Stefan Hartmann, Andreas Meister, Michael Schäfer, Stefan Turek (eds.)

International Workshop on Fluid-Structure Interaction

Theory, Numerics and Applications

kassel university press

Editors

Prof. Dr.-Ing. Stefan Hartmann
Division of Solid Mechanics
Institute of Applied Mechanics
Faculty of Mathematics/Computer Science and Mechanical Engineering
Clausthal University of Technology

Prof. Dr. rer. nat. Andreas Meister
Work Group of Analysis and Applied Mathematics
Department of Mathematics
University of Kassel

Prof. Dr. rer. nat. Michael Schäfer
Institute of Numerical Methods in Mechanical Engineering
Department of Mechanical Engineering
Technische Universität Darmstadt

Prof. Dr. rer. nat. Stefan Turek
Institute of Applied Mathematics
Faculty of Mathematics
Technische Universität Dortmund

Bibliographic information published by Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at http://dnb.d-nb.de.

ISBN print: 978-3-89958-666-4
ISBN online: 978-3-89958-667-1
URN: urn:nbn:de:0002-6673

Cover layout: Jörg Batschi Grafik Design, Kassel
Printed by: docupoint, Magedburg
Printed in Germany

Preface

Fluid-structure interactions (FSI), that means the interaction of some deformable structure with a surrounding or internal fluid flow, belong nowadays to the most important and challenging multi-physics problems which are aimed to treat by numerical simulations. Since such multi-physics problems, ranging from small-scale micro pumps over blood flow in arteries to objects in a wind tunnel up to huge buildings, involve more than one physical effect, the challenges with respect to mathematical modelling, numerical discretization and solution techniques and realization as software tools on modern computer architectures are still huge, even today, particularly if accurate, flexible and highly efficient simulation tools shall be realized.

Most of todays software packages for computational fluid dynamics (CFD) and computational structural mechanics (CSM) are already on a quite high level and allow, at least to some extent, the simulation of certain classes of FSI applications. However, although a lot of research has been invested in this challenging field, many of the key questions in FSI regarding accuracy, robustness, flexibility and efficiency of the developed simulation techniques have not been answered yet in a satisfying way: These key questions address, for instance, the appropriate coupling between structural and fluid parts, i.e. monolithic vs. partitioned coupling schemes, and the advantages and drawbacks of various discretization schemes used on the flow and on the structure side. Moreover, the reliability of the simulation results and underlying error estimators, or flexible data structures and geometry models for large geometrical or even topological changes, are still in the center of research activities. And, finally, the design of robust and efficient solvers for the resulting huge nonlinear systems and the realization of the numerical approaches as flexible software tools allowing techniques from high performance computing still have to be treated with very high priority.

This book contains the proceedings of a 3-day workshop on fluid-structure interactions held in Herrsching am Ammersee, Germany, in September 2008 which was organized by the Collaborative Research Center SFB TRR 30 *Process-integrated development of functional graded structures on the basis of thermo-mechanically coupled phenomena* and the Research Unit 493 *Fluid-Structure Interaction: Modelling, Simulation, Optimization,* both established by the Deutsche Forschungsgemeinschaft (DFG). Designed as a forum for presenting recent results as well as for exchanging ideas with leading international experts, this workshop consisted of 29 lectures on all kind of mathematical and computational aspects of fluid-structure interactions. The topics in this volume cover a broad spectrum of up-to-date FSI issues, ranging from more methodical aspects to applications and experimental settings for validation and evaluation.

We would like to thank the “Gesellschaft zur Förderung des technischen und kaufmännischen Führungsnachwuchses in Nordhessen Informatik (GFF-I)” for financial support, and Mrs. Ilona Blaschek for her enormous engagement and administrative work. Finally, we are grateful for the Deutsche Forschungsgemeinschaft (DFG). Without this financial support, neither many of the results presented in this book nor the book itself would have been possible.

Stefan Hartmann
Andreas Meister
Michael Schäfer
Stefan Turek

Clausthal, Kassel, Darmstadt, Dortmund March 2009

Content

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 7– 20
Herrsching am Ammersee, 29.9.-1.10.2008

Shape Optimization with Higher-Order Surfaces in Consideration of Fluid-Structure Interaction

G. Becker, U. Falk, M. Schäfer

This paper proposes the first step towards an efficient and entirely integrated shape optimization of fluid-structure interaction problems. The presented approach utilizes NURBS surfaces to approximate and alter grid based optimization surfaces. Due to the usage of the already availabe FSI grid generation routines it is possible to handle even large grid deformations. Furthermore, this approach allows the user to address the optimization surfaces easily via boundary condition. The design variables can be set very intuitively. The presented implementation works with any derivative free optimization tool.

1 Introduction

Fluid-structure interactions (FSI) arise in many disciplines and applications, e.g. elastic artery modeling, airfoil flutter or wind loads on structures (Mittal and Tezduyar, 1995; Kalro and Tezduyar, 2000; Tezduyar and Osawa, 2001). Therefore, the significance of efficient numerical methods to solve these problems has increased steadily. Thus, the demand for shape optimization has arisen, e.g. the drag and lift force optimization of airfoils. Throughout the last years, researchers have improved the numerical methods concerning fluid-structure interactions. Efficient codes are available now. Furthermore, in the field of structural mechanics shape optimization is already commercially receivable and fluid mechanics researchers make good progress. However, simulations combining fluid-structure interactions and shape optimizations have not been deeply investigated yet (Mohammadi and Pironneau, 2001).

Therefore, it is aspired to establish shape optimizations within fluid-structure interaction applications. Since the optimal shape is calculated, numerical shape optimization of coupled problems may reduce the costs of time consuming experiments (Harth and Schäfer, 2004).

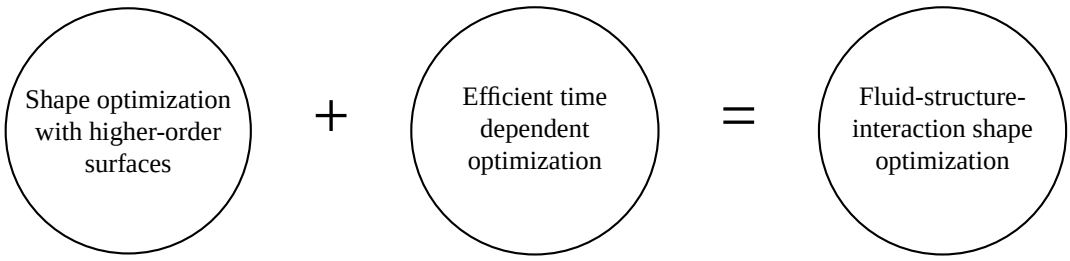


Figure 1: Classification of this paper's content within the context of FSI optimization

This paper proposes a new deformation approach for optimizing shapes. It allows a straightforward application towards coupled problems, i.e. the coupled code's routines for grid deformation and generation are utilized for the shape movement within the optimization process as well. The new method is implemented within the fluid region and works aside the fluid-structure coupling surfaces. Because of the time dependence in fluid-structure interactions, an efficient time dependent optimization approach needs to be considered. This paper handles the first step towards an efficient FSI optimization (see Fig. 1).

1.1 Shape optimization approaches

Samareh (1999) gives a general review of shape parameterization techniques for multidisciplinary optimization based on the work of Haftka and Grandhi (1986). In addition to the obvious discrete approach using the surface grid points as design variables, there are three general shape optimization techniques. The main intention of these methods is the reduction of design variables to accelerate the optimization.

The *basis vector approach* uses a small number of trial designs. These are weighted by design variables and added to the original design to achieve an optimal shape (Pickett et al., 1973). The trial designs are generated by a variety of methods, e.g. free form deformation (Sederberg and Parry, 1986; Barr, 1984). Drawback of this method is that the shape is restricted only to the possible combinations of the trial shapes.

CAD-based methods utilize geometric shape design parameters defined in a CAD model as design variables, e.g. the height of an object. First the shape is altered in an external CAD application. If necessary a new grid is generated in a meshing tool. A new objective function value is evaluated using this grid within a FEM- or FVM-solver (Olhoff et al., 1991; Grindeanu et al., 2002). The disadvantage of this approach is the time costly serial usage of different exteral programs.

The *polynomial and spline approach* utilizes polynomial functions to represent the shape being optimized. B-spline functions allow an easy and local control of the object's shape by manipulating the control points, which serve as design variables. Furthermore, B-spline surfaces can represent any shape and allows the approximation of given sets of grid points (Piegl, 1991; Tiller, 1983). It is possible to completely integrate them into the availabe FSI grid deformation routines. Hence, no external programms are necessary. Thus, nonuniform rational B-spline (NURBS) surfaces are chosen in this paper (Bazilevs et al., 2006, 2008).

1.2 Development environment

The applied code solves fluid-structure interactions via an implicit partitioned approach (Schäfer et al., 2006). It is based on FASTEST, a parallel multigrid flow solver, utilizing an entirely conservative finite-volume method to solve the incompressible Navier-Stokes equation on a non-staggered, block structured and cell centered grid (Durst and Schäfer, 1996). Furthermore, FEAP is used as a finite-element based structural solver (Taylor, 2003). The coupling interface is realized via MpCCI, which sends forces of the flow region to the structural solver and returns resulting deformations (MpCCI, 2006). For these, an efficient block-based grid deformation tool is implemented for the fluid region, allowing large grid movements by algebraic and elliptic mesh generation techniques (Schäfer et al., 2006).

2 Mathematical background

In this section the mathematical background of nonuniform rational B-spline surfaces (NURBS surfaces) and the approximation of a set of points with such surfaces is briefly reviewed.

According to Piegl (1991), NURBS curves are vector-valued piecewise rational polynomial functions (Piegl and Tiller, 1987; Rogers and Adams, 1976; Tiller, 1983). NURBS are used within this work to calculate an approximation of given surface grid points. Those represent the shape of the objects we like to optimize with respect to minimize a defined objective function. NURBS surfaces are a powerful technique to change the grid point position and therefore the shape of the objects by adjusting the control points. In contrast to the amount of grid points the amount of control points is small. Starting from splines the definition of NURBS surfaces is shown step by step. In addition the key benefits regarding the application of NURBS surfaces in shape optimization are stated.

A linear regression problem has to be solved in order to fit a NURBS surface to the provided surface grid points and gives us a first set of control points. Therefore, different methods of solving such overdetermined linear equation systems are given.

2.1 NURBS surfaces

2.1.1 Splines

According to Piegl and Tiller (1996), a spline $\mathcal{C} : [0,1] \to \mathbb{R}^n$ is a piecewise defined polynomial function of order $\tilde{p} \in \mathbb{N}$. Therefore, parameter values (knots)

$$\tilde{\boldsymbol{\varphi}} = (\tilde{\varphi}_1, \ldots, \tilde{\varphi}_{\tilde{m}}), \text{ where } \tilde{\varphi}_i \in [0,1],\ i \in \{1, \ldots, \tilde{m}\}, \tag{1}$$

need to be defined. They represent the endpoints of the piecewise defined polynomial functions. Let $\tilde{p} \geq 2$ and $\mathcal{C}_i := \mathcal{C}_{|_{[\tilde{\varphi}_i, \tilde{\varphi}_{i+1}]}} \in C^{\tilde{p}-2}[0,1]$. Then for $j = \{0, 1, \ldots, \tilde{p}-2\}$ we demand for the jth derivative

$$\mathcal{C}_i^{(j)}(\tilde{\varphi}_{i+1}) = \mathcal{C}_{i+1}^{(j)}(\tilde{\varphi}_{i+1}),\ \forall i \in \{1, \ldots, \tilde{m}-2\}. \tag{2}$$

Regarding the approximation of data points, splines offer the following advantages compared to a single polynomial:

- Due to the low degree of the piecewise defined spline-function, splines are easy to process and numerically stable even with a high number of interpolation points (Piegl and Tiller, 1996).
- Splines are well-suited to shape any design. With control points the shape can be locally manipulated (Piegl, 1991).

2.1.2 B-spline-surfaces

In contrast to general splines, B-splines are constructed using a particular class of polynomial functions. These functions are called B-spline basis functions (Tiller, 1983).

The finite polynomial B-spline surface $\mathcal{S} : [0,1] \times [0,1] \to \mathbb{R}^3$ is a product of two piecewise defined polynomial functions of order p, q respectively. Therefore, we define two knot vectors $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_{\hat{n}+p})$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_{\hat{m}+q})$ by

$$\xi_i = \begin{cases} 0 & \text{if} \quad i < p \\ \frac{i-p}{\hat{n}-p} & \text{if} \quad p \leq i \leq \hat{n} \\ 1 & \text{if} \quad i > \hat{n} \end{cases} \qquad \text{for} \quad i = 1, \ldots, \hat{n}+p \tag{3}$$

and $\boldsymbol{\nu}$ analogously.

According to Zeid (2005) the above definition generates a uniform knot vector, which leads to a *uniform* B-spline surface. Furthermore, the end knots are repeated p (resp. q) times. As a consequence the surface's corner points coincide with the corresponding control points. This allows a direct manipulation. A nonuniform knot vector would lead to a *nonuniform* B-spline surface (see Section 3.1.3).

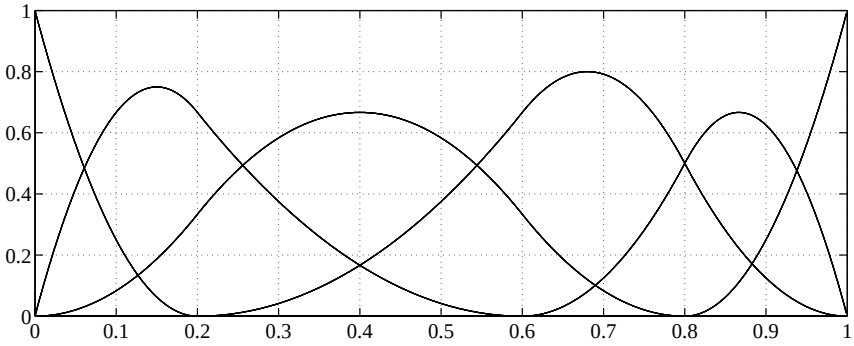


Figure 2: B-spline basis functions $\mathcal{N}_{i\in\{1,\ldots,6\}}^{p=3}$ using the nonuniform knot vector $\boldsymbol{\xi} = (0, 0, 0, 0.2, 0.6, 0.8, 1, 1, 1)$.

The B-Spline basis functions $\mathcal{N}^p, \mathcal{N}^q : [0,1] \to \mathbb{R}$, with $\mathcal{N}_i^p := \mathcal{N}^p_{|_{[\xi_i,\xi_{i+1}]}}$ ($\mathcal{N}_j^q$ using $\boldsymbol{\nu}$ analogously) are computed piecewisely and recursively with the Cox-deBoor recurrence (see De Boor (2001) and Fig. 2). Note, that we follow the convention of Farin et al. (2002): $0/0 := 0$.

$$\mathcal{N}_i^1(u) = \begin{cases} 1 & \text{if } \xi_i \leq u < \xi_{i+1} \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathcal{N}_i^p(u) = \frac{u - \xi_i}{\xi_{i+p} - \xi_i}\mathcal{N}_i^{p-1}(u) + \frac{\xi_{i+p+1} - u}{\xi_{i+p+1} - \xi_{i+1}}\mathcal{N}_{i+1}^{p-1}(u), \qquad (\mathcal{N}_j^q(v) \text{ analogously}). \tag{4}$$

Therefore, following Piegl and Tiller (1996),

$$\mathcal{S}(u,v) = \sum_{i=1}^{\hat{n}}\sum_{j=1}^{\hat{m}} \mathcal{N}_i^p(u)\mathcal{N}_j^q(v)\mathbf{P}_{i,j}, \quad u,v \in [0,1], \tag{5}$$

where $\mathbf{P}_{i,j} \in \mathbb{R}^3$ denotes the (i,j)th entry of $\mathbf{P} \in \mathbb{R}^{\hat{n}} \times \mathbb{R}^{\hat{m}} \times \mathbb{R}^3$.

2.1.3 Nonuniform rational B-spline surfaces (NURBS surfaces)

A NURBS surface is a B-spline surface with a nonuniform knot vector (Tiller, 1983) and is defined as follows:

$$\mathcal{S}(u,v) = \frac{\sum_{i=1}^{\hat{n}}\sum_{j=1}^{\hat{m}} w_{i,j}\mathbf{P}_{i,j}\mathcal{N}_i^p(u)\mathcal{N}_j^q(v)}{\sum_{r=1}^{\hat{n}}\sum_{s=1}^{\hat{m}} w_{r,s}\mathcal{N}_r^p(u)\mathcal{N}_s^q(v)}, \quad u,v \in [0,1] \tag{6}$$

with weights $\mathbf{w} = (w_{i,j})_{i,j}$ and the nonuniform knot vectors $\boldsymbol{\xi}, \boldsymbol{\nu}$. The computation of the nonuniform knot vectors can be found in Section 3.1.3.

Let

$$\mathcal{R}_{i,j}^{p;q}(u,v) = \frac{w_{i,j}\mathcal{N}_i^p(u)\mathcal{N}_j^q(v)}{\sum_{r=1}^{\hat{n}}\sum_{s=1}^{\hat{m}} w_{r,s}\mathcal{N}_r^p(u)\mathcal{N}_s^q(v)} \tag{7}$$

be a bivariate basis function (Piegl, 1991), thus

$$\mathcal{S}(u,v) = \sum_{i=1}^{\hat{n}}\sum_{j=1}^{\hat{m}} \mathbf{P}_{i,j}\mathcal{R}_{i,j}^{p;q}(u,v). \tag{8}$$

Piegl (1991) stated the following important properties of bivariate basis functions:

- Locality:

$$\mathcal{R}_{i,j}^{p;q}(u,v) = 0 \text{ if } u \notin [\xi_i, \xi_{i+p}] \text{ and } v \notin [\nu_j, \nu_{j+q}] \tag{9}$$

- Partition of unity:

$$\sum_{i,j} \mathcal{R}_{i,j}^{p;q}(u,v) = 1 \ \ \forall\, u,v \in [0,1] \tag{10}$$

- Differentiability:

$$\mathcal{R}_{i,j}^{p;q}(\xi_i, \nu_j) \in C^{p-k;q-l}, \text{ if } \quad \xi_i = \xi_{i+\hat{r}} \text{ for } \hat{r} \in \{1,\dots,k\};\ \nu_j = \nu_{j+\hat{s}} \text{ for } \hat{s} \in \{1,\dots,l\}, \tag{11}$$

  where k, l denote the amount of recurrences of the same knot value.

As a result we obtain the following properties for NURBS surfaces (Piegl, 1991):

- The surface is only affected in a p knot span (or q knot span respectively) if a control point is moved or a weight is changed.
- If all $w_{i,j} = 1$, Eq. (6) simplifies to Eq. (5). In particular, B-spline surfaces are a special case of nonuniform rational B-spline (NURBS) surfaces.

2.2 Approximation of surface grids with NURBS

Throughout this paper surface fitting is defined as the construction of a NURBS surface which fits a set of geometric data, such as the grid points representing the shape of an object during a numerical flow simulation.

The input of the fitting process consists of the geometric grid point data $\mathbf{X} \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^3$, the knot vectors $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$, the weights $\mathbf{w}$, the number of control points in u- and v-direction $\hat{n}$ and $\hat{m}$ as well as the orders p and q of the B-spline basis functions. $\mathbf{X}$ is specified by the problem; $\boldsymbol{\xi}, \boldsymbol{\nu}, \hat{n}, \hat{m}, p, q$ and $\mathbf{w}$ have to be preselected. The resulting output is a set of control points $\mathbf{P} \in \mathbb{R}^{\hat{n}} \times \mathbb{R}^{\hat{m}} \times \mathbb{R}^3$. According to Piegl and Tiller (1996) very little has been published on setting the weights in the fitting process, therefore we set

$$\mathbf{w} = \underline{1} \tag{12}$$

in this paper.

Given a set of data points $\mathbf{X}$ and weights $\mathbf{w} = \underline{1}$, a nonuniform B-spline surface approximating the given grid points has to be constructed. First, $\mathbf{X}$ has to be parameterized to receive the discrete points u_k, v_l (see Section 3.1.2). Thus, the following system of linear equations needs to be solved:

$$\mathcal{S}(u_k, v_l) = \sum_{i=1}^{\hat{n}} \sum_{j=1}^{\hat{m}} \mathcal{N}_i^p(u_k) \mathcal{N}_j^q(v_l) \mathbf{P}_{i,j}. \tag{13}$$

2.3 Linear regression problem

To finish the data fitting process presented in the previous section, Eq. (13) needs to be solved. However, there are notably less control points compared to surface grid points and hence significantly less unknowns compared to the provided equations. Thus, an overdetermined linear system has to be solved which is also known as a regression problem. For the examination of possible strategies (see also Duff and Reid (1976)) to solve such problems and to simplify matters this chapter considers the following overdetermined linear system of equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \text{ with } \mathbf{A} \in \mathbb{R}^m \times \mathbb{R}^n, \ \mathbf{x} \in \mathbb{R}^n, \ \mathbf{b} \in \mathbb{R}^m, \ m > n. \tag{14}$$

According to Peters and Wilkinson (1970) and using the simplified Eq. (14), a linear regression problem can be defined as follows:

$$\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2 = \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2. \tag{15}$$

In general $\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2 = r, \ r \geq 0$. The resulting value r is called the residuum of $\mathbf{x}^*$.

From a geometrical point of view the optimal solution of the minimization problem is orthogonal to the image space $\text{Im}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\}$, i.e.

$$\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2 = \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \Leftrightarrow \mathbf{A}\mathbf{x}^* - \mathbf{b} \perp \text{Im}(\mathbf{A}) \tag{16}$$

The following methods are suitable to solve such overdetermined linear system, i.e. the minimization problem defined in Eq. (15):

- Normal equations

- Singular value decomposition
- QR-decomposition

According to Dahmen and Reusken (2008) the major drawback of the normal equations approach is the squared condition number. This leads to a high sensitivity in respect to truncation errors during the utilized *Cholesky* decomposition and can result in an unstable solution process. Whereas the singular value decomposition needs considerably more time to solve a linear regression problem than the normal equations or the QR-decomposition. Furthermore, the QR-decomposition offers a stable algorithm. Thus, we choose the QR-decomposition to solve our overdetermined linear system of equations.

2.3.1 QR-decomposition

The QR-decomposition is based on the following property of orthogonal matrices:

$$\|\mathbf{QA}\|_2 = \|\mathbf{A}\|_2, \text{ since } \|\mathbf{Q}\|_2 = 1 \tag{17}$$

Thus, the linear regression problem as defined in Eq. (15) can be rewritten as:

$$\min_{x \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2 = \|\mathbf{QAx}^* - \mathbf{Qb}\|_2, \quad \text{with} \quad \mathbf{Q} \in \mathbb{R}^{m \times m}\ \mathbf{A} \in \mathbb{R}^{m \times n},\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{b} \in \mathbb{R}^m \tag{18}$$

where $\mathbf{Q}$ has to be chosen such that

$$\mathbf{QA} = \mathbf{R} := \begin{pmatrix} \tilde{\mathbf{R}} \\ 0 \end{pmatrix} \quad \text{with} \quad \tilde{\mathbf{R}} \in \mathbb{R}^n \times \mathbb{R}^n \tag{19}$$

and

$$\mathbf{Qb} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \quad \text{with} \quad \mathbf{b}_1 \in \mathbb{R}^n,\ \mathbf{b}_2 \in \mathbb{R}^{m-n} \tag{20}$$

where $\tilde{\mathbf{R}}$ is an upper triangular matrix. While $\mathbf{Q}$ having the characteristics of Eq. (19) and (20) the approximate solution of the overdetermined linear equation system can be achieved by

$$\mathbf{x}^* = \tilde{\mathbf{R}}^{-1}\mathbf{b}_1. \tag{21}$$

$\mathbf{Q}$ is the product of n orthogonal matrices $\mathbf{Q}_i$. In order to generate the upper triangular matrix $\tilde{\mathbf{R}}$, n matrices $\mathbf{Q}_i$ have to be chosen such that the multiplication by the matrix $\mathbf{A}$ results in zeros below the main diagonal in the ith column of matrix $\mathbf{A}$.

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \xrightarrow{\mathbf{Q_1 A}} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \xrightarrow{\mathbf{Q_2 Q_1 A}} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \xrightarrow{\mathbf{Q_3 Q_2 Q_1 A}} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 3: Creation of an upper triangular matrix with orthogonal matrices $\mathbf{Q}_i$

To generate the orthogonal matrices $\mathbf{Q}_i$ we apply the Householder reflection which is the most effective method according to Bärwolff (2007).

The Householder reflection uses a $(n-1)$ dimensional hyperplane through the origin and reflects $\mathbf{Q}_i$ at this plane. The corresponding hyperplane of $\mathbf{Q}_i$ is defined by the normal vector $\mathfrak{h}^i$ (Householder vector).

$$\mathfrak{h}^i = (\mathfrak{h}_1^i, \ldots, \mathfrak{h}_n^i)^T \quad \in \mathbb{R}^n, \quad \mathfrak{h}^i \neq 0 \tag{22}$$

Using the Householder-vector $\mathfrak{h}^i$ the orthogonal matrices $\mathbf{Q}_i$ can be defined as:

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{F} \end{bmatrix} \quad \text{with unity matrix } \mathbf{I} \in \mathbb{R}^{i-1} \times \mathbb{R}^{i-1},\ \mathbf{F} \in \mathbb{R}^{m-i+1} \times \mathbb{R}^{m-i+1}, \tag{23}$$

where

$$\mathbf{F} = \mathbf{I} - 2\frac{\mathfrak{h}^i(\mathfrak{h}^i)^T}{(\mathfrak{h}^i)^T\mathfrak{h}^i}. \tag{24}$$

In order to be able to reduce the matrix $\mathbf{A}$ to the upper triangular matrix $\tilde{\mathbf{R}}$ by multiplying it with the orthogonal matrices $\mathbf{Q}_i$ the components of the Householder-vector $\mathfrak{h}$ have to be selected as

$$\mathfrak{h}^i = \boldsymbol{\alpha}^i + \text{sign}(\alpha_i^i)\|\boldsymbol{\alpha}^i\|_2\mathbf{e}^i, \tag{25}$$

where $\boldsymbol{\alpha}^i$ is the ith column of the matrix $\mathbf{A}$ and $\mathbf{e}^i$ is the unit vector of the ith direction. The linear regression problem can be solved by QR-decomposition using the following algorithm (Dahmen and Reusken, 2008):

- QR-decomposition via Householder-reflection (complexity mn^2)
- Calculation of $\mathbf{Qb}$ (complexity $2mn$)
- Solving $\tilde{\mathbf{R}}\mathbf{x} = \mathbf{b}_1$ (complexity $\frac{1}{2}n^2$)

According to Dahmen and Reusken (2008) a unique solution of such an overdetermined linear equation system with the previous algorithm is possible if $\mathbf{A}$ has a full rank, i.e.

$$\text{rank}(\mathbf{A}) = n. \tag{26}$$

Schoenberg and Whitney (1953) stated that condition (26) is allways fulfilled if at least one of n parameterized gridpoints u_k exists between every pair of subsequent knots, i.e.

$$\xi_i < u_k < \xi_{i+1}, \quad i = 1, \ldots, \hat{n} + p, \quad k = 1, \ldots, n, \qquad (\boldsymbol{\nu}, v, m, \hat{m}, q \text{ analogously}) \tag{27}$$

For that reason Eq. (37) is chosen to generate the knot vectors $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$.

3 Implementation

In this section the workflow of *FASTEST ShapeOpt* using a derivative free optimizer is presented. The optimizer is treated as a black box, i.e. it is exchangeable by any derivative free optimization tool. For computing the example in section 4 the free available tool *DFO* by Conn et al. (1997) was implemented into the *FASTEST ShapeOpt* environment.

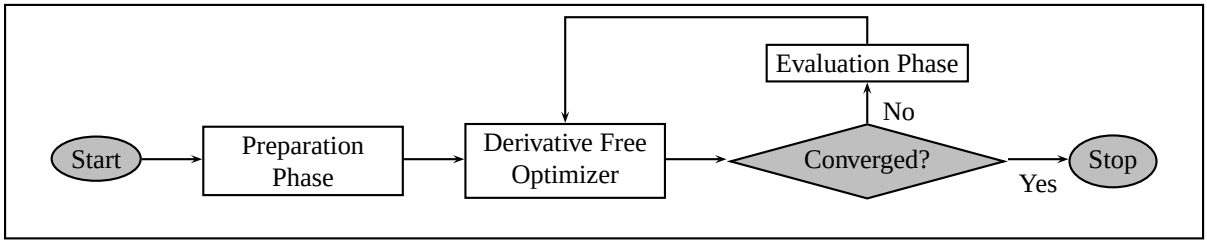


Figure 4: General workflow of *FASTEST ShapeOpt*, utilizing a derivative free optimizer as black box

FASTEST ShapeOpt is split into three main parts (see Figure 4). The preparation phase, where the optimization surface is approximated with a NURBS surface. Basis for the approximation is the blockstructured grid of the *FASTEST Flow solver*. Furthermore, the initial set of design variables is defined. A derivative free optimizer typically interpolates the objective function by a polynomial of fixed degree and computes the optimum based on it. Hence, this kind of optimizers need evaluations to enhance the approximation of the objective function with further values. A new set of design variables is passed to the evaluation phase in order to receive a new point of the objective function. Within this evaluation phase, a new surface shape based on the new set of design variables is calculated, the grid is updated and new flow properties are calculated to compute the new value of the objective function.

3.1 Preperation Phase

The preparation phase is necessary to approximate a given surface grid with a NURBS surface. Afterwards the surface's control points can be used as design variables (see section 3.2). In order to use this part of *FASTEST ShapeOpt*, a blockstructured grid of the flow domain needs to be created and the optimization surfaces are marked with a special optimization boundary condition. This boundary condition will be reassigned as "wall" after the surface grid points are stored. The preparation phase can be subdivided into the parts described in the following.

3.1.1 Grid data

First, we need to find all initial surface grid points lying on an optimization surface and store them into a tensor

$$\mathbf{X}_d \in \mathbb{R}^{n_d} \times \mathbb{R}^{m_d} \times \mathbb{R}^3,\ d \in \{1, \dots, \kappa\}, \tag{28}$$

where κ denotes the number of optimization surfaces, n_d and m_d are the number of grid points in the two spatial edge directions of each surface. Now the boundary condition "optimization surface" can be reassigned as "wall".

Henceforth we omit the surface number d to simplify matters and present all equations for a single optimization surface.

3.1.2 Parameterization

The next step is to parameterize $\mathbf{X} \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^3$ with $u_k, v_l \in [0, 1]$. We use the following parameterization method based on Ma and Kruth (1998). Let

$$\hat{u}_{1,l} = 0, \quad \hat{u}_{k,l} = \frac{\sum_{i=1}^{k-1} \left(\|\mathbf{X}_{i+1,l} - \mathbf{X}_{i,l}\|\right)^e}{\sum_{i=1}^{n-1} \left(\|\mathbf{X}_{i+1,l} - \mathbf{X}_{i,l}\|\right)^e}, \quad k \in \{2, \dots, n\},\ l \in \{1, \dots, m\} \tag{29}$$

and

$$\hat{v}_{k,1} = 0, \quad \hat{v}_{k,l} = \frac{\sum_{j=1}^{l-1} \left(\|\mathbf{X}_{k,j+1} - \mathbf{X}_{k,j}\|\right)^e}{\sum_{j=1}^{m-1} \left(\|\mathbf{X}_{k,j+1} - \mathbf{X}_{k,j}\|\right)^e}, \quad k \in \{1, \dots, n\},\ l \in \{2, \dots, m\}. \tag{30}$$

Now we can derive $\mathbf{u} = (u_1, \dots, u_n)$ and $\mathbf{v} = (v_1, \dots, v_m)$ with

$$u_1 = 0, \quad u_k = \frac{1}{m} \sum_{j=1}^{m} \hat{u}_{k,j}, \quad k \in \{2, \dots, n\} \tag{31}$$

and

$$v_1 = 0, \quad v_l = \frac{1}{n} \sum_{i=1}^{n} \hat{v}_{i,l}, \quad l \in \{2, \dots, m\}. \tag{32}$$

The parameter $e \in [0, 1]$, introduced by Jung and Kim (2000) offers the possibilty to influence the type of parameterization. For $e = 0$, a uniform parameterization, for $e = 1$, a chordlength parameterization and for $e = 0.5$, a centripedal parameterization is performed. However, e can be chosen within $[0, 1]$, dependent on the setting.

3.1.3 Knot vector

In order to calculate the knot vectors, several boundary conditions have to be considered. To obtain a unique set of control points, it is necessary to fulfill condition (26). Hence, at least one grid point needs to be located between

every pair of subsequent knots (Eq. (27) and see also Section 2.3.1). Furthermore, the grid density should be considered and the corner points of the NURBS surface should coincide with the corresponding control points.

Therefore, the following algorithm was developed: Let us consider the discrete parameters $u_k,\ k \in \{1,\dots,n\}$ and $v_l,\ l \in \{1,\dots,m\}$ as a data basis to compute the knot vectors $\boldsymbol{\xi}, \boldsymbol{\nu}$ using

$$\begin{aligned}\boldsymbol{\xi} &= (\underbrace{0,\dots,0}_{p}, \xi_{p+1},\dots,\xi_{\hat{n}}, \underbrace{1,\dots,1}_{p})\\ \boldsymbol{\nu} &= (\underbrace{0,\dots,0}_{q}, \nu_{q+1},\dots,\nu_{\hat{m}}, \underbrace{1,\dots,1}_{q})\end{aligned} \tag{33}$$

with length $p+\hat{n}$ and $q+\hat{m}$, respectively. Recall that $\hat{n}, \hat{m}$ denote the numbers of control points and $p, q \in \mathbb{N}$ the orders of the B-spline basis functions in each boundary direction. Additional conditions are $n > \hat{n} - p + 1$ and $m > \hat{m} - q + 1$, to assure having a grid point between two subsequent knots.

For the purpose of simplification the following algorithm is presented for the knot vector $\boldsymbol{\xi}$ ($\boldsymbol{\nu}$ analogously).

As we can see in Eq. (33), we need to calulate $(\hat{n} - p)$ entries for this vector. Therefore, we split all entries of vector $\mathbf{u} = (u_1,\dots,u_k,\dots,u_n)$ into $(\hat{n} - p)$ intervalls. To compute the interval length we introduce an integer G and the remainder R:

$$G = \left\lfloor \frac{n}{\hat{n} - p + 1} \right\rfloor \tag{34}$$

$$R = n \bmod (\hat{n} - p + 1) \tag{35}$$

We set the interval length L of the $\hat{n} - p$ intervals to

$$L = 2G + R. \tag{36}$$

Now the knot vector entries can be computed by

$$\xi_{p+h+1} = \frac{1}{2G+R} \sum_{i=hG+1}^{hG+L} u_k, \quad h \in \{0,1,\dots,\hat{n}-p-1\}. \tag{37}$$

3.1.4 Basis functions

With known knot vectors $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$, as well as all discrete grid points $u_k,\ k = 1,\dots,n$ and $v_l,\ l = 1,\dots,m$ we are able to calculate the B-spline basis functions by Eq. (4) and store them into the following matrices:

$$\mathbf{N}^p \in \mathbb{R}^{\hat{n}} \times \mathbb{R}^n, \text{ in } u\text{-direction and} \tag{38}$$

$$\mathbf{N}^q \in \mathbb{R}^{\hat{m}} \times \mathbb{R}^m, \text{ in } v\text{-direction.} \tag{39}$$

3.1.5 Solving the linear regression problem

Finally, using nonuniform B-spline surfaces (Eq. (13)) and grid points $\mathbf{X}_\gamma \in \mathbb{R}^n \times \mathbb{R}^m$ we have to solve the overdetermined linear equation system

$$\mathbf{X}_\gamma = (\mathbf{N}^p)^T\, \mathbf{P}_\gamma\, \mathbf{N}^q, \tag{40}$$

where $\gamma \in \{1,2,3\}$ denotes the directions in $\mathbb{R}^3$. For solving Eq. (40) we use the above presented QR-decomposition algorithm (Section 2.3.1) and obtain the initial set of control points $\mathbf{P}_\gamma \in \mathbb{R}^{\hat{n}} \times \mathbb{R}^{\hat{m}}$.

3.2 Initialize design variables

To finish the preparation phase a relation between the set of control points $\mathbf{P}^{(g)}$, where $\mathbf{P}^{(g)}_{i,j} \in \mathbb{R}^3$ denotes the (i,j)th entry at the gth evaluation call, and the following set of design variables has to be established.

$$\boldsymbol{\alpha}^{(g)} \in \mathbb{R}^M, \quad \text{subject to } L_\tau \le \alpha^{(g)}_\tau \le U_\tau,\ \tau \in \{1,\dots,M\}, \text{ and } \boldsymbol{\alpha} = (\alpha_\tau)_\tau \tag{41}$$

where $M \in \mathbb{N}$ denotes the number of design variables. The initial set of design variables is always defined as

$$\boldsymbol{\alpha}^0 = 0. \tag{42}$$

The control points of the gth evaluation call are derived as follows:

$$\mathbf{P}_{i,j}^{(g)} = \begin{pmatrix} a_{i,j}^{(g)} \\ b_{i,j}^{(g)} \\ c_{i,j}^{(g)} \end{pmatrix} + \mathbf{P}_{i,j}^{(g-1)}. \tag{43}$$

For each pair (i, j) we distinguish four cases for the usage of the control points:

- Case I: Fixed control points

$$a_{i,j}^{(g)} = b_{i,j}^{(g)} = c_{i,j}^{(g)} = 0. \tag{44}$$

- Case II: One design variable per control point utilizing a direction vector $\mathbf{s}^\tau = (s_i^\tau)_{i\in\{1,2,3\}}$

$$\begin{aligned} a_{i,j}^{(g)} &= s_1^\tau \alpha_\tau^{(g)}, \\ b_{i,j}^{(g)} &= s_2^\tau \alpha_\tau^{(g)}, \\ c_{i,j}^{(g)} &= s_3^\tau \alpha_\tau^{(g)}, \\ \|\mathbf{s}^\tau\|_2 &= 1. \end{aligned} \tag{45}$$

- Case III: Three design variables per control point in each spatial direction

$$\begin{aligned} a_{i,j}^{(g)} &= \alpha_\tau^{(g)}, \\ b_{i,j}^{(g)} &= \alpha_{\tau+1}^{(g)}, \\ c_{i,j}^{(g)} &= \alpha_{\tau+2}^{(g)}. \end{aligned} \tag{46}$$

- Case IV: The control point depends on another control point's design variable with a scaling factor $\mathbf{t}^\tau = (t_i^\tau)_{i\in\{1,2,3\}}$. This is an important feature to reduce the number of design variables, e.g. in symmetric cases.

$$\begin{aligned} a_{i,j}^{(g)} &= t_1^\tau a_{k,l}^{(g)},\ (i,j) \neq (k,l), \\ b_{i,j}^{(g)} &= t_2^\tau b_{k,l}^{(g)},\ (i,j) \neq (k,l), \\ c_{i,j}^{(g)} &= t_3^\tau c_{k,l}^{(g)},\ (i,j) \neq (k,l). \end{aligned} \tag{47}$$

The above definitions allow the user to connect the design variables intuitively with the control points. In this approach, the design variable's value $\alpha_\tau^{(g)} \in [L_\tau, U_\tau]$ represents the displacement along either a direction vector (case II) or a single direction in space (case III).

3.3 Evaluation Phase

The functions we intend to optimize frequently appear to be either unknown or not differentiable. Hence, we can only use the class of derivative free optimizers. These interpolate the objective function with a lower degree polynomial using known values of $\mathcal{F}(\boldsymbol{\alpha}^{(g)})$. Thus, the derivative is known and the optimizer can use standard optimization methods to identify the optimum. In order to compute such an interpolation, it is necessary to calculate the unknown relation between the design variables $\boldsymbol{\alpha}^{(g)}$ and the objective function $\mathcal{F}$ during an evaluation phase. Hence, the objective function

$$\mathcal{F}(f(\boldsymbol{\alpha}^{(g)})) : \Theta \to \mathbb{R}, \tag{48}$$

where Θ denotes an arbitrary set of parameters, is defined implicitly by a function

$$f(\boldsymbol{\alpha}^{(g)}) : \mathbb{R}^M \to \Theta, \tag{49}$$

which is computed by the *FASTEST Flow Solver*.

The evaluation phase is divided into three sections. First, a new grid is generated using a new set of design variables. Afterwards the flow problem is solved and finally a new value of the objective function is calculated and returned to the optimizer.

3.3.1 Grid generation

In the evaluation phase a new set of control points $\mathbf{P}^{(g)} \in \mathbb{R}^{\hat{n}} \times \mathbb{R}^{\hat{m}} \times \mathbb{R}^3$ is calculated with Eq. (43) using a new set of design variables $\boldsymbol{\alpha}^{(g)}$ and the already calculated B-spline basis functions (see Section 3.1.4) as input. Now, a new set of surface grid points $\mathbf{X}^{(g)} \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^3$ can be calculated using the NURBS equation (40), where $X^{(g)}_{i,j,k}$ denotes the (i,j,k)th entry of the tensor $\mathbf{X}^{(g)}$.

Note that this method generates the new surface grid $\mathbf{X}^{(g)}$ only. To move the remaining grid according to the displaced optimization surface, grid generation techniques already used for FSI grid displacements are applied. Within the FSI approach introduced in Section 1.2, the FSI grid generation routine receives computed displacements of the coupling surface from the structural solver *FEAP* (Schäfer et al., 2006). To utilize this routine for the purpose of shape optimization we need to compute the displacements between $\mathbf{X}^{(g)}$ and the original optimization surface $\mathbf{X}^0$, i.e.

$$\Delta_{i,j,k} = \left| X^{(g)}_{i,j,k} - X^0_{i,j,k} \right|, \quad \forall\, i,j,k. \tag{50}$$

The grid generation routine uses $\boldsymbol{\Delta} = (\Delta_{i,j,k})_{i,j,k}$ to generate new grids within the optimization surface's neighboring blocks. This is performed either via linear or transfinite interpolation or via elliptic grid generation (Thompson and Soni, 1999).

3.3.2 Flow solver

In contrast to many other shape optimization approaches this implementation generates the grid internally. In particular, the grid's topology remains unchanged. Hence, all flow related values of the control volume's centers which have been calculated in a former evaluation call can be used as initial values for the next solver run. This allows fast convergence and supersedes new mapping or restarting. Thus, a new evaluation run in a figurative sense is the same as a new FSI coupling step. Nevertheless, the full functionality of the flow solver can be applied (see Section 1.2 as well as Durst and Schäfer (1996)).

3.3.3 Objective function

After obtaining all problem related variables at the control volume centers, the objective function can be calculated. For instance

$$\min_{\boldsymbol{\alpha}^{(g)}} \mathcal{F}(f(\boldsymbol{\alpha}^{(g)}, \mathbf{x})), \tag{51}$$

where f denotes a function dependent on the set of design variable $\boldsymbol{\alpha}^{(g)}$ and $\mathbf{x}$, a further set of variables. The resulting value of the objective function is passed back to the optimizer.

4 Example: 2D Flow over a hill

A simple test example was set up to validate the implementation using DFO as optimization tool (Conn et al., 1997). It is based on the optimizer test of Burkhardt and Peterson (1995) being slightly modified.

The basic setup is a 2D-channel with a small hill (see Figure 5) to derange the ideal laminar Hagen-Poiseuille flow. The Hagen-Poiseuille flow of this channel can be described analytically by

$$\hat{u}(y) = \frac{\overline{u} y (3h - y)}{(1.5h)^2}, \tag{52}$$

where $\overline{u}$ denotes the maximum velocity in x-direction and $3h$ denotes the height of the channel.

We optimize the hill's shape with one design variable $\alpha^{(g)}$ to receive an ideal Hagen-Poiseuille flow downstream the hill. For this case we know the optimum shape in advance: The hill's height has to be zero. The ideal Hagen-Poiseuille flow and the actual flow at (x_0, y_i) are shown in Figure 6(a).

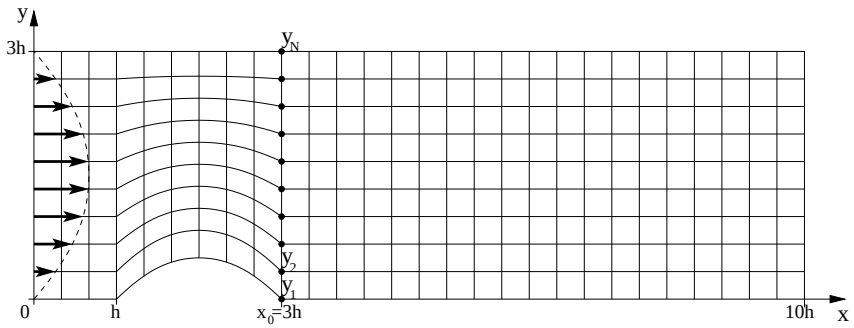


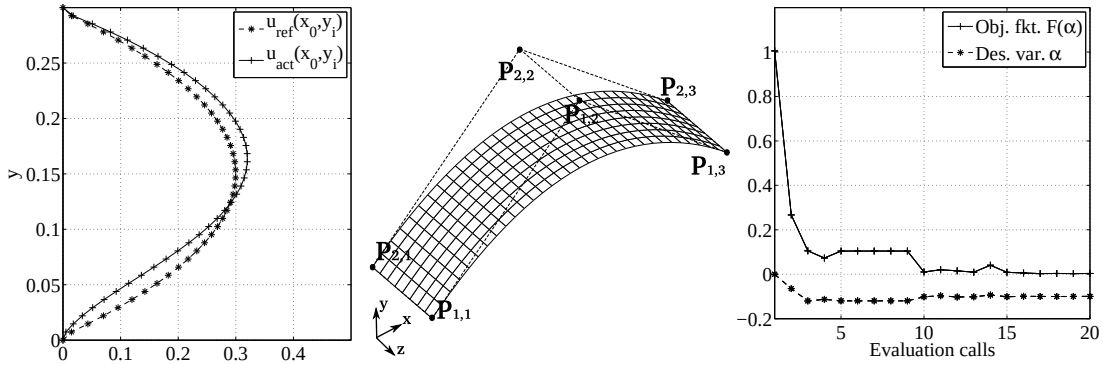
Figure 5: The flow problem's discretized domain

Thus, we define the objective function

$$\min_{\alpha^{(g)}} \mathcal{F}(u(\alpha^{(g)}, x, y), \hat{u}(y_i), x_0, \mathbf{y}) = \sum_{i=1}^{N} \left| u(\alpha^{(g)}, x_0, y_i) - \hat{u}(y_i) \right|, \quad \text{with } \mathbf{y} = (y_1, y_2, \ldots, y_N) \tag{53}$$

where $u(\alpha^{(g)}, x_0, y_i)$ denotes the current velocity in x-direction dependent on the shape of the flow domain (controlled by α), at the position $(x_0, y_i),\ i = 1, \ldots, N$ (see Figure 5).

Due to the 3D-based implementation of *FASTEST*, it is necessary to add a third direction in space to the flow domain. Note that we need to add several layers of control volumes in the new third direction to be able to use a geometric multigrid approach for solving. Hence, we need a NURBS surface instead of a NURBS spline to approximate the hill. In Figure 6(b) this surface is shown. It was approximated with $\hat{n}\hat{m}$ ($\hat{n} = 3$ and $\hat{m} = 2$) control points and orders $p = 3$ and $q = 2$.



(a) Analytic Hagen-Poiseille flow u_{ref} and the actual flow u_{act} at x_0

(b) Hill approximated with a NURBS surface

(c) Resulting values of the design variable α and the objective function $\mathcal{F}$

Figure 6: Flow profiles, optimization surface approximation and results

The *FASTEST flow solver* using a geometric multigrid on three grid levels is set to solve an incompressible, steady state and isotherm flow with Re $= 1$ and 40960 control volumes at the finest grid level. The grid deformation within the block above the hill is performed with a linear interpolation, while the remaining grid stays unchanged.

Furthermore, the relation between control points and design variables is set to

$$\mathbf{P}_{1,1}^{(g)} = \mathbf{P}_{2,1}^{(g)} = \mathbf{P}_{1,3}^{(g)} = \mathbf{P}_{2,3}^{(g)} = 0 \quad \text{(Case I)}, \tag{54}$$

$$\mathbf{P}_{1,2}^{(g)} = \begin{pmatrix} a_{1,2}^{(g)} \\ b_{1,2}^{(g)} \\ c_{1,2}^{(g)} \end{pmatrix} + \mathbf{P}_{1,2}^{(g-1)}, \quad \text{with } a_{1,2}^{(g)} = c_{1,2}^{(g)} = 0 \text{ and } b_{1,2}^{(g)} = \alpha^{(g)} \in [-0.12, 0] \quad \text{(Case II)}, \tag{55}$$

where $\alpha^{(0)} = 0$ and

$$\mathbf{P}_{2,2}^{(g)} = \begin{pmatrix} a_{1,2}^{(g)} \\ b_{1,2}^{(g)} \\ c_{1,2}^{(g)} \end{pmatrix} + \mathbf{P}_{2,2}^{(g-1)} \quad \text{(Case VI)}. \tag{56}$$

The initial position of control point $\mathbf{P}_{1,2}^{(0)}$ is $(0.2h, 0.1h, 0)$. After 20 evaluation runs, the optimizer returns a design variable with value $\alpha^{(20)} = 0.09997$, which computes a control point position $\mathbf{P}_{1,2}^{(20)} = (0.2h, 2.9h \times 10^{-6}, 0)$ applying Eq. (55). The objective function's value is expected to be zero. After 20 evaluation calls it is $\mathcal{F} = 27.1 \times 10^{-4}$ (see Fig. 6(c)). The result lies within the flow solver's and the optimizer's numerical accurracies.

5 Conclusions

The presented shape optimization approach utilizes NURBS surfaces to approximate and alter grid based shapes. This reduces the number of design variables and allows an accurate and reliable shape optimization. Furthermore, NURBS surfaces can represent any surface shape without restrictions.

In contrast to other shape optimization tools, this approach is fully integrated within the partitioned FSI environment. In particular, the FSI grid deformation routine is also applied to the shape optimization. It allows large deformations due to a blockwise generation of the grid. This is done either with a linear or transfinite interpolation or an elliptic grid generation. Furthermore, all flow related values can be reused as starting values for a new evaluation run. This allows fast grid deformation, flow solving and evaluating the objective function without using external software like CAD or meshing tools.

The complete functionality of the *FASTEST flow solver* can be used to solve laminar, turbulent (RANS, LES) and thermal flow problems. The implementation of *FASTEST ShapeOpt* allows the user to address the optimization surfaces easily via boundary condition. Furthermore, the design variables can be set intuitively. The presented implementation works with any derivative free optimization tool.

The next step is to add a sensitivity analysis and an efficient time dependent optimization strategy to this new approach to receive an efficient and accurate FSI shape optimization.

References

Barr, A. H.: Global and local deformations of solid primitives. *Computer Graphics*, 18, 3, (1984), 21–30.

Bärwolff, G.: *Numerik für Ingenieure, Physiker und Informatiker: Für Bachelor und Diplom*. Elsevier Spektrum Akad. Verl., München (2007).

Bazilevs, Y.; Calo, V.; Hughes, T.; Zhang, Y.: Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43, 1, (2008), 3–37.

Bazilevs, Y.; Calo, V.; Zhang, Y.; Hughes, T.: Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38, 4, (2006), 310–322.

Burkhardt, J.; Peterson, J.: Control of steady incompressible 2D channel flows. In: M. D. Gunzburger, ed., *Flow Control*, The IMA Volumes in Mathematics and its Applications, pages 111–126, Springer, New York (1995).

Conn, A. R.; Scheinberg, K.; Toint, P. L.: Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79, (1997), 397 – 414.

Dahmen, W.; Reusken, A.: *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg (2008).

De Boor, C.: *A practical guide to splines*. Springer (2001).

Duff, I. S.; Reid, J. K.: A comparison of some methods for the solution of sparse overdetermined systems of linear equations. *IMA Journal of Applied Mathematics*, 17, 3, (1976), 267–280.

Durst, F.; Schäfer, M.: A parallel block-structured multigrid method for the prediction of incompressible flows. *International Journal for Numerical Methods in Fluids*, 22, 6, (1996), 549–565.

Farin, G. E.; Hoschek, J.; Kim, M. S., eds.: *Handbook of computer aided geometric design*. Elsevier, Amsterdam (2002).

Grindeanu, I.; Kim, N. H.; Choi, K. K.; Chen, J.-S.: CAD-based shape optimization using a meshfree method. *Concurrent Engineering*, 10, 1, (2002), 55–66.

Haftka, R. T.; Grandhi, R. V.: Structural shape optimization–A survey. *Computer Methods in Applied Mechanics and Engineering*, 57, 1, (1986), 91–106.

Harth, Z.; Schäfer, M.: Numerical shape optimization for flow configurations. In: *Design Optimization International Conference*, Athens, Greece (2004).

Jung, H.; Kim, K.: A new parameterisation method for NURBS surface interpolation. *The International Journal of Advanced Manufacturing Technology*, 16, 11, (2000), 784–790.

Kalro, V.; Tezduyar, T. E.: A parallel 3d computational method for fluid-structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190, 3-4, (2000), 321 – 332.

Ma, W.; Kruth, J.-P.: NURBS curve and surface fitting for reverse engineering. *The International Journal of Advanced Manufacturing Technology*, 14, 12, (1998), 918–927.

Mittal, S.; Tezduyar, T. E.: Parallel finite element simulation of 3D incompressible flows: Fluid structure interactions. *International Journal for Numerical Methods in Fluids*, 21, 10, (1995), 933–953.

Mohammadi, B.; Pironneau, O.: *Applied shape optimization for fluids*. Oxford Science Publications, Oxford (2001).

MpCCI: *Mesh-based parallel code coupling interface*. Fraunhofer SCAI, 3.05 edn. (2006).

Olhoff, N.; Bendsøe, M. P.; Rasmussen, J.: On CAD-integrated structural topology and design optimization. *Computer Methods in Applied Mechanics and Engineering*, 89, 1-3, (1991), 259–279.

Peters, G.; Wilkinson, J. H.: The least squares problem and pseudo-inverses. *The Computer Journal*, 13, 3, (1970), 309–316.

Pickett, R. M.; Rubinstein, M. F.; Nelson, R. B.: Automated structural synthesis using a reduced number of design coordinates. *AIAA Journal*, 11, 4, (1973), 489–494.

Piegl, L.: On nurbs: A survey. *Computer Graphics and Applications, IEEE*, 11, 1, (1991), 55–71.

Piegl, L.; Tiller, W.: Curve and surface constructions using rational B-splines. *Computer-Aided Design*, 19, 9, (1987), 485–498.

Piegl, L.; Tiller, W.: *The NURBS book*. Springer Verlag, New York (1996).

Rogers, D. F.; Adams, J. A.: *Mathematical elements for computer graphics*. McGraw-Hill, New York (1976).

Samareh, J.: A survey of shape parameterization techniques. Tech. Rep. NASA-CP-1999-209136, National Aeronautics and Space Administration, Langley Research Center (June 1999).

Schäfer, M.; Heck, M.; Yigit, S.: An implicit partitoned method for the numerical simulation of fluid-structure interaction. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction: Modelling, Simulation, Optimization*, 53, pages 171–194, Springer, Berlin, Heidelberg (2006).

Schoenberg, I. J.; Whitney, A.: On Polya frequency function. III. The positivity of translation determinants with an application to the interpolation problem by spline curves. *Transactions of the American Mathematical Society*, 74, 2, (1953), 246–259.

Sederberg, T. W.; Parry, S. R.: Free-form deformation of solid geometric models. *Computer Graphics*, 20, 4, (1986), 151–160.

Taylor, R.: *FEAP: A finite element analysis program*. UC Berkeley, 7.5 edn. (2003).

Tezduyar, T.; Osawa, Y.: Fluid-structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191, 6-7, (2001), 717 – 726.

Thompson, J. F.; Soni, B.: *Handbook of grid generation*. CRC Press, Boca Raton, Fla. (1999).

Tiller, W.: Rational B-splines for curve and surface representation. *Computer Graphics and Applications, IEEE*, 3, 6, (1983), 61–69.

Zeid, I.: *Mastering CAD-CAM*. McGraw–Hill Higher Education, Boston (2005).

Address: G. Becker, Chair of Numerical Methods in Mechanical Engineering, Technische Universität Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany.
email: `becker@fnb.tu-darmstadt.de`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 21– 30
Herrsching am Ammersee, 29.9.-1.10.2008

On Coupling Schemes for Heat Transfer in FSI Applications

P. Birken, K. J. Quint, S. Hartmann, A. Meister

In this article, the coupling of the temperature-dependent, compressible Navier-Stokes equations solved by a compressible finite volume scheme together with the finite element solution of the heat equation is considered. The application is focused on the cooling process of a heated metal bar treated in the field of metal forming technology. This is done both by loose and strong numerical coupling methods based on the Backward-Euler scheme, where, particularly, Gauss-Seidel and fixed-point solvers are considered.

Keywords: Fluid-Structure Interaction, Thermal Coupling, Partitioned Approach

1 Introduction

Many industrial applications of metal forming involve a simultaneous or subsequent heat treatment. The purpose of this treatment is to improve the mechanical properties such as ductility, hardness, yield strength, or impact resistance. For this purpose the steel is heated up to a certain temperature (austenitic temperature) and then cooled with a critical rate. The thermal evolution (cooling rate) defines the final material properties and, accordingly, its prediction is of particular interest. This complicated process has to be handled by numerical simulations implying thermo-mechanical coupling effects in the gas (fluid-mechanical part), which is used for cooling the metal specimen, the heat transport within the solid (solid mechanical part) and thermo-mechanical coupling effects in the solid itself. The mechanical effects are out of the scope of the investigations here. We will treat the heat transfer from the solid region into the fluid region through a fluid-structure interaction problem.

In our application a metal bar is heated and then cooled at the surface by cold compressed air. This results in an unsteady thermal coupling problem, where the hot steel heats the cold air, which is of low to medium speed. The effect of radiation is neglected for the purpose of getting a more clear picture of the numerical methods with a special focus on the coupling procedure.

Thus, we will look at a model problem, which serves as a stepping stone for further work: the compressible Navier-Stokes-equations as a model for air, coupled along a non-moving boundary with the heat equation as a model for the temperature distribution in the steel. While a lot of work has been done on the thermal coupling of incompressible fluids with structure, we are looking at thermal coupling of a compressible fluid and a structure. Research on numerical simulation of this problem was so far mainly driven by problems where hot gas heats the structure, for example supersonic reentry of vehicles from space or heating of gas-turbine blades (Hinderks and Radespiel, 2006; Mehta, 2005). The results are mainly qualitative, describing numerical methods and the comparison of numerical results to experimental data, with the conclusion that the results are not always in agreement with experiments (Hinderks and Radespiel, 2006).

For the fluid-structure interaction, we consider a partitioned approach (Farhat, 2004), where different codes are used for the subproblems and the coupling is done by a master program which calls by interface functions the other codes. This allows to use existing software for the subproblems, by contrast to a monolithic approach, where a new code is tailored for the coupled equations. This problem is solved numerically using a finite volume method (FVM) for the fluid and a finite element method (FEM) for the heat equation as the methods for space-discretization. Another distinction is made between loose coupling and strong coupling approaches. In the first approach, only one step of each solver is performed in each time step, while the latter approach adds a convergence criterion and an inner loop. We will consider both loose and strong coupling and compare the results on the thermal coupling problem.

The method of lines then implies the time-discretization, where it is common to apply low order time integration in both methods, FVM and FEM, respectively and in the coupling solver.

2 Governing equations and discretization

In the following a thermal coupling problem is considered, where a fluid domain $\Omega_1 \subset \mathbb{R}^2$ and a structure domain $\Omega_2 \subset \mathbb{R}^2$ are given. Within the Ω_1-domain use is made of the temperature-dependent Navier-Stokes equations for compressible flow consisting of the continuity equation, the balance of momentum and the energy relation to describe the thermally coupled fluid flow. In the Ω_2-domain the transient heat equation is assumed. The domains meet at an interface Γ consisting of a curve in $\mathbb{R}^2$, where we require that temperature and heat flux are continuous. No further coupling conditions of the interface are taken into account. For the fluid use is made of the DLR TAU-Code, see (Gerhold et al., 1997), and for the structural part the in-house FEM-program TASAFEM for high-order time-integration is applied, see both (Hartmann, 2006) and, for example, (Hartmann, 2002).

To comply with the condition that temperature and heat flux are continuous at the interface Γ, a so-called Dirichlet-Neumann-coupling is used. Namely, the boundary conditions for the two solvers are chosen such that we prescribe Neumann data for one solver and Dirichlet data for the other. Following the analysis of Giles (1997), temperature is prescribed for the equation with smaller heat conductivity, namely the fluid and heat flux for the structure. Convergence of this approach has been proved for a system of coupled Laplace equations, but not for the case considered here.

2.1 Structure Discretization

The finite element code TASAFEM is a high-order time-integration program originally based on stiffly accurate, diagonally implicit Runge-Kutta methods, see (Ellsiepen and Hartmann, 2001), here extended to the unsteady heat conduction case. The heat conduction problem is, although unsteady, in a first approximation linear. We start from the balance of energy

$$\rho(\boldsymbol{x}) c_D \dot{\Theta}(\boldsymbol{x}, t) = -\operatorname{div} \boldsymbol{q}(\boldsymbol{x}, t), \tag{1}$$

where $\boldsymbol{x}$ defines the spatial coordinates and t the time. The dot symbolizes the time derivative and

$$\boldsymbol{q}(\boldsymbol{x}, t) = -\lambda \operatorname{grad} \Theta(\boldsymbol{x}, t)$$

denotes the heat flux vector depending by Fourier's law on the coefficient of heat conduction λ (which is assumed to define an isotropic heat conductivity). Furthermore, $\Theta(\boldsymbol{x}, t)$ is the absolute temperature, $\rho(\boldsymbol{x})$ the density and c_D denotes the specific heat at constant deformation. On the boundary, we have Neumann conditions, where the heat flux $\boldsymbol{q}(\boldsymbol{x}, t) \cdot \boldsymbol{n}(\boldsymbol{x}) = q(\boldsymbol{x}, t)$ is given on ∂A^q with the outer normal vector $\boldsymbol{n}(\boldsymbol{x})$. Furthermore, initial conditions $\Theta(\boldsymbol{x}, 0) = \Theta_0(\boldsymbol{x})$ are required.

In view of the classical finite element setting, multiplying Eq. (1) with a virtual temperature field $\delta\Theta(\boldsymbol{x})$, the weak formulation reads

$$\int_{\Omega_2} \rho c_D \dot{\Theta} \delta\Theta \, \mathrm{d}V = -\int_{\Omega_2} \lambda \operatorname{grad} \Theta \operatorname{grad} \delta\Theta \, \mathrm{d}V - \int_{\partial A^q} q \delta\Theta \, \mathrm{d}A. \tag{2}$$

Next, one inserts an ansatz

$$\Theta^h(\boldsymbol{x}, t) = \boldsymbol{N}^T(\boldsymbol{x}) \, \boldsymbol{\Theta}(t) \tag{3}$$

$$\delta\Theta^h(\boldsymbol{x}) = \boldsymbol{N}^T(\boldsymbol{x}) \, \delta\boldsymbol{\Theta} \tag{4}$$

into Eq. (2). The temperature gradient reads in matrix notation

$$\operatorname{grad} \Theta^h = \left\{ \begin{array}{c} \partial_{x_1} \Theta^h \\ \partial_{x_2} \Theta^h \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{N}_{,1}^T \\ \boldsymbol{N}_{,2}^T \end{array} \right\} \boldsymbol{\Theta}(t) = \boldsymbol{B}(\boldsymbol{x}) \boldsymbol{\Theta}(t) \tag{5}$$

with the temperature gradient- nodal temperature matrix $\boldsymbol{B}(\boldsymbol{x})$. If we insert ansatz (3) and (4) into the weak formulation (2), we obtain a system of ordinary differential equations

$$\boldsymbol{g}(t, \boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}) = \boldsymbol{M} \dot{\boldsymbol{\Theta}}(t) + \boldsymbol{K}(\boldsymbol{\Theta}) \boldsymbol{\Theta}(t) - \bar{\boldsymbol{q}}(t, \underline{\boldsymbol{u}}) = \boldsymbol{0}. \tag{6}$$

The heat flux on the coupling boundary is defined by

$$\bar{\boldsymbol{q}}(t,\underline{\boldsymbol{u}}) = -\int_{\partial A^q} \boldsymbol{N}\,\boldsymbol{q}^h(t,\underline{\boldsymbol{u}})\,\mathrm{d}A, \tag{7}$$

where $\underline{\boldsymbol{u}}$ was introduced to denote the dependence on the fluid data as explained in the next section.

$$\boldsymbol{M} = \int_{\Omega_2} \rho c_D \boldsymbol{N}\,\boldsymbol{N}^T\,\mathrm{d}V \qquad \boldsymbol{K} = \int_{\Omega_2} \lambda(\Theta^h)\boldsymbol{B}^T\boldsymbol{B}\,\mathrm{d}V \tag{8}$$

are the matrices concerned. In the case of a constant domain, no volumetric distributed heat sources and temperature-independent material parameters, a Backward-Euler step of Eq.(6) reads

$$[\boldsymbol{M} + \Delta t_n\,\boldsymbol{K}]\,\boldsymbol{\Theta}^{n+1} = \boldsymbol{M}\boldsymbol{\Theta}^n + \bar{\boldsymbol{q}}(t_{n+1},\underline{\boldsymbol{u}}^{n+1}) \tag{9}$$

implying the solution of a symmetric, sparse linear system of equations to obtain the nodal temperatures at time t_{n+1}.

2.2 Fluid discretization

Concerning the fluid part, the flow is assumed to be governed by the two dimensional temperature-dependent compressible Navier-Stokes equations. The common non-dimensional integral form of the corresponding conservation laws reads

$$\frac{d}{dt}\int_\sigma \boldsymbol{u}\mathrm{d}V + \sum_{j=1}^{2}\int_{\partial\sigma} \boldsymbol{f}_j^c(\boldsymbol{u})\,n_j\mathrm{d}A = \frac{1}{\mathrm{Re}_\infty}\sum_{j=1}^{2}\int_{\partial\sigma}\boldsymbol{f}_j^v(\boldsymbol{u})\,n_j\mathrm{d}A, \tag{10}$$

where $\boldsymbol{n} = \{n_1, n_2\}^T$ represents the outwards unit normal vector at the boundary of the control volume σ. Furthermore,

$$\boldsymbol{u} = \{\rho, \rho v_1, \rho v_2, \rho E\}^T$$

is the vector of the conserved variables and $\boldsymbol{f}_j^c, \boldsymbol{f}_j^v, j = 1, 2$ are the convective and viscous fluxes which are given by

$$\boldsymbol{f}_j^c(\boldsymbol{u}) = \begin{Bmatrix} \rho v_j \\ \rho v_1 v_j + \delta_{1j}p \\ \rho v_2 v_j + \delta_{2j}p \\ \rho H v_j \end{Bmatrix} \text{ and } \boldsymbol{f}_j^v(\boldsymbol{u}) = \begin{Bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \sum_{i=1}^2 v_i\tau_{ij} + \frac{\mu\gamma}{\mathrm{Pr}_\infty}\partial_{x_j}e \end{Bmatrix},$$

respectively. The quantity e denotes the internal energy, which is given by $e = E - 1/2\left(v_1^2 + v_2^2\right)$ and H is defined by $H = E + p/\rho$. The pressure is determined by the equation of state $p = (\gamma-1)\rho\left(E - 1/2\left(v_1^2 + v_2^2\right)\right)$, where γ denotes the ratio of specific heats. The temperature is given by $\Theta = \gamma(\gamma-1)\mathrm{Ma}_\infty^2 e$, where Ma_∞ denotes the Mach number at infinity. The elements of the shear stress tensor are

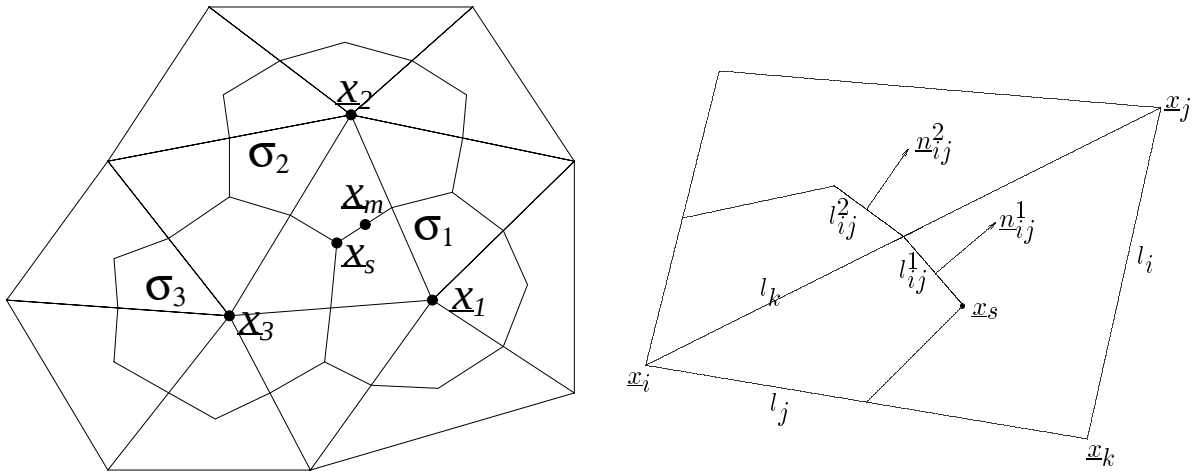
$$\tau_{ij} = \mu\left(\partial_{x_j}v_i + \partial_{x_i}v_j\right) + \delta_{ij}\tilde{\lambda}\left(\partial_{x_1}v_1 + \partial_{x_2}v_2\right),$$

with the viscosity assumed to follow the Sutherland law $\mu = \Theta^{1.5}(1+S)/(\Theta+S)$, where $S = 110K/\Theta_\infty$ and Θ_∞ denote the temperature at infinity measured in degree of Kelvin. Moreover, the relation between the thermal conductivity and the viscosity is defined by the Stokes' hypothesis to be $\tilde{\lambda} = -2/3\mu$ and Re_∞ and Pr_∞ denote the Reynolds and Prandtl number at infinity, respectively.

In order to solve (10) numerically, we consider a conforming triangulation $\mathcal{T}_h$ of the spatial domain is the sense of Delaunay, see (Friedrich, 1993). Based on this primary grid, we define a discrete control volume σ_i as the volume of the barycentric subdivision of $\mathcal{T}_h$ enclosing the node $\boldsymbol{x}_i = \{x_{i1}, x_{i2}\}^T$ and bounded by the straight line segments l_{ij}^k connecting the midpoint of the edge with the inner point $\boldsymbol{x}_s$ (see Fig. 1). For a detailed description, we refer to Meister and Sonar (1998). Utilizing our notion of control volumes and introducing the cell average on σ_i by $\boldsymbol{u}_i(t) := \int_{\sigma_i}\boldsymbol{u}(\boldsymbol{x},t)\mathrm{d}V/|\sigma_i|$ into the Navier-Stokes equations (10), we obtain the form

$$\dot{\boldsymbol{u}}_i(t) = \frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}\int_{l_{ij}^k}\sum_{\ell=1}^{2}\left(\frac{1}{\mathrm{Re}_\infty}\boldsymbol{f}_\ell^v(\boldsymbol{u}) - \boldsymbol{f}_\ell^c(\boldsymbol{u})\right)n_\ell\mathrm{d}A,$$

where $N(i)$ denote the index set of all control volumes σ_j neighboring box σ_i. To overcome the difficulty that the line integrals are usually not defined if $\boldsymbol{u}$ is discontinuous, we introduce the concept of numerical flux functions.



(a) Triangulation $\mathcal{T}_h$ (b) Boundary lines

Figure 1: General form of a control volume

Concerning the convective part, we make use of the well-known AUSMDV scheme, see (Wada and Liou, 1994). Furthermore, the viscous fluxes are discretized by central differences. Therefore, for each physical quantity ϕ appearing within the viscous flux, the unique linear distribution with respect to the triangle τ is calculated by means of the cell averages of the three adjacent control volumes σ_i satisfying $\sigma_i \cap \tau \neq \emptyset$. In this procedure the cell averages are considered to be located at the vertices of the triangle. Due to this reconstruction the value as well as the gradient of each quantity can easily be evaluated at the midpoint of the inner line segment $l_{ij}^k \subset \tau$. Thus, the semi-discrete form of the governing equations reads

$$\dot{\boldsymbol{u}}_i(t) = \frac{1}{|\sigma_i|} \sum_{j \in N(i)} \sum_{k=1}^{2} |l_{ij}^k| \left(\boldsymbol{h}^v(\boldsymbol{u}_i^{n+1}, \boldsymbol{u}_j^{n+1}, \boldsymbol{u}_m^{n+1}; \boldsymbol{n}_{ij}^k) - \boldsymbol{h}^c(\hat{\boldsymbol{u}}_i^{n+1}, \hat{\boldsymbol{u}}_j^{n+1}; \boldsymbol{n}_{ij}^k) \right),$$

where $\boldsymbol{h}^c$ is the AUDMDV flux and $\boldsymbol{h}^v$ corresponds to the discretization of the viscous fluxes. Note that the notation $\hat{\boldsymbol{u}}_i$ emphasizes that we increase the order of accuracy for the convective part by utilizing a well-known TVD-like reconstruction technique and, accordingly, $\hat{\boldsymbol{u}}_i$ denotes the one-sided limit with respect to the box σ_i at the midpoint of the line segment l_{ij}^k. If we write this as an equation for the complete domain, we obtain

$$\underline{\dot{\boldsymbol{u}}}(t) = \boldsymbol{\sigma}^{-1} \underline{\boldsymbol{h}}(\underline{\boldsymbol{u}}, \boldsymbol{\Theta}). \tag{11}$$

The underlined vectors represent the respective vectors on the whole fluid grid and we have included the dependence on the structural temperature on the coupling interface through the vector of the structure temperatures $\boldsymbol{\Theta}$. The matrix $\boldsymbol{\sigma}$ is a diagonal matrix with the volumes of the corresponding cells on the diagonal.

Similar to TASAFEM the restrictive time-step constraint of an explicit discretization technique for the time derivative is overcome using an Backward-Euler approach. Thus, the discrete form of the governing equations reads

$$\underline{\boldsymbol{u}}^{n+1} = \underline{\boldsymbol{u}}^n + \Delta t_n \boldsymbol{\sigma}^{-1} \underline{\boldsymbol{h}}(\underline{\boldsymbol{u}}^{n+1}, \boldsymbol{\Theta}^{n+1}), \tag{12}$$

where $\underline{\boldsymbol{u}}^{n+1} \approx \underline{\boldsymbol{u}}(t_{n+1})$ and $t_{n+1} = t_n + \Delta t_n$. It is easily seen that each time step within the fluid solver requires the solution of a sparse non-linear system of equations, which is performed by a dual time-stepping approach, see (Jameson, 2004). The precise choice of the solver for this non-linear system is not important here, so instead, a Newton-Krylov method could be used.

2.3 Coupled equations

If we combine the semidiscrete equations (6) for the domain Ω_1 and (11) for the domain Ω_2, we obtain a coupled system of ODEs

$$\begin{aligned} \underline{\dot{\boldsymbol{u}}}(t) &= \boldsymbol{\sigma}^{-1} \underline{\boldsymbol{h}}(\underline{\boldsymbol{u}}, \boldsymbol{\Theta}), \\ \boldsymbol{M}\dot{\boldsymbol{\Theta}}(t) &= -\boldsymbol{K}\boldsymbol{\Theta}(t) - \bar{\boldsymbol{q}}(t, \underline{\boldsymbol{u}}), \end{aligned} \tag{13}$$

where we prescribe the additional algebraic constraint that temperature and heat flux are continuous at the coupling interface Γ. The application of the Backward-Euler method to the coupled system is straightforward. The question is now, how the coupled system can be solved accurately and efficiently.

3 Fluid-Structure-Coupling

As described above, we pursue a partitioned approach. The technical difficulty of different programming languages (FORTRAN for TASAFEM and C++ for TAU) in the partitioned approach is dealt by means of the C++-library called Component Template Library (CTL), see (Matthies et al., 2006).

It is assumed that at time t_n the fluid data $\underline{\boldsymbol{u}}^n$, the structure data $\boldsymbol{\Theta}^n$ and a global step-size Δt_n are given. As described above, the fluid and the structural equations are both treated implicitly with associated solvers for the time-stepping procedure. In the coupling context, it is useful to regard the two solvers as mappings that, for given fixed data $\underline{\boldsymbol{u}}^n$ at t_n, respectively $\boldsymbol{\Theta}^n$, take an approximation of the boundary data at t_{n+1} from the other solver and provide a new approximation to their data at t_{n+1}, which provides new boundary data for the other solver. The fluid solver provides a solution to (12) and can be written as

$$\underline{\boldsymbol{u}}^{n+1} = \boldsymbol{F}(\boldsymbol{P}(\boldsymbol{\Theta})),$$

whereas the structure solver provides a solution to (9) and can be represented by

$$\boldsymbol{\Theta}^{n+1} = \boldsymbol{S}(\boldsymbol{q}_\Gamma(\underline{\boldsymbol{u}})).$$

$\boldsymbol{P}$ is a projection of the temperature onto the boundary of Ω_2 and $\boldsymbol{q}_\Gamma$ provides the boundary heat flux in the fluid. Using this notation, it is possible to define coupling methods. The most simple coupling procedures are loose coupling methods, where no convergence criterion is used in the coupling iteration. In particular, there is Gauss-Seidel coupling

$$\underline{\boldsymbol{u}}^{n+1} = \boldsymbol{F}(\boldsymbol{P}(\boldsymbol{\Theta}^n)), \tag{14}$$
$$\boldsymbol{\Theta}^{n+1} = \boldsymbol{S}(\boldsymbol{q}_\Gamma(\underline{\boldsymbol{u}}^{n+1})), \tag{15}$$

and Jacobi-coupling, which can be done in parallel:

$$\underline{\boldsymbol{u}}^{n+1} = \boldsymbol{F}(\boldsymbol{P}(\boldsymbol{\Theta}^n)), \tag{16}$$
$$\boldsymbol{\Theta}^{n+1} = \boldsymbol{S}(\boldsymbol{q}_\Gamma(\underline{\boldsymbol{u}}^n)). \tag{17}$$

These can be iterated leading to fixed point coupling, here for the Gauss-Seidel case:

$$\underline{\boldsymbol{u}}_{k+1}^{n+1} = \boldsymbol{F}(\boldsymbol{P}(\boldsymbol{\Theta}_k^{n+1})), \tag{18}$$
$$\boldsymbol{\Theta}_{k+1}^{n+1} = \boldsymbol{S}(\boldsymbol{q}_\Gamma(\underline{\boldsymbol{u}}_{k+1}^{n+1})), \quad k = 0, 1, \ldots. \tag{19}$$

As a fixed point equation this is given by

$$\boldsymbol{P}(\boldsymbol{\Theta}) = \boldsymbol{P}(\boldsymbol{S}(\boldsymbol{q}_\Gamma(\boldsymbol{F}(\boldsymbol{P}(\boldsymbol{\Theta}))))), \tag{20}$$

which can be used as a convergence criterion for the fixed point iteration. Various methods have been proposed to increase the convergence speed of the fixed point iteration by decreasing the interface error between subsequent steps, for example Relaxation (Le Tallec and Mouro, 2001; Küttler and Wall, 2008), Interface-GMRES (Michler et al., 2006) or ROM-coupling (Vierendeels et al., 2007). For the purpose of looking at the qualitative behavior of loose and strong coupling, it is sufficient to analyze the more simple methods described here.

4 Numerical Results

4.1 Test case

To analyze the properties of the coupling method, the test example is chosen as simple as possible. The reason is that this comparably simple coupling problem is already beyond the current solution theory, respectively convergence theory of numerical methods. Therefore, we choose a test case where the exact solutions for the uncoupled problems are known in order to make sure that no additional side effects are present, which cannot be controlled.

Accordingly, the cooling of a flat plate resembling a simple work piece is considered (described in Fig. 2 as solid). This example has also been studied by other authors (Yarrington and Thornton, 1994) and (Huebner et al., 2001, p. 465) in conjunction with the cooling of structural parts in hypersonic vehicles. There localized heating was of special interest. In our case the work piece is initially at a temperature of $\Theta(\boldsymbol{x}, 0) = 900$ K and is cooled by a constant air stream. The latter is modeled in a first approximation as a laminar flow along the plate, see Fig. 2. For

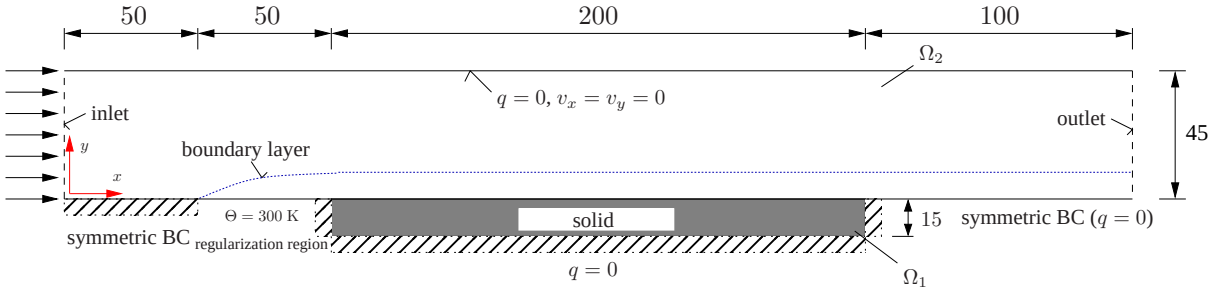
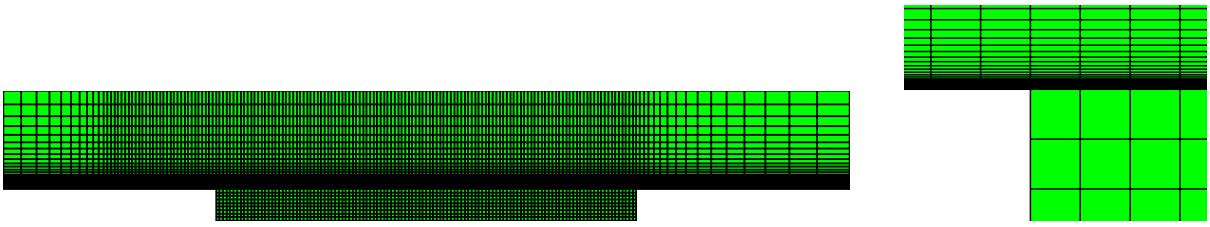


Figure 2: Test case for the coupling method

the work piece the following constant material properties are assumed: mass density $\rho = 7836\ \mathrm{kg/m^3}$, specific heat capacity $c_D = 443\ \mathrm{J/(kgK)}$ and thermal conductivity $\lambda = 48.9\ \mathrm{W/(mK)}$. The inlet is at the left, where the air enters the domain with an initial velocity of $\mathrm{Ma}_\infty = 0.8$ in horizontal direction and a temperature of 273 K. Then, there are two succeeding regularization regions of $50\ \mathrm{mm}$ to obtain an unperturbed boundary layer. In the first region, $0 \leq x \leq 50$, symmetry boundary conditions, $v_y = 0$, $q = 0$, are applied. In the second region, $50 \leq x \leq 100$, a constant wall temperature of 300 K is specified. Within this region the velocity boundary layer fully develops. The third part is the solid (work piece) of length $200\ \mathrm{mm}$, which exchanges heat with the fluid, but is assumed insulated otherwise, $q = 0$. Therefore, the corresponding Neumann boundary conditions are applied throughout. Finally, the flow domain is closed by a second regularization region of $100\ \mathrm{mm}$ with symmetry boundary conditions and the outlet.

The grid, see Fig. 3, in the structural part is chosen cartesian and equidistant, whereas the thinnest cells in the fluid



(a) Entire mesh (b) Mesh zoom

Figure 3: Full grid (left) and zoom into coupling region (right)

region have an aspect ratio of 1:200 and then become coarser in y-direction. The points of the primary fluid grid and the nodes of the structure grid match on the interface, which avoids additional difficulties from interpolation. Thus, we have 9660 cells in the fluid region and $n_x \times n_y = 120 \times 9 = 1080$ elements with $121 \times 10 = 1210$ nodes in the region of the structure.

To specify reasonable initial conditions within the fluid a steady state solution of the flow with constant wall temperature is computed. To cope with convergence problems we first compute a solution with a medium boundary temperature. In a second step the temperature at the boundary is increased up to the value $\Theta = 900$ K. Due to the constant boundary temperature we are able to compare the results with the theoretical solution of Blasius for the velocity boundary layer and of van Driest for the temperature boundary layer (Van Driest, 1952) and thereby verify the quality of our grid and our fluid solver. In the structure, a constant temperature of 900 K at $t = 0$ s is chosen throughout.

4.2 Numerical tests

In Fig. 4 one can see the temporal evolution of the temperature at the middle of the coupling interface. As expected,

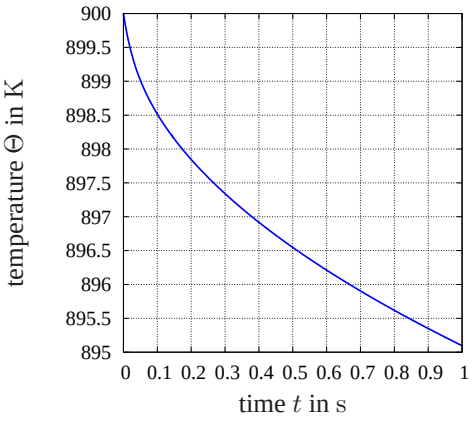
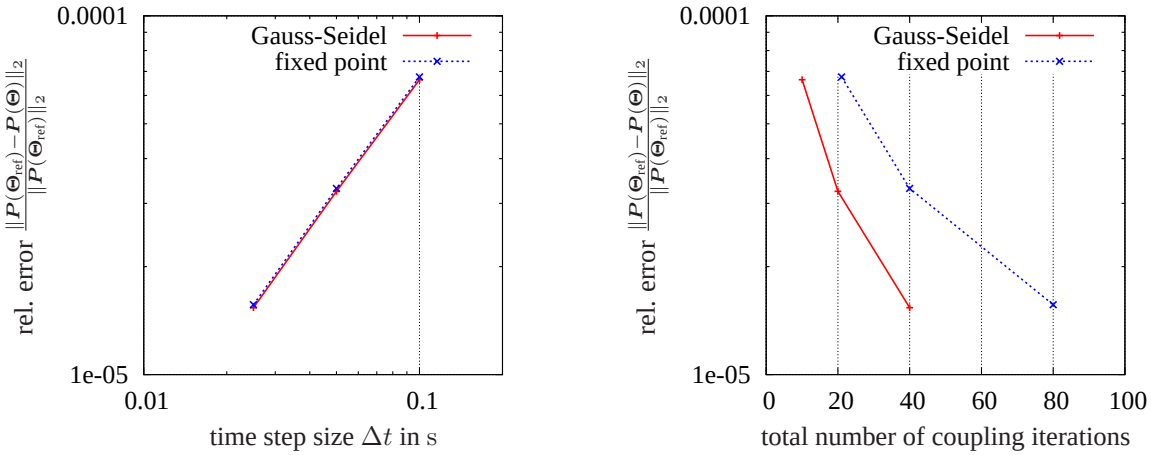


Figure 4: Temperature evolution at the middle point of the interface

the temperature decreases monotonously with a large gradient at the beginning of the process, which decreases in the following. At $t = 1$ s, the temperature has dropped down from 900 K to approximately 895 K. This solution is obtained using fixed point coupling and $\Delta t_n = 0.0025$ s. Since no exact solution is available, it will be used as reference solution.

As for strong coupling methods, the fixed point method is iterated until the vector 2-norm of the interface residual (20) has dropped below $\epsilon = 0.1$, i.e. $\left\| \boldsymbol{P}\left(\boldsymbol{\Theta}_{k+1}^{n+1}\right) - \boldsymbol{P}\left(\boldsymbol{\Theta}_{k}^{n+1}\right) \right\|_2 \leq \epsilon$. As mentioned before $\boldsymbol{P}$ is a projection of the temperature onto the boundary and k is the iteration number of the fixed point iteration. Except for the first time-step, two iterations are sufficient to fulfill this criterion. In this case, for $\Delta t = 0.1$ s and $\Delta t = 0.05$ s, three iterations are needed.

Next, the Gauss-Seidel coupling is compared with the iterated Gauss-Seidel (fixed point) coupling for time step sizes Δt of 0.1 s, 0.05 s and 0.025 s. To this end, we consider the resulting error at $t = 1$ s, using the 2-norm of the difference of temperatures at the interface to the reference solution. In Fig. 5(a) one can see the error over the



(a) Relative error over time-steps

(b) Relative error over coupling iterations

Figure 5: Relative error behavior

time step size and in Fig. 5(b), compared to the number of coupling iterations. As can be seen, in the investigated application fixed point coupling does not improve the accuracy of the solution if $\epsilon = 0.1$ is employed as dropping tolerance. The relative error of the numerical method is practically unaltered but the computational cost is at least doubled, depending on the specified tolerance ϵ of the fixed point coupling. On the other hand the time step size Δt has, as expected, a significant influence on the accuracy.

5 Conclusions

The coupling of the temperature-dependent compressible Navier-Stokes equations using a finite volume code and the heat equation using finite elements, both based on a Backward-Euler time-integration step, are considered. In the investigated test example the fluid cools the structure. We compare loose to strong coupling methods for this problem occurring in the field of hot metal forming processes and it can be seen that for a simple example of a plain metal specimen and a dropping tolerance of $\epsilon = 0.1$, loose coupling methods are completely sufficient.

Acknowledgements

We would like to thank Dr. Volker Hannemann from the DLR Göttingen for his help with the TAU-Code and Dr. Rainer Niekamp from the TU Braunschweig supporting us with the usage of CTL. This paper is based on investigations of the collaborative research center SFB/TR TRR 30, which is kindly supported by the German Research Foundation (DFG)

References

Ellsiepen, P.; Hartmann, S.: Remarks on the interpretation of current non-linear finite-element-analyses as differential-algebraic equations. *Int. J. Num. Meth. Eng.*, 51, (2001), 679–707.

Farhat, C.: CFD-based nonlinear computational aeroelasticity. In: E. Stein; R. de Borst; T. J. Hughes, eds., *Encyclopedia of Computational Mechanics, Volume 3: Fluids*, pages 459–480, John Wiley and Sons (2004).

Friedrich, O.: A new method for generating inner points of triangulations in two dimensions. *Comp. Meth. Appl. Mech. Eng.*, 104, (1993), 77–86.

Gerhold, T.; Friedrich, O.; Evans, J.; Galle, M.: Calculation of complex three-dimensional configurations employing the DLR-TAU-Code. *AIAA Paper*, 97-0167.

Giles, M.: Stability analysis of numerical interface conditions in fluid-structure thermal analysis. *Int. J. Num. Meth. in Fluids*, 25, (1997), 421–436.

Hartmann, S.: Computation in finite strain viscoelasticity: finite elements based on the interpretation as differential-algebraic equations. *Comp. Meth. Appl. Mech. Eng.*, 191, 13-14, (2002), 1439–1470.

Hartmann, S.: TASA-FEM: Ein Finite-Elemente-Programm für raum- und zeitadaptive gekoppelte Strukturberechnungen. Tech. Rep. 1/2006, Institute of Mechanics, University of Kassel, Kassel, Germany (2006).

Hinderks, M.; Radespiel, R.: Investigation of hypersonic gap flow of a reentry nosecap with consideration of fluid structure interaction. *AIAA Paper*, 06-1111.

Huebner, K. H.; Dewhirst, D. L.; Smith, D. E.; Byrom, T. G.: *The Finite Element Methods for Engineers*. John Wiley & Sons, 4th edn. (2001).

Jameson, A.: Aerodynamics. In: E. Stein; R. de Borst; T. J. Hughes, eds., *Encyclopedia of Computational Mechanics, Volume 3: Fluids*, pages 325–406, John Wiley and Sons (2004).

Küttler, U.; Wall, W.: Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Comp. Mech.*, 43, (2008), 61–72.

Le Tallec, P.; Mouro, J.: Fluid structure interaction with large structural displacements. *Comp. Meth. Appl. Mech. Eng.*, 190, (2001), 3039–3067.

Matthies, H. G.; Niekamp, R.; Steindorf, J.: Algorithms for strong coupling procedures. *Comp. Meth. Appl. Mech. Eng.*, 195, (2006), 2028–2049.

Mehta, R. C.: Numerical computation of heat transfer on reentry capsules at mach 5. *AIAA-Paper 2005-178*.

Meister, A.; Sonar, T.: Finite-volume schemes for compressible fluid flow. *Surv. Math. Ind.*, 8, (1998), 1–36.

Michler, C.; van Brummelen, E.; de Borst, R.: Error-amplification analysis of subiteration-preconditioned GMRES for fluid-structure interaction. *Comp. Meth. Appl. Mech. Eng.*, 195, (2006), 2124–2148.

Van Driest, E.: National advisory commitee for aeronautics (naca) - investigation of laminar boundary layer in compressible fluids using the crocco method. *NACA*.

Vierendeels, J.; Lanoye, L.; Degroote, J.; Verdonck, P.: Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Comp. & Struct.*, 85, (2007), 970–976.

Wada, Y.; Liou, M.-S.: A Flux Splitting Scheme with High-Resolution and Robustness for Discontinuities. *AIAA Paper*, 94–0083.

Yarrington, P. W.; Thornton, E. A.: Finite element analysis of low-speed compressible flows within convectively cooled structures. *Journal of Thermophysics and Heat Transfer*, 8–4.

Address: Dr. P. Birken (corresponding author), Department of Mathematics, University of Kassel, Heinrich-Plett Str. 40, 34132 Kassel, Germany. email: birken@mathematik.uni-kassel.de

Dipl.-Ing. K.-J. Quint, Institute of Mechanics, University of Kassel, Mönchebergstr. 7, 34105 Kassel, Germany. karsten.quint@uni-kassel.de

Prof. Dr.-Ing. St. Hartmann, Institute of Applied Mechanics, Clausthal University of Technology, Adolph-Roemer Str. 2a, 38678 Clausthal-Zellerfeld, Germany, email: stefan.hartmann@tu-clausthal.de

Prof. Dr. A. Meister, Department of Mathematics, University of Kassel, Heinrich-Plett Str. 40, 34132 Kassel, Germany. email: meister@mathematik.uni-kassel.de

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 31– 42
Herrsching am Ammersee, 29.9.-1.10.2008

FSI of the Turbulent Flow around a Swiveling Flat Plate Using Large–Eddy Simulation

M. Breuer, M. Münsch

The topic is of fluid–structure interaction plays a dominant role in many fields of engineering. Therefore, a strong need for appropriate numerical simulation tools exists with a variety of numerical and physical aspects. The special task of the present investigation is to take care about fluid–structure interaction of turbulent flows using modern simulation techniques such as large–eddy simulation. For that purpose, a new coupling scheme was recently developed which guarantees a strong coupling between the fluid and the structure but still relies on an explicit predictor–corrector scheme for time marching. Thus the specific features of numerical methods typically applied for large–eddy simulations are conserved and extended to the coupled problem. The methodology is briefly described in the paper. Furthermore, two test cases, a laminar and a turbulent flow around elastically supported cylindrical structures, were taken into account to validate the numerical scheme. Finally, the coupling algorithm is applied to a challenging test case, i.e., the turbulent flow around a swiveling flat at Re $\approx$ *68,000.*

1 Introduction

Fluid–structure interaction (FSI) is of major importance for many fields of engineering such as mechanical engineering (e.g. rotor blades and airfoils), process engineering (e.g. flexible tubes and stirrers), medicine technique (e.g. artificial heart valves and substitute vocal folds), or civil engineering (e.g. stadium roofage and suspension bridges). To solve these multi–physical problems, numerical simulations have become an important tool. In order to develop and investigate such numerical methods for the robust and efficient simulation of coupled problems related to the interaction of a fluid with a structure, the DFG special research group FOR 493 *'Fluid–Structure Interaction – Modeling, Simulation and Optimization'* was initiated. It is concerned with a variety of different aspects of FSI algorithms including different solvers for the fluid mechanical and the structural parts and especially with the coupling schemes.
To solve coupled FSI problems a partitioned approach can be used, i.e., two highly specialized codes for the fluid mechanical part and the structural part are coupled via a coupling interface. This has the advantage that adequate codes and with this the most accurate simulation approach can be applied on each problem of interest. In contrast to the counterpart, i.e., a monolithic approach based on a unique numerical method and a common code for both disciplines, the partitioned approach allows an easy exchange of single codes for each field separately. Thus it is much more flexible than the monolithic approach.
In terms of flow computation the simulation approach used strongly depends on the expected flow properties. For the computation of turbulent flows one can make use among others of three categories of numerical simulation methodologies. The most accurate one for turbulence computations by solving the Navier–Stokes equations without averaging or approximations for all motions in the flow field is the direct numerical simulation (DNS), see e.g., Ferziger and Perić (2002) and Breuer (2002). For this methodology a high number of grid points and time steps are required which makes the simulations expensive and restricts them to moderate Reynolds numbers. Averaging the Navier–Stokes equations leads to the second group of methods, the so-called Reynolds-averaged Navier Stokes (RANS) equations which have to be closed by a statistical turbulence model. The flow is described in a statistical sense which leads to a time–averaged flow field. In previous studies, laminar as well as turbulent FSI applications using the Reynolds–Averaged Navier–Stokes approach were investigated (Glück et al., 2001, 2003). For that purpose, a partitioned fully implicit scheme was applied which coupled a three–dimensional finite–volume based multi–block flow solver for incompressible fluids with a finite–element code for the structural problem. This coupling scheme works efficiently for large time step sizes typically used for implicit time–stepping schemes within RANS predictions. However, flow problems involving large–scale flow structures such as vortex shedding or in-

stantaneous separation and reattachment are often not reliably predicted by RANS and more advanced techniques such as large–eddy simulation (LES) are required (Breuer, 1998, 2000, 2002). The LES technique is based on spatial filtering of the Navier–Stokes equations. Here, the large scales are computed directly while the small scales are modeled leading to an unsteady field of physical values, which enables predictions of flows past bluff and inclined bodies and makes LES to the tool of choice regarding to FSI in turbulent flows when large–scale flow structures are expected.

2 Numerical Methodology

2.1 Governing Equations

For the intended purpose, a finite–volume scheme is used to discretize the filtered Navier–Stokes equations for an incompressible fluid based on a curvilinear, blockstructured body–fitted grid with colocated variable arrangement. The present study relies on the inhouse code FASTEST–3D (Durst and Schäfer, 1996; Durst et al., 1996) which was extended for the LES–FSI problem as will be described below. For the discretization standard schemes are applied. A second–order accurate central scheme is obtained by linear interpolation of the flow variables to the cell faces and a midpoint rule approximation of the integrals. Compared to computations on fixed grids, for FSI the grid movement caused by the structural displacement at the boundaries of the computational domain has to be taken into account. That is achieved in a consistent manner by applying the well–known Arbitrary–Lagrangian–Eulerian (ALE) formulation. Thus the mass and momentum conservation equations in integral form are formulated for a control volume (CV) with time-dependent volume $V(t)$ and surface $S(t)$. The governing equations in ALE formulation read:

Mass Conservation:

$$\frac{d}{dt}\int_{V(t)} \rho \, dV + \int_{S(t)} \rho(u_j - u_{g,j}) \cdot n_j \, dS = 0 \tag{1}$$

Momentum Conservation:

$$\frac{d}{dt}\int_{V(t)} \rho u_i \, dV + \int_{S(t)} \rho u_i(u_j - u_{g,j}) \cdot n_j \, dS = \int_{S(t)} \tau_{ij} \cdot n_j \, dS - \int_{S(t)} p \cdot n_i \, dS \tag{2}$$

Here, ρ denotes the density, p the pressure, u_i the velocity vector with three Cartesian components, and τ_{ij} the molecular momentum transport tensor. Furthermore, n_j stands for the unit normal vector directed outwards. Finally, $u_{g,j}$ describes the grid velocity with which the CV surface is moving. Since an incompressible fluid with temperature–independent fluid properties is considered here, the conservation equation for the energy can be omitted. To close the governing equations in case of LES, a classical Smagorinsky subgrid–scale model (Smagorinsky, 1963) with Van–Driest damping near solid walls or a dynamic model by Germano et al. (1991) and Lilly (1992) is applied.

2.2 Space Conservation Law

The decisive differences between these conservation equations and their counterpart on a fixed grid are the volume integrals which now describe the local changes within a moving or deformable CV and the additional mass and momentum fluxes resulting from the movement of the CV surface given by $u_{g,j}$. In order to end up with a closed system of equations again, this new unknown has to be determined. If the grid movement between two or more successive time steps is known, $u_{g,j}$ can in principle be calculated by a simple finite–difference approximation. However, this procedure is not recommended since it often leads to serious problems because the conservation principle is not necessarily satisfied and thus mass or momentum is lost. As a remedy the so-called *space conservation law (SCL)* (Demirdžić and Perić, 1988, 1990) should be taken into account in order to determine the unknown grid velocity $u_{g,j}$. It represents an extra conservation equation guaranteeing that no space is lost when the CV changes its shape or position. It reads:

$$\frac{d}{dt}\int_{V(t)} dV - \int_{S(t)} u_{g,j} \cdot n_j \, dS = 0 \tag{3}$$

In discretized form the *SCL* is expressed by the swept volumes of the corresponding cell faces. The nice feature of the *SCL* is that in combination with the mass conservation equation (1) on a moving grid, the original mass conservation equation for a fixed grid is obtained. Thus the pressure–correction scheme for the solution of the incompressible Navier–Stokes equations does not have to be changed. Furthermore, the *SCL* is applied to consistently determine the newly appearing grid fluxes in the momentum equations (2). Consequently, in the final numerical scheme no extra conservation equation has to be solved.

2.3 FSI Coupling Scheme

In LES small time steps are required to resolve the turbulent flow field in time. Accordingly, the usage of explicit time–marching schemes is preferred, i.e., a predictor–corrector scheme of second–order accuracy. In the predictor step an explicit three substeps low–storage Runge–Kutta scheme advances the momentum equations in time. In the following corrector step the mass conservation has to be fulfilled by solving a Poisson equation for the pressure correction based on an incomplete LU decomposition solver. Whereas the predictor step is solely carried out once per time step, the pressure–correction algorithm is repeated until a predefined convergence criterion ε is reached. It guarantees that the mass conservation equation at the end of the corrector step is numerically satisfied, e.g., $\Delta\dot{m} < \varepsilon = \mathcal{O}(10^{-8})$. For that purpose, typically 5 to 10 pressure–correction iterations are required.

For solving FSI problems, this scheme has to be adapted and a corresponding coupling scheme has to be incorporated. The coupling scheme developed (Breuer and Münsch, 2008a,b) is shown in Figure 1(a). The new time step starts with an estimation of the structural displacement $\tilde{X}$ and the structural velocity $\dot{\tilde{X}}$. Here, a second–order extrapolation for the displacement $\tilde{X}$ is used [Eq. (4)] taking the displacement values of the three former time steps denoted by the superscripts $^{t-1}$, $^{t-2}$, and $^{t-3}$ into account:

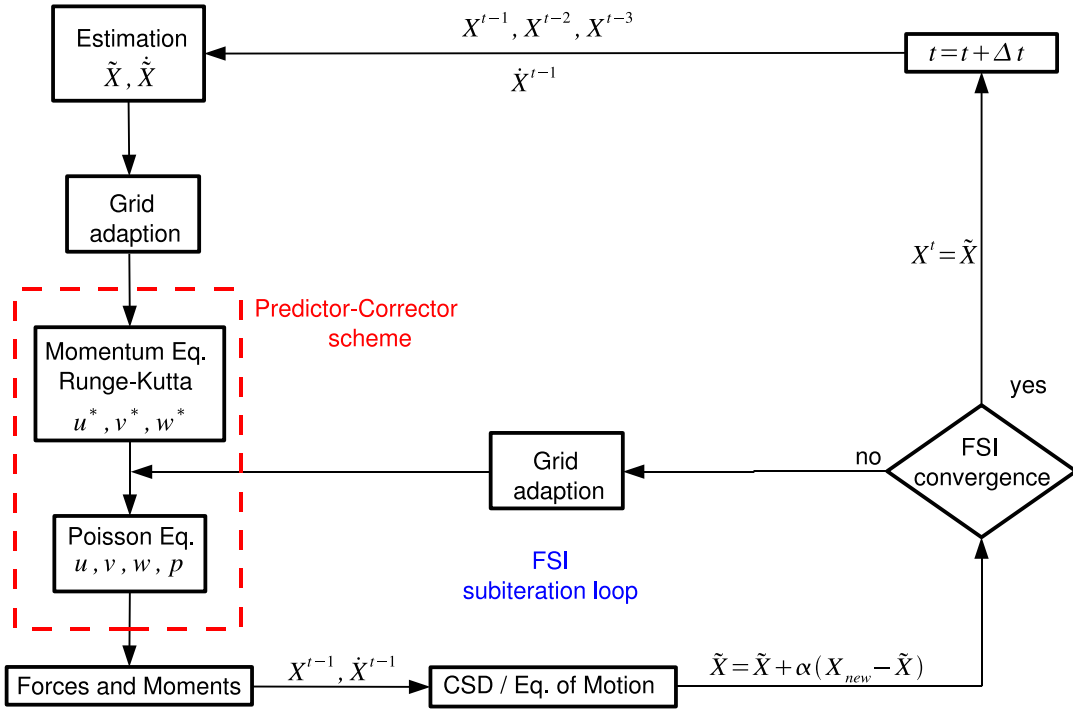
$$\tilde{X}^t = 3X^{t-1} - 3X^{t-2} + X^{t-3} \; . \tag{4}$$

In order to be consistent, a first–order extrapolation is applied on the structural velocity $\dot{\tilde{X}}$:

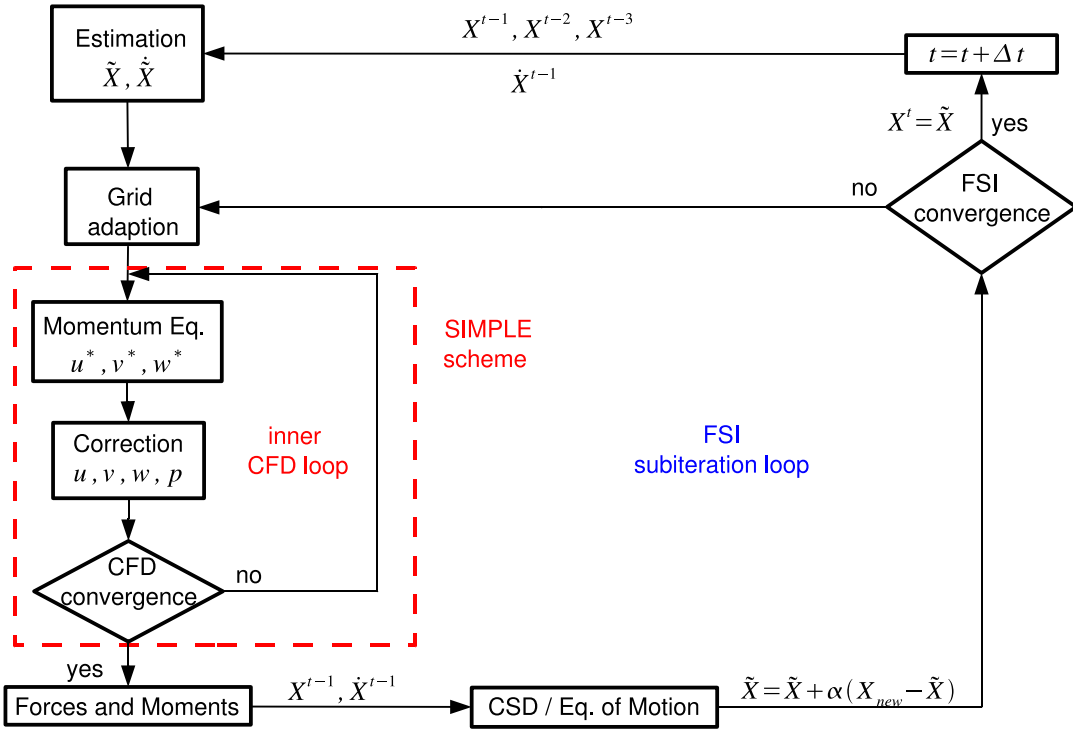
$$\dot{\tilde{X}}^t = 2\dot{X}^{t-1} - \dot{X}^{t-2} \; . \tag{5}$$

Thus the interface between the fluid and the structure, which represents a boundary of the CFD domain, is deformed. According to these estimated values at the boundary, the entire computational grid has to be adapted. Presently, this grid adjustment is performed based on a transfinite interpolation in each FSI–subiteration defined below. It has the advantage to be fast and efficient, but can lead to a deterioration of the grid quality regarding orthogonality and smoothness for larger deformations of the structure. In this case other techniques such as elliptic grid smoothing based on composite mapping as suggested by Spekreijse (1995) and used by Yigit et al. (2007) are more appropriate and will be used in the near future to maintain the grid quality within the coupled simulation.

After the grid adaption the predictor–corrector scheme is applied and a preliminary flow solution is obtained. Based on this solution the corresponding forces and/or moments on the structure are computed. These forces and moments are in general transferred to a computational structure dynamics code (CSD), or in the case of a rigid structure with only one or a few degrees of freedom, to an equation of motion describing the behavior of the structure. The resulting displacement X_{new} of the structure is underrelaxed by an underrelaxation factor α. If an FSI convergence criterion is already fulfilled, i.e., the dynamic equilibrium between fluid and structure is achieved, the computation would go on with the next time step. In this configuration the method described represents a *loose coupling scheme* which is generally only stable for low ratios of the fluid density to the structural density, i.e., a low so-called added mass effect typical for aeroelastic flow problems. Thus alternatives are required if the added mass effect plays a dominant role as it is the case in the examples shown in Sections 3 and 4.
Such an alternative is the scheme sketched in Figure 1(a). If the FSI convergence criterion is not fulfilled after the first sweep, a new grid adaption is done based on the underrelaxed displacement. Subsequently, the corrector step of the predictor–corrector scheme is performed again until mass conservation is accomplished. Afterwards the pressure and shear forces on the structure are recomputed and transferred to the structural solver leading to an update of the displacement X_{new}. This so-called FSI–subiteration loop is performed until the convergence criterion is fulfilled, typically only 3 to 5 times. Presently, the residual of the governing equation of motion has to reach a predefined minimum guaranteeing that the dynamic equilibrium between the fluid and the structure is achieved. In summary, a FSI coupling scheme is designed, which on the one hand is appropriate for an explicit time–stepping scheme and on the other hand avoids instabilities known from loose coupling schemes.
For comparison a coupling scheme for fully implicit time marching is presented in Figure 1(b). Beside the fact that



(a) explicit scheme



(b) implicit scheme

Figure 1: Coupling scheme for (a) an explicit time–marching scheme (presently applied), and for (b) an implicit time–marching scheme shown for comparison (see Glück et al. (2001, 2003)).

the flow is solved with a fully implicit scheme and therefore an inner CFD loop is necessary, both schemes differ with respect to the momentum equations within the FSI–subiteration loop. For the implicit scheme the momentum equations are solved repeatedly in each subiteration sweep. In contrast they are only solved once per time step for the explicit case which strongly reduces the computational effort. Furthermore, the number of FSI–subiterations required to reach the convergence criterion is typically at least one order of magnitude smaller within the explicit scheme compared to the implicit variant. In conclusion, instabilities due to the added mass effect known from loose coupling schemes are avoided by the newly developed coupling scheme. However, the explicit character of the time–stepping scheme is still maintained perceptibly reducing the computational effort.

3 Validation of the Coupling Scheme

3.1 Laminar Flow Around an Elastically Supported Circular Cylinder

The new coupling scheme was tested on laminar and turbulent flows around cylindrical structures. According to the paper of Zhou et al. (1999) the laminar flow around a circular cylinder at Re = 200 (based on the cylinder diameter $D = 6 \cdot 10^{-3}$m and the reference velocity $U_\infty = 0.514$ m/s) for different values of the reduced damping parameter Sg defined below has been computed. A sketch of the configuration including the elastically supported cylinder is given in Figure 2.

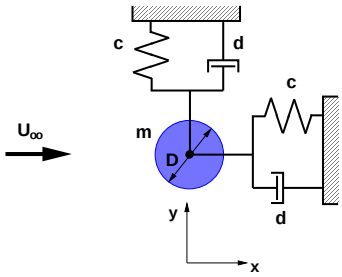


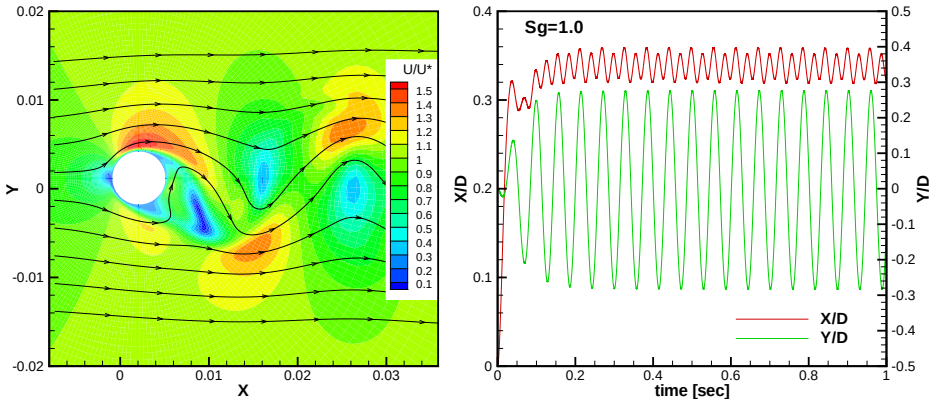
Figure 2: Spring-mass-damper model of the circular cylinder test case at Re = 200.

The flow prediction was performed on a grid with 262,144 control volumes. Since the flow is assumed to be two–dimensional, only 8 CVs are used in the spanwise direction and symmetry boundary conditions are applied. At the inflow a constant undisturbed inflow velocity U_∞ was set. A convective outflow boundary condition and a no–slip boundary condition were prescribed at the outflow and the cylinder surface, respectively. The time step size in this case was $\Delta t = 1.5 \cdot 10^{-5}$ seconds according to a CFL number of 0.2.

For the fixed cylinder case the well–known vortex shedding phenomenon in the wake of the cylinder occurs. The shedding frequency is measured to f_s = 16.7 Hz with a corresponding Strouhal number of $\mathrm{St} = f_s \cdot D/U_\infty \approx 0.2$. Starting from this initial solution for the fixed structure at an arbitrarily chosen time step, the cylinder is released. The movement of the center of the cylinder in x and y direction is described by equations of motion, i.e., a spring-mass-damper model [Eq. (6)], with m denoting the cylinder mass, the damping coefficient d and the spring constant c which are assumed to be identical for both directions.

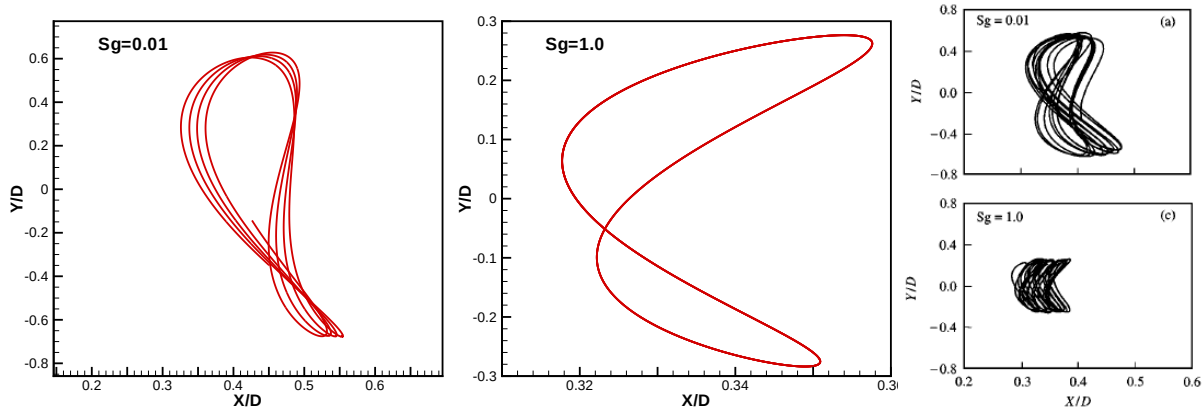
$$\begin{aligned} m\,\ddot{x}(t) + d\,\dot{x}(t) + c\,x(t) &= F_x(t) \\ m\,\ddot{y}(t) + d\,\dot{y}(t) + c\,y(t) &= F_y(t) \end{aligned} \tag{6}$$

Mass, damping and spring parameters are defined according to the reduced damping parameter $Sg = 8\,\pi^2\,St^2\,d^*\,M^*$ with mass ratio M^* and normalized damping parameter d^*. Here, the mass ratio is defined as $M^* = m^*/\rho D^2$ with normalized cylinder mass $m^* = m/l_c$, cylinder diameter D, fluid density ρ and cylinder length l_c. The normalized damping parameter d^* is equal to $d^* = d/l_c$. Spring constant c and cylinder mass m are given from $f_n = \sqrt{c/m}\,1/(2\pi) = 1.3 f_s$ with f_n denoting the natural cylinder frequency. These ordinary differential equations are solved numerically by a classical Runge–Kutta scheme. The flow field of the normalized velocity magnitude for Sg = 1.0 is shown exemplarily in Figure 3(a). Beside the vortex shedding phenomenon downstream of the cylinder a deflection of the recirculation area can be seen. These deflection indicates the movement of the cylinder, here in positive y–direction. Figure 3(b) shows the normalized displacements Δx/D and Δy/D of the cylinder as a function of time. Due to the drag force the cylinder is displaced in x–direction during the transition phase and starts to oscillate around Δx/D = 0.34. In y–direction the cylinder moves around Δy/D = 0. In Figure 4 the results for Sg = 0.01 and Sg = 1.0 are exemplarily shown. As observed by Zhou et al. (1999) the trace in the



(a) (b)

Figure 3: (a) Flow field of the normalized velocity magnitude for Sg = 1.0 and Re = 200; (b) Normalized displacements of the cylinder center as a function of time.



(a) (b) (c)

Figure 4: Normalized displacements of the cylinder center for (a) Sg = 0.01 and for (b) Sg = 1.0 in comparison with the original data set presented by Zhou et al. (1999)(c).

x–y plot describes an asymmetric 8 where the intersection points are found to be off–centered. For Sg = 0.01 the trace slightly changes from cycle to cycle, whereas for Sg = 1.0 nearly but not completely the same trace is observed. The oscillations found with Sg = 0.01 range from $(X/D)_{min}$ = 0.325 to $(X/D)_{max}$ = 0.55 and from $(Y/D)_{min}$ = -0.64 to $(Y/D)_{max}$ = 0.61. The results of Zhou et al. (1999) are limited by $(X/D)_{min}$ = 0.309 and $(X/D)_{max}$ = 0.47 in X-direction and by $(Y/D)_{min}$ = -0.62 and $(Y/D)_{max}$ = 0.56 in Y-direction. With increasing reduced damping parameter Sg the normalized displacements decrease (Note the different scalings of the axes in Figures 4 (a) to (c)). For Sg = 1.0 values of $(X/D)_{min}$ = 0.319, $(X/D)_{max}$ = 0.342, $(Y/D)_{min}$ = -0.29 and $(Y/D)_{min}$ = 0.28 are computed. Zhou et al. (1999) found $(X/D)_{min}$ = 0.283, $(X/D)_{max}$ = 0.386, $(Y/D)_{min}$ = -0.27 and $(Y/D)_{min}$ = 0.27, respectively. Thus, overall the results computed by the new coupling scheme show satisfactory correspondence with the results of Zhou et al. (1999). Here, the agreement of the Y/D values, which fits very well especially for Sg = 1.0, is better than for the X/D values. The deviations maybe generally attributed to the different solution techniques, i.e., the present finite–volume method versus the discrete–vortex method by Zhou et al. (1999), and/or to the unsteadiness of the flow leading to slightly different solutions due to bifurcation effects. Here, the maximal number of FSI subiteration was NFSI = 1.

3.2 Sub–critical Flow Around an Elastically Supported Square Cylinder

In a next step, the coupling scheme was tested for the turbulent flow around an elastically supported square cylinder at Re = $13,000$ as depicted in Figure 5. The Reynolds number is based on the edge length $b = 0.02$ m of the cylinder and the free–stream velocity U_∞ = 10 m/s. For the computation of the flow field a LES prediction with the Smagorinsky model and a Smagorinsky constant of C_S = 0.1 was performed. The grid consisted of 2,228,224 CVs, where 130 CVs are used to discretize the spanwise direction which has an extension of 4 times the edge length b.

At the inflow a velocity block profile was set with U_∞. The outflow is described by a convective outflow condition with $U_c = U_\infty$. In spanwise direction periodic boundary conditions were set. Slip–wall boundary conditions were used on the upper and lower face of the domain. The cylinder itself is defined by a no-slip boundary condition. The movement of the structure was described by the same equations of motion in x and y directions as for the previous test case.

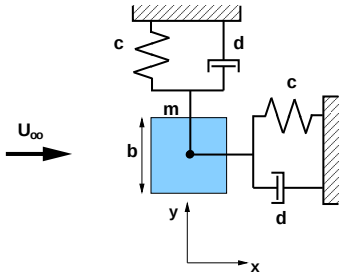


Figure 5: Spring-mass-damper model of the square cylinder test case at Re = $13,000$.

The aim of this computation is the comparison of both coupling schemes presented in Figure 1(a) and (b). It should be mentioned that according to their numerical characteristics both schemes use different time step sizes. In the implicit case, the time step size is set to $\Delta t = 10^{-4}$ seconds leading to a CFL number of 20, whereas the explicit scheme applied $\Delta t = 3 \cdot 10^{-6}$ seconds according to a CFL number of 0.59, which is much more appropriate for LES than the implicit counterpart.

In the sub–critical case considered, the oncoming flow is laminar. The flow separates at the leading and partially at the trailing edges forming free shear layers. Transition to turbulence takes place within the free shear layers. Alternately, vortices are shed from the cylinder. This vortex shedding observed past the cylinder makes the cylinder oscillate in x and y directions. Beside this there is a drag–induced movement of the cylinder in x direction. Figure 6 shows a snapshot of the normalized velocity magnitude $\sqrt{(U_x)^2 + (U_y)^2 + (U_z)^2}/U_\infty$ and the streamlines. The vortex shedding phenomenon is visible in the wake of the structure. The normalized displacement Δx/b and Δy/b

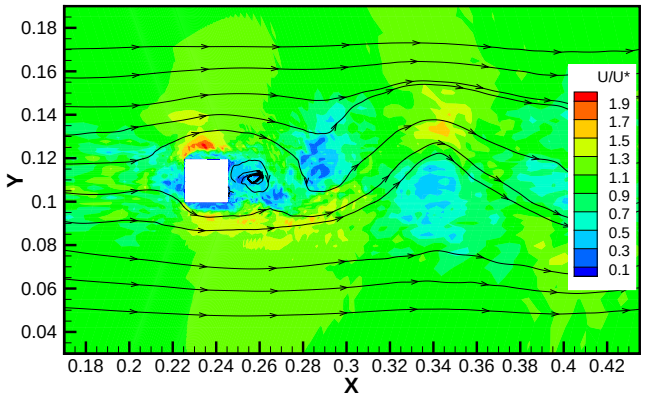


Figure 6: Normalized velocity magnitude and streamlines of the flow around an oscillating cylinder at Re = $13,000$.

are shown in Figure 7. For both time–marching schemes, i.e., both corresponding coupling schemes, a transitional behavior in the Δx/b component is visible which represents the initial excursion after the release of the square cylinder. Thus, starting from Δx/b = 0 and Δy/b = 0 the cylinder is displaced in x direction with an maximal displacement after about 0.1 seconds real time until the displacement reduces and the transitional time interval ends at about 0.5 seconds. Owing to the vortex shedding cycle the Δy/b component oscillates around Δy/b = 0.

Comparing the result for the two time–stepping schemes a perfect match of the displacement values is not visible. The x-displacement seems to be slightly overdetermined by the explicit time–marching scheme compared to the implicit method. Furthermore, the y-component only corresponds in the displacement magnitudes but not in their time histories. However, in general a perfect fit cannot be expected here. The differences are attributed to different starting time steps of the fluid–structure interaction within the LES and especially to the non–deterministic nature of turbulence.

For further investigations a Fast–Fourier–Transformation (FFT) was performed for the displacement data sets of Figure 7. For this purpose, the transitional period was neglected and displacements for time instants greater

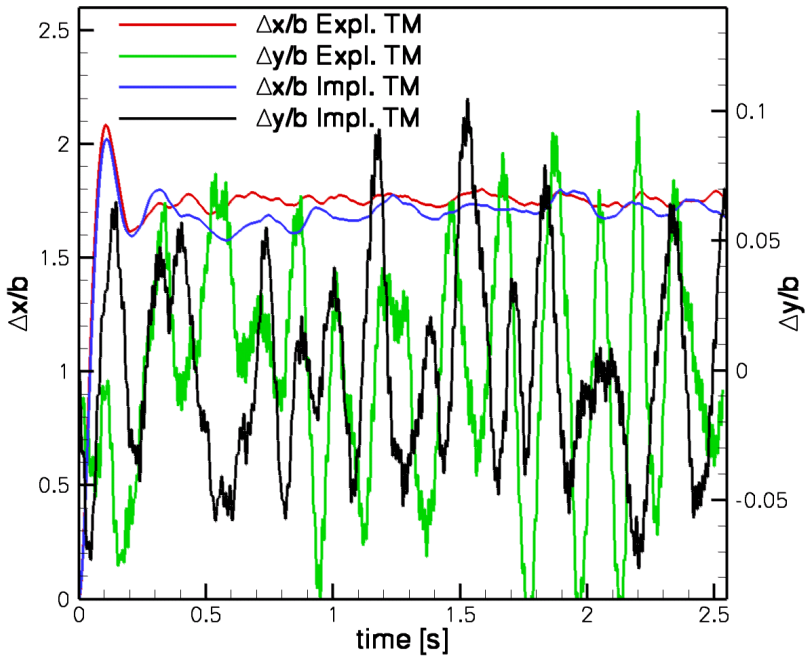


Figure 7: Normalized displacements of the square cylinder for the explicit and the implicit time–marching schemes.

than 0.8 seconds were used for the evaluation. The result for the FFT according to both directions are plotted in Figure 8. In the frequency spectra of the x–displacements no characteristic peak is visible for the data predicted by both time–marching schemes. In the spectra of the y–displacement a peak is visible for f = 69 Hz for the implicit time–marching scheme which equals the well–known shedding frequency of the von Kármán vortices as observed by Kniesburges (2007). This peak can also be found in the data predicted by the explicit time–marching scheme but with a slightly shifted frequency of f = 71 Hz.

In summary, the result for the square cylinder case obtained by different time–marching schemes and accordingly

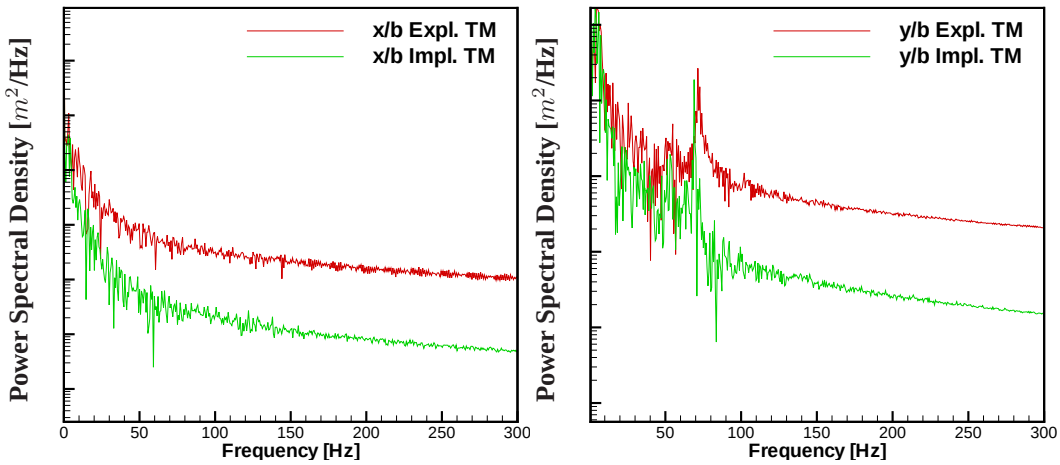


Figure 8: Fast–Fourier–Transformation of the cylinder displacements in (a) x–direction and (b) in y–direction.

with different coupling strategies show good correspondence and confirm the functionality of the developed explicit but strong coupling algorithm within a LES prediction. The maximal number of FSI–subiterations was very low, i.e., NFSI = 3, for the explicit time-stepping algorithm.

4 FSI Prediction of the Swiveling Plate

4.1 Numerical Setup

Besides the above mentioned test cases, this new FSI scheme was evaluated based on simulations of the flow around a swiveling flat plate at Re = $68,000$ for which experimental investigations were carried out at LSTM Erlangen (Gomes et al., 2009). In Figure 9 the computational domain is plotted. A flat plate of 177 mm $\times$ 64 mm $\times$ 2 mm with chord length c = 64 mm is placed in a channel with rectangular cross–section. The plate possesses one rotational degree of freedom around the axis in z–direction located 20 mm from the leading edge.

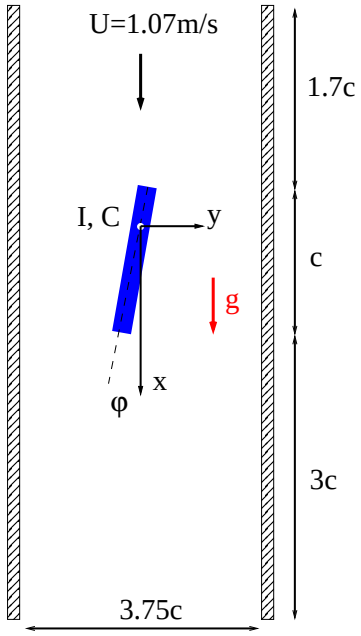


Figure 9: Setting of the computational domain for the channel flow around a flat plate at Re $\approx$ 68,000.

For the flow computation a LES using the Smagorinsky model (C_S = 0.1) with Van Driest damping near solid walls is performed. At the inflow a block velocity profile with U_∞ = 1.07 m/s is set. A convective outflow condition with the convective velocity $U_c = U_\infty$ is applied at the outlet of the domain. For a first computation of the FSI problem, only a section of one chord length of the full spanwise extension of the plate used in the experimental setup is taken into account. This leads to the usage of periodic boundary conditions in spanwise direction and thus the influence of the sidewalls is neglected. For the upper and lower walls of the channel as well as the swiveling plate itself, the no-slip condition is applied. This preliminary simulation was performed on a grid with 434,176 control volumes by resolving the spanwise direction with only 4 control volumes. For the non–deflected plate a wall–normal resolution given by y^+ = 3 for the wall–nearest grid point is applied to resolve the flow at the trailing edge of the plate. The time step size in this case was $\Delta t = 10^{-5}$ seconds. The response of the plate on the outer moment $M_z(t)$ imposed by pressure and shear forces is described by a spring–mass–model leading to an ordinary differential equation for the plate angle φ, which is solved by a classical Runge–Kutta scheme.

$$I\,\ddot{\varphi}(t) + C\,\sin(\varphi(t)) = M_z(t) \tag{7}$$

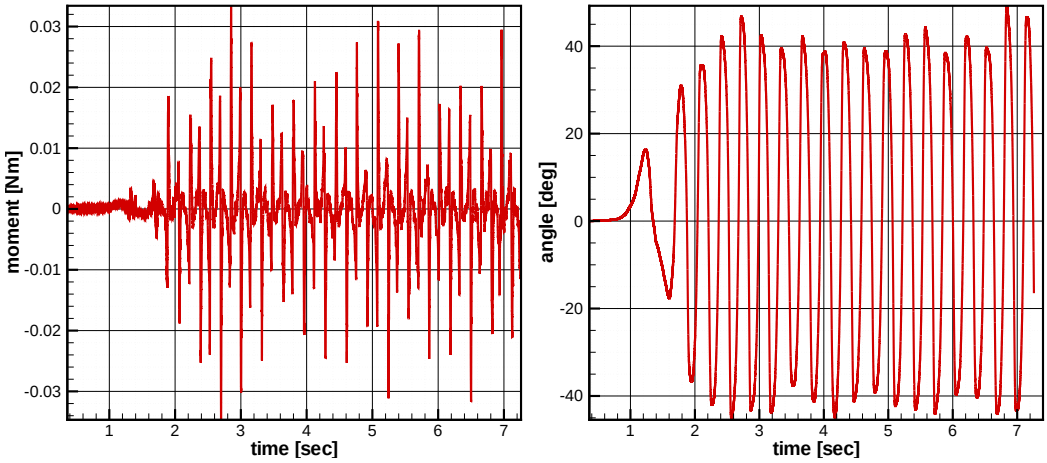
Here, I describes the moment of inertia of the plate and the axle and C is equal to the resulting moment due to gravity and buoyancy forces. The latter results from density difference between the fluid and the structure.

4.2 Results

In Figure 10 the development of the moment $M_z(t)$ and the resulting angle $\varphi(t)$ is shown. For initialization of the flow field, the plate is kept fixed, i.e., φ = 0°, for about 0.36 seconds real time. Then the plate is released and the swiveling motion starts. After a transitional time of approximately 2 seconds amplitudes in a range of φ_{min} = -45° to φ_{max} = 49° are observed. Amplitudes like this make great demands on the grid adaption but can barely be tackled with a transfinite interpolation in this case. Similar to the amplitudes also the frequency of the motion

obtained is not absolutely constant but takes values of about 3.14 Hz. The mean number of FSI–subiteration $\overline{NFSI}$ is equal to five for this case.

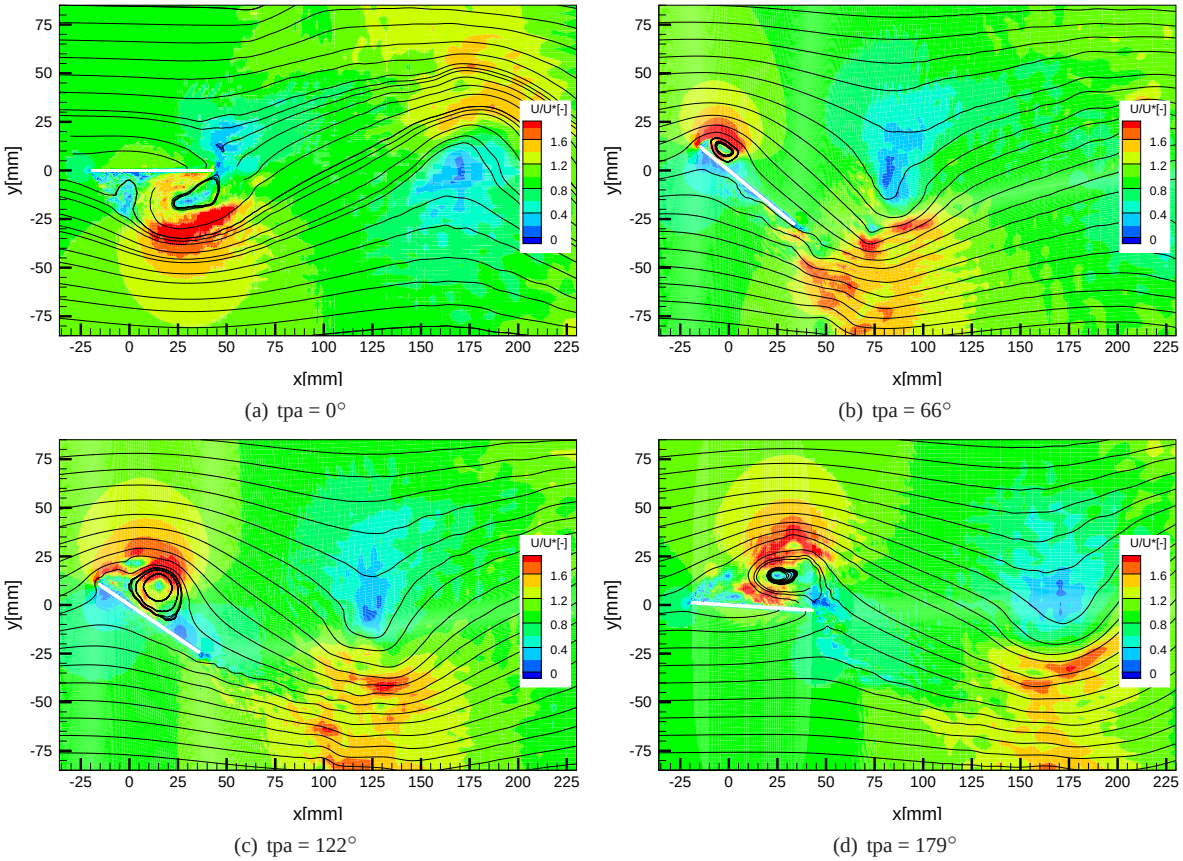
The instantaneous flow fields of the normalized velocity magnitude with the corresponding time-phase resolved



(a) (b)

Figure 10: (a) Computed angular moment acting on the plate as a function of time. (b) Resulting angular displacement of the plate.

plate angles for time-phase angles of 0°, 66°, 122° and 179° are shown in Figure 11. The time-phase angle is defined as tpa = t/T · 360° with T denoting the swiveling period and t describing the passed real time within the period. Thus the figure depicts half of a period of a full cycle.



(a) tpa = 0° (b) tpa = 66°

(c) tpa = 122° (d) tpa = 179°

Figure 11: Computed instantaneous velocity field and plate deflection at four different instants of the swiveling motion period at Re = 68,000.

Starting from a plate angle of 0° the plate starts to deviate from its initial position. With increasing plate angle also the velocity at the leading edge on the suction side of the plate increases and a vortex generates which can be seen in Figure 11(b). The vortex travels downstream (Figure 11(c)) and with it a zone of low pressure. Passing the center of rotation this vortex, i.e., the corresponding low pressure region, together with the higher pressure on the pressure side of the plate reduces the rotational speed of the plate and finally changes the direction of rotation. Then a similar cycle starts at the opposite side of the plate.
When comparing the computed results with the experimentally measured data, differences become apparent. The maximal values of the amplitudes are $\pm 27°$ and also the frequency with f = 2.49 Hz is clearly lower than the computed one. Possible reasons for the deviations observed might be the too coarse (spanwise) resolution or the restriction to a section of the rigid plate. Nevertheless, when comparing the velocity fields of a full cycle the computational and the experimental results show good qualitative correspondence concerning the spots of high and low velocity magnitudes (Gomes et al., 2009). A more detailed comparison has to be carried out when a coupled simulation without the restrictions mentioned above has been performed.

5 Conclusions and Outlook

FSI in turbulent flows requires advanced techniques such as LES for a reliable flow prediction. To resolve the turbulent structures in time, small time–step sizes are used for LES which makes explicit time–marching schemes such as predictor–corrector schemes favorable. To conserve the character of this explicit time–marching scheme on the one hand and to obtain a stable FSI algorithm on the other hand, a coupling scheme was proposed in which the corrector step and the structural computation are directly connected in a FSI–subiteration loop. Due to this feature, the dynamic equilibrium between the fluid and the structure is achieved within the explicit time–marching scheme. Compared to the implicit variant, the explicit scheme seems to be more efficient and less CPU–time consuming, but a detailed verification of this impression still has to be done.
Results of this coupling scheme have been presented for the flow around cylindrical structures for laminar (Re = 200) and sub–critical (Re = 13,000) flows. For the laminar case the result showed good compliance to the values presented by Zhou et al. (1999). The sub–critical test case was used to compare the results obtained by a classical fully implicit scheme (see Glück et al. (2001, 2003)) with the proposed explicit coupling scheme. Satisfying results were obtained with respect to the displacement magnitudes and frequencies of the structure oscillation.
A challenging test case was performed for the flow around a loosely mounted plate at Re = 68,000. Here, the results presently deviate from the experimental ones. As a consequence previously introduced simplifications of the computational case with respect to the experimental one have to be reconsidered, i.e., the computational domain has to be extended in spanwise direction and the effect of sidewalls has to be taken into account. That also includes that the spanwise resolution has to be increased.
In a next step, a study is thought to be done for more complex geometries like nominally two–dimensional flexible structures. For this purpose, the coupling with a CSD–code has to be realized. In addition, a more powerful grid adaption has to be introduced such as an elliptic grid smoothing algorithm proposed by Spekreijse (1995) for example. This has not only to be done for tackling more complex displacements of a flexible structure. More important is to maintain the grid quality within the FSI algorithm since the demands for LES are high (see Breuer and Münsch (2008b)).

Acknowledgement *The project is financially supported by the Deutsche Forschungsgemeinschaft within the research group* 'Fluid–Struktur–Wechselwirkung: Modellierung, Simulation, Optimierung' (FOR 493) *under contract number BR 1847/6.*

References

Breuer, M.: Large–eddy simulation of the sub-critical flow past a circular cylinder: Numerical and modeling aspects. *Int. Journal for Numerical Methods in Fluids*, 28, (1998), 1281–1302.

Breuer, M.: A challenging test case for large–eddy simulation: High Reynolds number circular cylinder flow. *Int. Journal of Heat and Fluid Flow*, 21, 5, (2000), 648–654.

Breuer, M.: *Direkte Numerische Simulation und Large–Eddy Simulation turbulenter Strömungen auf Hochleistungsrechnern*. Habilitationsschrift, Universität Erlangen–Nürnberg, Berichte aus der Strömungstechnik, ISBN 3-8265-9958-6, Shaker Verlag, Aachen (2002).

Breuer, M.; Münsch, M.: Fluid–structure interaction using LES – a partitioned coupled predictor–corrector scheme. In: *78th Annual GAMM Meeting, Universität Bremen, Germany, March 31 – April 4, 2008*, Proceedings in Applied Mathematics and Mechanics, WILEY–VCH Verlag GmbH & Co. KGaA, Weinheim (2008a).

Breuer, M.; Münsch, M.: LES meets FSI – important numerical and modeling aspects. In: *Seventh Int. ERCOFTAC Workshop on DNS and LES: DLES-7, Trieste, Italy, Sept. 8–10, 2008*, ERCOFTAC Series, Springer Science (2008b).

Demirdžić, I.; Perić, M.: Space conservation law in finite–volume calculations of fluid flow. *Int. Journal for Numerical Methods in Fluids*, 8, (1988), 1037–1050.

Demirdžić, I.; Perić, M.: Finite–volume method for prediction of fluid flows in arbitrarily shaped domains with moving boundaries. *Int. Journal for Numerical Methods in Fluids*, 10, (1990), 771–790.

Durst, F.; Schäfer, M.: A parallel block–structured multigrid method for the prediction of incompressible flows. *Int. Journal for Numerical Methods in Fluids*, 22, (1996), 549–565.

Durst, F.; Schäfer, M.; Wechsler, K.: Efficient simulation of incompressible viscous flows on parallel computers. *Notes on Numerical Fluid Mechanics*, 52, (1996), 87–101, Vieweg Verlag, Braunschweig.

Ferziger, J. H.; Perić, M.: *Computational Methods for Fluid Dynamics*. Springer Verlag, Berlin (2002).

Germano, M.; Piomelli, U.; Moin, P.; Cabot, W.: A dynamic subgrid–scale eddy viscosity model. *Phys. of Fluids A*, 3, 7, (1991), 1760–1765.

Glück, M.; Breuer, M.; Durst, F.; Halfmann, A.; Rank, E.: Computation of fluid–structure interaction on lightweight structures. *Int. Journal of Wind Engineering and Industrial Aerodynamics*, 89, 14–15, (2001), 1351–1368.

Glück, M.; Breuer, M.; Durst, F.; Halfmann, A.; Rank, E.: Computation of wind–induced vibrations of flexible shells and membranous structures. *Journal of Fluids and Structures*, 17, (2003), 739–765.

Gomes, J.; Münsch, M.; Breuer, M.; Lienhart, H.: Flow–induced oscillation of a flat plate — a fluid–structure interaction study using experiment and LES. In: *16. DGLR–Fach–Symposium der STAB, RWTH Aachen, Germany, Nov. 3–5, 2008*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, in press (2009).

Kniesburges, S.: *Simulation der Fluid-Struktur-Kopplung bei einer starren und einer flexiblen schwingenden Struktur*. Diplomarbeit, Universität Erlangen–Nürnberg, Lehrstuhl für Strömungstechnik (2007).

Lilly, D.: A proposed modification of the Germano subgrid–scale closure method. *Phys. of Fluids A*, 4, 3, (1992), 633–635,.

Smagorinsky, J.: General circulation experiments with the primitive equations I, the basic experiment. *Mon. Weather Rev.*, 91, (1963), 99–165.

Spekreijse, S. P.: Elliptic grid generation based on Laplace equations and algebraic transformations. *Journal of Computational Physics*, 118, (1995), 38–61.

Yigit, S.; Schäfer, M.; Heck, M.: Numerical investigation of structural behavior during fluid excited vibrations. *REMN*, 16, (2007), 491–519.

Zhou, C. Y.; So, R.; Lam, K.: Vortex–induced vibrations of an elastic circular cylinder. *Journal of Fluids and Structures*, 13, (1999), 165–189.

Address: M. Breuer (corresponding author), Department of Fluid Mechanics, Institute of Mechanics, Helmut–Schmidt–University Hamburg, Holstenhofweg 85, D–22043 Hamburg, Germany. email: `breuer@hsu-hh.de`
M. Münsch, Institute of Fluid Mechanics, University of Erlangen–Nürnberg, Cauerstr. 4, D–91058 Erlangen, Germany. email: `mmuensch@lstm.uni-erlangen.de`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 43– 54
Herrsching am Ammersee, 29.9.-1.10.2008

An Overset-Grid Strategy for Aeroacoustics and Aeroelasticity of Moving Bodies

F. Daude, P. Lafon, F. Crouzet and C. Bailly

A high-order finite-difference algorithm is proposed in the aim of LES for Computational Aeroacoustics (CAA) and Aeroelasticity applications. The subgrid scale dissipation is performed by the explicit high-order numerical filter used for numerical stability purpose. In order to tackle complex geometries and moving grids, while preserving grid quality, an overset-grid approach is used. High-order interpolations make it possible to ensure the communication between overlapping domains. The whole algorithm is validated on canonical flow problems to illustrate its capability to preserve accuracy for moving configurations.

1 Introduction

In a wide range of technical fields such as aircrafts, automotive engineering, trains, turbomachinery, power plants, non-linear interactions between the turbulent flow and the acoustic fields produce undesirable high pressure levels, see Colonius and Lele (2004). They are sources of noise pollution which is a major environmental issue. The radiated noise can also induce vibrations and damages. This is particularly the case in confined flows. In addition, turbulent flows or acoustic waves can couple with moving structures involving fluid/structure interaction (FSI). The energy industry has to deal with many FSI phenomena ranging from Vortex-Induced Noise or Vortex-Induced Vibrations (VIV) to aeroelasticity. The related applications are respectively cable aeolian tones, tube bundle vibrations, see Longatte et al. (2003), or blade flutter, see Crouzet (2006).

In many such coupled configurations, the calculation of both the unsteady flow and the radiated sound must be performed in the same computation. This is referred as Direct Noise Computation (DNC) in the literature, see Bailly et al. (2008). Using DNC is an efficient way to identify the fluid mechanism contributing to the sound production and therefore, a useful tool to reduce noise radiation. The feasibility of DNC is now demonstrated in the literature via Direct Numerical Simulation (DNS), see Colonius et al. (1997), Freund (2001), Gloerfelt et al. (2003), and Large-Eddy Simulation (LES), see Bodony and Lele (2005), Bogey and Bailly (2006, 2007), Emmert et al. (2007, 2008).

Application of compressible LES to computational aeroacoustics (CAA) problems makes it possible to tackle applications with industrial or practical relevance. The large disparity in the characteristic scales of the acoustic and the flow fluctuations, and the need to accurately resolve high wavenumber fluctuations require the use of numerical methods with minimal dissipation and dispersion errors, see Colonius and Lele (2004). In this context, the Dispersion-Relation-Preserving (DRP), see Tam and Webb (1993), or optimized high-order finite-difference schemes in conjunction with selective filter, see Bogey and Bailly (2004), are an attractive choice for LES to reduce both amplitude and phase numerical errors.

For moving grids, the mainly used method is the classical ALE method associated with deforming and/or remeshing procedures. In order to allow body displacements while preserving grid quality, the overset-grid (Chimera) method is best suited. It is based on different body-fitted overlapping grids associated with interpolation procedures for the communication between the different component grids. This method also makes it possible to tackle complex geometries on fixed or moving grids. In this context, a new numerical code called *Code_Safari* (Simulation of Aeroacoustic Flows And Resonance and Interaction) has been developed to handle industrial configurations. To maintain the high accuracy of the algorithm, the communication between non-coincident grids are

made by high-order interpolation, see Delfs (2001), Sherer and Scott (2005), Desquesnes et al. (2006).

2 Governing equations

2.1 Fluid dynamics

The three-dimensional Navier-Stokes equations are expressed in Cartesian coordinates for a viscous compressible Newtonian fluid. After the application of a general time-dependent curvilinear transformation $(x, y, z, t) \rightarrow (\xi, \eta, \zeta, \tau)$, see Viviand (1974) and Vinokur (1974), these equations are written in the following strong conservative form:

$$\partial_\tau \hat{U} + \partial_\xi \Big(E - E^\nu \Big) + \partial_\eta \Big(F - F^\nu \Big) + \partial_\zeta \Big(G - G^\nu \Big) = 0. \tag{1}$$

with $\hat{U} = U/J$ where $U = (\rho, \rho u, \rho v, \rho w, \rho e)$ is the vector of conservative variables, ρ is the density, u, v and w are the Cartesian velocity components of the vector $\vec{V}$, e is the total specific energy:

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \left(u^2 + v^2 + w^2 \right),$$

where p is the pressure, γ the specific heat ratio and J the Jacobian of the coordinate transformation $(x, y, z) \rightarrow (\xi, \eta, \zeta)$. E, F and G are the inviscid flux-vectors which can be expressed as:

$$E = \xi_t \hat{U} + \frac{1}{J} \begin{pmatrix} \rho\Theta_\xi \\ \rho u\Theta_\xi + p\xi_x \\ \rho v\Theta_\xi + p\xi_y \\ \rho w\Theta_\xi + p\xi_z \\ (\rho e + p)\,\Theta_\xi \end{pmatrix}, \quad F = \eta_t \hat{U} + \frac{1}{J} \begin{pmatrix} \rho\Theta_\eta \\ \rho u\Theta_\eta + p\eta_x \\ \rho v\Theta_\eta + p\eta_y \\ \rho w\Theta_\eta + p\eta_z \\ (\rho e + p)\,\Theta_\eta \end{pmatrix}, \quad G = \zeta_t \hat{U} + \frac{1}{J} \begin{pmatrix} \rho\Theta_\zeta \\ \rho u\Theta_\zeta + p\zeta_x \\ \rho v\Theta_\zeta + p\zeta_y \\ \rho w\Theta_\zeta + p\zeta_z \\ (\rho e + p)\,\Theta_\zeta \end{pmatrix}.$$

The contravariant velocity components Θ_ξ, Θ_η and Θ_ζ are defined as:

$$\Theta_\xi = u\xi_x + v\xi_y + w\xi_z, \quad \Theta_\eta = u\eta_x + v\eta_y + w\eta_z \quad \text{and} \quad \Theta_\zeta = u\zeta_x + v\zeta_y + w\zeta_z.$$

The quantities ξ_t, η_t and ζ_t are the time metrics; $\xi_x, \xi_y, \xi_z, \eta_x, \eta_y, \eta_z, \zeta_x, \zeta_y$ and ζ_z designate the spatial metrics. The subscripts denote the partial derivatives. E^ν, F^ν and G^ν are the viscous flux-vectors. Their expression are the same as in the case of time-invariant generalized coordinates, see Marsden et al. (2005) and Suh et al. (2006).

2.2 Geometrical conservation

With the strong-conservation form in Equation (1), the following relations must be satisfied numerically to ensure free-stream preservation when a finite-difference discretization is used, see Visbal and Gaitonde (2002):

$$\begin{cases} \left(\frac{1}{J}\xi_x\right)_\xi + \left(\frac{1}{J}\eta_x\right)_\eta + \left(\frac{1}{J}\zeta_x\right)_\zeta = 0 \\ \left(\frac{1}{J}\xi_y\right)_\xi + \left(\frac{1}{J}\eta_y\right)_\eta + \left(\frac{1}{J}\zeta_y\right)_\zeta = 0 \\ \left(\frac{1}{J}\xi_z\right)_\xi + \left(\frac{1}{J}\eta_z\right)_\eta + \left(\frac{1}{J}\zeta_z\right)_\zeta = 0 \\ \left(\frac{1}{J}\right)_\tau + \left(\frac{1}{J}\xi_t\right)_\xi + \left(\frac{1}{J}\eta_t\right)_\eta + \left(\frac{1}{J}\zeta_t\right)_\zeta = 0 \end{cases} \tag{2}$$

The last relation only concerns time-dependent meshes and is called the geometric conservation law (GCL), see Thomas and Lombard (1979). In order to satisfy the numerical metric error cancellation and to ensure the free-stream preservation, the spatial metrics are expressed in the conservative form proposed by Thomas and Lombard

(1979):

$$\left\{\begin{array}{rcl} \dfrac{1}{J}\xi_x &=& (y_\eta z)_\zeta - (y_\zeta z)_\eta \\ \dfrac{1}{J}\xi_x &=& (y_\zeta z)_\xi - (y_\xi z)_\zeta \\ \dfrac{1}{J}\xi_x &=& (y_\xi z)_\zeta - (y_\eta z)_\xi \end{array}\right. \tag{3}$$

Time metrics are used for moving/deforming grid computations. Their expression are given in the next section.

2.3 Application to moving grids

To tackle moving/deforming grid configurations, the respect of GCL is a key issue to enforce metric cancellation and free-stream preservation. In this aim, non-conservative corrector terms are used to ensure the GCL identity, as proposed by Visbal and Gaitonde (2002). In practice, the time derivative in Equation (1) is split into two parts and the second term is evaluated using the GCL condition. And, finally, the following equation is obtained:

$$\partial_\tau U + \underbrace{J\left\{\partial_\xi\Big(E-E^\nu\Big)+\partial_\eta\Big(F-F^\nu\Big)+\partial_\zeta\Big(G-G^\nu\Big)-U\left[\partial_\xi\Big(\frac{\xi_t}{J}\Big)+\partial_\eta\Big(\frac{\eta_t}{J}\Big)+\partial_\zeta\Big(\frac{\zeta_t}{J}\Big)\right]\right\}}_{\mathcal{R}}=0 \tag{4}$$

In addition, the time metrics are evaluated using the grid velocity $X_\tau = (x_\tau, y_\tau, z_\tau)^T$ via the following relations:

$$\left\{\begin{array}{rcl} \dfrac{\xi_t}{J} &=& -\Big(x_\tau\dfrac{\xi_x}{J}+y_\tau\dfrac{\xi_y}{J}+z_\tau\dfrac{\xi_z}{J}\Big) \\ \dfrac{\eta_t}{J} &=& -\Big(x_\tau\dfrac{\eta_x}{J}+y_\tau\dfrac{\eta_y}{J}+z_\tau\dfrac{\eta_z}{J}\Big) \\ \dfrac{\zeta_t}{J} &=& -\Big(x_\tau\dfrac{\zeta_x}{J}+y_\tau\dfrac{\zeta_y}{J}+z_\tau\dfrac{\zeta_z}{J}\Big) \end{array}\right. \tag{5}$$

These relations are similar to the classical ALE (Arbitrary Lagrangian Eulerian) expression in a finite-volume framework.

3 Numerical method

3.1 Discretization

First derivatives at interior grid points are determined using the optimized 11-point centered finite-difference scheme proposed by Bogey and Bailly (2004):

$$\partial_\xi E_{i,j,k} \approx \frac{1}{\Delta\xi}\sum_{m=1}^{5} s_m\Big(E_{i+m,j,k}-E_{i-m,j,k}\Big). \tag{6}$$

This non-dissipative scheme is optimized in the wavenumber space to reduce the dispersion error following the idea of Tam and Webb (1993). The linear analysis shows that this scheme is able to resolve accurately perturbations with only four points per wavelength such as shown in Figure 1 (a). The same scheme has been applied successfully for the direct computation of jet noise using LES, see Bogey and Bailly (2006, 2007).

The time integration is performed with the classical explicit four-stage Runge-Kutta scheme (RK4) yielding:

$$U^{(l)}_{i,j,k} = U^n_{i,j,k} - \Delta\tau\beta^{(l)}\mathcal{R}^{(l-1)}_{i,j,k} \quad \forall l \in \{1,...,4\} \tag{7}$$

with $U^{(0)} = U^n$ and $\mathcal{R}_{i,j,k}$ the discretization of the residual $\mathcal{R}$. The Courant-Friedrich-Lewy number in the ξ-direction is defined by:

$$\mathsf{CFL}_\xi = \frac{\Delta t\left(|\xi_t+\Theta_\xi|+c||\vec{\nabla}\xi||\right)}{\Delta\xi} \tag{8}$$

and the numerical stability requires to satisfy the following relation:

$$\mathsf{CFL} = \max(\mathsf{CFL}_\xi, \mathsf{CFL}_\eta, \mathsf{CFL}_\zeta) \leq 1$$

In the same way, the mesh displacement is linked to a new stability requirement. In order to introduce this new stability constraint, a 1-D model is considered. In the physical space, the time and spatial variables are independent which is equivalent to a non-linear advection equation in the computational domain using the chain-derivative rules:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 0 \quad \Longleftrightarrow \quad \partial_\tau x + \xi_t \partial_\xi x = 0$$

For this equation, the stability constraint is based on the ratio $C_\xi = \dfrac{|\xi_t|\Delta\tau}{\Delta\xi}$. According to Equations (5) and (8), it follows that:

$$\frac{|d_\xi|}{\Delta\xi}||\vec{\nabla}\xi|| < \mathsf{CFL}_\xi \tag{9}$$

where $d_\xi = \vec{V}_e.\vec{\nabla}\xi/||\vec{\nabla}\xi||$ is the displacement in the ξ-direction. Thus, the mesh displacement is limited and the maximal allowed value is driven by the CFL value linked by the fluid dynamics.

In order to ensure the synchronization between the flow variables and the grid coordinates, the RK4 scheme is also used for the grid motion:

$$X^{(l)}_{i,j,k} = X^n_{i,j,k} + \Delta\tau\beta^{(l)}(X_\tau)^{(l-1)}_{i,j,k} \quad \forall l \in \{1, ..., 4\} \tag{10}$$

with $X = (x, y, z)$, $X^{(0)} = X^n$ and $X^{n+1} = X^{(4)}$.

After the application of the Runge-Kutta scheme, the explicit optimized 11-point spatial low-pass filter proposed by Bogey and Bailly (2004) is used to remove spurious high-frequency spatial oscillations:

$$W^{(5)}_{i,j,k} = W^{(4)}_{i,j,k} - \sigma_f \left[F^\xi\left(W^{(4)}_{i,j,k}\right) + F^\eta\left(W^{(4)}_{i,j,k}\right) + F^\zeta\left(W^{(4)}_{i,j,k}\right)\right] \tag{11}$$

where

$$F^\xi\left(W^{(4)}_{i,j,k}\right) = d_0 W^{(4)}_{i,j,k} + \sum_{m=1}^{5} d_m \left(W^{(4)}_{i+m,j,k} + W^{(4)}_{i-m,j,k}\right)$$

with $0 \leq \sigma_f \leq 1$ for the filtering strength ; and $W = (\rho, \rho u, \rho v, \rho w, p)^T$.

This filter is optimized in the wavenumber space: the linear analysis shows that this filter only damps the perturbations not accurately resolved by the spatial scheme of Equation (6) as shown in Figure 1 (b).

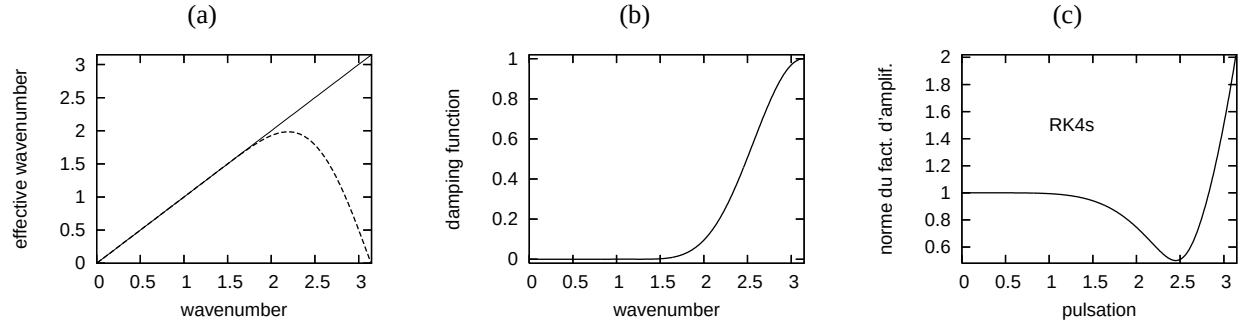


Figure 1: (a) Comparison between the exact – and the effective - - wavenumber of the spatial discretization; (b) Damping function of the selective filter as a function of the wavenumber $k\Delta x$; (c) Dissipative characteristic of the RK4 scheme as a function of the pulsation $\omega\Delta t$.

3.2 LES strategy

The LES strategy used in the present work is the same as the one employed by Bogey and Bailly (2006, 2007) and by Rizzetta et al. (2003). The compressible LES formalism of Vreman et al. (1995), is retained to express the filtered equations in conservative form. The selective filter used to improve the numerical stability of the centered non-dissipative spatial discretization is also employed to separate the large scales from the small ones. In addition, this linear filter takes into account the dissipative effects of the subgrid scales by draining energy at the cut-off frequency. Indeed, the selective filter leaves flow features larger than the cut-off wavelengths unaffected, while properly removing the energy being transferred to smaller wave lengths. In addition, the interactions between the resolved and the unresolved scales are neglected. Thus, no additional explicit subgrid scale model is used.

3.3 Linear Analysis

The von Neumann method is used to analyze the damping and dispersive properties of the algorithm presented previously. This analysis is only applied on linear equations with periodic boundary conditions. For non-linear equations, the results obtained with the linear analysis are not sufficient. However, linear stability is a necessary condition for non-linear problems, see Hirsch (1988).

The von Neumann method is applied to the global algorithm (spatial, temporal discretizations and low-pass filter) for the following linear advection equation:

$$\partial_t u + a\partial_x u = 0 \tag{12}$$

The algorithm can be decomposed into three steps as:

$$\begin{cases} R_i(u) = \dfrac{a}{\Delta x}\displaystyle\sum_{m=1}^{5} s_m(u_{i+m} - u_{i-m}) & \text{(spatial discretization)} \\ u_i^{(l)} = u_i^n - \Delta t \beta^{(l)} R_i(u^{(l-1)}) \qquad \forall l \in \{1,...,4\} & \text{(time discretization)} \\ u_i^{n+1} = u_i^{(4)} - \sigma_f \Big[d_0 u_i^{(4)} + \displaystyle\sum_{m=1}^{5} d_m \left(u_{i+m}^{(4)} + u_{i-m}^{(4)} \right) \Big] & \text{(low-pass filter)} \end{cases}$$

with $u_i^{(0)} = u_i^n$.

The von Neumann method is based on the Fourier transform. We consider a single harmonic $u_i^n = \hat{u}^n e^{Iik\Delta x}$ with $\hat{u}^n$ the amplitude, $k\Delta x$ the phase angle corresponding to the wavenumber k and $I^2 = -1$. In order to evaluate the algorithm amplification factor defined as $g = \hat{u}^{n+1}/\hat{u}^n$, the Fourier transform is applied to the three stages of the computation:

$$\begin{cases} \hat{R}(u) = I\dfrac{a}{\Delta x}\hat{u}k^*\Delta x & \text{with} \quad k^*\Delta x = 2\displaystyle\sum_{m=1}^{5} s_m \sin(mk\Delta x) & \text{(spatial discret.)} \\ \hat{u}^{(4)} = \left(1 + \displaystyle\sum_{l=1}^{4} \gamma_l (-\Delta t I \frac{a}{\Delta x} k^* \Delta x)^l\right) \hat{u}^n & \text{with} \quad \gamma_l = \displaystyle\prod_{q=4-l+1}^{4} \beta^{(q)} & \text{(time discret.)} \\ \hat{u}^{n+1} = (1 - \sigma_f \hat{D})\hat{u}^{(4)} & \text{with} \quad \hat{D} = d_0 + 2\displaystyle\sum_{m=1}^{5} d_m \cos(mk\Delta x) & \text{(low-pass filter)} \end{cases}$$

Finally, the amplification factor of the global algorithm can be written as:

$$g = (1 - \sigma_d \hat{D}) \left(1 + \sum_{l=1}^{4} \gamma_l (-I\sigma k^* \Delta x)^l\right) \tag{13}$$

with the CFL number $\sigma = \dfrac{a\Delta t}{\Delta x}$.

The amplification factor g which can be rewritten as $g = |g|e^{I\phi}$ is now compared with the exact factor: $g_{\text{ex}} = e^{-I\sigma k\Delta x}$. The algorithm damping property is given by the norm $|g|$ and the dispersive one by the relative phase error: $\phi + \sigma k\Delta x$. The results with $\mathsf{CFL} = 1$ and $\sigma_f = 0.2$ are displayed in Figure 2. With respect to the

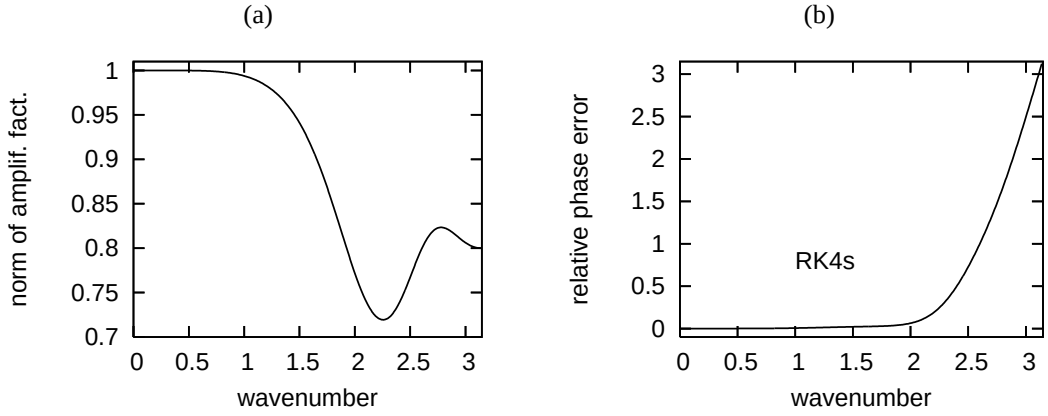


Figure 2: Damping and dispersion errors as a function of the wavenumber $k\Delta x$: (a) Norm of the amplification factor $|g|$; (b) relative phase error $\phi + \sigma k\Delta x$.

damping character of the spatial scheme and the linear filter presented in Figure 1, by taking $\mathsf{CFL} = 1$, the explicit time integration damages the upper bound of the range of well-resolved wavenumber: $k\Delta x \leq \pi/2$. To known quantitatively the accuracy domain of the global algorithm, an accuracy limit is estimated from the following arbitrary criterion:

$$|1 - \mathcal{H}| \leq 5 \times 10^{-4} \tag{14}$$

with the ratio $\mathcal{H} = g/g_{\text{ex}}$. The accuracy domain of the global algorithm is thus reduced to $0 \leq k\Delta x \leq 0.65$, that is to say in term of number of points per wavelength: $\lambda_a/\Delta x \approx 9.66$.

3.4 Boundary conditions

3.4.1 Wall boundaries

In order to preserve low-dissipation and low-dispersion properties near wall boundaries, 11-point non-centered finite-difference schemes in conjunction with explicit 11-point non-centered low-pass filter proposed by Berland et al. (2007) are used. These two procedures are optimized in the wavenumber space to recover the bandwidth properties of the centered ones in Equations (6) and (11). However, the non-centered schemes suffer from numerical instability. Therefore, in the case of strong flow gradients near wall boundaries, explicit centered filtering of lower order can optionally be used to ensure this numerical stability.

3.4.2 Non-reflecting boundary conditions

Inlet and outlet boundary conditions are based on the Thompson's characteristic boundary conditions, see Thompson (1990). The conditions are supposed to locally be one-dimensional and inviscid. Then, the convective terms in the boundary-normal direction are split into several waves with different characteristic velocities. Finally, the unknown incoming waves are expressed as a function of known outgoing waves. The 3-D far-field radiation boundary conditions generalized by Bogey and Bailly (2002) are applied on the boundaries only reached by acoustic perturbations.

4 Extension to complex geometries

The high-order finite-difference algorithm satisfying conservation laws on generalized coordinates are limited to cylindrical geometries. In order to go beyond this limit, overset-grid techniques are used with high-order interpolation procedure to preserve the high-order spatial accuracy, see Delfs (2001), Sherer and Scott (2005), Desquesnes et al. (2006). This is addressed in the following.

4.1 Overset-grid strategy and high-order interpolation

In order to handle complex configurations as those including multiple bodies, the high-order algorithm presented in the previous sections is extended to general overset-grid topologies. In practice, *Code_Safari* is interfaced with the freely available *Overture* library developed by the Lawrence Livermore National Laboratory, see Henshaw (1998). The mesh including different component grids are given by *Overture*. In addition, the interpolation data such as overlapping zones, interpolation stencils and offsets are generated with *Overture*.

In the overset-grid approach, points of the different overlapping regions are non coincident. Therefore, the communication between overlapping component grids is performed with high-order interpolation. Following Sherer and Scott (2005), high-order explicit non-optimized Lagrangian polynomials are used to perform the interpolation stage. The interpolation process is performed in the computational domain (ξ, η, ζ, τ) as in Figure 3. The evaluation of the variable ϕ at the point P is performed via the interpolation of ϕ at P as:

$$\phi_P \approx \sum_{i=0}^{M_\xi - 1} \sum_{j=0}^{M_\eta - 1} L_i^\xi L_j^\eta \phi_{I_Q+i, J_Q+j}. \tag{15}$$

where M_ξ and M_η are the interpolation stencil length in the ξ- and η-direction respectively. Q is the first donor point of the interpolation stencil (in green in Figure 3) and its coordinates are (I_Q, J_Q) in the computational domain. L_i^ξ and L_j^η are the Lagrangian coefficients in the two directions defined as:

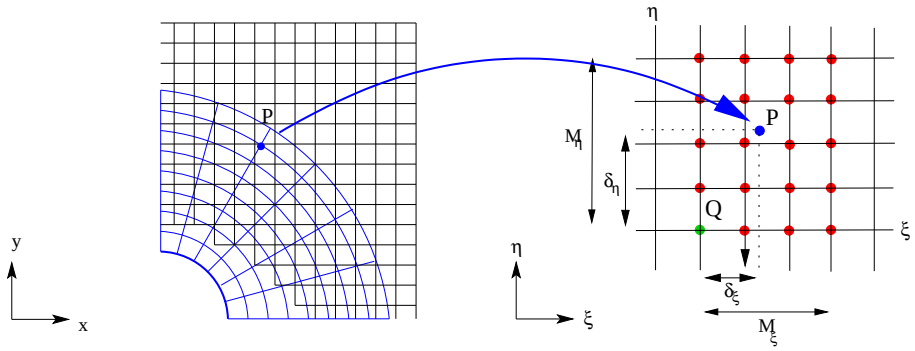


Figure 3: Example of a 2-D interpolation stencil: 2-D communication between a circular and a Cartesian component grids.

$$L_i^\xi = \prod_{m=0, m\neq i}^{M_\xi - 1} \frac{\delta_\xi - m}{i - m} \quad \text{and} \quad L_j^\eta = \prod_{m=0, m\neq j}^{M_\eta - 1} \frac{\delta_\eta - m}{j - m}$$

where δ_ξ and δ_η called the offsets are the coordinates of P, the receiver point, with respect to Q in the computational domain. For simplicity and isotropic reason, in the following, we have chosen $M_\xi = M_\eta = M$ which is also the Lagrangian polynomial order in the computational domain.

In addition, *Code_Safari* is parallelized by domain decomposition on each component grid for application to massively-parallel platforms. The communication between each domain is performed via the MPI library.

For moving grid applications, as the relative position of the overlapping grids changes continuously during the flow simulation, the interpolation data used for the communications between the component grids must be updated at each stage of the RK4 scheme. In practice, this updating is performed via the overlapping grid generator *Ogen* of the library *Overture*, see Henshaw (1998).

4.2 Linear analysis

The interpolation errors are assessed via a linear analysis to ensure that the interpolation procedure preserves the high accuracy of the present algorithm.

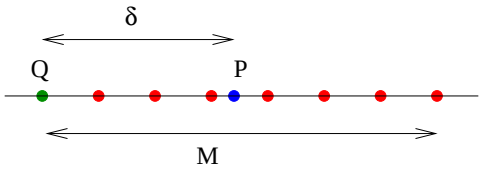


Figure 4: Example of a 1-D interpolation stencil.

In 1-D, the Lagrangian interpolation procedure in Equation (15) can be rewritten as follows:

$$\phi(x_P) \approx \sum_{i=0}^{M-1} L_i \phi(x_Q + i\Delta x) \quad \text{with} \quad L_i = \prod_{m=0, m\neq i}^{M-1} \frac{\delta - m}{i - m} \tag{16}$$

with $x_P = x_Q + \delta\Delta x$. The interpolation error is now quantified using a one-dimensional Fourier error analysis following Sherer and Scott (2005). Thus, we consider a single harmonic: $\phi(x) = e^{Ikx}$ as previously in Section 3 with the wavenumber k and $I^2 = -1$. The interpolation error factor can be defined as:

$$\mathcal{H}_{\text{itp}} = \frac{e^{I\delta k\Delta x}}{\sum_{i=0}^{M-1} L_i e^{Iik\Delta x}}$$

For a centered Lagrangian interpolation, we have $\delta \approx (M-1)/2$. The local error is displayed in Figure 5. The

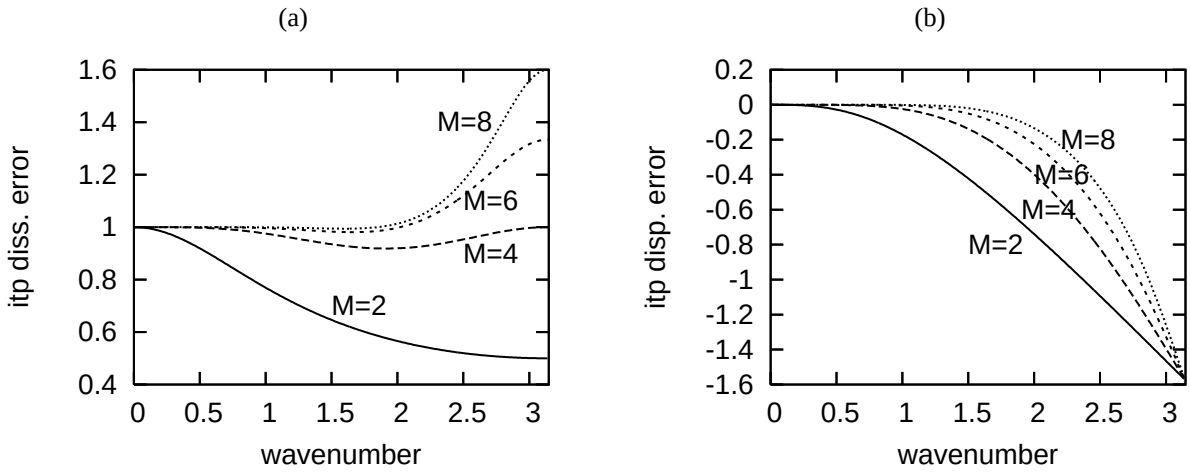


Figure 5: Local error of the interpolation process as a function of the wavenumber $k\Delta x$ with $M = 2, 4, 6$ and 8: (a) dissipation error or norm of $\mathcal{H}_{\text{itp}}$ and (b) dispersion error or phase of $\mathcal{H}_{\text{itp}}$.

Lagrangian interpolation procedure with $M = 2$ or $M = 4$ implies numerical errors in the wavenumber range not damped by the present algorithm according to the results in Section 3. This can lead to the generation of spurious waves. In contrast, Lagrangian interpolation with $M = 6$ or $M = 8$ seems to be suitable with the present numerical algorithm. To compare quantitatively the different polynomial interpolation, the limit accuracy limit in Equation (14) is still used: $|1 - \mathcal{H}_{\text{itp}}| \leq 5 \times 10^{-4}$. The accuracy domains are given in the table 1. The range of

M	$k_a\Delta x$	$\lambda_a/\Delta x$
2	0.04	169.81
4	0.34	18.48
6	0.65	9.59
8	0.90	6.94

Table 1: Accuracy limit of the Lagrangian polynomial interpolations with $M = 2, 4, 6$ and 8.

wavenumber well resolved by the present algorithm is thus incorporated in the one of the Lagrangian polynomial interpolation with $M = 6$ and $M = 8$.

5 Canonical tests on moving grids

The present high-order algorithm has shown to be suitable for LES of compressible flows with acoustic coupling on fixed grids in both subsonic and supersonic regimes, see Emmert et al. (2007, 2008). The validation procedure of the application of our high-order algorithm on dynamic meshes is performed in two stages. The first one concerns single-block computations in order to validate the calculation of the time metrics and the grid coordinate updating. Then, multi-block computations is used to couple the updating of the interpolation data with the numerical algorithm.

5.1 Inviscid vortex advection

The first validation test case is the vortex advection on a dynamically deforming 2-D mesh. The computational domain is taken as $[-2,2]\times[-1,1]$. Initially, an uniform mesh is retained with $\Delta x_0 = \Delta y_0 = 1/100$. The grid speed is analytically specified by the following equations:

$$\left\{\begin{array}{rcl} (x_\tau)_{i,j} & = & 2\pi\omega A_x \Delta x_0 \cos(2\pi\omega t)\sin\left(n_x\pi\dfrac{y_{i,j}(0)-y_{\min}}{y_{\max}-y_{\min}}\right)\alpha_x \\ (y_\tau)_{i,j} & = & 2\pi\omega A_y \Delta y_0 \cos(2\pi\omega t)\sin\left(n_y\pi\dfrac{x_{i,j}(0)-x_{\min}}{x_{\max}-x_{\min}}\right)\alpha_y \end{array}\right. \tag{17}$$

with

$$\alpha_x = \exp\left(-4\log(2)\frac{x_{i,j}(0)^2+y_{i,j}(0)^2}{(x_{\max}-x_{\min})^2}\right)$$

$$\alpha_y = \exp\left(-4\log(2)\frac{x_{i,j}(0)^2+y_{i,j}(0)^2}{(y_{\max}-y_{\min})^2}\right)$$

The grid coordinates are then provided via the RK4 scheme with the assumption that the grid speed is constant during a time step: $(X_\tau)^{(l-1)} = X_\tau^n \ \forall l \in \{1,..,4\}$ in Equation (10).

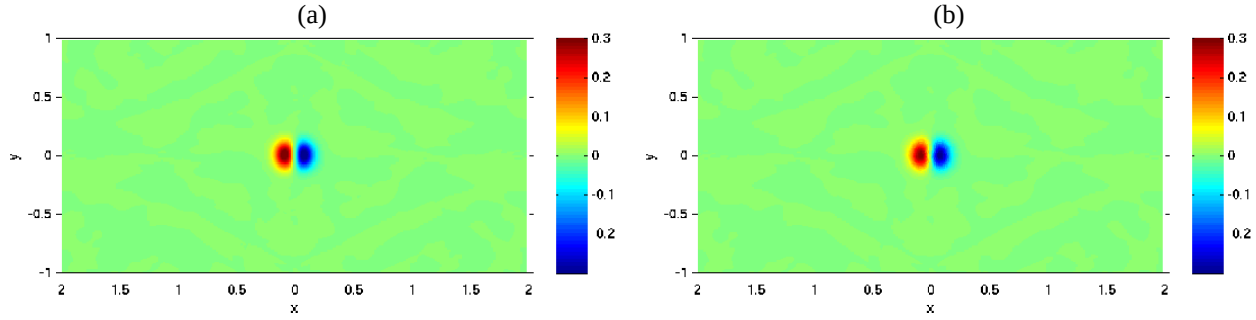


Figure 6: Comparison of the swirl velocity field: (a) in the static case ; (b) in the deforming case.

In fact, only the domain $[x_{\min}, x_{\max}]\times[y_{\min}, y_{\max}]$ is dynamically deformed. The different parameters are: $A_x = A_y = 2$, $n_x = n_y = 6$, $x_{\min} = y_{\min} = -0.5$, $x_{\max} = y_{\max} = 0.5$ and $\omega = 2$.

Two computations are performed: one on a static grid, the initial uniform grid, and the other with the grid velocity expressed in Equation (17). These two computations are performed with the same time step leading to CFL $= 0.5$, designed with the initial non deformed grid, in order to underline the effect of the mesh dynamic deformation on the high-order discretizations performance. The vortex is initially placed on $(x_c, y_c) = (0,0)$ and results given in this section are visualized when the vortex returns at its initial position. Comparison is given in Figures 6 and 7. The velocity fields in the static and deforming cases are similar which makes it possible to preserve the high-order schemes propagation properties on dynamically deforming meshes. The profile of the swirl velocity on $y = 0$ in Figure 7 shows the excellent agreement between the two computations. In addition, the dynamic deformation of the mesh implies a kind of numerical dissipation in the spatial high-order discretization characterized by a damping in the profile amplitude, as reported by Visbal and Gaitonde (2002).

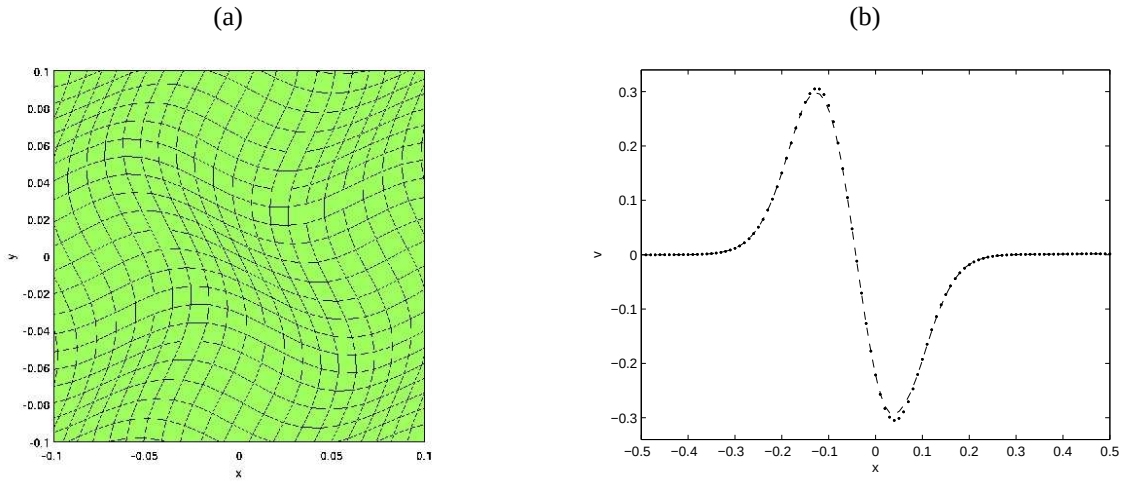


Figure 7: (a) Snapshot the grid deformation; (b) effect of the mesh dynamic deformation on the swirl velocity: . static case ; - - deforming case.

5.2 Cylinder advection in an inviscid uniform flow

The advection of a cylinder in a uniform flow at rest is now considered. The computational domain taken as $[-2, 2] \times [-2, 2]$ is divided in two grids. A cylindrical body-fitted grid moves with respect to a fixed Cartesian uniform grid with $\Delta x = \Delta y = 1/50$. The overlapping meshes are plotted in Figure 8. Initially, the center of the cylinder is located at $x_c = 0.85$.

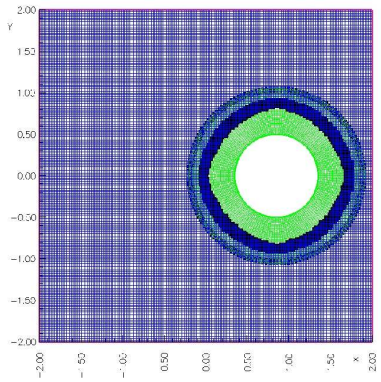


Figure 8: General view of the computational domain for the cylinder advection.

At every time step, a constant displacement of the cylindrical domain in the x-direction is imposed: $d_x = -0.08\Delta x$. Then, the mesh velocity is computed using the relation $x^{(l+1)} - x^{(l)} = d_x/4$ for every stage of the RK4 scheme. The computation is performed with CFL $= 0.5$. Thus, the cylinder is shifted at the Mach number $M_e \approx 0.3$. The radiation boundary conditions are applied to all the boundaries of the fixed Cartesian domain. Then, at the wall, a no-slip condition is used following the displacement of the cylinder.

The velocity field of the inviscid flow over an moving cylinder is plotted in Figure 9 for three different positions. First, a transient acoustic wave is generated by the initial motion of the cylinder. Then, the wave leaves the computational domain and a symmetric stationary solution with respect to the cylinder is reached.

6 Conclusion and future works

A numerical method has been described for performing compressible LES in CAA and aeroelasticity applications. The scheme is based on a 11-point explicit optimized finite-difference algorithm in conjunction with a 11-point optimized spatial low-pass filter. In order to address complex geometrical configurations, overlapping grids are used and the communications between domains are performed via high-order Lagrangian interpolation. The high-

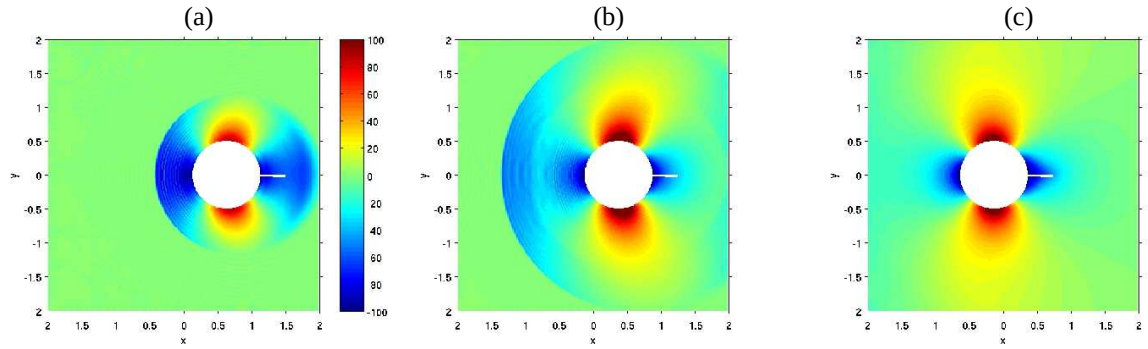


Figure 9: Time evolution of the streamwise velocity around the cylinder.

order overset-grid technique has proved to maintain the algorithm accuracy for moving grid applications.

To address fluid/structure interaction, the coupling algorithm between flow and structure has been implemented and a detailed validation procedure is in progress. The next configuration to be addressed is a moving cylinder in a 2-D low-Reynolds number flow. Forced oscillations will be simulated. Then the Vortex-Induced Vibration and the resulting radiated acoustic field will be computed for several mean flows. Finally the cylinder behaviour at lock-in will be investigated.

The choice of the time integration method is also to be considered. In the explicit method used in this work, the time step is imposed by stability constraints. However, the time step needed to respect the physical time scales of the turbulent flow may be larger. This is the case for turbulent wall-bounded flows, for example. The use of implicit time integration method would make it possible to circumvent the numerical stability by using a time step only driven by the flow physics, see Rizzetta et al. (2003), Daude et al. (2006).

Acknowledgments

This work is supported by the “Agence Nationale de la Recherche” under the reference ANR-06-CIS6-011. The authors want to thank Dr. Bill Henshaw for his useful advice concerning the overset strategy.

References

Bailly, C.; Bogey, C.; Marsden, O.: Advances in computational aeroacoustics: challenges and issues. *7th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements (ETMM7), Limassol, Cyprus, 4-6 June,*, pages 1–10.

Berland, J.; Bogey, C.; Marsden, O.; Bailly, C.: High-order, low dispersive and low dissipative explicit schemes for multiple-scale and boundary problems. *J. Comp. Phys.*, 224, 2, (2007), 637–662.

Bodony, D. J.; Lele, S. K.: On using large-eddy simulation for the prediction of noise from cold and heated turbulent jets. *Phys. of Fluids*, 17.

Bogey, C.; Bailly, C.: Three-dimensional non-reflective boundary conditions for acoustic simulations: far field formulation and validation test cases. *Acta Acustica*, 88, 4, (2002), 463–471.

Bogey, C.; Bailly, C.: A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *J. Comp. Phys.*, 194, 1, (2004), 194–214.

Bogey, C.; Bailly, C.: Large eddy simulations of transitional round jets: influence of the Reynolds number on flow development and energy dissipation. *Phys. of Fluids*, 213, 2, (2006), 777–802.

Bogey, C.; Bailly, C.: An analysis of the correlations between the turbulent flow and the sound pressure fields of subsonic jets. *J. Fluid Mech.*, 583, (2007), 71–97.

Colonius, T.; Lele, S. K.: Computational aeroacoustics: progress on nonlinear problems on sound generation. *Prog. Aerospace Sci.*, 40, (2004), 345–416.

Colonius, T.; Lele, S. K.; Moin, P.: Sound generation in a mixing layer. *J. Fluid Mech.*, 330, (1997), 375–409.

Crouzet, F.: A time-domain method for the vibration of mistuned bladed disk assemblies. *ASME Pressure Vessels and Piping Division Conference, Vancouver, BC, Canada, 2006, July 23-27, PVP2006-ICPVT11-93879*.

Daude, F.; Mary, I.; Comte, P.: Implicit time integration method for LES of complex flows. In: *Direct and Large-Eddy Simulation VI*, pages 771–778, Springer (2006).

Delfs, J. W.: An overlapped grid technique for high resolution CAA schemes for complex geometries. *AIAA Paper 2001-2199*.

Desquesnes, G.; Terracol, M.; Manoha, E.; Sagaut, P.: On the use of a high order overlapping grid method for coupling in CFD/CAA. *J. Comp. Phys.*, 220, 1, (2006), 355–382.

Emmert, T.; Lafon, P.; Bailly, C.: Computation of Aeroacoustic Phenomena in Subsonic and Transonic Ducted Flows. *AIAA Paper 2007-3429*.

Emmert, T.; Lafon, P.; Bailly, C.: Numerical study of aeroacoustic coupling in a subsonic confined cavity. *14th AIAA/CEAS Aeroacoustics Conference, Vancouver, Canada*.

Freund, J. B.: Noise sources in a low-Reynolds-number turbulent jet at Mach 0.9. *J. Fluid Mech.*, 438, (2001), 277–305.

Gloerfelt, X.; Bailly, C.; Juvé, D.: Direct computation of the noise radiated by a subsonic cavity flow and application of integral methods. *J. Sound Vib.*, 266, 1, (2003), 119–146.

Henshaw, W. D.: Ogen : An Overlapping Grid Generator for Overture. Tech. Rep. UCRL-MA-132237, Lawrence Livermore National Laboratory (1998).

Hirsch, C.: *Numerical computation of internal and external flows*. J. Wiley and Sons, New-York (1988).

Longatte, E.; Bendjeddou, Z.; Souli, M.: Methods for numerical study of tube bundle vibrations in cross-flows. *J. Fluids Struct.*, 18, 5, (2003), 513–528.

Marsden, O.; Bogey, C.; Bailly, C.: High-order curvilinear simulations of flows around non-Cartesian bodies. *J. Comput. Acoust.*, 13, 4, (2005), 731–748.

Rizzetta, D. P.; Visbal, M. R.; Blaisdell, G. A.: A time-implicit high-order compact differencing and filtering scheme for large-eddy simulation. *Int. J. Numer. Methods Fluids*, 42, (2003), 665–693.

Sherer, S. E.; Scott, J. N.: High-order compact finite-difference methods on general overset grids. *J. Comp. Phys.*, 210, 2, (2005), 459–496.

Suh, J.; Frankel, S. H.; Mongeau, L.; Plesniak, M. W.: Compressible large eddy simulation of wall-bounded turbulent flows using a semi-implicit numerical scheme for low Mach number aeroacoustics. *J. Comp. Phys.*, 215, 2, (2006), 526–551.

Tam, C. K. W.; Webb, J. C.: Dispersion-Relation-Preserving Finite Differences Schemes for Computational Acoustics. *J. Comp. Phys.*, 107, 2, (1993), 262–281.

Thomas, P. D.; Lombard, C. K.: Geometric conservation law and its application to flow computations on moving grids. *AIAA J.*, 17, 10, (1979), 1030–.

Thompson, K. W.: Time-dependent boundary conditions for hyperbolic systems, II. *J. Comp. Phys.*, 89, 2, (1990), 439–461.

Vinokur, M.: Conservation equations of gasdynamics in curvilinear coordinate systems. *J. Comp. Phys.*, 14, 105, (1974), 48–56.

Visbal, M. R.; Gaitonde, D. V.: On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes. *J. Comp. Phys.*, 181, 1, (2002), 155–185.

Viviand, H.: Formes conservatives des équations de la dynamique des gaz. *La Recherche Aérospatiale*, 158, (1974), 65–66.

Vreman, A. W.; Geurts, B. J.; Kuerten, J. G.: A priori tests of large-eddy simulation of compressible plane mixing layer. *J. Eng. Math.*, 118, 1, (1995), 24–37.

Address: F. Crouzet (corresponding author), EDF R&D, 1 av. du Général de Gaulle, 92141 Clamart Cedex.
email: `fabien.crouzet@edf.fr`.

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 55– 64
Herrsching am Ammersee, 29.9.-1.10.2008

An interface quasi-Newton algorithm for partitioned simulation of fluid-structure interaction

J. Degroote, R. Haelterman, S. Annerel, A. Swillens, P. Segers, J. Vierendeels

To solve a fluid-structure interaction problem efficiently in a partitioned way with a "black-box" flow solver and structure solver, an interface quasi-Newton technique with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) has been developed. In this work, the implementation of the IQN-ILS algorithm without explicit creation of the approximation for the inverse of the Jacobian is explained and it is demonstrated that the IQN-ILS technique can be implemented easily in simulation environments which currently use relaxation methods. Simulations of a flexible beam behind a cylinder and the propagation of a pressure wave in a carotid artery indicate that the IQN-ILS algorithm outperforms Aitken relaxation and that it has similar performance as the interface block quasi-Newton algorithm with approximate Jacobians from least-squares reduced-order models.

1 Introduction

The simulation of fluid-structure interaction (FSI) and other coupled problems has gained importance over the last decade. Nowadays, FSI simulations are used to avoid flutter of airplane wings (Farhat et al., 2006) and of blades in turbomachinery (Willcox et al., 1999). The design and analysis of artificial heart valves (dos Santos et al., 2006; Dumont et al., 2007), the prediction of the rupture of aneurysms or of the outcome of surgery (Taylor et al., 1999; Wilson et al., 2005) also rely on FSI simulations.

An FSI problem can be solved in either a monolithic or a partitioned way. The monolithic approach is to solve the flow equations and structural equations simultaneously such that their mutual influence can be taken into account during the solution process. In a partitioned FSI simulation, the flow equations and the structural equations are solved with separate codes which means that the flow does not change while the solution of the structural equations is calculated and vice versa. The partitioned approach thus requires a coupling algorithm to take the interaction between fluid and solid into account. However, the partitioned approach preserves software modularity and different, possibly more efficient solution techniques can be used for the flow equations and structural equations. This paper focuses on partitioned fluid-structure interaction simulations with "black-box" solvers.

In partitioned FSI simulations, coupling iterations between the flow solver and the structural solver have to be performed. In every coupling iteration, information on the fluid-structure interface is exchanged between the solvers. In case of a Dirichlet-Neumann decomposition of the FSI problem, the stress distribution on the fluid-structure interface is passed from the flow solver to the structural solver and the position of the interface is transferred the other way around. If the interaction between the fluid and the structure is weak, e.g. in aeroelastic simulations, so-called staggered or loosely-coupled methods (Farhat et al., 2006; Lesoinne and Farhat, 1998; Piperno et al., 1995) which require only one coupling iteration per time step can be used but these algorithms do not enforce the equilibrium conditions on the fluid-structure interface and are unstable in cases with strong interaction.

Gauss-Seidel coupling iterations are also unstable in simulations with an incompressible fluid that has a high added-mass effect (Causin et al., 2005; Förster et al., 2007). This instability has been explained by Causin et al. (2005) who rewrite the flow equations as an added-mass operator in the structural equations and by Degroote et al. (2008) who substitute a linearized model for the structure in the flow equations. Coupling algorithms that can be used in these simulations with strong interaction are e.g. Gauss-Seidel iterations with Aitken relaxation (Küttler and Wall, 2008) and the method with two reduced-order models from Vierendeels et al. (2007) which further will be denoted as interface block quasi-Newton of the Gauss-Seidel type with approximate Jacobians from least-squares reduced-order models (IBQN-LS) .

In Degroote et al. (2008), it is demonstrated that IBQN-LS outperforms Aitken relaxation for incompressible flow in a flexible tube. The difference in performance becomes larger when information from previous time steps is reused to improve the approximate Jacobians in the IBQN-LS method. However, the Aitken relaxation is significantly easier to implement as it only consists of a scalar relaxation factor which is modified in every coupling iteration. The IBQN-LS method requires action every time the flow problem or structural problem is solved whereas the Aitken relaxation only intervenes after the solution of both problems. Moreover, two linear systems with as dimension respectively the number of degrees-of-freedom in the interface's position and stress distribution have to be solved in every coupling iteration of the IBQN-LS method. These linear systems can be solved quickly and efficiently with a matrix-free iterative solver such that even large simulations with the IBQN-LS method are faster than with Aitken relaxation but the matrix-free implementation of IBQN-LS is more complex.

To combine a straightforward implementation with fast convergence, an interface quasi-Newton technique with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) has been developed. This technique uses quasi-Newton iterations to solve the fluid-structure problem reformulated as a set of equations with the discretized position of the fluid-structure interface as unknowns. The approximation for the *inverse* Jacobian is obtained from a least-squares reduced-order model with the technique from Vierendeels et al. (2007). Because the inverse of the Jacobian is approximated, no linear systems with as dimension the number of degrees-of-freedom in the interface's position have to be solved which is important for simulations with a high number of degrees-of-freedom in the interface's position. Moreover, the IQN-ILS method only intervenes in the coupling iterations after the solution of both the flow and structural problem like relaxation techniques and hence it can be implemented easily in an existing FSI framework that currently uses relaxation to accelerate the convergence of the coupling iterations.

The remainder of this work is organized as follows. Section 2 gives an abstract definition of the flow solver and structural solver and Section 3 explains the implementation of the IQN-ILS coupling technique with reuse of information from previous time steps. Section 4 demonstrates two examples for which the performance of IQN-ILS and IBQN-LS is similar and higher than the performance of Aitken relaxation with respect to both number of coupling iterations and duration of the simulation, followed by the conclusion in Section 5.

2 Definition of the flow solver and structural solver

This section gives a brief definition of the functions $\mathcal{F}$ and $\mathcal{S}$ that represent respectively the flow solver and structural solver. Because the solvers are black-box codes, the Jacobians of these functions are not available.

The expression

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}) \tag{1}$$

concisely represents several operations. First, the discretized position $\boldsymbol{x} \in \mathbb{R}^u$ of the fluid-structure interface is passed to the flow code and the grid of the fluid domain adjacent to the interface is adapted accordingly. Subsequently, the flow equations are solved for the fluid state in the entire domain which results in a distribution of the stress on the interface $\boldsymbol{y} \in \mathbb{R}^w$.

The actions of the structural solver are represented by

$$\boldsymbol{x} = \mathcal{S}(\boldsymbol{y}), \tag{2}$$

indicating that the stress on the interface is passed to the structural code which then solves the entire structural problem and returns the new position of the fluid-structure interface.

With these definitions, the FSI problem is given by

$$\mathcal{R}(\boldsymbol{x}) \equiv \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}) - \boldsymbol{x} = 0 \tag{3}$$

with $\mathcal{R}$ the residual operator of the coupled problem and with the symbol $\circ$ denoting that $\mathcal{F}(\boldsymbol{x})$ is calculated first and that the result is the argument of $\mathcal{S}$.

In the remainder of this paper, all values and functions are at the new time level $n+1$, unless indicated otherwise with a left superscript. A right superscript indicates the coupling iteration while a subscript denotes the element in a vector. Capital letters denote matrices, bold lower case letters indicate vectors and lower case letters represent scalars. Approximations are indicated with a hat. As the output of the solvers $\mathcal{F}$ and $\mathcal{S}$ is only an intermediate

value that is not passed on to the next coupling iteration, this value is indicated with a tilde. Once the final value that will be used in the next iteration has been calculated, the tilde is dropped. Depending on the context, the equality sign can denote either assignment or equality.

3 Interface quasi-Newton with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS)

The FSI problem reformulated as a set of nonlinear equations in the interface's position

$$\mathcal{R}(x) = 0 \tag{4}$$

can be solved with several techniques. If the Jacobian matrix $\mathrm{d}\mathcal{R}/\mathrm{d}\boldsymbol{x}$ would be available, equation (4) could be solved with Newton iterations

$$\text{solve} \quad \left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}^k} \Delta\boldsymbol{x}^k = -\boldsymbol{r}^k \tag{5a}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k. \tag{5b}$$

However, the exact Jacobian of $\mathcal{R}$ is unknown as the Jacobians of $\mathcal{F}$ and $\mathcal{S}$ are unavailable. Moreover, a linear system (5a) with as dimension the number of degrees-of-freedom in the position of the fluid-structure interface has to be solved in every Newton iteration. Although the number of degrees-of-freedom in the interface's position is generally smaller than the number of degrees-of-freedom in the entire fluid and structure domain, the Jacobian matrix $\mathrm{d}\mathcal{R}/\mathrm{d}\boldsymbol{x}$ is usually dense such that the solution of the linear system (5a) corresponds to a significant computational cost in large simulations.

If the Jacobian $\mathrm{d}\mathcal{R}/\mathrm{d}\boldsymbol{x}$ is approximated and quasi-Newtons iterations are performed, black-box solvers can be used but this approach does not avoid that the linear system (5a) has to be solved. It is more advantageous to approximate the *inverse* of the Jacobian instead by applying the least-squares technique introduced by Vierendeels et al. (2007) on a particular set of vectors as will be explained later. The quasi-Newton iterations with the approximation for the inverse of the Jacobian can be written as

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \widehat{\left(\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}^k}\right)^{-1}} \left(-\boldsymbol{r}^k\right) \tag{6}$$

and these iterations solve $\mathcal{R}(\boldsymbol{x}) = 0$ for the interface's position, hence the name interface quasi-Newton with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS). The initial guess for the quasi-Newton iterations is obtained from an extrapolation of the interface's position

$$\boldsymbol{x}^0 = \frac{5}{2}({}^n\boldsymbol{x}) - 2({}^{n-1}\boldsymbol{x}) + \frac{1}{2}({}^{n-2}\boldsymbol{x}), \tag{7}$$

based on the previous time steps. Lower order extrapolations are used in the first two time steps. The residual in Eq. (6) is calculated as

$$\boldsymbol{r}^k = \mathcal{R}(\boldsymbol{x}^k) = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k) - \boldsymbol{x}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k \tag{8}$$

and the quasi-Newton iterations in the time step have converged when $||\boldsymbol{r}^k||_2 \leq \epsilon$.

It can be seen from equation (6) that the approximation for the inverse of the Jacobian does not have to be created explicitly, a procedure to calculate the product of this matrix with the vector $-\boldsymbol{r}^k$ is sufficient. The vector $-\boldsymbol{r}^k$ is the difference between the desired residual, i.e. $\mathbf{0}$, and the current residual $\boldsymbol{r}^k$ and it is further denoted as $\Delta\boldsymbol{r} = \mathbf{0} - \boldsymbol{r}^k = -\boldsymbol{r}^k$. In this work, the matrix-vector product is calculated from information obtained during the previous quasi-Newton iterations. Equation (8) shows that the flow equations and structural equations are solved in quasi-Newton iteration k, resulting in $\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k)$ and the corresponding residual $\boldsymbol{r}^k$. The vectors $\tilde{\boldsymbol{x}}$ and $\boldsymbol{r}$ from all previous coupling iterations are also available, giving a set of known residual vectors

$$\boldsymbol{r}^k, \quad \boldsymbol{r}^{k-1}, \quad \ldots, \quad \boldsymbol{r}^1, \quad \boldsymbol{r}^0 \tag{9a}$$

and the corresponding set of vectors $\tilde{x}$

$$\tilde{\boldsymbol{x}}^{k+1}, \quad \tilde{\boldsymbol{x}}^k, \quad \ldots, \quad \tilde{\boldsymbol{x}}^2, \quad \tilde{\boldsymbol{x}}^1. \tag{9b}$$

The differences between all vectors from previous iterations (superscript i) and the most recent vector (superscript k) are calculated

$$\Delta \boldsymbol{r}^i = \boldsymbol{r}^i - \boldsymbol{r}^k \tag{10a}$$

$$\Delta \tilde{\boldsymbol{x}}^{i+1} = \tilde{\boldsymbol{x}}^{i+1} - \tilde{\boldsymbol{x}}^{k+1} \tag{10b}$$

for $i = 0, \ldots, k-1$. Each $\Delta \boldsymbol{r}^i$ corresponds to a $\Delta \tilde{\boldsymbol{x}}^{i+1}$ and these vectors are stored as the columns of the matrices

$${}^{n+1}V^k = \begin{bmatrix} \Delta \boldsymbol{r}^{k-1} & \Delta \boldsymbol{r}^{k-2} & \ldots & \Delta \boldsymbol{r}^1 & \Delta \boldsymbol{r}^0 \end{bmatrix} \tag{11a}$$

and

$${}^{n+1}W^k = \begin{bmatrix} \Delta \tilde{\boldsymbol{x}}^k & \Delta \tilde{\boldsymbol{x}}^{k-1} & \ldots & \Delta \tilde{\boldsymbol{x}}^2 & \Delta \tilde{\boldsymbol{x}}^1 \end{bmatrix}. \tag{11b}$$

The matrices ${}^{n+1}V^k$ and ${}^{n+1}W^k$ can be combined with those from s previous time steps (if at least s time steps have already been performed), giving

$$V^k = \begin{bmatrix} {}^{n+1}V^k & {}^{n}V & \ldots & {}^{n-s+2}V & {}^{n-s+1}V \end{bmatrix} \tag{12a}$$

and

$$W^k = \begin{bmatrix} {}^{n+1}W^k & {}^{n}W & \ldots & {}^{n-s+2}W & {}^{n-s+1}W \end{bmatrix}. \tag{12b}$$

The columns of the matrices ${}^{n-i+1}V$ and ${}^{n-i+1}W$ are calculated by subtracting the vector of the last iteration of time step $n-i+1$ from all previous vectors in that time step, in the same way as in equation (10). By including the information for s previous time steps, the convergence of the coupling iterations is accelerated remarkably. However, if information from too many time steps is reused, the convergence can slow down again as information from time step $n-s+1$ might no longer be relevant in time step $n+1$. The optimal value of s is problem dependent but the convergence of the coupling iterations does not change significantly near the optimum such that the performance of the method is robust with respect to the parameter s.

The number of columns in V^k and W^k is indicated with v and is generally much smaller than the number of rows u. Nevertheless, in simulations with a low number of degrees-of-freedom on the interface, it is possible that the number of columns has to be limited to u by discarding the rightmost columns.

The vector $\Delta \boldsymbol{r} = \boldsymbol{0} - \boldsymbol{r}^k$ is approximated as a linear combination of the known $\Delta \boldsymbol{r}^i$

$$\Delta \boldsymbol{r} \approx V^k \boldsymbol{c}^k \tag{13}$$

with $\boldsymbol{c}^k \in \mathbb{R}^v$ the coefficients of the decomposition. Because $v \leq u$, equation (13) is an overdetermined set of equations for the elements of $\boldsymbol{c}^k$ and hence the least-squares solution to this linear system is calculated. For that reason, the so-called economy size QR-decomposition of V^k is calculated using Householder transformations (Golub and Van Loan, 1996)

$$V^k = Q^k R^k \tag{14}$$

with $Q^k \in \mathbb{R}^{u \times v}$ an orthogonal matrix and $R^k \in \mathbb{R}^{v \times v}$ an upper triangular matrix. The coefficient vector $\boldsymbol{c}^k$ is then determined by solving the triangular system

$$R^k \boldsymbol{c}^k = {Q^k}^{\mathrm{T}} \Delta \boldsymbol{r} \tag{15}$$

using back substitution. If a $\Delta \boldsymbol{r}^i$ vector is (almost) a linear combination of other $\Delta \boldsymbol{r}^j$ vectors, one of the diagonal elements of R^k will (almost) be zero. Consequently, the equation corresponding to that row of R^k cannot be solved during the back substitution and the corresponding element of $\boldsymbol{c}^k$ is set to zero.

The $\Delta \tilde{\boldsymbol{x}}$ that corresponds to $\Delta \boldsymbol{r}$ is subsequently calculated as a linear combination of the previous $\Delta \tilde{\boldsymbol{x}}^i$, similar to equation (13), giving

$$\Delta \tilde{\boldsymbol{x}} = W^k \boldsymbol{c}^k. \tag{16}$$

From equation (8), it follows that

$$\Delta \boldsymbol{r} = \Delta \tilde{\boldsymbol{x}} - \Delta \boldsymbol{x} \tag{17}$$

and substitution of equation (16) in equation (17) results in

$$\Delta \boldsymbol{x} = W^k \boldsymbol{c}^k - \Delta \boldsymbol{r}. \tag{18}$$

Because the coefficients $\boldsymbol{c}^k$ are a function of $\Delta\boldsymbol{r}$, equation (18) shows how $\Delta\boldsymbol{x}$ can be approximated for a given $\Delta\boldsymbol{r}$. Hence, equation (18) can be seen as a procedure to calculate the product of the approximation for the inverse of the Jacobian and a vector $\Delta\boldsymbol{r} = -\boldsymbol{r}^k$

$$\Delta\boldsymbol{x} = \widehat{\left(\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_k}\right)^{-1}} \Delta\boldsymbol{r} = W^k\boldsymbol{c}^k + \boldsymbol{r}^k. \tag{19}$$

The relation between $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{x}$ is thus found by means of the $\Delta\tilde{\boldsymbol{x}}$ values. If a direct relation between $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{x}$ is constructed instead, the quasi-Newton iterations will not converge as the new input for $\mathcal{S}\circ\mathcal{F}$ would be a linear combination of the previous inputs.

The complete IQN-ILS technique is shown in Algorithm 1. For steady calculations, one time step ($n_{max} = 1$) has to be performed with the same algorithm. Because the matrices V^k and W^k have to contain at least one column, an relaxation with factor ω (line 9) is performed in the second coupling iteration of the first time step if information from the previous time steps is reused ($s > 0$) and in the second coupling iteration of every time step without reuse ($s = 0$).

Algorithm 1 Interface quasi-Newton with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS)

1: $n = 0$
2: **while** $n < n_{max}$ **do**
3: $\quad\boldsymbol{x}^0 = \frac{5}{2}({}^n\boldsymbol{x}) - 2({}^{n-1}\boldsymbol{x}) + \frac{1}{2}({}^{n-2}\boldsymbol{x})$ (if $n \geq 2$, lower order otherwise)
4: $\quad k = 0$
5: $\quad\tilde{\boldsymbol{x}}^1 = \mathcal{S}\circ\mathcal{F}(\boldsymbol{x}^0)$
6: $\quad\boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
7: $\quad$**while** $||\boldsymbol{r}^k||_2 > \epsilon$ **do**
8: $\quad\quad$**if** $k = 0$ and ($s = 0$ or $n = 0$) **then**
9: $\quad\quad\quad\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega\boldsymbol{r}^k$
10: $\quad\quad$**else**
11: $\quad\quad\quad$construct V^k and W^k as shown in equation (10) to (12)
12: $\quad\quad\quad$calculate QR-decomposition $V^k = Q^kR^k$
13: $\quad\quad\quad$solve $R^k\boldsymbol{c}^k = -{Q^k}^{\mathrm{T}}\boldsymbol{r}^k$ with back substitution
14: $\quad\quad\quad\boldsymbol{x}^{k+1} = W^k\boldsymbol{c}^k + \boldsymbol{r}^k$
15: $\quad\quad$**end if**
16: $\quad\quad k = k + 1$
17: $\quad\quad\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S}\circ\mathcal{F}(\boldsymbol{x}^k)$
18: $\quad\quad\boldsymbol{r}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k$
19: $\quad$**end while**
20: $\quad n = n + 1$
21: **end while**

Figure 1 depicts a simplified representation of the IBQN-LS, IQN-ILS and (Aitken) relaxation algorithm. The IBQN-LS algorithm creates a least-squares approximation for $\mathrm{d}\mathcal{F}/\mathrm{d}\boldsymbol{x}$ and $\mathrm{d}\mathcal{S}/\mathrm{d}\boldsymbol{y}$ and uses them to solve

$$\boldsymbol{y} - \mathcal{F}(\boldsymbol{x}) = 0 \tag{20a}$$

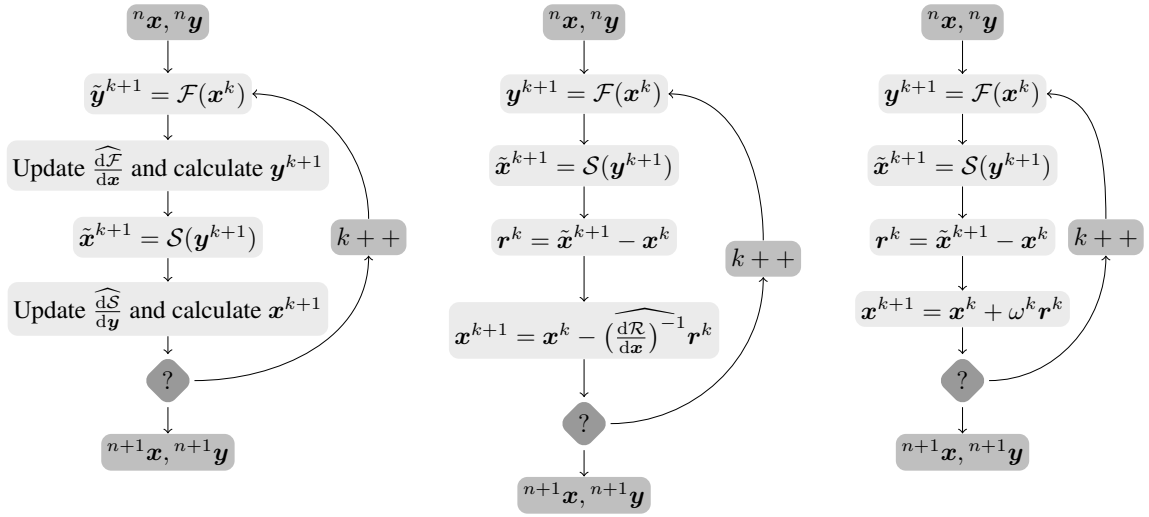
$$\boldsymbol{x} - \mathcal{S}(\boldsymbol{y}) = 0 \tag{20b}$$

with block quasi-Newton iterations of the Gauss-Seidel type. Relaxation methods solve $\boldsymbol{x} = \mathcal{S}\circ\mathcal{F}(\boldsymbol{x})$ with relaxed fixed-point iterations, possibly with adaptation of the relaxation factor ω^k in every coupling iteration.

Both Algorithm 1 and Figure 1 demonstrate that IQN-ILS can be implemented easily in an existing FSI framework with relaxation. A conditional statement has to be wrapped around the existing relaxation on line 9 such that the relaxation is only performed if not enough information is available to approximate the product of the inverse of the Jacobian and $-\boldsymbol{r}^k$, otherwise the procedure of the IQN-ILS algorithm on line 11 to 14 has to be executed. The IQN-ILS algorithm consists of vector operations, two matrix-vector products, a QR-decomposition and the solution of a triangular system which are all available in standard linear algebra packages. Moreover, as v is generally much smaller than u, the procedure described above is significantly faster and less memory consuming than explicit creation of the approximation for the inverse of the Jacobian as

$$\widehat{\left(\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_k}\right)^{-1}} = W^kR^{k^{-1}}{Q^k}^{\mathrm{T}} - I \tag{21}$$

with $I \in \mathbb{R}^{u \times u}$ the identity matrix. The results will demonstrate that only a small fraction of the total simulation time is spent on line 11 to 14.



(a) IBQN-LS (b) IQN-ILS (c) Relaxation

Figure 1: Simplified representation of the (a) IBQN-LS, (b) IQN-ILS and (c) (Aitken) relaxation algorithm for partitioned FSI with black-box solvers.

4 Results

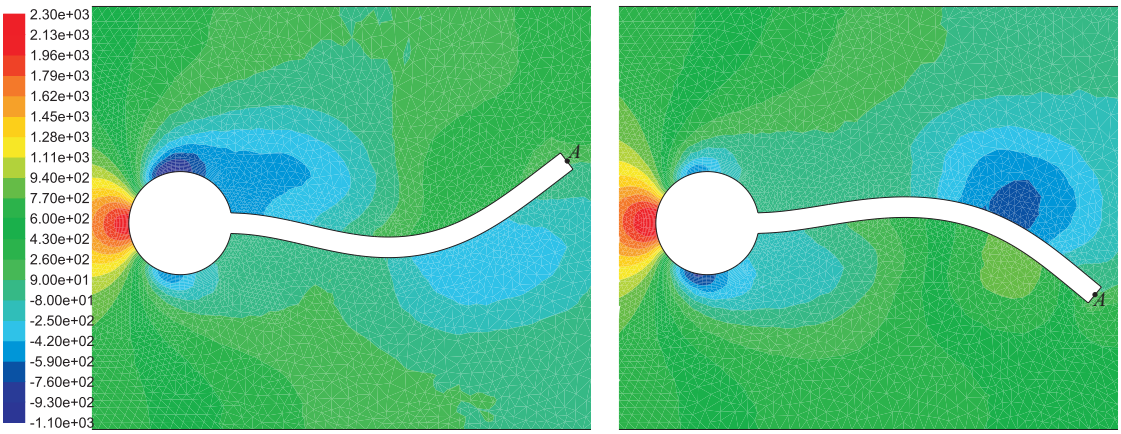
In this section, the IBQN-LS, IQN-ILS and Aitken relaxation are compared with respect to both the number of coupling iterations per time step and the total duration of the simulation. Two examples are given, namely the oscillation of a 2D flexible beam and the propagation of a pressure pulse in a 3D carotid artery. In both examples, the flow field is calculated by a finite volume flow solver with second order discretization of the pressure, second order upwind discretization of the momentum and first order time accuracy on moving meshes. The mesh of the fluid domain is adapted to the position of the fluid-structure interface with a spring analogy and the mesh is adapted if cells are too distorted. The structural solver is a finite element code with implicit time integration which takes geometric nonlinearity due to large deformations into account. The coupling algorithm, the flow solver and structural solver are all executed on a dedicated machine with two quad-core processors.

4.1 2D Flexible beam

The first example is the oscillation of a flexible beam, specifically the unsteady FSI2 test as described in detail in the benchmark paper by Turek and Hron (2006). The geometry consists of a horizontal channel of 0.41 m high which contains a rigid cylinder with center positioned 0.2 m above the bottom of the channel. A laminar, viscous flow with density 10^3 kg/m^3 and kinematic viscosity 10^{-3} m^2/s enters the channel from the left hand side with a parabolic velocity profile and a mean inlet velocity of 1.0 m/s. A constant pressure is imposed at the outlet of the channel and a no-slip condition is applied to the top and bottom of the channel. A linearly elastic beam with density 10^4 kg/m^3, Young's modulus $1.40 \cdot 10^6$ N/m^2 and Poisson's ratio 0.4 is attached to the right hand side of the cylinder.

The fluid domain is initially discretized with 24806 triangular and quadrilateral cells, a number that varies slightly due to mesh adaptation, and the solid domain consists of 1200 quadrilateral continuum elements. After validating the flow solver and the structural solver with the tests listed by Turek and Hron (2006), the FSI2 test has been executed. The pressure contours in the vicinity of the structure are shown in Figure 2.

The point at the right end of the beam (as indicated in Figure 2) is called A. The displacement u of this point in the x and y direction and the drag and lift on the entire structure are given in Table 1 and Figure 3 in the same format



(a) (b)

Figure 2: Pressure contours in the 2D unsteady FSI2 test with the flexible beam after (a) 12 s; (b) 16 s.

as in the benchmark paper. The residual $||r_i||_2$ is reduced five orders of magnitude with respect to its initial value and the same results have been found with IQN-ILS, IBQN-LS and Aitken relaxation. The deviations between the simulations and the benchmark are reasonable, given that the flow solver is only first order accurate in time.

	u_x [10^{-3} m]	u_y [10^{-3} m]	drag [N]	lift [N]
Simulation	-14.07±12.37[3.7]	1.18±76.5[1.9]	217.52±84.65[3.7]	-0.74±267.6[1.9]
Benchmark	-14.58±12.44[3.8]	1.23±80.6[2.0]	208.83±73.75[3.8]	0.88±234.2[2.0]

Table 1: Results of the 2D unsteady FSI2 test with the flexible beam. Displacement of point A in the x and y direction, and drag and lift on the entire structure. All data are given in the format mean±amplitude[frequency].

The number of coupling iterations per time step (averaged over the last period of the oscillation) and the relative duration of the simulations is given in Table 2. The notations IQN-ILS(s) and IBQN-LS(s) denote that information from the s previous time steps has been reused. When information from the last 3 time steps is reused, the average number of coupling iterations per time step reduces with approximately 30 % for both IQN-ILS and IBQN-LS compared to the simulations without reuse. The performance of IQN-ILS(3) and IBQN-LS(3) is comparable and both techniques complete the FSI2 test nearly twice as fast as Aitken relaxation.

In the FSI2 test with IQN-ILS(3), only 0.04 % of the total simulation time was spent on line 11 to 14 in Algorithm 1. The computational cost of the IQN-ILS algorithm is hence negligible compared to the computational cost of the flow problem and the structural problem. This can also be seen from Table 2 which shows that the difference between IQN-ILS and Aitken relaxation is almost the same with respect to average number of coupling iterations and duration of the simulation.

	Iterations	Duration
IQN-ILS	9.4	1.84
IQN-ILS(3)	6.1	1.07
IBQN-LS	7.2	1.54
IBQN-LS(3)	4.8	1.00
Aitken relaxation	9.9	1.81

Table 2: Number of coupling iterations per time step (averaged over the last period of the oscillation) and relative duration of the simulations for the 2D unsteady FSI2 test with the flexible beam. The notations IQN-ILS(s) and IBQN-LS(s) denote that information from the s previous time steps has been reused.

4.2 3D Carotid bifurcation

The second example is the propagation of a pressure wave in the geometry of a carotid artery. Fernandez and Moubachir (2005), among others, described the propagation of a pressure wave in a straight cylindrical tube and the same material properties and boundary conditions have been used in this work with a more complicated geometry.

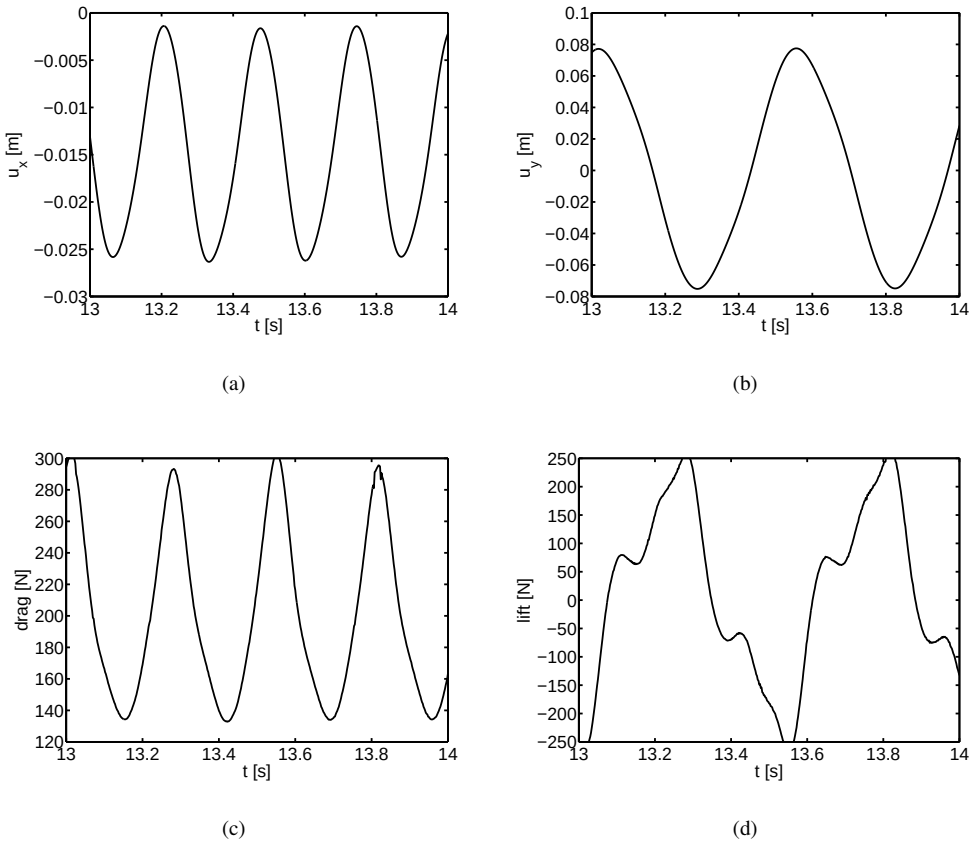


Figure 3: Results of the 2D unsteady FSI2 test with the flexible beam. Displacement of point A in (a) the x direction and (b) the y direction, (c) drag and (d) lift on the entire structure.

The model for the carotid artery is based on a geometry from a healthy volunteer in which an eccentric plaque was added. Starting from CT-scans, the geometry was three-dimensionally reconstructed. The fluid domain is initially discretized with 148054 tetrahedral cells and the structural model consists of one layer with 19062 triangular shell elements. The distance between the inlet at the common carotid artery and the two outlets (interna and externa) is 0.095 m. This simulation does not aim at reproducing the reality but nevertheless it is a tough test for the algorithms.

The solid is linear elastic with Young's modulus $3{\cdot}10^5\,\mathrm{N/m^2}$ and Poisson's ratio 0.3 and it has a density of $1200\,\mathrm{kg/m^3}$. The structure is clamped at both ends. The viscosity of the fluid is $0.003\,\mathrm{Pa{\cdot}s}$ and its density is $1000\,\mathrm{kg/m^3}$. At the inlet, a pressure of 1333.2 Pa is applied during 0.003 s and 0 Pa is applied afterwards. At the outlets a pressure of 0 Pa is imposed and the fluid-structure interface is a no-slip boundary. The wave propagation is followed during 0.01 s with time steps of 0.0001 s. The pressure contours on the fluid-structure interface are shown in Figure 4.

The average number of coupling iterations per time step and the relative duration of the simulations with IQN-ILS, IBQN-LS and Aitken relaxation are given in Table 3. The residual is reduced with 3 orders of magnitude with respect to its initial value in the time step. The simulation with Aitken relaxation has been stopped after 10 time steps because the convergence criterion has not been reached in any of the first 10 time steps within 50 coupling iterations. In this simulation with a much higher number of degrees-of-freedom, the IQN-ILS algorithm is 10 % faster than the IBQN-LS algorithm although the average number of coupling iterations is only 5 % lower because no linear systems have to be solved in the IQN-ILS method. The linear systems in the IBQN-LS method are solved with a matrix-free iterative solver which reduces the L2-norm of the residual of the linear systems by a factor 10^{-10}.

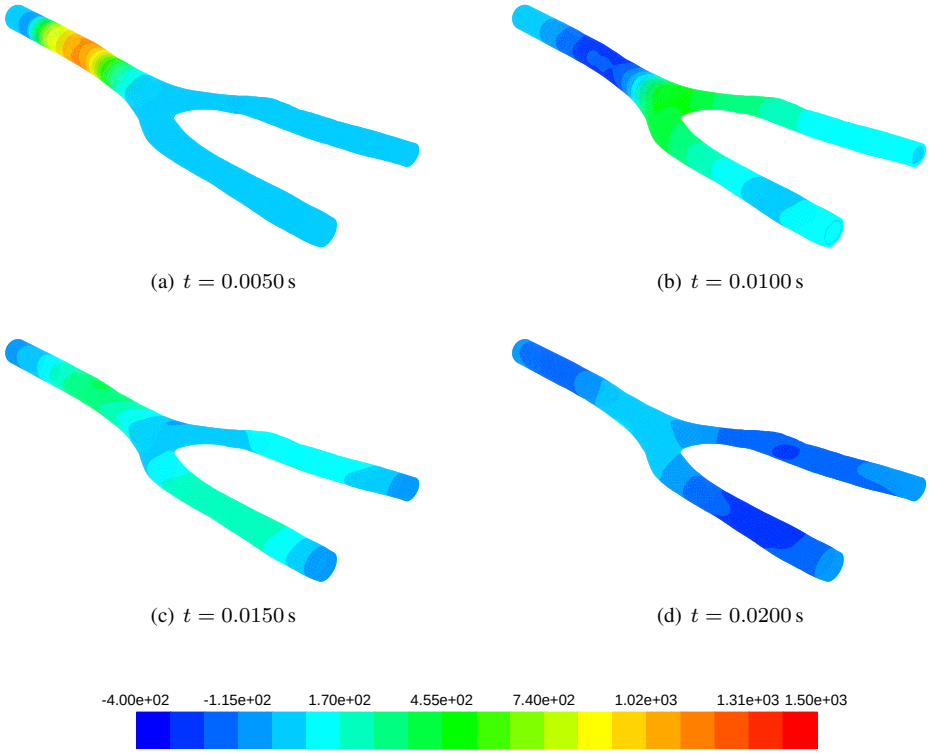


Figure 4: Pressure contours on the fluid-structure interface for the simulation of the wave propagation in the carotid bifurcation.

	Iterations	Duration
IQN-ILS(2)	11.9	1.00
IBQN-LS(2)	12.5	1.11
Aitken relaxation	—	—

Table 3: Number of coupling iterations per time step (averaged over the entire simulation) and relative duration of the simulations for the 3D propagation of the pressure wave in the carotid artery. The notations IQN-ILS(s) and IBQN-LS(s) denote that information from the s previous time steps has been reused. The simulation with Aitken relaxation has been stopped after 10 time steps because the convergence criterion had not been reached in any of those time steps within 50 coupling iterations.

5 Conclusion

The interface quasi-Newton technique with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS) has a similar performance as the interface block quasi-Newton technique (IBQN-LS), but it is significantly easier to implement. Moreover, it avoids that linear systems with as dimension the number of degrees-of-freedom in the position of the fluid-structure interface have to be solved which is an advantage for large simulations. In the examples presented here, IQN-ILS outperforms Aitken relaxation and it has been demonstrated how the IQN-ILS algorithm can be implemented in existing FSI codes that currently rely on relaxation.

Acknowledgments

J. Degroote gratefully acknowledges a Ph.D. fellowship of the Research Foundation - Flanders (FWO). A. Swillens is funded by a grant of the Special Fund for Scientific Research of Ghent University (BOF).

References

Causin, P.; Gerbeau, J.-F.; Nobile, F.: Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194, 42-44, (2005), 4506–4527.

Degroote, J.; Bruggeman, P.; Haelterman, R.; Vierendeels, J.: Stability of a coupling technique for partitioned solvers in FSI applications. *Computers & Structures*, 86, (2008), 2224–2234, dOI: 10.1016/j.compstruc.2008.05.005.

dos Santos, N.; Gerbeau, J.-F.; Bourgat, J.-F.: Partitioned FSI strategy for simulations of a thin elastic valve. In: P. Wesseling; E. Oñate; J. Périaux, eds., *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, pages 1–10, ECCOMAS, Delft, The Netherlands (2006).

Dumont, K.; Vierendeels, J.; Kaminsky, R.; Van Nooten, G.; Verdonck, P.; Bluestein, D.: Comparison of the hemodynamic and thrombogenic performance of two bileaflet mechanical heart valves using a CFD/FSI model. *Journal of Biomechanical Engineering - Transactions of the ASME*, 129, 4, (2007), 558–565.

Farhat, C.; van der Zee, K.; Geuzaine, P.: Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006), 1973–2001.

Fernandez, M.; Moubachir, M.: A Newton method using exact Jacobians for solving fluid-structure coupling. *Computers & Structures*, 83, (2005), 127–142.

Förster, C.; Wall, W.; Ramm, E.: Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196, 7, (2007), 1278–1293.

Golub, G. H.; Van Loan, C. F.: *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edn. (1996).

Küttler, U.; Wall, W.: Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43, 1, (2008), 61–72, DOI 10.1007/s00466-008-0255-5.

Lesoinne, M.; Farhat, C.: A higher-order subiteration free staggered algorithm for non-linear transient aeroelastic problems. *AIAA Journal*, 36, 9, (1998), 1754–1756.

Piperno, S.; Farhat, C.; Larrouturou, B.: Partitioned procedures for the transient solution of coupled aeroelastic problems - Part I: model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124, (1995), 79–112.

Taylor, C.; Draney, M.; Ku, J.; Parker, D.; Steele, B.; Wang, K.; Zarins, C.: Predictive medicine: Computational techniques in therapeutic decision-making. *Computer Aided Surgery*, 4, (1999), 231–247.

Turek, S.; Hron, J.: Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction – Modelling, Simulation, Optimisation*, no. 53 in Lecture Notes in Computational Science and Engineering, pages 371–385, Springer, Berlin (2006), ISBN 3-540-34595-7.

Vierendeels, J.; Lanoye, L.; Degroote, J.; Verdonck, P.: Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers & Structures*, 85, 11–14, (2007), 970–976.

Willcox, K.; Paduano, J.; Peraire, J.: Low order aerodynamic models for aeroelastic control of turbomachines. In: *40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials (SDM) Conference*, pages 1–11, St Louis, MO, USA (april 1999).

Wilson, N.; Arko, F.; Taylor, C.: Predicting changes in blood flow in patient-specific operative plans for treating aortoiliac occlusive disease. *Computer Aided Surgery*, 10, 4, (2005), 257–277.

Address: Prof. J. Vierendeels (corresponding author), Department of Flow, Heat and Combustion Mechanics, Ghent University, Sint-Pietersnieuwstraat 41, Ghent, Belgium.
email: `Jan.Vierendeels@UGent.be`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 65– 80
Herrsching am Ammersee, 29.9.-1.10.2008

Efficient Fluid-Structure Interaction Simulation of Viscoplastic and Fracturing Thin-Shells Subjected to Underwater Shock Loading

Ralf Deiterding, Fehmi Cirak, and Sean P. Mauch

The fluid-structure interaction simulation of shock-loaded thin-walled structures requires numerical methods that can cope with large deformations as well as local topology changes. We present a robust level-set-based approach that integrates a Lagrangian thin-shell finite element solver with fracture and fragmentation capabilities into an Eulerian Cartesian fluid solver with embedded boundary and mesh adaptation capability. As main computational applications, we consider the plastic deformation and rupture of thin plates subjected to explosion and piston-induced pressure waves in water.

1 Introduction

The construction of efficient and scalable algorithms for simulating shock-driven fluid-structure interaction (FSI) problems is an area of active research. The discretizations both in fluid and solid are usually time-explicit and therefore computationally comparably inexpensive. On the other hand, major geometric complexities, such as large structural deformations, fracture, and even fragmentation might have to be considered. An approach to this problem is to employ an immersed or embedded boundary method in the fluid solver (Mittal and Iaccarino, 2005), in which moving solid structures slide through a fixed (Eulerian) fluid background mesh.

We have developed a generic software framework for shock-driven FSI simulation, named *Virtual Test Facility* (Deiterding et al., 2006b), that imposes embedded moving wall boundary conditions on a Cartesian fluid solver with a ghost fluid approach, as proposed by Fedkiw (2002) and Arienti et al. (2003). A scalar level set function storing the distance to the nearest element of the solid's triangulation is utilized to represent the embedded geometry on the fluid grid and block-based dynamic mesh adaptation is employed to mitigate boundary approximation inaccuracies. The Virtual Test Facility (VTF) specifically targets coupled problems in the high-speed regime, such as the transient deformation of metallic structures due to explosive detonations or the fracture and fragmentation of brittle or ductile materials under shock wave impact, cf. Aivazis et al. (2000) and Mauch et al. (2003). Computational fluid and solid mechanics solvers are time-accurate and consider all arising supersonic wave phenomena (e.g., shear and dilatation waves in the viscoplastic solid, shock waves in the compressible fluid) correctly. For coupling, a temporal splitting technique, in which solvers exchange data only at the interface between disjoint computational domains after consecutive time steps, is adopted. For compressible fluids, stable solutions are obtained reliably with such a *weakly coupled* method, when the evolving interface geometry and velocities are imposed as boundary conditions on the fluid solver and the hydrodynamic pressure is used as force boundary condition acting on the solid exterior, cf. Löhner et al. (2003), Cirak and Radovitzky (2005), and Specht (2000).

While the VTF approach has been successfully applied to verification and validation examples driven by shock and detonation waves in gases, cf. Deiterding et al. (2006a) and Deiterding et al. (2006c), we focus in here primarily on thin-walled solid structures subjected to strong pressure waves in water. In Section 2, we sketch the adaptive Cartesian finite volume fluid solver with level-set-based embedded boundary capability and briefly describe the employed multiphase Riemann solver. Section 3 outlines the solid mechanics solver that has been developed to enable FSI simulations of thin-walled (possibly fracturing) shell structures. In Section 4, we outline the highly efficient auxiliary algorithm based on geometric characteristic reconstruction and scan conversion that we have developed to transform evolving triangulated surface meshes efficiently into signed or unsigned distance functions. The fluid-structure coupling algorithm, highlighting its incorporation into the adaptive fluid mesh refinement framework is detailed in Section 5. The simple verification test of an elastic beam under shock loading is

discussed in Section 6.1. Sections 6.2 and 6.3 present FSI simulation results for two setups involving viscoplastic deformation and fracture of thin circular isotropic metal plates subjected to shock loadings from or comparable to underwater explosions. Good agreement with experimental results and the level of detail provided make the computations excellent test cases for explicit FSI software.

2 Eulerian fluid mechanics solver

In this section, we are concerned with the construction of an Eulerian fluid solver suitable for efficient fluid-structure coupling. Although the presentation is tailored to the two-component solver employed throughout this paper, the concepts are equally applicable to other conservation laws with or without source terms, cf. Deiterding et al. (2006c).

2.1 Governing equations

The simulation of trans- or supersonic wave phenomena in fluids requires the consideration of the compressibility while viscosity can typically be neglected. The basic system of governing equations are the Euler equations:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \,, \quad \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = 0 \,, \quad \partial_t (\rho E) + \nabla \cdot ((\rho E + p)\mathbf{u}) = 0 \,. \tag{1}$$

Herein, ρ is the density, $\mathbf{u}$ the velocity vector and E the specific total energy. In order to close (1), an equation of state $p = p(\rho, e)$ is required for modeling the dependency of the hydrostatic pressure p on density ρ and specific internal energy $e := E - \frac{1}{2}\mathbf{u}^T\mathbf{u}$. For a single polytropic gas, the equation of state (EOS) reads

$$p = (\gamma - 1)\rho e \tag{2}$$

with γ denoting the constant adiabatic exponent. For Eq. (2), the speed of sound in the fluid c is found to be $c = \left(\gamma\, p/\rho\right)^{1/2}$. For very high pressures, as they appear for instance in underwater explosions, a simple extension of Eq. (2) to the *stiffened* gas EOS of the form

$$p = (\gamma - 1)\rho e - \gamma p_\infty \tag{3}$$

is sufficient to model pressure waves in liquids with speed of sound $c = \left(\gamma\,(p + p_\infty)/\rho\right)^{1/2}$. Equation (3) becomes the EOS of a multi-component mixture if we assume a model based on the volume fractions α^i, with $\sum_{i=1}^m \alpha^i = 1$, that defines the mixture quantities as

$$\rho = \sum_{i=1}^m \alpha^i \rho^i \,, \quad \rho\mathbf{u} = \sum_{i=1}^m \alpha^i \rho^i \mathbf{u}^i \,, \quad \rho e = \sum_{i=1}^m \alpha^i \rho^i e^i \,, \quad \frac{p}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i p^i}{\gamma^i - 1} \,, \quad \frac{\gamma p_\infty}{\gamma - 1} = \sum_{i=1}^m \frac{\alpha^i \gamma^i p^i_\infty}{\gamma^i - 1} \,,$$

and in which each component satisfies a stiffened gas EOS of the form $p^i = \left(\gamma^i - 1\right)\rho^i e^i - \gamma^i p^i_\infty$. At this point, several possibilities would exist to derive different sets of governing transport equations for a two-fluid model, however, we choose to follow the approach of Shyue (1998) that supplements system (1) with the two advection equations

$$\frac{\partial}{\partial t}\left(\frac{1}{\gamma - 1}\right) + \mathbf{u} \cdot \nabla \left(\frac{1}{\gamma - 1}\right) = 0 \,, \quad \frac{\partial}{\partial t}\left(\frac{\gamma p_\infty}{\gamma - 1}\right) + \mathbf{u} \cdot \nabla \left(\frac{\gamma p_\infty}{\gamma - 1}\right) = 0 \,. \tag{4}$$

Abgrall (1996) proved that a multi-component continuum scheme needs to satisfy Eq. (4.1) in the discrete sense to prevent unphysical oscillations at material boundaries. Although different scheme alterations are possible to satisfy this requirement, cf. Abgrall and Karni (2001), the utilization of (4) in the governing equations and therefore direct discretization together with (1) is the simplest remedy to the problem, cf. Shyue (1998) and Shyue (2006).

2.2 Finite volume scheme

The appropriate discretization technique for hyperbolic problems with discontinuities (shocks, material boundaries, etc.) is the finite volume approach. For simplicity, we restrict ourselves in the following to the two-dimensional case and assume an equidistant discretization with mesh widths Δx_1, Δx_2 and a constant time step Δt. Since the equations (4) are not in conservation form, we use the Wave Propagation Method by LeVeque (2002) to discretize the system (1), (4). An explicit two-dimensional wave propagation scheme has the formal structure

$$\mathbf{Q}^{l+1}_{jk} = \mathbf{Q}^l_{jk} - \frac{\Delta t}{\Delta x_1}\left(\mathcal{A}^- \Delta_{j+\frac{1}{2},k} + \mathcal{A}^+ \Delta_{j-\frac{1}{2},k}\right) - \frac{\Delta t}{\Delta x_2}\left(\mathcal{B}^- \Delta_{j,k+\frac{1}{2}} + \mathcal{B}^+ \Delta_{j,k-\frac{1}{2}}\right) \,.$$

While the *fluctuations* $\mathcal{A}^{\pm}\Delta$, $\mathcal{B}^{\pm}\Delta$ can be supplemented with (limited) second-order and cross-derivative contributions resulting in a truly multi-dimensional high-resolution Godunov-type method (LeVeque, 2002), the key component is an approximate Riemann solver that provides a linearized decomposition of the Riemann problem (RP) normal to each cell boundary. In here, we use the HLLC[1] approach by Toro et al. (1994) that is tailored specifically for the Euler equations and approximates the RP (here x_1-direction) with three discontinuous jumps by

$$\mathbf{q}^{HLLC}(x_1,t) = \begin{cases} \mathbf{q}_l\,, & x_1 < s_l\,t, \\ \mathbf{q}_l^\star\,, & s_l\,t \le x_1 < s^\star\,t, \\ \mathbf{q}_r^\star\,, & s^\star\,t \le x_1 \le s_r\,t, \\ \mathbf{q}_r\,, & x_1 > s_r\,t, \end{cases}$$

which is also depicted in Fig. 1. For the wave speeds $s_{l/r}$ we use the estimations $s_l = \min\{u_{1,l}-c_l, u_{1,r}-c_r\}$, $s_r = \max\{u_{1,l}+c_l, u_{1,r}+c_r\}$ suggested by Davis (1988) and $s^\star$ is given in the HLLC approach by

$$s^\star = \frac{p_r - p_l + s_l u_{1,l}(s_l - u_{1,l}) - \rho_r u_{1,r}(s_r - u_{1,r})}{\rho_l(s_l - u_{1,l}) - \rho_r(s_r - u_{1,r})}\,.$$

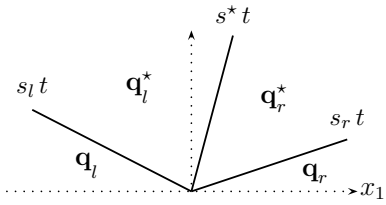


Figure 1: Approximation of the RP by the HLLC scheme.

Conservation arguments and consideration of the structure of the RP for Euler equations lead to the specification of the unknown solution values as

$$\mathbf{q}_k^\star = \left[\eta, \eta s^\star, \eta u_2, \eta\left[\frac{(\rho E)_k}{\rho_k} + (s^\star - u_{1,k})\left(s_k + \frac{p_k}{\rho_k(s_k - u_{1,k})}\right)\right], \frac{1}{\gamma_k - 1}, \frac{\gamma_k p_{\infty,k}}{\gamma_k - 1}\right]^T, \quad \eta = \rho_k \frac{s_k - u_{1,k}}{s_k - s^\star}$$

for $k = \{l, r\}$, cf. Toro (1999). Knowledge of the intermediate state then allows the direct evaluation of the *waves* as $\mathcal{W}_1 = \mathbf{q}_l^\star - \mathbf{q}_l$, $\mathcal{W}_2 = \mathbf{q}_r^\star - \mathbf{q}_l^\star$, $\mathcal{W}_3 = \mathbf{q}_r - \mathbf{q}_r^\star$ and by setting $\lambda_1 = s_l$, $\lambda_2 = s^\star$, $\lambda_3 = s_r$ the fluctuations in the x_1-direction are defined as $\mathcal{A}^-\Delta = \sum_{\lambda_\nu<0} \lambda_\nu\,\mathcal{W}_\nu$, $\mathcal{A}^+\Delta = \sum_{\lambda_\nu\ge 0} \lambda_\nu\,\mathcal{W}_\nu$ for $\nu = \{1, 2, 3\}$.

Note that the robustness and positivity preservation of the HLLC approach is essential for obtaining reliable simulation results when multiple fluids with disparate material properties are involved as it the case in the computations presented in the Sections 6.2 and 6.3.

2.3 Numerical treatment of thin-walled structures

Geometrically complex moving boundaries are incorporated into the finite volume scheme by using some of the cells as ghost cells for enforcing immersed moving wall boundary conditions, cf. Arienti et al. (2003) and Fedkiw et al. (1999). The boundary geometry is mapped onto the Cartesian mesh by employing a scalar level set function ϕ that stores the distance to the boundary surface and allows the efficient evaluation of the boundary outer normal in every mesh point as $\mathbf{n} = -\nabla\phi/|\nabla\phi|$. Since, throughout this paper, we employ only thin-shell and beam solid mechanics solvers that discretize the structure with a topologically possibly open lower dimensional manifold surface mesh, we use *unsigned* distance as level set information and employ those fluid cells as exterior ghost cells for which the cell *midpoint* satisfies $\phi < h/2$.[2] The latter condition is a straightforward, unambiguous solution to achieve the mandatory thickening of the surface mesh by the element thickness h. The contour line $\phi = h/2$ effectively represents the embedded boundary for the fluid solver (depicted as dotted line around shell elements in Fig. 2). The hydrodynamic load on each thin element is then evaluated as the difference between the approximated pressure values at $\phi = h/2$ in the positive and negative direction of each element's normal, i.e. $p^F := p^+ - p^-$.

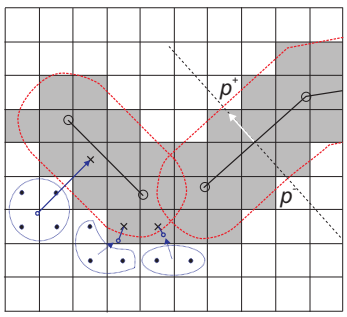


Figure 2: Ghost cells (shaded gray) around shell elements (dark segments) and construction of mirrored values.

For the governing equations (1), the boundary condition at a rigid wall moving with velocity $\mathbf{v}$ is $\mathbf{u}\cdot\mathbf{n} = \mathbf{v}\cdot\mathbf{n}$. Enforcing the latter with ghost cells, in which the discrete values are located at the cell centers, requires the mirroring of the values ρ, $\mathbf{u}$, p, $1/(\gamma-1)$, and $\gamma p_\infty/(\gamma-1)$ across the embedded boundary. The normal velocity in the ghost cells is set to $(2\mathbf{v}\cdot\mathbf{n} - \mathbf{u}\cdot\mathbf{n})\mathbf{n}$, while the mirrored tangential velocity remains unmodified. Mirrored values are constructed by calculating spatially interpolated values in the point $\tilde{\mathbf{x}} = \mathbf{x} + 2\phi\mathbf{n}$ from neighboring interior cells. We employ a dimension-wise

[1] HLLC:Harten-Lax-van Leer Riemann solver with restored Contact surface

[2] For topologically closed boundary surfaces it is also possible to use signed distance instead, cf. (Deiterding, 2006) and (Cirak and Radovitzky, 2005).

linear interpolation for this operation, but it has to be emphasized that directly near the boundary the number of interpolants needs to be decreased to ensure the monotonicity of the numerical solution. This property is essential in simulating hyperbolic problems with discontinuities. Figure 2 also highlights the necessary reduction of the interpolation stencil for some exemplary cases. The interpolation locations are indicated by the origins of the arrows normal to the contour line that defines the embedded boundary. After the application of the numerical scheme, cells that have been used to impose internal boundary conditions are set to the entire state vector of the nearest cell in the fluid interior. This operation ensures proper values in case such a cell becomes a regular interior cell in the next step due to boundary movement. The consideration of **v** in the ghost cells also guarantees that the embedded boundary propagates at most one cell in every time step.

Note that the described technique does not require a modification of the numerical stencil itself and is therefore generically applicable, but causes a slight diffusion of the boundary location throughout the method and results in an overall non-conservative scheme. The boundary undergoes a staircase approximation that potentially can give rise to considerable errors in the computed solution. However, by refining the embedded boundary, typically up to the highest available resolution, with the dynamic mesh adaptation method described in next subsection, we alleviate these problems effectively. A refinement criterion based on $\phi \equiv 0$ has been implemented for this purpose.

2.4 Structured adaptive mesh refinement

In order to supply a fine local temporal and spatial resolution efficiently, we use the block-structured adaptive mesh refinement (SAMR) method by Berger and Colella (1988). Characteristic for the SAMR method is that a specific finite volume method is technically not implemented in a cell-based fashion, but rather in a routine operating on equidistant subgrids. The subgrids become computationally decoupled during one update cycle through the use of ghost or halo cells. Starting from the base mesh on level 0, the time step size and all spatial mesh widths on level $l > 0$ are r_l-times finer than on level $l - 1$ and a time-explicit finite volume scheme will (in principle) remain stable on all levels of the recursively nested refinement hierarchy. Ghost cell values at coarse-fine interfaces are constructed by interpolating coarse level data, which mandates a recursive order of update (cf. Sec. 5.1). SAMR in the VTF is provided generically by the AMROC (Adaptive Mesh Refinement in Object-oriented C++) framework (Deiterding, 2002) that can be used on all parallel systems that provide the MPI library.

3 Lagrangian thin-shell solver

The Kirchhoff-Love thin-shell model applied here for three-dimensional thin-shell simulation has been discretized with smooth subdivision finite elements, as previously described by Cirak and Ortiz (2001) and Cirak et al. (2000). Notably, the underlying kinematic assumptions allow for finite strains, displacements and rotations. The subdivision shell elements have also been extended to the range of applications that involve fracture and fragmentation, cf. Cirak et al. (2005). Thereby, fracture initiation and propagation is considered as a progressive failure phenomenon in which the separation of the crack flanks is modeled with a cohesive law. In the present implementation, cohesive interface elements are inserted at all inter-element edges and constrain the opening of the crack flanks to the deformation of the shell mid-surface and its normal.

3.1 Governing equations in weak form

To kinematically describe a possibly fractured thin-shell as sketched in Fig. 3(a), we first consider the shell in its undeformed configuration $\overline{V}$. The position vector φ of a material point on the undeformed shell $\overline{\varphi}$ of a material point on the undeformed shell body is assumed to be

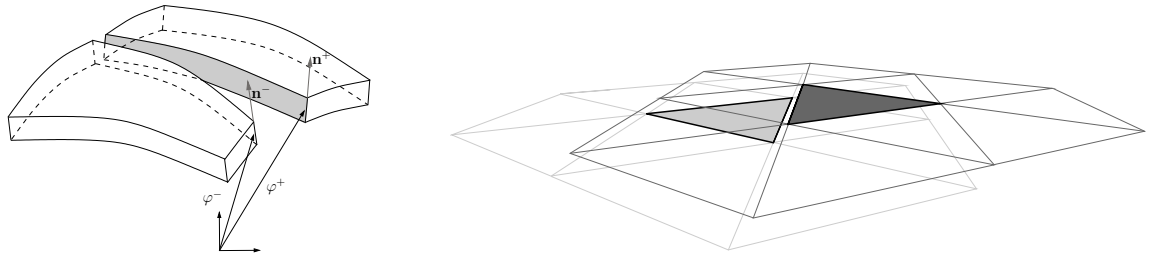
$$\overline{\varphi} = \overline{\mathbf{x}} + \theta_3 \overline{\mathbf{n}} \tag{5}$$

with the uniform thickness $\overline{h}$ and $-\overline{h}/2 \leq \theta_3 \leq \overline{h}/2$. The position vector of the shell mid-surface is denoted by $\overline{\mathbf{x}}$ and its out-of-surface unit normal by $\overline{\mathbf{n}}$. In other words, the shell mid-surface represents a two-dimensional manifold in $\mathbb{R}^3$. The deformation mapping φ maps the shell body into the deformed configuration V

$$\varphi = \mathbf{x} + \theta_3 \lambda \mathbf{n} \tag{6}$$

where **x** and **n** are the deformed mid-surface and its normal. The thickness stretch parameter λ is the ratio of the deformed shell thickness h to the reference thickness $\overline{h}$. In the presence of a crack, the deformation is discontinuous across the crack and has a jump, i.e.

$$[\![\varphi]\!] = \varphi^+ - \varphi^- = [\![\mathbf{x}]\!] + \theta_3 [\![\mathbf{n}]\!] , \tag{7}$$



(a) (b)

Figure 3: (a) Fractured shell body: opposite crack flanks and corresponding normals. (b) A cohesive edge and its two adjacent elements.

where the superscripts $+$ and $-$ refer to the opposing crack flanks. Further, the first term describes the discontinuity of the deformation of the shell mid-surface, and the second term the discontinuity in the shell out-of-surface normal. The discontinuities in the deformations can also be interpreted as the opening displacement of the crack. Further, note that the Kirchhoff-Love constraint is satisfied, i.e. $\mathbf{x} \cdot \mathbf{n} = 0$, on both sides of the crack.

A standard semi-inverse approach is followed for obtaining the shell equilibrium equations in weak form. To this end, the assumed reduced kinematic equations for the shell body (Equations (5) and (7)) are introduced into the conventional virtual work expression for the three-dimensional body. As previously mentioned, we consider fracture as a gradual separation phenomenon, resisted by cohesive tractions. Consequently, the internal virtual work expression contains the virtual work of the cohesive interface ($\delta\Pi_{C,\text{int}}$) in addition to the virtual work of the bulk material ($\delta\Pi_{S,\text{int}}$)

$$\delta\Pi_{S,\text{int}} + \delta\Pi_{C,\text{int}} - \delta\Pi_{\text{ext}} = 0 \tag{8}$$

with the external virtual work $\delta\Pi_{\text{ext}}$ and

$$\delta\Pi_{S,\text{int}} = \int_{\overline{\Omega}} \int_{-\overline{h}/2}^{\overline{h}/2} \mathbf{P} : \delta\mathbf{F}\, \mu\, d\theta^3 d\overline{\Omega}\,, \quad \delta\Pi_{C,\text{int}} = \int_{\overline{\Gamma}_C} \int_{-\overline{h}/2}^{\overline{h}/2} \mathbf{T} \cdot [\![\varphi]\!]\, \mu\, d\theta^3 d\overline{\Gamma}_C\,,$$

where $\mathbf{P}$ is the first Piola-Kirchhoff stress tensor, $\mathbf{T}$ the related traction vector at the cohesive surface, and $\mathbf{F}$ the deformation gradient. The virtual work expression for the bulk material is integrated over the undeformed shell mid-surface $\overline{\Omega}$ and for the cohesive interface over the crack path $\overline{\Gamma}_C$. The scalar factor μ accounts for the curvature of the shell in the volume computation, cf. Cirak and Ortiz (2001).

3.2 Subdivision thin-shell elements

Next, we briefly outline the discretization of the governing equation (8) firstly for the non-fractured case. A detailed presentation of the used subdivision finite element discretization technique can be found in (Cirak et al., 2000) and (Cirak and Ortiz, 2001). In this approach, the reference ($\overline{\mathbf{x}}$) and deformed ($\mathbf{x}$) shell surfaces are approximated using smooth subdivision surfaces belonging to the Sobolev space H^2 with square-integrable curvatures. The subdivision interpolation within one element is accomplished with shape functions, which have support on the element as well as on the one-ring of neighboring elements. The overlapping local subdivision interpolants, each defined over one patch, together lead to a global interpolant with square-integrable curvatures. Importantly, smoothness is achieved without introducing nodal rotations as degrees of freedom. The absence of nodal rotations is particularly appealing in the presence of finite rotations.

In the presence of fracture, the smoothness and/or continuity of the interpolation has to be relaxed and the subdivision interpolant needs to be modified, cf. Cirak et al. (2005). The topological changes necessary to the non-local subdivision functions and the underlying control mesh in order to describe the dynamic propagation of a single crack are complicated. Therefore, we chose to pre-fracture the element patches, such that each patch possesses its own nodes and acts independently for the purpose of interpolation, see Fig. 3(b). Prior to crack nucleation, the coupling of the elements is enforced by applying stiff elastic cohesive interfaces at all edges. Once fracture nucleates along an element edge, the element patches on both sides of the cracked edge interact through cohesive tractions. The cohesive tractions are self-balanced internal forces derived from a cohesive fracture model.

3.3 Constitutive models for the shell

An irreversible cohesive constitutive model as proposed by Ortiz and Pandolfi (1999) is used for modeling the cracks. Thereby, the opening displacement $[\![\varphi]\!]$ plays the role of a deformation measure while the traction $\mathbf{T}$ is the conjugate stress measure. Further, a scalar effective opening displacement is defined by

$$\delta = \sqrt{\beta^2 |\delta_t|^2 + |\delta_n|^2}\,,$$

where δ_t and δ_n are the tangential and normal displacement components of $[\![\varphi]\!]$ with respect to the crack surface. The parameter β assigns different weights to the tangential and normal opening displacements. The cohesive tractions $\mathbf{T}$ are given by

$$\mathbf{T} = \frac{t}{\delta}(\beta^2 \delta_t + \delta_n)\,.$$

The scalar effective traction t is computed from a cohesive law as shown in Fig. 4. In addition to the parameter β, the model parameters are the maximal tensile stress σ_c and the critical opening displacement δ_c. The following relationship between the cohesive law and the critical fracture energy rate G_c exists:

$$G_c = \int_0^\infty t\, d\delta$$

which can be used for determining δ_c. For further details see Ortiz and Pandolfi (1999).

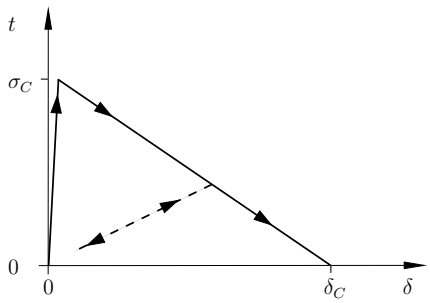


Figure 4: Irreversible linear cohesive law. Note that the initial stiff elastic response enforces the displacement continuity prior to crack initiation at σ_c. The dashed line represents the loading-unloading rule.

The inelastic behavior of the bulk material, i.e. the relation between $\mathbf{P}$ and $\mathbf{F}$, is described with a conventional J_2 viscoplasticity model with isotropic power-law hardening as described by Cuitino and Ortiz (1992). The power-law hardening for the flow stress g has the form

$$g(\epsilon^p) = \sigma_y \left(1 + \frac{\epsilon^p}{\epsilon_0^p}\right)^{1/n},$$

where σ_y is the initial yield stress, ϵ^p and ϵ_0^p are the total and the reference plastic strains, respectively, and $1/n$ is the hardening exponent. The rate-dependent behavior is described in terms of the effective von Mises stress σ_{eff} with a power viscosity law and constant rate sensitivity

$$\sigma_{\text{eff}} = g(\epsilon^p) \left(1 + \frac{\dot{\epsilon}^p}{\dot{\epsilon}_0^p}\right)^{1/m},$$

where $\dot{\epsilon}_0^p$ is the reference plastic strain rate and $1/m$ the strain rate sensitivity exponent.

The thin-shell typical plane stress condition is enforced with a local Newton-Raphson iteration at each quadrature point, cf. deBorst (1991). Thereby, the thickness stretch parameter λ (Eq. 6) is the unknown variable in the iteration.

4 Efficient level set evaluation

In Section 2, we have sketched the concept of employing a distance function to represent a complex embedded boundary on a Cartesian mesh. While distance functions are easily prescribed for single elementary geometric objects, their evaluation can be cumbersome for complex shapes. In coupled Eulerian-Lagrangian simulations, this complex shape is defined by the deforming shell surface mesh.

One can efficiently compute the distance on a grid by solving the eikonal equation with the method of characteristics and utilizing polyhedron scan conversion, cf. Mauch (2003). For a given grid point, the relevant closest point on the triangular mesh lies on one of the primitives (faces, edges and vertices) that comprise the surface. The characteristics emanating from each of these primitives form polyhedral shapes. Such a *characteristic polyhedron* contains all of the points which are possibly closest to its corresponding face, edge or vertex. The closest points to a triangle face must lie within a triangular prism defined by the face and its normal; the closest points to an edge

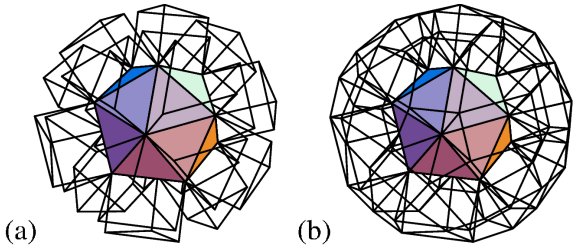


Figure 5: The characteristic polyhedra for faces and edges of an icosahedron.

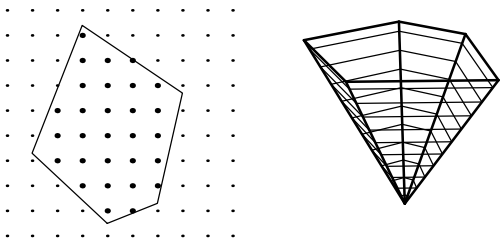


Figure 6: Slicing of a polyhedron to form two-dimensional polygons (right) and scan conversion of an exemplary case (left).

lie in a cylindrical wedge defined by the line segment and the normals to the two incident faces (see Fig. 5 for face (a) and edge (b) polyhedra for a particular example). Analogously, polygonal pyramids emanating from the vertices are also possible (not shown). We then determine the grid points that lie inside a characteristic polyhedron with polyhedron scan conversion. The polyhedron is first sliced along each sheet of the grid lattice to produce polygons, cf. Fig. 6. Simple geometric formulas are finally used to calculate the distance once a polyhedron has been assigned uniquely to each grid point.

By utilizing the outlined techniques, and evaluating the distance exactly only within a small distance around the surface, a highly efficient algorithm can be formulated that has linear computational complexity both in the number of Cartesian mesh points and the surface triangles, cf. Mauch (2003) and Deiterding et al. (2006c).

5 Fluid-structure coupling

The explicit fluid and solid solvers are weakly coupled by successively applying appropriate boundary conditions in a time-operator splitting technique. In the case of inviscid flows, the compatibility conditions are the continuity of the velocity component normal to the embedded boundary u_n in solid (S) and fluid (F), i.e. $u_n^S = u_n^F$, and the continuity of the normal component of the solid's Cauchy traction vector, $p^S = (\sigma \mathbf{n})\mathbf{n}$ with $\sigma = 1/det(\mathbf{F})\mathbf{FP}$, and the hydrodynamic pressure p^F, i.e $p^S = p^F$. We use the following update algorithm to implement these coupling conditions numerically:

update $\phi(t)$
$\mathbf{v}_F^{+/-} := \mathbf{u}^S(t)$
update_fluid(Δt)
$p^S := p^F(t + \Delta t)$
update_solid(Δt)
$t := t + \Delta t$

After evaluating the distance function ϕ for the currently available shell surface mesh, the embedded wall boundary velocities for the fluid solver are set to the solid velocities in the nearest shell element midplane. The same velocity $\mathbf{v}$ is enforced in the fluid on upper ($^+$) and lower ($^-$) side of each element. After setting embedded rigid wall boundary conditions and the fluid update, a new hydrodynamic pressure load $p^F := p^+ - p^-$ on each shell element (compare Fig. 2) is derived by evaluating $p^{\pm}$ with the linear interpolation / extrapolation operation already sketched in Section 2.3. With these new boundary conditions, the cycle is completed by advancing the solid by Δt, which in practice is typically done by taking multiple, smaller time steps in the solid solver to effectively accommodate the more restrictive stability condition in the solid.

5.1 Application of SAMR in the fluid solver

While the implementation of a loosely coupled FSI method is straightforward with conventional solvers with consecutive time update, the utilization of the recursive SAMR method with hierarchical time step refinement in the fluid is non-apparent. In the VTF, we treat the fluid-solid interface as a discontinuity that is a-priori refined at least up to a coupling level l_c. The resolution at level l_c has to be sufficiently fine to ensure an accurate wave transmission between fluid and structure, but might not be the highest level of refinement. To incorporate the fluid-structure data exchange into the recursive SAMR algorithm it has to be ensured that the updated mesh positions and nodal velocities are received *before* a regridding of the coupling level l_c is initiated and that the hydrodynamic pressure loadings

on the interface are evaluated *after* the highest available refinement level has reached the same discrete time as the updated level l_c. We visualize the data exchange between solid and SAMR fluid solver in Fig. 7 for an exemplary SAMR hierarchy with two additional levels with $r_{1,2} = 2$. Figure 7 pictures the recursion in the SAMR method by numbering the fluid update steps (F) according to the order determined by the SAMR method. The order of the solid update steps (S) on the other hand is strictly linear. The red diagonal arrows correspond to the sending of the interface pressures p^F from fluid to solid at the end of each time step on level l_c. The blue upward arrows visualize the sending of the interface mesh and its nodal velocities $\mathbf{u}^S$ after each solid update. The modification of refinement meshes is indicated in Fig. 7 by the gray arrows; the initiating base level, that remains fixed throughout the regridding operation, is indicated by the gray circles.

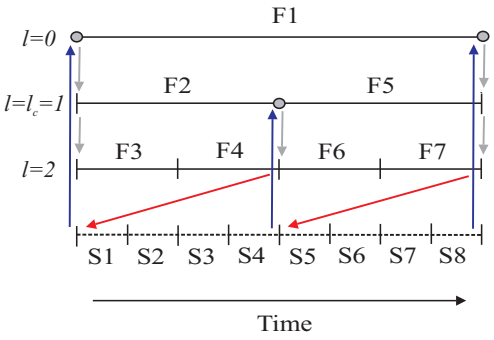


Figure 7: Data exchange between the recursive fluid solver and the linear thin-shell solver during one SAMR root level time step.

5.2 Software implementation

The fluid-structure coupling software VTF is a collection of C++ classes for implementing high-speed FSI problems on distributed memory machines. At present, only the loose coupling of time-explicit solvers is supported. The design follows a classical framework approach in which the instantiation of the main objects is usually done in a short generic main program and customization is achieved through subclass derivation.

In our current implementation, computational fluid and solid dynamics solvers are parallelized separately for distributed memory machines using independent rigorous domain decomposition methods. In order to facilitate an efficient communication of the distributed fluid-shell boundary information we have implemented a non-blocking high-level communication library that determines the necessary point-to-point communication patterns by intersecting Cartesian bounding boxes enclosing the local domains. Details on this communication library and also a detailed algorithmic description of the coupled SAMR method can be found in (Deiterding et al., 2006c).

6 Computational results

In the following sub-sections, we consider three different configurations to demonstrate the versatility of our fluid-structure coupling approach. While Section 6.1 discusses a verification test of linearly elastic motion due to the impact of a shock wave in air, the computations in Sections 6.2 and 6.3 simulate viscoplastic deformation and fracture driven by strong pressure waves in water. In both cases the induced solid motion causes tension in the liquid that results in cavitation when the pressure falls below the water vapor pressure. Since cavitation inception limits the maximal tension, and thereby the minimal pressure the liquid can support, we employ an additional pressure cutoff model (see also Xie et al. (2006)) in these computations. It is implemented by applying the non-conservative energy correction

$$E := \frac{p_c + \gamma p_\infty}{\rho(\gamma - 1)} + \frac{1}{2}\mathbf{u}^T\mathbf{u}\,, \quad \text{for } p < p_c \tag{9}$$

after every fluid time step and its purpose is to limit all hydrodynamic pressures to the cutoff value p_c.

6.1 Verification

As a first test for the coupled method, we consider the verification configuration of a thin-walled steel panel impacted by a planar shock wave in air ($\gamma^A = 1.4$, $p^A_\infty = 0$) proposed by Giordano et al. (2005). The panel has the thickness $h = 1\,\mathrm{mm}$ and extends $50\,\mathrm{mm}$ from a mounting with forward-facing step geometry into which it is firmly clamped. Figure 8 depicts the computational domain and initial conditions. Inflow boundary conditions are applied on the left side, rigid wall boundary conditions anywhere else.

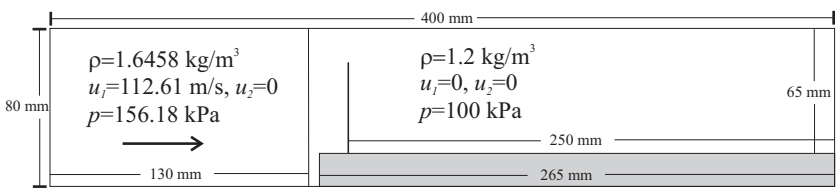
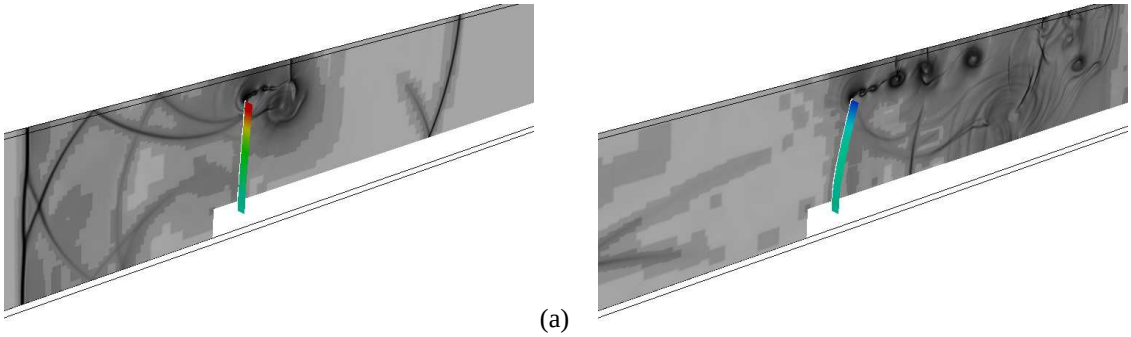


Figure 8: Geometry of the computational setup for the shock-panel testcase.



(a) (b)

Figure 9: Snapshots at $t = 0.43\,\mathrm{ms}$ (a) and $t = 1.56\,\mathrm{ms}$ (b) visualize the evolution of the fluid mesh adaptation (different levels indicated by gray scales) according to the flow field and the deflection of the thin-shell panel.

.

First, we model the panel as a one-dimensional elastic beam immersed into a two-dimensional fluid domain and solve the Euler-Bernoulli beam equation

$$\rho_s h \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial \bar{x}^4} = p^F \tag{10}$$

for $\rho_s = 7600\,\mathrm{kg/m^3}$, $E = 220\,\mathrm{GPa}$, and $I = h^3/12$ to calculate the deflection $w(\bar{x}, t)$ of the beam middle axis with updated hydrodynamic loading p^F after every time step of the fluid solver. Equation (10) is approximated with a straightforward time-implicit finite difference approach and the resulting linear system is solved by QR decomposition. An equidistant mesh of 101 points is used to discretize the beam middle axis. The fluid mesh uses a base grid of 320×64 cells and allows up to two additional levels of dynamic isotropic refinement (based on ϕ and scaled gradients of ρ and p) with refinement factors $r_{1,2} = 2$. Calculating $18,660$ coupled time steps at $l_c = 2$ to $t_e = 5.0\,\mathrm{ms}$ required $\sim 12.3\,\mathrm{h}$ CPU on a $3.4\,\mathrm{GHz}$ processor.

Second, the panel is modeled as a two-dimensional plate strip of $5\,\mathrm{mm}$ width (325 triangular elements) and embedded into a three-dimensional fluid base mesh of $320 \times 64 \times 2$ cells. The structural motion is computed with the previously described three-dimensional thin-shell finite element solver. Linear elastic material behavior with the parameters given above and the Poisson ratio $\nu = 0.3$ is assumed. Figure 9 shows the dynamic bending of the plate strip and the evolving fluid mesh adaptation with two additional levels (depicted by gray scales) as the initial shock is partially reflected (a) and increased vortex shedding occurs at the panel tip (b). The three-dimensional computation reached $t_e = 5.0\,\mathrm{ms}$ after $18,000$ coupled time steps, where 5 solid solver sub-steps were used, and required $\sim 322\,\mathrm{h}$ CPU.

A comparison of the predicted panel tip displacement versus time for both FSI simulations is given in Fig. 10. Slight divergences at later times are expectable considering the apparent differences in beam and shell theory. However, both FSI configurations show excellent agreement at earlier times and are in proximity to a-priori predictions based on Eq. (10) alone (Beam) for the constant instantaneous loading $p^F \equiv 100\,\mathrm{kPa}$ that has been found to be a rough approximation for the pressure loading for approximately $t < 2.5\,\mathrm{ms}$, cf. Giordano et al. (2005).

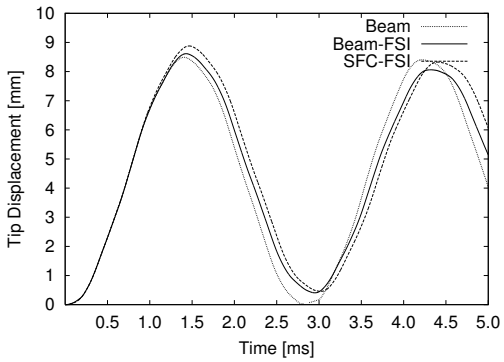
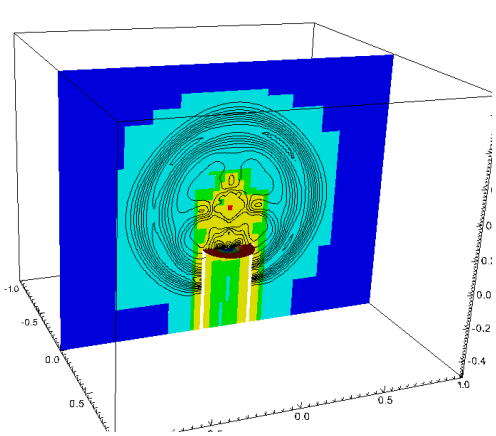


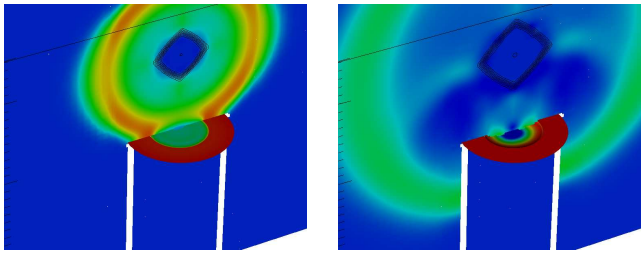
Figure 10: Tip motion for both FSI computations and reference computation with constant loading.

6.2 Underwater explosion

As a first test for shock waves in water, we simulate a fluid-structure experiment by Ashani and Ghamsari (2008). A small charge ($m_{C4} = 20\,\mathrm{g}$ and $m_{C4} = 30\,\mathrm{g}$) of the explosive C4 ($1.34\times$ TNT) is detonated in a water-filled basin at the standoff distances $d = 25\,\mathrm{cm}$ or $d = 30\,\mathrm{cm}$ above a circular air-backed aluminum plate (exposed radius $85\,\mathrm{mm}$) of thickness $h = 3\,\mathrm{mm}$. We model the basin with a fluid domain of $2\,\mathrm{m} \times 1.6\,\mathrm{m} \times 2\,\mathrm{m}$, where the origin is placed at the domain center. Outflow is assumed at all domain boundaries. In analogy to the experiment, air-backed conditions are represented by inserting a rigid cylinder of radius $150\,\mathrm{mm}$ from $x_2 = -0.8\,\mathrm{m}$ to $x_2 = 0$ into the domain. The cylinder is sealed by the test plate of radius $150\,\mathrm{mm}$, discretized with 8148 triangles, which is treated as rigid for $r > 85\,\mathrm{mm}$. The material parameters for viscoplastic material behavior of aluminum, that were used in these simulations, are detailed in Table 1. It is assumed that the aluminum is strain-rate insensitive



(a)



(b) (c)

Figure 11: (a) Isolines of p on domains of refinement levels (indicated by color) at $t = 0.31\,\mathrm{ms}$. (b), (c) The plane shows a color plot of p and isolines of α^A, the plate displays the normal vertex velocity at $t = 0.14\,\mathrm{ms}$ and $0.31\,\mathrm{ms}$.

and strain-softening effects have not been considered. The cylinder is filled with air ($\gamma^A = 1.4$, $p^A_\infty = 0$) at density $\rho^A = 1.29\,\mathrm{kg/m^3}$, the basin with water ($\gamma^W = 7.415$, $p^W_\infty = 296.2\,\mathrm{MPa}$) at $\rho^W = 1027\,\mathrm{kg/m^3}$, which are both initially at rest and assumed to be at atmospheric pressure $p_0 = 100\,\mathrm{kPA}$ (the static pressure increase due the water depth can safely be neglected). The modification by Eq. (9) with $p_c = -1\,\mathrm{MPa}$ is employed in the fluid solver to consider cavitation inception. The shock from the explosion is modeled as a spherical energy deposition ($m_{C4} \cdot 6.06\,\mathrm{MJ/kg}$) uniformly distributed over a sphere of radius $5\,\mathrm{mm}$ of air at temperature $1500^o\mathrm{C}$ located at $(0, d, 0)$.

The fluid domain is discretized with an SAMR base mesh of $50 \times 40 \times 50$ cells. Four additional levels with refinement factors $r_{1,2,3} = 2, r_4 = 4$ are employed. The highest level refinement is static and restricted to the explosion center. Fluid mesh adaptation on all other levels is dynamic and based on ϕ and the scaled gradient of p. However, refinement at levels 2 and 3 is restricted to the immediate vicinity of the structure and the shock as it impinges onto it. Figure 11(a) depicts a snapshot of the fluid mesh in a plane through the center of the domain for the case $m_{C4} = 20\,\mathrm{g}$, $d = 25\,\mathrm{cm}$. The FSI simulation uses $l_c = 3$ with 2 solid solver sub-steps, and 1296 coupled time steps were computed to reach the final time $t_e = 1\,\mathrm{ms}$.

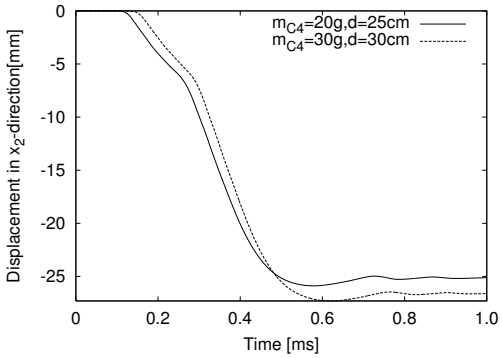


Table 1: Material properties for aluminum.

Mass density	$\rho_s = 2719\,\mathrm{kg/m^3}$
Young's modulus	$E = 69\,\mathrm{GPa}$
Poisson's ratio	$\nu = 0.33$
Yield stress	$\sigma_y = 217.6\,\mathrm{MPa}$
Reference plastic strain	$\epsilon^p_0 = 0.025$
Hardening exponent	$1/n = 0.6$
Rate sensitivity exponent	$1/m = 0$

Figure 12: Left: center displacement versus time.

The impact of the spherical shock onto the plate and its partial reflection are visualized in graphics (b) and (c) of Fig. 11, respectively. The induced motion of the exposed part of the test specimen is clearly visible. Figure 12 displays the plate center motion versus time for both cases considered. Note that during the first $\sim 0.2\,\mathrm{ms}$ after the shock impact the deformation occurs with constant velocity since the water near the plates cavitates and does not transmit significant forces onto the plate. The maximal computed deflection for the case $m = 20\,\mathrm{g}$, $d = 25\,\mathrm{cm}$ is $25.88\,\mathrm{mm}$, for the case $m_{C4} = 30\,\mathrm{g}$, $d = 30\,\mathrm{cm}$ it is $27.31\,\mathrm{mm}$. Those values compare reasonably well to the experimental measurements of $28.83\,\mathrm{mm}$ and $30.09\,\mathrm{mm}$ by Ashani and Ghamsari (2008), where the differences are primarily due to our rather simplistic modeling of the initial shock wave created by the explosion. Both computations were run on 12 nodes of a parallel cluster with Intel-$3.4\,\mathrm{GHz}$-Xeon dual processors (10 nodes fluid, 2 nodes solid dynamics solver) and required $\sim 130\,\mathrm{h}$ CPU each ($\sim 5.4\,\mathrm{h}$ wall time).

6.3 Plate deformation from water hammer

The final configuration considered is an experimental setup developed by Deshpande et al. (2006) that reproduces loading conditions of large-scale underwater explosions in the laboratory. By firing a steel projectile onto a piston

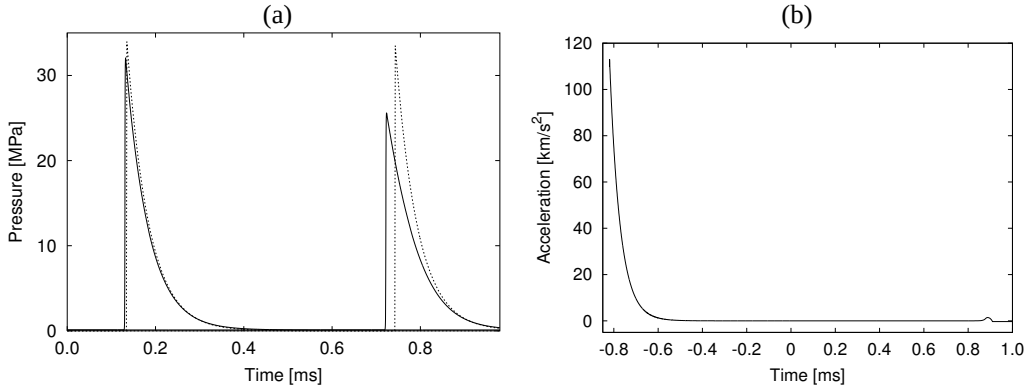


Figure 13: Loading conditions for $p_{\max} = 34\,\mathrm{MPa}$. (a) Comparison of the traveling wave approximation Eq. (12) (dotted) with computed pressure traces (solid) at $x_1 = 1.1\,\mathrm{m}$ (left) and $x_1 = 0.2\,\mathrm{m}$ (right). (b) Computed piston acceleration for the entire simulation time.

inserted into the end of a water shock tube, a strong pressure wave is created that propagates through the water column and impinges onto a circular copper plate sealing the other end. The shock tube has a length of $1.3\,\mathrm{m}$ and a radius of $32\,\mathrm{mm}$, which is modeled with an additional signed distance level set function. The tube is filled with water ($\gamma^W = 7.415$, $p_\infty^W = 296.2\,\mathrm{MPa}$, $p_c = 0$) of density $\rho^w = 1000\,\mathrm{kg/m^3}$ at atmospheric pressure $p_0 = 101.3\,\mathrm{kPa}$. The copper plate has a thickness of $0.25\,\mathrm{mm}$ and a radius of $r = 56\,\mathrm{mm}$, but is unconstrained only for $r < 32\,\mathrm{mm}$. Again, the plate is air-backed ($\rho^A = 1.0\,\mathrm{kg/m^3}$, $\gamma^A = 1.4$, $p_\infty^A = 0$) at pressure p_0. The center of the plate is initially located at the coordinate origin and the shock tube middle axis is aligned with the x_1-axis.

Verification of loading conditions

The motion of the piston creates a quasi one-dimensional pressure spike with exponential decay rate that propagates through the water column. We incorporate the piston movement into the computational setup by employing a second signed distance level set function that corresponds to the piston boundary in contact with the fluid. The level set is initially positioned at $x_1 = 1.3\,\mathrm{m}$ and assumed to move with constant velocity b_0. During a simulation, we integrate the law of motion for the piston

$$\bar{m}\dot{b} = -(\bar{p} - p_0) \tag{11}$$

with the forward Euler method and update level set position and velocity b in direction of the tube middle axis (to be used as wall normal velocity v_n for this level set, cf. Sec. 2.3) in every time step. As the piston is constrained in all other directions, it suffices to consider the hydrodynamic pressure averaged across the piston boundary $\bar{p}$ and to use the averaged piston mass per unit area $\bar{m}$. In all subsequent computations the value $\bar{m} = 74.1\,\mathrm{kg/m^2}$ was used.

By assuming the wave to propagate with the constant speed of sound in water $c^w = 1482\,m/s$, a traveling wave solution of the form

$$p(x_1, t) = \begin{cases} p_{\max} \exp\left(-\dfrac{t - x_1/c^w}{\bar{m}/(\rho^w c^w)}\right) & , \quad t \geq \dfrac{x_1}{c^w} \\ 0 & , \quad \text{otherwise} \end{cases} \tag{12}$$

can be derived for the pressure evolution at a fixed spatial location that is found to be in reasonable agreement with experimental measurements close to the piston boundary (Deshpande et al., 2006). The maximum of the traveling wave is set to $p_{\max} = \rho^w c^w b_0$, which permits the computation of the initial piston velocity b_0 for given $p_{\max}$. Plot (a) of Fig. 13 compares Eq. (12) evaluated at the locations $x_1 = 1.1\,\mathrm{m}$ and $x_1 = 0.2\,\mathrm{m}$ with pressure traces derived from a one-dimensional unigrid finite volume simulation with 2700 cells on the domain $[0\,\mathrm{m}, 1.35\,\mathrm{m}]$ for $p_{\max} = 34\,\mathrm{MPa}$. The agreement at early times is very good, verifying the correctness of our computational setup. At later times, the simulation necessarily differs from the unaltered (non-dispersive) traveling wave, because our computational model considers the density rise in the compression wave and the resulting change of the speed of sound. Plot (b) of Fig. 13 displays the computed piston acceleration, where the time is shifted by $-0.82\,\mathrm{ms}$ setting the start of the FSI simulations to $t = 0$.

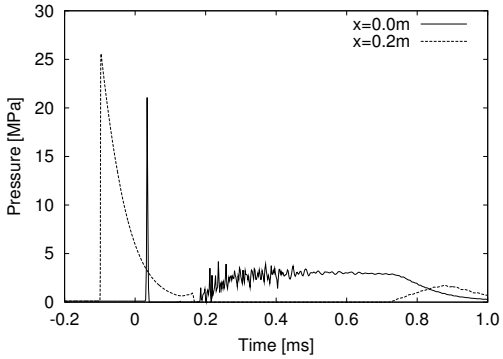


Figure 14: Fluid pressure traces along tube middle axis at $x_1 = 0$ and $x_1 = 0.2\,\mathrm{m}$ for $p_{\max} = 34\,\mathrm{MPa}$.

Table 2: Material and cohesive model parameters for annealed copper.

Mass density	$\rho_s = 8920\,\mathrm{kg/m^3}$
Young's modulus	$E = 130\,\mathrm{GPa}$
Poisson's ratio	$\nu = 0.31$
Yield stress	$\sigma_y = 38.5\,\mathrm{MPa}$
Reference plastic strain	$\epsilon_0^p = 0.0091$
Hardening exponent	$1/n = 0.627$
Reference plastic strain rate	$\dot{\epsilon}_0^p = 1000$
Rate sensitivity exponent	$1/m = 1.0$
Maximal tensile stress	$\sigma_c = 525\,\mathrm{MPa}$,
Critical opening displacement	$\delta_c \approx 0.3\,\mathrm{mm}$
Tangential weight parameter	$\beta^2 = 0.8$

Fluid-structure interaction simulations

While the thin copper plate exhibits just viscoplastic deformation for $p_{\max} = 34\,\mathrm{MPa}$, fracture occurs for larger values of $p_{\max}$. All computations were therefore run with activated cohesive interface element capability of the thin-shell solver. The parameters for the J_2 plasticity and the cohesive interface model used for annealed copper are given in Table 2. The mesh for the plate considers the mounting holes for purely optical reasons and consists of 8896 triangles. In order to ensure the correct boundary conditions throughout the whole simulation, the three-dimensional fluid domain covers with $[-0.146\,\mathrm{m}, 1.35\,\mathrm{m}] \times [-0.04\,\mathrm{m}, 0.04\,\mathrm{m}] \times [-0.04\,\mathrm{m}, 0.04\,\mathrm{m}]$ the entire shock tube.

To concentrate the computational resources for the fluid in the region of interest, the computation uses an SAMR base mesh of $374 \times 20 \times 20$ cells and two additional levels with refinement factors $r_{1,2} = 2$. The fluid mesh is always fully refined along the plate boundary and at the front of the incoming pressure wave. The shock tube boundary is statically refined at level 2 for $x_1 < 0.206\,\mathrm{m}$ and at level 1 for $x_1 < 0.43\,\mathrm{m}$. The coupling level for the fluid-structure data exchange is set to $l_c = 2$. The simulations were run on 12 nodes of a parallel cluster consisting of Intel-$3.4\,\mathrm{GHz}$-Xeon dual processors CPUs, where 6 nodes were employed for the fluid and the solid solver, respectively. Using 2 solid solver sub-steps the case $p_{\max} = 34\,\mathrm{MPa}$ took 4120 coupled time steps to reach $t_e = 1.0\,\mathrm{ms}$, which required $\sim 48\,\mathrm{h}$ wallclock time ($\sim 1150\,\mathrm{h}$ CPU).

Results from the non-rupture case are displayed in the Figs. 14 and 15. Figure 14 shows pressure traces along the tube middle axis at the locations $x_1 = 0.2\,\mathrm{m}$ and $x_1 = 0$. The impact of the pressure wave onto the plate at $t \approx 0.03\,\mathrm{ms}$ and the water cavitation immediately after can be clearly inferred. An expansion wave due to the resulting structural motion travels upstream through the water column inducing a small piston acceleration around $t = 0.9\,\mathrm{ms}$, cf. Fig. 13(b).

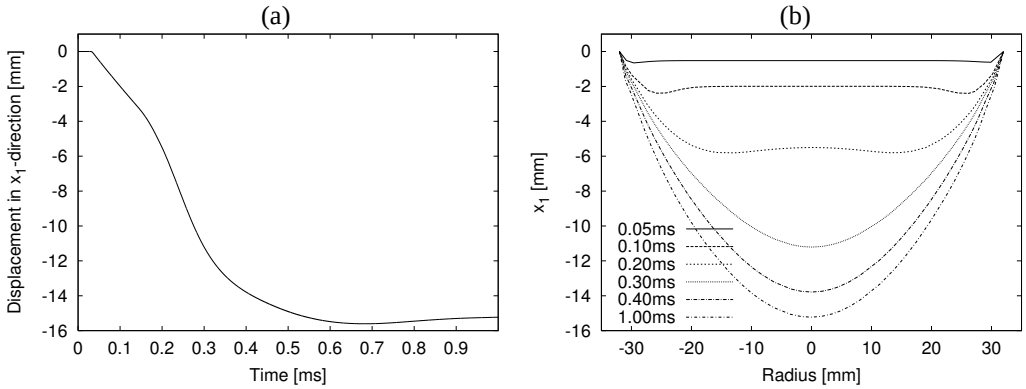


Figure 15: (a) Plate center displacement versus time and (b) deflection history of the plate midplane during the coupled simulation.

Similarly to the previous section, the plate deformation is separated into two phases. As can be seen in Fig. 15(a) cavitation leads to a constant displacement velocity during the first $\sim 0.15\,\mathrm{ms}$. The deflection of the plate is characterized by the appearance of a stationary plastic hinge at the boundary and a second instationary plastic hinge traveling towards the plate center, cf Fig. 15(b). Shortly after the fluid expansion wave leaves the plate

boundary, a hydrodynamic pressure of $\sim 4\,\mathrm{MPa}$ builds up again directly at the plate until it declines finally from $t \approx 0.76\,\mathrm{ms}$ on, cf. Fig. 14. During this phase, the plate deformation is nonlinear and the plate deflection is convex. The displacement reaches its maximum at $t \approx 0.76\,\mathrm{ms}$ and remains almost unaltered until the end of the simulation at $t_e = 1.0\,\mathrm{ms}$. A visual comparison of the finally deformed solid mesh in the FSI simulation with a photograph of a target plate by Deshpande et al. is given in Fig. 16. The agreement is apparently very good. The obtained maximum deflection of $15.6\,\mathrm{mm}$ is also in good agreement with the analytic estimate of $16.1\,\mathrm{mm}$ by Qui et al. (2004) (Equation 21a in (Qui et al., 2004)). Note that Qui et al.'s estimate is for an ideally plastic material. In contrast, the presented computations include a strain-hardening effect, which has a reducing effect on the maximum displacements. Further, Qui et al. base their analysis on the traveling wave (12) that overpredicts the pressure maximum exposed to the plate, cf. Fig. 13(a).

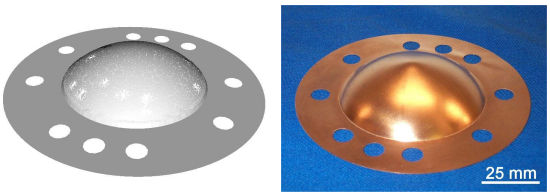


Figure 16: Comparison of simulated specimen with the experiment for $p_{\max} = 34\,\mathrm{MPa}$.

We present two exemplary results for plate rupture. When $p_{\max}$ is moderately increased, the plate first bulges severely but then undergoes localized cracking at the plate center. This failure mode is shown in Fig. 17(a) for $p_{\max} = 64\,\mathrm{MPa}$ that displays the situation at $t = 0.85\,\mathrm{ms}$ simulation time when the fracture pattern is clearly established and the water splashes into the air behind the plate. Note also the similarity in longitudinal cracking behavior in simulation and experiment. When the loading is increased further, rupture occurs primarily at the supports. We display only one case for the high loading $p_{\max} = 173\,\mathrm{MPa}$ that clearly exhibits tearing at the support from the initial impulse, cf. Fig. 17(b). Finally, it is worth mentioning that the reliable simulation of cracking phenomena with two coupled explicit solvers mandatorily requires a careful computational analysis of the stability condition in each sub-solver after every time step, which in our case also considers the temporal recursion in the SAMR approach. See Deiterding et al. (2006c) for details.

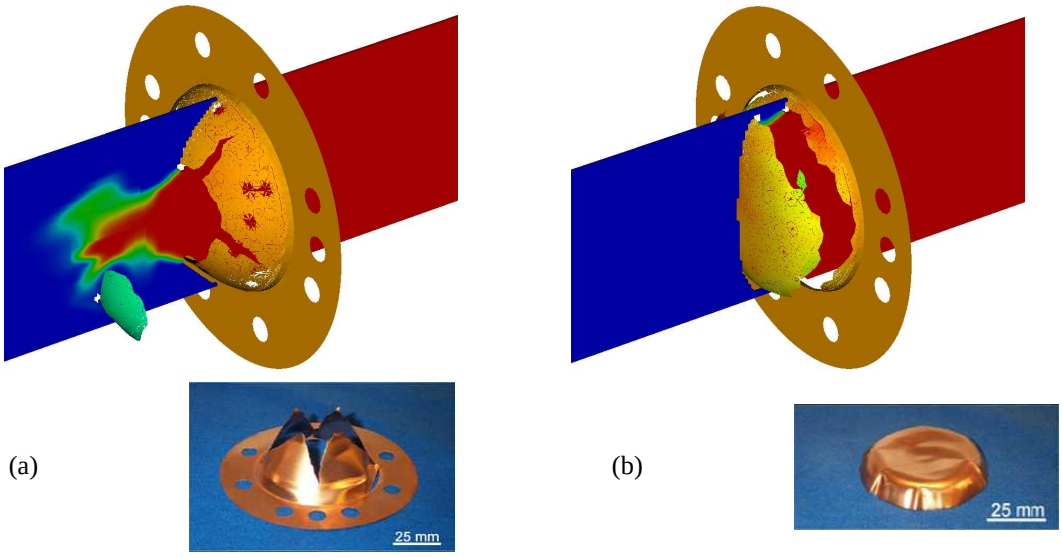


Figure 17: Plate fracture for $p_{\max} = 64\,\mathrm{MPa}$ at $t = 0.85\,\mathrm{ms}$ (a) and $p_{\max} = 173\,\mathrm{MPa}$ at $t = 0.177\,\mathrm{ms}$ (b), when the fracture patterns are fully established. The color midplane showing the volume fraction α^W in the fluid visualizes the water splash.

7 Conclusions

We have presented all computational components of a level-set-based fluid-structure coupling approach for the time-accurate simulation of thin flexible shells responding dynamically to strong pressure waves in water. The approach has been demonstrated to handle arbitrarily evolving thin-shells surrounded by fluid without problems. Three different fluid-structure interaction configurations of increasing complexity have been given to verify and validate the approach. The obtained results are found to be in good to excellent agreement with experimental observations, depending on the level of effort spent to reproduce the hydrodynamic loading conditions and to carefully adjust material model parameters that cannot be deduced unambiguously from tabulated data, namely for

viscoplasticity and fracture. The level of detail given should allow the easy reproducibility of our results thereby proving good verification tests for explicit FSI software. The integrated implementation of all presented components, including further documented explicit FSI testcases, is freely available from http://www.cacr.caltech.edu/asc.

Acknowledgements

We thank Vikram S. Deshpande from the Department of Engineering of the University of Cambridge for providing the plate photographs shown in the Figs. 16 and 17.

R. D. is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. All computations were performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

References

Abgrall, R.: How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *J. Comput. Phys.*, 125, (1996), 150–160.

Abgrall, R.; Karni, S.: Computations of compressible multifluids. *J. Comput. Phys.*, 169, (2001), 594–523.

Aivazis, M.; Goddard, W.; Meiron, D.; Ortiz, M.; Pool, J.; Shepherd, J.: A virtual test facility for simulating the dynamic response of materials. *Comput. Science & Engineering*, 2, 2, (2000), 42–53.

Arienti, M.; Hung, P.; Morano, E.; Shepherd, J. E.: A level set approach to Eulerian-Lagrangian coupling. *J. Comput. Phys.*, 185, (2003), 213–251.

Ashani, J. Z.; Ghamsari, A. K.: Theoretical and experimental analysis of plastic response of isotropic circular plates subjected to underwater explosion loading. *Mat.-wiss. u. Werkstofftechn.*, 39, 2, (2008), 171–175.

Berger, M.; Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82, (1988), 64–84.

Cirak, F.; Ortiz, M.: Fully c^1-conforming subdivision elements for finite deformation thin-shell analysis. *Int. J. Numer. Meth. Engineering*, 51, (2001), 813–833.

Cirak, F.; Ortiz, M.; Pandolfi, A.: A cohesive approach to thin-shell fracture and fragmentation. *Computer Methods in Appl. Mechanics and Engineering*, 194, (2005), 2604–2618.

Cirak, F.; Ortiz, M.; Schröder, P.: Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int. J. Numer. Meth. Engineering*, 47, (2000), 2039–2072.

Cirak, F.; Radovitzky, R.: A Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets. *Computers & Structures*, 83, (2005), 491–498.

Cuitino, A.; Ortiz, M.: A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics. *Engineering Computations*, 9, (1992), 437–451.

Davis, S. F.: Simplified second-order Godunov-type methods. *SIAM J. Sci. Stat. Comp.*, 9, (1988), 445–473.

deBorst, R.: The zero-normal stress condition in plane-stress shell elasto-plasticity. *Comm. Meth. Appl. Numer. Meth.*, 7, (1991), 29–33.

Deiterding, R.: AMROC - Blockstructured Adaptive Mesh Refinement in Object-oriented C++. Available at http://amroc.sourceforge.net (2002).

Deiterding, R.: An adaptive Cartesian detonation solver for fluid-structure interaction simulation on distributed memory computers. In: A. Deane, ed., *Parallel Computational Fluid Dynamics - Theory and Application, Proc. Parallel CFD 2005 Conference*, pages 333–340, Elsevier (2006).

Deiterding, R.; Cirak, F.; Mauch, S. P.; Meiron, D. I.: A virtual test facility for simulating detonation-induced fracture of thin flexible shells. In: V. N. Alexandrov; G. D. van Albada; P. M. Sloot; J. Dongarra, eds., *Proc. 6th Int. Conf. Computational Science, Reading, UK, May 28-31, 2006*, vol. 3992 of *Lecture Notes in Computer Science*, pages 122–130, Springer (2006a).

Deiterding, R.; Radovitzki, R.; Mauch, S.; Cirak, F.; Hill, D. J.; Pantano, C.; Cummings, J. C.; Meiron, D. I.: Virtual Test Facility: A virtual shock physics facility for simulating the dynamic response of materials. Available at http://www.cacr.caltech.edu/asc (2006b).

Deiterding, R.; Radovitzky, R.; Mauch, S. P.; Noels, L.; Cummings, J. C.; Meiron, D. I.: A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. *Engineering with Computers*, 22, 3-4, (2006c), 325–347.

Deshpande, V. S.; Heaver, A.; Fleck, N. A.: An underwater shock simulator. *Royal Society of London Proceedings Series A*, 462, 2067, (2006), 1021–1041.

Fedkiw, R. P.: Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.*, 175, (2002), 200–224.

Fedkiw, R. P.; Aslam, T.; Merriman, B.; Osher, S.: A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152, (1999), 457–492.

Giordano, J.; Jourdan, G.; Burtschell, Y.; Medale, M.; Zeitoun, D. E.; Houas, L.: Shock wave impacts on deforming panel, an application of fluid-structure interaction. *Shock Waves*, 14, 1-2, (2005), 103–110.

Jones, N.: *Structural Impact*. Cambridge University Press (1989).

LeVeque, R. J.: *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, New York (2002).

Löhner, R.; Baum, J.; Charman, C.; Pelessone, D.: Fluid-structure interaction simulations using parallel computers. vol. 2565 of *Lecture Notes in Computer Science*, pages 3–23, Springer (2003).

Mauch, S. P.: *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. Ph.D. thesis, California Institute of Technology (2003).

Mauch, S. P.; Meiron, D. I.; Radovitzky, R.; Samtaney, R.: Coupled Eulerian-Lagrangian simulations using a level set method. In: K. Bathe, ed., *2nd M.I.T. Conference on Computational Fluid and Solid Mechanics*, M.I.T, Cambridge, MA (2003).

Mittal, R.; Iaccarino, G.: Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37, (2005), 239–261.

Ortiz, M.; Pandolfi, A.: Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *Int. J. Numer. Meth. Engineering*, 44, (1999), 1267–1282.

Qui, X.; Deshpande, V. S.; Fleck, N. A.: Dynamic response of a clamped circular sandwich plate subject to shock loading. *Int. J. Numer. Meth. Engineering*, 71, (2004), 637–645.

Shyue, K.-M.: An efficient shock-capturing algorithm for compressible multicomponent problems. *J. Comput. Phys.*, 142, (1998), 208–242.

Shyue, K.-M.: A volume-fraction based algorithm for hybrid barotropic and non-barotropic two-fluid flow problems. *Shock Waves*, 15, (2006), 407–423.

Specht, U.: *Numerische Simulation mechanischer Wellen an Fluid-Festkörper-Mediengrenzen*. No. 398 in VDI Reihe 7, VDU Verlag, Düsseldorf (2000).

Toro, E. F.: *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, Heidelberg, 2nd edn. (1999).

Toro, E. F.; Spruce, M.; Speares, W.: Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4, (1994), 25–34.

Xie, W. F.; Liu, T. G.; Khoo, B. C.: Application of a one-fluid model for large scale homogeneous unsteady cavitation: The modified Schmidt model. *Computers & Fluids*, 35, (2006), 1177–1192.

Address: R. Deiterding (corresponding author), Oak Ridge National Laboratory, P.O. Box 2008, MS-6367, Oak Ridge, TN 37831, USA.
email: deiterdingr@ornl.gov.

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 81– 96
Herrsching am Ammersee, 29.9.-1.10.2008

Wind Induced Ovalling Oscillations of Thin-Walled Cylindrical Structures

D. Dooms, G. De Roeck, G. Degrande

Circular cylindrical shells, like silos, are perceptible to wind induced ovalling oscillations, an aeroelastic phenomenon, where the cross section deforms as a shell without bending deformation of the longitudinal axis of symmetry. During a storm in October 2002, ovalling was observed on several empty silos of a group of forty silos in the port of Antwerp. A coupled numerical analysis of the wind flow around one silo aims to predict the ovalling onset flow velocity.

A three-dimensional finite element model of the silo is sequentially coupled to the three-dimensional wind flow around a single cylinder with a conventional serial staggered algorithm and an iteratively staggered algorithm. In the iteratively staggered algorithm, the transferred displacements are not relaxed. The accuracy of the conventional serial staggered algorithm is lower than the accuracy of the iteratively staggered algorithm. The response of the silo is dominated by the eigenmodes with circumferential wavenumber $n = 3$ and 4.

In order to reduce the computational cost of the fluid-structure interaction computations, the model of the structure is reduced to two dimensions using the finite strip method. This finite strip model of the cylinder is coupled to a two dimensional model of the flow. The coupling procedure is validated by means of experimental results of wind tunnel tests reported in the literature (Laneville and Mazouzi, 1995). As ovalling occurs when the negative aerodynamic damping cancels the structural modal damping, the structural damping is reduced step-by-step in order to find the damping ratio at which ovalling occurs.

1 Introduction

Circular cylindrical shells are widely used in civil engineering structures as silos, chimneys and water towers. The use of high tensile strength aluminium has resulted in slender structures that are more perceptible to wind induced ovalling oscillations, an aeroelastic phenomenon, where the cross section deforms as a shell without bending deformation of the longitudinal axis of symmetry. Resonance occurs when the negative aerodynamic damping cancels the structural modal damping. During a storm in October 2002, ovalling was observed on several empty silos of a group consisting of forty silos in the port of Antwerp (figure 1). The forty silos are placed in five rows of eight silos with gaps of 30 cm between two neighbouring silos. Similar cases indicate that storm damage is mainly located on silos on the corners of the group.

A fluid-structure interaction analysis aims to predict the ovalling onset flow velocity and to investigate in a later stage the influence of the distance between the silos. The configuration of this group strongly modifies the pressure distribution around the silos (Dooms et al., 2006a).

The structural mode shapes and eigenfrequencies of a single silo are computed with a three-dimensional finite element model. The transient three-dimensional incompressible turbulent flow around a single cylinder at $\mathrm{Re} = 12.4 \times 10^6$ is computed using the Shear Stress Transport turbulence model (Menter, 1994). The three-dimensional finite element model of the silo is sequentially coupled to the three-dimensional wind flow around a single cylinder with two different algorithms. In order to reduce the computational cost of the fluid-structure interaction computations, the model of the structure is reduced to two dimensions using the finite strip method. This finite strip model of the cylinder is coupled to a two dimensional flow. The coupling procedure is validated by means of experimental results of wind tunnel tests reported in the literature (Laneville and Mazouzi, 1995).

Figure 1: Photograph of the silo group.

2 Silo structure

The silos are circular cylindrical shell structures with a diameter of $D = 5.5\,\mathrm{m}$ and a height of $25\,\mathrm{m}$. One cylinder consists of 10 aluminium sheets with a height of $2.5\,\mathrm{m}$ and a thickness that decreases with the height from $10.5\,\mathrm{mm}$ at the bottom to $6\,\mathrm{mm}$ at the top. The length-to-radius ratio $L/R = 9.1$ and the radius-to-thickness ratio ranges from $R/t = 262$ at the bottom to $R/t = 458$ at the top. At the top and the bottom, a cone is welded to the cylinder at an angle of $15°$ and $60°$ with the horizontal plane, respectively. The bottom of the cylinder is supported by an octagonal beam and is bolted to the beam at 4 points around the circumference. The silos are made of aluminium with a Young's modulus $E = 67600 \times 10^6\,\mathrm{N/m^2}$, a Poisson's ratio $\nu = 0.35$ and a density $\rho = 2700\,\mathrm{kg/m^3}$.

A finite element model of the silo has been validated by means of modal parameters derived from in situ measurements of radial accelerations under ambient loading (Dooms et al., 2006c). For the eight central aluminium sheets (with a height of $2.5\,\mathrm{m}$ each), sixty 8-node quadrilateral shell elements are used around the circumferential direction and sixteen elements are used along the vertical direction. In the zones near the lower and upper edges of the silo and on both cones, smaller finite elements are used as to take into account the local bending waves (Billington, 1965). The total number of shell elements is equal to 18988. The eigenfrequencies and mode shapes of a circular cylindrical shell structure are very sensitive to the boundary conditions imposed on the axial displacements u_z, while the influence of the boundary condition for the rotation φ_θ is almost negligible (Forsberg, 1964). At the bottom of the cylinder all degrees of freedom are constrained in 4 points around the circumference. Figure 2 shows a top and a three-dimensional view of the five mode shapes with the lowest eigenfrequencies. Each mode shape is referred to by a couple (m, n), where m denotes the half wave number in the axial direction ($m/2$ is the number of axial waves) and n is the number of circumferential waves. Two similar mode shapes with possibly slightly different eigenfrequencies are associated with every couple (m, n). The in situ measurements indicate that the eigenmodes with $(1, 3)$ or $(1, 4)$ have the highest contribution to the response of the silos under wind loading (Dooms et al., 2003).

In order to limit the computational cost for the fluid-structure interaction computations, a coarser finite element model without the local refinements near the boundaries is used. In this model, the cylindrical part of the silo is meshed with sixty shell elements in the circumferential direction and twenty in the axial direction. The total number of shell elements in this model is equal to 1756. Table 1 compares the eigenfrequencies of the coarser model with eigenfrequencies of the experimentally validated model. For the lowest eigenfrequencies around $4\,\mathrm{Hz}$, the differences between eigenfrequencies computed with the two models are smaller than 2%.

In order to advance the finite element solution of the structure in time, the Newmark method with $\alpha = 0.25$ and $\delta = 0.5$ is used. The time step Δt is equal to $0.005\,\mathrm{s}$, which is small enough to compute accurately the contributions of eigenmodes up to $10\,\mathrm{Hz}$. For the structure Rayleigh damping is added. The damping matrix $\mathbf{C}$ is constructed as a linear combination of the mass and the stiffness matrix:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{1}$$

The modal damping ratios ξ_k at two different frequencies ω_k determine the multipliers α and β for respectively

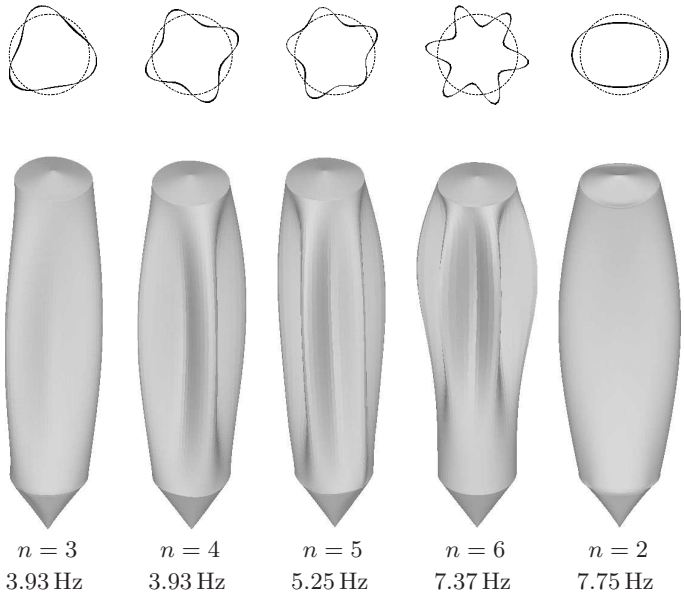


Figure 2: Top and three-dimensional view of the five mode shapes of the silo with the lowest eigenfrequencies.

(m, n)	Coarser mesh		Validated mesh	
(1,2)	7.90	8.80	7.75	8.48
(1,3)	4.00	4.00	3.93	3.93
(1,4)	3.93	4.05	3.93	4.04
(1,5)	5.37	5.37	5.25	5.25
(1,6)	7.37	7.37	7.37	7.37
(1,7)	9.72	9.72	9.72	9.72
(2,4)	8.71	8.97	8.71	8.94
(2,5)	5.93	5.93	5.56	5.56
(2,5)	8.08	8.08	8.01	8.01
(2,6)	9.29	9.49	9.29	9.39

Table 1: Eigenfrequencies (in Hz) computed with the coarser and the validated three-dimensional finite element model.

the mass and the stiffness matrix by means of the following system of equations:

$$2\omega_k \xi_k = \alpha + \beta \omega_k^2 \tag{2}$$

A modal damping ratio $\xi_1 = 0.25\%$ at $f_1 = 3.93\,\mathrm{Hz}$ and $\xi_2 = 0.50\%$ at $f_2 = 20\,\mathrm{Hz}$ corresponds to $\alpha = 0.078\,\mathrm{s}^{-1}$ and $\beta = 0.75 \cdot 10^{-4}\,\mathrm{s}$. The modal damping ratios are estimated from the measured modal damping ratios for all eigenmodes between $3.93\,\mathrm{Hz}$ and $20\,\mathrm{Hz}$ (Dooms et al., 2003).

3 Turbulent air flow around a single silo

First, the wind flow around a silo is simplified to a two-dimensional unsteady flow around a cylinder. In all computations air with a density $\rho = 1.25\,\mathrm{kg/m^3}$ and a dynamic viscosity $\mu = 1.76 \times 10^{-5}\,\mathrm{Pa\,s}$ is used. The mean wind velocity at a height of $z = 30\,\mathrm{m}$ for terrain category II (BIN, 1995) is equal to $v_m = 31.84\,\mathrm{m/s}$. At the inlet, a turbulence intensity $I = \sqrt{2k/3}/v_m = 1.00\%$ is imposed.

As the Reynolds number $\mathrm{Re} = Dv_m/\nu = 12.4 \times 10^6$ is larger than $\mathrm{Re} = 3.5 - 6 \times 10^6$, the regime of the flow around the cylinder is post-critical (Zdravkovich, 1997). The wake and the shear layers are fully turbulent and the boundary layers become fully turbulent prior to separation. In the boundary layers, the transition from laminar to turbulent flow takes place between the stagnation and the separation point. In the post-critical regime, regular vortex shedding reappears, while it was absent at lower Reynolds numbers.

The pressure coefficient is a dimensionless expression for the pressure at the cylinder's surface:

$$C_\mathrm{p} = \frac{p - p_\mathrm{f}}{\frac{\rho v_\mathrm{f}^2}{2}} \tag{3}$$

where p_f and v_f are the free stream pressure and velocity, respectively. Eurocode 1 (BIN, 1995) describes the pressure coefficient as a function of the angle for $\mathrm{Re} = 10^7$. At the stagnation point, the pressure coefficient is equal to 1. The minimum value of the pressure coefficient $C_\mathrm{p}^\mathrm{min}$ equals -1.5 at an angle of $75°$. The boundary layer separates at $105°$. After separation, the base pressure coefficient C_p^b is constant and equal to -0.8. Zdravkovich (1997) gives an overview of available experimental data. There is a lack of detailed experimental data at post-critical Reynolds numbers, while available data show considerable scatter, which may be explained by the high sensitivity of the flow to perturbations due to surface roughness and free-stream turbulence.

A two-dimensional URANS simulation is performed with the CFX finite volume code (Ansys, 2005a). The High Resolution spatial discretization is used, which is an automatically determined blend of a first and a second order accurate scheme. The coupled algorithm solves the momentum and continuity equation as a single system.

The boundaries of the computational domain should be sufficiently far from the region close to the silo where the accuracy of solution is important. Behr et al. (1995, 1991) suggest a distance of at least $8D$ for the inlet and the lateral boundaries and a distance of $22.5D$ for the outlet, with D the diameter of the silo. Here, a distance of $9D$ is adopted for the inlet and the lateral boundaries and $30D$ for the outlet. Results on a larger problem domain, where the inlet and the lateral boundaries are located at $12D$ and the outlet at $40D$, are comparable.

Far away from the silo wall, an unstructured mesh, consisting of triangles, is used. Close to the silo wall and in the wake of the silo, the mesh is structured and consists of quadrilaterals. The nodes next to the silo wall are placed in the logarithmic law region, where a logarithmic relation exists between the dimensionless wall distance $y^+ = v_\tau y/\nu$ and the dimensionless velocity $v^+ = v/v_\tau$, where $v_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity, y the wall distance and τ_w the wall shear stress:

$$v^+ = \frac{1}{\kappa} \ln y^+ + B \tag{4}$$

The von Karman constant κ is equal to 0.41 and B is equal to 5.2. In the outer layer ($y^+ > 50$), direct effects of the viscosity on the main flow are negligible. Nevertheless, the region of validity of the logarithmic law can be extended to $y^+ > 30$. The mesh consists of 53530 elements and 90798 nodes. At time $4.68\,\mathrm{s}$ the dimensionless wall distance y^+ of the nodes next to the silo wall varies from 0 at the stagnation points to 220 (figure 3). In practice, so-called wall functions apply the logarithmic law as boundary conditions on the nodes next to the silo wall.

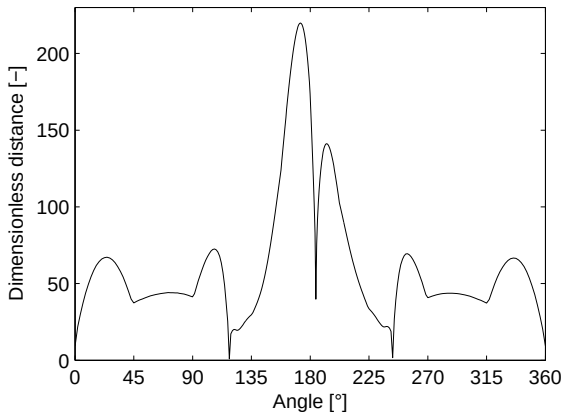


Figure 3: Dimensionless distance y^+ of the nodes next to the silo wall as a function of the angle θ for a single silo.

The Shear Stress Transport (SST) (Menter, 1994) turbulence model combines through a blending function the robust and accurate formulation of the k-ω model in the near wall region with the free stream independence of the k-ε model in the outer part of the boundary layer. It is used as it is suited to predict the onset and amount of separation under adverse pressure gradients and produces of all turbulence models in CFX the best overall

correspondence with a set of experimental results from the literature (Zdravkovich, 1997) for the flow around a cylinder (Dooms et al., 2006b).

As in the post-critical regime regular vortex shedding is present, an unsteady RANS computation (Iaccarino et al., 2003) is performed. The transient solution is integrated by the three-point backward difference scheme with a time step $\Delta t = 0.005\,\mathrm{s}$, which corresponds to approximately 100 time steps per vortex shedding period. Within every time step, 5 iterations are performed to obtain a converged solution. The computed time window corresponds to ten vortex shedding periods. 937 time steps are computed, which results in a time window of $4.68\,\mathrm{s}$. The vortex shedding frequency $f_{\mathrm{vs}} = 2.13\,\mathrm{Hz}$ is described by the dimensionless Strouhal number:

$$\mathrm{St} = \frac{f_{\mathrm{vs}} D}{v_{\mathrm{f}}} = 0.37 \tag{5}$$

For flows with $\mathrm{Re} \approx 10^7$, experimental values for the Strouhal number (Zdravkovich, 1997) range from 0.27 to 0.32. Eurocode 1 suggests a constant value of 0.2, independent of the Reynolds number. If the streamlines would separate a little bit earlier from the cylinder wall, the Strouhal number would be reduced. Placing the lateral boundaries further away might decrease the Strouhal number as well.

Figure 4 shows the time history and the frequency content of the pressure in a point at the silo's surface. The vortex shedding frequency and some higher harmonics are clearly visible in the frequency content.

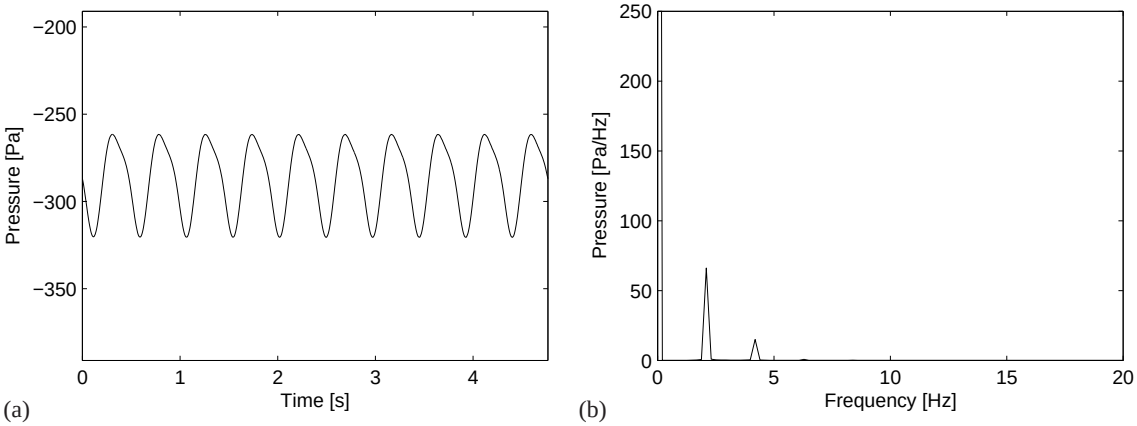


Figure 4: (a) Time history and (b) frequency content of the pressure in a point at the silo's surface for a single silo.

Figure 5 shows the time average and the standard deviation of the pressure p. The stagnation pressure at the windward side and the suction in the wake are clearly visible. The largest time variations of the pressure occur in the wake.

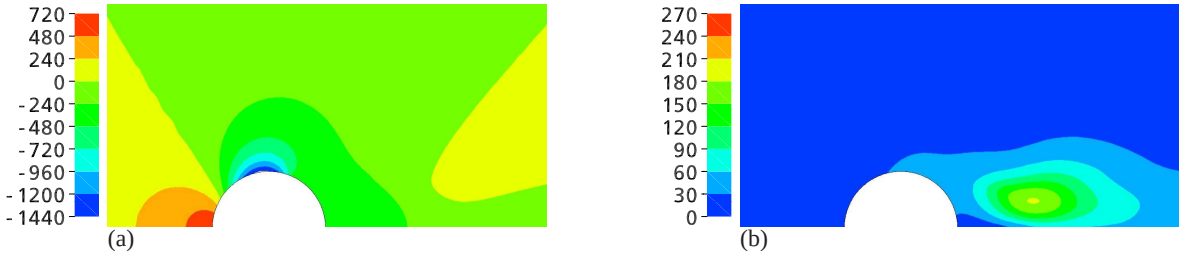


Figure 5: (a) Time average and (b) standard deviation of the pressure p for a single silo.

Figure 6 compares the time-averaged pressure coefficient $\overline{C_{\mathrm{p}}}$ of the transient computation with experimental data from the literature. The range of the available experimental data for the pressure coefficients at Reynolds numbers from 0.73×10^7 to 3.65×10^7 is plotted in light grey at the background. The time average of the transient computation predicts a too low minimum pressure coefficient $\overline{C_{\mathrm{p}}}^{\mathrm{min}}$, while the base pressure coefficient $\overline{C_{\mathrm{p}}}^{\mathrm{b}}$ is quite high. The maxima and minima of the pressure coefficient during the transient computation are depicted as well.

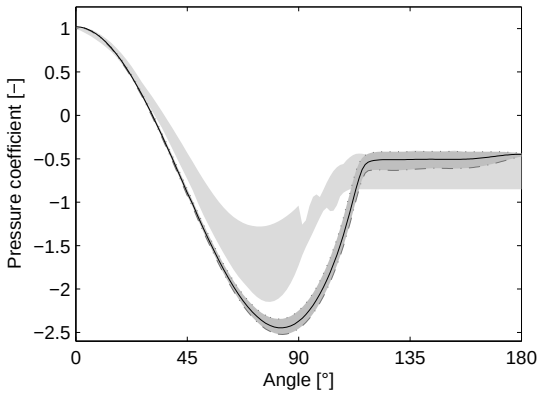


Figure 6: Comparison of the time average (solid line), minimum (dash-dotted line) and maximum (dotted line) of the pressure coefficient with experimental data for a single silo.

The drag coefficient $\overline{C_\mathrm{d}} = \int_0^{2\pi} \overline{C_\mathrm{p}} \cos\alpha d\alpha$ is, according to Eurocode 1, equal to 0.72 for a smooth surface ($k/b = 10^{-5}$). Drag coefficients derived from the experimental pressure coefficients vary from 0.45 to 0.73. The computations predict 0.35, which is an underestimation.

Next, in order to couple the flow to the three-dimensional shell model of the structure, the three-dimensional wind flow around a single cylinder is modelled with symmetry boundary conditions on top and bottom surfaces. The velocity profile at the inlet is uniform along the height. The three-dimensional mesh is obtained by copying the mesh used for the two-dimensional flow around a single silo 12 times in the axial direction. The number of elements in the axial direction is sufficient to compute the flow around a silo which deforms according to an eigenmode with $m = 1$. This number is however too low to generate any variation in the vortex shedding pattern along the axial direction. The total number of elements is equal to 642360.

In order to obtain an initial solution for the fluid-structure computation, first 1080 time steps are computed just for the fluid field, which results in a time window of $5.4\,\mathrm{s}$. Figure 7 shows the time history and the frequency content of the pressure between $t = 3.55\,\mathrm{s}$ and $t = 5.4\,\mathrm{s}$ at the cylinder's surface at mid-height for $\theta = 112°$, $\theta = 174°$ and $\theta = 180°$. The vortex shedding frequency at $2.16\,\mathrm{Hz}$ and a higher harmonic at $4.31\,\mathrm{Hz}$ are clearly visible in the frequency content. This vortex shedding frequency is slightly higher than the frequency obtained with a two-dimensional computation ($2.13\,\mathrm{Hz}$).

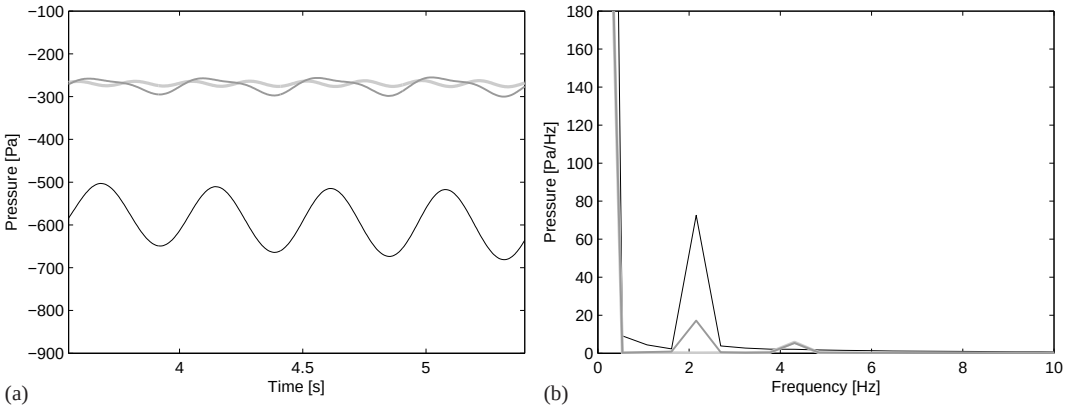


Figure 7: (a) Time history and (b) frequency content of the pressure at the cylinder's surface at mid-height for $\theta = 112°$ (—), $\theta = 174°$ (—) and $\theta = 180°$(—). The angle $\theta = 0°$ coincides with the stagnation point.

4 Fluid-structure interaction

In this section the wind induced ovalling oscillations of the silos located in the port of Antwerp, are studied for one silo using fluid-structure interaction. The shell model of the structure is coupled with the three-dimensional incompressible turbulent wind flow around a cylinder (figure 8).

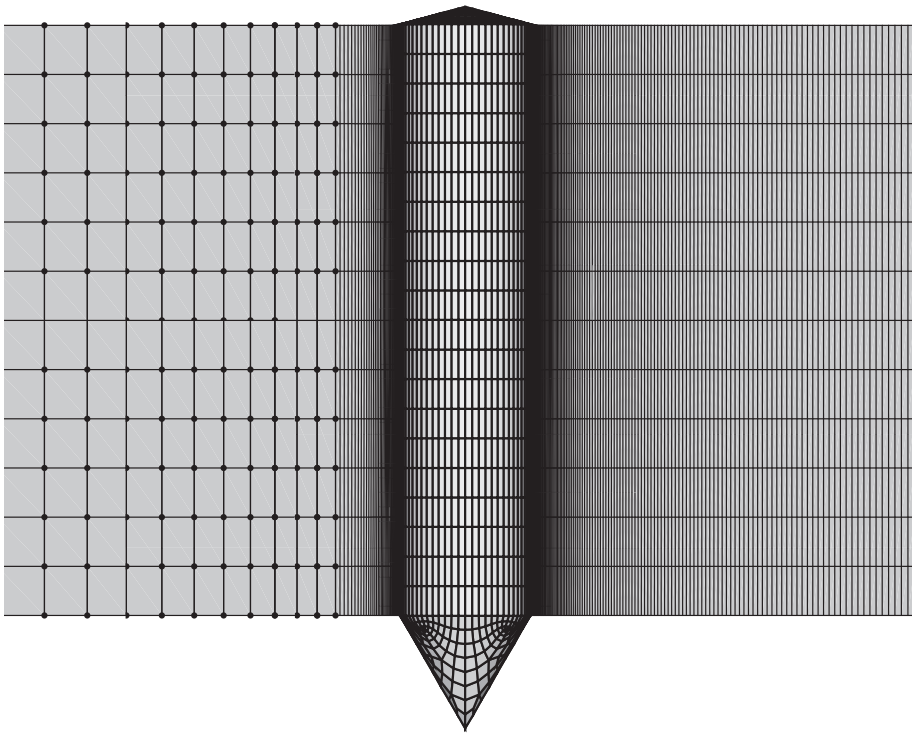


Figure 8: Model for the coupled simulation of the three-dimensional wind flow around a cylinder and the response of the silo structure.

Between the structure and a cylindrical surface with a diameter equal to twice the silo diameter, the fluid flow is computed on a deforming mesh. The Navier-Stokes equations for incompressible flow are written in an Arbitrary Lagrangian Eulerian formulation (Hughes et al., 1981):

$$\nabla_{\mathbf{x}} \cdot \mathbf{v} = 0 \tag{6}$$

$$\left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\boldsymbol{\chi}} + (\mathbf{c} \cdot \nabla_{\mathbf{x}})\mathbf{v} + \nabla_{\mathbf{x}} p^{\star} = 2\nu \nabla_{\mathbf{x}} \cdot \boldsymbol{\epsilon} + \mathbf{b} \tag{7}$$

where $\boldsymbol{\chi}$ and $\mathbf{x}$ denote the referential domain and the spatial domain respectively. The convective velocity $\mathbf{c}$ is defined as the difference between the material particle velocity $\mathbf{v}$ and the grid point velocity $\widehat{\mathbf{v}}$. On the outer boundaries of the fluid domain, the grid point velocity $\widehat{\mathbf{v}}$ should remain zero, while on the fluid-structure interface the grid point velocity $\widehat{\mathbf{v}}$ should be equal to the velocity of the structure.

The grid point displacements $\widehat{\mathbf{u}}$ of the fluid mesh are obtained by diffusing the displacements of the structure through this domain:

$$\nabla \cdot (k \nabla \widehat{\mathbf{u}}) = 0 \tag{8}$$

As to preserve the quality of the mesh in refined regions, the diffusivity of a finite volume is equal to the inverse of its volume.

For the load transfer between the non-matching grids, the traction is first integrated over the surface of the fluid independently of the structure. The resulting nodal force is transferred to the structure using a point to element mapping: each node of the fluid mesh is mapped onto one element of the mesh of the structure. The transferred force is distributed over the nodes of this element according to the value of the shape functions of the structure. As the sum of the shape functions is equal to one the total force is conserved. For the displacement transfer, the consistent interpolation method is used. The consistent interpolation method is based on the same point to element mapping: each node of the fluid mesh is mapped onto one element of the mesh of the structure. The value of the displacement at the projection of the fluid node onto the structural element is interpolated using the shape functions of the structure mesh.

As initial conditions the undeformed structure and the transient solution of the fluid flow without interaction at $t = 5.4\,\mathrm{s}$ are used. 500 time steps are computed, which results in a time window of $2.5\,\mathrm{s}$.

First the conventional serial staggered algorithm (algorithm A) is applied, in which every field is computed once at each time level t^n. In a second computation an iteratively staggered algorithm (algorithm B) ensures the equilibrium between the two fields at each time level t^n. A relaxation parameter $\omega^{n+1(i)}$ can be introduced for the interface displacements:

$$\mathbf{u}_\Gamma^{n+1(i+1)} = \omega^{n+1(i)} \widetilde{\mathbf{u}}_\Gamma^{n+1(i+1)} + (1 - \omega^{n+1(i)}) \mathbf{u}_\Gamma^{n+1(i)} \tag{9}$$

$$= \mathbf{u}_\Gamma^{n+1(i)} + \omega^{n+1(i)} (\widetilde{\mathbf{u}}_\Gamma^{n+1(i+1)} - \mathbf{u}_\Gamma^{n+1(i)}) \tag{10}$$

The interface displacements $\mathbf{u}_\Gamma^{n+1(i+1)}$ that will be transferred in the next iteration to the fluid, are a linear combination of the newly computed interface displacements $\widetilde{\mathbf{u}}_\Gamma^{n+1(i+1)}$ in the structure and the interface displacements $\mathbf{u}_\Gamma^{n+1(i)}$ transferred to the fluid at the beginning of the iteration. In this case the interface displacements are not relaxed.

The convergence of the interfield iterations is reached if:

$$\frac{\|\widetilde{\boldsymbol{\phi}}_\Gamma^{n+1(i+1)} - \boldsymbol{\phi}_\Gamma^{n+1(i)}\|_{\mathrm{L}_2}}{\|\widetilde{\boldsymbol{\phi}}_\Gamma^{n+1(i+1)}\|_{\mathrm{L}_2}} < \epsilon_{\boldsymbol{\phi}} \tag{11}$$

where $\boldsymbol{\phi}_\Gamma$ are the interface displacements or forces. Maximum four interfield iterations are needed to obtain a relative change of the transferred quantities smaller than $\epsilon_{\boldsymbol{\phi}} = 0.001$ (equation (11)).

Figure 9a compares the time history of the radial displacements in three points at mid-height of the silo computed with algorithm A and B. Within this short time window both computations are stable. For a more rigourous evaluation of the stability more time steps should be computed. Clearly, the results computed with algorithm A show differences that increase in time with the results of algorithm B because the accuracy of the conventional serial staggered algorithm is lower than the accuracy of the iteratively staggered algorithm. The accuracy of the conventional serial staggered algorithm could be improved by the use of a prediction for the structural displacements and a corrected fluid force, but this option is not available in the coupling between Ansys and CFX. As the staggered coupling algorithm is stable for this example, the time step could be as well reduced in order to improve the accuracy. This might be cheaper than the use of interfield iterations.

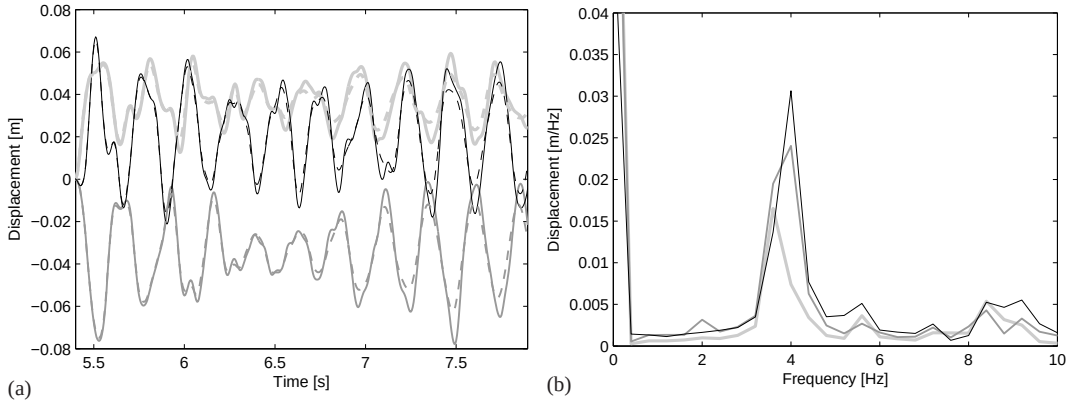
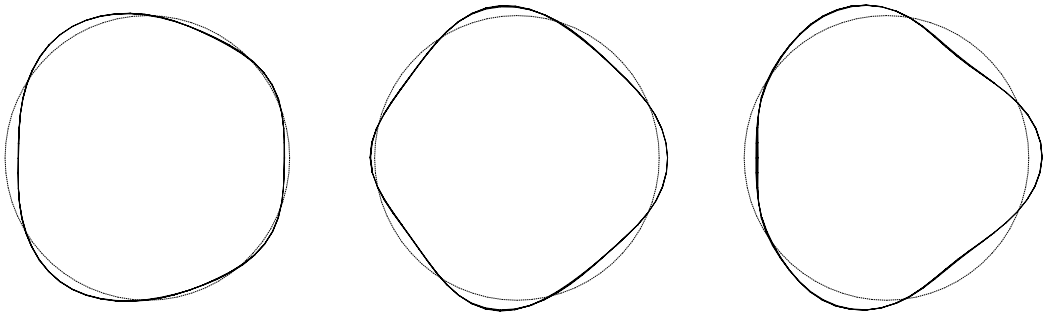


Figure 9: (a) Time history of the radial displacements at mid-height for $\theta = 66°$ (—), $\theta = 120°$ (—) and $\theta = 180°$ (—) computed with algorithm A (dashed lines) and algorithm B (solid lines) and (b) frequency content of these radial displacements computed with algorithm B. The angle $\theta = 0°$ coincides with the stagnation point.

Figure 9b shows the frequency content of the radial displacements in the same points computed with algorithm B. The response of the silo is dominated by eigenmodes (1,3) and (1,4) around $4\,\mathrm{Hz}$. The peak around $2\,\mathrm{Hz}$ indicates the effects of vortex shedding on the silo structure. The smaller peaks above $4\,\mathrm{Hz}$ are related to the eigenmodes with higher frequencies.

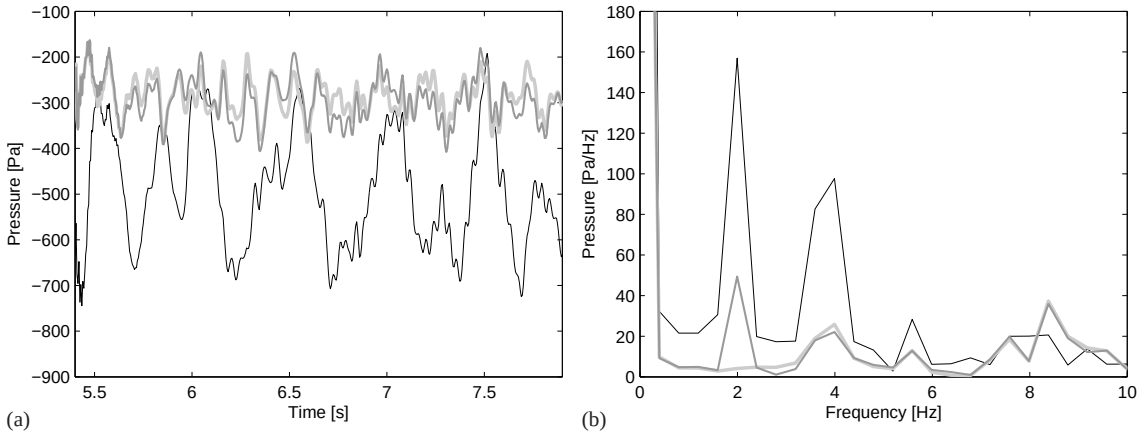
Figure 10 shows the deformations (enlarged with a factor 5) of the structure between a height of $11.25\,\mathrm{m}$ and $13.75\,\mathrm{m}$ at three different times. At all times a antinode faces the free stream direction. At $t = 5.905\,\mathrm{s}$ and $t = 6.805\,\mathrm{s}$ the response is dominated by eigenmodes (1,4) with respectively a negative and a positive radial displacement at $\theta = 0°$. At $t = 7.250\,\mathrm{s}$ eigenmode (1,3) is dominant. The maximal radial displacement is $0.105\,\mathrm{m}$ at $t = 6.415\,\mathrm{s}$ and occurs at a height of $15\,\mathrm{m}$ and $\theta = 0°$.



(a) (b) (c)

Figure 10: Deformations (enlarged with a factor 5) of the structure between $11.25\,\mathrm{m}$ and $13.75\,\mathrm{m}$ high at (a) $t = 5.905\,\mathrm{s}$, (b) $t = 6.805\,\mathrm{s}$ and (c) $t = 7.250\,\mathrm{s}$. The wind flows from the left.

Figure 11 shows the time history and frequency content of the pressure between $t = 5.4\,\mathrm{s}$ and $t = 7.9\,\mathrm{s}$ at the cylinder's surface at mid-height for $\theta = 112°$, $\theta = 174°$ and $\theta = 180°$. The comparison with figure 7 indicates that the structural deformations influence the pressure field near the wall. While the pressure time histories of the flow simulation around a rigid silo mainly showed contributions at 2.16 and $4.31\,\mathrm{Hz}$, in the coupled computation contributions are present as well at higher frequencies. An important contribution is present around $4\,\mathrm{Hz}$ which corresponds to the eigenfrequencies of eigenmodes (1,3) and (1,4). Due to the interaction the magnitude of the pressure fluctuations around $2\,\mathrm{Hz}$ has clearly increased, which indicates an amplification of the vortex shedding.



(a) (b)

Figure 11: (a) Time history and (b) frequency content of the pressure at the cylinder's surface at mid-height for $\theta = 112°$ (—), $\theta = 174°$ (—) and $\theta = 180°$(—) using algorithm B. The angle $\theta = 0°$ coincides with the stagnation point.

The pressure field on the vertical plane through the cylinder axis parallel with the inlet flow direction at $t = 7.9\,\mathrm{s}$ is shown in figure 12. The pressure field behind the cylinder is clearly three-dimensional. Figure 13 shows the pressure at the cylinder's surface along the height at five time levels for three circumferential angles. At the beginning of the simulation (at $t = 5.4\,\mathrm{s}$), the pressure is constant along the height in the stagnation point ($\theta = 0°$) and for $\theta = 180°$. At $\theta = 112°$ the pressure varies slightly with the height. During the coupled simulation the largest variations along the height occur at $\theta = 112°$, but also at $\theta = 0°$ and $\theta = 180°$ considerable variations take place.

At every time level the radial displacements along the circumference at mid-height are decomposed into a Fourier series of modes with circumferential wavenumbers n. For each mode principle component analysis (Pearson, 1901) of the time series yields the position of the first and the second principle component with respect to the silo. For all circumferential wavenumbers n the first principle component is positioned roughly with an antinode facing the flow and the second with a node facing the flow. Figure 14 shows the time history and frequency content of the first and the second principle component corresponding to circumferential wavenumbers $n = 2$, 3 or 4. The response of the silo mainly consists of modes with circumferential wavenumber $n = 3$ and 4. Their contribution varies strongly with time.

Figure 12: Pressure field on the vertical plane through the cylinder axis parallel with the inlet flow direction at $t = 7.9\,\mathrm{s}$.

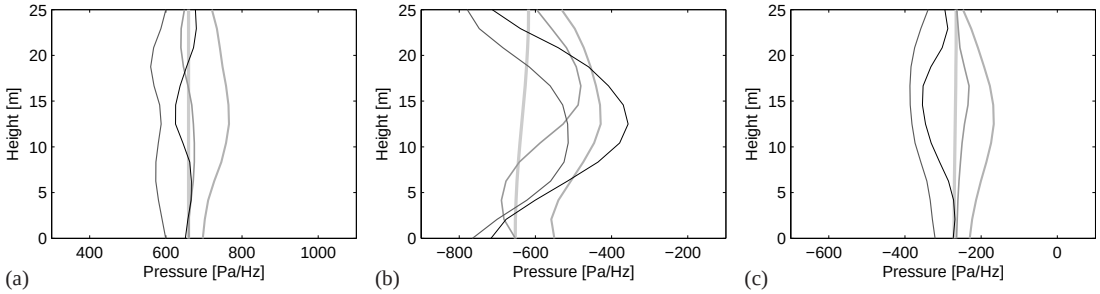


Figure 13: Pressure at the cylinder's surface along the height for (a) $\theta = 0°$, (b) $\theta = 112°$ and (c) $\theta = 180°$ at $t = 5.4\,\mathrm{s}$ (—), $t = 5.475\,\mathrm{s}$ (—) and $t = 5.925\,\mathrm{s}$ (—) $t = 6.35\,\mathrm{s}$ (—) $t = 6.605\,\mathrm{s}$ (—).

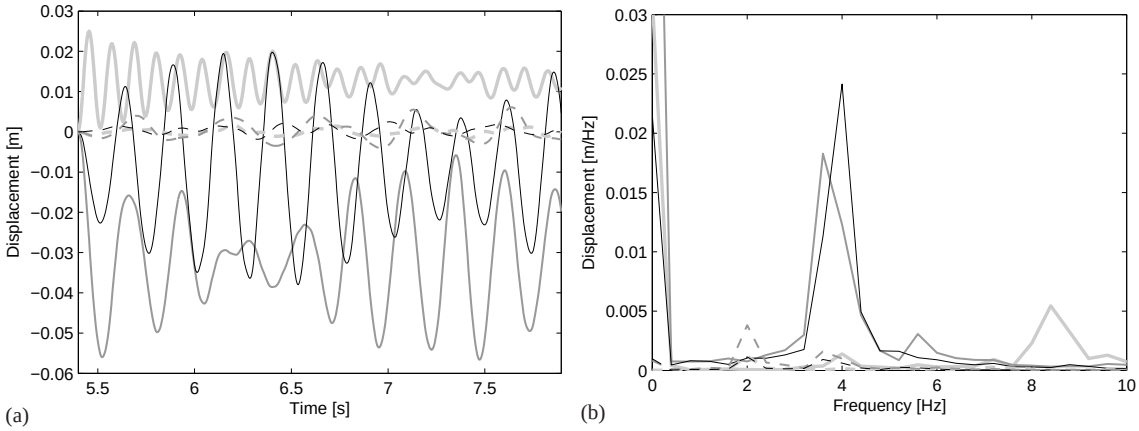


Figure 14: (a) Time history and (b) frequency content of the first (solid line) and second principal component (dashed line) of the displacements at mid-height with circumferential wavenumber $n = 2$ (—), $n = 3$ (—) and $n = 4$ (—).

In order to evaluate the occurrence of ovalling oscillations, the response of the structure should be computed during a much longer time interval (e.g. $40\,\mathrm{s}$), as the structure is suddenly loaded and the modal damping ratios are very low. However, the computation times using a single processor are very high. The major part is spent on the fluid partition.

5 The finite strip method

A reduction of the dimensions of the problem decreases the computational cost substantially. The finite strip method enables to build an approximate model of the structure in two dimensions. This finite strip model of the silo is coupled to a two dimensional flow.

Using a finite strip formulation (Cheung, 1976), the displacements of the three-dimensional structure are decomposed into a series of orthogonal functions that satisfy a priori the Dirichlet boundary conditions in the axial z-direction and a two-dimensional displacement field in the (r, θ)-plane. The use of orthogonal functions results in a decoupled system of equations for every term in the series. The sine functions reflect that the radial and circumferential displacements are assumed to be zero at both ends of the cylinder, while the cosine function allows for free axial displacements at both ends:

$$\begin{bmatrix} u_r(r,\theta,z,t) \\ u_\theta(r,\theta,z,t) \\ u_z(r,\theta,z,t) \end{bmatrix} = \sum_{m=0}^{\infty} \begin{bmatrix} \sin(\frac{m\pi z}{h}) & 0 & 0 \\ 0 & \sin(\frac{m\pi z}{h}) & 0 \\ 0 & 0 & \cos(\frac{m\pi z}{h}) \end{bmatrix} \begin{bmatrix} u_{rm}(r,\theta,t) \\ u_{\theta m}(r,\theta,t) \\ u_{zm}(r,\theta,t) \end{bmatrix} \tag{12}$$

The coupling procedure is validated by means of experimental results of wind tunnel tests reported in the literature. Laneville and Mazouzi (1995) studied the ovalling phenomenon of five different cylindrical shells in a wind tunnel. The cylinders were clamped at their base to a turntable over a height of 10 cm and did not have an end plate at the top. The flexible part of the shell was located outside the boundary layer of the wind tunnel. At the top of the cylinder, an adjustable false ceiling was fixed within a distance of 0.02 times the diameter of the cylinder in order to reduce the three-dimensional effects in the flow. The cylinders were made of aluminium with a Young's modulus $E = 68960\,\mathrm{MPa}$, a density $\rho = 2643\,\mathrm{kg/m^3}$ and a Poisson's ratio $\nu = 0.3$. Cylinder L1, with a height of $91\,\mathrm{cm}$, a radius of $6\,\mathrm{cm}$ and a shell thickness of $0.0127\,\mathrm{cm}$, started vibrating in mode (1,3) at a wind speed of $5\,\mathrm{m/s}$ (figure 15).

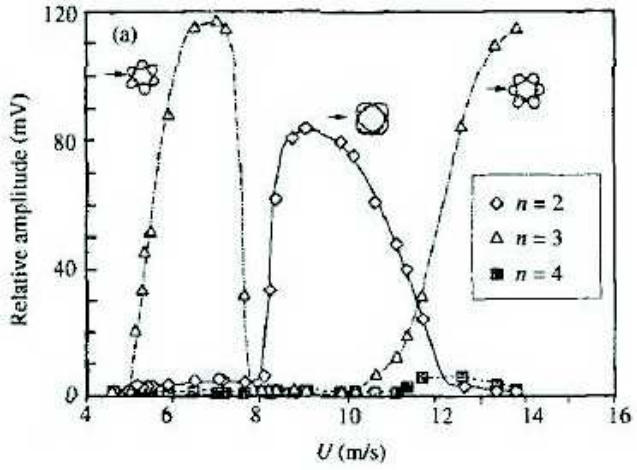


Figure 15: Mode occurrence and vibration amplitude as a function of the flow velocity (Laneville and Mazouzi, 1995).

The finite strip formulation of equation (12) is justified for structures which are simply supported at both ends. Here, the cylinder is clamped at its base and open at the top. If, however, a quarter (co)sine is assumed for the variation of the displacements in the axial direction, the eigenmodes can be approximated by doubling the height of the cylinder in the finite strip model: finite strip model 1 has a height of $2 \times (0.91 - 0.10) = 1.62\,\mathrm{m}$. The series in equation (12) can be limited to the term $m = 1$. The resulting 2-node elements have four degrees of freedom per node: the displacements in the radial, circumferential and axial directions and the rotation around the z-axis.

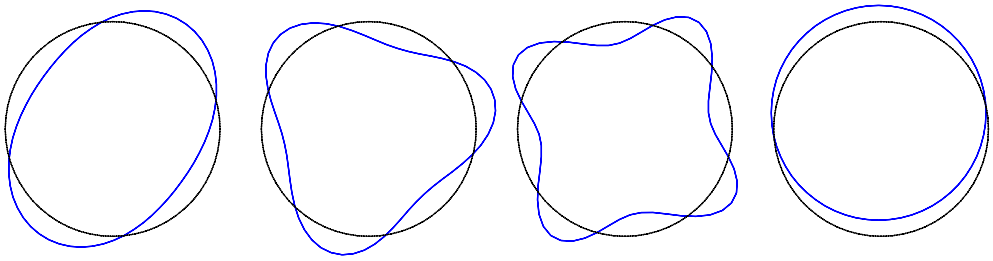
As CFX (Ansys, 2005a) inherently is a three-dimensional fluid solver, the fluid mesh is one element thick in the third dimension and symmetry is applied to both planes. As to enable coupling with this fluid model, a finite strip model with one element in the third dimension is obtained by combining a dummy four node Kirchhoff plate element with two user defined matrix elements (Ansys, 2005b). The finite strip elements are implemented through the user-defined elements. The stiffness matrix of the user defined elements is independent of the deformation, so that geometrical non-linear effects are not included in these terms. The dummy plate elements transfer the fluid loads to the two user defined matrix elements. They are $1.62\,\mathrm{m}$ high and have a Young's modulus and a density which is a factor 10^{-6} smaller than the values of aluminium.

The lowest eigenfrequency computed with model 1 corresponds well to the measured value (table 2). In order to achieve a better correspondence for the other eigenfrequencies, in a second model, the height is changed to $1.55\,\mathrm{m}$,

the Young's modulus to $65000\,\mathrm{MPa}$ and the density to $2875\,\mathrm{kg/m^3}$. Figure 16 shows the four mode shapes with the lowest eigenfrequencies of model 2.

	Measured	Model 1	Model 2
n	f [Hz]	f [Hz]	f [Hz]
2	47.0	47.1	46.9
3	64.5	68.7	64.5
4	117.5	127	118
1		127	129

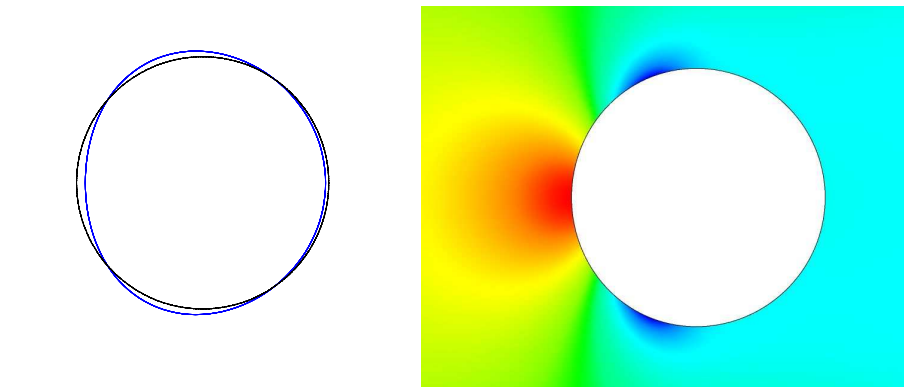
Table 2: Comparison of the eigenfrequencies for $m = 1$ of the measurements and different finite strip models.



a. $(1,2)$ $46.9\,\mathrm{Hz}$ b. $(1,3)$ $64.5\,\mathrm{Hz}$ c. $(1,4)$ $118\,\mathrm{Hz}$ d. $(1,1)$ $129\,\mathrm{Hz}$

Figure 16: Eigenmodes of cylinder L1 computed with a finite strip model.

First a steady-state FSI computation is performed. At the inlet a wind velocity of $7\,\mathrm{m/s}$ is chosen. At this wind speed the vibration amplitude for mode $(1,3)$ was maximal (figure 15). Figure 17a shows the deformations of the cylinder enlarged with a factor five. The largest displacement is $0.84\,\mathrm{mm}$. The pressure distribution around the deformed cylinder is shown in figure 17b.



(a) (b)

Figure 17: (a) Deformed structure (enlarged with a factor five) and (b) pressure distribution around the cylinder for steady-state FSI computation.

The results of the steady FSI computation are used as initial conditions for a transient FSI computation. For both fluid and structure the time step is chosen as $\Delta t = 0.0002\,\mathrm{s}$. This is small enough to compute accurately the contributions of eigenmodes up to $250\,\mathrm{Hz}$ (figure 16) and to be able to calculate the vortex shedding in the wake of the cylinder. 1000 time steps are computed, which results in a time window of $0.2\,\mathrm{s}$. The iteratively staggered algorithm without relaxation is employed. Maximum ten interfield iterations are performed in order to obtain a relative change of the transferred quantities smaller than $\epsilon_\phi = 0.001$.

As ovalling occurs when the negative aerodynamic damping cancels the structural modal damping, the structural damping is reduced step-by-step in order to find the damping ratio at which ovalling occurs. The damping ratio

is supposed to be equal for the two lowest eigenmodes, at $f_1 = 46.9\,\text{Hz}$ and at $f_2 = 64.5\,\text{Hz}$. The value varies from 0.0596% over 0.0238%, 0.0149%, 0.0134% to 0%. Figure 18 shows the radial displacements at $\theta = 180°$ and $\theta = 90°$. For the highest damping ratio 0.0596%, the response of the structure is damped out and shows a stable behaviour. For the four lower damping ratios, the response of the structure becomes unstable during the computations. The lower the structural damping is, the sooner the computations become unstable. From a certain time step, the convergence criterium for the forces (11) is not satisfied anymore after ten interfield iterations. For the three lowest damping ratios, in subsequent time steps the convergence criterium for the displacements (11) and the convergence criterium for the flow computations are not satisfied anymore and the computations finally crash due to excessive deformations of the structure.

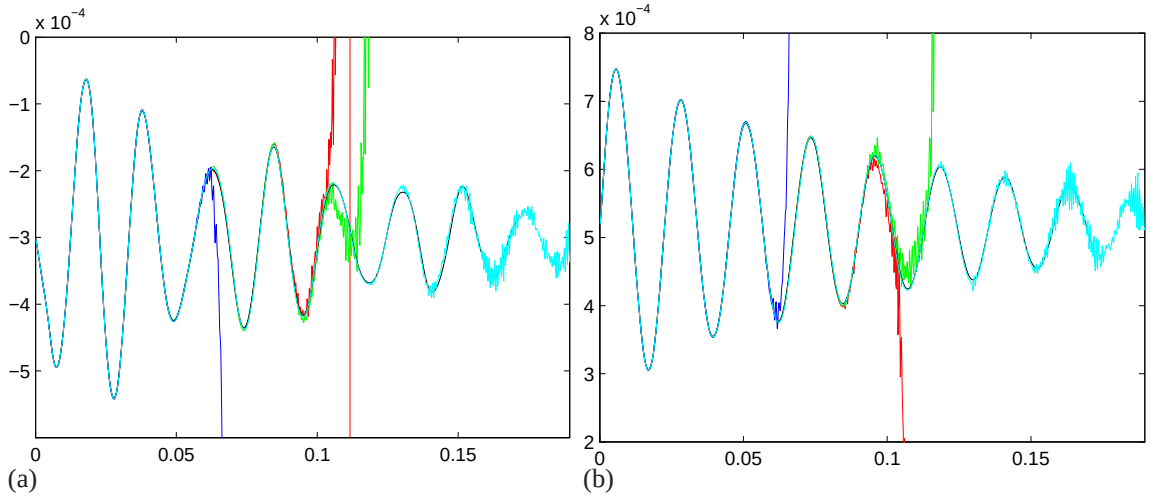


Figure 18: Structural displacements for (a) $\theta = 180°$ and (b) $\theta = 90°$ for different structural damping ratios: 0%(—), 0.0134% (—), 0.0149%(—), 0.0238%(—) and 0.0596%(—).

Figure 19 shows the deformations of the structure for respectively the damping ratios 0%, 0.0134% and 0.0149% during the time step before the collapse of the structure. The deformed shape consists of a combination of the eigenmode (1,19) with another eigenmode (1,2) or (1,3). The eigenmode (1,19) has an eigenfrequency of $2907\,\text{Hz}$. At this frequency the modal damping ratio is respectively equal to 0%, 0.35% and 0.39%. Further research should clarify the cause of these instabilities and the presence of eigenmode (1,19) in the response of the structure.

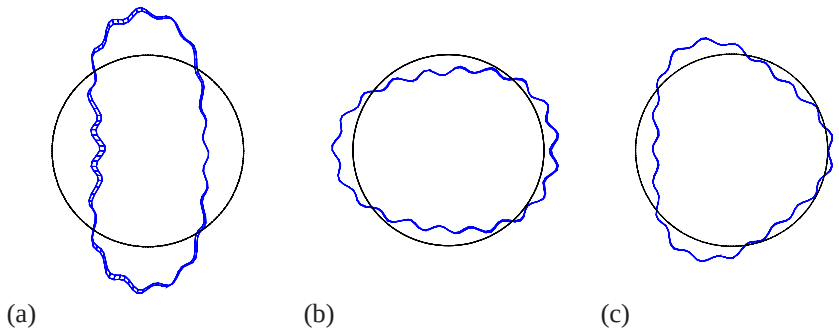


Figure 19: Deformations of the structure for the damping ratios (a) 0%, (b) 0.0134% and (c) 0.0149% during the time step before the collapse of the structure.

6 Conclusion

During a storm in October 2002, ovalling was observed on several empty silos of a group of forty silos in the port of Antwerp. A coupled numerical analysis of the wind flow around one silo is performed.

The differences between the eigenfrequencies of a validated three-dimensional finite element model of the silo structure and a coarser model without the local refinements near the boundaries are quite small. The eigenmodes with the lowest eigenfrequencies, around $4\,\text{Hz}$, have a circumferential wavenumber $n = 3$ and 4.

The two-dimensional wind flow around a single cylinder is modelled at a Reynolds number of $12.4 \cdot 10^6$. An unsteady incompressible RANS simulation using the Shear Stress Transport turbulence model is performed. The vortex shedding frequency is equal to $2.13\,\mathrm{Hz}$. The three-dimensional wind flow around a single cylinder yields a slightly higher vortex shedding frequency ($2.16\,\mathrm{Hz}$).

The three-dimensional finite element model of the silo is coupled to the three-dimensional wind flow around a single cylinder. The structure and the fluid are sequentially coupled with a conventional serial staggered algorithm and an iteratively staggered algorithm. In the iteratively staggered algorithm, the transferred displacements are not relaxed. Maximum four interfield iterations are needed in order to obtain a relative change of the transferred quantities smaller than 0.001. The accuracy of the conventional serial staggered algorithm is lower than the accuracy of the iteratively staggered algorithm. The response of the silo is dominated by the eigenmodes with circumferential wavenumber $n = 3$ and 4. The structural deformations influence the pressure field near the wall: the pressure fluctuations at the vortex shedding frequency have clearly increased and due to the interaction pressure fluctuations are present as well at the lowest eigenfrequencies of the structure.

In order to reduce the computational cost of the fluid-structure interaction computations, the model of the structure is reduced to two dimensions using the finite strip method. This finite strip model of the cylinder is coupled to a two dimensional flow. The coupling procedure is validated by means of experimental results of wind tunnel tests reported in the literature (Laneville and Mazouzi, 1995). The lowest eigenfrequencies and mode shapes of the finite strip model correspond well with experimental results. As ovalling occurs when the negative aerodynamic damping cancels the structural modal damping, the structural damping is reduced step-by-step in order to find the damping ratio at which ovalling occurs.

For the highest damping ratio 0.0596%, the response of the structure is damped out and shows a stable behaviour. For the four lower damping ratios, the response of the structure becomes unstable during the computations and the computations finally crash due to excessive deformations of the structure. Further research should clarify the cause of these instabilities.

References

Ansys: *Ansys CFX-Solver Theory Guide Ansys CFX Release 10.0*. Ansys, Inc. (October 2005a).

Ansys: *Ansys Theory reference, Ansys Release 10.0*. Ansys, Inc. (July 2005b).

Behr, M.; Hastreiter, D.; Mittal, S.; Tezduyar, T.: Incompressible flow past a circular cylinder: Dependence of the computed flow field on the location of the lateral boundaries. *Computer Methods in Applied Mechanics and Engineering*, 123, (1995), 309–316.

Behr, M.; Liou, J.; Shih, R.; Tezduyar, T.: Vorticity streamfunction formulation of unsteady incompressible flow past a cylinder : Sensitivity of the computed flow field to the location of the outflow boundary. *International Journal for Numerical Methods in Fluids*, 12, (1991), 323–342.

Billington, D.: *Thin shell concrete structures*. McGraw-Hill (1965).

BIN: *NBN ENV 1991-2-4: Eurocode 1: Basis of design and actions on structures - Part 2-4 : Actions on structures - Wind actions*. Belgisch Instituut voor Normalisatie (November 1995).

Cheung, Y.: *Finite strip method in structural analysis*. Pergamon Oxford (1976).

Dooms, D.; De Roeck, G.; Degrande, G.: Influence of the group positioning of cylinders on the wind pressure distribution in the post-critical regime. In: P. Wesseling; E. Oñate; J. Périaux, eds., *Proceedings of European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, Egmond aan Zee, the Netherlands (September 2006a).

Dooms, D.; De Roeck, G.; Degrande, G.: Reynolds Averaged Navier Stokes simulation of the post-critical flow around a single circular cylinder and groups of cylinders. In: *Proceedings of the 7th National Congress on Theoretical and Applied Mechanics*, National Committee for Theoretical and Applied Mechanics, Mons, Belgium (May 2006b).

Dooms, D.; Degrande, G.; De Roeck, G.; Reynders, E.: Finite element modelling of a silo based on experimental modal analysis. *Engineering Structures*, 28, 4, (2006c), 532–542.

Dooms, D.; Jacobs, S.; Degrande, G.; De Roeck, G.: Dynamische analyse van een groep silo's onder dynamische windbelasting: In situ metingen op een lege hoeksilo. Report to Ellimetal BWM-2003-09, Department of Civil Engineering, K.U.Leuven (June 2003).

Forsberg, K.: Influence of boundary conditions on the modal characteristics of thin cylindrical shells. *AIAA Journal*, 2, 12, (1964), 2150–2157.

Hughes, T.; Liu, W.; Zimmermann, T.: Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29, (1981), 329–349.

Iaccarino, G.; Ooi, A.; Durbin, P.; Behnia, M.: Reynolds averaged simulation of unsteady separated flow. *International Journal of Heat and Fluid Flow*, 24, 2, (2003), 147–156.

Laneville, A.; Mazouzi, A.: Ovalling oscillations of cantilevered cylindrical shells in cross-flow: New experimental data. *Journal of Fluids and Structures*, 9, 7, (1995), 729–745.

Menter, F.: Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32, 8, (1994), 1598–1605.

Pearson, K.: On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 6, (1901), 559–572.

Zdravkovich, M.: *Flow around circular cylinders. Vol 1: Fundamentals*. Oxford University Press (1997).

Address: D. Dooms (corresponding author), Department of Civil Engineering, K.U.Leuven, Kasteelpark Arenberg 40, B-3001 Leuven, Belgium.
email: david.dooms@bwk.kuleuven.be.

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 97– 114
Herrsching am Ammersee, 29.9.-1.10.2008

A Computational Environment for Membrane-Wind Interaction

T. Gallinger, A. Kupzok, U. Israel, K.-U. Bletzinger and R. Wüchner

Within this paper the development of a computational environment for the specific case of light-weight structure-wind interaction is addressed. To highlight modularity a partitioned solution approach is chosen in combination with Dirichlet-Neumann type coupling and adaptive under-relaxation. The specific needs of the single-fields, like turbulent wind flow in the atmospheric boundary layer and dynamic structural behavior with large displacements, are addressed and the simulation techniques to cope with these are given. Emphasis is placed on the software realization at the coupled interface, for which a seperate programme was developed. Different application examples of the environment are given, in which important single aspects are proven and a complex real-world application is treated.

1 Introduction

Predicting and analysing structural behavior of optimized and light-weight structures like membranes and very thin shells are topics of high interest in structural engineering. One of the most decisive influences on this special type of structures is wind, which necessitates an in-depth analysis of this phenomenon. In general, the influence of wind is difficult to predict. The standard procedure in civil engineering is to reduce complexity by introducing appropriate assumptions. However, this approach involves the risk of overestimating or neglecting effects which result from the strong coupling of the two different physical fields. On the one hand, this leads to structures being oversized and inefficient due to the need for large safety factors, on the other hand this can also lead to damages or even catastrophic failures. A prominent example for this is the Tacoma Narrows bridge desaster.

Thus, there is a need to reproduce the coupled behavior between light-weight structures and wind as accurately as possible or necessary. In this regard, the numerical simulation of aeroelastic effects is a promising complement to and enhancement of current methods in use. The purpose of this research is the development of a computational environment for the design and analysis of light-weight structures under wind influence, being able to capture all relevant phenomenona and helping to improve design and sizing. The physical phenomena of this specific case, that have to be captured by the environment, can be briefly sketched as:

- Represent wind flow within the atmospheric boundary layer with influence of surrounding buildings and landscape. This leads to characteristic Reynolds numbers in the order of 10^6 to 10^7.
- Simulate behavior of light-weight prestressed structures, which are a result of form finding processes. Typically with complex and non-parametric geometries and, due to high flexibility, large dispacements under wind influence.
- The combination of wind and high flexibility leads to aeroelastic effects, that have a significant influence on the structural behavior and, therefore, cannot be neglected.

Summarizing the physical phenomena leads to the following requirements for the computational environment on modeling and numerics:

- Ability to represent the important characteristics of physical wind by modeling highly turbulent flows in a steady and unsteady regime.

- Form finding of prestressed membrane and cable structures.
- Geometric nonlinear dynamic and static computations of membranes and thin shells.
- Simulation of aeroelasticity by efficient, stable, and reliable algorithms.
- Execution of massively parallel computations to cope with real-world problems.

This paper is organized as follows. In section 2, the physics of the problems are stated by the underlying set of governing equations and models. Section 3 describes the selected numerical approach to solve the physics. Section 4 gives an insight into the realized software implementation. In Section 5 different examples are shown: First some principle examples to proof important single aspects of the environment separated from each other and then a realistic example of a real-world structure showing the capabilities of the developed environment. The paper ends by a conclusion and an outlook on the next planned steps.

2 Governing Equations

The whole simulation domain Ω is split into a fluid domain Ω^F and a structure domain Ω^S, with $\Omega = \Omega^F \cup \Omega^S$ and the interface between fluid and structure denoted by Γ^{FS}, with $\Gamma^{FS} = \Omega^F \cap \Omega^S$.

The fluid flow on a moving domain can be assumed to be incompressible and is described by the momentum equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \frac{\partial \overline{\mathbf{d}}}{\partial t})\nabla \mathbf{u} = -\frac{1}{\rho_F}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) \qquad in \quad \Omega^F$$

and the continuity equation

$$\nabla \cdot \mathbf{u} = 0,$$

which set up together with boundary and inital conditions

$$\mathbf{u}(x,t) = \mathbf{u}_\Gamma(t), \quad p(x,t) = p_\Gamma(t) \quad on \quad \Gamma^F$$

$$\mathbf{u}(x,0) = \mathbf{u}_0(x), \quad p(x,0) = p_0(0)$$

the well-known Navier Stokes equations (NSE). Herein, $\mathbf{u}$ denotes the fluid velocity, $\frac{\partial \overline{\mathbf{d}}}{\partial t}$ the observer velocity and p the pressure.

The set of equations describing the structure problem is given by the momentum equation

$$\rho_S \frac{d^2 \mathbf{d}}{dt^2} - \nabla \cdot (\mathbf{F} \cdot \mathbf{S}) = \mathbf{f} \quad in \quad \Omega^S,$$

the constitutive equation

$$\mathbf{S} = \mathbf{C} : \mathbf{E},$$

giving a relation between the second Piola-Kirchhoff stresses $\mathbf{S}$ and the Green-Lagrange strains $\mathbf{E}$, the kinematic equation

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}),$$

with $\mathbf{F} = \nabla \mathbf{d}$ as the deformation gradient. The boundary and initial conditions are

$$\mathbf{d}(x,t) = \mathbf{d}_\Gamma(t), \quad \mathbf{S}(x,t) = \mathbf{S}_\Gamma(t) \quad on \quad \Gamma^S$$

$$\mathbf{d}(x,0) = \mathbf{d}_0(x), \quad \mathbf{S}(x,0) = \mathbf{S}_0(0).$$

Herein, $\mathbf{d}$ denotes the structural displacements and $\mathbf{f}$ the sum of external and internal forces. At the interface between fluid and structure Γ^{FS} certain conditions have to fulfilled.These are the continuity of displacements

$$\overline{d}_{\Gamma^{FS}} = d_{\Gamma^{FS}},$$

and continuity of surface traction

$$\overline{t}_{\Gamma^{FS}} = t_{\Gamma^{FS}}.$$

3 Partitioned Solution Approach

In principle, different approaches to solve the above stated fluid-structure interaction problem exist. The most common ones are monolithic and partitioned strategies. Within a monolithic strategy the whole nonlinear coupled problem, containing terms from the fluid and the structure field and interfield terms, is solved simultaneously. The solution can be carried out very efficiently, but full access to the field solvers is required, which spoils software modularity and makes the use of black-box solvers impossible. Moreover, the consideration of the two different physics leads to huge coefficient matrices which might even be ill-conditioned.

Within a partitioned strategy, the single fields are solved independently from each other, exchanging and preserving interface conditions via their boundary conditions. This gives the possibility of chosing the best suited solver and methodology from a broad range of existing ones for the specific single field. Within the context of simulating interaction of complex geometries with turbulent flow, a partitioned method is known to be the most convenient and flexible approach (Cebral, 1996; Farhat and Lesoinne, 2000; Kalro and Tezduyar, 2000; Stein et al., 2001; Wüchner, 2006). In the context of this work, a partitioned method is therefore chosen. In the following, the numerical treatment of the single fields is given, followed by a description of the interface treatment.

3.1 Fluid Field

The fluid flow has to follow the physical principle of the Navier Stokes equations in its incompressible form on a moving domain. Therefore, the equations are given within an ALE framework. For the discretization of the underlying equations, the cell centered Finite Volume Method (FVM) on unstructured grids is used and a segregated solution approach is chosen.

Because FSI simulations show naturally a transient characteristic, the segregated pressure based solution algorithm PISO (Issa, 1985; Ferziger and Perić, 1999) for solving the transient fluid flow is applied. This means, that the momentum equation is linearized, the coupled system of equations is solved in a sequence and the pressure equation enforces the conservation of mass. According to Jasak (1996), the PISO algorithm splits up into the following steps:

- **Momentum predictor:** Solve the momentum equation, given by

$$a_P U_P = H(U) - \sum_f S(p)_f.$$

  The index P denotes values at the cell centre, f values at faces, that belong to the current cell. a_P is a matrix coefficient that belongs to the cell velocity, and S is the face area vector. Because the pressure gradient is not known, use the pressure field from the previous time step or coupled subiteration. This gives an approximate solution of the new velocity field, but it is not divergence free.

- **Pressure solution:** Assemble the $H(U)$ operator, given by

$$H(U) = -\sum_N a_N U_N + \frac{U^0}{\Delta t},$$

  using the predicted velocities. The index N denotes all neighboaring values. Setup the pressure equation

$$\sum_f S \cdot [(\frac{1}{a_P})_f (\nabla p)_f] = \sum_f S \cdot (\frac{H(U)}{a_P})_f.$$

  The Solution gives an estimate of the new pressure field.

- **Explicit velocity correction:** Use the new pressure field to evaluate the conservative face fluxes

$$F = S \cdot U_f = S \cdot [(\frac{H(U)}{a_P})_f - (\frac{1}{a_P})_f (\nabla p)_f].$$

  Correct the velocities in an explicit manner by solving

$$U_P = \frac{H(U)}{a_P} - \frac{1}{a_P} \nabla p$$

- **Loop until convergence:** Setup new $H(U)$ operator based on corrected velocities. Repeat steps **Pressure solution** and **Explicit velocity correction** until convergence.

It should be noted, that for steady-state simulations the SIMPLE algorithm is applied, which is a simplification of the above stated PISO algorithm. Examining the time fraction, which is needed within one time step to solve the evolving equations, leads to the observation that arround 50% to 80% of the total simulation time is spent for solving the pressure equation due to its elliptic nature. Thus, it is of main interest to solve this equation in a most efficient way. The procedure applied follows the recently published work of Jasak et al. (2007). It is a combination of a Krylov space and multigrid method with a strong smoother within the multigrid levels. The Krylov method used is of type Conjugate Gradient (CG), a standard solver for linear systems. Its efficiency depends extremly on the chosen preconditioner, which is traditionally the Incomplete Cholesky factorisation (ICCG). However, as proposed in Jasak et al. (2007), here an Agglomerative Algebraic Multigrid Solver (AAMG) is used as preconditioner to lower the condition number. This leads together with the strong smoother of type Gauss-Seidel to a substantial performance improvement compared to standard ICCG or AMG solvers.

The schemes applied for discretization of temporal derivatives, convective and diffusive terms are all seccond order in space and time. In the majority of the performed calculations, the following schemes are used: The temporal derivative is approximated by the implict, second-order accurate three time level scheme, referred to as the backward scheme, the diffusive terms by Gaussian integration with central differences and the convective terms by a TVD limited form of central differences.

Because the spatial domain shape changes in time, a mesh movement technique is necessary. This is done by prescribing the boundary velocity of the fluid domain $\partial_t \overline{d}_\Gamma$ and solving a Laplacian equation within the domain of type $\nabla \cdot (\gamma \nabla (\partial_t \overline{d}_\Gamma))$. The diffusion coefficient γ is defined per finite volume. Typically, it is chosen to be dependent on the nearest distance l between the current cell center and the moving boundary, by the formula $\gamma = \frac{1}{l^m}$ and m being equal to 2. More details on the implementation are given in Jasak and Tukovic (2007). It is noted, that the combination of the implemented moving mesh technique and the backward scheme for the temporal discretization is known to satisfy the space conservation law, otherwise errors in the form of artificial mass sources would occur (Demirdĉić and Perić, 1988; Förster et al., 2006; Tukovic and Jasak, 2008).

A specific topic within the computational modeling of wind flow is the treatment of the turbulent flow behavior. Wind flow is naturally highly turbulent, with typical Reynolds numbers beyond the order of 10^6. There exist different methodologies to capture turbulent effects, the most common ones are Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and methods based on Reynolds Averaging (RANS). DNS and LES are computationally extremly demanding and, in the case of complex and large civil engineering structures, the computational power is currently simply not available to solve problems with this approach. Therefore, RANS is the method of choice, which use is also widespread in industrial flows application. In this work, the Shear Stress Transport (SST) model (Menter and Esch, 2001; Menter et al., 2003) is used, which is a zonal formulation of a two-equation eddy-viscosity model based on blending functions. The $k - \epsilon$ model is employed in the free-shear layers of the flow. In the near wall layer, however, the $k - \epsilon$ model is known to overpredict turbulent kinetc energy k, what leads to incorrect prediction of separation points in the flow. Therefore, in the boundary layers the $k - \omega$ model is used, which is known to give substantially more accurate results. It has recently been shown, that this model gives good results for wind-induced flows in comparison to wind-tunnel data (Yang et al., 2008).

3.2 Structure Field

The principal structure problem, as stated above, is also known as Cauchy's first equation of motion. It is of transient and geometric nonlinear nature, leading to a description that distinguishes between undeformed configuration $\mathbf{X}$ and deformed configuration $\mathbf{x}$. Due to the limitation on large deformations but small strains, the St. Venant-Kirchhoff material law is used to set up the constitutive equation.

The geometry of a membrane is given as a two-dimensional surface in space, which is described by the two surface parameters Θ^1 and Θ^2. The curvilinear base vectors are defined by

$$\mathbf{g}_\alpha = \mathbf{x}_{,\alpha} = \frac{\partial \mathbf{x}}{\partial \Theta^\alpha} \qquad \text{as well as} \qquad \mathbf{G}_\alpha = \mathbf{X}_{,\alpha} = \frac{\partial \mathbf{X}}{\partial \Theta^\alpha}$$

The deformation of a point at the membrane surface depends on the difference of its location in space:

$$\mathbf{d}(\Theta^1, \Theta^2, t) = \mathbf{x}(\Theta^1, \Theta^2, t) - \mathbf{X}(\Theta^1, \Theta^2).$$

The spatial discretization of these equations is based on the Finite Element Method (FEM) with a Lagrangian description, using an isoparametric element concept. This leads to the semi-discrete equations of motion

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{r_{int}}(d) = \mathbf{f}_{ext},$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ the damping matrix, $\mathbf{r_{int}}$ the vector of internal forces and f_{ext} the vector of external forces. The external forces consist of forces arising within the single-field problem, like self-weight or snow, and forces from the surrounding fluid field, and therefore $\mathbf{f}_{ext} = \mathbf{f}_{ext,S} + \mathbf{f}_{ext,FSI}$.

The temporal discretization of the nonlinear problem is based on a time integration algorithm. It should ensure second-order accuracy and high numerical stability. Therefore, the implicit generalized-α scheme is applied. Additionally, it offers the possibility of controlled numerical damping of high-frequency noise and minimizing errors in the lower modes of interest. The structures under consideration are thin-walled light-weight structures like shells and membranes, which are based on the assumption that the negligibly small thickness h stays constant during deformation. The latter has an important impact on the formulation of the coupling interface, i.e. data transfer at the interface and the fluid field solver must be able to handle infinitely thin surfaces.

Membrane structures are typically supported at their edges by prestressed edge cables, leading also to a prestressed state in the undeformed configuration. The undeformed configuration of a membrane structure is therefore a state of equilibrium between prestresses in the edge cables and internal stresses in the membrane. The shape of the membrane is not known a priori and an additional analysis has to be performed in advance, being known as form finding procedure. The need for special form finding procedures results from the singularity of the inverse problem of finding the corresponding shape to a given stress distribution. This difficulty is overcome by the Updated Reference Strategy (URS) (Bletzinger et al., 2005; Wüchner and Bletzinger, 2005; Bletzinger et al.), a regularization technique based on a homotopy mapping.

3.3 Coupling Strategy

There exist several different strategies to handle surface coupled partitioned analysis. The problem of the interaction of wind and light-weight membranes belongs to the type of physically strongly coupled problems, because the large deformations influence the surrounding fluid flow. The requirements for the coupling algorithm can be stated as follows: Most important is to ensure a very stable and highly efficient coupling. Additionally, it has to be easy to implement, to ensure the possibility of simple and fast replacement of the field solvers. It has been shown recently (Küttler and Wall, 2008), that a fixed-point scheme with adaptive under-relaxation is a good choice, if the under-relaxation parameter is chosen in an efficient manner and it shows good performance also compared to other, more sophisticated, methods. The under-relaxation of the transferred quantities is described by $\tilde{\mathbf{d}}_{\Gamma,i+1} = r_i \cdot \overline{\mathbf{d}}_{\Gamma,i+1} + (1 - r_i) \cdot \overline{\mathbf{d}}_{\Gamma,i}$ with the under-relxation parameter r_i in subiteration i. The calculation of the under-relaxation parameter is based on Aitkens formula in the formulation of Irons & Tuck (Wall, 1999; Mok, 2001). The Aitken factor is given by:

$$\mu_i = \mu_{i-1} + (\mu_{i-1} - 1) \cdot \frac{(\Delta\overline{\mathbf{d}}_{\Gamma,i} - \Delta\overline{\mathbf{d}}_{\Gamma,i+1})^T \cdot \Delta\overline{\mathbf{d}}_{\Gamma,i+1}}{(\Delta\overline{\mathbf{d}}_{\Gamma,i} - \Delta\overline{\mathbf{d}}_{\Gamma,i+1})^2}.$$

The relaxation parameter is then given by $r_i = 1 - \mu_i$.

The coupling algorithm for one time step within a typical transient FSI calculation is given in a schematic manner in algorithm 1.

4 Computational Concept

4.1 Requirements and Overview of Realization

The applications to be treated within the software environment range from small two dimensional principle examples up to complex real-world applications with several millions of degrees of freedom. This leads to the following requirement specifications for the software devlopment process:

- A highly modular environment, so that for each specific problem the best-suited single field solvers can be adopted easily.

Algorithm 1 Coupling algorithm

for $t = 0$ to $t = t_{tot}$ **do**
  while (!converged) **do**
    Apply interface forces $\sigma^n_{\Gamma,i}$
    Solve $\mathbf{M}\ddot{d}^n + \mathbf{r^n_{int}}(d^n) = f^n_{ext,i}(\sigma^n_{\Gamma,i})$
    Evaluate interface displacements $d^n_{\Gamma,i}$

    Calculate relaxation parameter r_i
    Relax displacements $\tilde{d}^n_{\Gamma,i} = r_i \overline{d}^n_{\Gamma,i} + (1 - r_i)\overline{d}^n_{\Gamma,i-1}$

    Apply boundary displacement $\tilde{d}^n_{\Gamma,i}$
    Move fluid mesh $\nabla \cdot (\gamma \nabla(\partial_t \tilde{d}^n_{\Gamma,i}))$
    Apply PISO, solve for u and p
    Evaluate interface forces $\sigma^n_{\Gamma,i+1}$

    Proof convergence, e.g. $\|\sigma^n_{\Gamma,i+1} - \sigma^n_{\Gamma,i}\|_{L^2} < \epsilon$
    $i \leftarrow i + 1$
  end while
  $n \leftarrow n + 1$
end for

- Use of different communication concepts, because different codes offer different possibilities to connect at their interfaces.
- Possibility to perform massively parallel computations on clusters and supercomputers to simulate real-world applications within reasonable time.
- Handle different interface discretizations with non-matching surface meshes, resulting from different discretization techniques and demands of the specific single fields.

To fulfill all the above stated requirements, a software realization was chosen based on three different codes (see figure 1). Two codes representing the single-field sovers for fluid and structure, both being able to work in parallel, and a third code being located between the field solvers. This third code, called coupling code, operates as a master process, controlling the whole simulation, and working as an interface between fluid and structure field.

The codes that are used within the scope of this paper, are: (I) for the structure field the in-house code CARAT, (II) for the coupling the in-house code CoMA, (III) for the fluid a solver based on OpenFOAM (Weller and Fureby (1998), www.openfoam.org), a freely available set of object-oriented libraries for finite volume discretization. It should also be noted that the commercial CFD software package CFX-11 is also used at the institute within the coupling environment (Wüchner et al., 2007), what highlights the modularity of the software concept. In what follows, the emphasis is placed on CoMA, as it is an substantial part of the software concept.

4.2 Code Coupling via CoMA

The name CoMA stands for **C**oupling f**o**r **M**ultiphysics **A**nalysis. It is a programme responsible for the simulation control within a surface-coupled simulation and is located as a central process between the single-field solvers. It offers a great variety of features, making it a powerful tool for all kind of surface coupled simulations. It is a fully object-oriented programme written in C++, making it easy to extend or adapt to new needs and allowing extensive code-reuse. It is able to communicate with single-field codes working in parallel, allowing communication between the fields based on the Message Pasing Interface (MPI) standard, which is the first choice, if the single field solvers allow access to the source code, or based on files and directories, if the access to the source code of the single field solvers is limited, like in commercial software packages. It can handle surface discretizations, that differ on the two fields, and do not match. At the surface, flux quantities and field quantities can be mapped. In what follows the different parts of CoMA will be explained in more detail.

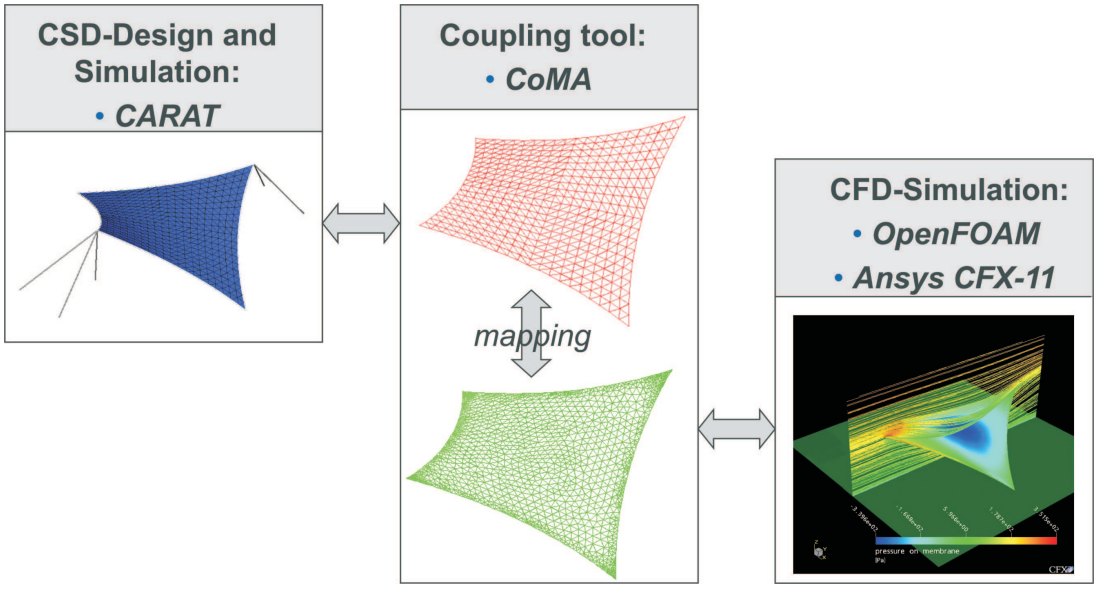


Figure 1: Partitioned approach - Software overview

4.2.1 Programme Work Flow

The work flow of CoMA can be subdivided into two main parts: Firstly, an initialisation part, in which the communication is set up, the surface meshes are received and the mapping between the surface meshes is performed. The second part represents the loop over the time steps and sub-steps, in which an exchange, mapping and convergence check of the quantities is performed. A pseudo-code representation of the work flow for a usual Dirichlet-Neumann coupling strategy is given in algorithm 2.

Algorithm 2 CoMA workflow

Initialise Communication environment
Receive surface meshes
Initialise mapping operator $\mathfrak{L}$
for $t = 0$ to $t = t_{tot}$ **do**
  while (!converged) **do**
    Receive structure displacements $d^n_{\Gamma,i}$
    Map $d^n_{\Gamma,i} \xrightarrow{\mathfrak{L}} \overline{d}^n_{\Gamma,i}$
    Calculate relaxation parameter r_i
    Relax displacements $\tilde{d}^n_{\Gamma,i} = r_i \overline{d}^n_{\Gamma,i} + (1 - r_i)\overline{d}^n_{\Gamma,i-1}$
    Send displacements $\tilde{d}^n_{\Gamma,i}$
    Receive fluid forces $\overline{\sigma}^n_{\Gamma,i+1}$
    Map $\overline{\sigma}^n_{\Gamma,i+1} \xrightarrow{\mathfrak{L}} \sigma^n_{\Gamma,i+1}$
    Send forces $\sigma^n_{\Gamma,i+1}$
    Proo convergence $\|\sigma^n_{\Gamma,i+1} - \sigma^n_{\Gamma,i}\|_{L^2} < \epsilon$
    $i \leftarrow i + 1$
  end while
  $n \leftarrow n + 1$
end for

4.2.2 Communication Concept

The great flexibility of the communication concept is one of the main features within CoMA. It provides the possibility to communicate with other processes on a file-based interface - this allows to couple with commercial tools - or, what is of particular use within massively parallel computations on supercomputers and clusters, establishes a communication based on the Message Passing Interface (MPI) standard, which is the de-facto standard for parallel

executed programmes. A communication based on MPI has several advantages: It allows fast, reliable and efficient communication between processes, what is of main interest within parallel computations. As far as it is known to the authors, almost all modern programmes that offer the possibility to be executed in parallel do this based on MPI communication routines. This means, that for the use within a coupling environment the already existing implementation of the MPI routines just has to be adapted to allow specific communication with CoMA. But this can be also be seen as a disadvantage. The existing communication concept in single-field codes comes out of a long and error-prone development process. Therefore, major changes to the existing routines for single-field communication are not desired, because this would lead to a long-lasting search for errors and bugs. As a conclusion, it was one of the main goals to develop a communication concept, that requires as few as possible changes to existing procedures.

The main idea of the communication concept can be described as follows: The communication within parallel executed programmes is based on communicators, typically MPI_COMM_WORLD, a global communicator in which all started processes of the programme are part of. If a coupled computation is performed, all single-field programmes and CoMA are started together and form one global MPI group, consisting of several processes of different kind. Now, the single-field processes of the same kind setup a processgroup and are grouped together into a subfield with a certain sub-communicator (e.g. MPI_COMM_FLUID, MPI_COMM_STRUCTURE). The existing single-field communication routines now just have to base their communication not on the formerly used MPI_COMM_WORLD communicator, but on the now available sub-field communicator. This allows to re-use all existing single-field communication routines by only changing the communicator type. For the communication between the single-field processes and CoMA the MPI_COMM_WORLD communicator is used. CoMA just communicates with a subset of the single-field processes, because not all processes possess a coupled boundary. A graphical representation of the communicator concept is given in figure 2. Also, a limitation was set to just use MPI commands declared within MPI Standard 1.1, to allow interoperability with programmmes based on this standard. This type of programming model is known as MPMD - Multiple-Programme Multiple-Data programming. It should also be noted, that a communication based on the TCP/IP level could easily be implemented in CoMA due to its modular and flexible programme structure, but has not been done up to now due to a lack of necessity.

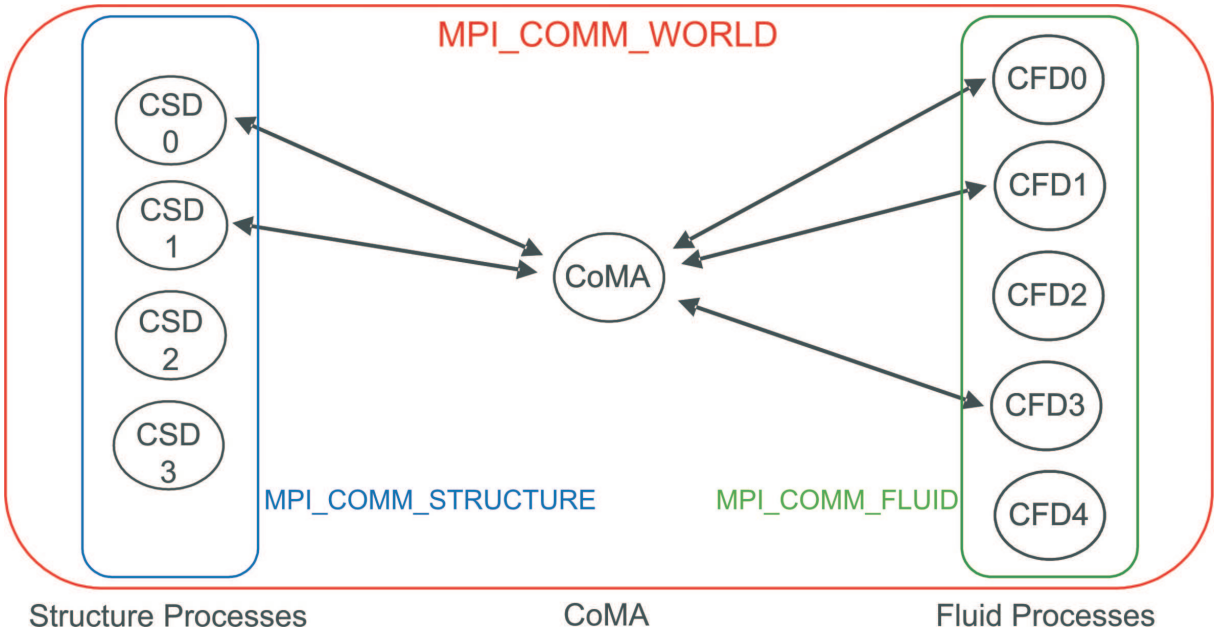


Figure 2: Communication and communicator concept

4.2.3 Surface Data Transfer

The handling of data transfer between the different surface meshes of fluid and structure is one of the most important tasks in doing FSI simulations. After transferring specific data from one to the other surface mesh, it acts as an interface boundary condition for the respective single field. Achieving high numerical accuracy in this operation is one of the primary conditions for successfull FSI calculations. In realistic simulations, like the interaction of wind and membranes, the subproblems have different resolution requirements and use different discretization tech-

niques, leading to the fact of non-matching surface meshes with different surface element types. Basically, there exist two different methodologies for a conservative data transfer on non-coincident surfaces: Methods based on interpolation and Mortar methods (Felippa et al., 2001). Within CoMA, an interpolation based method is chosen. In the case of wind-structure interaction, or aeroelasticity in general, the fluid surface mesh is generally much finer than the structure's one. For this case an interpolation method shows a similar accuracy compared to Mortar (Farhat et al., 1998; Farhat, 2004), accompanied with less implementation and numerical effort.

The procedure of the mapping algorithm is described exemplarily with the node-based quantities displacements and forces. This choice doesn't imply any restriction of the developed software but demonstrates the necessary mappings for the coupling strategy mentioned above. First of all, a neighbourhood search between the surface mesh nodes of the fluid mesh and the midpoint of the triangulated structure mesh is performed. This provides the projection points P_j of the fluid nodes onto the structural surface elements. The displacement field d of a structural element in the context of Finite Element Methods is given by shape functions and nodal values:

$$\mathbf{d} = \sum_i N_i \mathbf{d}_i.$$

To transfer the displacement field onto the fluid surface mesh, it is evaluated at the projection points P_j of the fluid nodes within the current structure element, described by their location in natural coordinates ξ_j and η_j, and applied as a displacement boundary condition to the according fluid node (see left picture in figure 3):

$$\overline{\mathbf{d}}_{P_j} \approx \mathbf{d}(\xi_j, \eta_j) = \sum_i N_i(\xi_j, \eta_j)\mathbf{d}_i.$$

This interpolation method can be used for all field variables, e.g. pressures, temperatures or velocities.

The transfer of forces, which belong to the type of integrated variables, follows a load-conservative approach (Cebral and Löhner, 1997). It is also based on an interpolation principle. The fluid forces are known at the projection points P_j of the fluid nodes within the structure element. At structure node P_i the nodal force f_i is evaluated as a sum of the fluid forces f_j within the element, weighted by shape functions (see right picture in figure 3):

$$f_i = \sum_j N_i(\xi_j, \eta_j) f_j.$$

Under the prerequisite of $\sum_j N_j = 1$, it follows that:

$$\sum_i f_i = \sum_i \sum_j N_i(\xi_j, \eta_j) f_j = \sum_j \sum_i N_i(\xi_j, \eta_j) f_j = \sum_j f_j,$$

what serves as a proof of the postulated conservation property of the mapping algorithm. Illustratively speaking, this means that all forces wich act at an element are transferred to the nodes as forces with the same total magnitude.

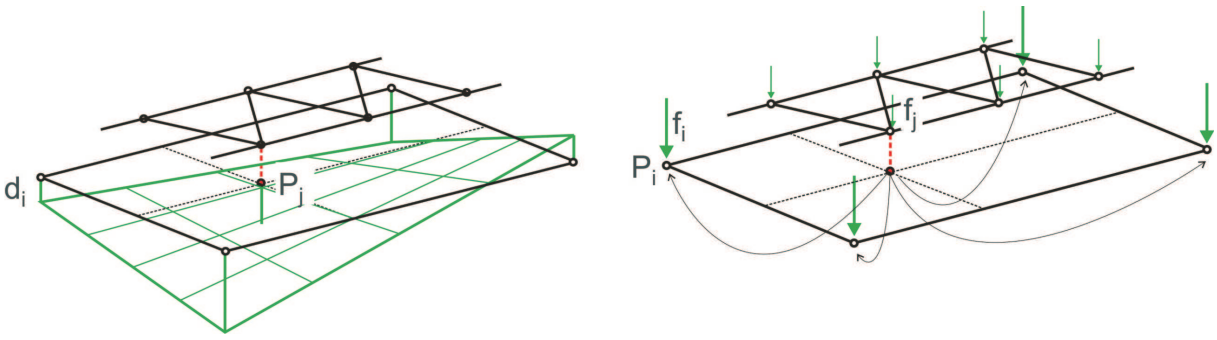


Figure 3: Interpolation of field variables (left) and forces (right)

Remarkably, the implementation of the surface data transfer in CoMA is not limited to forces and displacements, but can be used for any surface information, like temperature distribution, pressure or sensitivities (Israel et al., 2008). There exists also the possibility of handling multiple interfaces within one simulation with separate mapping of quantities. This is especially useful for computations with extremely thin membranes, where the mapping of quantities on the top and bottom surface has to be separated from each other, otherwise leading to errors within the neighbourhood search. But this feature can also be used to handle simulations with more than one elastic object, e.g. one using a geometric nonlinear and the other one using a geometric linear approach. An example for a transfer of quantities at a curved surface is given within the next section.

5 Examples

In this section different examples are shown. The first two verify important single aspects of the environment, namely the mapping of field-quantities on doubly-curved surfaces and the ability to perform coupled computations in the steady-state and transient, but not turbulent regime. The third example shows a real-world application, a complex light-weight membrane structure and its behavior due to wind influence.

5.1 Field Mapping on non-matching Surface Meshes using CoMA

The correct transfer of surface data at the interface Γ^{FS} between fluid and structure domain is a major aspect in doing FSI simulations. The accuracy of the whole simulation is affected by the procedure applied at the interface. In this test case a realistic example for the data transfer algorithm is constructed. The surface treated is inspired by the structural example of a 4-point tent, which is a 3-dimensional doubly curved free form surface and cannot be described analytically. The two surface meshes, composed of triangular elements, are non-matching. One side S^S is a regular mesh, consisting of 1600 elements and representing a typical structure mesh for this example. The other side S^F is an unstructured mesh. It consists of 4854 elements, is refined at the boundary, and should therefore represent a typical fluid mesh in this example. A picture of the two meshes is given in figure 4.

Figure 4: Structured and unstructured surface meshes

Now the distribution of a field variable, e.g. displacements, is specified on the structured surface mesh and mapped onto the unstructured one. The distribution follows an analytical function in global coordinates. Evaluating this function at the node positions P_j^S of the structured mesh leads to the prescribed nodal values $\mathbf{f}_{P_j^F}$. After the data transfer, the nodal values at the unstructured mesh $\mathbf{f}_{P_i^S}$ are known and can be compared to the values given by the analytical P_j^F function, by evaluating the coordinates of the specific node. The analytical function is prescribed on the projection to the ground view and is given by

$$\mathbf{F}(x,y) = cos(C_1 y) \cdot [sin(C_2 x) + C_3 \cdot sin(C_4 x) + C_5 \cdot (1 - C_6 x^2)],$$

which is a superposition of different functions. The constants $C_{1..6}$ are chosen to scale the function to the geometry of the domain. The error at point P_j^F is then computed by

$$e_j = \frac{\left\| (\mathbf{F}_{P_j^F} - \mathbf{f}_{P_j^F}) \right\|}{max(\mathbf{F}_{P_j^F})} \cdot 100.$$

The influence of the difference between the continuous function and the interpolation of values is neglected. A contour plot of the computed error is given in figure 5. It can be seen that the error in the field itself is nearly zero. Just some single elements, located directly at the surface boundary, show higher errors. But this effect can be neglected in nearly all simulations. As a result, it can be said, that the mapping and data transfer algorithms implemented in CoMA show good accuracy in the case of complex surfaces and meshes and can therefore be used in real-world simulations.

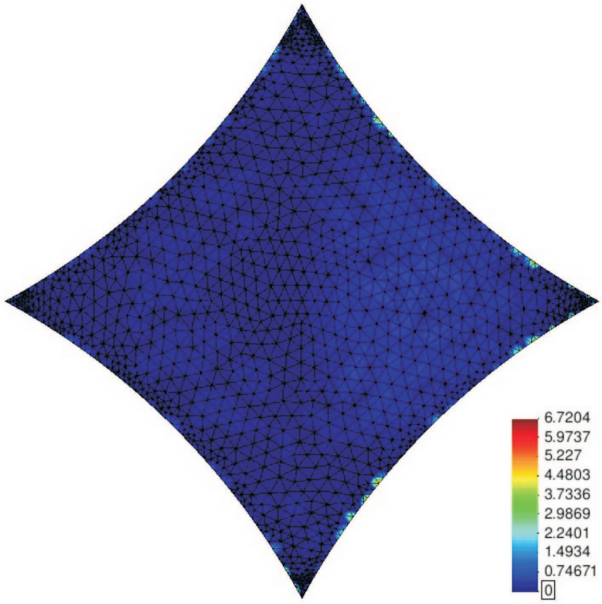


Figure 5: Error distribution on unstructured mesh $[\%]$

5.2 Coupled Computation in laminar Regime

After evaluating an important single aspect of the computational environment, in this example a coupled computation is carried out. The testcase has to ensure the following prerequisits: Two-dimensionality to keep computational time low, test transient and steady-state behavior and ensure a laminar flow regime, because the treatment of turbulence is of great complexity, so it should be kept out to focus on the coupling issues. The testcase developed within the DFG research group 493 is chosen (Turek and Hron, 2006), because it fulfills all the requirements. As a positive side effect, it contains also test scenarios for the single field solvers, so that these can be tested separate.

The test case is described as follows: A fixed, rigid cylinder is placed slightly unsymmetric into a wall-bounded channel. An elastic beam is attached at the back of the cylinder. The geometric representation of the computational domain is given in figure 6 and a description including all details can be found in Turek and Hron (2006). If a flow is imposed onto the system, the flow regime behind the cylinder influences the beam and leads to deformations. Up to a certain Reynolds number, the flow is steady, if the Reynolds number exceeds a certain value, the flow gets unsteady and Karmann vorticies occur behind the cylinder. It should be noted, that only laminar cases are taken into account.

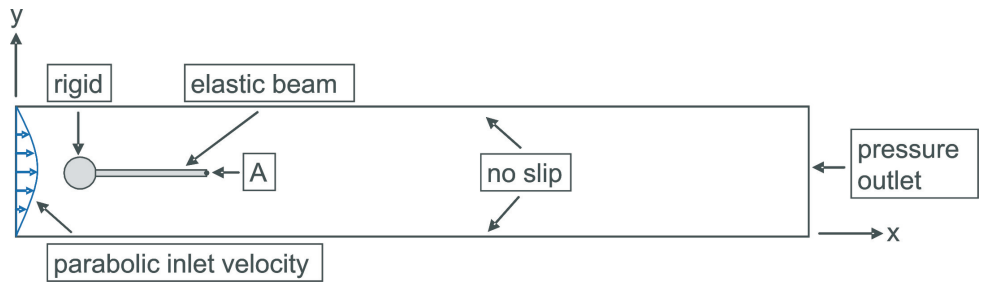


Figure 6: Computational domain

Different mesh levels are examined on both fields. The results shown in this contribution are those, which give the best compromise between accuracy and numerical effort. The principle setup of the software environment is three-dimensional. So for the simulation of this two-dimensional case, one element is used for the discretization in the transversal direction and all effects of the third direction are avoided by proper choice of the boundary conditions. The structure, a cantilever beam, is modeled using 4-noded shell elements. The total number of shell elements is 30 and the mid-surface of the shell is located in the transverse direction of the domain. The geometric representation of the interface between fluid and structure domain is evaluated by a projection of the element surface along the nodal director vector with a factor of $\pm 0.5 \cdot t_{Shell}$, representing the lower and upper surface of the shell. The fluid

is modeled based on a block-structured grid. The total number of fluid elements is 54000. The values taken as reference are the results published in Turek and Hron (2006).

5.2.1 Single Field Computation

First, the single field solvers are tested seperatly from each other fortheir steady and unsteady, respectively static and dynamic behavior. In the structure case, three different tests are performed. As load the gravitational force in $-y$ direction is applied. The first two cases, CSM1 and CSM2, are static calculations with varying modulus of elasticity. The third case, CSM3, consists of a transient computation, in which the gravitational forc is applied starting at time $t = 0$, leading to a harmonic oscillation of the beam as described in Turek and Hron (2006). The results, given in table 1, show the displacements in y-direction of node **A**. The differences to the reference solution are below 0.2%, so it can be said, that the structure field solver works properly and is therefore evaluated. The fluid case also consists of three different tests, in which the inlet velocity is differed. The first and seccond case, CFD1 and CFD2, reaches a steady-state, the third case, CFD3, is of transient nature. The results, given in table 2, show the drag and lift coefficients, evaluated by integrating over the cylinder and beam surface. It can be seen, that for steady-state the differences are below 1%, and for unsteady below 2%. As a consequence, it can be said, that the fluid solver works properly and is verified.

	reference	own	difference
CSM1	-0.06610	-0.06610	0.00%
CSM2	-0.01697	-0.01697	0.00%
CSM3	± 0.06516	± 0.06546	0.45%

Table 1: CSD benchmark results

	Lift			Drag		
	reference	own	difference	reference	own	difference
CFD1	1.119	1.117	0.18%	14.29	14.31	0.14%
CFD2	10.530	10.440	0.85%	136.70	137.52	0.60%
CFD3	± 437.81	± 445.35	1.72%	± 5.61	± 5.50	1.96%

Table 2: CFD benchmark results

5.2.2 Coupled Computation

In the coupled computation an interaction between the surrounding fluid and the elastic beam is simulated. Three different cases are examined, called FSI1 to FSI3, in which the inlet velocity, the structure density and the elastic modulus are varied. FSI1 results in a steady, whereas FSI2 and FSI3 in unsteady solutions, showing periodic oscillations of the beam with differing frequency and amplitude. The transient cases are strongly coupled, therefore the above described fully-implicit coupling scheme with adaptive under-relaxation is applied. The results are given in table 3, examining the y-displacement of point **A** at the beams tip as being the most significant one. It can be seen, that the differences are below 3.5%, which is in the range of the results of all other groups contributing to this benchmark computation. Therefore, also the coupled computation is verified.

	reference	own	difference
FSI1	0.0008209	0.0007999	2.55%
FSI2	± 0.08060	± 0.08338	3.45%
FSI3	± 0.03438	± 0.03473	1.02%

Table 3: FSI benchmark results

5.3 Mobile Canopy Structure

After verifying important single aspects of the developed computational environment, in this example a real-world application is examined. It shows the capabilities of the environment to handle complex geometries and simulate

challenging structural behavior in higly turbulent atmospheric boundary layer flow. The example is inspired by a cooperative work of the architectural department of the Technische Universität München (Gengnagel, 2005) and the University of Dundee. The structure is a canopy, used as a tribune roof in e.g. open-air cultural and sports events. Due to the demand of mobility, lightness is one of the major design parameters. The structure is constructed as follows: Two curved cantilever arms, build as space frames, are clamped at the ground and span over a distance of $11.5m$. A membrane is spanned between the cantilever arms having a width of $6.5m$. A cable is fixed to the membrane at its center axis and the ground. Prestress is applied onto the cable, leading to a deformation of the membrane and the cantilevers towards the ground. Due to this deformation, prestress is applied onto the whole system, leading to a major increase in stiffness under the influence of external loads. A representation of the structure model is given in figure 7. Due to its extreme lightness wind influence is one of the major aspects in the design and sizing of this structure. Therefore, a numerical simulation of the wind impact on the system is performed to gain detailed insight into the structural behavior.

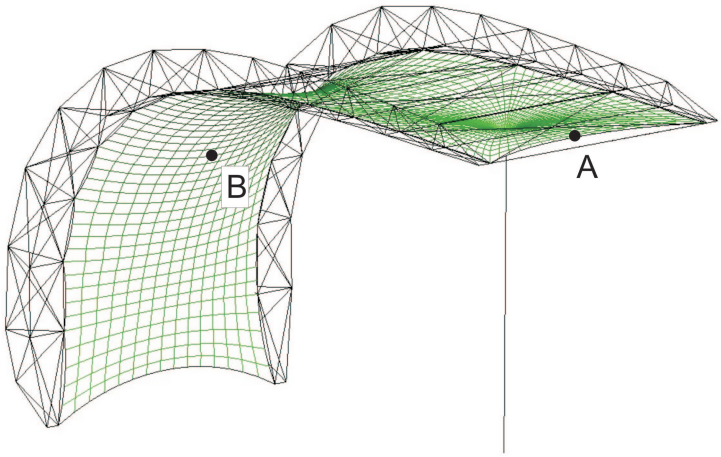


Figure 7: Mobile canopy structure model

5.3.1 Computational Models

The structure model is a combination of prestressed trusses and membranes. The initial geometry of the model is found by a form finding procedure, assuming an isotropic prestress distribution. All computations are performed considering fully geometric nonlinear kinematics. The fluid domain is discretized by an unstructured tetrahedral mesh. The wind flow within the atmospheric boundary layer is simulated by a proper choice of the boundary conditions. A wind speed of $U_{ref} = 15\frac{m}{s}$ in a reference height of $z_{ref} = 10m$ is assumed, what corresponds to a strong wind and intensity 6 on the Beaufort scale. The canopy is exposed to the roughness length of $z_0 = 0.1m$. Following the work of Richards and Hoxey (1993) and Hargreaves (2007), the inlet velocity is given by

$$U(z) = \frac{u_*}{\kappa} ln(\frac{z + z_0}{z_0}),$$

where u_* is the friction velocity and κ is von Karman's constant. The turbulent kinetic energy and the dissipation rate are given by

$$k = \frac{u_*^2}{\sqrt{C_\mu}} \quad and \quad \epsilon = \frac{u_*^3}{\kappa(z + z_0)},$$

with C_μ is a model constant, usually set to 0.09. Under these assumptions, a Reynolds number of $Re = 5 \cdot 10^6$ is reached. The turbulent effects are modeled by the RANS methodology, using the $k - \omega$ SST turbulence model. First, CFD-only simulations are performed, until a steady-state is reached. This state is used as starting point for the coupled computations. A geometric representation of the cantilever arms in the fluid domain is neglected, because the major influence of the fluid flow on the structure is captured by the flow effects on the membrane surface. At the fluid-structure interface non-matching surface meshes occur. Because of the small membrane thickness, the data transfer at the lower and upper surface of the interface has to be done separated from each other. Otherwise, the neighbourhood search between surface nodes of structure and fluid mesh would mix up the two sides and result in errors.

5.3.2 Simulation Results

Three different simulations are carried out, examining major influences of wind onto the structure. The first two analyse the static deformation of the structure under the two main wind directions directly from the back and from the front, which is referred to as case 1 and 2. Therefore, a constant velocity profile is applied at the inlet and a coupled computation towards a steady-state is performed. Numerical efficiency is reached by an explicit coupling scheme in combination with fluid subcycling and under-relaxation. Two points on the membrane surface are examined in more detail, the locations of Points **A** and **B** are shown in figure 7. The results of the coupled computation can be seen in figure 8 for case 1 and figure 9 for case 2, giving a contour plot of the displacements in z-direction and the displacements d_z of point **A** and $\|\mathbf{d}\|$ of point **B** versus the number of coupled iterations. It is noted, that due to under-relaxation a smooth convergence behavior is reached and the computations are nearly converged after only 35 coupled iterations. Wind blowing from the back is uncritical to the structural behavior, but wind from the front leads to a great reduction of the applied prestress in the back region of the membrane, as can be seen from the positive z-displacements in this region. The reason is, that the wind pressure acts against the existing curvature, which is therefore reduced. For a further increase of wind speed, the risk of a snap through in this region of the membrane is expected. A snap through should be avoided by any means, because the load carrying behavior would be changed dramatically.

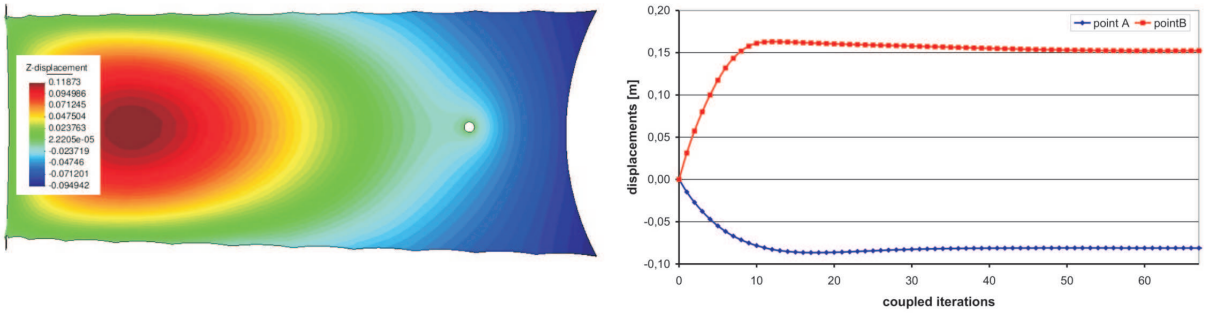


Figure 8: Case 1 - Contour lines of z-displacements $[m]$ and displacements versus coupled iterations

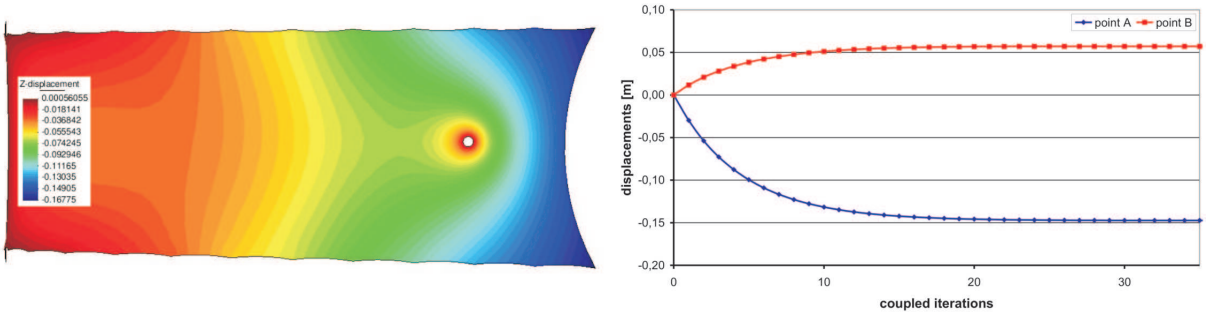


Figure 9: Case 2 - Contour lines of z-displacements $[m]$ and displacements versus coupled iterations

To further examine the structural behavior, a third coupled simulation is performed, called case 3. This calculation is of transient and fully implicit nature. Two artificial wind gusts are applied at the inflow, blowing from front, with a maximum inlet velocity of $30\frac{m}{s}$ at the reference height. The temporal variation of the inlet velocity profile is given by $u_*(t) = 20\frac{m}{s}$ with $t \in [0; 4s]$ and $u_*(t) = 20\frac{m}{s} + 10\frac{m}{s} \cdot sin^2(\frac{t}{4s} \cdot 2\pi)$ with $t \in]4; 10s]$. The total time of the simulation is $10.0s$. Again, the result of a steady-state fluid-only solution is used as initial solution at $t = 0s$. The results can be seen in figure 10. It is noted, that the displacements follow directly the applied wind inflow. This is due to a very small mass inertia of the system as a result of the high lightness. It is also noted, that under the high wind level a snap-through of the membrane in the back region occurs, what results in extremly high displacements due to diminishing geometric stiffnesses.

In summary, the simulations lead to the following results: Wind from back is uncritical for the structure, but wind from front diminishes the applied prestress and gives the risk of a snap-through in the membrane. This was shown in a transient calculation with an increased wind speed level. As impact for the design, it is stated that the applied prestress in this simulations has to be increased to prevent snap-through.

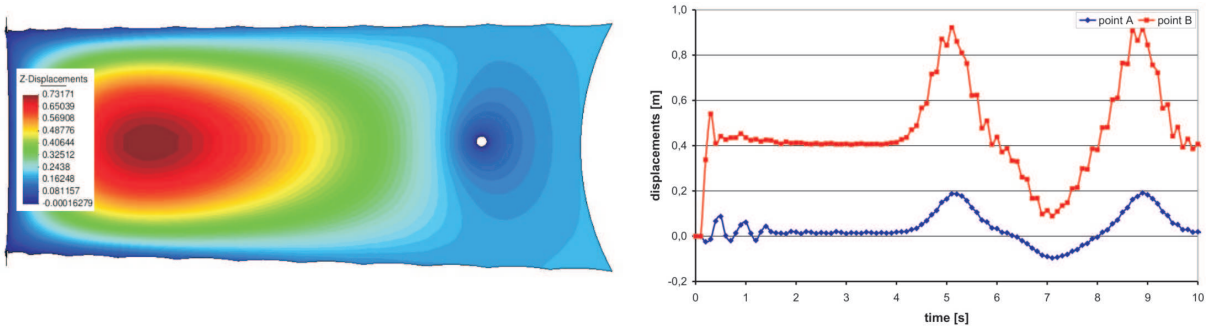


Figure 10: Case 3 - Contour lines of z-displacements $[m]$ at $t = 5.1s$ and displacements versus time

6 Conclusion and Outlook

The development of a general and modular framework for the simulation of light-weight membrane structures interacting with wind is described within this contribution. The evolving strongly coupled system is solved by a partitioned strategy. The realistic modeling of turbulent wind flow leads to large systems, that have to be solved massively parallel. The resulting specific requirements on the framework are handled by a software scheme based on three codes, the two single field solvers and the central master code CoMA. CoMA is a general tool responsible for simulation control and data transfer on non-matching surface meshes. Emphasis is placed on the efficiency and flexibility of the communication concept as an important aspect in MPMD systems. This allows to use CoMA in any surface-coupled analysis with arbitrary coupling quantities.

The next steps will include a further investigation of the correct modeling of wind-induced effects on light and flexible structures. This comprises a further study of the correct simulation of natural wind (e.g. appropriate formulation of inflow conditions, influence of wall modeling, domain size) and a continued consolidation of the coupling procedure with regard to efficiency, reliability and stability. Moreover, the elaboration of selected benchmark tests for verification and validation purposes will be done. Finally, the correct interpretation of the gained results with respect to the design of structures under wind influence is needed and is part of ongoing investigations.

Acknowledgement

The authors would like to acknowledge the cooperative work with Professor Christoph Gengnagel, formerly Technische Universität München, now architectural department of the Universität der Künste, Berlin, who inspired the example of the mobile canopy structure.

References

Bletzinger, K.-U.; Firl, M.; Linhard, J.; Wüchner, R.: Optimal shapes of mechanically motivated surfaces. *Computer Methods in Applied Mechanics and Engineering*, in press, doi:10.1016/j.cma.2008.09.009.

Bletzinger, K.-U.; Wüchner, R.; Daoud, F.; Camprubi, N.: Computational methods for form finding and optimization of shells and membranes. *Computer Methods in Applied Mechanics and Engineering*, 194, (2005), 3438–3452.

Cebral, J.; Löhner, R.: Conservative load projection and tracking for fluid-structure problems. *AIAA Journal of Aircraft*, 35(4), (1997), 687–692.

Cebral, J. R.: *Loose Coupling Algortihms for Fluid-Structure Interaction*. Ph.D. thesis, George Mason University (1996).

Demirdĉić, I.; Perić, M.: Space conservation law in finite volume calculations of fluid flow. *International Journal for Numerical Methods in Fluids*, 8, (1988), 1037–1050.

Farhat, C.: *Encyclopedia of Computational Mechanics 3*, chap. 13, pages 459–480. John Wiley & Sons, Ltd. (2004).

Farhat, C.; Lesoinne, M.: Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182, (2000), 499–515.

Farhat, C.; Lesoinne, M.; LeTallec, P.: Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157, (1998), 95–114.

Felippa, C. A.; Park, K. C.; Farhat, C.: Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190, (2001), 3247–3270.

Ferziger, J.; Perić, M.: *Computational Methods for Fluid Dynamics*. Springer New York (1999).

Förster, C.; Wall, W.; Ramm, E.: On the geometric conservation law in transient flow calculations on deforming domains. *International Journal for Numerical Methods in Fluids*, 50, (2006), 1369–1379.

Gengnagel, C.: *Mobile Membrankonstruktionen*. Ph.D. thesis, TU München (2005).

Hargreaves, W. N., D.M.: On the use of the k-ϵ model in commercial cfd software to model the neutral atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 95(5), (2007), 355–369.

Israel, U.; Stavropoulou, E.; Barcelos, M.; Gallinger, T.; Bletzinger, K.-U.; Wüchner, R.: Shape optimization in partitioned analysis of fluid-structure interaction. *Fluid-Structure Interaction. Theory, Numerics and Applications.*.

Issa, R.: Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62, (1985), 40–65.

Jasak, H.: *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Ph.D. thesis, Imperial College London (1996).

Jasak, H.; Jemcov, A.; Maruszewski, J.: Preconditioned linear solvers for large eddy simulation. In: *CFD 2007 Conference, CFD Society of Canada* (2007).

Jasak, H.; Tukovic, Z.: Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30, (2007), 1–18.

Kalro, V.; Tezduyar, T.: A parallel 3d computational method for fluid-structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190, (2000), 321–332.

Küttler, U.; Wall, W. A.: Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1), (2008), 61–72.

Menter, F.; Esch, T.: Elements of industrial heat transfer predictions. In: *16th Brazilian Congress of Mechanical Engineering (COBEM)*, pages 26–30 (2001).

Menter, F.; Kuntz, M.; Langtry, R.: Ten years of industrial experience with the sst turbulence model. *Turbulence, Heat and Mass Transfer*, 4, (2003), 625–632.

Mok, D. P.: *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*. Ph.D. thesis, Institut für Baustatik der Universität Stuttgart (2001).

Richards, P.; Hoxey, R.: Appropriate boundary conditions for computational wind engineering models using the k-epsilon turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics*, 46, (1993), 145–153.

Stein, K.; Benney, R.; Tezduyar, T.; Leonard, J.; Accorsi, M.: Fluid-structure interactions of a round parachute: modeling and simulation techniques. *Journal of Aircraft*, 38, (2001), 800–808.

Tukovic, Z.; Jasak, H.: Simulation of free-rising bubble with soluble surfactant using moving mesh finite volume/area method. In: SINTEF/NTNU, ed., *6th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway (2008).

Turek, S.; Hron, J.: Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. *LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING*, 53, (2006), 371–385.

Wall, W.: *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. Ph.D. thesis, Institut für Baustatik der Universität Stuttgart (1999).

Weller, H., H.G.; Tabor G.; Jasak; Fureby, C.: A tensorial approach to cfd using object oriented techniques. *Computers in Physics*, 12(6), (1998), 620–631.

Wüchner, R.: *Mechanik und Numerik der Formfindung und Fluid-Struktur-Wechselwirkung von Membrantragwerken*. Ph.D. thesis, Lehrstuhl für Statik der TU München (2006).

Wüchner, R.; Bletzinger, K.-U.: Stress-adapted numerical form finding of pre-stressed surfaces by the updated reference strategy. *International Journal for Numerical Methods in Engineering*, 14(2), (2005), 131–145.

Wüchner, R.; Kupzok, A.; Bletzinger, K.-U.: A framework for stabilized partitioned analysis of thin membrane-wind interaction. *International Journal for Numerical Methods in Fluids*, 54, (2007), 945–963.

Yang, W.; Quan, Y.; Jin, X.; Tamura, Y.; Gu, M.: Influences of equilibrium atmosphere boundary layer and turbulence parameter on wind loads of low-rise buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 96 (10-11), (2008), 2080–2092.

Address: T. Gallinger (corresponding author), Lehrstuhl für Statik, Technische Universität München, Arcisstrasse 21, 80333 München.
email: gallinger@bv.tum.de

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 115– 130
Herrsching am Ammersee, 29.9.-1.10.2008

FSI*ce – A Modular Simulation Environment for Fluid-Structure Interactions

B. Gatzhammer, M. Mehl

From the conceptual point of view, modularity and flexibility are inherent to partitioned fluid-structure interaction simulations or, in a broader sense, multi-physics simulations. In fact, they are the big advantage of partitioned approaches in comparison to monolithic approaches. However, this implies the need for software tools that ensure the independence of the involved components and, at the same time, allows for sophisticated coupling strategies going much beyond a pure data exchange. For this purpose, we developed the coupling tool FSI❄ce that will be described in detail in this article including all current functionalities and future potentials.

1 Introduction

Multi-physics simulations have become more and more the focus of computational engineering in the last years. Increasing computer performance has allowed for the inclusion of more and more physical realism in the computational models. Just think of the inclusion of fluid-structure interactions in the evaluation of skyscraper static, of dam simulations covering fluid dynamics, structural mechanics, soil behaviour and their interactions, of aero acoustic simulations in aircraft construction coupled with structure reactions, and many other examples. In the future, this trend is supposed to continue taking into account more and more physical fields to enable highly realistic and accurate simulations. To illustrate this with an example, we could imagine to enhance the simulation of blood flow in a vein by taking into account not only the wall of the vein but also the surrounding tissue with different material properties and partly own dynamics (bones, skin, muscles,...).

To establish a simulation environment for such simulations in a fast and efficient way and, in particular, to cope with the continuous model development and enhancement, partitioned approaches have proven to be highly suitable. In contrast to monolithic methods that establish and solve one system of equations for the complete scenario, partitioned approaches use existing and well-tested codes for each of the involved single-physics problems and couple the codes together using an additional piece of software. Whereas the advantage of monolithic approaches is a higher robustness, partitioned approaches offer a very high flexibility in terms of simulated physical phenomena and solvers used.

To realise the coupling of several codes, different approaches and software tools are available. The most prominent commercial software is `MpCCI` Ahrem et al. (2001) that leans strongly on the `MPI`-syntax and offers functions supporting the transfer of data from one code to another as well as predefined interpolation and projection routines for the mapping of data between the in general non-matching grids of the solver codes at the coupling interface[1]. The Model Coupling Toolkit `MCT` is a scientific library providing programming interfaces for data exchange, interpolation, and manipulation of data for non-matching and even partitioned solver grids Larson et al. (2005). It is widespread in climate simulation. `C-SAFE` Parker and und T. Harman (2006) is a component-oriented realisation of a coupling unit. Although it is restricted to Cartesian solver grids, the successful integration of a fluid-structure interaction model Parker and und T. Harman (2006) shows the general potential of component based approaches.

[1] For fluid-structure interactions, the coupling interface is the contact surface between fluid and structure, for example.

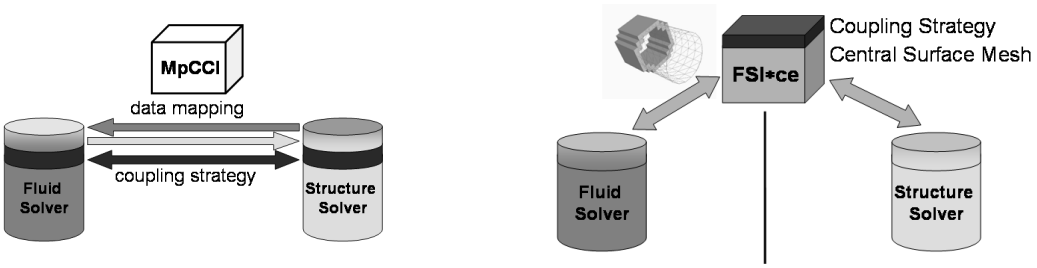


Figure 1: Schematic view of the coupling of two codes for the partitioned simulation of fluid-structure interactions with MpCCI (left) and FSI❄ce (right).

2 Motivation for the Development of FSI❄ce

We listed some commercial and scientific coupling units in 1 and saw that there is no generally applicable framework available that goes far beyond supporting data exchange and data manipulation between two or more solvers. To motivate the development of a our coupling tool for the partitioned simulation of fluid-structure interactions, we have a look at the realisation of the coupling with the help of the most common of the mentioned tools, MpCCI. Figure 1 shows the principle of the coupling in a schematic manner. We start with two given solvers: for the simulation of fluid dynamics on the one hand and the simulation of structural mechanics on the other hand. To make the codes work together, the first thing we have to do is implement the data exchange between the two solvers at the so-called wet surface that is the contact interface between fluid and structure. For this purpose, we can resort to given interpolation and projection methods of MpCCI. However, if we have particular requirements such as higher order discretisations in one of our solvers, new mapping methods have to be established and implemented. The second implementation task concerns the coupling control that is the choice of the coupling strategy in time (explicit/implicit/multigrid/...), the convergence control, the time stepping etc. of the whole coupled simulation. As MpCCI is a tool mainly designed for data exchange, this is done in one or even both of the solvers. If we now exchange one of the solvers, for example, we have to change the implementation of the data mapping in both the new and the remaining solver as well as parts of or even the whole coupling control. Thus, we loose the flexibility to easily exchange solvers and coupling strategies independently, although this flexibility seems to be the main advantage and an inherent property of the partitioned simulation of multi-physics problems.

In order to improve this situation, we developed FSI❄ce Brenk et al. (2005, 2006); Brenk (2007) with the intention to completely hide the components of a partitioned simulation from each other and to provide an easy-to-use tool for the coupled simulation of fluid-structure interactions in a first step but also of general multi-physics scenarios in the future. To separate the coupling strategy and control from the solvers, the first idea behind FSI❄ce was to completely integrate these aspects into a separate coupling unit instead of the solvers. This leads to a client-server approach with FSI❄ce acting as a client and the two solvers acting as servers receiving jobs and queries for data from FSI❄ce Brenk et al. (2005). As a consequence, we can provide coupling strategies that then can be used for arbitrary combinations of solvers. The second idea enforcing the independency of the solvers from each other is to introduce a central coupling mesh in addition to the two solver grids. This coupling mesh discretises the wet surface and acts as a common point of contact for the solvers. That is, both solvers map their simulation data to and get updated data back from it. Thus, a solver, once being prepared for the use with FSI❄ce, can be coupled with an arbitrary partner solver without further changes Brenk et al. (2005, 2006, 2007). Figure 1 displays the resulting setup: Two solvers implementing a data mapping to and from the central mesh and FSI❄ce providing the whole coupling control and the central mesh itself.

Flexibility and ease of use are very important ingredients of a tool designed for the coupling of codes for partitioned multi-physics simulations. However, these two tasks were not the only motivation for establishing FSI❄ce as an own numerical unit and not only a data mapping library. FSI❄ce also provides the possibility to implement numerical coupling methods at a central position that could not or hardly be implemented in one of the solvers. We list some examples here, taking into account that this list of course can not be complete.

2.1 The Transient Localised Lagrangian Multiplier Method

A first and very obvious example is the method of Localised Langrangian Multipliers (LLM) as proposed in Park et al. (2001); Ross (2006). It introduces an interface frame between fluid and structure similar to the central mesh

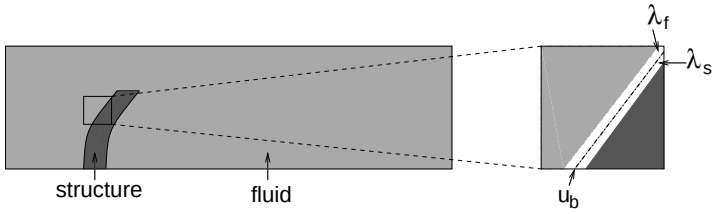


Figure 2: Introduction of a central interface frame describing the wet surface in the transient Localized Lagrangian Multiplier method Ross (2006). The continuity of the interface variable between the interface frame and the solvers is enforced with the help of two Lagrangian multipliers λ_f (for the fluid side) and λ_s (for the structure side).

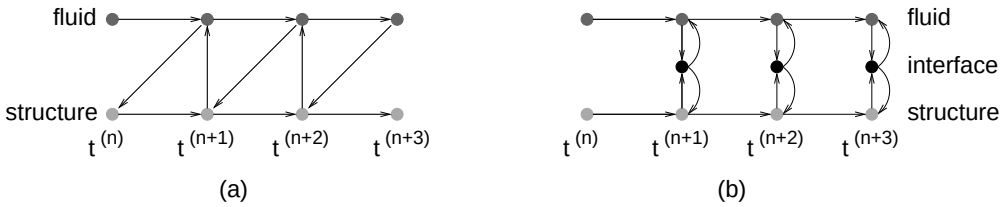


Figure 3: Comparison of the staggered explicit coupling (a) and the transient LLM method (b).

in FSI❄ce (see Figure 2).

This interface frame holds informations on the state of the interface. Depending on the actual method, this state might be described by the speed or the displacement of the interface, e.g. The resulting system of equations involves the fluid equations, the structure equations, and three additional systems of equations for the interface frame state and the two Lagrangian multipliers. These three additional systems constitute side conditions for the fluid and the structure solver including continuity of velocities or displacements and Newton's third law. Instead of the commonly applied solution strategy for the explicit coupling, we can now execute a time step both for the fluid and the structure simultaneously and, then, use a projection method to establish a so-called interface equation for the Lagrangian multipliers and the interface state. These interface data are in turn used to update the fluid and structure state. Figure 3 compares the common 'staggered' explicit discretisation to this new approach. To solve the interface equation, an autonomous numerical component such as FSI❄ce is required. Ross (2006) proved that this method is unconditionally stable for some for the example of the linear piston problem as long as the time stepping schemes of both the fluid and the structure solver are unconditionally stable. This can not be achieved with the staggered explicit coupling.

Table 1 shows that the solvers have to be only slightly changed to be able to update data at their wet surface according to the Lagrangian multipliers whereas all other work is and has to be done by the coupling component.

2.2 Implicit Coupling with Reduced Order Models

A second example is a coupling method using reduced order models to perform a Newton iteration for the non-linear Dirichlet-Neumann interface equation Vierendeels (2006a,b)

operation	solvers	FSI❄ce
solve single-physics problems	x	
collect wet surface data (including Lagrangian multipliers)		x
solve the interface equation		x
send Lagrangian multipliers to solvers		x
update the wet surface according to the Lagrangian multipliers	x	

Table 1: Algorithmic steps in a transient LLM method Ross (2006) and their allocation in the solvers or the coupling component.

operation	solvers	FSI❄ce
solve single-physics problems	x	
collect wet surface data		x
establish reduced order models		x
perform Newton iteration on the interface		x
send interface data to solvers		x
update wet surface boundary conditions	x	

Table 2: Algorithmic steps of a fluid-structure interaction simulation using an implicit coupling strategy based on reduced order models Vierendeels (2006a,b) and their allocation in the solvers or the coupling component.

$$D^{(n+1)} = \underbrace{R_s \circ \text{structure\_solve}(R_f \circ \text{fluid\_solve}(D^{(n+1)}, F^{(n)}), S^{(n)})}_{=: B\left(D^{(n+1)}\right)}, \tag{1}$$

where $D^{(n+1)}$ is the structure displacement at the wet surface at time $t^{(n+1)} = t^{(n)} + dt$, R_s denotes an operator that restricts the structure state to the displacements at the wet surface, structure_solve and fluid_solve are arbitrary structure and fluid solvers, R_f computes the forces exerted on the structure by the fluid on the wet surface from the fluid state. $F^{(n)}$ and $S^{(n)}$ are the fluid and structure state at time $t^{(n)}$. To solve this method with the help of a Newton method, we have to perform iterations

$$D_{i+1}^{(n+1)} = D_i^{(n+1)} + \left(I - J\left(D_i^{(n+1)}\right)\right)^{-1} \left(B\left(D_i^{(n+1)}\right) - D^{(n+1)_i}\right), \tag{2}$$

where $J\left(D_i^{(n+1)}\right)^{-1}$ denotes the Jacobian of $B\left(D_i^{(n+1)}\right)$. This Jacobian is always costly to compute. In the case of black-box solvers, it is even impossible to compute it in any other way than with the help of finite differences that is out of question in terms of efficiency. However, the Jacobian can be approximated by replacing it by the Jacobian computed from reduced order models for the fluid and the structure solver. For these reduced order models, we can exactly and efficiently determine the Jacobian and the application of the inverse of $\left(I - J\left(D_i^{(n+1)}\right)\right)$ on $B\left(D_i^{(n+1)}\right)$, respectively Vierendeels (2006a,b). The reduced order models themselves are established during the run of the simulation using the input-output relation of the solvers as a data basis. Table 2 shows that in this case, there are a lot of non-trivial actions that have to be taken by a separate coupling component both from a logical point of view (as they concern the coupling, not the solvers) and from a practical point of view (as the method has particularly been designed for black box solvers).

2.3 Multigrid

A third example that requires a central control of the partitioned simulation is a multigrid method not only for the involved solver but for the whole coupled simulation environment. That is, we have to couple fluid and structure solver not only on the fine grid but also on coarser grids. After performing a few iterations for the fine grid equation

$$\begin{pmatrix} F_h \\ S_h \\ W_h \end{pmatrix}^{(n+1)} = \text{FSI}_h \left(\begin{pmatrix} F_h \\ S_h \\ W_h \end{pmatrix}^{(n+1)} \right) \tag{3}$$

with the fluid state variables F_h, the structure state variables S_h, the interface state variables W_h, and the discretised operator of the fluid-structure interaction equation FSI_h, we switch to the coarse grid equation

$$\begin{pmatrix} F_H \\ S_H \\ W_H \end{pmatrix}^{(n+1)} = \text{FSI}_H \left(\begin{pmatrix} F_H \\ S_H \\ W_H \end{pmatrix}^{(n+1)} \right) + R\text{FSI}_h \left(\begin{pmatrix} \bar{F}_h \\ \bar{S}_h \\ \bar{W}_h \end{pmatrix}^{(n+1)} \right) - \text{FSI}_H R \left(\begin{pmatrix} \bar{F}_h \\ \bar{S}_h \\ \bar{W}_h \end{pmatrix}^{(n+1)} \right), \tag{4}$$

operation	solvers	FSI❄ce
solve/smooth single-physics problems	x	
collect wet surface data		x
define how to establish the coarse grid		x
restrict the fine grid solution	○	x
modify the right hand side on coarse grids	○	x
change the level of resolution	x	x
interpolate the coarse grid solution	○	x
update the fine grid solution	○	x
send interface data to solvers		x
update wet surface boundary conditions	x	

Table 3: Algorithmic components of a multigrid method for the whole coupled simulation of a fluid-structure interaction scenario and their allocation in the solvers or the coupling component. Circles indicate that these steps are allocated in the solvers only in case of a method where the solvers only smooth and not solve on each grid level.

where $\bar{F}_h^{(n+1)}$, $\bar{S}_h^{(n+1)}$, and $\bar{W}_h^{(n+1)}$ denote the state approximation for fluid, structure, and interface after the fine grid iterations. This coarse grid equation is then solved by a recursive call of the multigrid solver. To finish the multigrid cycle, the coarse grid approximation is used to update the fine grid approximation according to

$$\begin{pmatrix} F_h \\ S_h \\ W_h \end{pmatrix}^{(n+1)} = \begin{pmatrix} \bar{F}_h \\ \bar{S}_h \\ \bar{W}_h \end{pmatrix}^{(n+1)} + I \begin{pmatrix} F_H \\ S_H \\ W_H \end{pmatrix}^{(n+1)} - IR \begin{pmatrix} F_h \\ S_h \\ W_h \end{pmatrix}^{(n+1)} \tag{5}$$

and some more post smoothing iterations on the fine grid are performed. In this very general formulation, we did not further specify the equation (3) and the kind of smoothing iterations that are performed on each grid level. Bijl et al. (2006) showed that the runtime already for a one-dimensional linear piston problem could be reduced by a factor of 10 with a two-level method.

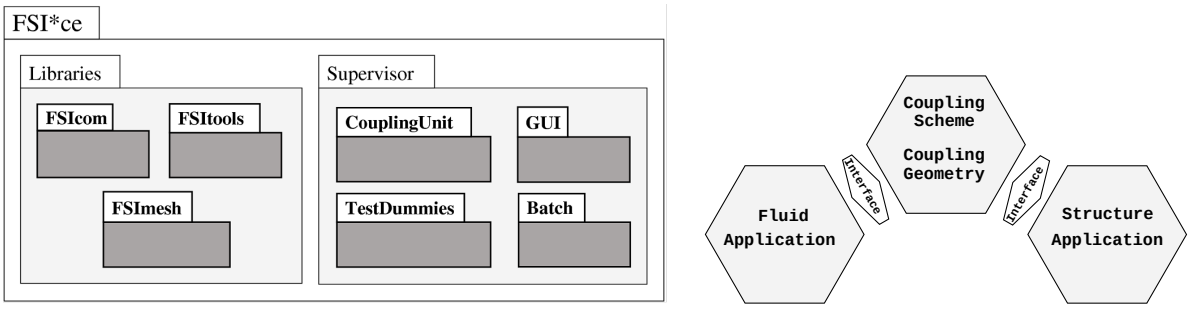
Depending on what solver we use and in how far we can change the solver code, smoothing iterations can either include the complete solution of the fluid and the structure equation on the respective level or only some iterations for both fluid and structure. The first case will be required if we use black-box solvers. In this case, equation (3) will be an interface equation such as equation (2), only. In the second case, the error after the smoothing iterations is not only due to errors in the interface equation but also due to not yet converged solvers for fluid and structure. Such, equation (3) will cover the whole coupled fluid-structure system. Table 3 gives an overview of the algorithmic parts that have to be done in the solvers and those that are done in the coupling component. Circles mark actions that have to be taken by the solvers only in the second case, where the solvers also only smooth instead of solving. It becomes obvious, that even in that case, only standard multigrid functionality (in the sense of a full approximation scheme for nonlinear equations) is required from the solvers. However, the single components of this multigrid functionality have to be addressable separately by the coupling component. Besides, all other functionality and the overall control is integrated in the coupling component, which discharges the user from the re-implementation of a lot of functional elements for each solver pair and, second, in particular for the definition of the coarse grid, is required also from a methodological point of view. Only the central unit can decide how to establish a suitable overall coarse grid from the two solver fine grids and the wet surface.

3 Structure and Functionality of FSI❄ce

After we have motivated the development of FSI❄ce by showing the potential of a coupling unit with a numerical functionality going far beyond data mapping, interpolation and projection mechanism, we will describe the actual realisation of FSI❄ce in this section. FSI❄ce has a modular structure that ensures an easy extendibility (see Sect. 3.1). The main components of FSI❄ce are the coupling mesh (Sect. 3.2), the support for data mapping (Sect. 3.3), and the coupling strategies (Sect. 3.4). Additionally, we shortly present the application programming interface of FSI❄ce which is the essential ingredient for a user friendly tool (Sect. 3.5).

3.1 Modular Structure

A well defined structure is the essential basis for software to last. It simplifies (or even enables) the maintenance and extension of software; and a software that is not maintained nor extended will be thrown away soon. Thus, one goal for the development of FSI❄ce is to maintain a well-thought modular structure of the source code project. The current structure of FSI❄ce is illustrated in Figure 4 (a). It consists of two main parts: first, the libraries, which are meant to be included by a user of FSI❄ce. They contain the coupling mesh (see Sect. 3.2), support for data mapping to and from the coupling mesh (see Sect. 3.3), the communication API (see Sect. 3.5), and implementations of communication mechanisms with MPI and sockets. The second part of FSI❄ce is the coupling supervisor. The coupling supervisor acts as the server in the coupled simulation (Sect. 2) and cares for the exact execution of the chosen coupling schemes. Figure 4 (b) illustrates the client-server concept, and shows the place of the coupling supervisor in the frame of the coupled simulation.



(a) Modular structure of the software package FSI❄ce.

(b) Modular client-server-based realization of FSI at application level.

Figure 4: Modular concept of FSI❄ at internal design and external application level.

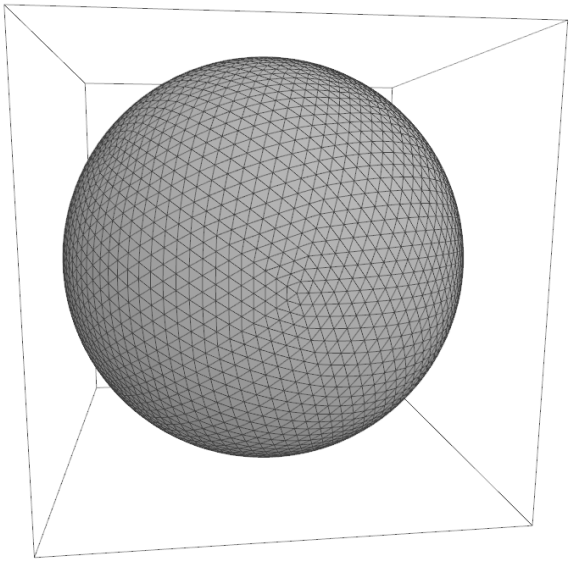
3.2 Coupling Mesh

A unique feature of FSI❄ce among existing coupling tools is the use of an additional mesh instance - the coupling mesh - dedicated for the exchange of data located on the common surface of the coupled solvers. At first sight, this might appear to be an additional complication of the whole coupling process. However, it enables to decouple the solvers from each other and to make them easily and independently exchangeable. Furthermore, it allows to introduce advanced coupling methodologies such as the ones introduced in Section 2.

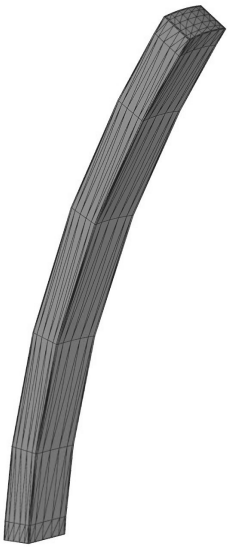
The coupling mesh is chosen as a triangulation of the surface(s) between the single solvers domains. It can be described by a vf-graph, i.e. a graph consisting of **v**ertices interconnected by **t**riangle faces. Figure 5 shows two examples for triangulated surfaces.

The triangle edges are of no importance currently, and are, hence, not considered in the implementation of the coupling mesh. For a pure node centered data transfer from solver to solver, even the triangles would not be necessary. However, since the coupling mesh is not only used for such a simple mapping but also for numerical operations one the triangulation itself (think of coarsening of the triangulation for multigrid methods, e.g.) and to derive geometric relations between the triangulation and the solver grid nodes such as the position of a point or voxel relative to the coupling surface, it is necessary to also have a geometric description of the surface and its orientation and not only the locations of the nodes carrying the data.

The functionality of the coupling mesh is two-fold: First, it holds the topological information of the common interface of fluid and structure and, thus, administrates also all movements and deformations of this interface. Solver grids are adapted after each coupling step according to changes in the interface description. In the setup-phase of a coupled simulation, the coupling mesh can either be derived from a solver grid or, vice versa, act as an input for the generation of a solver grid. The second functionality of the solver grid is storing, modifying, receiving, and sending all values that are to be exchanged between the involved solvers. In the case of fluid-structure interactions and a Dirichlet-Neumann coupling, these are the forces exerted on the structure surface by the fluid, and the displacements of the structure resulting in prescribed velocities for the fluid field. For the realisation of other coupling methods or further multi-physics problems, however, the coupling mesh allows for the storage of arbitrary kinds of vectorial or scalar data sets, annotated to the nodes of the triangular mesh.



(a) Sphere 3d triangulated representation by coupling mesh.

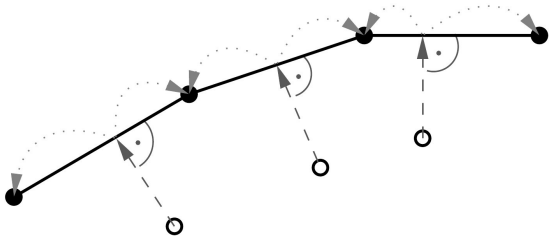


(b) Cantilever 3d triangulated representation by coupling mesh.

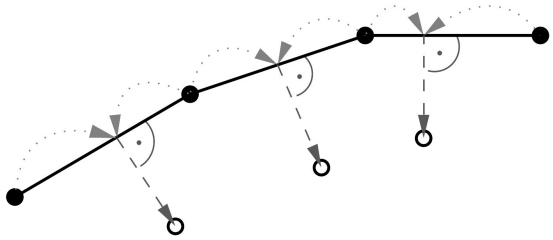
Figure 5: Examples for triangulated surface meshes.

3.3 Support for Data Mapping

The data mapping between the solver grids and the coupling mesh will often require some projection and interpolation of data values from the unknowns stored in the non-matching solver grids to the nodes of the coupling mesh. FSI❄ce supports some simple projection and interpolation methods. Figure 6 illustrates for example the currently implemented mapping to and from the coupling mesh in 2d for forces and displacements on the fluid side. Forces are mapped from the fluid grid to the coupling mesh and displcements vice versa from the coupling mesh to the fluid grid. This mapping is conservative with respect to the total force exerted on the structure and consistent with respect to displacements that is mapping a constant displacement on the coupling mesh to a constant displacement on the respective boundary of the fluid grid[2]



(a) Conservative mapping from solver nodes (empty circles) to coupling mesh nodes (filled circles). The geometric position of the solver nodes is projected orthogonally onto the nearest triangle (dashed lines). By usage of the parametric description of the projected point, i.e. the description by barycentric coordinates, the data value of the projected point can be decomposed and interpolated onto the triangle nodes (dotted lines). The sum of the forces stays constant.

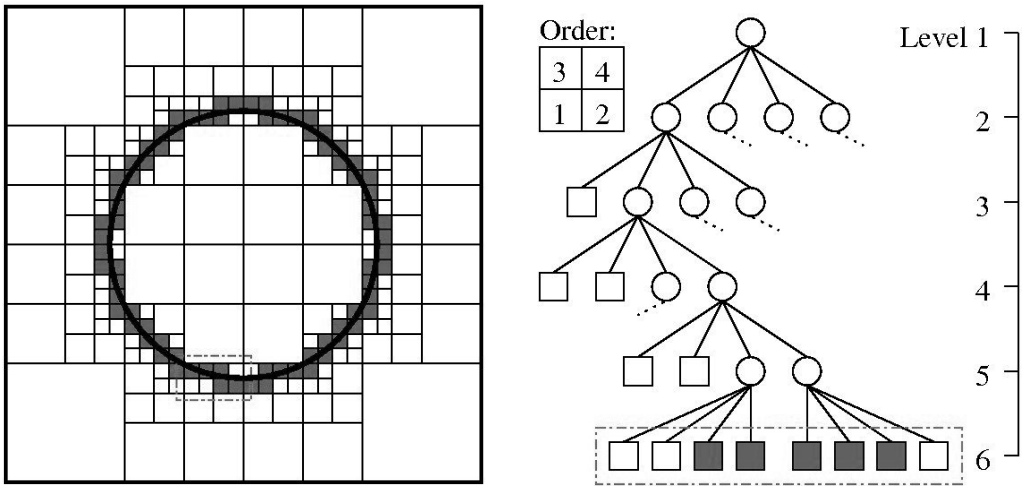


(b) Consistent mapping from coupling mesh nodes (filled circles) to solver nodes (empty circles). The geometric position of the solver nodes is projected orthogonally onto the nearest triangle (dashed lines). By usage of the parametric description of the projected point, i.e. the description by barycentric coordinates, data values are fetched according to their weight from the triangle nodes (dotted lines). The sum of the fetched data values is assigned to the corresponding orthogonally projected solver nodes. The sum of data values is different on coupling and solver grid, but the mean value stays constant (in the sense of a consistent mapping).

Figure 6: Conservative mapping from solver grid nodes to coupling mesh nodes (a), and reversed consistent mapping from coupling grid nodes to solver grid nodes (b).

More specialised mapping methods as required for higher order solver discretisations, for example, have to be

[2]Since all our examples have been computed with a coupling mesh identical to the surface mesh of the structure solver Adhoc Düster et al. (2004), data mapping methods on the structure side are still work in progress.



(a) Domain with circle decomposed into Cartesian quadtree-cells. The leaves of the quadtree are marked as empty (white fill) if the circle does not intersect with the quadtree cell or marked as full (grey fill), if the circle intersects the quadtree cell. Empty cells are not refined any more, while full cells are refined up to a given level.

(b) Tree representation of the domain decomposition from (a). Circles represent refined quadtree cells, squares leaves. The filled squares on the bottom correspond to the leave cells in (a), which are marked by a dash-dotted line. The legend on the top left shows the correspondence of tree nodes to quadtree-cells in (a).

Figure 7: Quadtree example.

implemented by the user itself.

One general functionality that is essential for any kind of data mapping methods, however, can be used in all cases: In order to find the 'mapping goal', that is the closest coupling mesh triangle for an arbitrary solver grid node, a nearest neighbour search has to be performed. The computational complexity of a neighbour search is $O(N \cdot M)$, where N is the number of fluid nodes and M the number of coupling mesh nodes. Assuming the number of fluid and structure nodes to be in the same order of magnitude, we would have to solve a problem that has a computational complexity growing quadratic with the number of nodes. This is not a severe problem for coupling surfaces with a moderate number of unknowns that is for simple geometric objects contained in a fluid flow domain with a much higher amount of unknowns to be solved for. However, it can prolongate the simulation time of a coupled problem unnecessarily, when very fine resolutions for rather large coupling surfaces are needed such as in the case of domains with flexible boundaries in combination with turbulent flows, for example.

In order to speed up the neighbour search process, we introduce an octree, the 3d variant of a spacetree. Figure 7 illustrates the basic concept of the octree: the spatial domain is recursively decomposed into regular subquadrants, but only at elements intersected by the coupling mesh. The recursive refinement is stopped by some maximal depth criteria. The leaves of the octree data structure contain references to the triangles and nodes of the coupling mesh which are (partly) contained in the subquadtrant represented by the respective leave.

Now, the neighbourhood information is already available in the octree and the number of triangles to be checked can be reduced to the ones contained in octree leaves neighbouring the considered solver grid node. It is not sufficient to only look at the octree cell containing this grid node. In addition, also the direct neighbour cells need to be considered since a point at the boundary of the octree cell might have a closer neighbour in a neighbouring octree cell, than in the one it is located in. However, these cells can be efficiently identified in a top-down run over the octree. This concept reduces the computational complexity to $O(N \log M)$ and leads to a negligible runtime requirement for the neighbour search compared to that of the fluid solver, for example Daubner (2005); Brenk et al. (2008). The octree functionality is provided to the user of FSI❄ce in a library (see Sect. 3.1). Hence, he can implement the mapping functionality for the solvers with less efforts.

3.4 Coupling Strategies

FSI❄ce currently incorporates several standard coupling schemes offering a solution procedure for the partitioned coupled problem in time. It includes a staggered weak (i.e. explicit) and a strong (i.e. implicit) coupling scheme with interface iterations. Independent from the coupling scheme chosen, subcycling can be applied for one or both solvers. In addition, non-coupled pre-computations can be performed by one solver, which will be done by the

fluid solver in a typical FSI scenario to achieve a physically realistic initial state. Figure 8, 9, 10 and 11 illustrate and describe these coupling schemes.

Taking advantage of the client-server concept described in Section 2, all the coupling schemes are implemented in the coupling supervisor, saving the user from having to implement coupling schemes themselves and in the solvers, which would result in the need to re-implement the coupling strategy after each solver exchange. The solvers only have to provide a few function calls to the coupling supervisor, which is much easier and incurs much less implementation efforts. We describe the work to be done by the user in the next section treating the application user interface (API) if FSI❄ce.

3.5 Coupling Application Programming Interface (API)

Although the implementation work in the solvers to be coupled with FSI❄ce is kept to a minimum by the client-server based approach, there are still modifications to be done in a solver being prepared for coupled simulations with FSI❄ce. The essential functionalities to control the coupled simulation are bundled in the coupling API, which is shown in listing 1. The first three functions initiate and register a solver with the coupling supervisor, exchange the coupling mesh and relevant data such as the time step length computed, and finalize the contact to the coupling supervisor. The last three functions are there for the solver to query the state of the coupled simulation.

```
void FSI_Init ();
void FSI_Data_Exchange ( timeStepLength );
void FSI_Finalize ();

int FSI_Is_new_interface_values ();
int FSI_Is_implicit_converged ();
int FSI_Is_running ();
```

Listing 1: FSI❄ce coupling API, allowing the solver to be controlled by the coupling supervisor. Function parameters are mostly omitted.

Listing 2 shows the main parts of a solver code. A time stepping loop contains the determination of the length of the next time step, the computation of the flow field of the next time step and the storage of these new values. The second and third step must be separated if a strong coupling scheme is chosen, since the interface iterations require not to advance in time before the convergence of the iterations.

```
while (more time steps)
    Set time step length
    Compute values of next time step
    Store values of next time step
end while
```

Listing 2: Main parts of a solver code.

The coupling API can now be integrated into this code as shown in Listing 3. Obvious are the calls to `FSI_Init()` and `FSI_Finalize()`, which must take place before and after the time stepping loop. The duration of the solver's simulation is now controlled via the function `FSI_Is_running()`. Since reading data from the coupling mesh might have an impact on the computation time, the function `FSI_Is_new_interface_values()` can determine, whether new interface values have been obtained or not. A solver performing pre-computations or subcycling (see Sect. 3.4) does not receive new interface values in every solver-local computation step. A similar function could prevent the writing of coupling data to the coupling mesh, but is not yet implemented. Then, the call to `FSI_Data_exchange()` performs the actual communication of the state of the simulation and the coupling mesh with the coupling supervisor. The computed time step length is sent to the coupling supervisor, which in turn computes a new maximal time step limit for the next time step communicated back to the solver. Note that the solver is always allowed to compute a time step shorter than that proposed by the coupling supervisor, since solver-internal time step length limitations, such as the CFL criterion for a fluid solver, are unknown to the coupling supervisor, but still need to be respected by the solver. The function `FSI_Is_implicit_converged()` must be used, when an implicit coupling scheme is applied. The convergence test is performed within the function

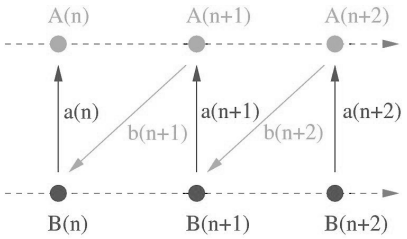


Figure 8: Basic weak coupling scheme with staggered solution procedure. The dashed lines indicate the time axis for each solver. The solution procedure starts at time $t(n)$, where n denotes the time step, with solver B sending its initial interface values $a(n)$ to solver A (this step can be omitted, if solver A is supplied with proper initial interface values). Solver A computes its next time step $t(n+1)$ and sends the obtained interface values $b(n+1)$ to solver B, which in turn advances to the next time step.

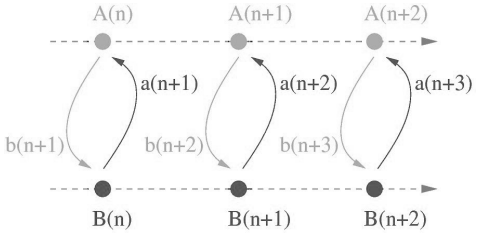


Figure 9: Strong coupling scheme with interface iterations. The dashed lines indicate the time axis for each solver. The solution procedure starts with solver A (but could equally start with solver B) at time $t(n)$. Solver A computes a first prediction $b(n+1)$, for solver B's interface values of the next time step and sends it to solver B. B uses the obtained prediction to compute itself a prediction $a(n+1)$ for solver A's interface values and returns it to solver A. This procedure continues (next would be solver A to compute $b(n+1)$), with both solvers always using the most current interface values, until convergence of the interface values is achieved. Then, solver A and solver B can advance simultaneously to time step $t(n+1)$.

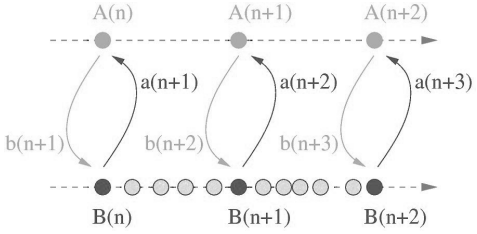


Figure 10: Subcycling solver B combined with a strong coupling scheme. The dashed lines indicate the time axis for each solver. The solution procedure follows the description of Figure 9, with a difference for solver B when advancing to the next time step. Then, solver B computes non-coupled time steps (indicated by circles filled with lighter color) always using the interface values obtained by A at time $t(n)$ until it reaches time $t(n+1)$. Only then a new interface iteration starts. Subcycling can be equally combined with any other of the discussed coupling schemes.

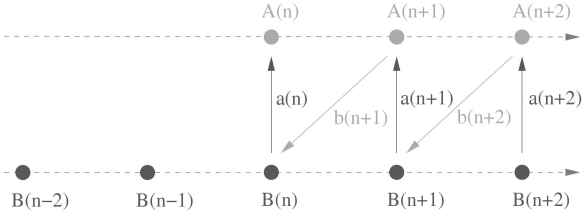


Figure 11: **Pre-computations performed by solver B prior to the coupled simulation.** (The dashed lines indicate the time axis for each solver). Pre-computations can be used to obtain physical meaningful initial values for the solver starting the computation of the coupled problem.

`FSI_Data_exchange()` by default (but can be specialized by the user) and checks the convergence of the interface values of both solvers involved in the coupled simulation.

```
FSI_Init ()
while (FSI_Is_running())
    if (FSI_Is_new_interface_values())
        Read coupling data from coupling mesh
    Set time step length
    Compute values of next time step
    Write coupling data to coupling mesh
    FSI_Data_exchange ()
    if (FSI_Is_implicit_converged())
        Store values of next time step
end while
FSI_Finalize ()
```

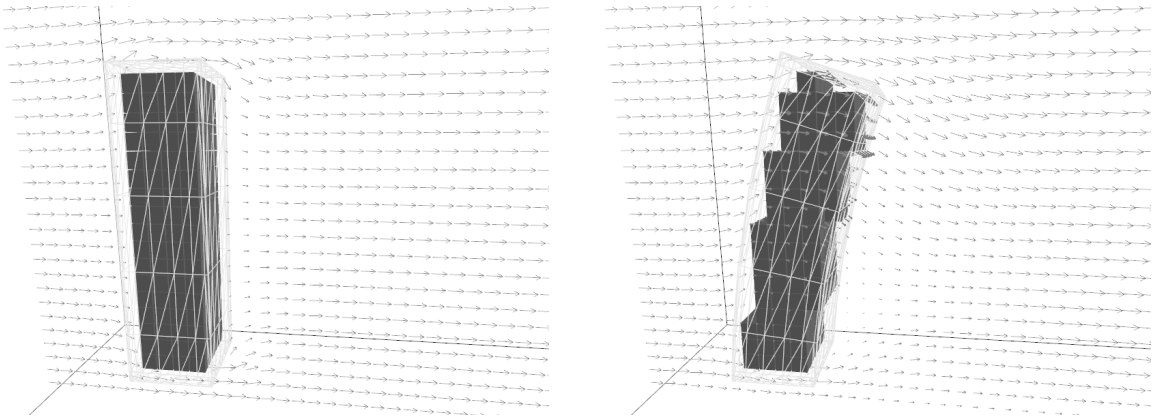
Listing 3: Coupling API integrated into a solver code.

Having done the implementation tasks described here enables the coupling supervisor to completely control the coupled simulation. A user can choose between all the coupling schemes implemented in the coupling supervisor, without any additional changes in the solver code.

4 Numerical Examples

We computed several FSI scenarios with FSI❄ce as coupling tool. To realize the partitioned simulations, we used our fluid solver F3F and the structure solver AdhoC^4 Düster et al. (2004), which is developed at the Chair for Computation in Engineering of the Civil Engineering department at TU München. F3F was initially developed to simulate the phenomena of blister formation in boiling liquids by Emans (2003), but then extended to perform partitioned coupled FSI simulations in Brenk (2007) and Gatzhammer (2008). It is a finite volume based solver for the incompressible Navier-Stokes equations and works with Cartesian grids. Geometies are represented by the marker-and-cell approach introduced in Harlow and Welch (1965). The Cartesian grids in combination with the marker-and-cell approach are perfectly suited for fixed grid approaches to FSI problems, and allow arbitrarily large deformations with even topology changes.

Figure 12 shows visualizations of the flow field with the geometry discretised on the Cartesian grid and by the surrounding coupling mesh.



(a) (b)

Figure 12: FSI coupling test case of a bending tower in a channel flow. Snapshot (a) is taken at the beginning of the simulation, while snapshot (b) is taken after the tower has started to bend in flow direction. In (b), the velocities at the Cartesian mesh nodes are visualized by vectors.

We tested the implicit coupling scheme with a channel flow driving the movement of a spherical obstacle inserted into the flow. The resulting forces in flow direction are shown in Figure 13. The jumps in the force values (marked

by arrows) are due to the discretization of the obstacle on the Cartesian grid, which allows for step-wise movements only. The resolution of the grid chosen was rather low. Hence, there are significant oscillations. However, it is visible that these jumps lead to oscillations for the explicit scheme especially in the right end of the figure, and the discrete movements of the obstacle occur at different times, since the jumps are not synchronized for explicit and implicit methods.

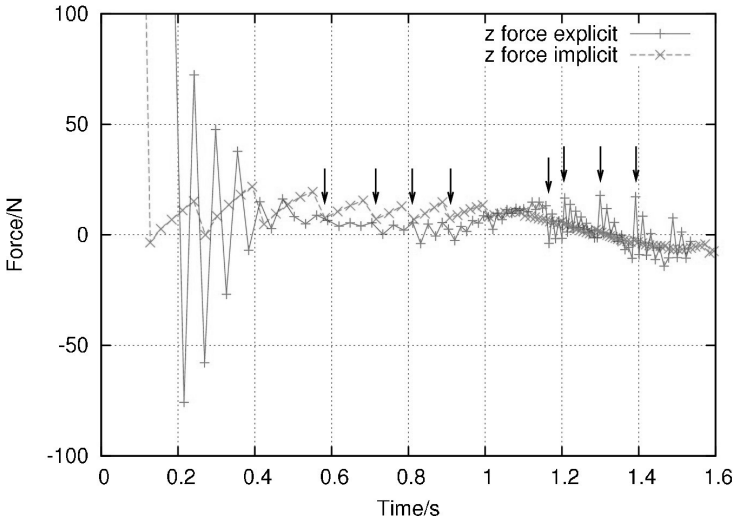


Figure 13: Comparison of the forces acting on a free moving sphere in a channel flow computed by an implicit and an explixit coupling scheme.

Finally, we set up a three-dimensional scenario for quantitative comparisons with the benchmark scenario as proposed by Hron and Turek (2006). The preparations to compute this scenario are currently still work in progress. Thus, we only show a screenshot of the setup of the benchmark in Figure 14. The fixed cylinder obstacle is modeled by the fluid solver internally, while the deformable cantilever attached to it is coupled to the structure solver AdhoC. In the figure, the coupling mesh is displayed as overlay on the cantilever.

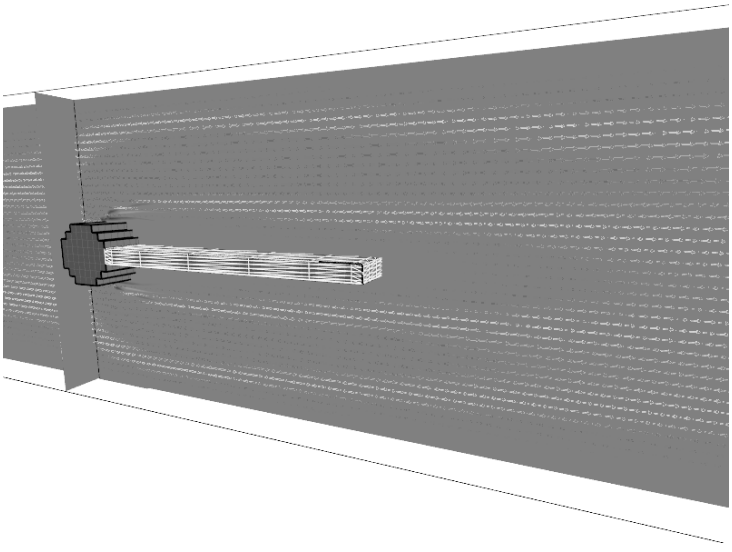


Figure 14: Setup of an FSI benchmarking scenario with fixed cylinder and deformable and coupled cantilever attached to it.

5 Outlook

Currently, we improve the data mapping from the solver grids to the coupling mesh, with the goal to completely encapsulate the data mapping functionality into FSI❄ce. The task is a library-like mapping component, where the user can choose the mapping algorithm and only needs to implement some API calls into his solver codes. This

approach does not prevent the user from implementing specialized mapping strategies, to pertain the error order of high order mesh elements, for example. What changes, is the place the user has to implement this functionality, which is shifted from the solver to the coupling tool. There, a component based strategy is chosen, which allows other users to reuse implemented mapping schemes and, thus, further enhances the value of FSI❄ce.

The component based direction is also in the focus of our work on other parts of FSI❄ce such as the data structure used to exchange data between the coupled solvers, which must not be limited to a triangulated surface mesh. In particular, we are working on a generalization of our coupling tool to arbitrary multi-physics problems, which is a natural step in the sense that the functionality FSI❄ce provides should not only be solver independent, but also independent of the characteristics of the physical domains considered.

A topic will to address in the near future is the parallelisation of the data mapping between solver and coupling mesh. Due to the encapsulation of the data mapping functionality into FSI❄ce, this task can be completely taken from the users of FSI❄ce. Hence, a simulation program does not have to take care whether the access to the coupling data structure is parallel or not.

Future goals are the implemention of multigrid schemes with coupling on all grid levels and the realisation of grid based partitioned FSI simulations. These capabilities are necessary to simulate very large FSI problems as occurring in the simulation of very large floating structures, for example.

6 Summary

In Section 2, we showed the motivation to develop a new coupling tool, which is the lack of a well designed coupling tool exploiting the inherent flexibility of partitioned coupled simulations. Thanks to the client-server concept for coupling the solvers and an additional mesh instance representing the coupling surface, the solvers can be hidden from each other and, as a consequence, can be kept free of any direct dependency on each other. The localisation of all coupling related functionalities into an additional software unit, the coupling supervisor, minimises the implementation efforts on the user side, since it allows to reuse all functionality such as coupling schemes for all solvers coupled with the help of FSI❄ce. The tasks remaining for the user of FSI❄ce are the implementation of the calls to the coupling API prescribed by FSI❄ce and an implementation of the mapping to and from the coupling mesh. The latter one can be simplified a lot by using the supplied neighbour search functionality, which delivers the closest neighbouring points and triangles on the coupling surface efficiently due to the octree acceleration.

Furthermore, we presented three examples for advanced coupling schemes in Section 2, which are the method of Localized Langrangian Multipliers, an implicit coupling scheme with reduced order models, and a multigrid scheme coupling all mesh levels together. We showed that, due to our coupling concept, the efforts undertaken by a user to employ one of those coupling schemes are kept minimal, provided the coupling schemes are supported by the external coupling unit. A comparison of different coupling methods is then relatively easy to achieve and allows to gain further flexibility and adaptability to the specifics of a coupled problem.

In Section 3, we showed the current structure and functionality of our coupling tool. We explained the structure of the coupling mesh employed to transport interface data between the solvers. Crucial for a user is the support for data mapping, which is given in the form of neighbour search functionality accelerated by an octree data structure. We also presented the coupling schemes supported by FSI❄ce, which are an explicit and an implicit coupling scheme with interface iterations. In addition, subcycling and pre-computations can be applied, which give further flexibility to realise FSI simulations.

Finally, in Section 4, we gave some examples of test scenarios computed with our coupling tool. Further validation is necessary, especially a benchmark scenario allowing to perform quantitative comparisons of results in order to fully prove the capabilities of our coupling tool.

Acknowledgements

This work was supported by the *German Research Foundation*, research unit 493, and the International Graduate School of Science and Engineering at the Technische Universität München, project 3.10. This support is thankfully acknowledged.

References

Ahrem, R.; Post, P.; Steckel, B.; Wolf, K.: MpCCI: A tool for coupling CFD with other disciplines. In: *Proceedings of the 5^{th} World Conference in Applied Fluid Dynamics, CFD - Efficiency and the Economic Benefit in Manufacturing* (2001).

Bijl, H.; van Zuijlen, A. H.; Bosscher, S.: Two level algorithms for partitioned fluid-structure interaction computations. In: P. Wesseling; E. Oñate; J. Périaux, eds., *ECCOMAS CFD 2006, European Conference on Computational Fluid Dynamics*, TU Delft (2006).

Brenk, M.: *Algorithmic Aspects of Fluid-Structure Interactions on Cartesian Grids (German: Algorithmische Aspekte der Fluid-Struktur-Wechselwirkung auf kartesischen Gittern)*. Ph.D. thesis, TU München (2007).

Brenk, M.; Bungartz, H.-J.; Daubner, K.; Mehl, M.; Muntean, I.; Neckel, T.: An Eulerian approach for partitioned fluid-structure simulations on Cartesian grids. *Computational Mechanics*, accepted.

Brenk, M.; Bungartz, H.-J.; Mehl, M.; Mundani, R.-P.; Düster, A.; Scholz, D.: Efficient interface treatment for fluid-structure interaction on cartesian grids. In: *ECCOMAS COUPLED PROBLEMS 2005, Proc. of the Thematic Conf. on Computational Methods for Coupled Problems in Science and Engineering*, International Center for Numerical Methods in Engineering (CIMNE) (2005).

Brenk, M.; Bungartz, H.-J.; Mehl, M.; Neckel, T.: Fluid-structure interaction on cartesian grids: Flow simulation and coupling environment. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction*, no. 53 in LNCSE, pages 233–269, Springer (2006).

Brenk, M.; Bungartz, H.-J.; Muntean, I. L.; Neckel, T.: Simulating large particle movements in drift ratchets using cartesian grids. In: P. Wesseling; E. Oñate; J. Périaux, eds., *ECCOMAS COUPLED PROBLEMS 2007, Proc. of the Thematic Conf. on Computational Methods for Coupled Problems in Science and Engineering*, pages 397–399, International Center for Numerical Methods in Engineering (CIMNE) (2007).

Daubner, K.: *Data Exchange and Geometry Treatment for the Simulation of Fluid-Structure Interactions with Partitioned Aproaches (german: Datenaustausch und Geometriebehandlung bei der Simulation von Fluid-Struktur-Wechselwirkungen mit partitionierten Ansätzen)*. Diploma thesis, Institut für Informatik, TU München (2005).

Düster, A.; Bröker, H.; Heidkamp, H.; Heißerer, U.; Kollmannsberger, S.; Krause, R.; A.Muthler; Niggl, A.; Nübel, V.; Rücker, M.; Scholz, D.: *AdhoC^4 – User's Guide*. Lehrstuhl für Bauinformatik, TU München (2004).

Emans, M.: *Numerische Simulation des unterkühlten Blasensiedens in turbulenter Strömung: Ein Euler-Lagrange-Verfahren auf orthogonalen Gittern*. Ph.D. thesis, Institut für Informatik, TU München (2003).

Gatzhammer, B.: *A Partitioned Approach for Fluid-Structure Interaction on Cartesian Grids*. Master's thesis, Fakultät für Informatik, Technische Universität München (Aug. 2008).

Harlow, F. H.; Welch, J. E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Physics of fluids*, 8, 12, (1965), 2182–2189.

Hron, J.; Turek, S.: Proposal for numerical benchmarking of fluid-structure interaction between elastic object and laminar incompressible flow. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction*, no. 53 in LNCSE, pages 371–385, Springer (2006).

Larson, J.; Jacob, R.; Ong, E.: The model coupling toolkit: A new fortran90 toolkit for building multiphysics parallel coupled models. *Int. J. High Perf. Comp. App.*, 19, 3, (2005), 277–292.

Park, K. C.; Felippa, C. A.; Ohayon, R.: Partitioned formulation of internal fluid-structure interaction problems by localized Lagrange multipliers. *Comput. Methods Appl. Mech. Engrg.*, 190, (2001), 2989–3007.

Parker, S. G.; und T. Harman, J. G.: A component-based parallel infrastructure for the simulation of fluid-structure interaction. *Engineering with Computers*, 22, 3–4, (2006), 277–292.

Ross, M.: *Coupling and Simulation of Acoustic Fluid-Structure Interaction Systems Using Localized Lagrange Multipliers*. Ph.D. thesis, University of Colorado (2006).

Vierendeels, J.: Implicit coupling of partitioned fluid-structure interaction solvers using reduced-order models. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction*, no. 53 in LNCSE, pages 1–18, Springer (2006a).

Vierendeels, J. A.: Strong coupling of partitioned fluid-structure interaction problems with reduced order models. In: P. Wesseling; E. Oñate; J. Périaux, eds., *ECCOMAS CFD 2006, European Conference on Computational Fluid Dynamics*, TU Delft (2006b).

Address: Dr. M. Mehl (corresponding author), Department of Computer Science, Technische Universität München, Boltzmannstr. 3, 85748 Garching.
email: mehl@in.tum.de

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 131– 146
Herrsching am Ammersee, 29.9.-1.10.2008

Reference Test Cases for Fluid-Structure Interaction Studies

J. Pereira Gomes, H. Lienhart

The swiveling motion of a flexible structure immersed in a flowing fluid can become self-excited as a result of different fluid-structure interaction mechanisms. The present study aimed to identify and decouple the different fluid-structure interaction self-exciting mechanisms and to understand the influence of the physical parameters on the different exciting processes. The parameters involved the fluid viscosity, the incoming fluid velocity and the geometric and mechanical properties of the structure. To achieve this goal, the two-dimensional self-excited periodic swiveling motion of flexible structures was investigated in both laminar and turbulent uniform flows. The results obtained for a specific structure model are presented. It consisted of a $0,04$ *mm thick stainless-steel membrane attached to a cylindrical front body. At the trailing edge of the flexible membrane, a rectangular mass was considered. The entire structure model was free to rotate around an axle located in the central point of the front body. During the experimental investigation, the general character of the dynamic response of the structure model was first investigated as a function of the incoming flow velocity. The tests in laminar flows were performed in a polyglycol syrup (dynamic viscosity:* $1,64 \times 10^{-4}$ *m²/s) to maintain the Reynolds number smaller than* 270*. The tests in turbulent flows were conducted in water for Reynolds numbers up to* 44000*. In both cases, the maximum incoming velocity tested was* 2 *m/s. Subsequently, the relevant fluid-structure self-exciting swiveling modes were characterized in more detail as far as the flow velocity field and structure deflection modes are concerned. Thus, detailed measurements are presented at* $1,07$ *m/s and at* $1,45$ *m/s in laminar and at* $0,68$ *m/s in turbulent flows. The measurements were performed using a particle image velocimetry (PIV) system complemented by a time-phase detector to obtain accurate time-phase resolved measurements and aimed at the flow velocity and also the structure deflection. In this way, it was possible to characterize the flow velocity field and also the deflection of the structure over an entire period of the motion.*

1 Introduction

The mechanisms which lead the vibration of a flexible structure immersed in a flowing fluid to become self-excited are very difficult to predict and, at the same time, very sensitive to the mechanical properties of the structure and to the properties of the incoming flow. If a structure swivels in a flowing fluid, the added damping can become negative as a result of different mechanisms by which energy is transferred from the flow to the body. In such cases, the overall damping of the structure-fluid system decreases and, in some cases, completely disappears, resulting in a periodic coupled unsteady movement of the flow and structure motion. Depending on whether the fluctuation of the flow plays a significant role in the excitation process or not, the excitation is called either flow-induced or movement-induced excitation (MIE). In the case of flow-induced excitation, one can further distinguish between the extraneously-induced and instability-induced excitation (EIE and IIE).

From the numerical point of view, these self-excitation problems have attracted increasing interest in computational mechanics due to their practical relevance in many fields of engineering and in many other sciences (Naudascher and Rockwell (1980), Naudascher and Rockwell (1994); Souli and Hamdouni (2007)). The numerical simulations require coupling algorithms between computational fluid dynamics (CFD) and computational structural dynamics (CSD) which are not yet considered as validated tools (Bungartz and Schäfer (2006)). In this respect, the present investigation aimed to reproduce the self-excited coupled cyclic movement of relatively simple but meaningful two-dimensional structures in uniform flows. The primary objectives of these controlled experiments were to identify, and decouple, the different fluid-structure interaction self-exciting mechanisms and to understand the influence of the physical parameters on the different exciting processes for a relatively simple structure geometry. The parameters involved the incoming fluid characteristics and the geometric and mechanical properties of the structure. The present study also investigated both laminar and turbulent flow regimes, to account for the influence

of the Reynolds number on the self-exciting process. To control the Reynolds number and the flow velocity independently, the viscosity of the test liquid was controlled during the tests. Second, the experimental investigation addressed the need for experimental data on reference test cases. It provided a reliable data base on specific, well-defined reference test cases to be used as a diagnostic and validation tool for numerical models for fluid-structure interaction simulations. The data base created from these reference test cases included the time-phase resolved characterization of the flow velocity field and the mechanical behavior of the structure, such as its deflection, principal deflection modes, periodic motion amplitude and frequency. The present paper reports the results of the experimental investigation performed on one of the investigated structure models in both laminar and turbulent flows.

2 Experimental set-up

2.1 Test definition

The project aimed to characterize the resulting two-dimensional periodic swiveling motion of relatively simple flexible structures driven by uniform incoming laminar and turbulent flows. The project requirements for periodicity and reproducibility of the resulting flow and structure motion imposed stringent restrictions on the selection of models to be considered. Moreover, the set of models to be tested should make it possible to understand the influence of different parameters such as shape, mass and momentum of inertia on the fluid-structure interaction self-excitation mechanisms. Finally, a set of models mainly consisting of a flexible panel attached to a free rotating front solid cylinder was considered. The present paper shows the results obtained for one of that set of selected models (figure 1). It consisted of a $0,04$ mm thick stainless-steel sheet attached to a 22 mm diameter aluminum cylindrical front body. At the trailing edge of the membrane, a 10 mm $\times$ 4 mm rectangular stainless-steel mass was located. All the structure was free to rotate around an axle located in the central point of the front cylinder. Both the front cylinder and the rear mass were considered rigid. The flexible section of the structure proved to show linear mechanical behavior within the range of forces acting on it during the tests and the Youngs modulus was measured as 200 kN/mm^2. The overall spanwise direction of the model was chosen to match the dimensions of the test section to guarantee the two-dimensionality of the test case.

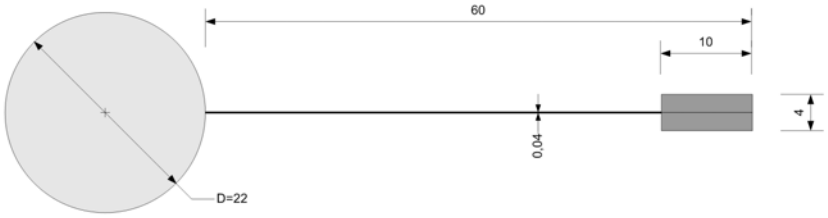


Figure 1: Structure model geometry (all dimensions in mm).

The tests were conducted in a vertical, closed-loop tunnel capable of operating with different viscous liquids with a kinematic viscosity up to 5×10^{-4} m^2/s. The special capability of the tunnel to operate with different viscous liquids allowed control of the Reynolds number of the tests for the same inlet flow velocity. For the investigation in the laminar regime, polyethylene glycol (polyglycol) PG-12000 syrup was used as the test fluid. Because the physical properties of this kind of syrups are sensitive to the liquid temperature, this value had to be controlled during the tests. The tests were finally conducted for a controlled temperature of 25 °C with an uncertainty of $0,5$ °C. Within this temperature range, the mixture could be considered incompressible and Newtonian; its kinematic viscosity and density were measured to be constant at $1,64 \times 10^{-4}$ m^2/s and 1050 kg/m^3, respectively. On the other hand, the tests in turbulent flows were conducted in water at a temperature of 22 °C $^{+0°C}_{-1°C}$ ($\rho_w = 998$ kg/m^3; $\nu_w = 0,97 \times 10^{-6}$ m^2/s). For both the laminar and turbulent regimes, the test were performed for a flow velocity range up to 2 m/s.

The test section has an overall length of 338 mm and a cross-sectional area of 180 mm $\times$ 240 mm. The structure was mounted 55 mm downstream of the inlet plane of the test section. Opting for a vertical tunnel, the gravity force was aligned with the x-axis and so it did not introduce any asymmetry. The experiment domain of the tests is represented in figure 2. Special attention was given to the model support on the test section. Low-friction bearings were used for this specific task to guarantee a frictionless rotational degree of freedom of the front cylinder. The

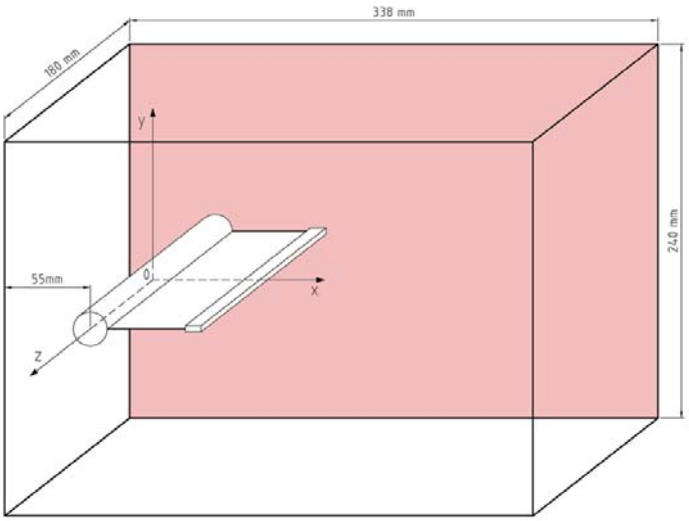


Figure 2: Experimental domain.

supporting system also considered a non-contacting magnetic position sensor to measure the angular position of the structure front cylinder. The output of this sensor was used during the test to measure the time-dependent angle of the structure front body and to determine the beginning of a new model swiveling motion period.

2.2 Measuring apparatus

The specific requirements of the present project imposed several challenges when it came to acquiring the experimental data and to reconstruct them. The first difficulty was that measurements were performed in different liquids with a wide range of viscosity. The others were related to the nature of the resulting movement observed in periodic fluid-structure interaction problems. First, the periodicity of the structure motion is sensitive to the flow conditions and structure mechanical properties; therefore, there are cycle-to-cycle variations of the period time. Second, the velocity of the structure within a motion period is not predefined (as it is for crank shaft-driven set-ups), which makes it impossible to reconstruct the time-phase resolved data from position-resolved measurements. To overcome those problems, a particle image velocimetry (PIV) system connected to an in-house designed time-phase detector was implemented to obtain accurate time-phase resolved measurements of the resulting coupled structure and flow motion in a wide range of viscous liquids. The time-phase detector was developed based on a 1 MHz absolute clock to provide a detection accuracy of 2 μs. During the tests, two signals were detected and recorded: the internal triggering signal of the PIV cameras to indicate the instant of the measurement and the signal of a magnetic angular position sensor connected to the structure to indicate the beginning of a new swiveling cycle. Thus, the measurements were reconstructed introducing the time-phase angle $tpa = (t/T) \times 360\,^\circ$ where T is the period of the structure swiveling motion and t is the delay of the measurements with respect to the beginning of the cycle. The measurements were resolved in the time-phase space with a resolution of $2,5\,^\circ$ within an angle averaging slot of $\pm 0,5^\circ$. The task of measuring the two-dimensional flow field around the model was performed using a PIV system that consisted of two 1280 pixel $\times 1024$ pixel synchronized cameras and a double-head pulsed Nd:YAG laser with a wavelength of 532 nm. Opting for the solution of two parallel cameras, it was possible to cover an extended 272 mm $\times 170$ mm flow field area while keeping the spatial resolution as low as 133 μm $\times 133$ μm per CCD pixel. The measurement location was set to the center plane of the test section ($z = 0$ mm; see figure 2). Two laser sources were used to illuminate the flow. This solution was adopted because the flexible structure was an opaque body which creates an unsteady dark shadow region when illuminated by just one light source. This behavior not only reduced the measuring area to almost one side of the flexible structure, but also made the masking of the PIV images in post-processing difficult to perform. Using one laser source to illuminate each side of the structure, the dark region behind the structure was extinguished and all the flow surrounding the structure was accessible to PIV measurements. As seeding particles, 10 μ m mean diameter hollow glass spheres were chosen to be used in water. They provided a good match of density and enough scattering signal over the all the measuring area. During the laminar tests, in polyglycol syrups, silver-coated hollow glass spheres with the same diameter were adopted as seeding particles. They produce higher signal levels in high light-absorbing media compared with non-coated hollow glass spheres. The major drawback of the silver-coated glass spheres is related to their density; the relative density of this kind of particles is about $1,4$. Nevertheless, this drawback was acceptable because of the

high viscosity of the fluid and the velocity of the flow during the tests. To determine the position of the structure, the PIV system was modified to provide it with structure deflection analysis capabilities. The idea behind this set-up was to use the PIV system to acquire images from the swiveling structure and to use specially developed software to analyze and reconstruct the time-dependent deflection of the structure. The major advantage of this approach was that the same measuring system as used for the velocity field measurements could be employed. The cameras were now located in such a way as to acquire images of the flexible structure illuminated by the laser sheet from each side of the model. No seeding was used during these tests. Quantitative analysis was performed after image acquisition in Matlab workspace by a script developed for the specific task. The software analyzed and compared the PIV images of both sides of the model and reconstructed the time-dependent image of the light sheet reflected by the structure. For that purpose, it mapped the pixel value in the grayscale of the entire image and detected the line resulting from the intersection of the laser sheet and the structure and the edges of the rear mass. With the information on the position of the membrane and on the time-phase detector module, the algorithm finally computed all the relevant time-phase resolved data of the structure movement, such as time-phase resolved angle of attack of the front body, structure deformation shape and coordinates of the structure trailing edge. Based on these data, the modes present in the structure were identified and characterized. More detailed descriptions of the measurement techniques and their adaptation to the present task are given elsewhere (Gomes and Lienhart (2006a); Gomes and Lienhart (2006b)).

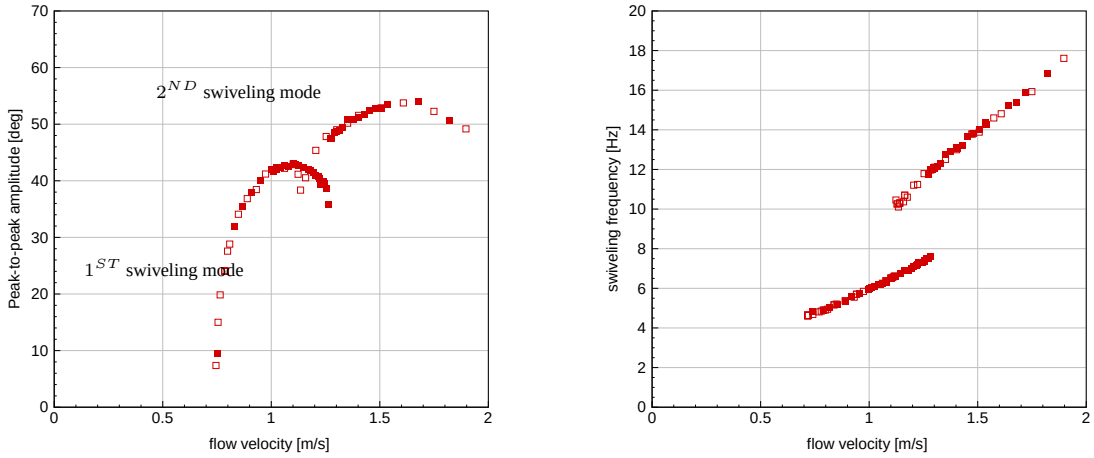
3 Results

To assist the understanding of the results, they are divided according to the flow regime. Thus, in section 3.1 the results obtained in the laminar regime are presented and in section 3.2 the results in turbulent flows. For each regime, the structure model was first tested at different incoming flow velocities up to 2 m/s. These results defined the general character of the dynamic response of the structure model as a function of the incoming flow velocity and showed the different swiveling modes exhibited by the structure. In the second stage, detailed measurements were conducted at selected velocities to characterize each combined flow and structure swiveling mode, and section 3.1 includes detailed measurements obtained in laminar flow at $1,07$ m/s and $1,45$ m/s. In turbulent flows, detailed measurements were performed only for one approaching flow velocity, $0,68$ m/s. These measurements are presented in section 3.2.

3.1 Results in laminar flows

Figure 3 shows the dynamic response of the structure in laminar flows. The Reynolds number, based on the diameter of the front cylindrical body, reached the maximum value of 270 at 2 m/s. At very low flow velocities, it was not possible to identify any kind of motion. On increasing the flow speed, it was observed that the minimum velocity needed to excite the movement of the structure varied slightly from test to test. Nevertheless, in all cases it was possible to achieve a periodic cyclic swiveling motion for velocities slightly smaller then 1 m/s. It should be mentioned that as soon as the structure started to swivel, its motion frequency coincided with the line in figure 3 independently of the velocity value at which the movement started. From the instant that the structure started to swivel and for all the range of velocities tested, the resulting motion proved to be symmetric and periodic. The RMS value of the cycle-to-cycle structure motion period was measured to be less than 1%.

The most obvious aspect revealed by figure 3 is the existence of two distinctive structure swiveling motion modes separated by a pronounced, well-defined hysteretic region. For both swiveling modes, the frequency of the resulting motion increased linearly with the velocity of the incoming flow. Whereas the frequency increased with the flow velocity, the amplitude of the structure motion showed a maximum value for each swiveling motion mode. The first swiveling mode, registered for incoming flow velocities up to approximately $1,1$ m/s, was characterized by the fact that the deflection of the structure model was strongly governed by the first bending mode of the membrane. In connection with this, the movement of the rear mass was in concordance with the movement of the front body. The second swiveling mode, observed for incoming velocities higher than $1,3$ m/s, was characterized by the fact that the rear mass motion is in opposition to the movement of the structure front body. At the same time, higher bending modes were present in the deflection of the structure. The conclusion about the deflection modes exhibited by the structure was supported by visualizations performed during the tests at different flow speeds. Within the transition region, from $1,1$ m/s to $1,3$ m/s, the structure presented a hysteretic behavior where both swiveling modes could be observed depending on the previous frequency of the structure. In both cases, a delay of the movement of the rear mass in relation to the front body was identified. This delay is a consequence of the flexibility of the membrane and it is a function of the mechanical properties of the structure. After identifying the two different



(a) (b)

Figure 3: Structure swiveling frequency (a) and structure front body swiveling peak-to-peak amplitude (b) versus incoming flow velocity (solid squares correspond to measurements acquired while increasing and open squares while decreasing the flow velocity).

self-excited swiveling modes of the flow-structure system, each was characterized in more detail. To this end, two flow velocities, $1,07$ m/s and $1,45$ m/s, were selected as representative of the swiveling modes. Both values are located close to the velocity of the maximum structure amplitude excitation (figure 3(a)).

Results at 1,07 m/s

In the following figures, the characterization of the structure movement for an incoming flow velocity of $1,07$ m/s is shown. At this velocity, the Reynolds number is approximately 140, based on a front body diameter of 22 mm and a kinematic viscosity of the polyglycol syrup of to $1,64 \times 10^{-4}$ m^2/s.

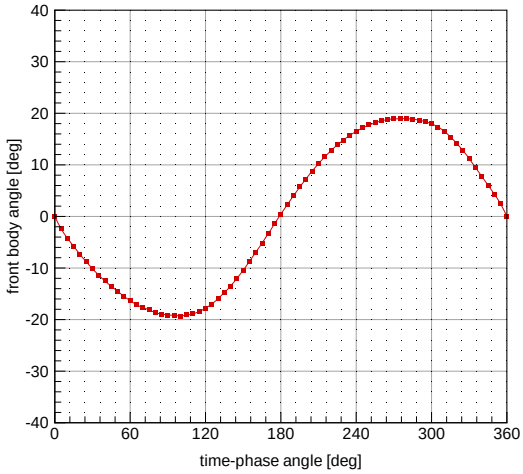


Figure 4: Time-phase resolved front body angle within a period of motion at $1,07$ m/s ($Re \approx 140$).

Figure 4 shows the evolution of the angle of the structure front body within the swiveling motion averaged period and figures 5 and 6 present the successive positions of the structure membrane and the coordinates of the trailing edge during the same period. The time-phase resolution in figure 5 was set to $30\,^\circ$ whereas that used for figure 6 was $5°$. As far as the flow field surrounding of the structure model is concerned, figure 7 compiles the time-

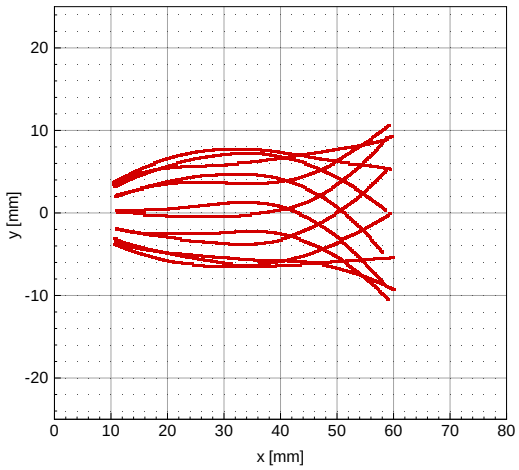


Figure 5: Membrane deflection within a period of motion at $1,07$ m/s ($Re \approx 140$).

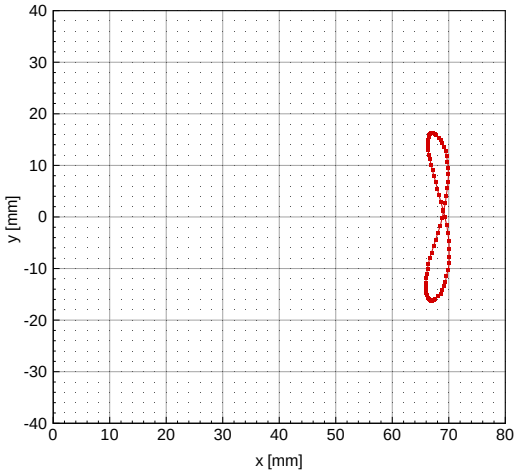
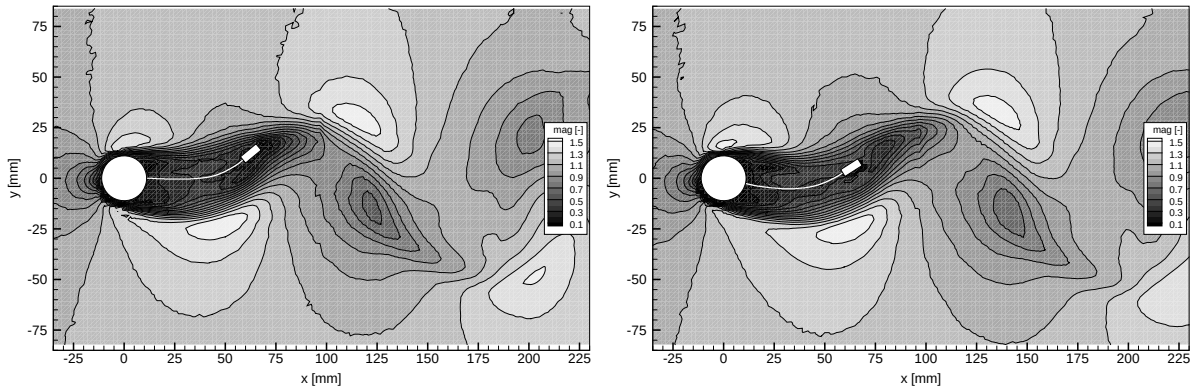


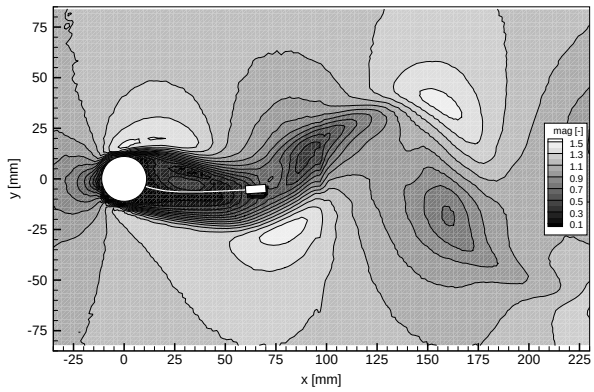
Figure 6: Trailing edge coordinates within a period of motion at $1,07$ m/s ($Re \approx 140$).

phase resolved combined flow field and structure deflection at eight instants of the reference swiveling period. The successive results indicate that the movement of the trailing edge is in-phase, but delayed, with respect to the movement of the front body. This delay could be quantified by comparing the time-phase resolved angle of the front body with the y-coordinate of the structure trailing edge. Thus, at $1,07$ m/s the delay of the trailing edge with respect to the front body movement, as time-phase angle, was computed to be approximately $60\,^{\circ}$.

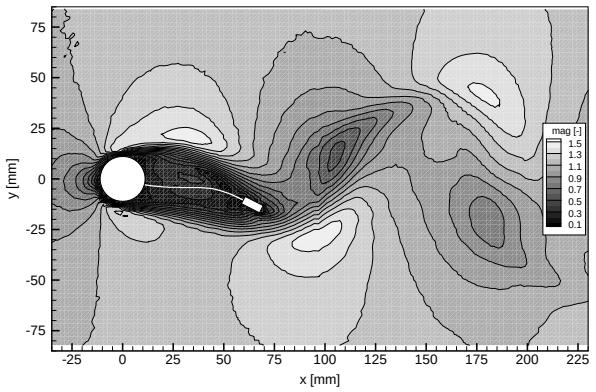


(a) tpa=0°

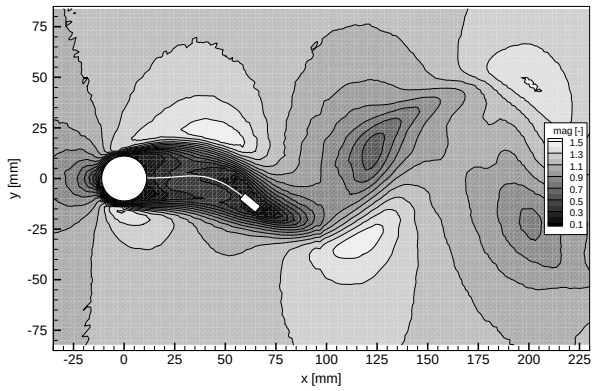
(b) tpa=45°



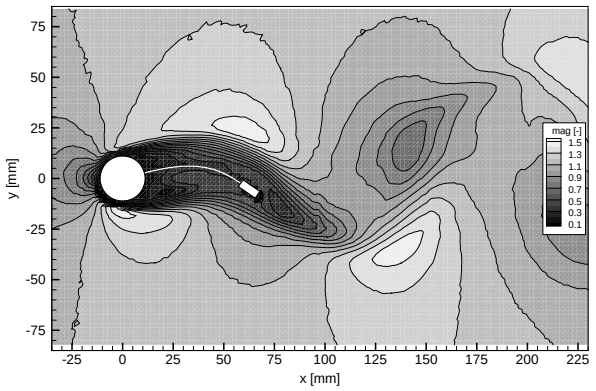
(c) tpa=90°



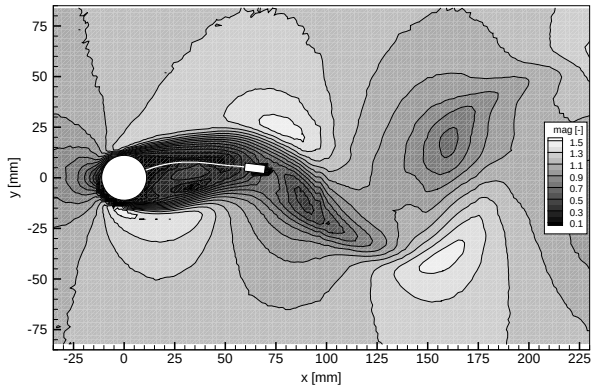
(d) tpa=135°



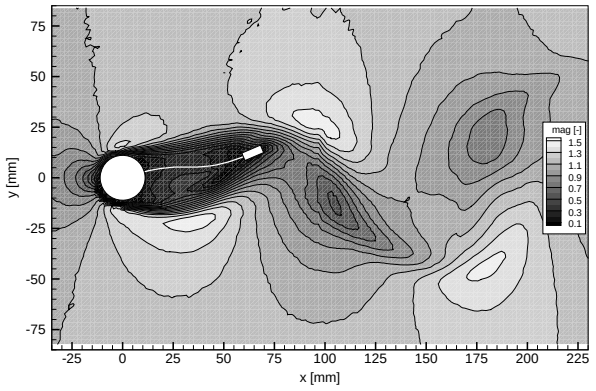
(e) tpa=180°



(f) tpa=225°



(g) tpa=270°



(h) tpa=315°

Figure 7: Time-phase resolved combined flow field/structure deflection measurements at eight different instants of the swiveling motion period at $1,07$ m/s ($Re \approx 140$).

Results at 1,45 m/s

A similar set of measurements were performed for an incoming flow velocity of $1,45$ m/s, which corresponds to a Reynolds number close to 190. At $1,45$ m/s, the structure exhibited a more complex and faster swiveling motion. The resulting motion frequency was measured to be $13,58$ Hz and the maximum front body angular amplitude was $\pm 22°$. In figure 8 the angle of the front body within the averaged period of motion is displayed. The time-phase delay of the trailing edge excursion in relation to the front body movement increased to about 210 °.

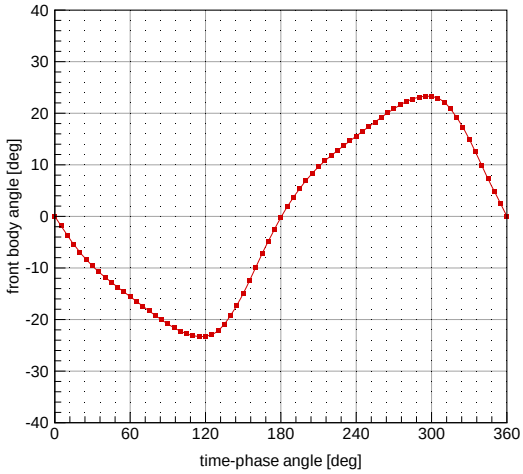


Figure 8: Time-phase resolved front body angle within a period of motion at $1,45$ m/s ($Re \approx 190$).

Figures 9 and 10 show the time-phase resolved position of the structure membrane and of the structure model trailing edge within the averaged swiveling motion. Now the collection of the membrane deformations display a pronounced node, indicating the existence of higher bending modes in the structure deflection. In figures 9 and 10, time-phase angle resolutions of $30\,°$ and $5°$ were used. Regarding the unsteady flow field results, figure 11 presents the velocity field around the structure for eight, $45\,°$ equally distant, time-phase angles within the reference movement cycle.

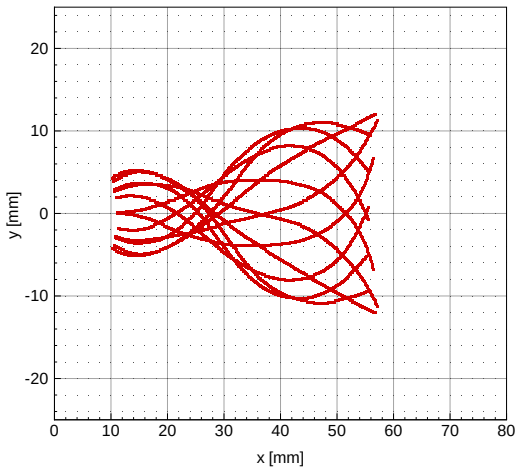


Figure 9: Membrane deflection within a period of motion at $1,45$ m/s ($Re \approx 190$).

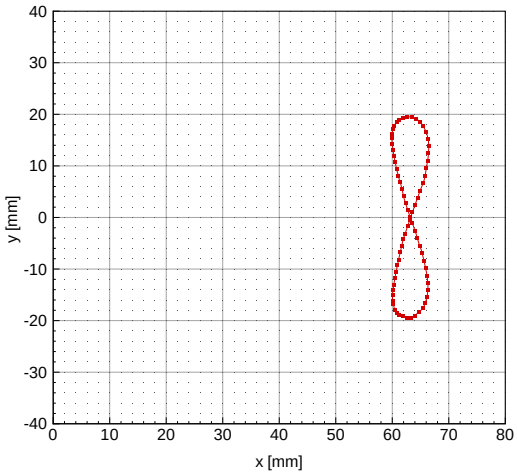
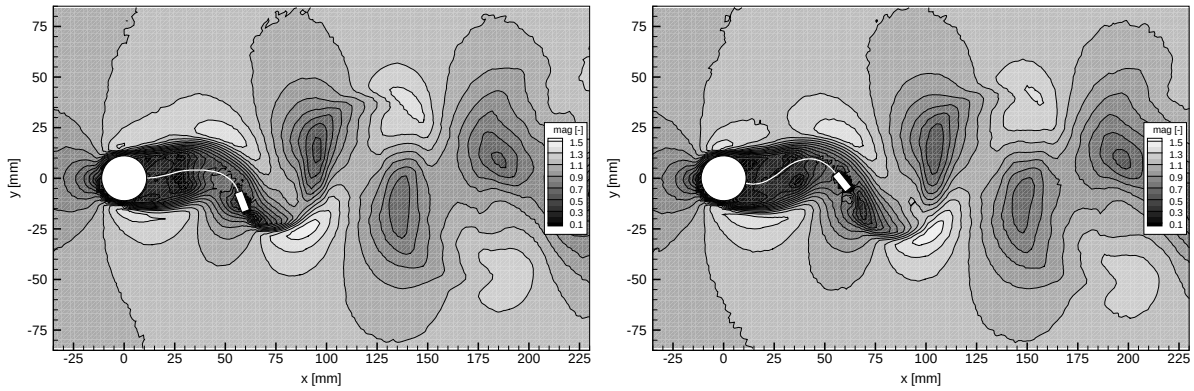
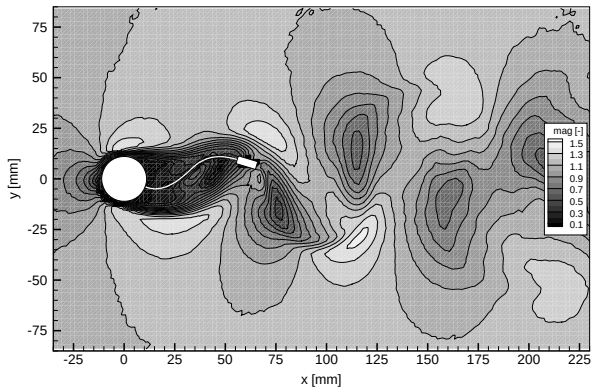


Figure 10: Trailing edge coordinates within a period of motion at $1,45$ m/s ($Re \approx 190$).

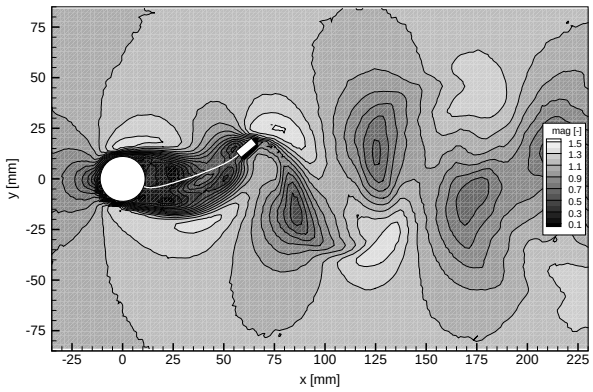


(a) tpa=0°

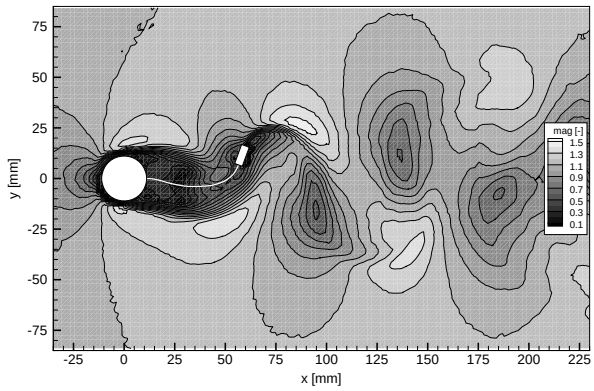
(b) tpa=45°



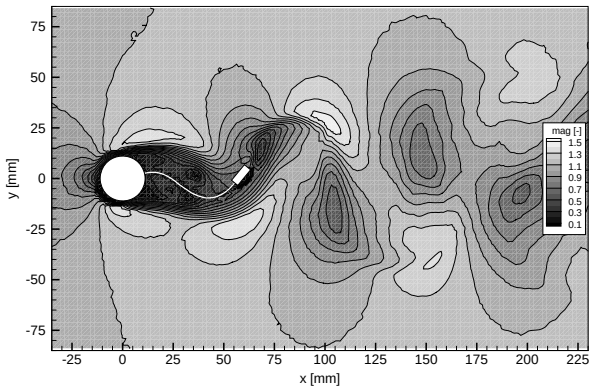
(c) tpa=90°



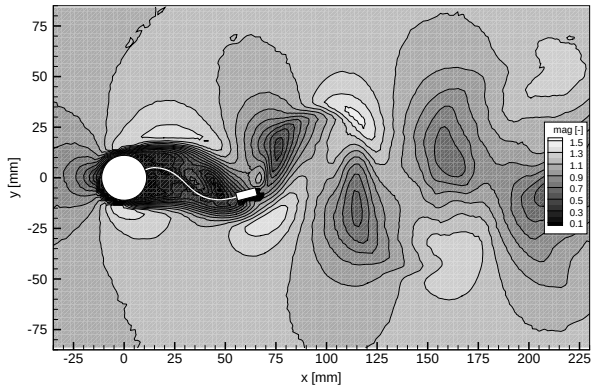
(d) tpa=135°



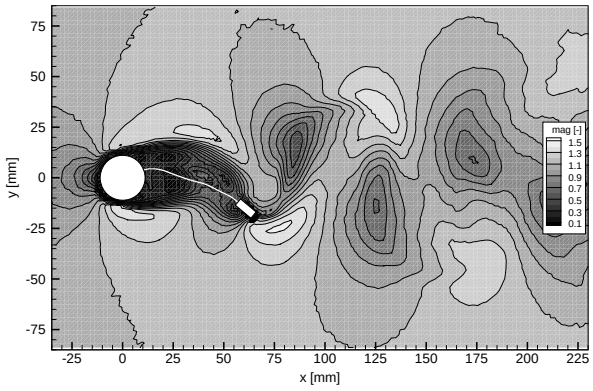
(e) tpa=180°



(f) tpa=225°



(g) tpa=270°

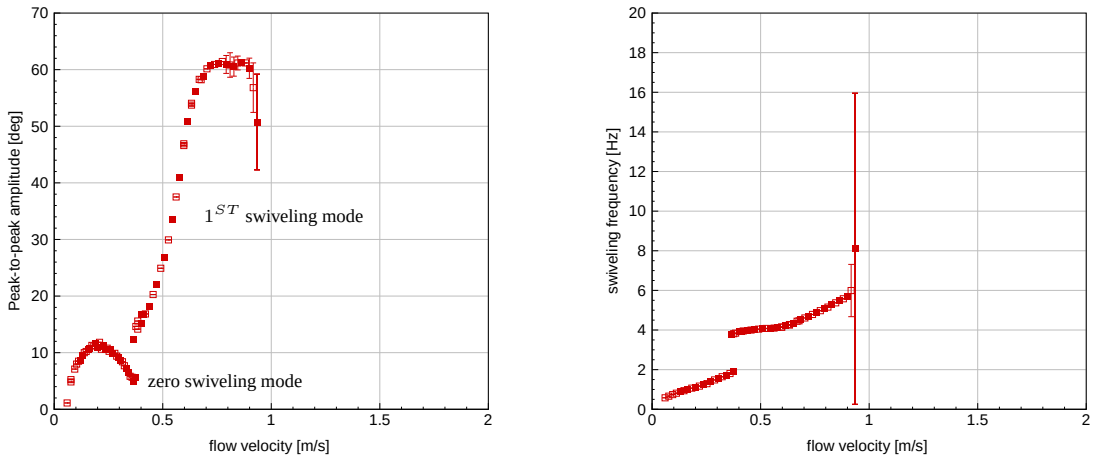


(h) tpa=315°

Figure 11: Time-phase resolved combined flow field/structure deflection measurements at eight different instants of the swiveling motion period at $1,45$ m/s ($Re \approx 190$).

3.2 Results in turbulent flows

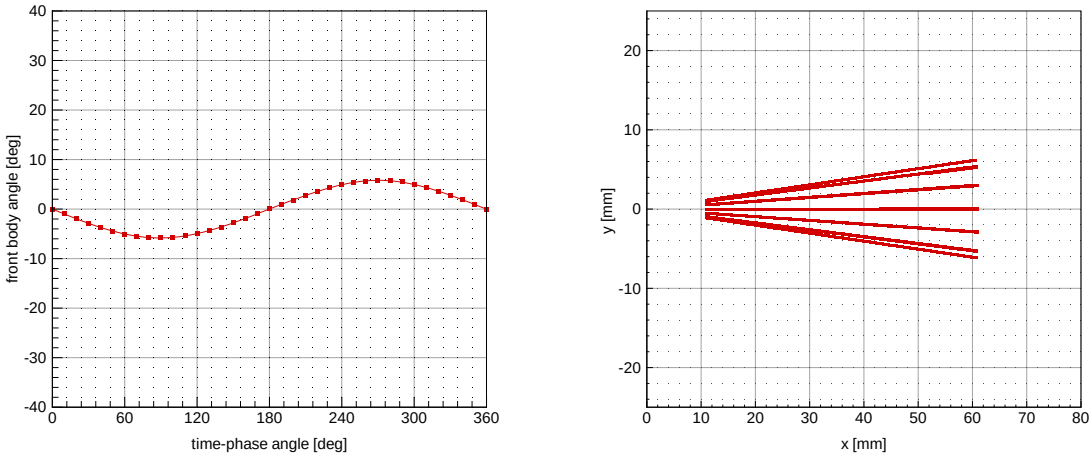
In the turbulent regime, using water as the working fluid, the structure proved to have the same well-defined multi-swiveling mode behavior as observed in laminar flows. Figure 12 presents the dynamic response of the structure versus the incoming water flow velocity up to 2 m/s ($Re \approx 44000$). Now the structure could be excited to a periodic swiveling motion at very low flow velocities. Visualizations showed that this first excited mode corresponded to the rigid body motion mode, i.e. the structure swiveled in the fluid around its free rotating axle without changing its original and straight shape. Because it corresponded to the rigid body motion, this mode was named the zero swiveling mode. As an example of this zero mode, figure 13 shows the structure behavior at approximately $0,19$ m/s in water. A swiveling mode transition was registered for a flow velocity close to $0,4$ m/s. The transition between modes is abrupt and it was not possible to observe either any evolution of the structure motion during the transition or any hysteretic behavior. In the new swiveling mode, the structure deflection was dominated by the first bending mode of the membrane. The behavior of the structure in this mode is all respects similar to the first swiveling mode observed in laminar flows. For both swiveling modes shown in figure 12, the amplitude of the structure movement was limited and exhibited a local maximum value. The structure movement frequency increased approximately linearly with the velocity of the incoming flow in both modes. The only exception occurred in the first mode at approximately 0.6 m/s where a change in slope was registered because of a phase-locked occurrence. On further increasing the incoming flow velocity, an unusual behavior was observed: as soon as the amplitude of the structure started to decrease, after reaching the local maximum, the motion characteristics degraded very rapidly. This effect was supported by the RMS value presented in figure 12. The coupled movement became non-periodic and non-symmetric and led to a rapid destruction of the structure. Therefore, no measurements could be obtained for flow velocities higher than $0,9$ m/s. This sequence of facts occurred when the structure swiveling frequency was showing the first signals of transition to a new, second swiveling mode. In the range in which the structure movement is periodic and reproducible, up to $0,9$ m/s, the RMS value of the cycle-to-cycle motion period remained lower than 1%.



(a) (b)

Figure 12: Structure swiveling frequency (a) and structure front body swiveling peak-to-peak amplitude (b) versus incoming flow velocity (solid squares correspond to measurements acquired while increasing and open squares while decreasing the flow velocity).

Despite the trivial rigid body mode, the only self-exciting swiveling mode that could be characterized in detail using the present structure configuration was the first one. Therefore, further investigations were performed in a water flow at $0,68$ m/s. The decision regarding that velocity followed the same criteria as used for the laminar investigations.



(a) (b)

Figure 13: Time-phase resolved front body angle (a) and membrane deflection (b) within a period of motion at $0,19$ m/s.

Results at 0,68 m/s

At $0,68$ m/s, the Reynolds number of the measurements performed in water was about 15000, based on the diameter of the front cylinder and on the water properties. Under such conditions, the structure exhibited a $4,45$ Hz periodic swiveling motion with the rear mass delayed, in time-phase angle, $95\,^{\circ}$ in relation to the front body. Once again, the characterization of the resulting movement is based on the angle of the structure front body, trailing edge position and structure deflection. Figures 14 to 16 show the time-phase resolved evolution of these quantities within the averaged period of motion. The same resolutions as used to generate the laminar results were used in the following figures.

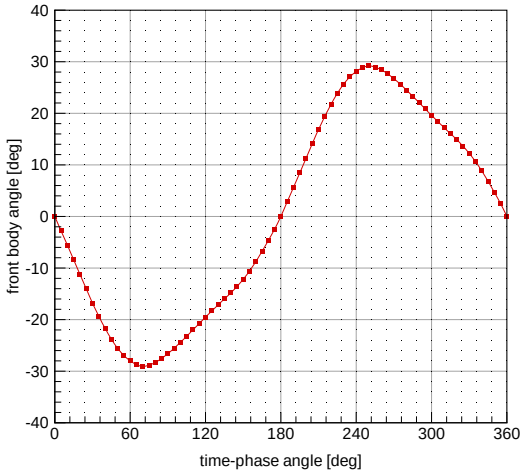


Figure 14: Time-phase resolved front body angle within a period of motion at $0,68$ m/s ($Re \approx 15000$).

Figure 17 shows the flow velocity field results obtained in water at $0,68$ m/s for eight successive time-phase angles measured within a period of the structure motion. The time-phase angles presented correspond to those adopted in section 3.1.

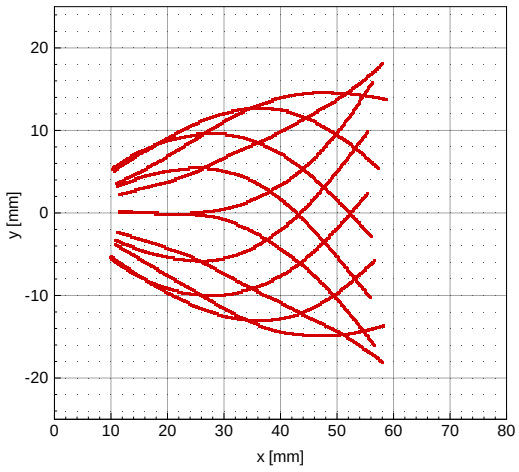


Figure 15: Membrane deflection within a period of motion at $0,68$ m/s ($Re \approx 15000$).

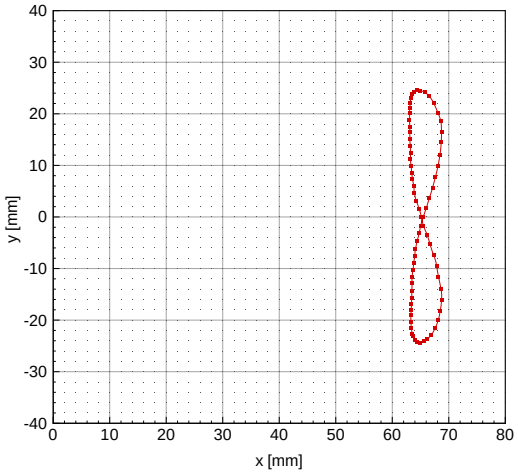


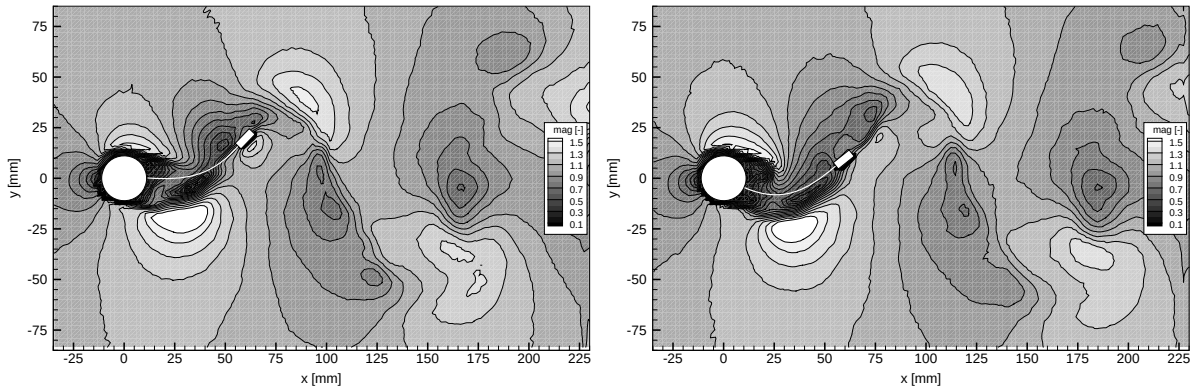
Figure 16: Trailing edge coordinates within a period of motion at $0,68$ m/s ($Re \approx 15000$).

4 Discussion of the results

The self-excited two-dimensional movement of a flexible structure in a uniform flow up to 2 m/s was investigated. The tests included laminar measurements performed in a polyglycol syrup with a kinematic viscosity of $1,64 \times 10^{-4}$ m^2/s and turbulent measurements conducted in water. The structure model consisted of a thin metal sheet attached to an aluminum front cylindrical free rotating body. At the trailing edge of the metal sheet, a rear mass was located. During the tests, this model proved to perform a very reproducible and symmetric two-dimensional motion within the flow velocity range tested. The maximum RMS value of the cycle-to-cycle fluctuation of the structure movement period was measured to be less than 1%.

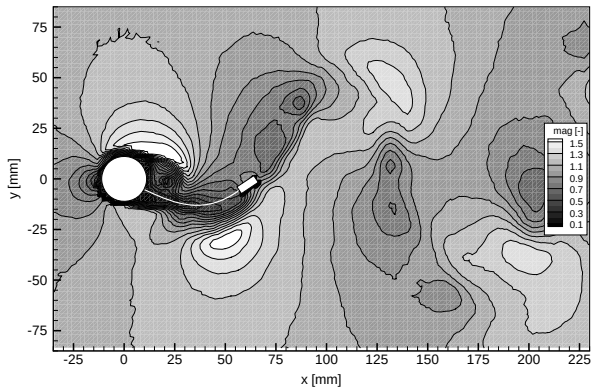
The analysis of the structure model dynamic response proved the existence of different structure swiveling modes. In the laminar regime, it was possible to observe two swelling modes depending on the approaching flow velocity. For both modes, the structure movement frequency increased linearly whereas the front body amplitude presented a local maximum. The maximum excitation occurred for $1,1$ m/s and $1,6$ m/s for the first and second excitation mode, respectively. The corresponding movement frequencies for these two instants of maximum excitation were measured to be around $6,5$ Hz and 15 Hz. The first mode was excited for the first time at $0,8$ m/s. For this flow velocity, the corresponding Strouhal number ($St \approx 0,175$) and the first natural frequency of the structure ($N_1 = 5,9$ Hz) showed a strong interconnection between the movement excitation and the classical von Karman vortex shedding triggered by the structure front cylinder. Despite a small delay, the trailing edge movement could be considered in phase with the angular movement of the front body. Concerning the deflection of the structure, this mode was characterized almost exclusively by the existence of the first bending mode. At $1,07$ m/s, the structure vibrated around its first natural frequency; more precisely, the coupled fluid and structure unsteady motion was registered to occur at $6,38$ Hz associated with a maximum excursion of the front body and trailing edge of 19 ° and 16 mm, respectively.

The transition to the second, more complex mode was observed between $1,12$ m/s and $1,3$ m/s and it showed a strong hysteretic behavior. A similar behavior has already been reported by Parkinson and Smith (1964) and Novak (1972) for square cross-section prisms in cross flow. The second swiveling mode was characterized by a vortex shedding frequency much lower than the second natural frequency of the structure ($N_2 = 27,4$ Hz), indicating a self-exciting mechanism of a different type than the first one. The trailing edge was now almost in phase opposition in relation to the front body position and the structure deflection was mainly characterized by the second structure bending mode and higher. The presence of the second bending mode justified the pronounced node observed in the structure deformation within a movement period. At $1,45$ m/s, the front cylinder reached a maximum deflection of $26°$ and the trailing edge excursion was limited to 19 mm.

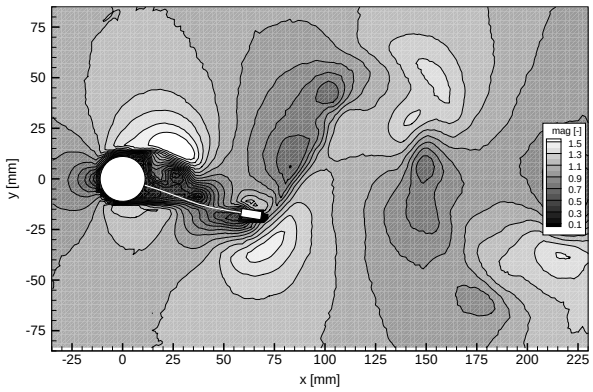


(a) tpa=0°

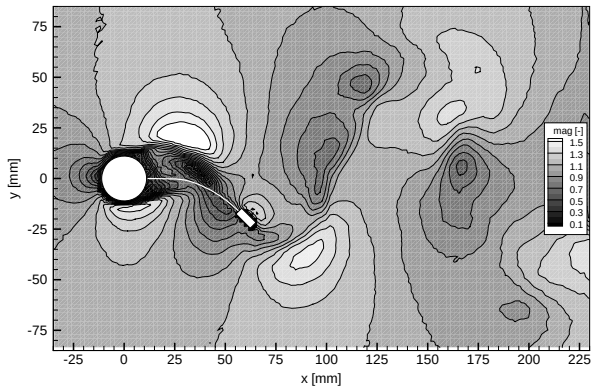
(b) tpa=45°



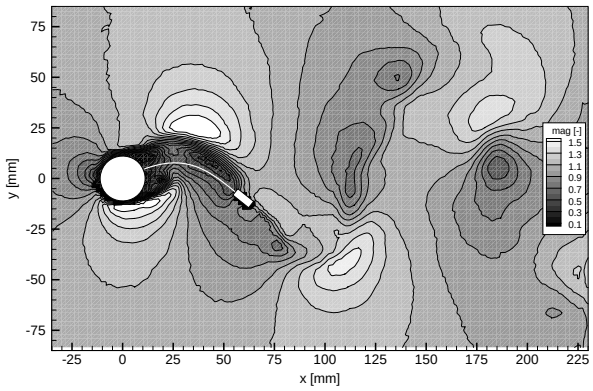
(c) tpa=90°



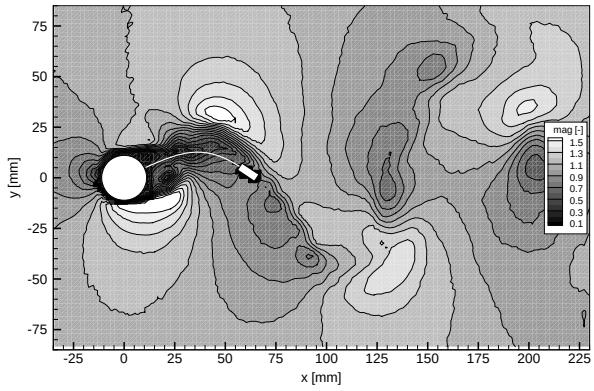
(d) tpa=135°



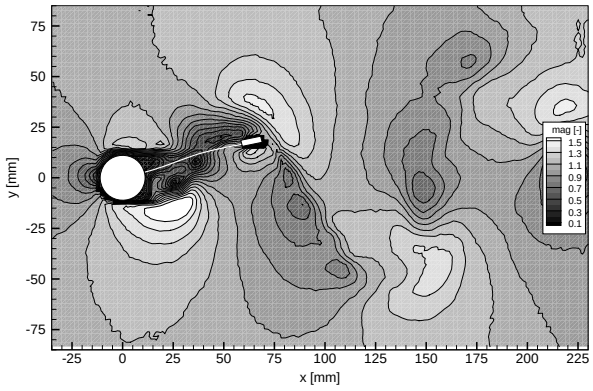
(e) tpa=180°



(f) tpa=225°



(g) tpa=270°



(h) tpa=315°

Figure 17: Time-phase resolved combined flow field/structure deflection measurements at eight different instants of the swiveling motion period at $0,68$ m/s ($Re \approx 15000$).

In the turbulent regime, one more mode was observed for very small inlet flow velocities. The lowest mode to be excited in water was the rigid body mode (referred to as the zero swiveling mode). This mode started to be observed at a very small approaching velocity and it was characterized by small structure deflections and movement frequencies. The maximum rigid body excitation was registered for $0,2$ m/s at the same time as the structure swiveled at about $1,1$ Hz. The transition to the first self-excitation mode was registered at about $0,4$ m/s and no hysteresis was observed. In this mode, the maximum excitation of the structure was achieved for $0,8$ m/s corresponding to a movement frequency of 5 Hz. Detailed measurements at $0,68$ m/s registered a $4,45$ Hz self-exciting resulting motion associated with a maximum excursion of the front body and the trailing edge of $29°$ and 25 mm, respectively. Finally, at $0,9$ m/s, the structure start to give indications of a swiveling mode transition; however, beyond this value it was not possible to register the dynamic behavior of the structure model. On increasing the velocity, the resulting movement of the model became unstable and non-reproducible, finally leading to the failure of the structure.

In a similar way as for the first laminar swiveling mode, a direct connection between the movement excitation and the classical von Karman vortex shedding behind the structure front cylinder was proved to exist. This is supported for the first swiveling mode observed in turbulent flows by the relation between the first natural frequency of the structure (N_1) and the turbulent Strouhal number ($St \approx 0,21$). The same direct relation was observed for the rigid body swiveling mode (or rigid body mode) on comparing the Strouhal number and the rigid body natural frequency of the structure ($N_0 = 0,19$ Hz).

Based on the results, one can conclude that the first swiveling mode is similar in nature in both the laminar and turbulent regimes. The main differences between the two are only connected with the fact that for turbulent flow the damping of the coupled system was significantly reduced. Thus, for the second case, higher excursions from the trailing edge and front body and also higher deflection of the flexible part of the structure were observed. Because of the lower damping imposed by the fluid, the structure was exposed to higher accelerations during its swiveling movement in the turbulent tests. Another difference appears when comparing the movement of the structure trailing edge. At higher Reynolds number, the area covered by the "figure-of-eight" shaped trajectory was considerably bigger than that obtained in the laminar tests. This is related to the delay registered between the rear mass and the front body movement. In both cases, the movement of the rear mass could be considered to be in concordance with the front cylinder rotation, but the delay increased with the Reynolds number: whereas at $Re = 140$ the delay was measured to be close to $60°$, at $Re = 15000$ it reached $95°$.

Comparing both self-excited coupled motions, one may conclude that both are triggered by the vortex shedding created around the front cylinder. Because the natural frequencies of the structure are constant and the Strouhal number is not so sensitive to the Reynolds number in the range 140-15000, the resulting movements have the same response as far as frequency of the movement versus approaching flow velocity is concerned. Considering all evidences, it can be concluded that both first swiveling modes correspond to instability-induced excited fluid-structure interaction cases. The same applies to the rigid body mode observed in the turbulent tests. In the laminar regime, a similar rigid body mode could not to be registered. The excitation process responsible for the second mode observed in laminar flow is more difficult to examine. However, the first results indicate strongly that this mode can be attributed to movement-induced excitation.

5 Acknowledgments

The present research project is part of the DFG Forschergruppe 493 - Fluid-Struktur-Wechselwirkung: Modellierung, Simulation, Optimierung. The authors gratefully acknowledge the financial support for their research work through the German Science Foundation (DFG) - Germany - and Fundação para a Ciência e a Tecnologia (FCT) - Portugal. In addition the authors acknowledge the scientific support given by the Erlangen Graduate School in Advanced Optical Technologies (SAOT) - Germany - to the present investigation.

References

Bungartz, H. J. and Schäfer, M. (2006): Fluid-Structure Interaction. Modelling, Simulation, Optimization. Lecture Notes in Computational Science and Engineering. Springer-Verlag.

Gomes, J. P. and Lienhart, H. (2006a): Time-resolved PIV/DMI measurements on two-dimensional fluid-structure interaction problems. 13th Int. Symposium on Applications of Laser Techniques to Fluid Mechanics Proceedings, Lisbon, Portugal.

Gomes, J. P. and Lienhart, H. (2006b): Experimental Study on a Fluid-Structure Interaction Reference Test Case: Fluid-Structure Interaction. Modelling, Simulation, Optimization. Lecture Notes in Computational Science and Engineering. Springer-Verlag. pp. 356-370.

Naudascher, E. and Rockwell, D. (1980): Practical Experiences with Flow-Induced Vibrations. Springer.

Naudascher, E. and Rockwell, D. (1994): Flow-Induced Vibrations - An engineering guide. Balkema.

Novak, M. (1972): Galloping Oscilations of Prismatic Structures. Journal of the Engineering Mechanics Division. Vol. 98 pp. 27-46.

Parkinson, G.V. and Smith, J.D. (1964): The Square Prism as an Aeroelastic Non-linear Oscillator. Quarterly, Journal of Mechanics and Applied Mathematics. Vol. 17 pp. 225-239.

Souli, M. and Hamdouni, A. (2007): Fluid Structure Interaction Industrial and Academic Applications. Revue Européenne de Mécanique Numérique. Vol. 16 n°. 3-4.

Address: Institute of Fluid Mechanics, University of Erlangen-Nuremberg
Cauerstr. 4, 91058 Erlangen, Germany
email: jorge.gomes@lstm.uni-erlangen.de; hermann.lienhart@lstm.uni-erlangen.de

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 147– 154
Herrsching am Ammersee, 29.9.-1.10.2008

Application of Monolithic Fluid-Structure Interaction to Identify Hydroelastic Instabilities of a Slide Gate Chain in Axial Flow

B. Hübner

During operation of a bottom outlet with a closed slide gate and an opend roller-mounted gate, a high velocity water flow (up to 80 m/s) occurs in the duct of the slide gate chain. The flow induces large amplidude chain vibrations perpendicular to the broadside of chain elements. In order to investigate the excitation mechanism and to compare different cross-sectional shapes of the lifting device by means of numerical simulation, a simplified model is developed. The chain is placed in the center of a square channel and modeled as a continuous plate structure with reduced stiffness to account for joints between the elements. For numerical simulation, a monolithic coupling of Navier-Stokes fluid dynamics and non-linear structural dynamics applies. The calculation model is based on a unified space-time finite element method. Stability limits of the simplified chain structure are identified by means of transient simulations with slowly increasing inflow velocity. In addition to the chain model with rectangular cross-section, also a cable with circular cross-section, equal axial stiffnes, and low bending stiffness is regarded, leading to a stability limit at higher inflow velocities. Thus, by using a cable instead of a chain for slide gate operation, the situation may be improved, but high amplitude flow induced vibrations cannot be prevented completely.

1 Introduction

In most freshwater reservoirs, a bottom outlet is designed to empty the reservoir for maintenance reasons. At high watermarks and the danger of dam overflow, it may also be used as an emergency outlet. The present paper regards a bottom outlet in which a slide gate and a roller-mounted gate are arranged subsequently. During normal outlet operation, the slide gate is closed, and the roller-mounted gate starts to open. Now, the water is flowing 11 meters in upward direction through the slide gate duct before it is flowing back in an additional duct and passing the roller-mounted gate. When the roller-mounted gate is half opened, the water flow inside the slide gate duct reaches a certain but not exactly known velocity which induces high amplitude vibrations of the slide gate chain perpendicular to the broadside of chain elements. The unstable system behavior remains when the roller-mounted gate is fully opened. In this case, a flow velocity of approx. $80\,\mathrm{m/s}$ has been measured in the duct of the slide gate chain. Figure 1 shows a view into the duct including the chain. The overall length of the duct is 50 meters, but only the lower 11 meters are filled with water.

Figure 1: View into the duct of the slide gate chain

Goal of this work is to study the excitation mechanism and to compare stability limits of lifting devices with different cross-sectional shapes to find an improved solution for slide gate operation. For this purpose, a simplified model of the coupled system suitable for numerical simulations is developed and described in section 3. Previously, the numerical solution procedure for analyzing strongly coupled fluid-structure interaction is summarized in section 2. Numerical results for a chain like structure with rectangular cross-section and an alternative cable structure with circular cross-section and very low bending stiffness are compared in section 4.

2 Numerical solution procedure

For numerical simulation of the coupled system in time domain, a monolithic coupling of incompressible Navier-Stokes fluid dynamics and geometrically non-linear structural dynamics applies. The solution procedure is based on a unified space-time finite element method which is described in detail by Hübner et al. (2004). The formulation of viscous fluid dynamics in moving domains follows the work of Tezduyar et al. (1992a) and Tezduyar et al. (1992b). The simultaneous solution procedure is applied by Hübner and Dinkler (2005) to identify hydroelastic instabilities of a fluid-conveying cantilevered pipe system which has been defined and experimentally investigated by Païdoussis and Semler (1998).

The governing equations for both solid and fluid are formulated in velocity variables and discretized with stabilized and time-discontinuous space-time finite elements leading to a rate formulation of structural dynamics, see Knippers and Harbord (1994) and Hughes and Hulbert (1988). A continuous finite element mesh applies to the entire spatial domain. Hence, velocity variables at the interface belong to both fluid and structural domain, fulfilling the coupling conditions automatically. The discretized model equations are assembled altogether in a single set of algebraic equations, considering the two-field problem as a whole. The space-time finite element method provides a consistent discretization of both space and time. By applying isoparametric space-time elements, which are adaptable in time direction, the method discretizes model equations in moving domains in a natural way and satisfies inherently the geometric conservation law, see Lesoinne and Farhat (1996). At discrete time levels, the interpolation of velocity, stress and pressure variables is discontinuous in time using independent degrees of freedom for the values at the end of the previous and at the beginning of the actual time slab. This time-discontinuous Galerkin formulation results in a fully implicit time integration scheme which is A-stable and of third-order accurate, see Johnson (1993). A Petrov-Galerkin stabilization of space-time elements prevents numerical oscillations in case of convection dominated flows and allows the application of equal order interpolations for velocities and pressure. The mixed-hybrid velocity-stress formulation of structural elements prevents all kinds of locking effects.

The highly nonlinear system of discretized model equations for solid and fluid dynamics as well as the movement of the fluid mesh has to be solved iteratively. Due to the application of a continuous finite element mesh for the entire domain, a monolithic formulation of solid and fluid in a single system of algebraic equations is obtained. A Picard iteration scheme linearizes all equations, leading to a relatively simple fixed point type solution procedure. Subsequently, the mesh movement procedure based on a pseudo-elastic approach applies. Thus, the entire set of model equations for solid, fluid, and fluid mesh movement is solved in a single iteration loop. In case of strong structural nonlinearities, the iteration is accelerated by applying a relaxation scheme for the calculation of structural displacements. For the relaxation parameter, values between 0.7 and 1.0 are appropriate in most cases. For solving the large sparse system of linearized equations, a preconditioned BiCGStab solver is used. However, a major difficulty is the choice of appropriate preconditioning methods. The matrix is not diagonally dominant and may be ill-conditioned if different scales are present in fluid and structure. Therefore, a direct LU-factorization of the system matrix is used for preconditioning during a variable number of time steps. When system characteristics change significantly, the system matrix has to be factorized again.

3 Modeling of chain and cable structure in axial flow

In order to investigate flow induced vibrations and to find an improved design for the slide gate lifting device, the existing chain structure is compared with a cable model. A clear advantage of the cable is the non-existence of bolt connections which therefore cannot be destroyed by large amplitude vibrations, but the bending stiffness is much smaller and may lead to a lower stability limit or higher amplitudes. However, the contact surface for the flow is smaller and the flow around the vibrating cable is more complex. Therefore, a simplified model of the coupled system suitable for efficient numerical analyses is regarded to compare stability characteristics of chain and cable structure. Both lifting devices are placed in the center of a water filled quadratic channel, see figure 2, and modeled as elastic solids with adapted stiffness properties to approximately account for the real structural behavior.

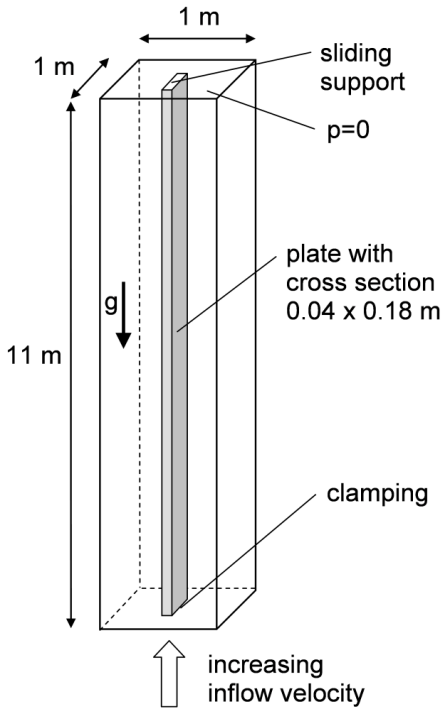


Figure 2: System configuration of the simplified model

3.1 Channel flow

For the flow simulation, an incompressible and viscous fluid including buoyancy effects is assumed and described by the instationary Navier-Stokes equations. At the lower end of the channel (inlet), a slowly increasing inflow velocity is imposed by $v(t) = 40\,\mathrm{m/s} + t \cdot 0.5\,\mathrm{m/s^2}$. At the upper end of the channel (outlet) zero pressure boundary conditions apply, see figure 2. With a density of $\rho = 1\,\mathrm{t/m^3}$ and a viscosity of $\mu = 10^{-6}\,\mathrm{t/(m\,s)}$, the Reynolds number based on the channel width is $Re = 4 \cdot 10^7$ for a flow velocity of $40\,\mathrm{m/s}$. Thus, the flow field is clearly turbulent. However, turbulence modeling is not taken into account, since boundary layers are very thin compared to the cross-sectional dimensions, and the velocity distribution over the cross-section is nearly constant. Moreover, the determination of friction losses is not considered in this investigation, and the onset of instabilities which is mainly governed by inertia effects may even be predicted by a potential flow theory, see e.g. Païdoussis (1998) and Païdoussis (2004). Furthermore, the non-linear time domain simulation of the coupled system over a large period with a small time step of $\Delta t = 0.005\,\mathrm{s}$ has to be performed with a comparable coarse fluid mesh in order to get reasonable simulation times.

Therefore, slip boundary conditions apply at channel walls, and only at the fluid-structure interface, no-slip conditions are present allowing for the development of a boundary layer. However, for the applied meshes (see next section), the boundary layer thickness of the simulation model is mainly defined by the thickness of the first fluid element layer at the interface which is in the range of the maximum thickness of the real turbulent boundary layer. Thus, the simplified flow model allows for a proper approximation of the real velocity profile in the slide gate duct. In order to prove that viscosity and boundary layer effects including the wall shear stress only have little influence on stability characteristics of the coupled system, the (laminar) viscosity is 10-fold increased, and the result is compared to the original case.

3.2 Chain with rectangular cross-section

The original chain is modeled as a beam like plate structure with rectangular cross-section and discretized with 3-dimensional solid elements for geometrically non-linear elastodynamics. From the entire chain, only the lower part which is immersed in water is regarded. At the upper end where the chain is cut off, a sliding support with fixed rotation applies, and at the lower end where the slide gate is attached, a clamping is placed. To account for the joints between chain elements, Young's modulus of the elastic solid, given by $E = 10^8\,\mathrm{kN/m^2}$, is reduced

to approximately half the value of steel. However, keep in mind, that also the real chain has a finite bending stiffness against deflections perpendicular to the broadside. For density and dead load, the real values of steel of $\rho = 7.8\,\mathrm{t/m^3}$ and $\gamma = 78\,\mathrm{kN/m^3}$ apply. Beside the damping due to flow effects, only a very small volume proportional damping coefficient of $b = 0.01\,\mathrm{t/(m^3\,s)}$ acts on the structure. The system configuration with all dimensions of the simplified chain model is given in figure 2. For numerical analyses, symmetry is considered to reduce the model size, see finite element mesh of a cross-sectional plane in figure 3.

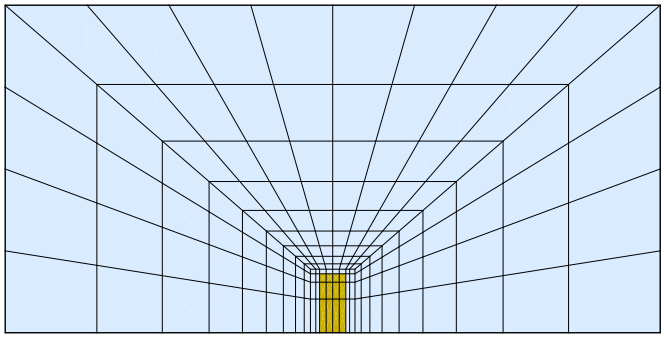


Figure 3: Cross-sectional mesh for the chain model

3.3 Cable with circular cross-section

Also for the alternative cable structure, symmetry conditions apply, and only half of the model is discretized, see the finite element mesh of a cross-sectional plane given in figure 4. However, this restricts the solution to in-plane vibration modes, even though three-dimensional vibration shapes are quite possible in case of circular cables. Upper and lower boundary conditions as well as the coupling to the fluid domain are equal to the numerical model of the chain structure. The cable diameter of $D = 0.09\,\mathrm{m}$ is chosen to match approximately the cross-sectional area of the chain model. In order to model real cable behavior exhibiting high axial stiffness and very low bending stiffness by means of an elastic solid, different Young's moduli are used in the center and at the outer region of the cross-section. For the two elements in the center of the symmetric mesh, a value of $E_i = 1.26 \cdot 10^9\,\mathrm{kN/m^2}$ applies. For the outer elements, a much smaller value of $E_o = 10^7\,\mathrm{kN/m^2}$ is used. Now, the axial stiffness of $EA = 7.2 \cdot 10^5\,\mathrm{kN}$ is equal to the chain model, while the bending stiffness of $EJ = 65\,\mathrm{kNm^2}$ is clearly smaller. The values for density, dead load and damping are taken from the chain model.

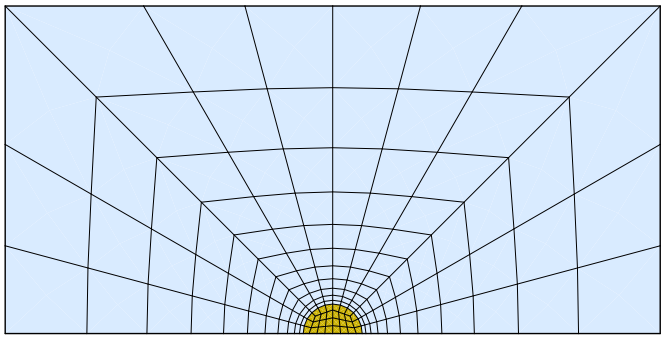


Figure 4: Cross-sectional mesh for the cable model

4 Comparison and evaluation of results

The simplified numerical model of the coupled system may not describe exactly the real physics of chain and cable structure in the turbulent flow field of the slide gate duct. However, the simplifications are comparable for both cases, and therefore, the numerical results allow a qualitative comparison of the sensitivity to flow induced vibrations and hydroelastic instabilities. In order to determine the limit velocity for the occurrence of hydroelastic instabilities leading to large amplitude vibrations, a horizontal impulse load is acting every 15 seconds on the upper end of the structure, while the inflow velocity increases continuously. The first impulse load acts after

one second of calculation time when a stationary flow field and an equilibrium state of the structure has been developed. The stability limit is given by the point in time at which the vibration amplitudes due to the impulse load start to increase. When the first impulse load acts at $40.5\,\mathrm{m/s}$ inflow velocity on the original chain model, the structure responds initially with a damped vibration behavior. However, after 5.6 seconds of simulation time at $v = 42.8\,\mathrm{m/s}$ inflow velocity, the dynamic behavior changes and the system gets unstable. See figure 5, where time histories of horizontal displacements at center and upper end of the chain model are shown for the first 12 seconds of simulation time. The vibration frequency which is increasing continuously with the inflow velocity, becomes $f = 1.4\,\mathrm{Hz}$ at the stability limit. The chain model with 10-fold increased fluid viscosity exhibits nearly identical stability characteristics. The stability limit is reached at a flow velocity of $v = 43.2\,\mathrm{m/s}$, and the corresponding frequency is given by $f = 1.4\,\mathrm{Hz}$, too, see displacement time histories of the entire calculation in figure 6. Thus, it can be concluded that boundary layer effects have only minor influence on stability characteristics which are mainly governed by inertia effects.

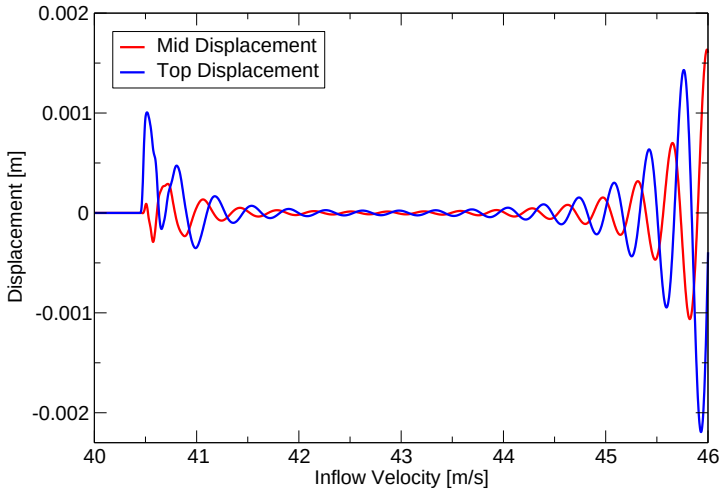


Figure 5: Time histories of horizontal displacements near to the satbility limit for the chain model

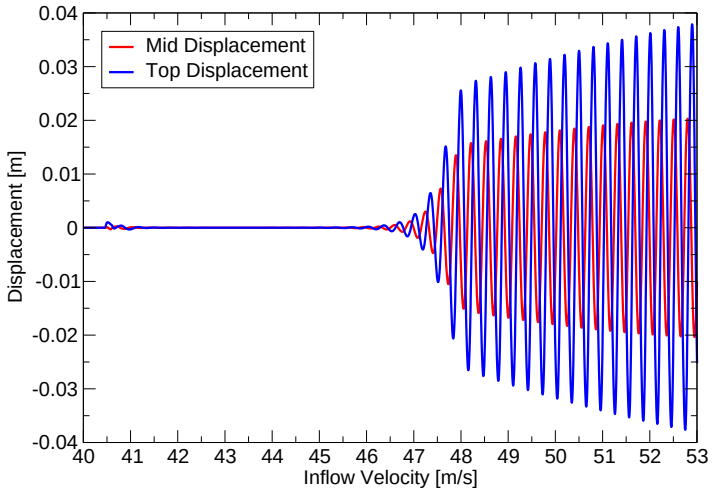


Figure 6: Time histories of horizontal displacements for the chain model with 10-fold fluid viscosity

In case of the cable structure, the impulse load induces much larger initial deflections in the stable regime, because bending stiffness and flow resistance are clearly smaller. Nevertheless, the vibration due to the first impulse load decrease nearly to zero, and the hydroelastic instability occurs much later. At $v = 50.0\,\mathrm{m/s}$ inflow velocity, the cable system tends to get unstable, but the amplitudes increase only slightly. The real stability limit exhibiting strongly increasing amplitudes is first reached at an inflow velocity of $v = 60.2\,\mathrm{m/s}$, see time histories of displacements in figure 7. Stability limits and corresponding frequencies of all regarded models are compared in table 1. Snap shots of the deformed cable structure during a single period of the coupled vibration are shown in figure 8. The type of motion which is quite similar for the chain models can be classified as a kind of travelling wave flutter, comparable to a flag in wind flow.

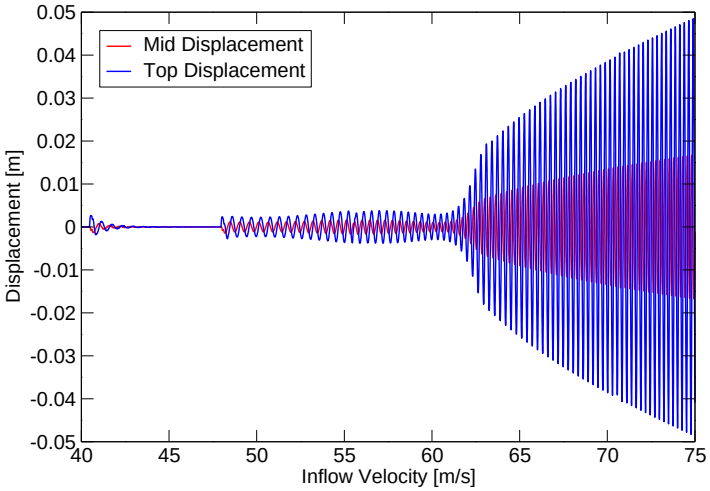


Figure 7: Time histories of horizontal displacements for the cable model

model	velocity at stability limit	corresponding frequency
rectangular chain	42.8 m/s	1.4 Hz
chain (10-fold visc.)	43.2 m/s	1.4 Hz
circular cable (1st)	50.0 m/s	1.9 Hz
circular cable (2nd)	60.2 m/s	2.8 Hz

Table 1: Comparison of stability limits for chain and cable model

Although the bending stiffness of the cable model is clearly smaller compared to the chain model, instability occurs at much higher flow velocities. This behavior is caused by the different cross-sectional shapes leading to different flow situations. In case of the chain model, the flow is syncronized over the entire contact surface (broadside of chain elements) leading to approximately two-dimensional flow behavior. In contrast, the contact surface of the cable is clearly smaller and the circular shape causes a fully 3-dimensional and more complex flow situation. This results in the favorable stability characteristics of the cable model, at least if only in-plane vibrations are regarded. However, for both lifting devices, large amplitude flow induced vibrations have been detected within the operating range of up to $80\,\mathrm{m/s}$ flow velocity. Nevertheless, the cable seems to have clear advantages over the chain structure.

5 Conclusion

In order to find reasons for large amplitude flow induced vibrations of a slide gate chain and to look for an improved design, numerical simulations of simplified chain and cable models have been performed and compared. The simplified models consist of continuous elasic solids with adapted stiffness properties to approximately account for the real behavior of chain and cable, respectively. An incompressible and viscous fluid has been assumed without special attention on turbulence modeling. However, by comparing results for different viscosities, it has been shown that boundary layer effects have very little influence on stability characteristics. For the transient simulation of the strongly coupled system, a monolithic approach to fluid-structure interaction based on space-time finite elements has been applied. Stability limits of $43\,\mathrm{m/s}$ and $60\,\mathrm{m/s}$ flow velocity have been detected for chain and cable, respectively. Thus, by using a cable instead of a chain structure for slide gate operation, the situation may be improved, especially because a cable does not feature bolt connections which can be destroyed by strong vibrations. However, high amplitude flow induced vibrations cannot be prevented completely if the flow velocity in the slide gate duct reaches values up to $80\,\mathrm{m/s}$.

References

Hübner, B.; Dinkler, D.: A simultaneous solution procedure for strong interactions of generalized newtonian fluids and viscoelastic solids at large strains. *International Journal for Numerical Methods in Engineering*, 64, (2005), 920–939.

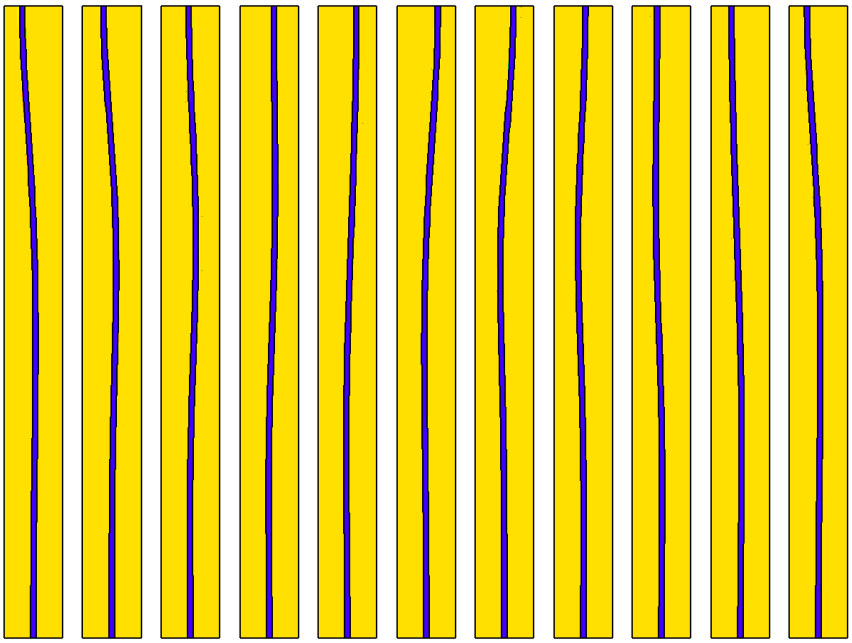


Figure 8: Snap shots of the deformed cable structure (2-times magnified) during a coupled vibration period

Hübner, B.; Walhorn, E.; Dinkler, D.: A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193, (2004), 2087–2104.

Hughes, T. J. R.; Hulbert, G. M.: Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66, (1988), 339–363.

Johnson, C.: Discontinuous galerkin finite element methods for second order hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 107, (1993), 117–129.

Knippers, J.; Harbord, R.: A mixed hybrid FE formulation for solution of elasto-viscoplastic problems. Part II: Dynamic loading conditions and bending problems. *Computational Mechanics*, 13, (1994), 231–240.

Lesoinne, M.; Farhat, C.: Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134, (1996), 71–90.

Païdoussis, M. P.: *Fluid-Structure Interactions: Slender Structures and Axial Flow*, vol. 1. Academic Press, San Diego (1998).

Païdoussis, M. P.: *Fluid-Structure Interactions: Slender Structures and Axial Flow*, vol. 2. Academic Press, San Diego (2004).

Païdoussis, M. P.; Semler, C.: Non-linear dynamics of a fluid-conveying cantilevered pipe with a small mass attached at the free end. *International Journal of Non-Linear Mechanics*, 33, (1998), 15–32.

Tezduyar, T. E.; Behr, M.; Liou, J.: A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain / space-time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94, (1992a), 339–351.

Tezduyar, T. E.; Behr, M.; Mittal, S.: A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain / space-time procedure: Ii. Computation of free-surface flows, wo-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94, (1992b), 353–371.

Address: Dr.-Ing. B. Hübner, Voith Siemens Hydro Power Generation GmbH & Co. KG, Basic Development, Alexanderstr. 11, 89522 Heidenheim, Germany.
email: `bjoern.huebner@vs-hydro.com`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 155– 294
Herrsching am Ammersee, 29.9.-1.10.2008

Shape Optimization in Partitioned Analysis for Fluid-Structure Interaction

U. Israel, E. Stavropoulou, M. Barcelos, T. Gallinger, K.-U. Bletzinger, R. Wüchner

A modular framework for gradient based optimization in partitioned analysis for fluid-structure interaction (FSI) is presented. To guarantee flexibility for several technical applications of shape optimal design for structures subject to fluid flow the software environment is designed for the Nested Analysis and Design (NAND) concept and partitioned FSI schemes using three field analyses including structure, fluid mesh and fluid. The developed framework provides the possibility to include and combine several approaches for optimization methods, shape definition and sensitivity analysis.
For shape definition, parametric and non parametric methods are discussed and illustrated by examples realized within this framework. A special focus is set on the Updated Reference Strategy as an effective and efficient way to obtain free form shapes of membranes for the design process. Incorporating this nonlinear form finding process into the optimization frame allows for finding real free form surfaces by using only a small number of optimization variables. For applications using these shape definitions numerical and semianalytical sensitivity method are discussed. In particular, a numerical and a direct sensitivity analysis method based on structural sensitivity equation are presented. Examples using different shape definitions and sensitivity analyses are shown for verification of the optimization framework.

1 Introduction and Motivation

In recent years good advances in FSI simulation have been made and optimization of FSI problems became a in-demand research topic. Good experience with optimization methods on fluid and structure side influenced the development of optimization of coupled systems (Thévenin and Janiga, 2008; Mohammadi and Pironneau, 2004; Jameson, 2004, 1995; Bletzinger et al., 2008; Haftka and Grandhi, 1986; Arora and Wang, 2005). Several approaches towards the optimization in fluid-structure interaction exist (Sobieszczanski-Sobieski and Haftka, 1997; Haftka et al., 1992; Bletzinger et al., 2006; Soto and Löhner, 2001; Lund et al., 2003; Maute et al., 2001, 2003).
Evolutionary algorithms are effective in finding global optima and simple to implement, but cause high computational costs because of slow convergence. Using gradient based algorithms, the computational effort can be extremely reduced. Gradients indicate the direction of a better solution and therefore fast convergence is achieved. The crucial point is to obtain the gradient information needed for the optimization procedure. Various approaches on numerical and semianalytical sensitivity analysis exist and will be discussed. For problems with low number of optimization variables numerical approaches are adequate, but the numerical effort increases with increasing number of optimization variables. Since high shape freedom is guaranteed by higher number of optimization variables in shape optimization, more complex methods are required here.
From the technical point of view, it is important whether partitioned or monolithic schemes are used. For big technical problems which require the use of specific and well tested simulation tools the partitioned approach offers the required modularity, but coupling of the single field codes becomes necessary. Basing optimization on a partitioned scheme, coupling has to be done during the whole process and therefore high demands arise in software engineering.

2 Requirements for Optimal Shapes of Lightweight Structures

2.1 Handling Lightweight Structures

The vision behind this work is simulation and shape optimization of large and lightweight structures interacting with fluid flow, especially in the area of aerospace, mechanical and civil engineering. Wide span membrane roofs and shell structures as well as inflatable and pressure dominated structures are typical applications in lightweight construction. These types of problems cause large numbers of degrees of freedom and, therefore, require efficient simulation tools. To introduce best suited codes for the single fields the partitioned approach guarantees the needed modularity for big technically relevant problems. Section 4 of this paper and reference (Gallinger et al., 2008) describe this approach. For the communication during runtime and the data transfer between non matching grids on the common interface a coupling tool is required. The huge number of degrees of freedom and the herewith connected high computational effort in optimization renders gradient based algorithms indispensable. In section 3 additional information is given.
Towards shape optimization of these complex systems, optimization methods for optimal design of shell and membrane structures subject to fluid flow have been developed for a steady state solution with the aim to optimize several practically relevant criteria like structural deformations and stresses. Correspondingly, displacement and force formulations are chosen as objectives. The special case of compliance minimization is also considered regarding the possibility to design optimal stiffened structures with respect to a specific loading.

2.2 Guaranteeing Shape Diversity

In shape optimization a maximum freedom in shape design for shell and membrane structures and definition of structural optimization criteria is focused. In general this also means the introduction of a huge number of design variables and accordingly increasing numerical effort. Several methods to reduce this effort can be applied. This can be done directly by reducing the number of design variables in an adequate way or by introducing adjoint methods in sensitivity analysis.
For shells the Computer Aided Geometric Design (CAGD) method allows for a small number of design variables by linking structural FE meshes to design elements. In this case, the connection between the FE-node locations and the design variables is established by means of mathematical relations. Nevertheless, the design freedom is strongly dominated by the choice of these relations. Therefore also CAGD free methods which guarantee the maximum design freedom by taking the nodal positions of the finite element mesh as design variables are needed. Then a huge number of design variables arise. To overcome the resulting intense numerical effort the adjoint formulation for sensitivity analysis must be introduced.
For structures acting as membranes mechanic based strategies are necessary. Here the Updated Reference Strategy (i.e. a generally formulated form finding method) is presented as an effective and efficient way to obtain free form shapes of membranes for the design process. Including the nonlinear form finding process into the optimization frame allows for finding real free form surfaces by just using a small number of design variables. This means that although a non-parametric shape definition is used and efficient reduction of number of design variables is achieved by applying a mechanics based criterion. Here, the form finding method is used as an effective non parametric shape description method to obtain free form shapes with a reduced number of optimization variables.
The provided design descriptions for the structural shape require multiple and flexible methods for sensitivity analysis. Numerical and semianalytical methods are required for applications using the shape definitions mentioned above. Numerical methods are needed for free form membranes, since the form finding process is nonlinear and inverse. Semianalytical direct methods are best suited for structures defined by CAGD. For the CAD free optimization semianalytical adjoint methods are required.
In section 6 and 7 shape description methods and sensitivity analysis are discussed.

2.3 Developing a Modular Software Environment

The desired variety and flexibility to all these mentioned possibilities can be guaranteed in a modular framework for gradient based optimization in partitioned analysis for fluid-structure interaction. To achieve the modularity and flexibility for various problems of shape optimal design for structures subject to fluid flow the software environment is designed for the Nested Analysis and Design (NAND) concept and a partitioned FSI scheme using a three field analysis including structure, fluid mesh and fluid. This involves the interaction of an optimizer, a coupled

analysis and a coupled sensitivity analysis. Coupling and data transfer has to be extended to optimization and the sensitivity analysis, resulting in special demands on the coupling tool. Finally, the developed framework provides the possibility to include and combine different approaches for optimization methods, shape definition and sensitivity analysis.
The code coupling plays a central role to ensure modularity in the use of fluid solvers as well as to handle coupled sensitivity analysis. Therefore the development of an independent coupling tool was one of the main working steps.
In section 5, the developed framework including the code coupling is presented.
Additionally, to enhance the powerful structural analysis and optimization tool CARAT it is a desired feature to plug in and run a FSI analysis with any fluid code. Thus, the possibility to use an arbitrary fluid solver as blackbox for the sensitivity analysis is exploited as far as possible, regarding reduction of numerical effort.

3 Gradient Based Optimization

In the following, the formulation and the solution strategy of the optimization problem is described. The optimization problem can be stated as follows:

$$f(s) \rightarrow \min$$

$$\begin{aligned} \text{such that} \quad & g_j(s) \leq 0 \quad j \in [1, m] \\ & h_j(s) = 0 \quad j \in [m+1, m+N] \end{aligned}$$

$$\text{where} \quad \underline{s_l} \leq s_l \leq \bar{s_l} \quad s \in \mathbb{R}^n$$

where f,g_j,h_j are the objective function, the inequality and the equality constraints, respectively. They are functions of $s = (s_1, \cdots, s_n)^T$ which is the vector of the optimization variables. The existence of the constraints changes totally the way to treat the problem. Consequently, constrained and unconstrained problems form two big classes of optimization problems. Unconstrained optimization problems using gradient based methods are discussed in the sequel. On the one hand, these methods are more efficient, have better convergence rates and require few evaluations of the objective function. On the other hand, derivative information has to be provided and because of that problems might occur when the objective function has discontinuities or kinks.
In general, gradient-based methods start with an initial guess s_0 and in each iteration the solution is updated as follows :

$$s_{i+1} = s_i + \alpha_i D_i$$

where $D^i = D^i(\nabla f(s_i); \nabla^2 f(s_i)) \in \mathbb{R}^n$ is the search direction and α_i is the step size, which is usually calculated with a line-search algorithm. Steepest descend algorithm is one of the simplest and most popular algorithms of this class because of its efficiency and robustness. This algorithm is using the property of the "steepest descent" of the negative gradient direction ($D = -\nabla f(s)$). The algorithm is as follows:

i Initialize $i = 0 : s_0$

ii Evaluate $D^i = -\nabla_{s_0} f$

iii Update $s_{i+1} = s_i + \alpha_i D_i$

iv If $i = k$ stop. maximum number of iterations k is reached.
otherwise: convergence check
if converged stop
else:$i = i + 1$ go to ii.

The technical implementation of an optimization problem is done using the Nested Analysis and Design (NAND) approach (Arora and Wang, 2005). Typically, in each optimization step the objective has to be evaluated by performing a full analysis with the actual design. Accordingly, the NAND consists of three independent modules, optimization, analysis and sensitivity analysis, which are called successively during the iterations of the optimization algorithm (figure 1). On the optimization level the design variables are updated. With the actual design the objective is evaluated on the analysis level. Finally, using information from the analysis, the objective function gradient for the next step can be calculated in the sensitivity analysis. The NAND approach provides high modularity because of exchangeability of the modules.

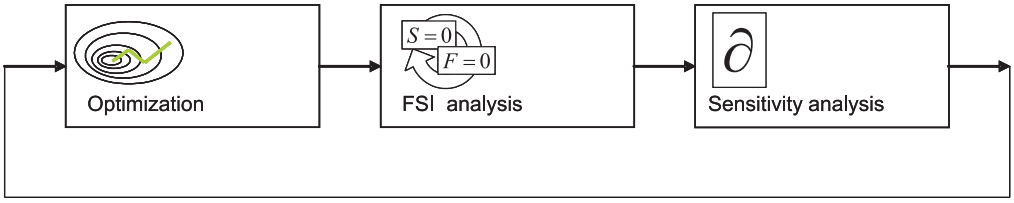


Figure 1: Concept of Nested Analysis and Design: Optimization, analysis and sensitivity analysis

4 Partitioned FSI Analysis

A three field analysis including structure, fluid mesh and fluid is performed. The fields are described by nonlinear equations, which are coupled through state variables. For solving the corresponding discrete equations of structure $S(s,u,v,w) = 0$, fluid mesh $M(s,u,v,w) = 0$ and fluid $F(s,u,v,w) = 0$ a staggered approach is used. The variables of the coupled discrete equations are: structure displacement u, fluid mesh position w and fluid state v. The structural problem is solved by a finite element method using an in-house computational tool called CARAT (Wüchner et al., 2007; Wüchner, 2006), and the fluid mesh motion and the fluid problems are solved by the finite volume method using an open source library called OpenFoam (Jasak and Tukovic, 2007). The coupled problem is solved to achieve a steady state solution. The data transfer between non-matching boundary meshes is done by an in-house coupling tool called CoMA (Israel et al., 2007).
Detailed information on the coupling schemes and the software environment is given in (Gallinger et al., 2008). For further information the reader can refer to (Ferziger and Perić, 1999; Wall, 1999; Felippa et al., 2001) .

5 A Framework for Gradient Based Optimization and Partitioned fluid-structure Interaction

5.1 Nested Analysis and Design for Partitioned Analyses

The challenging task is to combine properly the partitioned fluid-structure interaction approach and optimization using the NAND approach. Since the analysis as well as the sensitivity analysis and the optimization module concern fluid and structure field, all these modules need to communicate between each other according to the developed NAND method. In particular they need an individual coupling of fluid and structure field corresponding to the information which has to be exchanged. The design of the framework is depicted in figure 2.
On the optimizer level the update of the design variables and therefore the shape is done during the optimization iterations i. Each design change of the structure has to be passed to the fluid as a base for the next FSI analysis. On this level information is just passed in one direction and due to the known design increment no iteration is needed. Nevertheless the data transfer on the non matching meshes has to be taken into account. Since the design change of the structure corresponds to a position update of the structural finite element nodal coordinates this transfer can be performed similar to the transfer of structural displacements and therefore the same technique can be used.
The data transfer on the analysis level is done by transferring displacements from the structure to the fluid and loads from the fluid to the structure. Iterations are necessary here and a convergence check has to be performed. Each circle contains a full nonlinear or linear structural analysis.
Coupled sensitivity analyses can differ strongly. Data transfer on this level depends on the type of sensitivity analysis. Numerical methods using external finite differences get gradient information from solving the coupled system for reference and perturbed states. This means that the sensitivity analysis includes the perturbation of the design variables and a FSI analysis for the perturbed state. The coupling information then is analogue to the coupling information of the optimization and the analysis module, namely a design update and an iterative passing of displacements and forces. In semianalytical approaches, the coupling data differs. By evaluating the objective function gradient applying the chain rule terms corresponding to fluid and structure sides occur. These terms are obtained in the respective codes by different methods. In general the data which has to be passed is derivative information with respect to state variables or design variables depending on the type of semianalytical sensitivity analysis. In section 7 detailled information about the coupled sensitivity analysis is given.
Finally, the framework is designed to be modular with regard to the possible combinations of different approaches for optimization methods, shape definition and sensitivity analysis. On the optimizer level the optimization algorithms itself and shape definitions are defined and the user can choose between the available tools. On the analysis level several techniques to perform a fluid-structure interaction simulation are implemented and can be chosen. As sensitivity analysis numerical and semianalytical methods are possible.

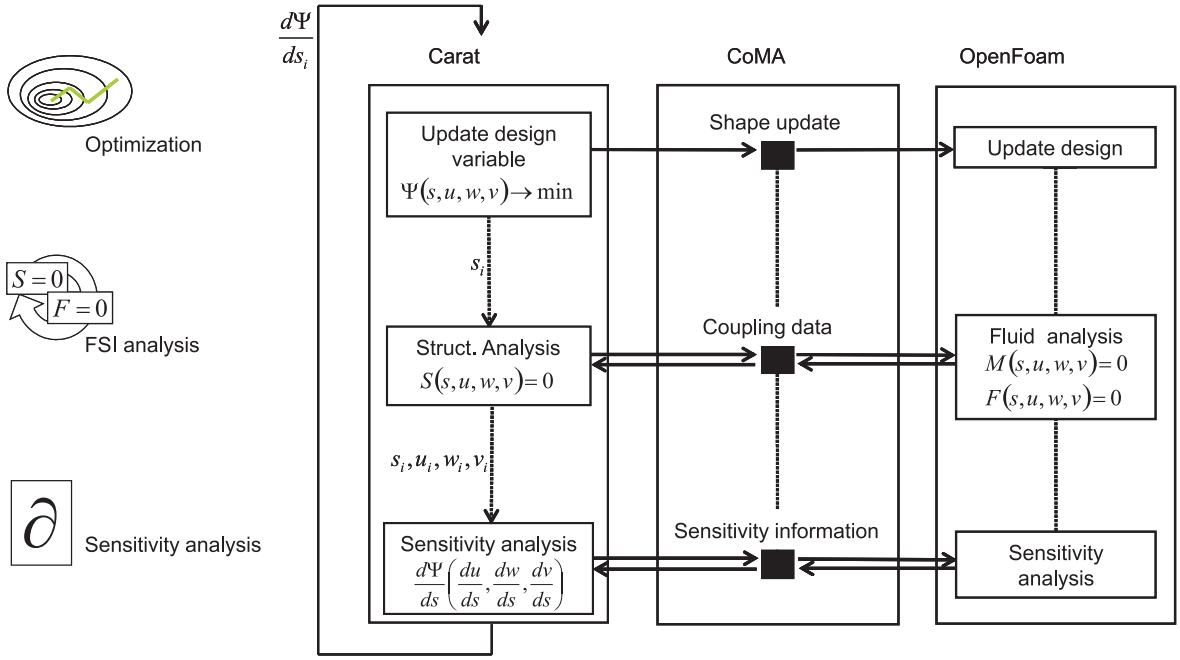


Figure 2: Optimization framework according to the NAND approach for partitioned FSI simulation

5.2 Code Coupling

To realize the coupling within the framework an independent coupling tool called CoMA (Coupling for Multi-physics Analysis) was developed. The main task of this tool is to perform communication between the codes and data mapping between the generally non matching grids on the levels of optimization, analysis and sensitivity analysis, respectively. Since control is done by CoMA, any kind of optimization and coupling procedure can be realized by simply modifying the encapsulated overall algorithmic strategy. To sum up, for any surface coupled problem CoMA provides the following main features:

- Compatibility to various codes including black box software.
- Possibility to perform parallel computations.
- Data transfer between non matching grids by interpolation and conservative summation.
- Central datastructure for any mesh based quantities.
- Data exchange: file based or by MPI.
- Control of coupled computation and convergence control.
- Coupled sensitivity analysis.

For any further information on application of CoMA in coupled analysis the reader can refer to (Israel et al., 2007; Gallinger et al., 2008) and for information about load and motion transfer to (Farhat et al., 1998; Farhat, 2004) A special focus is set here on the ability to handle coupled sensitivity analyses by transferring sensitivity information described in detail in section 7.4.

6 Shape Definition

6.1 CAGD Based Shape Optimization

State of the art in the field of structural mechanics is the separation of the geometrical model and the calculation model. The geometrical model is based on CAD or CAGD and the structural model is the discretized version of the CAD/CAGD model. The design variables for the optimization process are the geometry parameters of the CAGD-model. Thus, high numerical efficiency is reached because of a low number of optimization variables.
As an example for a CAGD based shape optimization the shell structure depicted in Fig. 3 is chosen. The geometry

is modeled by the use of Bezier Patches, whereas not more than 10 variables are needed as optimization variables, making use of the structural symmetry. The result shows the optimum shape within the possible domain for maximum stiffness (Bletzinger et al., 2005; Haftka and Grandhi, 1986).

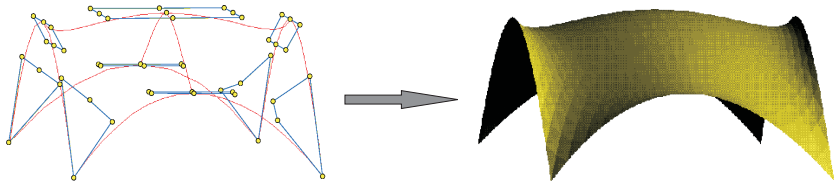


Figure 3: CAGD-model and optimal shape

6.2 CAGD Free Shape Optimization

A disadvantage of optimization based on CAGD models is that by the choice of the geometric design parameters also a restriction of the design space and possible solutions is made. These restrictions can be overcome, if the nodal coordinates of the finite element model are used as optimization parameters (Haftka and Grandhi, 1986; Pironneau, 1984). Thus all possible solutions which can be described by the discretization chosen for analysis can be achieved. Furthermore, by omitting the generally complicated selection of geometry based optimization parameters and the herewith combined definition of a CAGD model the overall modeling and setup are simpler and faster. However, parameter free optimization leads to a very high number of parameters. Figure 4 shows the results of a parameter free optimization towards the maximization of stiffness of a thin cantilever shell made out of metal loaded as shown.The model consists of approximately 500 design variables. The optimal shape (most right) is reached after 19 iterations.

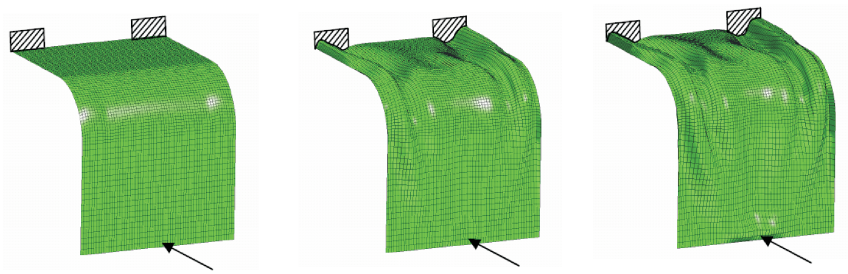


Figure 4: Shape optimization of a cantilever shell for compliance problem using the CAGD free method

6.3 Shape Optimization using Form Finding of Membranes

Membrane structures are very attractive alternatives to span large distances. They are light, elegant, and efficient. The material is optimally used since the structures are subjected only to membrane tension stresses. The anisotropy and distribution of pre-stress together with the choice of edge cable lengths and forces as well as the position of supports define the overall shape and appearance of the structure. The art of form finding is to play with these parameters to find the shape of the membrane, which is naturally defined by the equilibrium of the surface and edge cable forces. The idea is closely related to the determination of minimal surfaces so the resulting shapes are physically inspired (“soap film analogy”). Mathematically spoken, the task is to determine geometrical shapes solving an inverse mechanical problem. This results in singular expressions during the numerical solution of the principle of virtual work. To overcome this problem several form finding methods were developed. Within this contribution, the updated reference strategy (URS) is successfully used. The basic idea of the URS is the modification of the originally singular virtual work expression ($\delta W_{\boldsymbol{\sigma}} = 0$) by adding a stabilizing part $\delta W_{\mathbf{S}}$ (in terms of $PK2$ stresses $\mathbf{S}$ rather than Cauchy stresses $\boldsymbol{\sigma}$). These are formulated in that way, that they fade out as the solution is approached and the original, unmodified solution is received. The complete weak form of the stabilized

form finding scheme (URS) states as follows:

$$\begin{aligned}\delta W_\lambda &= \lambda\,\delta W_{\boldsymbol{\sigma}} + (1-\lambda)\,\delta W_{\mathbf{S}} \\ &= \lambda \left[h \int_A det\mathbf{F}\left(\boldsymbol{\sigma}_{pre}\cdot\mathbf{F}^{-T}\right) : \delta\mathbf{F}\;\mathrm{d}A \right] \\ &\qquad + (1-\lambda)\left[h \int_A (\mathbf{F}\cdot\mathbf{S}_{pre}) : \delta\mathbf{F}\;\mathrm{d}A \right] = 0\end{aligned} \tag{1}$$

For more details the reader can refer to (Bletzinger et al., 2005, 2006, 2008; Wüchner and Bletzinger, 2005; Wüchner, 2006).

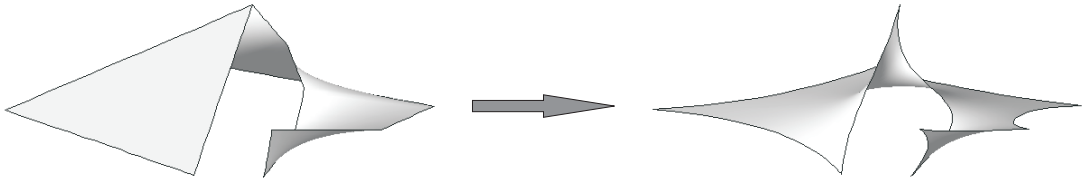


Figure 5: Membrane formfinding by URS: Initial geometry and minimal surface

The concept of optimizing a membrane structure can have many perspectives. For instance, form finding and structural optimization can generate optimal shapes inspecting the optimization problem from different views.
On the one hand form finding is restricted on membrane and shell structures and determines the shape from an inverse formulation of equilibrium due to a given stress distribution acting on the deformed structure. Control variables that could influence the final shape are the prestress ratio between the membrane and the cables, the coordinates of the supports, i.e. the Dirichlet boundary conditions. But form finding doesn't introduce a way to investigate the relation between the control variables and the external loading.
On the other hand, structural optimization determines the shape from a combination of different criteria that can be chosen according on the different needs of the problem. But in order to control the shape of a membrane many design variables are needed and consequently the solution becomes complex and computationally expensive. Furthermore, a quite intuitive relation has to be chosen between the design variables (linking rule) which depends on each specific problem.
Combining form finding and optimization we can achieve both: physically inspired shapes and a systematic treatment of the control variables. With this way we decrease the number of design variable and we obtain a control of the total structure with only a few significant design variables by still omitting an a priori fixed mathematical relation (like in CAGD) which leads to "real" free form shapes. As a result, we have a physical linking between the design variables and the total computational time is reduced significantly.
In the following, the "form finding enriched shape optimization in FSI" is described (figure 6). The shape of the membrane structure is computed by the URS. More precisely, after every update of the design variables form finding is performed to smoothen the shape and update the design. As design variables the control parameters of form finding are used. These are e.g. the distribution of prestress of the membrane and the edge cables or the position of the supports.

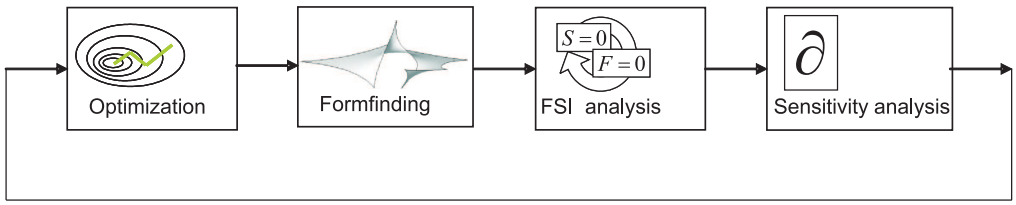


Figure 6: form finding enriched shape optimization in FSI

7 Multifield Sensitivity Analysis

7.1 Direct and Adjoint Approaches in Multifield Semianalytical Sensitivity Analysis

Optimization in fluid-structure interaction means minimizing the objective function ψ and in addition fullfilling the governing equations of structure, fluid mesh and fluid (Bletzinger et al., 2006; Barcelos and Maute, 2008; Barcelos et al., 2006; Maute et al., 2001, 2003; Soto and Löhner, 2001; Sobieszcanski-Sobieski, 1990; Löhner, 2008; Etienne and Pelletier, 2005).

$$\begin{aligned} &\psi(s,u,w,v) \rightarrow min \\ &S(s,u,w,v) = 0 \\ &M(s,u,w,v) = 0 \\ &F(s,u,w,v) = 0 \end{aligned} \tag{2}$$

$S(s,u,w,v)$, $M(s,u,w,v)$ and $F(s,u,w,v)$ are the the coupled system discrete equations corresponding to structure, fluid mesh and fluid. These equations depend on the design variables s and the state variables of structure u, fluid mesh w and fluid v, which again are dependent on the design. A gradient based optimization method requires the computation of the objective function gradient ψ with respect to the design variables s. Therefore a consistent sensitivity analysis formulation has to be based on the coupled system discrete equations. To obtain the desired gradient numerical approaches are costly because for each design variable at least two objective function evaluations have to be performed and therefore are recommended only for a small number of design variables. For bigger problems a analytical or semianalytical sensitivity analysis becomes indispensable.
Like state equations the objective function generally depends on the design variables s and the state variables of structure, fluid mesh and fluid. Accordingly, the gradient of the objective function ψ with respect to design variables s is written as

$$\frac{d\psi}{ds} = \frac{\partial\psi}{\partial s} + \frac{\partial\psi}{\partial u}\frac{du}{ds} + \frac{\partial\psi}{\partial w}\frac{dw}{ds} + \frac{\partial\psi}{\partial v}\frac{dv}{ds} \tag{3}$$

In analytical and semianytical methods the terms are determined individually. Here $\frac{\partial\psi}{\partial u}$, $\frac{\partial\psi}{\partial w}$ and $\frac{\partial\psi}{\partial v}$ are known by the definition of the objective function and are in general easy to obtain. The state variable derivatives $\frac{du}{ds}$, $\frac{dw}{ds}$ and $\frac{dv}{ds}$ are the unknowns and have to be obtained by additional equations.
Using the so called direct sensitivity method the unknowns can be determined directly by using the sensitivity equations, derived from the governing equations for structure, fluid mesh and fluid, respectively.

$$\begin{aligned} \frac{dS}{ds} &= \frac{\partial S}{\partial s} + \frac{\partial S}{\partial u}\frac{du}{ds} + \frac{\partial S}{\partial w}\frac{dw}{ds} + \frac{\partial S}{\partial v}\frac{dv}{ds} = 0 \\ \frac{dM}{ds} &= \frac{\partial M}{\partial s} + \frac{\partial M}{\partial u}\frac{du}{ds} + \frac{\partial M}{\partial w}\frac{dw}{ds} + \frac{\partial M}{\partial v}\frac{dv}{ds} = 0 \\ \frac{dF}{ds} &= \frac{\partial F}{\partial s} + \frac{\partial F}{\partial u}\frac{du}{ds} + \frac{\partial F}{\partial w}\frac{dw}{ds} + \frac{\partial F}{\partial v}\frac{dv}{ds} = 0 \end{aligned} \tag{4}$$

A nonlinear equation system arises from these equations and can be solved in a staggered manner for the state variable derivatives. Then, the objective function gradient is computed as a byproduct of the solution of the sensitivity equations which is the derivative of the state variables.
Alternatively the adjoint sensitivity method can be used by formulating the optimization problem as a constraint problem. The Lagrange function which has a stationary point at the optimum can be developed as follows. Here the discrete equations for structure, fluid mesh and fluid are taken as equality constraints and λ_u, $\lambda_w$and λ_v are the corresponding Lagrange multipliers, which can be identified as adjoint variables.

$$L = \psi + \lambda_u^T S + \lambda_w^T M + \lambda_v^T F \rightarrow stat \tag{5}$$

The solution can be determined by variation of L leading to

$$\begin{aligned} \delta L &= \frac{\partial L}{\partial s}\delta s + \frac{\partial L}{\partial u}\delta u + \frac{\partial L}{\partial w}\delta w + \frac{\partial L}{\partial v}\delta v + \delta\lambda_u^T S + \delta\lambda_w^T M + \delta\lambda_v^T F \\ &= (\frac{\partial\psi}{\partial s} + \lambda_u^T\frac{\partial S}{\partial s} + \lambda_w^T\frac{\partial M}{\partial s} + \lambda_v^T\frac{\partial F}{\partial s})\delta s + (\frac{\partial\psi}{\partial u} + \lambda_u^T\frac{\partial S}{\partial u} + \lambda_w^T\frac{\partial M}{\partial u} + \lambda_v^T\frac{\partial F}{\partial u})\delta u \\ &+ (\frac{\partial\psi}{\partial w} + \lambda_u^T\frac{\partial S}{\partial w} + \lambda_w^T\frac{\partial M}{\partial w} + \lambda_v^T\frac{\partial F}{\partial w})\delta w + (\frac{\partial\psi}{\partial v} + \lambda_u^T\frac{\partial S}{\partial v} + \lambda_w^T\frac{\partial M}{\partial v} + \lambda_v^T\frac{\partial F}{\partial v})\delta v \\ &+ \delta\lambda_u^T S + \delta\lambda_w^T M + \delta\lambda_v^T F = 0 \end{aligned} \tag{6}$$

and resulting in the following highly nonlinear set of equations for the adjoint variables and the coupled system equilibrium.

$$\begin{bmatrix} \lambda_u \\ \lambda_w \\ \lambda_v \end{bmatrix} = \begin{bmatrix} \frac{\partial S}{\partial u} & \frac{\partial S}{\partial w} & \frac{\partial S}{\partial v} \\ \frac{\partial M}{\partial u} & \frac{\partial M}{\partial w} & \frac{\partial M}{\partial v} \\ \frac{\partial F}{\partial u} & \frac{\partial F}{\partial w} & \frac{\partial F}{\partial v} \end{bmatrix}^{-T} \cdot \begin{bmatrix} \frac{\partial \psi}{\partial u} \\ \frac{\partial \psi}{\partial w} \\ \frac{\partial \psi}{\partial v} \end{bmatrix} \tag{7}$$

Again, numerical solution techniques using a staggered scheme e.g. according to the nested analysis and design approach can be applied. Then, in the analysis level the equilibrium is guaranteed and the state variables are determined and used for linearization of the adjoint system.
It can be identified that for direct methods the amount of solution procedures depends on the number of design variables whereas the system of equations of the adjoint method has to be solved once and its size depends on the number of constraints. Therefore direct methods are used for problems with small number of design variables. On the contrary adjoint methods are used for high number of design variables and small number of equations. Independent of using direct or adjoint method the crucial point is the determination of the partial derivatives of the discrete state equations with respect to the state variables, which are necessary for solving for either the state variable derivative or the adjoint variables. In section 7.3 a solution method based on the structure sensitivity equation is presented.

7.2 A Numerical Sensitivity Analysis for FSI

In pure numerical sensitivity analysis the objective function gradient is computed using finite differences according to the common schemes like central, forward or backward finite differencing by calculating the objective function for reference and perturbed states. Since numerical effort increases quickly by extending the design space numerical method are used when only small numbers of design variables are needed.
The first approach presented in this work is a numerical method which exploits the possibility of doing finite differences by solving the coupled system for a reference and a perturbed state. The objective function gradient is computed as

$$\frac{d\psi}{ds} = \frac{\psi(u_{pert}, v_{pert}, w_{pert}, s+\varepsilon) - \psi(u_{ref}, v_{ref}, x_{ref}, s)}{\varepsilon} \tag{8}$$

The Accuracy of the gradient is very good in general and of course depends on the size of the perturbation ε, which is specific for the application and has to be carefully chosen by the user.
Figure 7 illustrates the technical implementation of the numerical method for iteration step i of the optimization procedure. x are the coordinates of the mechanical model, i.e. the finite element nodal coordinates, which depend on the design variables s. With a defined perturbation on the design variable the objective can be calculated on the resulting mechanical model by performing a FSI analysis. Knowing objective values for the reference and the perturbed state the gradient $\frac{d\psi}{ds}$ can be approximated. By shape description using mathematical relations like CAGD the dependency $\frac{dx}{ds}$ is known. Using mechanics motivated shape descriptions like form finding it generally has to be determined numerically.

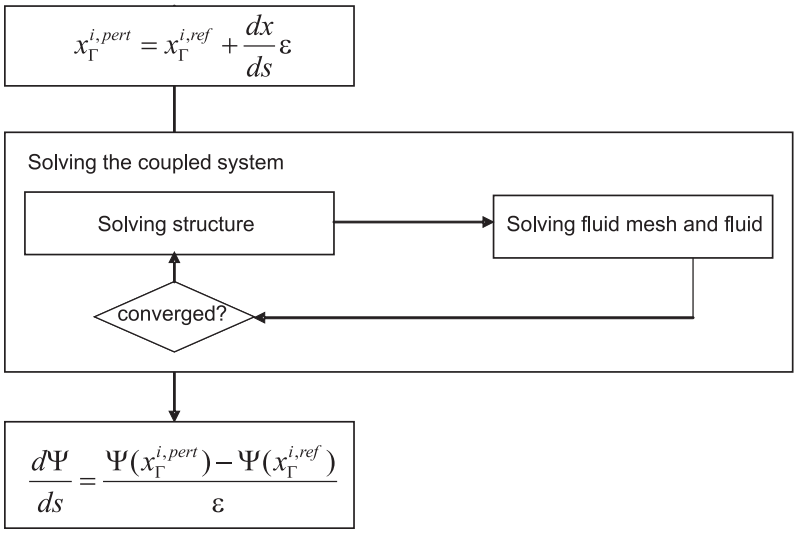


Figure 7: flowchart of numerical sensitivity analysis

7.3 A Direct Sensitivity Analysis based on Structural Sensitivity Equation

The second approach is a direct sensitivity analysis method developed for the structural analysis and optimization tool CARAT. The possibility to develop sensitivities on the structural side are far as possible and to use a fluid solver including fluid mesh as black box is exploited. Doing this the partial derivatives of fluid mesh and fluid state equations remain unknown and therefore the corresponding sensitivity equations are not available to solve for the derivatives of the state variables. Instead, an analysis tool to compute fluid mesh and fluid state variables w and v from the structural displacement u and the design variables s is applied. The equations to be solved reduce to the structural sensitivity equation (first equation of equation 4), which has the size of a structural solution and for a linear structure is written in term of the stiffness matrix K,

$$K\frac{du}{ds} = \frac{dK}{ds}u - \frac{dv}{ds} \tag{9}$$

The unknown $\frac{du}{ds}$ is computed in an iterative procedure illustrated in figure 8 for the optimization step i. The structural equation is derived semianalytically, using finite difference approximations for $\frac{dv}{ds}$ and $\frac{dK}{ds}$. The term $\frac{dv}{ds}$ could be obtained numerically by performing a FSI analysis for a perturbed state, but efficiency would not be gained in comparison with a numerical method. Therefore an iterative procedure reducing effort by performing a pure CFD analysis was developed. Actually, $\frac{dv}{ds}$ is obtained by using the coupled system steady state as a reference and by passing the displacement field, which is expected from a design perturbation ε to the fluid solver to compute a perturbed state: Using the state variable derivative $\frac{du}{ds}$ from the previous sensitivity analysis iteration step $j-1$ the displacement field with respect to a design change ε for the actual iteration step j can be predicted as:

$$u_\Gamma^{i,j,pert} = u_\Gamma^{i,j,ref} + \frac{du}{ds}^{i,j-1}\varepsilon \tag{10}$$

Since this displacement field results from a perturbed design the corresponding shape has to be passed additionally to the fluid solver:

$$x_\Gamma^{i,j,pert} = x_\Gamma^{i,j,ref} + \frac{dx}{ds}\varepsilon \tag{11}$$

With known displacements a pure CFD analysis gives the fluid state variables for calculating $\frac{dv}{ds}$ numerically as

$$\frac{dv}{ds} = \frac{v(u_\Gamma^{i,j,pert}, x_\Gamma^{i,j,pert}) - v(u_\Gamma^{i,j,ref}, x_\Gamma^{i,j,ref})}{\varepsilon} \tag{12}$$

and consequently the structural sensitivity equation

$$K^i\frac{du}{ds}^{i,j} = \frac{\partial K^i}{\partial x}\frac{dx}{ds}u^i - \frac{dv}{ds}^{i,j} \tag{13}$$

can be solved for the state variable derivative $\frac{du}{ds}$, which is used in the subsequent iteration step. To obtain good accuracy very few iterations are needed. With the converged value the gradient is computed according to equation 3.

7.4 Coupled Sensitivity Analysis

Special attention should be payed to the coupling within the sensitivity analysis resulting from the partitioned approach. It was mentioned before that the type of coupling data to be transferred between the non matching grids depends on the type of sensitivity analysis and that for the semianalytical approach the coupling data is derivative information. Using the sensitivity equations for a semianalytical sensitivity analysis the partial derivatives appearing in the equation are extracted from the respective codes and always correspond to the individual discretization of the domains. Solving according to the direct or the adjoint method requires a format defined by the sensitivity equations. Therefore sensitivities indicating structural behavior w.r.t. fluid variables and vice versa have to be transferred and transformed to the required size by the coupling tool CoMA according to equation 14. τ denotes the transfer operation on the common surface and the indices S, M and F indicate the discretization of the state variables corresponding to the respective fields.

$$\begin{bmatrix} \frac{\partial S}{\partial u_S} & \frac{\partial S}{\partial w_F} & \frac{\partial S}{\partial v_F} \\ \frac{\partial M}{\partial u_S} & \frac{\partial M}{\partial w_F} & \frac{\partial M}{\partial v_F} \\ \frac{\partial F}{\partial u_S} & \frac{\partial F}{\partial w_F} & \frac{\partial F}{\partial v_F} \end{bmatrix} = \begin{bmatrix} \frac{\partial S}{\partial u_S} & \tau(\frac{\partial S}{\partial w_S}) & \tau(\frac{\partial S}{\partial v_S}) \\ \tau^{-1}(\frac{\partial M}{\partial u_F}) & \frac{\partial M}{\partial w_F} & \frac{\partial M}{\partial v_F} \\ \tau^{-1}(\frac{\partial F}{\partial u_F}) & \frac{\partial F}{\partial w_F} & \frac{\partial F}{\partial v_F} \end{bmatrix} \tag{14}$$

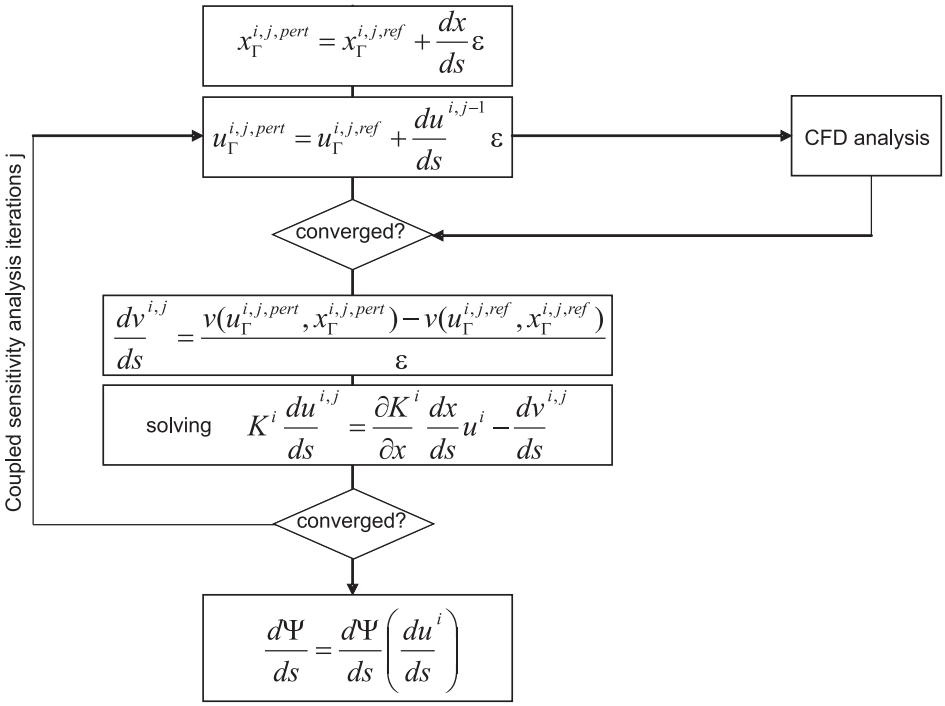


Figure 8: flowchart of direct sensitivity analysis based on structural sensitivity equations.

8 Applications

8.1 Designing a Half Sphere Shell Structure subject to Fluid Flow

For verification of the framework, for testing and comparing the numerical and direct sensitivity analysis a first academic example was computed. It is a linear half sphere shell structure in a viscous laminar flow ($Re = 25$). The structure is computed with CARAT using shell elements, the fluid flow is computed with OpenFoam and the coupling is performed with CoMA. The system setup is shown in figure 9. A steady state flow is reached deforming the structure as shown in figure 10 scaled by the factor 5.

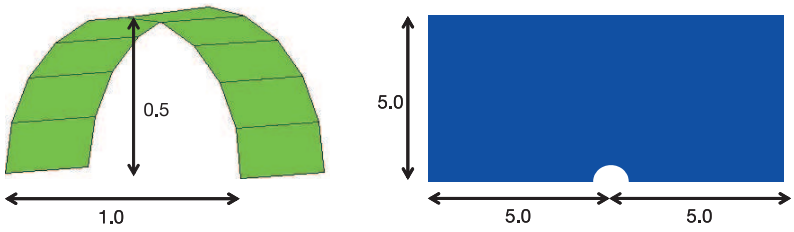


Figure 9: Setup of a half sphere shell structure in fluid flow

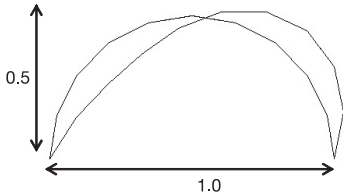
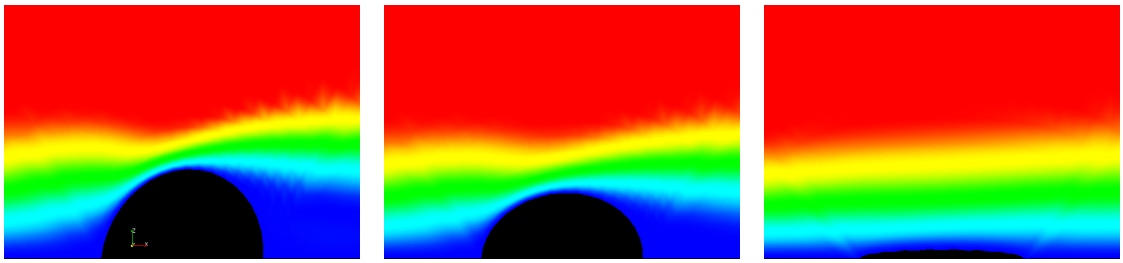


Figure 10: Deformation of a half sphere shell structure in fluid flow

The structure design is described by the CAGD concept using one single design element representing the whole geometry of the structure. The design element is controlled by one parameter, the total height of the structure. Thus, the finite element nodes which always lie on the design elements surface are linked to this CAGD parameter. Modifying the height parameter the finite element nodes move in the same ratio as the highest point in vertical direction, causing a flattening or a growth of the shell.
The CAGD parameter is chosen as optimization variable. The gradients of the objective function were computed by numerical and semianalytical direct methods and achieve the same results. Results for two different objective functions are shown in figure 11: The vertical displacement of the highest point of the structure is chosen as a structure based objective and the drag force as a flow based objective. Figure 11 (a) shows the initial design.
For the reduction of the vertical displacement the objective reduces with decreasing height of the structure. A reduction of 35% after 25 optimization iterations is achieved. The converged state is shown in figure 11 (b) when the lower box constraint which restricts the optimization variable and therefore the flattening of the structure to a minimum height are reached.
For the drag reduction the lower box contraint is set to zero height, because the optimum is expected to be found here. It is obvious that the drag is minimal for a totally flat structure. 83% reduction is reached after 80 optimization iterations. The design corresponding to this state is shown in figure 11 (c).



(a) (b) (c)

Figure 11: Minimization of vertical displacement and drag: initial design (a), optimum designs for displacement (b) and drag (c).

8.2 Designing a Four Point Tent Membrane Structure subject to Fluid Flow

In the following, form finding is embedded in the previous framework and the procedure is presented by an example, a four point tent membrane structure. The tent is modelled with membrane elements for the surface and truss elements for the cables. The initial geometry for the optimization loop is obtained with form finding from the geometry shown in figure 12. Such a procedure is robust and only 10 iterations are needed to obtain a converged shape (figure 12).
The objectives in this case are the displacement at a node, the $\mathcal{L}^2$ and the $\mathcal{L}^\infty$ norm of the structural displacements. The respective design variable is the vertical position of the heighest nodes which remains equal during optimization for sake of symmetry.

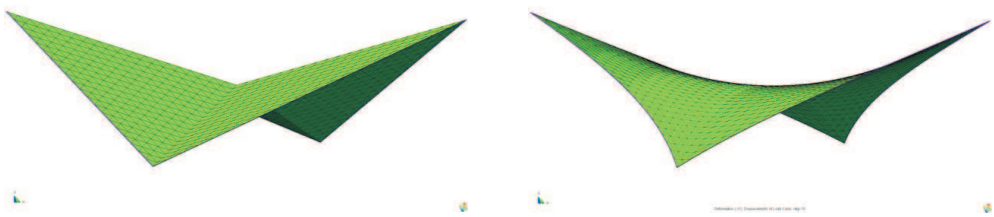


Figure 12: Initial and converged configuration of form finding of four point tent

In the structure side a geometric nonlinear analysis is performed. On the other hand, the fluid flow can be considered laminar ($Re < 1000$) and the SIMPLE algorithm is used to solve the Navier-Stokes equations for the steady state solution. The structural deformation, the flow field and the pressure on the structure's surface resulting from the FSI analysis of the initial design of the structure is shown in figure 13.

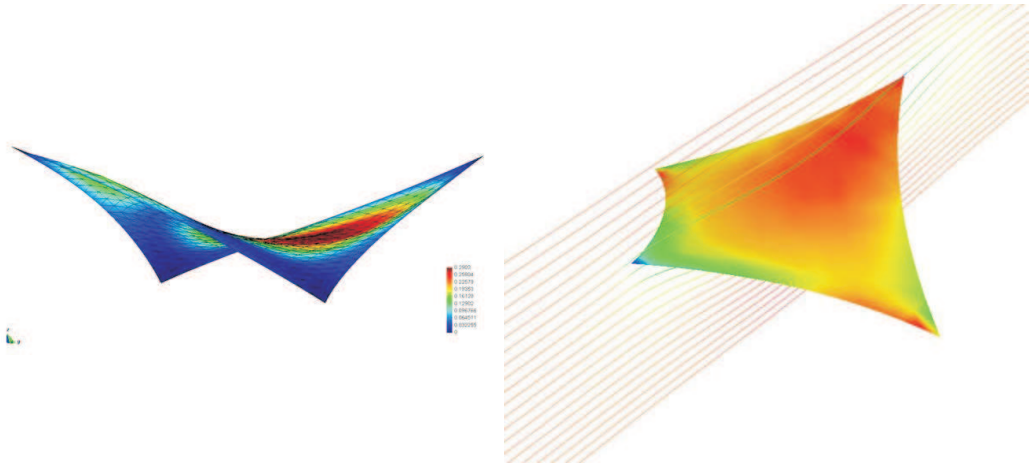


Figure 13: Four point tent in fluid flow: Deformation caused by flow, fluid velocity and surface pressure

For the sensitivity analysis the numerical approach is used (section 7.2). The numerical approach is very well suited in such a framework since the number of design variables is small and form finding is used to determine the final geometry as described in the sequence. After sensitivity analysis the design variable (position of highest nodes) is updated and the resulting shape is shown in the figure 14. In order to obtain a physical shape a mechanical correction is needed. For this reason form finding is applied resulting in the mechanically correct shape.

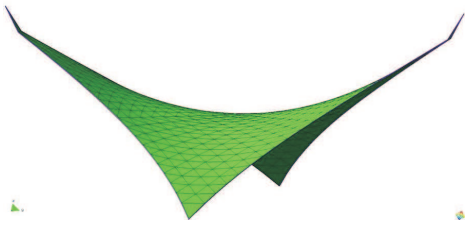


Figure 14: Update of design variables of four point tent

The results of an optimization with the $\mathcal{L}^2$ norm of the total displacement as objective is shown in figure 15. The objective is reduced with increasing height of the structure. Since the structural behaviour becomes stiffer with increasing curvature this result is intuitive. It can be seen that the solution converges to a non trivial optimum. This is because the objective increases for higher side faces of the structure, which are very sensitive to fluid loads.

A second membrane example was carried out using the form finding method for shape optimization. Figure 16 shows the real structure called ”Tanzbrunnen” realized in Cologne by Frei Otto and the optimized structure with the displacement field for the reduction of the $\mathcal{L}^2$ norm of the structural displacements by modifying the position of the corner nodes.

9 Outlook

The optimization framework can be enhanced by adding more complex methods in sensitivity analysis. Higher efficiency can be achieved using partial derivatives of fluid mesh and fluid. Therefore special functions have to be implemented in the fluid code and a new solution method for the sensitivity equations is required. Furthermore, when CAGD free approaches will be used and systems with many design variables arise the adjoint sensitivity formulation has to be implemented.
Since coupled sensitivity analysis belongs to both fields, fluid and structure, the coupling tool CoMA will be enhanced to control the coupled sensitivity analysis.

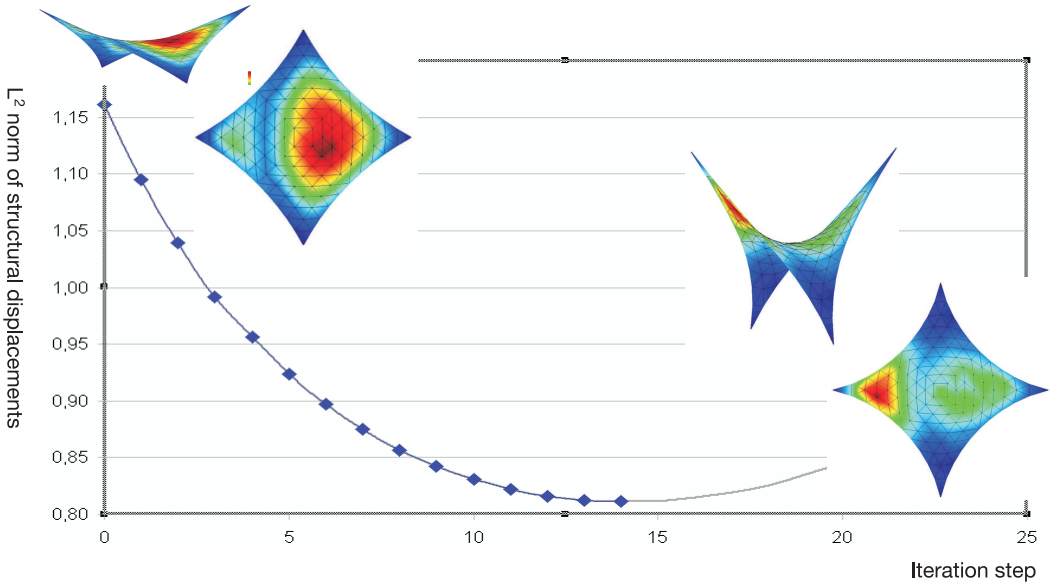


Figure 15: Reduction of $\mathcal{L}^2$ norm of structural displacements of four point tent

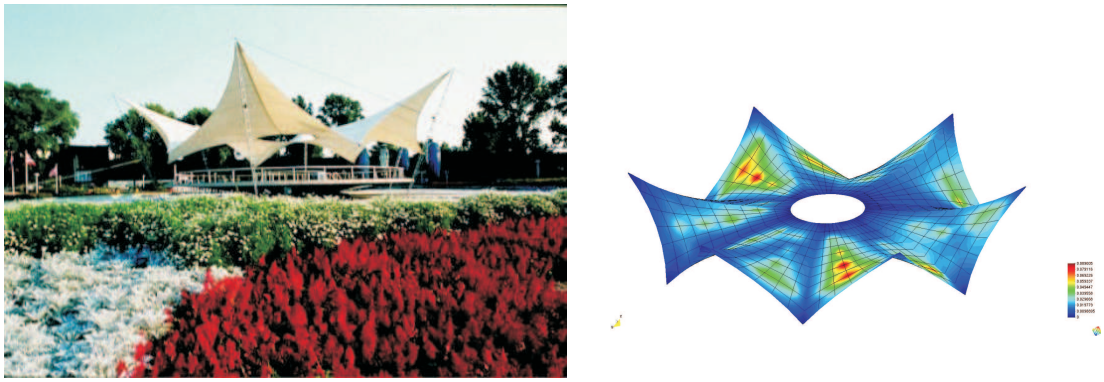


Figure 16: Real structure (taken from (Otto, 2008)) and the optimized structure in fluid flow (from the left) with displacement field for the reduction of the $\mathcal{L}^2$ norm of the structural displacements of the "Tanzbrunnen"

10 Conclusions

In this contribution a method for gradient based optimization using partitioned FSI schemes was presented. For the sake of maximum design freedom in structural shape various methods suited for different amounts of design variables were discussed. Moreover, the need for different shape description methods and therefore for several methods in sensitivity analysis was pointed out. The special advantage of embedding the form finding method into the optimization procedure was highlighted. Finally, a framework for gradient based optimization in FSI according to the NAND approach developed for including a variety of algorithmic approaches was presented and it's abilities were shown by illustrative examples.

11 Acknowledgment

This project is part of the German Research Unit 493: Fluid-Structure Interaction: Modeling, Simulation, Optimisation (DFG-493, 2008) funded by the Deutsche Forschungsgemeinschaft (German Research Foundation).

References

Arora, J.; Wang, Q.: Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30, (2005), 251–272.

Barcelos, M.; Bavestrello, H.; Maute, K.: A schur-newton-krylov solver for steady-state aeroelastic analysis and design sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006), 2050–2069.

Barcelos, M.; Maute, K.: Aeroelastic design optimization for laminar and turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 197, (2008), 1813–1832.

Bletzinger, K.-U.; Firl, M.; Linhard, J.; Wüchner, R.: Optimal shapes of mechanically motivated surfaces. *Computer Methods in Applied Mechanics and Engineering*, in press, doi:10.1016/j.cma.2008.09.009.

Bletzinger, K.-U.; Gallinger, T.; Kupzok, A.; Wüchner, R.: Partitioned strategies for optimization in FSI. In: *European Conference on Computational Fluid Dynamics (ECCOMAS CFD)* (2006).

Bletzinger, K.-U.; Wüchner, R.; Daoud, F.; Camprubi, N.: Computational methods for form finding and optimization of shells and membranes. *Computer Methods in Applied Mechanics and Engineering*, 194, (2005), 3438–3452.

DFG-493: Forschergruppe 493: Fluid-Struktur-Wechselwirkung: Modellierung, Simulation, Optimierung: http://fsw.informatik.tu-muenchen.de/index.php (2008).

Etienne, S.; Pelletier, D.: A general approach to sensitivity analysis of fluid-structure interactions. *Journal of Fluids and Structures*, 21, (2005), 169–186.

Farhat, C.: *CFD-based nonlinear computational aeroelasticity, Encyclopedia of Computational Mechanics 3 (E. Stein, R. Borst and T.J.R. Hughes, eds.)*, chap. 13, pages 459–480. John Wiley & Sons, Ltd. (2004).

Farhat, C.; Lesoinne, M.; LeTallec, P.: Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157, (1998), 95–114.

Felippa, C. A.; Park, K. C.; Farhat, C.: Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190, (2001), 3247–3270.

Ferziger, J.; Perić, M.: *Computational Methods for Fluid Dynamics*. Springer New York (1999).

Gallinger, T.; Kupzok, A.; Israel, U.; Bletzinger, K.-U.; Wüchner, R.: A computational environment for membrane-wind interaction. In: *Fluid-Structure Interaction. Theory, Numerics and Applications* (2008).

Haftka, R.; Grandhi, R.: Structural shape optimization - a survey. *Computer Methods in Applied Mechanics and Engineering*, 57, (1986), 91–106.

Haftka, R.; Sobieszczanski-Sobieski, J.; Padula, S.: On options for interdisciplinary analysis and design optimization. *Structural Optimization*, 4, (1992), 65–74.

Israel, U.; Kupzok, A.; Lähr, A.; Gallinger, T.; Wüchner, R.; Bletzinger, K.-U.: Eine flexible Kopplungsschnittstelle für partitionierte Berechnungen von Mehrfeldproblemen. In: *19. Forum Bauinformatik*, pages 61–68 (2007).

Jameson, A.: Optimum aerodynamic design using CFD and control theory. In: *AIAA Computational Fluid Dynamics Conference, 12th, San Diego*, pages 926–949 (1995).

Jameson, A.: *Aerodynamics, Encyclopedia of Computational Mechanics 3 (E. Stein, R. Borst and T.J.R. Hughes, eds.)*, chap. 11, pages 325–406. John Wiley & Sons, Ltd. (2004).

Jasak, H.; Tukovic, Z.: Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30, (2007), 1–18.

Löhner, R.: *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods, 2nd Edition*. Wiley (2008).

Lund, E.; Møller, H.; Jakobsen, L.: Shape design optimization of stationary fluid-structure interaction problems with large displacements and turbulence. *Struct. Multidisc. Optim.*, 25, (2003), 383–392.

Maute, K.; Nikbay, M.; Farhat, C.: Coupled sensitivity analysis and optimization of three-dimensional non-linear aeroelastic systems. *AIAA Journal*, 39, (2001), 2051–2061.

Maute, K.; Nikbay, M.; Farhat, C.: Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56, (2003), 911–933.

Mohammadi, B.; Pironneau, O.: Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36, (2004), 255–279.

Otto, F.: www.freiotto.com (2008).

Pironneau, O.: *Optimal shape design for elliptic systems*. Springer, New York (1984).

Sobieszcanski-Sobieski, J.: Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28(1), (1990), 153–160.

Sobieszczanski-Sobieski, J.; Haftka, R.: Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*, 14, (1997), 1–23.

Soto, O.; Löhner, R.: CFD shape optimization using an incomplete-gradient adjoint formulation. *International Journal for Numerical Methods in Engineering*, 51, (2001), 735–753.

Thévenin, D.; Janiga, G.: *Optimization and Computational Fluid Dynamics*. Springer Berlin (2008).

Wall, W.: *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. Ph.D. thesis, Institut für Baustatik der Universität Stuttgart (1999).

Wüchner, R.: *Mechanik und Numerik der Formfindung und Fluid-Struktur-Wechselwirkung von Membrantragwerken*. Ph.D. thesis, Lehrstuhl für Statik der TU München (2006).

Wüchner, R.; Bletzinger, K.-U.: Stress-adapted numerical form finding of pre-stressed surfaces by the updated reference strategy. *International Journal for Numerical Methods in Engineering*, 14(2), (2005), 131–145.

Wüchner, R.; Kupzok, A.; Bletzinger, K.-U.: A framework for stabilized partitioned analysis of thin membrane-wind interaction. *International Journal for Numerical Methods in Fluids*, 54, (2007), 945–963.

Address: U. Israel, Lehrstuhl für Statik, Technische Universität München, 80290 München, Germany.
email: `israel@bv.tum.de`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 171– 178
Herrsching am Ammersee, 29.9.-1.10.2008

Advanced Software Engineering for FSI Applications

Dominik Jürgens

Engineering multi-physical systems is a challenging task, which requires the simulation of interacting dynamical systems governed by differing model equations. For the development of a simulator for coupled analysis of such phenomena, reuse of already available solvers for the subsystems is often desirable. Against the common goal of using solvers for the subsystems as black-boxes — which aims at a common coupling methodology shared by the FSI community — often both programs need to be in depth understood by the coupling research group and black-box solvers need to be opened. In this article, the idea of a skeleton based architectural software design is proposed, which aims at conserving separation of concerns, so that subsystem solvers remain black-boxes. A goal of the proposed architecture is a deeper separation of coupling algorithms from technical concerns. In contrast to former approaches in this domain, we do not focus on providing a precast solution; we aim at highly parametrised and therefore flexible design patterns. The long term objective is to provide a flexible and open template framework for scientists in this field to accelerate the development of individual coupled simulators for scientific applications.

1 Component Based Design in FSI Applications and its Intrinsic Limitations

This section reviews general functional decomposition of software systems and discusses its application in coupled simulations. After this general discussion we will identify intrinsic properties of present-day coupling solutions and give rise to our motivation for proposing a paradigm shift in the development of coupling algorithms.

1.1 Short History on Functional Decomposition

To reuse software is a very fundamental idea. The development of reusable algorithms started with the development of assembler languages, which are the human readable form of machine programs. Later programming paradigms became independent of the machines, and compilers were used to transform conceptional statements into executable machine programs. This approach made it possible to write programs in a coherent way, hiding the instruction encoding. These higher programming languages provide predefined structures for general purpose programming tasks; for example loops or parametrised subprograms are provided. These abstractions hide technical details as register allocation, naming and jumps in the program sequence; they make programming less error prone and the resulting programs better understandable.

The introduction of support for explicit interfaces into several programming languages extended the possibility of making subsystems exchangeable (for a discussion of this aspect see Coulange (1998)). A superordinate concept is *programming with language independent components*, provided by software component frameworks (see Szyperski (1997)). These black-box paradigms support a functional decomposition of complex systems into independent subsystems. It is applicable if considered subsystems deal with a certain functional task, for which a specific common interface can be defined for a certain class of subsystems.

An interface for making a software subsystem reusable can only be defined, if —metaphorically speaking — its *contact surface* does not need to be changed to provide full functionality in different contexts; the *interface* of the subsystem to its environment has to be constant. To efficiently compose software from black-boxes, a point of minimal effort in composition *energy* must be found[1]. If a program provides functionality which varies in a functional sense, a constant interface can often not be found and a black-box view on the software can therefore

[1] Here *energy* means communication, programming and computational overhead.

not be provided. Reusing such codes in a classical programming language with its focus put on functional system composition is most likely inefficient or not maintainable; the code has to be reimplemented. A discussion of problems in functional decomposition of software systems can be found in Czarnecki and Eisenecker (2000).

An example for FSI code that is context dependent when it is written in a classical programming languages, is code for time-integration; it is reimplemented in every coupling project and not available as library function. Reusing such code may be possible in theory, but different data structures and different procedures to gain coupling data and to control the overall process make it hard to implement it as library function.

1.2 Relationship Between Partitioned Analysis and Functional Decomposition

Partitioned analysis and the so-called *partitioned approach* to coupled simulation considers available simulators as black-boxes and introduces a coupling procedure between them. In simulation of time-dependent problems implicit time-integration schemes can be used to achieve a physically strong coupling between the partitions (see for example Matthies et al. (2006b)).

The term black-box generally describes a system which can only be accessed through an interface that hides details of the system implementation. In FSI the term black-box is used as a synonym for software which provides finite approximations of the physical behaviour of the considered partitions; its black-box interface is often not considered and does not appear explicitly in simulator or coupling code. This abstract view with a *virtual* interface often results in entanglement of codes and cannot obscure the fact that many little details are to be considered and a lot of technical problems are to be solved to transform CFD or CSD code into a real black-box. The cost of making a solver a black-box depends on the realisation of the respective solver and on the flexibility it provides. For example Kassiotis and Hautefeuille (2008) present how the FEM code *FEAP* was encapsulated into a software component called *CoFEAP*; the solution allows to use FEAP through a black-box interface and the use of component instances over a computer network. CoFEAP provides an interface which is flexible enough to be used in different contexts, it is not specialised for FSI applications (as it would be needed to use MpCCI, see Fraunhofer (2008b)).

Software component technology allows to create software-black-boxes with context independent interfaces. Therefore given source-code is connected with an explicitly defined interface. Software components have been successfully applied in the development of customised solvers for partitioned analysis (see Matthies et al. (2006a)). In this way, modern paradigms from computer science are already applied in the development of coupled simulations. For projects in this context, advances in knowledge on abstract component based software systems can help to understand and solve intrinsic problems in the domain of FSI simulation.

1.3 Why Coupling is not a Functional Thing

The goal of software components is to make algorithms reusable and exchangeable at the same time. These goals are also desirable for coupling code, but coupling procedures would have to become black-boxes. To ask for a coupling black-box, or to define a general interface for any coupling procedure, appears therefore attractive. In order to find such general interface, many questions have to be ultimately answered: Can we find a fixed set of parameters for a coupling black-box, independent of the order of differential coupling on the coupling interface? Is the black-box interface independent of the provided coupling algorithm?

It turns out, that introducing a coupling interface implies a reduction of flexibility and applicability of its black-box implementations: this is not meaningful for a domain of active research. A functional black-box abstraction for coupling procedures is counterproductive. Here a dilemma of FSI research begins; a black-box for coupling simulators accelerates the development of customised software for specific simulation tasks — which necessarily occur in research — but a precast black-box solution is not flexible enough to be efficiently applicable in research, where the focus is put on development of new algorithms.

Such questions are common to the development of black-box algorithms from any domain; to consider a system as a black-box is not reasonable if its interface is context dependent or under development (and therefore not constant).

1.4 Limits of Current Approaches

The Mesh-based parallel Code Coupling Interface (MpCCI) provided by Fraunhofer (2008a) is an elaborated tool for coupling simulation codes. The methodology to couple codes with MpCCI can be compared with the one provided by the message passing interface (MPI). MPI provides a single program multiple data view (SPMD) on parallel software. The control flow of MPI-programs is distributed over the parallel instances which makes parallel MPI-programs hard to understand, debug and maintain as, for example, Eidson et al. (2003) discussed.

MpCCI extends the functionality of MPI through interpolation support; i.e. it provides mesh-based data exchange with interpolation. MpCCI and equally MPI require explicit send and receive calls in the communicating codes, in order to realise interaction between the distributed subprograms. This paradigm enforces therefore a deep knowledge of the code, which needs to be modified to become a parallel or coupled program, respectively. In Fraunhofer (2008b) the architecture of MpCCI is described. The interface to simulator code is realised by so-called adaptors, which integrate a certain solver into the MpCCI framework. Thereby any MpCCI-adaptor provides the same predefined interface, which makes the simulators exchangeable. These interfaces are not context independent, but specifically designed to match the requirements of coupled simulation.

In contrast to a black-box interface, the MpCCI interface does not make the solver context independent; it provides an interface for the specific context of coupling. Our focus with the approach presented in this article is to develop a hierarchy of context independent simulator interfaces, which can be ubiquitously used for a larger set of applications. In MpCCI a separation of concerns in this general sense is not supported. Instead simulation providers have to change their code to provide a MpCCI interface, and they have to change it again for any other use. Simulator reuse is not only suggestive in FSI, also optimisation, uncertainty quantification and domain decomposition are fields of application. A generic approach for simulator coupling would help to create a common basis of knowledge and technology for all of the mentioned fields.

Furthermore, MpCCI is a tool for the specific task of transferring grid-based data between two solvers; it does neither provide higher functionality like implicit time-integration schemes nor the support for customised interpolation or contact algorithms. MpCCI is a precast solution, not focusing on the support of the development of new coupling algorithms, which in contrast to this is a major goal of the architecture we aim for.

2 Skeleton Based Design for Reusable Coupling (Sub-)Algorithms

In this section we first abstractly discuss skeleton based software design, then we present possible applications in FSI and finally we discuss advantages of the proposed ideas.

2.1 The Idea of Algorithmic Skeletons

Classical programming languages provide general purpose abstractions like *loops* or *procedures*. In contrast, skeletons can be used to model and provide special purpose abstractions. A skeleton is a blueprint of an algorithm providing *slots*, where user-defined code can be *plugged in* to specialise the abstract specification so that it matches specific requirements. The skeleton approach to algorithm design is based on the assumption that similar applications require similar algorithms or at least similar algorithmic structure. An implication of skeleton based design is a decomposition of the overall system into functional subsystems. These functional subsystems are not unique to a specific skeleton, but can be used anywhere, where their function is required. Thereby skeletons assist code-reuse, they enable customisation, testing of individual subsystems and provide intrinsic support for parallelisation (see Darlington et al. (1993)).

2.2 Example for an Algorithmic Skeleton

A facility to systematically modify code is not provided in classical programming languages; the modification of a program cannot be expressed with functional abstractions. Skeletons provide a programming interface to code-generators. An illustrative example is a *divide and conquer* skeleton. Divide and conquer is a general pattern in algorithm design, where a task is decomposed recursively into independent subtasks. The recursive decomposition of the problem can be made until the subproblem — which is to be solved — is trivial and can be solved directly. To construct the overall result, the results of the subtasks have to be merged in a final step.

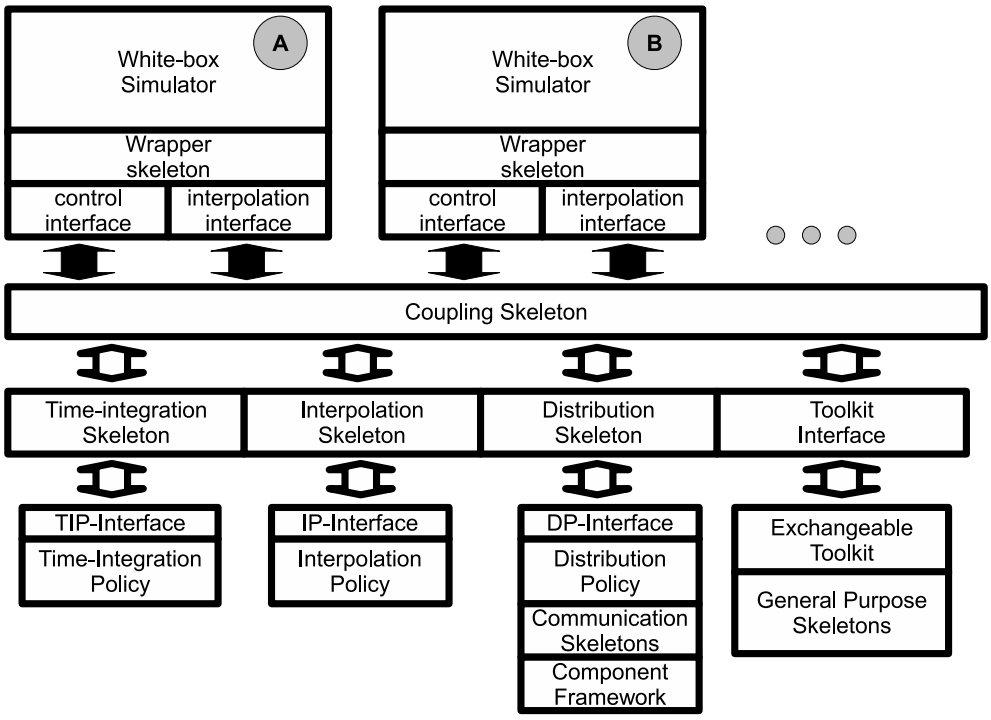


Figure 1: The figure shows a possible structure of a skeleton based architecture for coupling simulators. The skeletons represent abstract algorithms providing slots to plug-in specific behavioural modules. These modules as well as the skeletons provide explicit interfaces which make them exchangeable. Instead of reimplementing algorithms in different projects, abstract algorithms may be provided and reused.

Abstractly seen, divide and conquer requires five parameters (or slots) which are mostly functional:

$a \to Bool$ decide whether a task is trivial or not

$a \to b$ directly solve a trivial problem

$a \to (a_1, a_2, ...)$ break down a nontrivial task into smaller subtasks

$(b_1, b_2, ...) \to b$ merge previously computed results

a the task to be solved

The last mentioned parameter "a" is the only parameter in a classical sense, the other parameters are functional; they change the behaviour of the generated program. The application of this skeleton is demonstrated in Darlington et al. (1993). In this paper the skeleton is used to generate code which can be executed on a parallel machine without the need for any knowledge of parallel programming.

We can think of a skeleton as a template function with functional parameters. In classical programming languages functional parameters are technically provided by function-pointers. But function-pointers have certain restrictions with respect to performance and flexibility. Furthermore a systematic behavioural parametrisation of a program cannot be efficiently modelled with the help of function pointers. Classical languages do not provide functions or types with *behavioural parameters* which can be used to modify their precast implementation. Algorithmic skeletons provide such mapping in a high-performance and flexible way because code-generation supports *inlining* and *compiler optimisation.*

2.3 Algorithmic Skeletons in FSI Applications

In FSI applications a number of specific skeletons have to be provided for different parts of the overall system, one of them is a skeleton for a data transfer component similar to MpCCI (Fraunhofer (2008a)). Instances of the

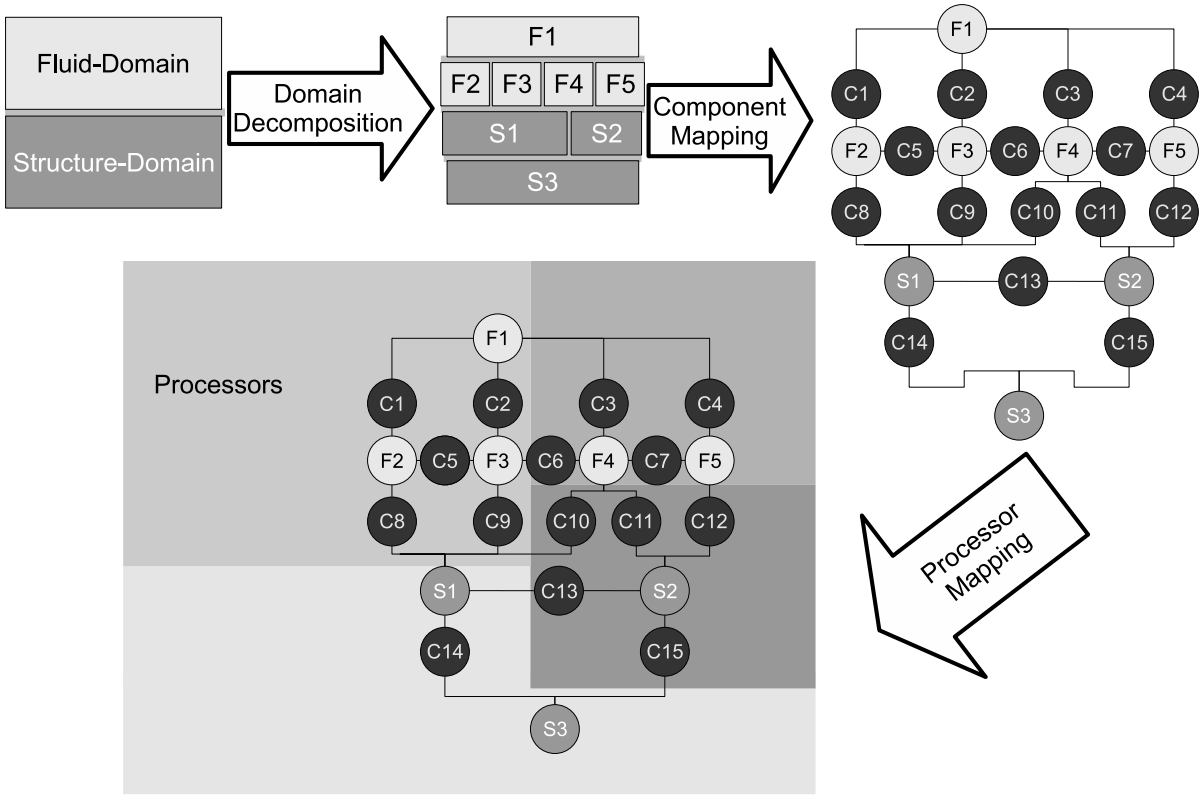


Figure 2: The Figure shows a possible decomposition of a coupled simulation with a fluid and a structure domain. In a domain decomposition step the domains are partitioned for optimising load balancing. Each partition is then modelled by a separate simulator component (F1..5 are the fluid subdomains and S1..3 are the structure subdomains, respectively). The simulator components are independent and can even implement different models. Instances of the coupling components C1..14 — which are implemented with the help of the presented skeleton — are created to couple respective fields. These coupling components do not need to be homogeneous, they may implement different algorithms. The mapping of the components to the processors in the last step can be done arbitrarily, which is a property of the software component approach. Dynamic allocation of the resources for optimised load balancing can be realised by using *component migration* to dynamically move a running component to another processor. This is an interesting feature on machines with non-uniform memory architecture.

skeletons we aim for — where its provided *slots* are occupied by specific implementations — may provide the same functionality as MpCCI. Nonetheless it is only a special case. Figure 1 shows a layout sketch of a skeleton based architecture for coupled simulation, where modules for functional slots are termed as *policies*. Skeletons in this architecture are used to provide code generators for recurrent programing tasks in simulator coupling:

- Writing wrapper code for a specific simulator to provide a coupling interface
- Implementing a time-integration scheme
- Implementing an interpolation scheme
- Managing distributed instances and connect solvers with common boundaries

By providing an architecture instead of a precast solution, customisations become implicitly supported; in this way the coupling algorithm is modelled as white-box and only common structure is fixed.

2.3.1 General Architecture for Coupled Simulation on Distributed Memory Machines

An important feature of MpCCI is an optimal communication strategy in distributed coupled simulations, where communication only occurs between processors of coupled domains (see Fraunhofer (2008b)). In a skeleton based component architecture, this goal can also be achieved and furthermore, general domain decomposition algorithms or a coupling of different models — i.e. potential flow in the far field, incompressible flow near the boundary layer — may be formulated in a natural way. Domain decomposition can be considered as a special case of coupled simulation, where domains are homogeneous — i.e. the meshes and models match.

Therefore, the context independent component interfaces we focus on can be used to implement a general parallel coupling environment. Assuming that a meaningful partitioning of the domain is given, each subdomain can be simulated in a separate simulator component. The general idea is demonstrated in figure 2.

2.4 The Vision: Plug-and-Play Algorithms

At this point we want to present further advantages of the proposed skeleton based coupling architecture. If new algorithms like higher order time-integration schemes can be provided as skeletons, research groups would be able to exchange coupling algorithms in their software simply, by using a different skeleton or policy. A broader scientific exchange would be possible and individual research projects may be able to focus on specific tasks, which results in a better separation of concerns. It becomes possible for a group to publish achievements in a reusable way, so that other groups can take the provided solution and plug it into their project code. Reimplementations of algorithms — necessary in classical programming — can be avoided.

The need for reimplementing code in classical programming has a number of disadvantages for algorithm development; time is consumed in reimplementing the code, testing it, and finally in integrating the new algorithm into the existing project code. Skeletons instead can be verified and requirements they put to their context can be statically asserted by introducing black-box interfaces between respective subsystems. The explicit interfaces allow the respective black-box subsystem to be tested for assert certain predefined semantics, so that quality enhancements and better maintainability of the software can be achieved. Finally, scientific results become better reproducible in that way.

3 Outlook and future work

Skeleton based software is not a new technology (see Cole (1989)); today it is technically implemented by technologies like C++-templates (see *policy-based design* in Alexandrescu (2001)). The problem with these technologies is that they are complex and therefore hard to understand and handle.

The goals of the envisioned skeleton based architecture can only be achieved if the methodology is simple and does not enforce users to become specialists in this field. It is the same as with general purpose programming languages: users do not want (and need) to understand how the compiler builds assembler code to implement a `for`-loop. For that reason, the mightiness of complex technologies like C++-templates has to be reduced to a level of abstraction which is specific enough to model FSI-algorithms. For achieving this goal, a programming language providing exactly the mentioned special purpose abstractions is to be developed.

For providing a user-interface to the code generation process, a domain specific language (DSL) (see Sloane et al. (2003)) is to be defined. The technical realisation of the skeletons can be done with the help of C++-templates, which provide abstractions for programming general purpose code-generators. The envisioned DSL will help to hide the complexity of the template-based implementation from the user of the algorithmic skeletons.

3.1 The Anticipated Paradigm-Shift

The paradigm-shift we anticipate is an introduction of a new concern into coupled simulation. The concerns which are already established in the scientific community are mathematics, physics and engineering. Computer science, as another important contributor, is by now involved by providing general purpose programming constructs like standard programming languages or by providing precast black-box solutions like MpCCI. The goal of providing specialised abstractions by developing new languages is not yet followed.

With our future work, we want to open this door to provide special purpose abstractions and a specialised programming language, to provide a higher degree of flexibility in supporting algorithm development. We think building a common basis of algorithmic knowledge is very important in an area of active research. A positive effect of skeletons compared to programming technologies as C++-templates is their formal definition; skeleton contributions of research groups can be described in the common language of mathematics, without having to describe things in natural or pseudo languages.

Acknowledgement Thanks to Rainer Niekamp for his time; let our discussions be as fruitful as in the past.

References

Alexandrescu, A.: *Modern C++ design: generic programming and design patterns applied*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2001).

Cole, M.: *Algorithmic Skeletons: Structured Management of Parallel Computation*. Research Monographs in Parallel and Distributed Computing. Pitman (1989).

Coulange, B.: *Software Reuse*. Springer Verlag, London (1998).

Czarnecki, K.; Eisenecker, U. W.: *Generative programming: methods, tools, and applications*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000).

Darlington, J.; Field, A. J.; Harrison, P. G.; Kelly, P. H. J.; Sharp, D. W. N.; Wu, Q.; While, R. L.: Parallel programming using skeleton functions. In: A. Bode; M. Reeve; G. Wolf, eds., *PARLE '93: Parallel Architectures and Languages Europe*, pages 146–160, Springer-Verlag, Berlin, DE (1993).

Eidson, T.; Dongarra, J.; Eijkhout, V.: Applying aspect-orient programming concepts to a component-based programming model. In: *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 205.2, IEEE Computer Society, Washington, DC, USA (2003).

Fraunhofer: MpCCI – Multidisciplinary Simulation through Code Coupling, `http://www.mpcci.de/` (2008a).

Fraunhofer: MpCCI 3.0.6-21 Documentation (2008b).

Kassiotis, C.; Hautefeuille, M.: *coFeaps Manual* (2008).

Matthies, H. G.; Niekamp, R.; Srisupattarawanit, T.: Scientific computing with software-components. *IACM Expressions*, 19, (2006a), 27–31.

Matthies, H. G.; Niekamp, R.; Steindorf, J.: Algorithms for strong coupling procedures. *Comp. Meth. Appl. Mech. Engrng.*, 195, (2006b), 2028–2049.

Sloane, T.; Mernik, M.; Heering, J.: When and how to develop domain-specific languages (2003).

Szyperski, C.: Component software - beyond object-oriented programming (1997).

Address: Institute of Scientific Computing, TU Braunschweig, Hans-Sommer-Strae 65, Braunschweig.
email: `d.juergens@tu-bs.de.`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 179– 188
Herrsching am Ammersee, 29.9.-1.10.2008

On Some Open Problems in Fluid-Structure Interaction

Rainald Löhner, Joseph D. Baum and Orlando A. Soto

Key Words: *Fluid-Structure Interaction*

ABSTRACT

We consider some of the open problems in fluid-structure interaction:

- The lack of 'optimistic' proofs about the order of temporal convergence of the coupled system if staggered or loose coupling techniques are used;
- The lack of a systematic and unified treatment of information transfer between structural models that are of lower order than the surface of the fluid domain;
- The shortcomings in techniques for the treatment of large body/surface motion within flow solvers; options presently advocated are either embedded/immersed techniques that are poorly suited for RANS applications, or ALE/body fitted techniques that are poorly suited for applications with change of topology and/or crack formation; and
- The achievable accuracy for coupled nonlinear problems, and the implications this has for numerical methods.

1 Introduction

Over the last three decades, the trend in each of the individual disciplines that are required in order to predict the behaviour of processes or products - fluid dynamics, structural mechanics, combustion, heat transfer, control, acoustics, electromagnetics, etc. - has followed the typical bottom-up direction. Starting from sufficiently simple geometries and equations to have an impact and be identified as 'computational', more and more realism was added at the geometrical and physics level. While the engineering process (Figure 1) follows the line: project, objectives, optimization goals, discipline, problem definition, gridding, solution of the PDE and evaluation, the developments (in particular of software) in the Computational Sciences tend to run in the opposite direction: once a solver is developed, grid generation becomes the next bottleneck. With automatic grid generation, the fast preparation of run-time data (i.e. the link CAD - problem definition - grid generation) requires attention. Once the workflow in an individual discipline has been sufficiently automated, the link to other disciplines becomes attractive. And as multi-disciplinary runs become routine, we may envision complete multidisciplinary optimization and project management. With the advancement of numerical techniques and the advent first of affordable 3-D graphics workstations and scalable compute servers, and more recently of personal computers with sufficiently large memory and 3-D graphics cards, public domain and commercial software for each of the 'computational core disciplines' has matured rapidly and received wide acceptance in the design and analysis process. Most of these packages (9) are now at the threshold mesh generator: pre-processor. This has prompted the development of the next logical step: multi-disciplinary links of codes, a trend that is clearly documented by the growing number of publications and software releases in this area.

The desire to solve multidisciplinary problems can not only be explained by the maturity of discipline codes. Indeed, the biggest payoffs expected from multidisciplinary analysis are increased **insight** into complex physical phenomena and industrial processes, leading to a more comprehensive **optimization** of products and processes.

The space of possible approximations for coupled fluid, structure and thermal analysis is shown in Figure 2. Note

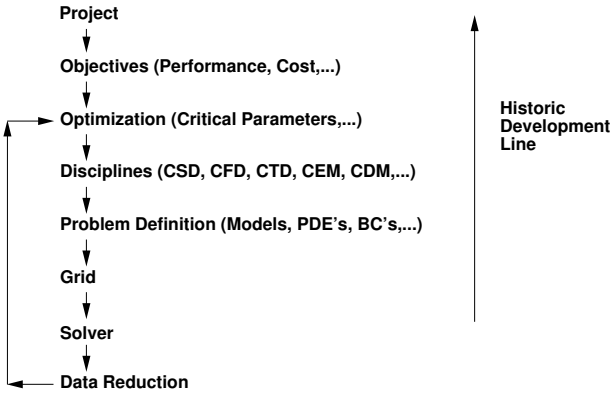


Figure 1: Design and Analysis Process in Engineering

that for each discipline, different levels of physical approximations and realism are possible. The CPU cost, as well as model preparation times, can vary by orders of magnitude as one moves away from the origin. Application areas associated with particular locations in this fluid- structure- thermal- analysis space have been compiled in Figure 3. These only represent the better known application classes of what is a fast-growing range of possibilities.

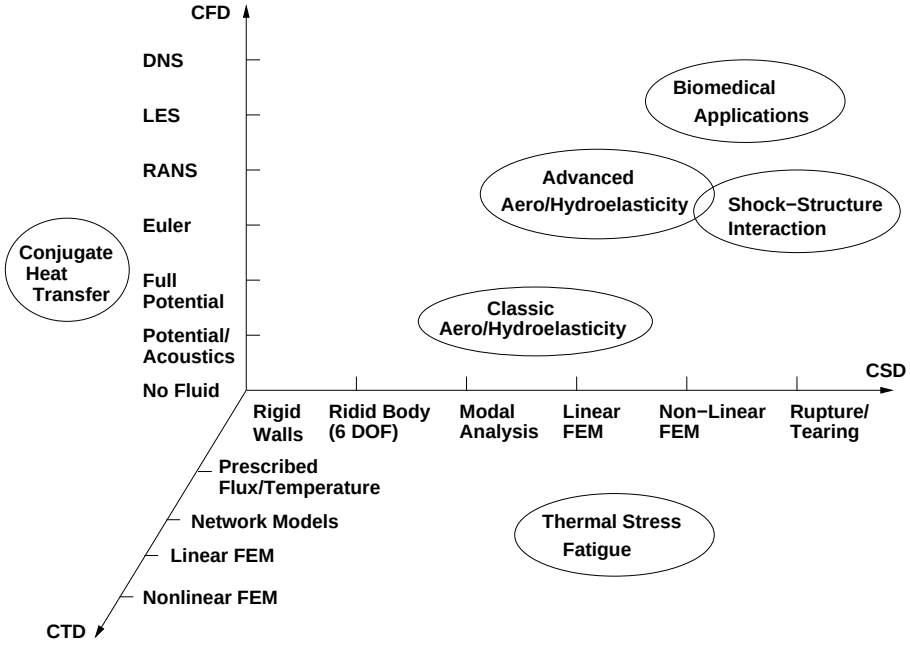


Figure 2: CFD/CSD/CTD Space

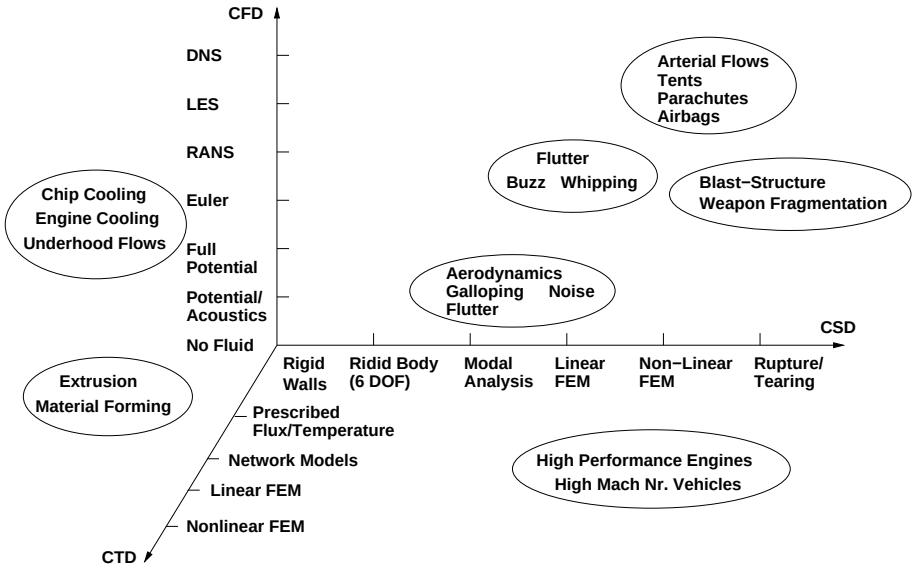


Figure 3: CFD/CSD/CTD Application Areas

More than a decade has passed since the first large-scale fluid-structure interaction calculations using nonlinear structural and fluid models were attempted (17; 6; 19). While many practical problems have been solved successfully (i.e. meaningful insight has been obtained from these coupled simulations), a number of problems already encountered early on persist. The present paper draws attention to some of these. It is the authors' hope that mathematicians and engineers will focus their attention on these so that a firmer theoretical foundation of these methods can be achieved, the robustness of algorithms is improved, and the range of applicability is expanded.

2 Theory

Consider the simplest case of a coupled fluid-structure-thermal problem. Assume that the grids are matching exactly at the interfaces, and that the discretizations are such that the degrees of freedom (positions, velocities, temperatures) are the same as well. Integrating implicitly the complete system in time yields the following matrix system:

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{sf} & \mathbf{K}_{st} \\ \mathbf{K}_{fs} & \mathbf{K}_{ff} & \mathbf{K}_{ft} \\ \mathbf{K}_{ts} & \mathbf{K}_{tf} & \mathbf{K}_{tt} \end{bmatrix} \cdot \begin{pmatrix} \Delta\mathbf{u}_s \\ \Delta\mathbf{u}_f \\ \Delta\mathbf{u}_t \end{pmatrix} = \begin{pmatrix} \mathbf{r}_s \\ \mathbf{r}_f \\ \mathbf{r}_s \end{pmatrix} \quad , \tag{1}$$

where the sub-indices s, f, t stand for structure, fluid and thermal fields, $\mathbf{u}$ are the unknowns, $\mathbf{r}$ the right-hand sides (sum of internal and external forces/ fluxes), the diagonal sub-matrices are the ones usually obtained for each sub-discipline, and the off-diagonal sub-matrices represent the coupling between disciplines.
As long as the grids are not moving and the continuity across fields is guaranteed, a formal analysis similar to that of single-discipline codes can be made. This means that formal orders of convergence for spatial and temporal scales can be obtained.
However, if the fluid mesh moves (ALE frame), or the grids at field interfaces are non-matching, it suddenly becomes much more difficult to obtain formal orders of convergence. Even in the case of just a flowfield with prescribed mesh motion, work by Farhat et al. (12; 13) yields only the following rather pessimistic estimate: Given a p-th order time-accurate scheme on a fixed mesh, and if the discrete geometric conservation law is satisfied to p-th order, the overall scheme on a moving mesh is at least of 1st order. Imagine the average code developer that has spent months developing p-th order accurate schemes, only to find that when the mesh is moved the best that can be hoped for is 'at least 1st order'. Intuitively, this estimate seems overly pessimistic, and it is surprising that we do not even have some simple empirical benchmarks tests to ascertain if this is indeed so.

3 Treatment of Lower Dimensional Abstractions

Lower dimensional abstractions are very common in computational structural dynamics (CSD). Examples include trusses, beams, plates and shells. In many cases the use of these abstractions from 3-D bodies implies a reduction in CPU and memory requirements of several orders of magnitude without omission of relevant physics, and it is only natural that these are widely used. For example, the CSD discretization of the F16 shown in Figure 4 contains all the relevant eigenmode data for an aeroelastic analysis.

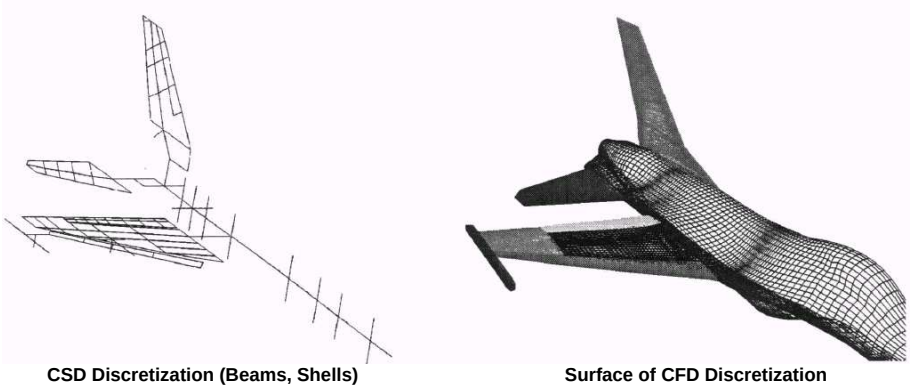


Figure 4: Aeroelastic Simulation of F16: CSD Model and Surface of CFD Mesh

Problems can arise, however, when load and position information has to be transferred to and from these lower dimensional abstractions to the surface of a 3-D computational fluid dynamics (CFD) mesh. Figures 5,6 illustrate possible cases.

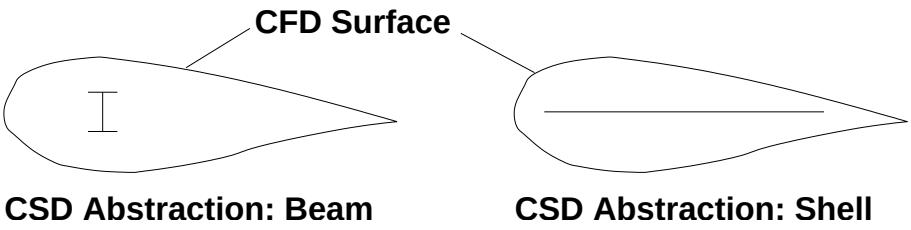


Figure 5: Lower Dimensional Abstractions in CSD

If the beam is associated with a complete airfoil section (a common situation in the aerospace industry) the section is moved/rotated rigidly. Care has to be taken how to define the surface section associated with a beam element (this is particularly the case for curved sections). Moreover, when several beams come together, the situation becomes rather complicated, and one typically ends up with messy, problem specific coding. As far as the authors are aware, thus far a general formulation to link beams to 3-D CFD surfaces has not been found.

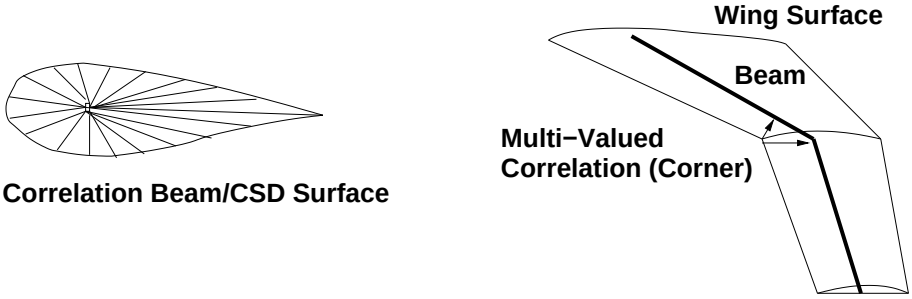


Figure 6: Load and Position Transfer for Beams

If the beams are free standing (e.g. columns), a general (if not elegant) way of handling them is as follows: for

each beam element, construct a set of 'loaded triangles' that reflect the cross-section shape of the beam (circular, square, I-section, L-section, etc.). As the beams can come together at arbitrary angles, these sections are closed. Forces are then obtained for the triangles, and resulting nodal forces and moments can be computed for the beams. Needless to say, this approach only works for embedded grids (1; 20), as there is no guarantee that the triangular elements will not intersect with very small angles or leave small gaps, making body-fitted grids nearly impossible. Alternatively, one could construct a small tetrahedral mesh for each beam and use an immersed of ficticious domain method (2; 28; 3; 4; 14; 22) to obtain the loads.

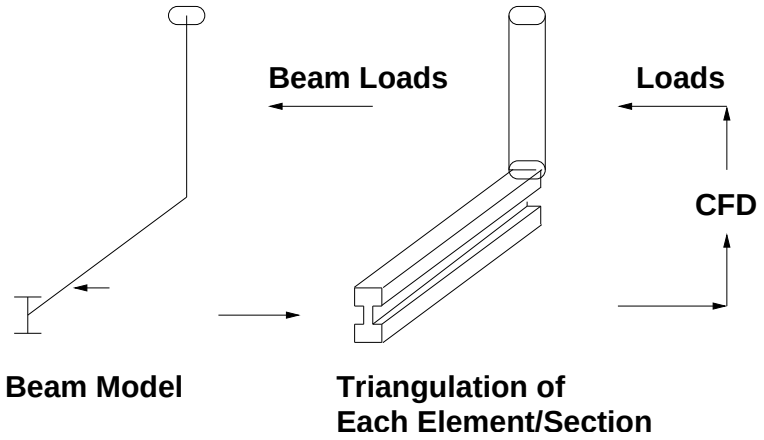


Figure 7: Load Transfer Via Local Triangulation of Beams

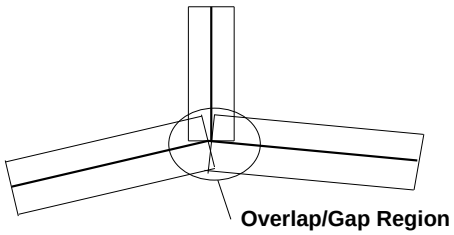
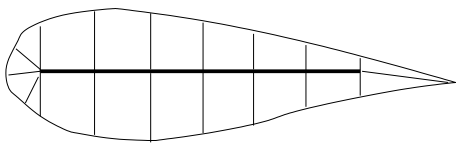


Figure 8: Possible Problems at Joints/Corners

For shells, the situation is usually not as bad as for beams. However, as Figure 9 shows, there may be areas of the CFD surface mesh that are not covered by CSD shells. How to proceed in such cases is again left to a certain degree of trial and error. Usually, all points that may be associated with a shell are moved in accordance with the shell displacements and rotations. However, 'end sections', such as leading and trailing edges, can only be moved rigidly based on the closest shell edge associated with them. As with beams, this does not present problems if the edges are straight (e.g. the leading edge of a wing). However, for corners, a CFD surface mesh point may be associated with several edges, requiring 'problem specific' coding. As with beams, a general way of treating such cases has proven elusive thus far.



CSD Abstraction: Shell

Figure 9: Load and Position Transfer for Shells

4 CFD Specific Problems

The main impediment to routine FSI calculations with large mesh movement and RANS grids has been the handling of moving bodies in the mesh. Two complimentary possibilities that have been widely pursued are the use of body-fitted grids and embedded/immersed techniques. None of these is universally applicable.

The **body fitted** approach is characterized by:

- The need for a fairly clean surface to define the computational domain and/or the wetted surfaces;
- Accurate and straightforward application of boundary conditions;
- Near-optimal grids for Reynolds-averaged Navier-Stokes applications (i.e. highly stretched grids in boundary layers and wakes);
- Moving grids, which implies the use of an arbitrary Lagrangian Eulerian reference frame for the partial differential equations solved;
- The need for sophisticated mesh movement strategies in order to avoid tangling and frequent remeshing;
- The need for a meshing tool as part of the solver in order to remesh automatically regions of deformed/bad elements;
- After remeshing, the solution will have to be interpolated from the old to the new mesh; for compressible flows, this may not present major problems; however, for incompressible flows the interpolated fields will not be divergence free, prompting spurious 'spikes' in the pressure and the need of divergence cleanup tools.

The **embedded/immersed** approach is characterized by:

- The ability to handle 'dirty' surfaces, such as those that may arise due to contact/penetration;
- The possibility of a reduction of local order of PDE approximation (i.e. solution degradation) near embedded/immersed boundaries due to imposition of boundary conditions;
- The difficulty of introducing stretched elements to resolve boundary layers;
- The necessity of adaptivity for most cases, making mpi-based parallelization difficult;
- If the CSD surface moves, points that 'emerge' on the other side of a structure will have to be extrapolated from neighbours; as before, experience indicates that this does not present problems for compressible flows; however, for incompressible flows the extrapolated velocity will not be divergence free, prompting spurious 'spikes' in the pressure and the need of local divergence cleanup tools;
- For problems with moving boundaries the information required to build the proper boundary conditions for elements close to the surface or inside the bodies can take a considerable amount of time; and
- For fluid-structure interaction problems, obtaining the information required to transfer forces back to the structural surface can also be time-consuming.

Most of the elements required to achieve fast and reliable codes for each of these approaches have been worked on during the last two decades. For mesh movement, nonlinear spring (5; 11; 10; 25), Laplacian-based (18) or Elasticity-based (24; 23) smoothing techniques have been proposed. Projective prediction (21) and linelet preconditioning (26) is used extensively to reduce CPU requirements. For local/global remeshing, see (16), (27). In order to relate embedded/immersed CSD data to the CFD mesh, optimal spatial data structures have been devised (20; 21). The treatment of embedded/immersed boundaries to achieve higher order accurary in the CFD solvers has also received considerable attention (2; 28; 3; 4; 20; 14; 22).
Nevertheless, given the high CPU and memory requirements of LES and DNS calculations for realistic Reynolds-numbers ($Re > 10^6$), RANS simulations will remain relevant for at least another two decades. Therefore, the need to be able to cover dirty geometries with RANS-suitable grids remains an active area of research.

5 Achievable Accuracy and Accuracy Requirements

Let us consider, as an example, a typical fluid-structure interaction calculation with nonlinear CFD and CSD effects: blasts on buildings (7; 8).
A simulation of this kind will start with a description of the building that is suitable for a CSD calculation. Here we already encounter a major source of possible errors. Very few buildings are built completely 'CAD to mortar', implying that many material and geometrical parameters need to be estimated or guessed. Moreover, for obvious practical reasons, this already implies a considerable amount of abstraction: columns may become beams, floors and windows are represented by shells, some walls will require reinforced concrete modeling, and machinery may be modeled as solid blocks. Furthermore, the material models will have to be chosen. A fluid dynamicist, used to 'clean' materials like air or C4, can only marvel at the multitude of models and parameters encountered in materials such as concrete, wood, or composites. A typical CSD code suitable for impact will have a dozen models for concrete. Each of these models will have anywhere from 5 to 15 parameters. Which ones to choose for a particular building depends very much on the experience and the familiarity with runs like these of the analyst. In most cases, more than one set of parameters is run so as to gauge their sensitivity. The important point is that after all this abstraction and modeling, the error that can be hoped for will exceed 1%, no matter how accurate the numerics are.

Given that large plastic deformation will occur during the run, it is advisable to define the wetted surfaces (which form the boundaries of the fluid domain) from the CSD model. In this way, the two grids can be 'glued' together. For Euler and LES/DNS simulations, no further modeling is needed. However, for RANS simulations, the structural dynamics expert has to marvel at the multitude of turbulence models available in typical CFD codes. Which one to choose again depends very much on the experience and the familiarity with runs like these of the analyst. If we add to this the variability of high explosive materials, the error that can be hoped for will greatly exceed 1%, no matter how accurate the numerics are.

Once the run starts, as the grids will not be matching exactly, interpolation and projection will introduce further errors. This will again limit the achievable accuracy. Finally, if cracks form or breakup occurs, topology changes and the associated interpolation errors during remeshing or solution extrapolation will limit accuracy.

It is therefore highly likely that such complex coupled engineering problems will never be solved to better than 1% accuracy. This is not a calamity, as many of these calculations nevertheless yield meaningful insight that can be used for decision-making, design and optimization. On the other hand, attempting runs with very high order FEM, FVM of FDM schemes seems contrary to wisdom in this context.

6 Conclusions and Outlook

Fluid-structure interaction techniques have flourished over the last decade. While many commercial packages at present advertise such capabilities, it is fair to say that these options are seldomly used for production runs. The present paper has considered some open problems in what has become a large field of research. In particular:

- The lack of 'optimistic' proofs about the order of temporal convergence of the coupled system if staggered or loose coupling techniques are used;
- The lack of a systematic and unified treatment of information transfer between structural models that are of lower order than the surface of the fluid domain;
- The shortcomings in techniques for the treatment of large body/surface motion within flow solvers; options presently advocated are either embedded/immersed techniques that are poorly suited for RANS applications, or ALE/body fitted techniques that are poorly suited for applications with change of topology and/or crack formation; and
- The achievable accuracy for coupled nonlinear problems, and the implications this has for numerical methods.

It is the authors' hope that mathematicians and engineers will focus their attention on these so that a firmer theoretical foundation of these methods can be achieved, the robustness of algorithms is improved, and the range of applicability is expanded.

7 Acknowledgements

This research was partially supported by DTRA and NRL.

References

[1] M.J. Aftosmis, M.J. Berger and G. Adomavicius - A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries; *AIAA*-00-0808 (2000).

[2] P. Angot, C.-H. Bruneau and P. Fabrie - A Penalization Method to Take Into Account Obstacles in Incompressible Viscous Flows; *Numerische Mathematik* 81, 497-520 (1999).

[3] Baaijens, F.P.T. - A Fictitious Domain/Mortar Element Method for Fluid- Structure Interaction; *Int. J. Num. Meth. Fluids* 35, 734-761 (2001).

[4] E. Balaras - Modeling Complex Boundaries Using an External Force Field on Fixed Cartesian Grids in Large-Eddy Simulations; *Comp. Fluids* 33, 375-404 (2004).

[5] J.T. Batina - Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes; *AIAA J.* 28, 8, 1381-1388 (1990).

[6] J.D. Baum, H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman - A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; *AIAA*-96-0795 (1996).

[7] J.D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone and C. Charman - A Coupled CFD/CSD Methodology for Modeling Weapon Detonation and Fragmentation; *AIAA*-99-0794 (1999).

[8] J.D. Baum, O.A. Soto, M.E. Giltrud, R. Löhner, C. Charman, J. Wolfson, G. Hegemeier and K. Arnett - Modeling of Steel Plate Response to Blast Loading Using a Coupled CFD/CSD Methodology; *Proc. MABS-20 Conf.*, Oslo, Norway, September (2008).

[9] For CFD: Fluent, Star-CD, CFX, Pam-Flow, Cart3d, Arc3d, CFL3D, etc.; for CFD: NASTRAN, ANSYS, ABAQUS, Pam-Solid, Cosmic-NASTRAN, etc.

[10] C. Degand and C. Farhat - A Three-Dimensional Torsional Spring Analogy Method for Unstructured Dynamic Meshes; *Comp. Struct.* ,80, 305-316 (2002).

[11] C. Farhat, C. Degand, B. Koobus and M. Lesoinne; Torsional Springs for Two-Dimensional Dynamic Unstructured Fluid Meshes; *Comp. Meth. App. Mech. Eng.* ,163, 231-245 (1998).

[12] C. Farhat and P. Geuzaine - Design and Analysis of Robust ALE Time-Integrators for the Solution of Unsteady Flow Problems on Moving Grids; *Comp. Meth. Appl. Mech. Eng.* 193, 39-41, 4073-4095 (2004).

[13] P. Geuzaine, C. Grandmont and C. Farhat - Design and Analysis of ALE Schemes With Provable Second-Order Time-Accuracy for Inviscid and Viscous Flow Simulations; *J. Comp. Phys.* 191, 1, 206-227 (2003).

[14] A. Gilmanov and F. Sotiropoulos - A Hybrid Cartesian/Immersed Boundary Method for Simulating Flows with 3-D, Geometrically Complex Moving objects; *J. Comp. Phys.* 207, 457-492 (2005).

[15] M. Lesoinne and Ch. Farhat - Geometric Conservation Laws for Flow Problems With Moving Boundaries and Deformable Meshes, and Their Impact on Aeroelastic Computations; *Comp. Meth. Appl. Mech. Eng.* 134, 71-90 (1996).

[16] R. Löhner - Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing; *Computer Systems in Engineering* 1, 2-4, 257-272 (1990).

[17] R. Löhner, C. Yang, J. Cebral, J.D. Baum, H. Luo, D. Pelessone and C. Charman - Fluid-Structure Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids; *AIAA*-95-2259 [Invited] (1995).

[18] R. Löhner and Chi Yang - Improved ALE Mesh Velocities for Moving Bodies; *Comm. Num. Meth. Eng.* 12, 599-608 (1996).

[19] R. Löhner, C. Yang, J. Cebral, J.D. Baum, H. Luo, D. Pelessone and C. Charman - Fluid-Structure-Thermal Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids; *AIAA*-98-2419 [Invited] (1998).

[20] Löhner, R., J.D. Baum, E. Mestreau, D. Sharov, C. Charman and D. Pelessone - Adaptive Embedded Unstructured Grid Methods; *Int. J. Num. Meth. Eng.* 60, 641-660 (2004).

[21] R. Löhner - *Applied CFD Techniques*; J. Wiley & Sons (2008).

[22] R. Mittal and G. Iaccarino - Immersed Boundary Methods; *Annu. Rev. Fluid Mech.* 37, 239-261 (2005).

[23] E. Nielsen and W. Anderson - Recent Improvements in Aerodynamic Design and Optimization on Unstructured Meshes; *AIAA*-01-0596 (2001).

[24] Sackinger, P., P. Schunk and R. Rao - A Newton-Raphson Pseudo-Solid Domain Mapping Technique for Free and Moving Boundary Problems: A Finite Element Implementation; *J. Comp. Phys.* 125, 83-103 (1996).

[25] K.A. Sorensen, O. Hassan, K. Morgan and N.P. Weatherill - A Multigrid Accelerated Time-Accurate Inviscid Compressible Fluid Flow Solution Algorithm Employing Mesh Movement and Local Remeshing; *Int. J. Num. Meth. in Fluids* 43, 5, 517-536 (2003).

[26] O. Soto, R. Löhner and F. Camelli - A Linelet Preconditioner for Incompressible Flows; *Int. J. Num. Meth. Heat and Fluid Flow* 13, (1), 133-147 (2003).

[27] U. Tremel, K.A. Sorensen, S. Hitzel, H. Rieger, O. Hassan, N.P. Weatherill - Parallel Remeshing of Unstructured Volume Grids for CFD Applications; *Int. J. Num. Meth. Fluids* 53, 8, 1361-1379 (2006).

[28] Turek, S. *Efficient Solvers for Incompressible Flow Problems*; Springer Lecture Notes in Computational Science and Engineering 6, Springer (1999).

Address: Center for CFD, Dept. of Computational and Data Sciences
MS 6A2, George Mason University, Fairfax, VA 22030-4444, USA
rlohner@gmu.edu, www.scs.gmu.edu/∼rlohner
Advanced Concepts Group
SAIC, McLean, VA 22102, USA
joseph.d.baum@saic.com, orlando.a.soto@saic.com

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 189– 196
Herrsching am Ammersee, 29.9.-1.10.2008

ALE and Fluid Structure Interaction. Application to Industrial Problems

E. Longatte , Z. Ozdemir, M. Souli

Multi-physics numerical computation requires a good choice of code coupling schemes when several single-physics codes are involved. Several methods have been used in fluid structure interaction problems involving either partitioned or monolithic procedure. The present work is devoted to simulation of fluid structure interaction for flow-induced vibration problems using a partitioned procedure, which combines the advantages of both weak and strong coupled schemes. Specific attention is paid to energy conservation at the fluid structure interface, especially for moving structure and when strong non linear behaviour may occur in both fluid and structure systems. In the present paper, three coupling algorithms are used and compared on a simple one dimensional problem. Numerical results are discussed in terms of energy conservation properties.

1 Introduction

Study of dynamical structures submitted to cross flows like heat exchanger, tube bundles or panel flutter, requires fluid and structure solvers to be solved at the same time. This can be performed by several ways. A first method consists in solving fluid and structure equations in a single system using monolithic algorithm. This is a strong coupling process ensuring energy conservation of the full-coupled fluid structure system. However this approach is often hard to set up for industrial applications as it requires significant developments in fluid and structure solvers, this difficulty can be overcome by using a partitioned procedure ensuring an external coupling of separated codes. This method is easier to set up and it allows independent model developments in both fluid and structure solvers. The procedure is iterative and each iteration is made of three steps: first computation of fluid forces acting on the structure by solving a CFD problem; second estimation of structure displacement and velocity induced by fluid forces, solving a structure dynamic problem; finally actualization of the fluid domain according to the structure wall motion. This approach has a great flexibility due to its modularity. The partitioned procedure may rely on several kinds of time coupling schemes, explicit or implicit. In Explicit coupling, fluid and structure computations are staggered in time. All explicit coupling algorithms inherently introduce energy because it is impossible to predict correctly the structural displacement inducing correct forces when solving the fluid problem.

The partitioned procedure may rely on several schemes, synchronous or asynchronous explicit algorithms or implicit time coupling schemes. With explicit schemes, fluid and structure computations are staggered in time and, as a result, artificial energy may be produced at the fluid structure interface. In order to reduce interface energy imbalance using explicit methods, explicit asynchronous schemes have been introduced to solve problems in aeroelasticity by Piperno et al. (1995) , Farhat et al. (1995). Implicit staggered method have also been introduced by several authors; Hermann and Steindorf (1999) , Tallec and Mouro (2001) , Mani (2003); to ensure better energy conservation.

This paper is devoted to the presentation of several explicit and implicit code coupling schemes. Their different properties are presented and results are compared to analytical solution for a simple one dimensional problem.

2 Computational process

2.1 Code coupling procedure

The fluid structure code coupling procedure is based on time incremental method and each time step is made of three steps: first computation of fluid forces acting on the structure, second resolution of structure dynamics equation, then fluid mesh update by using a fluid mesh displacement process like an Arbitrary Lagrangian Eulerian (ALE) formulation Hughes et al. (1981) , Souli and Zolesio (2001).

In their previous research work, Farhat and Lesoinne (1997) have shown that the loss in time accuracy and numerical stability of the partitioned procedure can be due to the lack of energy conservation at the fluid structure interface, this energy is numerically, and hence artificially created at the interface by staggering explicit methods. Explicit coupling methods cannot conserve energy at the interface, however, the coupling force passed from the fluid to the structure can be properly adjusted in order to control the imbalance energy and reduce it as much as possible. For this purpose an evaluation of the energy performed at the fluid structure interface using fluid structure coupling schemes is considered below.

On one hand energy variation induced by fluid computation at each time step is expressed by the following equation:

$$\Delta E_f^{n+1} = -F_f^{n+1}(X_m^{n+1} - X_m^n) \tag{1}$$

where F_f^{n+1} designates fluid forces acting on the structure at time t^{n+1}, X_m^n and X_m^{n+1} are fluid domain boundary displacement (or boundary mesh displacement) at times t^n and t^{n+1}.

On the other hand, the structure displacement can be modeled by using a classical structural dynamic equation of the form:

$$M_s A_s + C_s V_s + K_s X_s = F_s \tag{2}$$

where F_s designates forces acting on the structure, depending on the numerical scheme we are using, the force $F_s$can be similar or different from the fluid force F_f in Equation (1), an appropriate choice of the force F_s may reduce energy dissipation at the fluid structure interface. In Equation (2), A_s, V_s and X_s are structure acceleration, velocity and displacement. To illustrate energy variation measured by structure computation, let us take an example where Equation (2) is solved using a Newmark algorithm for time integration, with the Newmark parameters $\alpha = \beta = 1/2$.

$$M_s A_s^{n+1} + C_s V_s^{n+1} + K_s X_s^{n+1} = F_s^{n+1}$$

Terms are estimated at first order as follows:

$$V_s^{n+1} = V_s^n + \frac{\Delta t}{2}(A_s^n + A_s^{n+1}) \tag{3}$$
$$X_s^{n+1} = X_s^n + \frac{\Delta t}{2}(V_s^n + V_s^{n+1})$$

Structure energy is the sum of kinetic energy and potential energy. Hence energy variation provided by structure computation between times $t^n$and t^{n+1} can be written as:

$$\begin{aligned}\Delta E_s^{n+1} = E_s^{n+1} - E_s^n(4) &= \frac{1}{2}\left(V_s^{n+1} + V_s^n\right).M_s\left(V_s^{n+1} - V_s^n\right) + \frac{1}{2}\left(X_s^{n+1} + X_s^n\right).K_s.\left(X_s^{n+1} - X_s^n\right) \\ &= \Delta t.V_s^{n+1/2} M_s A_s^{n+1/2} + \Delta t.X_s^{n+1/2} K_s V_s^{n+1/2} \\ &= \Delta t.V_s^{n+1/2}\left(M_s A_s^{n+1/2} + K_s X_s^{n+1/2}\right) \\ &= \Delta t.V_s^{n+1/2}\left(F_f^{n+1/2} - C_s V_s^{n+1/2}\right)\end{aligned} \tag{4}$$

where $V_s^{n+1/2} = \frac{V_s^{n+1}+V_s^n}{2}$ and $X_s^{n+1/2} = \frac{X_s^{n+1}+X_s^n}{2}$

Finally one gets:

$$\Delta E_s^{n+1} = \left(X_s^{n+1} - X_s^n\right)\frac{F_s^{n+1} + F_s^n}{2} - \Delta t.V^{n+1/2}.C_s.V^{n+1/2} \tag{5}$$

2.2 Interface Conditions

The energy variation induced by the second term of the Equation (5) is due to the structure damping *Cs* and does not account for artificial interface energy. To reduce energy imbalance at the interface, the following relation must be satisfied :

$$\Delta E_s^{n+1} = \left(X_s^{n+1} - X_s^n\right) \frac{F_s^n + F_s^{n+1}}{2} = -F_f^n \left(X_m^{n+1} - X_m^n\right) = \Delta E_f^{n+1} \tag{6}$$

Values of X_s^n and F_f^n are the structural displacements and fluid forces at the fluid structure interface estimated by the structure and fluid solvers. Code coupling schemes are based on this relation and displacement X_m^n and force F_s^n must be built to minimize energy variation. In the following section, three explicit and implicit code coupling schemes are presented.

3 Staggered Solvers

3.1 Explicit Synchronous Algorithm (ESA)

With explicit scheme, structural displacements X_s^{n+1} at time t^{n+1} are induced from fluid force computation F_f^n at time t^{n+1} based on the fluid mesh position X_m at time t^{n+1}. The synchronous scheme gives a prediction of fluid structure interface position at time t^{n+1} by using previous positions at times t^n and t^{n+1}. The following integration scheme is applied:

$$X_m^{n+1} = X_s^n + \alpha_0 \Delta t V_s^n + \alpha_1 \Delta t \left(V_s^n - V_s^{n-1}\right) \tag{7}$$

where, X_s^n designates structure displacement at time t^n, X_m^n is the mesh displacement at time t^n, V^n and $V^{n-1}$are the structure velocity at times t^n, and t^{n-1} andα_0 and $\alpha_1$are specific code coupling scheme coefficients to be chosen.

Fluid forces F_s^{n+1} acting on the structure are calculated and structure displacement X_s^{n+1} is deduced from the mechanical equation, where α_0 and $\alpha_1$are chosen to get high order accuracy in the code coupling scheme. For α_0=1 and α_1=0.5 a second order code coupling scheme in time is obtained.

This algorithm was introduced by Farhat et al. (1995) and Farhat and Lesoinne (1997), where good results have been obtained in aeroelasticity problems like flow past panel flutter. However it does not satisfy the geometric conservation law at the interface (CGL, Thomas and Lombard (1979)).

3.2 Explicit Asynchronous Algorithm (EAA)

With an asynchronous code coupling scheme fluid and structure problems are not solved at the same time step. Fluid computation is expressed at time $t^{n+1/2}$ and structure computation at time t^{n+1} .

The following prediction of first order is used for fluid mesh displacement:

$$X_m^{n+1/2} = X_s^n + \frac{\Delta t}{2} V_s^n \tag{8}$$

Then fluid forces are computed at time t^{n+1}. This procedure ensures geometry mesh conservation and it ensures also displacement and velocity continuity at the fluid structure interface according to the Geometric Conservation Law (CGL, Thomas et Lombard, 1979). For example, by using the previous Newmark structure solver algorithm and the trapezoidal rule,the energy conservation property is deduced from the following relation:

$$V_m^n = \frac{X_m^{n+1/2} - X_m^{n-1/2}}{\Delta t} = \frac{X_s^n - X_s^{n-1}}{\Delta t} + \frac{V_s^n - V_s^{n-1}}{2} = V_s^n \tag{9}$$

whereV_m^n is the mesh velocity at time t^n.

3.3 Implicit Algorithm (IA)

An implicit code coupling scheme is also possible by using an iterative method. This method has been used by several authors for different applications, Hermann and Steindorf (1999), Abouri et al. (2003) and Longatte et al. (2003b). This algorithm uses convergent explicit predictions of the coupled fluid structure system. Subcycling is involved to get convergence for each sub-system and a criteria based on the fluid force or structure velocity is used at each time step to stop the numerical process. For the computation of the fluid and structure variables at step t^{n+1}, the following steps are used :

1. fluid force computation $(F_f^{n+1})^k$
2. prediction of structure displacement $(X_s^{n+1})^k$
3. deformation of the current geometry $(\Omega_f^{n+1})^k$
4. determination of the new geometry $(\Omega_f^{n+1})^{k+1}$ and forces $(F_f^{n+1})^{k+1}$
5. calculation of the error estimator: $E = \frac{\left|\left(F_f^{n+1}\right)^{k+1} - \left(F_f^{n+1}\right)^k\right|}{\left|\left(F_f^{n+1}\right)^0\right|}$

If the error estimator is smaller than a critical value then the next time step is incremented. The process restarts from the initial state t^n and the last velocity estimation$(V_s^{n+1})^k$ is used for the next subcycling of the algorithm. To illustrate the property of the previously mentioned explicit and implicit code coupling schemes (ESA, EAA, IA), a one-dimensional test case is investigated below. This case is described in detail by Longatte et al. (2003a) and Longatte et al. (2009).

4 One dimensional FSI Application

In this part a one-dimensional test case is considered. It involves two structures with nodal mass linked by a spring with a stiffness and no damping, Figure 1.

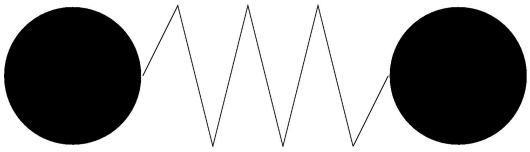


Figure 1: Simple 1D FSI problem

The system satisfies the following mass spring equations :

$$M_1 \frac{d^2 X_s^1}{dt^2} + K_s X_s^1 = K_s X_s^2 \tag{10}$$

$$M_2 \frac{d^2 X_s^2}{dt^2} + K_s X_s^2 = K_s X_s^1 \tag{11}$$

where M_1, M_2, X_1 and X_2 are the structure masses and displacements, respectively.

If the following initial conditions on initial structure displacement and velocities V_1 and V_2 are imposed:

$$X_1(0) = A \tag{12}$$

$$V_2(0) = V_1(0) = 0 \tag{13}$$

and with the additional condition: $M_2 = \frac{M_1}{2}$an analytical solution is given by

$$X_s^1(t) = A\cos(\omega t) \tag{14}$$

$$X_s^2(t) = -2X_s^1(t) \tag{15}$$

where $\omega = \sqrt{3K_s/M_1}$ designates the system circular frequency and *A* the amplitude displacement.

4.1 Interface Conditions

In order to study interface energy variation, fluid and structure energy E_f and E_s can be estimated under the assumption that the first equation of the system describes the structure displacement and the second equation describes the fluid forces at the coupling interface. Thus, fluid forces acting on structure is given by $F_f^n = K_s.X_1^n$. If the explicit synchronous or asynchronous code coupling scheme is chosen, the conditions to minimize numerical energy are deduced from Equation (6).

These extrapolations are not necessary with an implicit code coupling scheme because fluid force is predicted by a series of subcycling in the fluid computation.

4.2 Scheme properties

In what follows, one compares energy conservation properties of ESA, EAA and IA schemes by using one-dimensional test-case. Fluid and structure energy variation are reduced by using scheme ESA or scheme IA. For different time step sizes, energy error is plotted for the three different algorithms in Figures 2 and 3 using $\Delta t = 10^{-3}$ and $\Delta t = 10^{-5}$.

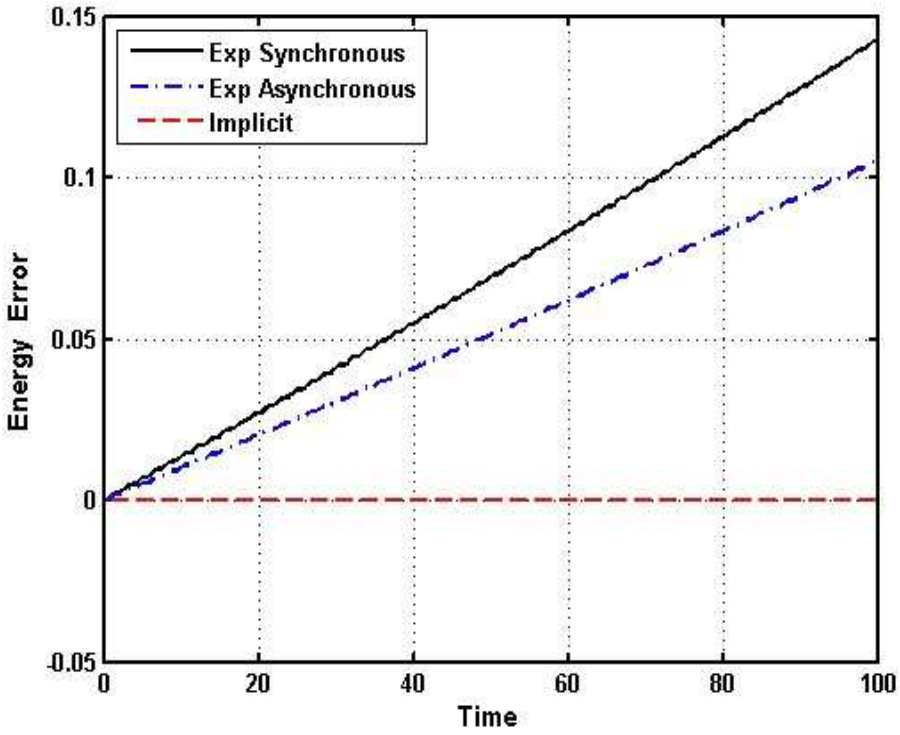


Figure 2: Comparison of the energy variation for 3 numerical schemes (Δt=10^{-3})

For the implicit staggered algorithm, energy conservation is satisfied, no artificial energy has been produced by the algorithm. However for the two explicit methods, artificial energy has been induced in the system. The explicit staggered schemes can not satisfy in the same time velocity and displacement continuity and it leads to energy imbalance. From this simple one dimensional example, it is obvious that the explicit asynchronous algorithm (EAA) gives better energy conservation than the explicit synchronous algorithm (ESA), and numerical error decrease with time step.

One can conclude that numerical damping created by implicit or explicit asynchronous code coupling schemes is lower than damping generated by explicit synchronous scheme. Besides a comparison with a fully implicit monolithic procedure by using a fluid structure finite element code is achieved in Table 4.2. The two masses and the spring are modeled as two discrete elements and a linearelement. As shown for the partitioned implicit code

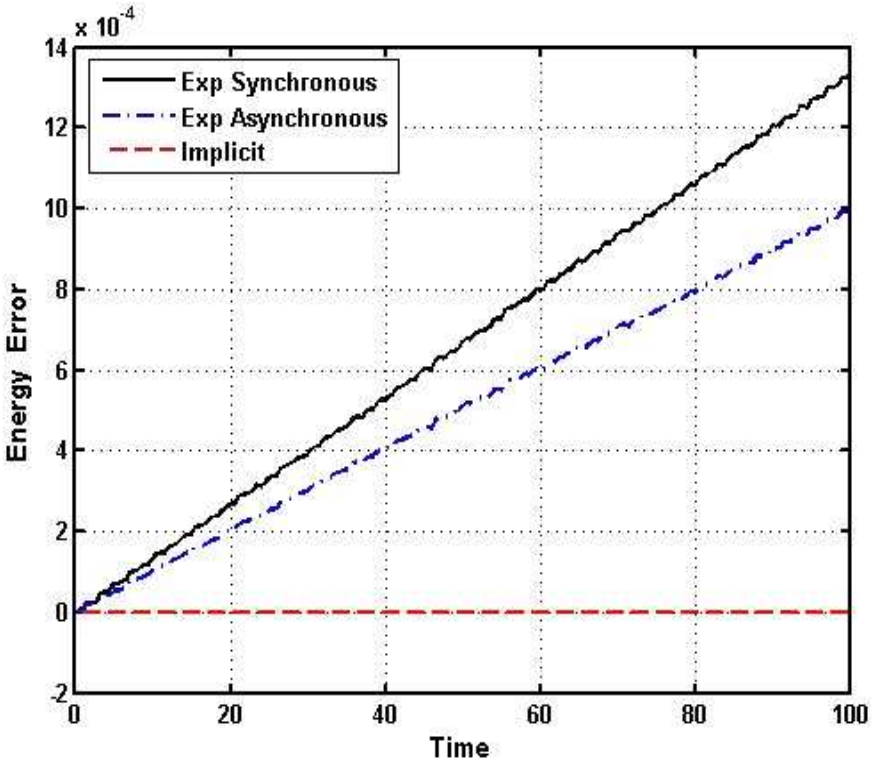


Figure 3: Comparison of the energy variation for 3 numerical schemes (Δt=10^{-5})

Coupling scheme	*Error on displacement*	*Error on energy*
Explicit synchronous	7.08910^{-5}	$8.67\ 10^{-5}$
Explicit asynchronous	5.40410^{-5}	$6.97\ 10^{-6}$
Implicit	$1,85110^{-5}$	9.0210^{-12}
Analytical solution	0	0

Table 1: Comparison of analytical and numerical solutions

coupling scheme, the monolithic procedure provides good results in terms of energy conservation. A fully implicit monolithic process involves a strong coupling solver.

5 Conclusion

Coupling codes method are often the choice for solving complex fluid structure interaction problems. This approach presents a great flexibility and modularity, a validated CFD and structural dynamics codes can be coupled using an efficient coupling interface, this method takes advantage of the parallel processing within each analysis code. Unlike coupled fluid structure algorithms which present a large size of the coupled fluid-structure interaction, the decoupled linear systems solved at each non linear Newton iteration present are reasonably well conditioned.

In this paper, numerical schemes for code coupling and an evaluation method for energy conservation for time integration algorithm in partitioned procedures are presented and discussed in terms of energy conservation properties. This method was validated on a simple mass spring system problem.

References

Abouri, D.; Parry, A.; Hamdouni, A.: Fluid-rigid body dynamic interaction in complex industrial flow. *Chapter Advances in fluid Mechanics: Fluid Structure interaction II, Wit Press*, 0, (2003), 295–305.

Farhat, C.; Lesoinne, M.: Improved staggered algorithms for serial and parallel solution of three- dimensional non-

linear transient aeroelastic problems. *Center for Aerospace Structures - 97-11, University of Colorado, Boulder, Colorado, AIAA Journal*, 0, (1997), –.

Farhat, C.; Lesoinne, M.; Maman, N.: Mixed explicit implicit time integration of coupled aeroelastic probems: three field formulation, geometric conservation and distribution solution. *International Journal for Numerical Methods in Fluids.*, 21, (1995), 807–835.

Hermann, G.; Steindorf, J.: Efficient partitioned procedures for computation of fluid-structure interaction on parallel computers. *Developments in Computational Mechanics with High Performance Computing, B.H.V, Civil-Comp Press, Edinburgh*, 0, (1999), 127–136.

Hughes, T.; Liu, W.; Zimmerman, T.: Lagrangian eulerian finite element formulation for viscous flows. *Comput.Methods Appl.Mech.Engrg*, 29, (1981), 329–349.

Longatte, E.; Bendjeddou, Z.; Souli, M.: Application of arbitrary lagrange euler formulations to flow-induced vibration problems. *Journal of Pressure Vessel Technology*, 125, (2003a), 411–417.

Longatte, E.; Bendjeddou, Z.; Souli, M.: Methods for numerical study of tube bundle vibrations in cross-flows. *Journal of Fluids and Structures*, 18, (2003b), 513–528.

Longatte, E.; Verreman, V.; Souli, M.: Time marching for simulation of fluid-structure interaction problems. *Journal of Fluids and Structures*, 25, (2009), 95–111.

Mani, S.: Truncation error and energy conservation for fluid-structure interactions. *Comput. Methods Appl. Mech. Engrg*, 192, (2003), 4769–4804.

Piperno, S.; Farhat, C.; Larrouturou, B.: Partitioned procedures for the transient solution of coupled aeroelastic problems. part i: Model problem, theory and two-dimensional application. *Comput.Methods Appl.Mech.Engrg*, 124, (1995), 79–112.

Souli, M.; Zolesio, J.: Arbitrary eulerian lagrangian methods in fluid mechanics. *Comput.Methods Appl.Mech.Engrg*, 191, (2001), 451–466.

Tallec, P. L.; Mouro, J.: Fluid structure interaction with large structural displacement. *Comput. Methods Appl. Mech. Engrg*, 190, (2001), 3039–3067.

Thomas, P.; Lombard, C.: Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17, (1979), 1030–1037.

Address: E. Longatte (corresponding author),
Electricit de France- R&D Division
Fluid Mechanics and Heat Transfer Department
email: `elisabeth.longatte@edf.fr`
Z. Ozdemir, M. Souli,
Universit des Sciences et Technologies de Lille I Laboratoire Mecanique de Lille, France email: `zuhal.ozdemir@ed.univ-lille1.fr , mhamed.souli@univ-lille1.fr`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 197– 204
Herrsching am Ammersee, 29.9.-1.10.2008

Transonic Flutter Prediction for a Generic Fighter Configuration

S. Marques, K. Badcock, H. Khodaparast, J. Mottershead

The computation of transonic aeroelastic stability, can require significant computational resources. The approach used in the current paper, is to view the problem of computing flutter onset as a stability problem for a steady state of the coupled fluid-structural system. The stability boundary is determined by solving non-linear eigenvalue problem. An eigenvalue solver based on the Schur Complement Method, is used to compute flutter at transonic conditions of realistic configurations. The development of a generic fighter configuration, based on published data is described. Aerodynamic surface pressures were matched to publicly available data and the structural model was updated according to measured natural frequencies. Results validate the efficiency and robustness of the Schur method against a model wing and a typical generic fighter configuration.

1 Introduction

Advanced fighter aircraft can suffer from aeroelastic instabilities in the transonic regime. Typical aerodynamic analysis tools for flutter calculations rely on panel methods, Yurkovich (2003), coupled with a Computational Structural Dynamics (CSD) model. Such methods are based on linear aerodynamics and are insensitive to the physics that dominate transonic flows. Computational Fluid Dynamics (CFD) can improve on linear methods. However, when computing flows with shock-waves, CFD based flutter analysis requires time-accurate simulations, which are impractical for parametric studies.

The coupled system stability can be studied by analysing the response in time to an initial disturbance, Woodgate et al. (2005). This approach requires unsteady calculations. With today's computational resources, this is still a formidable task and unfeasible when a large parameter space needs to be analysed. To overcome these difficulties, several alternative methods have been investigated such as data driven models, Lucia et al. (2004), or the Harmonic Balance method, Thomas et al. (2006). A particularly attractive technique is to study the stability of the coupled system steady state directly. Assuming a Hopf Bifurcation, stability is lost when the system Jacobian matrix has a pair of eigenvalues which cross the imaginary axis. The idea is to compute when this happens at a computational cost much less than time domain analysis.

The paper presented here, is a result of a continuous effort, Badcock et al. (2004), Badcock et al. (2005), Woodgate and Badcock (2007), in applying eigenvalue stability analysis to flutter problems. The latest development divides the coupled system into a framework consistent with the Schur Complement Method, Badcock and Woodgate (2008) and applies model updating techniques to improve the structural model, Friswell and Mottershead (1995). The formulation is described and applied to mode tracking and flutter analysis of two distinct configurations: a model wing and a realistic generic fighter configuration.

2 Aeroelastic Stability Formulation

2.1 Coupled Formulation

The coupled CFD-CSD semi-discrete system, can be written as:

$$\frac{d\mathbf{w}}{dt} = \mathbf{R}\left(\mathbf{w}, \mu\right) \tag{1}$$

where:

$$\mathbf{w} = [\mathbf{w}_f, \mathbf{w}_s]^T \tag{2}$$

is a vector containing the fluid and structural unknowns, respectively. The residual $\mathbf{R}$:

$$\mathbf{R} = [\mathbf{R}_f, \mathbf{R}_s]^T \tag{3}$$

is a vector containing the fluid residual, $\mathbf{R}_f$, and the structural residual, $\mathbf{R}_s$. The residual also depends on the parameter μ, which is independent of $\mathbf{w}$; μ usually takes the form of dynamic pressure or altitude.

The Euler equations describing the flow field are discretised on curvilinear multi-block body conforming grids, using a cell-centred finite volume method, which converts the partial differential equations into a set of ordinary differential equations. The convective terms are discretised using Osher's upwind method, Osher and Chakravarthy (1983). Monotone Upwind Scheme for Conservation Laws (MUSCL) variable extrapolation (see van Leer (1974)) is used to provide second-order accuracy with the Van Albada's limiter to prevent spurious oscillations around shock waves. Further details on the flow solver are given by Badcock et al. (2000). The structural residual is built up from the modal form of the structural equations, consisting of a linear restoring force and a general force formed by projecting the fluid surface forces onto the mode shapes. The structural model is built out of simplified components such as beams and plates.

2.2 Schur Complement Eigenvalue Solver

As described above, the stability of the equilibria point of equation 1 is determined by eigenvalues of the Jacobian matrix, $\mathbf{A} = \frac{\partial \mathbf{R}}{\partial \mathbf{w}}$. Stability is considered to be lost through a Hopf Bifurcation where a complex conjugate pair of eigenvalues of $\mathbf{A}$ crosses the imaginary axis. The calculation of the Jacobian $\mathbf{A}$ is most conveniently done by partitioning the matrix as:

$$\mathbf{A} = \begin{bmatrix} \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{w}_f} & \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{w}_s} \\ \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{w}_f} & \dfrac{\partial \mathbf{R}_s}{\partial \mathbf{w}_s} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{ff} & \mathbf{A}_{fs} \\ \mathbf{A}_{sf} & \mathbf{A}_{ss} \end{bmatrix} \tag{4}$$

$\mathbf{A}_{ff}$ corresponds to the Jacobian of the fluid system and has been derived analytically. In the current work, and as is conventional in aircraft aeroelasticity, the structure is modelled by a small number of modes, and so the number of fluid unknowns is far higher than the structural unknowns. This means that the Jacobian matrix has a large, but sparse, block $\mathbf{A}_{ff}$ surrounded by thin strips for $\mathbf{A}_{fs}$ and $\mathbf{A}_{sf}$. The term $\mathbf{A}_{fs}$ arises from the dependence of the CFD residual on the mesh motion and speeds, which in turn depend on the structural solution. These terms are currently evaluated using finite differences. The term $\mathbf{A}_{sf}$ is due to the dependence of the generalised forces on the surface pressures. Finally, $\mathbf{A}_{ss}$ is the Jacobian of the structural equations with respect to the structural unknowns. The details of the Jacobian calculation can be found in Badcock et al. (2004).

To compute the onset of instability, the aeroelastic eigenvalue problem needs to be solved. This corresponds to the normal modes under load from the aerodynamic forces. The eigenvalue problem is formulated as:

$$\begin{bmatrix} \mathbf{A}_{ff} & \mathbf{A}_{fs} \\ \mathbf{A}_{sf} & \mathbf{A}_{ss} \end{bmatrix} \mathbf{p} = \lambda \mathbf{p} \tag{5}$$

where $\mathbf{p}$ is the complex eigenvector and λ is the corresponding eigenvalue. The eigenvector is also partitioned according to the fluid and structural components:

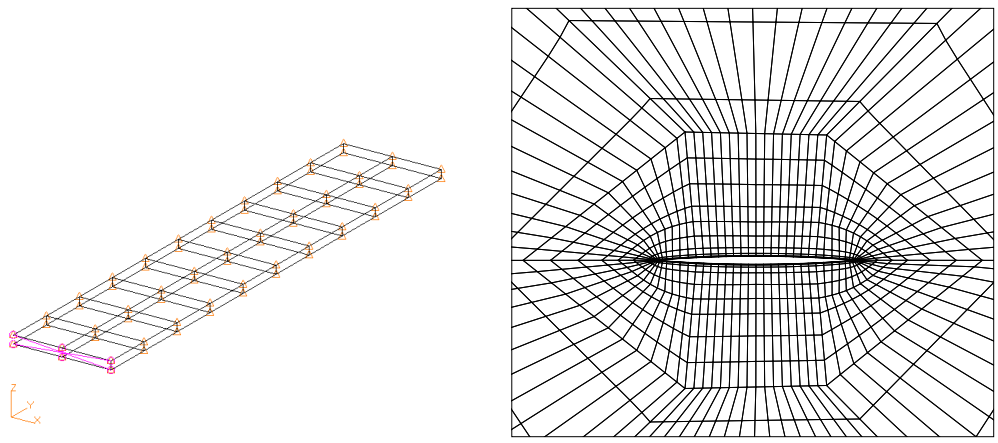
$$\mathbf{p} = [\mathbf{p}_f, \mathbf{p}_s]^T \tag{6}$$

It can be shown, Bekas and Saad (2005), that for a matrix partitioned as $\mathbf{A}$, if λ is an eigenvalue of $\mathbf{A}$ but not an eigenvalue of $\mathbf{A}_{ff}$, then it must also be an eigenvalue of $\mathbf{S}(\lambda)$:

$$\mathbf{S}(\lambda)\, \mathbf{p}_s = \lambda \mathbf{p}_s \tag{7}$$

where $\mathbf{S}(\lambda)$ is given by:

$$\mathbf{S}(\lambda) = \mathbf{A}_{ss} - \mathbf{A}_{sf} \left(\mathbf{A}_{ff} - \lambda \mathbf{I}\right)^{-1} \mathbf{A}_{fs} \tag{8}$$



a-) b-)

Figure 1: Structural and CFD grid

This small non-linear eigenvalue problem can be solved by a Newton-type approach. However, forming $\mathbf{S}$ still requires solving $\left(\mathbf{A}_{ff} - \lambda\mathbf{I}\right)^{-1}\mathbf{A}_{fs}$, which is still too expensive for every combination of λ and μ. To overcome this, the following expansion of $\left(\mathbf{A}_{ff} - \lambda\mathbf{I}\right)^{-1}$ is considered:

$$\left(\mathbf{A}_{ff} - \lambda\mathbf{I}\right)^{-1} \approx \mathbf{A}_{ff}^{-1} + \lambda\mathbf{A}_{ff}^{-1}\mathbf{A}_{ff}^{-1} + \lambda^2\mathbf{A}_{ff}^{-1}\mathbf{A}_{ff}^{-1}\mathbf{A}_{ff}^{-1}\ldots \tag{9}$$

This allows pre-computation of the coefficient matrices in the expansion.

In order to converge, the series expansion requires λ to be small. If λ_0 is the eigenvalue of $\mathbf{A}_{ss}$, then λ is considered a small change from λ_0. This allows reformulation of $\mathbf{S}(\lambda)$, by using λ_0 as a shift to the full eigenvalue problem and therefore replacing $\mathbf{A}_{ff}$ by $\mathbf{A}_{ff} - \lambda_0\mathbf{I}$ and $\mathbf{A}_{ss}$ by $\mathbf{A}_{ss} - \lambda_0\mathbf{I}$. The $\mathbf{S}$ matrix is reformulated as:

$$\mathbf{S}\left(\lambda\right) = \left(\mathbf{A}_{ss} - \lambda_0\mathbf{I}\right) - \mathbf{A}_{sf}\left(\mathbf{A}_{ff} - \lambda_0\mathbf{I} - \lambda\mathbf{I}\right)^{-1}\mathbf{A}_{fs} \tag{10}$$

and eq.9 is now given by:

$$\left(\mathbf{A}_{ff} - \lambda_0\mathbf{I} - \lambda\mathbf{I}\right)^{-1} \approx \left(\mathbf{A}_{ff} - \lambda_0\mathbf{I}\right)^{-1} + \lambda\left(\mathbf{A}_{ff} - \lambda_0\mathbf{I}\right)^{-2} + \lambda^2\left(\mathbf{A}_{ff} - \lambda_0\mathbf{I}\right)^{-3}\ldots \tag{11}$$

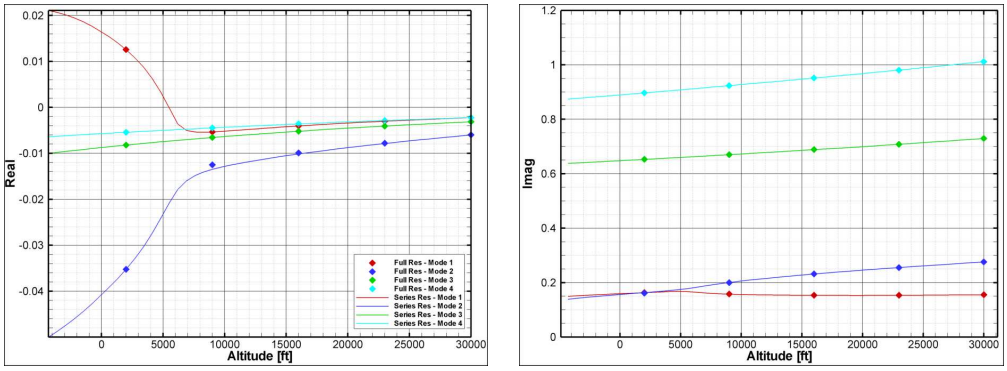
The non-linear problem is solved for λ and the eigenvalue for the original system is given by $\lambda_0 + \lambda$. Further details on the implementation of this method is given by Badcock and Woodgate (2008).

3 Results

3.1 Goland Wing

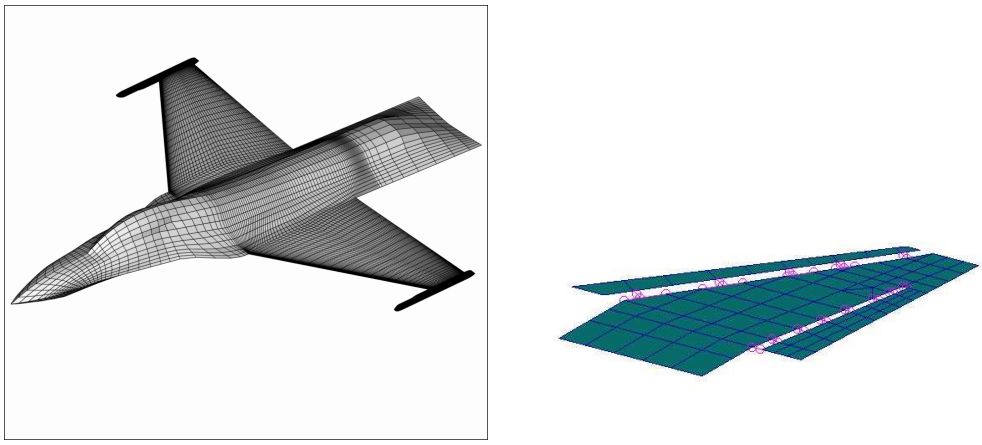
The first test case presented here is the Goland wing. This case has been the subject of several computational studies. The Goland wing uses a 4% thick parabolic aerofoil; it is a rectangular wing with dimensions 20×6ft. A detailed description of the structural model used in this work is given by Beran et al. (2004) and is shown in figure 1-a). The Goland wing is very flexible and flutter can occur even at low Mach numbers.

The multi-block CFD grid for the wing follows a O-O topology and a slice normal to the surface is shown in figure 1-b). A fine grid was generated with about 236 thousand points, from which a coarser level was extracted by removing every other point and resulted in a mesh with 35 thousand points. The first four mode shapes were obtained with MSC Nastran and mapped onto the CFD grid. Details of the mapping process are given by Rampurawala (2006). Initial tests were performed at a freestream Mach number of 0.5 and 0° angle of attack. Altitude was used as a bifurcation parameter. The initial matrices were computed at an altitude of 30000ft and the altitude range went down to 10000ft below sea level. The coarse grid calculations were performed on a single desktop computer. Five full evaluations of $\mathbf{S}(\lambda)$ at equally spaced altitudes were also done, to assess the accuracy of the series expansion. The total runtime using the series expansion was 12 minutes, while the five full evaluations took about 25 minutes.



a-) b-)

Figure 2: Variation of real and imaginary eigenvalues with altitude



a) Surface grid b-) Half-span structural model

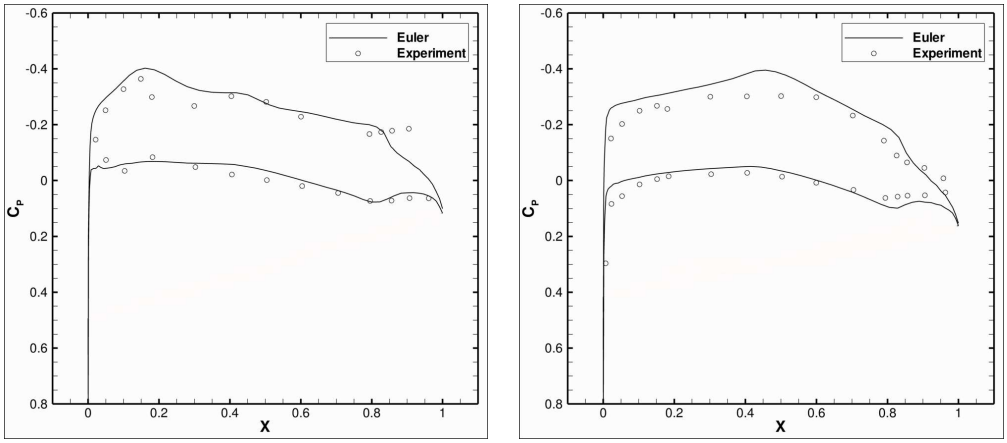
Figure 3: CFD grid and structural model

Figure 2 is obtained by calculating the eigenvalue of the normal mode at the highest altitude. At the highest altitude, the influence of the fluid is small and the eigenvalues depend mostly on the normal modes. As the bifurcation parameter is reduced, the influence of the fluid on the eigenvalues increases and the remainder of the graph is filled. At about 6000ft, the first bending and torsional modes begin to interact leading to flutter. The graphs also show the excellent level of agreement, between the Full Evaluation of eq. 8 or approximating it by using the series expansion defined in eq. 9.

3.2 Generic Fighter Configuration

The generic fighter configuration, shown in figure 3 is based on the F-16 fighter. The objective of this model is to represent a realistic aircraft, both in terms of aerodynamics and structural dynamics. The aeroelastic behaviour of the F-16 has been the focus of several investigations, Denegri (2000), Geuzaine et al. (2003). Furthermore, there is substantial information in the public domain for this aircraft; this allowed the development of a test case model that is representative of a typical fighter configuration.

The overall dimensions of the model have been scaled based on the F-16. A block structured grid was generated which has 890 thousand points and 240 blocks for the full configuration. The full span grid was obtained by reflecting a half span grid. The surface grid is shown in figure 3-a) and shows that the points are concentrated on the wing which contributes most to the aeroelastic response. The wing section consists of a NACA64A204 profile, with a wing root twist of -1° leading to a wing tip twist of -2.4°. The wing twist was evaluated by comparing surface pressures against results given by Denegri and Dubben (2005); figure 4 shows the comparison of the surface pressures at two locations along the span. An illustration of the surface flow field, corresponding to the case quoted above is also given in figure 5. The structural model consists of three main elements, the wing, leading and trailing edge flap, as shown in figure 3-b). The three components are modelled with shell



a-) 59% Span b-) 85% Span

Figure 4: Surface pressure comparison: $M = 0.85$, $\alpha = 2.12°$

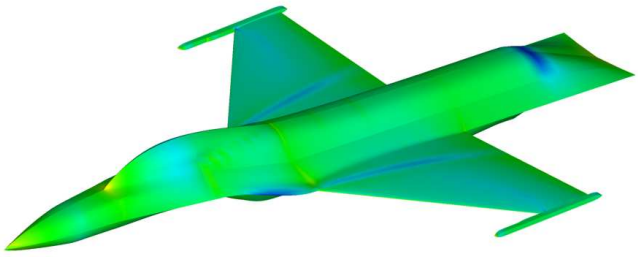
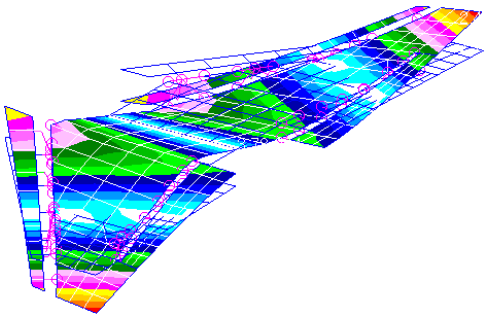


Figure 5: Surface pressure flow field: $M = 0.85$, $\alpha = 2.12°$

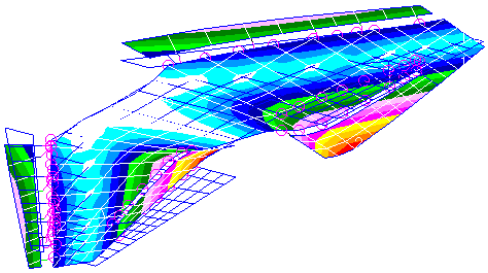
elements. The leading and trailing edge devices are connected to the main wing by spring elements. Material and model properties, such as density, Young's modulus and the several thicknesses are calculated based on model updating against mode frequencies. The following parameters were used: density, Young's modulus, directional spring stiffnesses $(k_x, k_y, k_z, k_\theta, k_\xi, k_\zeta)$, 13 thicknesses along the span. Denegri (2000) identified two modes that contributed to flutter. An optimisation problem was set up, using the two frequencies identified experimentally as an objective function. The structural model was given the 21 parameters and the objective function is given by eq. 12. The optimisation process, selects the parameters that minimise the overall sum described by eq. 12.

$$f = min\left[\sum_{i=1}^{n} W_i \left(\frac{\omega_i^e}{\omega_i^a}\right)^2\right] \tag{12}$$

where W_i represents the model parameters, given in table 1, ω_i^e and ω_i^a represent the experimental and model frequencies respectively for modes 2 and 3. Table 2 shows the initial and final results for the frequencies for the modes identified by Denegri (2000) as contributing to flutter. The structural model mode shapes and the mapping onto the CFD grid for the two relevant modes are shown in figure 6. The flutter calculations for this case were performed at M=0.85 and $0°$ angle of attack. All calculations were performed on 32 processors. The steady state was calculated in about 10min. Precomputing the matrices for the series approximations for 10 modes took about 12h. An altitude sweep was performed between 3000m down to 6000m below sea-level. This set of results is illustrated in figure 7. No flutter is observed for this configuration.

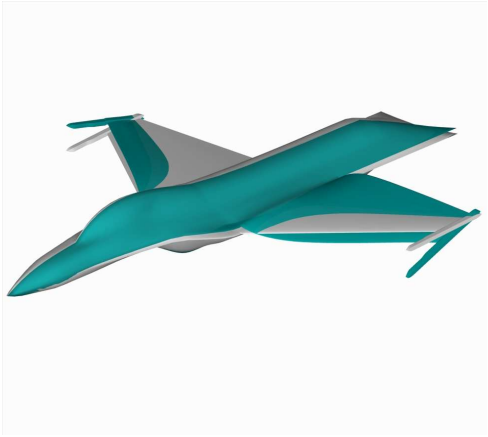


a) Displacement - Structural Model - Mode 2



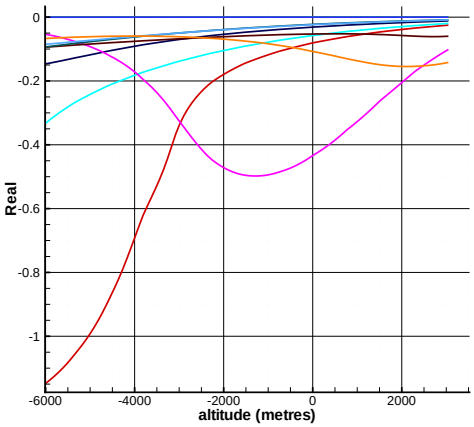
b-) Displacement - Structural Model - Mode 3

a) Displacement - CFD Model - Mode 2

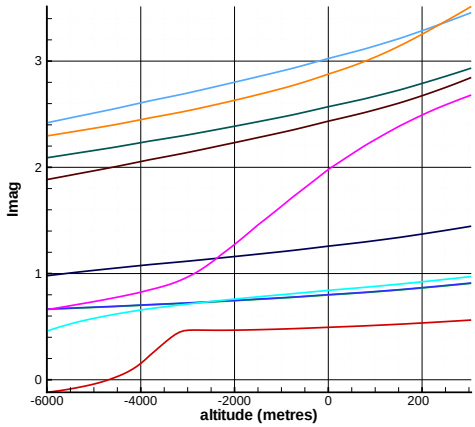


b-) Displacement - CFD Model - Mode 3

Figure 6: Mode Shapes



a)



b-)

Figure 7: Variation of real and imaginary eigenvalue components against altitude

Model Paramenter - W_i	Initial Value	Final Value	**Model Paramenter** - W_i	Initial Value	Final Value
Density	$4000kg/m^3$	$4010.4kg/m^3$	Young's Modulus	$1.0 \times 10^{11}N/m^2$	$6.437 \times 10^{10}N/m^2$
k_x	$1 \times 10^7 N/m$	$1 \times 10^7 N/m$	k_y	$1 \times 10^7 N/m$	$1 \times 10^7 N/m$
k_z	$1 \times 10^7 N/m$	$9.48 \times 10^6 N/m$	k_θ	$1 \times 10^7 N/m$	$9.93 \times 10^6 N/m$
k_ξ	$1 \times 10^7 N/m$	$9.84 \times 10^6 N/m$	k_ζ	$1 \times 10^7 N/m$	$1 \times 10^7 N/m$
t_1	$0.200m$	$0.144m$	t_2	$0.088m$	$0.060m$
t_3	$0.088m$	$0.058m$	t_4	$0.088m$	$0.043m$
t_5	$0.088m$	$0.159m$	t_6	$0.088m$	$0.095m$
t_7	$0.088m$	$0.136m$	t_8	$0.088m$	$0.091m$
t_9	$0.088m$	$0.099m$	t_{10}	$0.088m$	$0.088m$
t_{11}	$0.088m$	$0.076m$	t_{12}	$0.088m$	$0.060m$
t_{13}	$0.088m$	$0.053m$			

Table 1: Model Parameters Definition

Mode	Initial FE Model (Hz)	Denegri data (Hz)	Final FE Model	Mode Shape
1	7.329	–	3.920	symmetric
2	11.983	9.191	9.191	anti-symmetric
3	17.165	9.964	9.964	anti-symmetric
4	21.396	–	22.452	anti-symmetric
5	31.019	–	22.608	symmetric
6	34.380	–	24.020	anti-symmetric
7	41.109	–	26.772	symmetric
8	41.217	–	31.292	anti-symmetric
9	44.905	–	40.040	symmetric
10	45.504	–	41.695	anti-symmetric

Table 2: Structural Model Updating: Frequency matching

4 Conclusion

A methodology to construct and analyse realistic aeroelastic models has been presented. Model updating techniques allow the development of structural models that provide equivalent outputs to real structures. The Schur Complement method permits fast and robust computation of critical eigenvalues and conditions that lead to the onset of flutter. The methodology shown here also provides detailed information about modal interaction that cause instabilities. The computations on two distinct test cases, Goland wing and a generic fighter configuration, validated the methods presented in this paper.

5 Acknowledgements

This work is funded by the European Union for the Marie Curie Excellence Team ECERTA under contract MEXT-CT-2006 042383. The first author would like to thank the financial support provided by Gesellschaft zur Frderung des wissenschaftlichen Fhrungskrftenachwuchses.

References

Badcock, K.; Richards, B.; Woodgate, M.: Elements of Computational Fluid Dynamics on block structured grids using implicit solvers. *Progress in Aerospace Sciences*, 36, (2000), 351–392.

Badcock, K.; Woodgate, M.: Prediction of Bifurcation Onset of Large Order Aeroelastic Models. *49th Structural Dynamics and Materials Conference*, Schaumberg, Illinois, (2008), AIAA paper 2008–1820.

Badcock, K.; Woodgate, M.; Richards, B.: The Application of Sparse Matrix Techniques for the CFD based Aeroelastic Bifurcation Analysis of a Symmetric Aerofoil. *AIAA Journal*, 42(5), (2004), 883–892.

Badcock, K.; Woodgate, M.; Richards, B.: Direct Aeroelastic Bifurcation Analysis of a Symmetric Wing Based on the Euler Equations. *Journal of Aircraft*, 42(3), (2005), 731–737.

Bekas, K.; Saad, Y.: Computation of Smallest Eigenvalues using Spectral Schur Complements. *SIAM Journal of Scientific Computing*, 27(2), (2005), 458–381.

Beran, P.; Knot, N.; Eastep, F.; Synder, R.; Zweber, J.: Numerical Analysis of Store-Induced Limit Cycle Oscillarion. *Journal of Aircraft*, 41(6), (2004), 1315–1326.

Denegri, C.: Limit Cycle Oscillation Flight Test Results of a Fighter with External Stores. *Journal of Aircraft*, 37(5), (2000), 761–769.

Denegri, C.; Dubben, J.: F-16 Limit Cycle Oscillation Analysis Using Transonic Small-disturbance Theory. In: *AIAA-2005-2296 - 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Austin, Texas (2005).

Friswell, M.; Mottershead, J.: *Finite Element Model Updating in Structural Dynamics*. Kluwer Academic Press, Dordrecht (1995).

Geuzaine, P.; Gregory, B.; Harris, C.; Farhat, C.: Aeroelastic Dynamic Analysis of a Full F-16 Configuration for Various Flight Conditions. *AIAA Aircraft*, 41(3), (2003), 363–371.

Lucia, D.; Beran, P.; Silva, W.: Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1-2), (2004), 51–117.

Osher, S.; Chakravarthy, S.: Upwind schemes and boundary conditions with applications to Euler equations in general geometries. *Journal of Computational Physics*, 50, (1983), 447–481.

Rampurawala, A.: *Aeroelastic Analysis of Aircraft with Control Surfaces using CFD*. Ph.D. thesis, The University of Glasgow, Glasgow (March 2006).

Thomas, J.; Dowell, E.; Hall, K.; Denegri, C.: An investigation of the sensitivity of F-16 fighter flutter onset and Limit Cycle Oscillations to Uncertainties. In: *AIAA-2006-1847 - 47th AIAA/ASME/ASCE/AHS/ASC Structures. Structural Dynamics and Materials Conference*, AIAA, Newport, Rhode Island (2006).

van Leer, B.: Towards the ultimate conservative difference scheme II: monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics*, 14(4), (1974), 361–374.

Woodgate, M.; Badcock, K.: On the fast prediction of Transonic Aeroelastic Stability and Limit Cycles. *AIAA Journal*, 45(6), (2007), 1370–1381.

Woodgate, M.; Badcock, K.; Rampurawala, A.; Richards, B.; Nardini, D.; Henshaw, M.: Aeroelastic Calculations for the Hawk Aircraft Using the Euler Equations. *Journal of Aircraft*, 42(4), (2005), 1005–1012.

Yurkovich, R.: Status of Unsteady Aerodynamic Prediction for Flutter of High-Performance Aircraft. *Journal of Aircraft*, 40(5), (2003), 832–842.

Address: S. Marques (corresponding author), CFD Laboratory, Liverpool University.
email: `s.marques@liverpool.ac.uk`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 205– 214
Herrsching am Ammersee, 29.9.-1.10.2008

Solving FSI problems with high resolution and using a component framework in parallel

J. Rang, J. Schöön, H. G. Matthies

In this note fluid-structure interaction problems are treated with a partitioned approach. For the fluid part of the problem the finite volume code OpenFOAM and for the structure part the finite element code ParaFep is used. The communication between the different codes is realised with the help of the component template library (CTL). Therefore the different solvers are transformed into so-called software components which are controlled from a third code. The final non-linear system is solved with an iterative method, i.e. a staggered scheme or a block-Gauss-Seidel method. The implementation is explained and some numerical results show the advantages of the numerical method.

1 Introduction

The interaction of fluid and structure appears in several research areas (Dinkler and Rang, 2008), e.g. the numerical simulation of offshore wind turbines, see (Srisupattarawanit et al., 2004) and (Meyer and Matthies, 2004), or of biomechanical processes.

In the case of fluid-structure interaction problems a fluid and a structure problem are solved which are coupled with the boundary conditions defined on the so-called interface, i.e. a connecting part of the boundary. The computation of the numerical solution needs the simultaneous solution of the strong coupled equations of both problems. Often problems of fluid-structure interaction have a strong dependency between fluid and structure (Morand and Ohayon, 1995) and (Le Tallec and Mouro, 2001).

For building a monolithic solver (Rugonyi and Bathe, 2000), it is difficult to find a free available software system which knows at the same time finite elements, finite volumes and different time discretisation schemes for first and second order systems. This is one reason to use a modular approach and partitioned methods (Rugonyi and Bathe, 2000; Felippa and Park, 1980; Mok and Wall, 2001; Piperno et al., 1995; Matthies and Steindorf, 2002; Matthies et al., 2006; Steindorf, 2002), i.e. the fluid and the structure problem are solved numerically by two different codes. In our approach we use OpenFOAM for solving the fluid and ParaFep (Niekamp, 2001) for solving the structure part. In this note we concentrate on the communication of the different solvers and on the solution strategies of the non-linear system. The communication between the solvers can be realised with the help of the Component Template Library (CTL) (Niekamp, 2005), i.e. the solvers are transformed into software components and are controlled from outside with a central unit. Since OpenFOAM and ParaFep can be executed in parallel it is possible that the FSI simulation runs in parallel, too. Moreover it is in our approach possible that only one of the codes is executed in parallel.

One disadvantage of the partioned approach is the solution of the final non-linear system. Since we are using two different codes a part of this non-linear system is known by OpenFOAM and the other is known by ParaFep. Therefore we are using iterative methods as the staggered scheme or the block-Gauss-Seidel-scheme to solve this non-linear system.

The paper is structured as follows. First we give a short introducting into the two subproblems. In chapter 3 the coupling is discused and the Component Template Library (CTL) is introduced. Finally we present some numerical examples which show that our implementations give good results.

2 The subproblems

The fluid problem is described by the incompressible Navier-Stokes equations and the elastic structure is modelled with equations from elasto-dynamics.

The description of the fluid in domains with moving boundaries can be directly included into the equations, i.e. the Arbitrary-Lagrange-Eulerian (ALE)-Formulation can be chosen (Nath, 1973). The fluid field is described with a local or an Eulerian approach and the motion of the boundary with a Lagrangian approach. The combination of both approaches can be done with a modification of the convective velocities in a local representation. The discretisation in space can be done with finite volumes and in the time with the Crank-Nicolson scheme. A program code which knows both discretisation schemes is OpenFOAM which is an open source code. It is programmed in the programming language C++. The main application is computational fluid dynamics (CFD), i.e. the incompressible Navier-Stokes equations. OpenFOAM can be executed in parallel with the help of MPI and it supports mesh motion and topology changes. Moreover a pre- and postprocessor is included (FoamX and ParaFOAM).

The equations of the structure should be formulated in the case of fluid-structure interaction problems in such a way that large deformations of the structure are possible (Wriggers, 2008). Therefore we use equations of geometric non-linear elasto-dynamics which are represented in a Lagrangian formulation. The equations are discretised in space with finite elements and in time with the Newmark scheme (Matthies et al., 2006). The code ParaFep is used to solve this part of the problem. ParaFep is an in-house code which was developed at the institute of Structural and Numerical Mechanics in Hannover. The main application area are structural mechanics problems, i.e. problems from elasto-dynamics. ParaFep supports adaptivity and can be executed in parallel.

3 The coupling and the Component Template Library

Since we are using two different codes for the fluid and the structure problem it is computational too expensive to solve the final non-linear system in one program. Therefore the partitioned approach is used, i.e. each subproblem is solved by its own code. It is of course clear that either a very small time step size or some iterative method for solving the non-linear system should be used to get a good numerical approximation.

There are several reasons why partioned methods should be used. One advantage of this technique is the re-usability of existing software which can be used without larger changes or improvements. Moreover the best codes for each subproblem can be used and included into the FSI-simulation.

The numerical solution depends strongly on the coupling of both codes (Morand and Ohayon, 1995) and (Le Tallec and Mouro, 2001). One possibility are staggered schemes which are often called loosely coupled. In this approach the non-linear system is only solved one time in each time step. Since the coupling conditions are not satisfied, only relative small time step sizes can be used. Otherwise numerical instabilities will arise (Steindorf, 2002).

The block-Gauss-Seidel method iterates the solution of the non-linear system. This approach is implicit, and therefore the coupling is strong. But the block-Gauss-Seidel method may also fail for some strongly coupled problems (Matthies et al., 2006). For this class of problems it is possible to use a block-Newton method, (Steindorf, 2002) where the final system is solved with Newton's method. But this method needs cross-derivatives which are not explicitely known in a partitioned scheme. A simplification is the inexact block-Newton method (Steindorf, 2002). In this case the cross derivatives are approximated, (Steindorf, 2002). In this note we use the staggered and the block-Gauss-Seidel scheme for solving the non-linear system.

One question is the implementation of the whole concept. The simplest idea is that one of the solvers is the master unit. In this case the coupling algorithm can be implemented in one of the codes but this approach has several disadvantages. Libraries used by the codes may be incompatible and the implementation is difficult since at least one code has to be modified.

Another approach are software components (Szyperski, 1997). A software component is a piece of software which offers (via an interface) a predefined service and which is able to communicate with other components. Clemens Szyperski and David Messerschmitt define several criteria for software components: Software components should be multiple-usable which means that they can be executed in parallel. The implementation of them should be realised in a non-context-specific way, i.e. they are exchangeable. Software components can be composed with

other components and they are encapsulated, i.e. they are non investigable through its interfaces. Moreover software components are units which are independent of deployment and versioning.

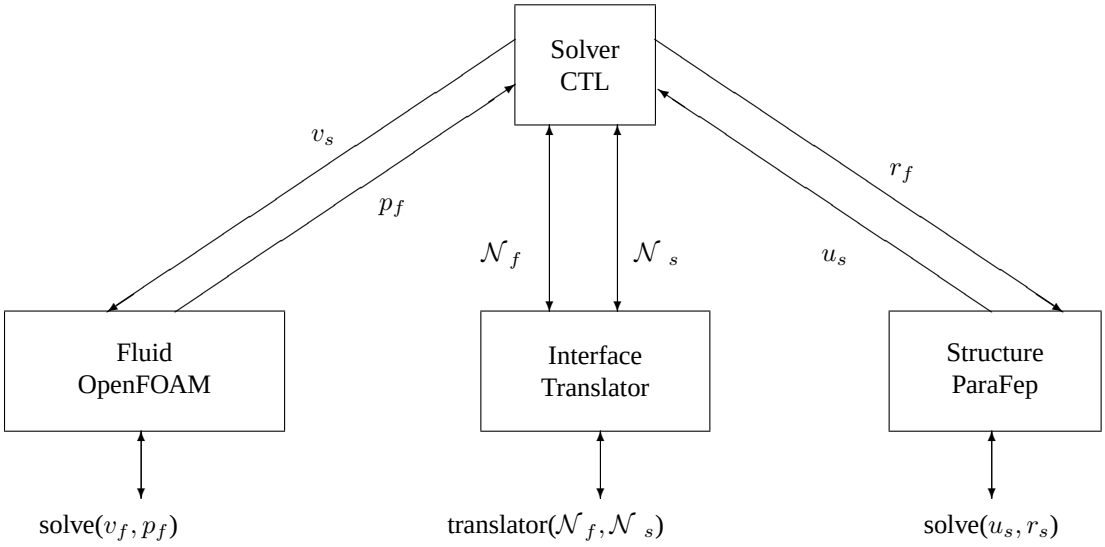


Figure 1: Communication between OpenFOAM, ParaFep, and the CTL

In our implementation the master unit is a third code which controls the fluid and the structure code from outside. Both codes are transformed into software components and the communication between the master unit and these software components is done by using the Component Template Library (CTL) (Niekamp, 2005). The CTL as a generative component framework introduces a high-performance component model and allows the development of high-performance applications with negligible overhead. It furthermore helps to integrate different programming languages and paradigms as well as an extensible amount of communication protocols including MPI. Furthermore CTL components can themselves be intrinsically parallel (i.e. using IMPI on a cluster).

Figure 1 shows a realization for the fluid-structure interaction. A component called *Solver* initialises first all variables and gives the velocity v_f to the fluid solver OpenFOAM. We have build a software component of OpenFOAM such that it can simulate with this information one time step. The numerical results are written into files which can be read by the solver. It is important for the simulation that the pressure p_f can be transformed with the help of the translator to interface nodes of the structure and into forces f_s. These forces are given to the structure solver ParaFep which simulates a time step and which returns displacements u_s.

Figure 2 shows how OpenFOAM (Version 1.2) is called in a coupling process. The C++ frontend `Ofoam` interfaces to OpenFOAM. The interfacing can be done via file editing and system calls. Of course the usage of `Ofoam` requires an OpenFOAM model, called a *case* in the OpenFOAM world. Additionally this model has to fit to the requirements of the used OpenFOAM solver. In the initialisation phase of an `Ofoam` instance the model is geometrically discretised by calling OpenFOAM's mesher `blockMesh`. During the proper FSI run `Ofoam` gets the mesh motion at the FSI boundary as velocities. The velocities are computed by $\frac{du}{dt}$, where du is the displacement change during the time step dt. `Ofoam` writes these velocities to the mesh motion file `motionU` of the model's start-time directory. Subsequently the solver `icoFoamAutoMotionMod` is called via a system call to solve the fluid problem. This solver is only an extension of `icoFoamAutoMotion`. It writes the pressure at the boundary to an additional file named `pBoundary`. This approach is more economic than extracting these pressures from pressure file `p`. `Ofoam` reads these pressures and returns them to the calling instance.

4 Numerical examples

We test our implementation with several test examples. The first one is a flexible beam which is connected with a quadratic rigid-body. At certain fluid velocities vortices break off and the elastic beam begins to move. Further

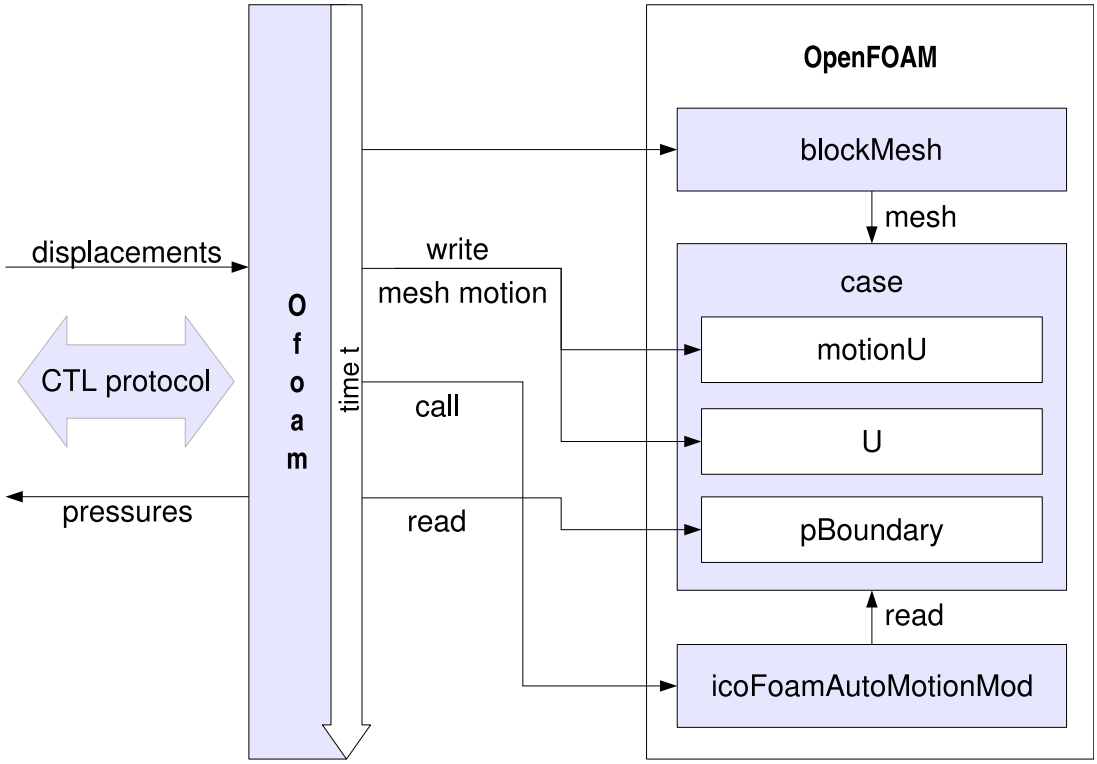


Figure 2: Usage of OpenFOAM (Version 1.2)

examples are an elastic glass fibre in a soap film and an elastic column in a flow.

4.1 Elastic cantilever beam

Our first example is an elastic cantilever beam from (Wall and Ramm, 1998). In this example the fluid induces oscillations of a flexible beam. The elastic structure is connected with a quadratic rigid-body from which at certain fluid velocities and Reynolds numbers vortices break off (see Figure 3). The elastic structure moves due to this vortices.

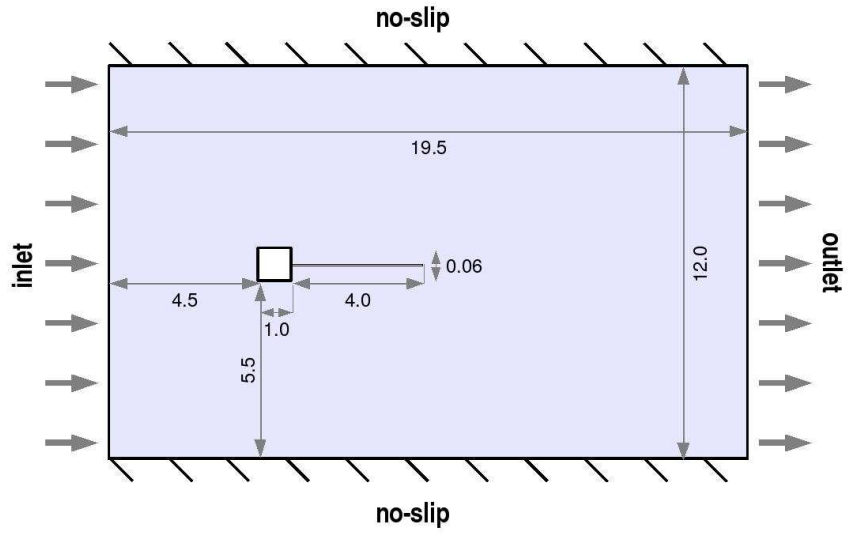


Figure 3: Elastic cantilever beam: Domain

As parameters for the simulation we choose for the structure the elasticity module $E = 2,5 \cdot 10^5$ N/m^2, the

Poisson number $\nu_s = 0.35$ and the density $\rho_s = 100\,\text{kg/m}^3$. For the fluid the density $\rho_f = 1.18\,\text{kg/m}^3$, the

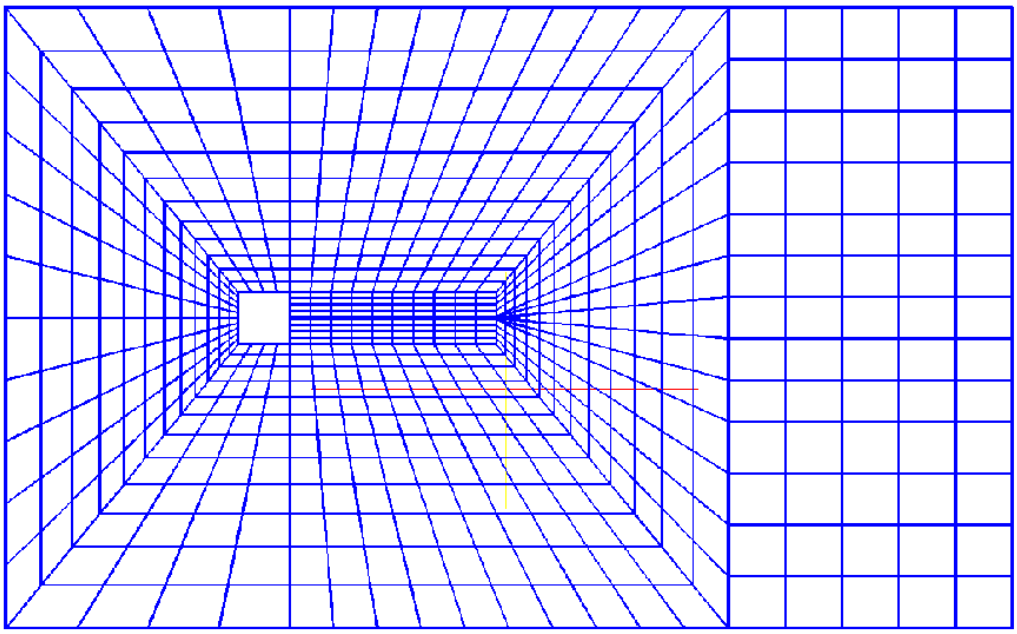


Figure 4: Elastic cantilever beam: Coarse mesh

viscosity $\nu_f = 1.82 \cdot 10^5 \text{Ns/m}^2$ and the inflow velocity $v_x = 0.513$ m/s, $v_y = 0$ m/s are chosen. The width of the elastic structure is $d = 0.6$ mm. All the dimensions and the boundary conditions are presented in Figure 3. A coarse discretisation of the fluid domain can be found in Figure 4.

For our numerical computations we refine the coarse mesh (see Figure 4) three times. We have 48448 degrees of freedom for the velocity and for the pressure. In Figure 5 we present the numerical results after 14.8 and 15 s. It can observed that the elastic beam shows large displacements and that the amplitude of the movement is approximately 0.9 cm. This result fits to other calculations presented in (Steindorf, 2002; Walhorn et al., 2001; Wall, 1999).

4.2 Glass fibre in a soap film

For this problem we have a thin glass fibre (thickness approx. 0.1 mm) which is fixed with a stainless-steel rod in a flowing soap film (1.5 % dawn dish detergent, density $\rho_f = 1000\text{k/m}^3$). The length of the glass fibre varies between 1 and 5 cm, the elasticity module is given by $E = 2 \cdot 10^{-6}\text{Nm}$. The geometry of the problem is shown in Figure 6.

There are many good arguments to consider this problem which is due to (Alben et al., 2002) and (Alben and Shelley, 2004). The geometry is relatively simple and two-dimensional. Moreover the fluid motion is laminar and there exists a unique stationary geometry at the final position of the glass fibre for every combination of fluid velocity and stiffness of the beam.

This test case offers two possibilities to compare the numerical results with the experimental data ((Alben et al., 2002) and (Alben and Shelley, 2004)). The first one is a comparison of the final deviation of the glass fibre which depends on the parameter of the fluid- and the structure problem, i.e. the densities of fluid and structure, the elasticity module, etc. Is is possible to reduce the number of free parameters such that only one remains. But this is only possible if the viscosity, the length and the thickness of the fibre are not changing too much. In this case a non-dimensional parameter η (see (Alben et al., 2002) and (Alben and Shelley, 2004)) can be defined

$$\eta = U\sqrt{\frac{\rho_f f L^3}{2E}},$$

where U is the velocity of the fluid and L is the length of the glass fibre. Since no experimental geometries with the final position of the glass fibre are printed in the references (Alben et al., 2002) and (Alben and Shelley, 2004),

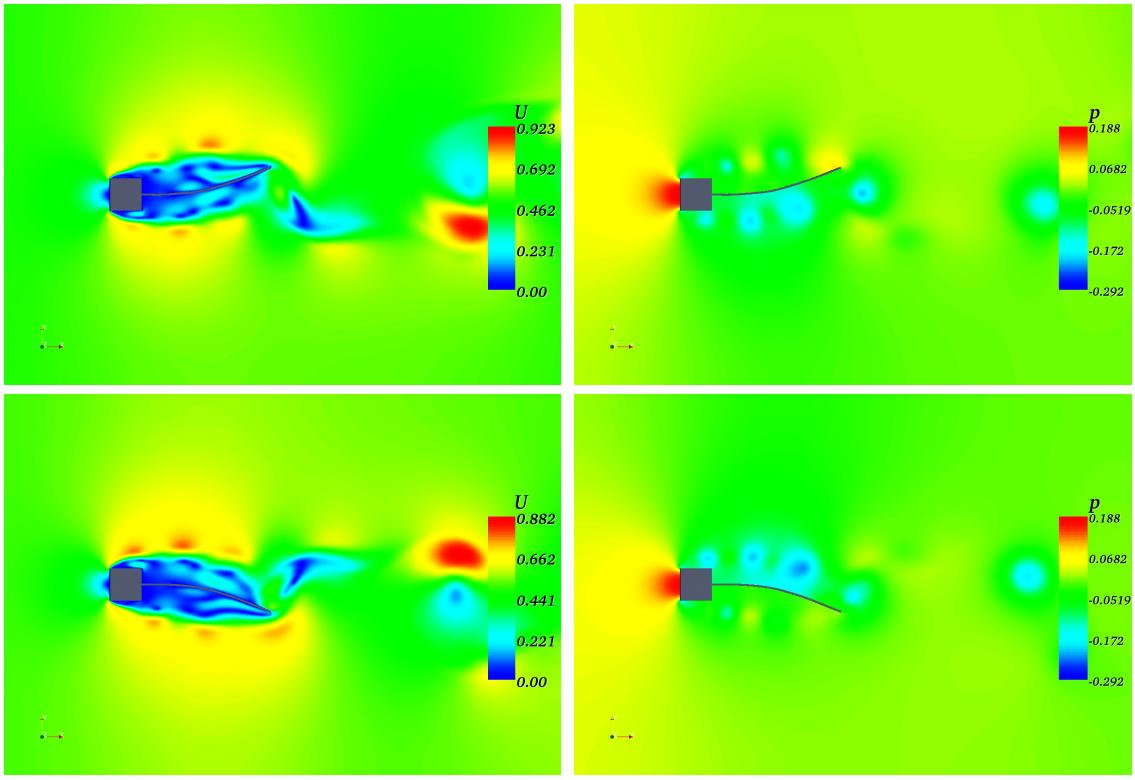


Figure 5: Numerical results for the flexible beam after 14.8 and after 15.0s

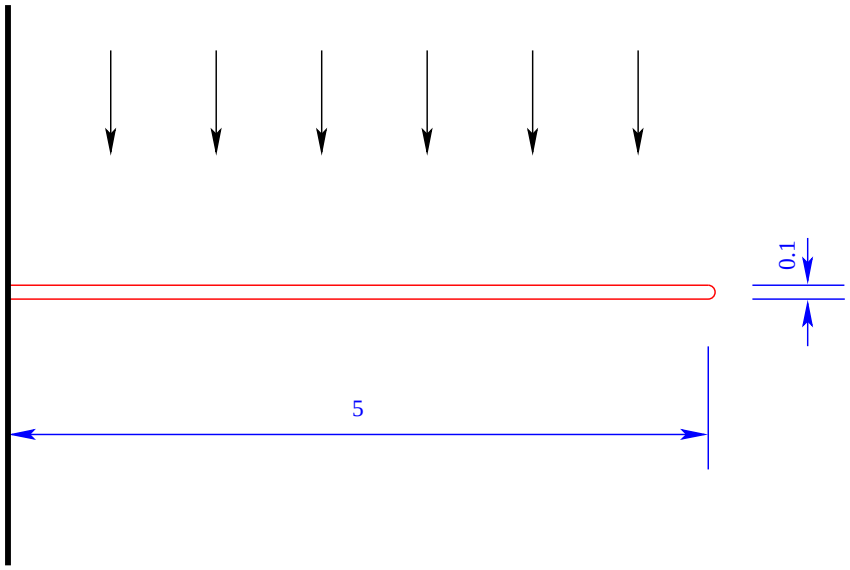


Figure 6: Glass fibre in a soap film (Measurements are in mm)

a direct comparison is not possible. Therefore a comparison with a mathematical model is possible (see Figure. 7). There are quiet large differences between the experimental geometries for $\eta = 16$ and our computation. These differences appear since no re-meshing is done and the deformations of the mesh are quiet large.

The theory and the measurements offer a second possibility for the comparison of experiments and numerical simulations since the ratio of effeminateness of the fibre and the drag can be considered. The drag coefficient

$$C_d := 2 \cdot \text{drag}/(\rho_f f L U^2)$$

is nearly independent from η (for small η), but decreases then with $\eta^{-2/3}$, if the glass fibre is moving under the influence of the fluid (see Figure 8).

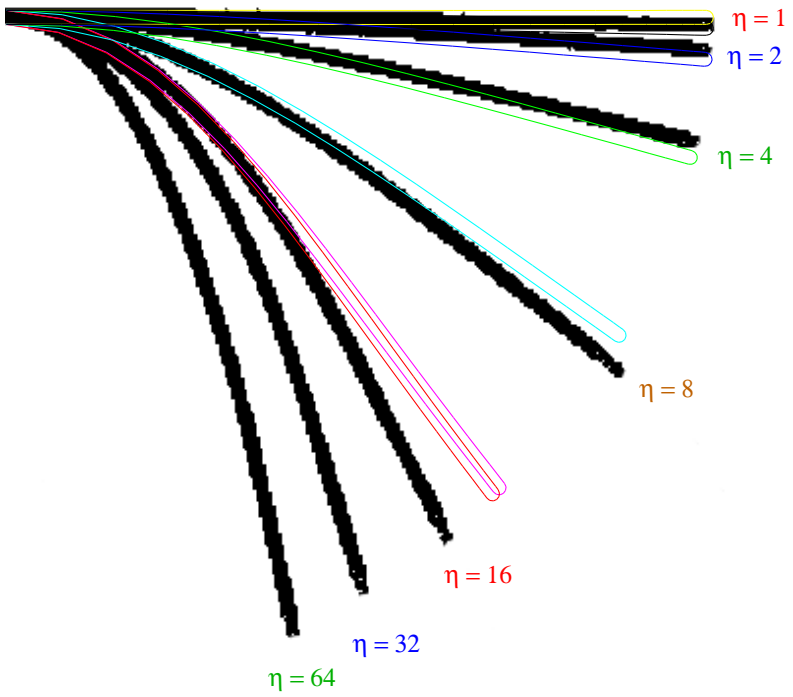


Figure 7: Comparison of displacements of the glass fibre for different η. Black: Experiment from (Alben and Shelley, 2004), Coloured: FSI-Simulation

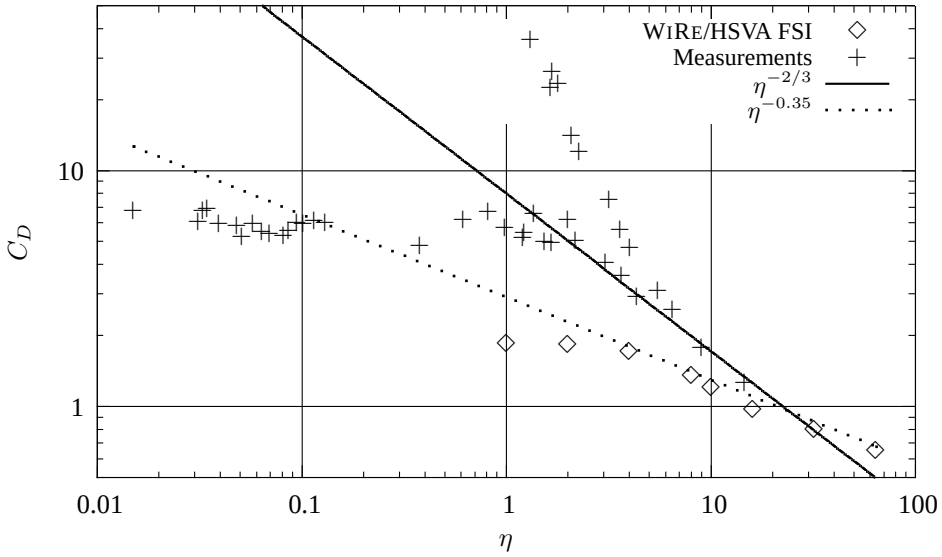


Figure 8: Drag of the glass fibre in the soap film, FSI-results in comparison with measured data (Alben et al., 2002) and (Alben and Shelley, 2004).

The differences between our FSI-simulations and the measure data are due to the fact that a shorter glass fibre is used in our calculation. A longer glass fibre blocks a greater part of the profile of the channel, such that the pressure difference between upper and bottom side of the fibre increases. Since $C_d \approx 2$ our FSI simulation seems to be correct. This is a similar result to the measurements of a plate in a tow-dimensional free flow (see (Hoerner, 1958)).

Column in a flow

A rectangular freestanding column with an inflow of 1 m/s from the left is our first 3D test case. The upper part

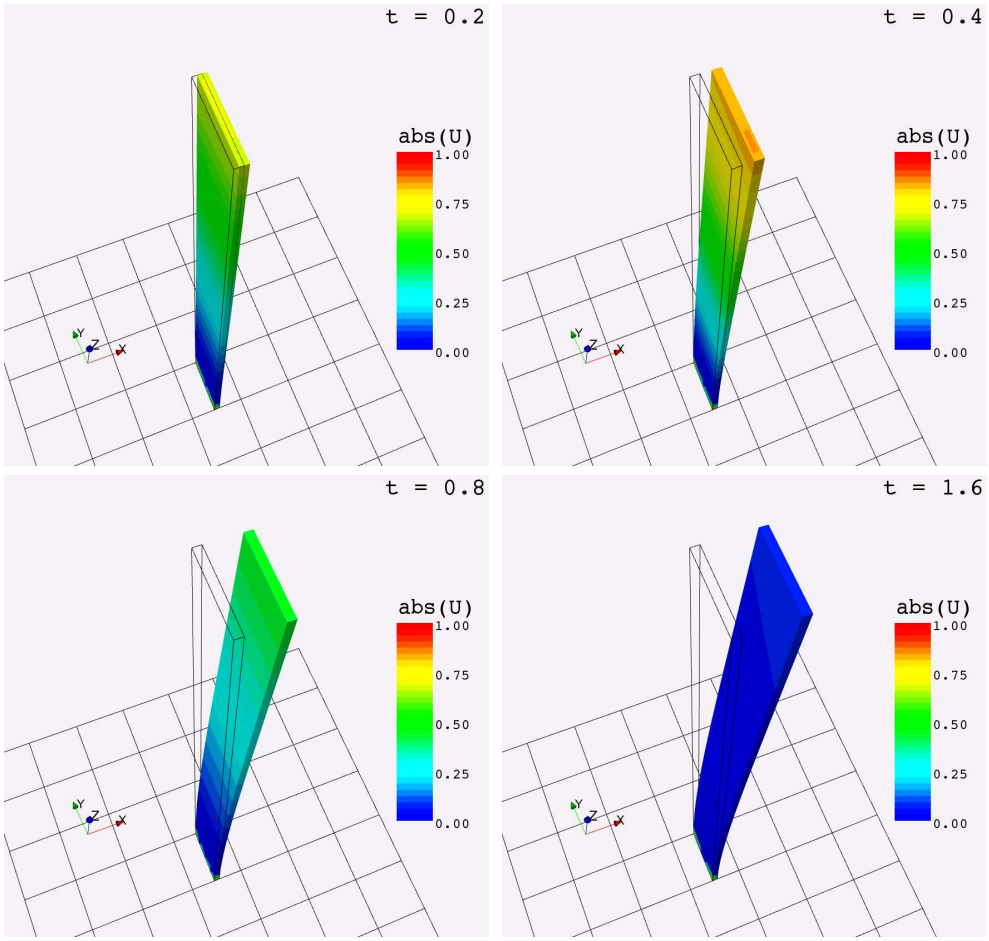


Figure 9: Deviation of the column. "$abs(U)$" is the velocity of the deviation in m/s.

of the column is bent due to the influence of the flow pressure and it reaches the neutral position quickly (see Figure. 9. The final deviation has the correct magnitude for the structure and the fluid which we consider. For this test case no comparable data from experiments or other numerical approximations are known. Nevertheless the numerical results show that our implementation give good results.

5 Summary

In this note we have presented a strategy for solving fluid-structure interaction problems with a partioned method. The fluid and the structure code are realized as so-called software components and the communication between solver and software components is done with the help of the Component Template Library (CTL). Finally we present 2 and 3 dimensional examples which show that our algorithm gives quiet accurate results.

References

Alben, S.; Shelley, M.: How flexibility induces streamlining in a two-dimensional flow. *Physics of Fluids*, 16, 5, (2004), 1694–1713.

Alben, S.; Shelley, M.; Zhang, J.: Drag reduction through self-similar bending of a flexible body. *Nature*, 420, (2002), 479–481.

Dinkler, D.; Rang, J., eds.: *Wechselwirkung von Struktur und Fluid - Abschlussbericht eines Graduiertenkollegs*. Wolfram Schmidt Buchbinderei & Druckerei, Braunschweig (2008).

Felippa, C.; Park, K.: Staggered transient analysis procedures for coupled mechanical systems: Formulation. *Comput. Methods Appl. Mech. Eng.*, 24, (1980), 61–111.

Hoerner, S. F.: *Fluid-Dynamic Drag*. S. F. Hoerner, New Jersey, USA, 2nd edn. (1958).

Le Tallec, P.; Mouro, J.: Fluid-structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Eng.*, 190, 24-25, (2001), 3039–3067.

Matthies, H. G.; Niekamp, R.; Steindorf, J.: Strong coupling methods. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006), 2028–2049.

Matthies, H. G.; Steindorf, J.: Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction. *Comput. Struct.*, 80, (2002), 1991–1999.

Meyer, M.; Matthies, H. G.: State-space representation of instationary two-dimensional airfoil aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics*, 92, 3-4, (2004), 263–274.

Mok, D. P.; Wall, W. A.: Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In: K. S. W. A. Wall, K.-U. Bletzinger, ed., *Trends in Computational Structural Mechanics*, pages 689–698, CIMNE, Barcelona (2001).

Morand, H.; Ohayon, R.: *Fluid-Structure Interaction*. John Wiley & Sohn, Chichester (1995).

Nath, B.: *Dynamics of structure-Fluid systems*, vol. 9 of *Advances in Hydrosciences*. Academic Press, London (1973).

Niekamp, R.: *Verteilte Algorithmen für adaptive Berechnungen in der Mechanik*. Ph.D. thesis, Univerität Hanover, Hanover (2001).

Niekamp, R.: CTL Manual for Linux/Unix. `http://www.wire.tu-bs.de/forschung/projekte/ctl/files/manual.pdf`.

Piperno, S.; Farhat, C.; Larrouturou, B.: Partitioned procedures for the transient solution of coupled aeroelastic problems. I: Model problem, theory and two-dimensional application. *Comput. Methods Appl. Mech. Eng.*, 124, 1-2, (1995), 79–112.

Rugonyi, S.; Bathe, K.-J.: On the analysis of fully-coupled fluid flows with structural interactions — a coupling and condensation procedure. *Int. J. of Comp. Civil and Struct. Eng.*, 1, (2000), 29–41.

Srisupattarawanit, T.; Lehmann, L.; Matthies, .: Simulations of offshore wind turbines with a coupled multi-physics model. In: *Proc. 6th World Congress on Computational Mechanics in conjunction with APCOM'04*, Beijing, China (2004).

Steindorf, J.: *Partitionierte Verfahren für Probleme der Fluid-Struktur Wechselwirkung*. Dissertation, TU Braunschweig, Braunschweig (2002).

Szyperski, C.: *Component Software - Beyond Object-Oriented Programming*. Addison-Wesley (1997).

Walhorn, E.; Hbner, B.; Dinkler, D.: Raum-Zeit Finite-Elemente-Methode zur Analyse von Fluid-Struktur-Wechselwirkungen. *PAMM*, 1, (2001), 81–82.

Wall, W.: *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. Ph.D. thesis, Universität Stuttgart (1999).

Wall, W.; Ramm, E.: Fluid-structure interaction based upon a stabilized (ale) finite element method. Bericht 98/16, SFB 404, Universität Stuttgart (1998).

Wriggers, P.: *Nonlinear finite element methods*. Springer, Berlin (2008).

Address: J. Rang (corresponding author), H. G. Matthies, Institute of Scientific Computing, TU Braunschweig, Hans-Sommer-Str. 65, 38106 Braunschweig
email: `j.rang@tu-bs.de.`
J. Schöön, Hamburgische Schiffbau-Versuchsanstalt GmbH, Bramfelder Strae 164, 22305 Hamburg
email: `schoon@hsva.de.`

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 215– 230
Herrsching am Ammersee, 29.9.-1.10.2008

Numerical simulation of fluid-structure interaction with application to aneurysm hemodynamics

M. Razzaq, S. Turek, J. Hron, J. F. Acker, F. Weichert, M. Wagner, I. Q. Grunwald, C. Roth, and B. F. Romeike

As an example for fluid-structure interaction in biomedical problems, the influence of endovascular stent implantation onto cerebral aneurysm hemodynamics is numerically investigated. The aim is to study the interaction of the elastic walls of the aneurysm with the geometrical shape of the implanted stent structure for prototypical 2D configurations. This study can be seen as a basic step towards the understanding of the resulting complex flow phenomena so that in future aneurysm rupture shall be suppressed by an optimal setting for the implanted stent geometry. From the mathematical side, numerical techniques for solving the problem of fluid-structure interaction with an elastic material in a laminar incompressible viscous flow are described. An Arbitrary Lagrangian-Eulerian (ALE) formulation is employed in a fully coupled monolithic way, considering the problem as one continuum. The mathematical description and the numerical schemes are designed in such a way that more complicated constitutive relations (and more realistic for biomechanics applications) for the fluid as well as the structural part can be easily incorporated. We utilize the well-known Q_2P_1 finite element pair for discretization in space to gain high accuracy and perform as time-stepping the 2nd order Crank-Nicholson, resp., Fractional-Step-θ-scheme for both solid and fluid parts. The resulting nonlinear discretized algebraic system is solved by a Newton method which approximates the Jacobian matrices by the divided differences approach, and the resulting linear systems are solved by iterative solvers, preferably of Krylov-multigrid type. Preliminary results for the stent-assisted occlusion of cerebral aneurysm are presented. Since these results are currently restricted to 2D configurations, the aim is not to predict quantitatively the complex interaction mechanisms between stents and elastic walls of the aneurysm, but to analyse qualitatively the behaviour of the elasticity of the walls vs. the geometrical details of the stent for prototypical flow situations.

1 Introduction

In this contribution, we consider the general problem of viscous flow interacting with an elastic body which is being deformed by the fluid action. Such a problem is of great importance in many real life applications, and typical examples of this type of problem are the areas of biomedical fluids which include the influence of hemodynamic factors in blood vessels, cerebral aneurysm hemodynamics, joint lubrication and deformable cartilage and blood flow interaction with elastic veins (Appanaboyina et al., 2008), (Valencia et al., 2008), (Fernandez et al., 2008), (Tezduyar et al., 2007), (Tezduyar et al., 2008). The theoretical investigation of fluid-structure interaction problems is complicated by the need of a mixed description for both parts: While for the solid part the natural view is the material (Lagrangian) description, for the fluid it is usually the spatial (Eulerian) description. In the case of their combination some kind of mixed description (usually referred to as the Arbitrary Lagrangian-Eulerian description or ALE) has to be used which brings additional nonlinearity into the resulting equations (see (Hron and Turek, 2006b)).

The numerical solution of the resulting equations of the fluid-structure interaction problem poses great challenges since it includes the features of structural mechanics, fluid dynamics and their coupling. The most straightforward solution strategy, mostly used in the available software packages (see for instance (Hron et al., 2002)), is to decouple the problem into the fluid part and solid part, for each of those parts using some well established method of solution; then the interaction process is introduced as external boundary conditions in each of the subproblems. This has the advantage that there are many well tested numerical methods for both separate problems of fluid flow and elastic deformation, while on the other hand the treatment of the interface and the interaction is problematic due to high stiffness and sensitivity. In contrast, the monolithic approach discussed here treats the problem as a

single continuum with the coupling automatically taken care of as internal interface.

Beside a short description of the underlying numerical aspects regarding discretization and solution procedure for this monolithic approach (see (Razzaq et al., 2008), (Hron and Turek, 2006a)), we concentrate on prototypical numerical studies for 2D aneurysm configurations. The corresponding parametrization was based on abstractions of biomedical data (i.e., cutplanes of 3D specimens from New Zealand white rabbits as well as computer tomographic and magnetic resonance imaging data of human neurocrania). In our studies, we allow the walls of the aneurysm to be elastic and hence deforming with the flow field in the vessel. Moreover, we examine several configurations for stent geometries which clearly influence the flow behavior inside of the aneurysm such that a very different elastic displacement of the walls is observed too. We demonstrate that either the elastic modeling of the aneurysm walls as well as the proper description of the geometrical details of the shape of the aneurysm and particularly of the stents is of great importance if the complex interaction between structure and fluid shall be quantitatively analyzed in future, especially in view of more realistic blood flow models and anisotropic constitutive laws of the elastic walls.

2 Fluid-structure interaction problem formulation

The general fluid-structure interaction problem consists of the description of the fluid and solid fields, appropriate interface conditions at the interface and conditions for the remaining boundaries, respectively. In this paper, we consider the flow of an incompressible Newtonian fluid interacting with an elastic solid. We denote the domain occupied by the fluid by Ω_t^b and the solid by Ω_t^s at the time $t \in [0,T]$. Let $\Gamma_t^0 = \bar{\Omega}_t^b \cap \bar{\Omega}_t^s$ be the part of the boundary where the elastic solid interacts with the fluid. In the following, the description for both fields fields and the interface conditions are introduced. Furthermore, discretization aspects and solution procedures are presented in the next section.

2.1 Constitutive relations for the fluid

The fluid is considered to be **Newtonian**, **incompressible** and its state is described by the *velocity* and *pressure* fields $\mathbf{v}^b$, p^b respectively. The constant density of the fluid is ρ^b and the kinematic viscosity is denoted by ν^b. The balance equations are:

$$\rho^b \frac{D\mathbf{v}^b}{Dt} = \operatorname{div} \sigma^b, \quad \operatorname{div} \mathbf{v}^b = 0 \quad \text{in} \quad \Omega_t^b \tag{1}$$

In order to solve the balance equations we need to specify the constitutive relations for the stress tensors. For the fluid we use the incompressible Newtonian relation

$$\sigma^b = -p^b \mathbf{I} + \mu(\nabla \mathbf{v}^b + (\nabla \mathbf{v}^b)^T), \tag{2}$$

where μ represents the dynamic viscosity of the fluid and p^b is the Lagrange multiplier corresponding to the incompressibility constraint in (1). The material time derivative depends on the choice of the reference system. There are basically 3 alternative reference systems: the Eulerian, the Lagrangian, and the Arbitrary Lagrangian-Eulerian formulation. The most commonly used description for the fluid-structure interaction is the ALE description. For the ALE formulation presented in this paper, corresponding discretization techniques are discussed in section 3. Let us remark that also nonnewtonian flow models can be used for modeling blood flow, for instance of Power Law type or even including viscoelastic effects (see (Damanik et al., 2008)) which is planned for future extensions.

2.2 Constitutive relations for the structure

The structure is assumed to be **elastic** and **compressible**. Its configuration is described by the displacement $\mathbf{u^s}$, with velocity field $\mathbf{v^s} = \frac{\partial \mathbf{u}^s}{\partial t}$. The balance equations are:

$$\rho^s \frac{\partial \mathbf{v}^s}{\partial t} + \rho^s (\nabla \mathbf{v}^s)\mathbf{v}^s = \operatorname{div} \sigma^s + \rho^s \mathbf{g}, \quad \text{in} \quad \Omega_t^s. \tag{3}$$

Written in the more common Lagrangian description, i.e. with respect to some fixed reference (initial) state Ω^s, we have

$$\rho^s \frac{\partial^2 \mathbf{u}^s}{\partial t^2} = \operatorname{div}(J\sigma^s \mathbf{F}^{-T}) + \rho^s \mathbf{g}, \quad \text{in} \quad \Omega^s. \tag{4}$$

The constitutive relations for the stress tensors for the compressible structure are presented, however, also incompressible structures can be handled in the same way (see (Hron and Turek, 2006b)). The density of the structure in the undeformed configuration is ρ^s. The material elasticity is characterized by a set of two parameters, the Poisson ratio ν^s and the Young modulus E. Alternatively, the characterization is described by the Lam*é* coefficients λ^s and the shear modulus μ^s. These parameters satisfy the following relations

$$\nu^s = \frac{\lambda^s}{2(\lambda^s+\mu^s)} \qquad E = \frac{\mu^s(3\lambda^s+2\mu^2)}{(\lambda^s+\mu^s)} \tag{5}$$

$$\mu^s = \frac{E}{2(1+\nu^s)} \quad \lambda^s = \frac{\nu^s E}{(1+\nu^s)(1-2\nu^s)}, \tag{6}$$

where $\nu^s = 1/2$ for a incompressible and $\nu^s < 1/2$ for a compressible structure. In the large deformation case it is common to describe the constitutive equation using a stress-strain relation based on the Green Lagrangian strain tensor E and the 2.Piola-Kirchhoff stress tensor $S(E)$ as a function of E. The 2.Piola-Kirchhoff stress can be obtained from the Cauchy stress σ^s as

$$\mathbf{S}^s = J\mathbf{F}^{-1}\sigma^s\mathbf{F}^{-T}, \tag{7}$$

and the Green-Lagrange tensor E as

$$E = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - I). \tag{8}$$

In this paper, the material is specified by giving the Cauchy stress tensor σ^s by the following constitutive law for the St.Venant-Kirchhoff material for simplicity

$$\sigma^s = \frac{1}{J}\mathbf{F}(\lambda^s(trE)I + 2\mu^s E)\mathbf{F}^T \qquad \mathbf{S}^s = \lambda^s(trE)I + 2\mu^s E. \tag{9}$$

J denotes the determinant of the deformation gradient tensor $\mathbf{F}$, defined as $\mathbf{F} = I + \nabla\mathbf{u}^s$. Similar as in the case of more complex blood flow models, also more realistic constitutive relations for the anisotropic behavior of the walls of aneurysms can be included which however is beyond the scope of this contribution.

2.3 Interaction conditions

The boundary conditions on the fluid-solid interface are assumed to be

$$\sigma^b n = \sigma^s n, \quad \mathbf{v}^b = \mathbf{v}^s, \quad \text{on} \quad \Gamma_t^0, \tag{10}$$

where n is a unit normal vector to the interface Γ_t^0. This implies the no-slip condition for the flow and that the forces on the interface are in balance.

3 Discretization and solution techniques

In this study, we restrict at the moment to two dimensions which allows systematic tests of the proposed methods for biomedical applications in a very efficient way such that the qualitatitive behaviour can be carefully analyzed. The corresponding fully implicit, monolithic treatment of the fluid-structure interaction problem suggests that an A-stable second order time stepping scheme and that the same finite elements for both the solid part and the fluid region should be utilized. Moreover, to circumvent the fluid incompressibility constraints, we have to choose a stable finite element pair. For that reason, the conforming biquadratic, discontinuous linear Q_2P_1 pair, see Figure 1 for the location of the degrees of freedom, is chosen which will be explained in the next section.

3.1 Space discretization

Let us define the usual finite dimensional spaces U for displacement, V for velocity, P for pressure approximation as follows

$$\begin{aligned}
U &= \{\mathbf{u} \in L^\infty(I,[W^{1,2}(\Omega)]^2), \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega\},\\
V &= \{\mathbf{v} \in L^2(I,[W^{1,2}(\Omega_t)]^2) \cap L^\infty(I,[L^2(\Omega_t)]^2), \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega\},\\
P &= \{p \in L^2(I,L^2(\Omega))\},
\end{aligned}$$

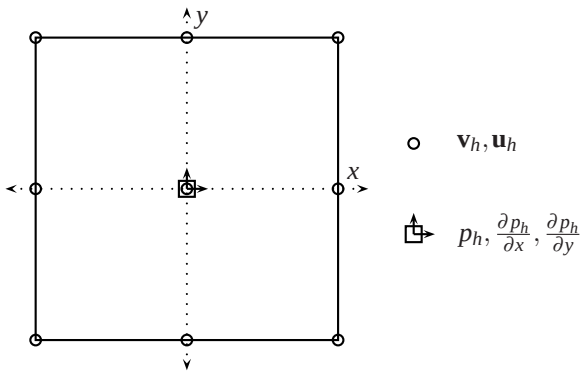


Figure 1: Location of the degrees of freedom for the Q_2P_1 element.

then the variational formulation of the fluid-structure interaction problem is to find $(\mathbf{u},\mathbf{v},p) \in U \times V \times P$ such that the equations are satisfied for all $(\zeta,\xi,\gamma) \in U \times V \times P$ including appropriate initial conditions. The spaces U,V,P on an interval $[t^n,t^{n+1}]$ would be approximated in the case of the Q_2,P_1 pair as

$$\begin{aligned} U_h &= \{\mathbf{u}_h \in [C(\Omega_h)]^2, \mathbf{u}_h|_T \in [Q_2(T)]^2 \quad \forall T \in \mathscr{T}_h, \mathbf{u}_h = \mathbf{0} \text{ on } \partial\Omega\}, \\ V_h &= \{\mathbf{v}_h \in [C(\Omega_h)]^2, \mathbf{v}_h|_T \in [Q_2(T)]^2 \quad \forall T \in \mathscr{T}_h, \mathbf{v}_h = \mathbf{0} \text{ on } \partial\Omega\}, \\ P_h &= \{p_h \in L^2(\Omega_h), p_h|_T \in P_1(T) \quad \forall T \in \mathscr{T}_h\}. \end{aligned}$$

Let us denote by $\mathbf{u}_h^n$ the approximation of $\mathbf{u}(t^n)$, $\mathbf{v}_h^n$ the approximation of $\mathbf{v}(t^n)$ and p_h^n the approximation of $p(t^n)$. Consider for each $T \in T_h$ the bilinear transformation $\psi_T : \hat{T} \to T$ to the unit square T. Then, $Q_2(T)$ is defined as

$$\mathrm{Q}_2(T) = \left\{q \circ \psi_T^{-1} : q \in \text{span} < 1,x,y,xy,x^2,y^2,x^2y,y^2x,x^2y^2 > \right\} \tag{11}$$

with nine local degrees of freedom located at the vertices, midpoints of the edges and in the center of the quadrilateral. The space $P_1(T)$ consists of linear functions defined by

$$\mathrm{P}_1(T) = \left\{q \circ \psi_T^{-1} : q \in \text{span} < 1,x,y > \right\} \tag{12}$$

with the function value and both partial derivatives located in the center of the quadrilateral, as its three local degrees of freedom, which leads to a discontinuous pressure. The inf-sup condition is satisfied (see (Boffi and Gastaldi, 2002)); however, the combination of the bilinear transformation ψ with a linear function on the reference square $P_1(\widehat{T})$ would imply that the basis on the reference square did not contain the full basis. So, the method can at most be first order accurate on general meshes (see (Arnold et al., 2002), (Boffi and Gastaldi, 2002))

$$\|p - p_h\| = O(h). \tag{13}$$

The standard remedy is to consider a local coordinate system (ξ,η) obtained by joining the midpoints of the opposing faces of T (see (Arnold et al., 2002), (Rannacher and Turek, 1992), (Turek, 1999)). Then, we set on each element T

$$P_1(T) := \text{span} < 1,\xi,\eta > . \tag{14}$$

For this case, the inf-sup condition is also satisfied and the second order approximation is recovered for the pressure as well as for the velocity gradient (see (Boffi and Gastaldi, 2002), (Gresho, 1990))

$$\|p - p_h\| = O(h^2) \quad \text{and} \quad \|\nabla(u - u_h)\|_0 = O(h^2). \tag{15}$$

For a smooth solution, the approximation error for the velocity in the L_2-norm is of order $O(h^3)$ which can easily be demonstrated for prescribed polynomials or for smooth data on appropriate domains.

3.2 Time discretization

In view of a more compact presentation, the applied time discretization approach is described only for the fluid part (see (Razzaq, 2009) for more details). In the following, we restrict to the (standard) incompressible Navier-Stokes equations

$$\mathbf{v}_t - \nu\Delta\mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v} + \nabla p = \mathbf{f}, \quad \text{div}\mathbf{v} = 0, \quad \text{in} \quad \Omega \times (0,T], \tag{16}$$

for given force $\mathbf{f}$ and viscosity ν, with prescribed boundary values on the boundary $\partial\Omega$ and an initial condition at $t = 0$. Then, the usual θ-scheme for time discretization reads:

Basic θ-scheme: Given $\mathbf{v}^n$ and $K = t_{n+1} - t_n$, then solve for $\mathbf{v} = \mathbf{v}^{n+1}$ and $p = p^{n+1}$

$$\frac{\mathbf{v} - \mathbf{v}^n}{K} + \theta[-\nu\Delta\mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v}] + \nabla p = \mathbf{g}^{n+1}, \quad \text{div}\,\mathbf{v} = 0, \quad \text{in} \quad \Omega \tag{17}$$

with right hand side $\mathbf{g}^{n+1} := \theta\mathbf{f}^{n+1} + (1-\theta)\mathbf{f}^n - (1-\theta)[-\nu\Delta\mathbf{v}^n + \mathbf{v}^n\cdot\nabla\mathbf{v}^n]$.

The parameter θ has to be chosen depending on the time-stepping scheme, e.g., $\theta = 1$ for the Backward Euler (BE), or $\theta = 1/2$ for the Crank-Nicholson-scheme (CN) which we prefer. The pressure term $\nabla p = \nabla p^{n+1}$ may be replaced by $\theta\nabla p^{n+1} + (1-\theta)\nabla p^n$, but with appropriate postprocessing, both strategies lead to solutions of the same accuracy. In all cases, we end up with the task of solving, at each time step, a nonlinear saddle point problem of given type which has then to be discretized in space as described above.

These two methods, CN and BE, belong to the group of *One-Step-θ-schemes*. The CN scheme can occasionally suffer from numerical instabilities because of its only weak damping property (not strongly A-stable), while the BE-scheme is of first order accuracy only (however: it is a good candidate for steady-state simulations). Another method which has proven to have the potential to excel in this competition is the Fractional-Step-θ-scheme (FS). It uses three different values for θ and for the time step K at each time level. In (Razzaq et al., 2008), (Turek et al., 2006) we additionally described a modified Fractional-Step-θ-scheme which particularly for fluid-structure interaction problems seems to be advantageous. A detailed description will appear in the thesis (Razzaq, 2009).

3.3 Solution algorithms

The system of nonlinear algebraic equations arising from the governing equations described above reads

$$\begin{bmatrix} S_{\mathbf{uu}} & S_{\mathbf{uv}} & 0 \\ S_{\mathbf{vu}} & S_{\mathbf{vv}} & kB \\ c_{\mathbf{u}}B_s^T & c_{\mathbf{v}}B_f^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f_u} \\ \mathbf{f_v} \\ f_p \end{bmatrix} \tag{18}$$

which is a typical saddle point problem, where S describes the diffusive and convective terms from the governing equations. The above system of nonlinear algebraic equations (18) is solved using Newton method as basic iteration. The basic idea of the Newton iteration is to find a root of a function, $\mathbf{R}(\mathbf{X}) = \mathbf{0}$, using the available known function value and its first derivative, where $\mathbf{X} = (\mathbf{u}_h, \mathbf{v}_h, p_h) \in U_h \times V_h \times P_h$. One step of the Newton iteration can be written as

$$\mathbf{X}^{n+1} = \mathbf{X}^n - \left[\frac{\partial\mathbf{R}}{\partial\mathbf{X}}(\mathbf{X}^n)\right]^{-1}\mathbf{R}(\mathbf{X}^n). \tag{19}$$

---

1. Let $\mathbf{X}^n$ be some starting guess.
2. Set the residuum vector $\mathbf{R}^n = \mathbf{R}(\mathbf{X}^n)$ and the tangent matrix $\mathbf{A} = \frac{\partial\mathbf{R}}{\partial\mathbf{X}}(\mathbf{X}^n)$.
3. Solve for the correction $\delta\mathbf{X}$
$$\mathbf{A}\delta\mathbf{X} = \mathbf{R}^n.$$
4. Find optimal step length ω.
5. Update the solution $\mathbf{X}^{n+1} = \mathbf{X}^n - \omega\delta\mathbf{X}$.

Figure 2: One step of the Newton method with line search.

This basic iteration can exhibit quadratic convergence provided that the initial guess is sufficiently close to the solution. To ensure the convergence globally, some improvements of this basic iteration are used. The damped Newton method with line search improves the chance of convergence by adaptively changing the length of the correction vector. The solution update step in the Newton method (19) is replaced by

$$\mathbf{X}^{n+1} = \mathbf{X}^n - \omega\delta\mathbf{X}, \tag{20}$$

where the parameter ω is determined such that a certain error measure decreases (see (Turek, 1999), (Hron and Turek, 2006a) for more details). The Jacobian matrix $\frac{\partial \mathbf{R}(\mathbf{X^n})}{\partial \mathbf{X}}$ can be computed by finite differences from the residual vector $\mathbf{R}(\mathbf{X})$

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]_{ij}(\mathbf{X}^n) \approx \frac{[\mathbf{R}]_i(\mathbf{X}^n + \alpha_j \mathbf{e}_j) - [\mathbf{R}]_i(\mathbf{X}^n - \alpha_j \mathbf{e}_j)}{2\alpha_j}, \tag{21}$$

where $\mathbf{e}_j$ are the unit basis vectors in $\mathbb{R}^n$ and the coefficients α_j are adaptively taken according to the change in the solution in the previous time step. Since we know the sparsity pattern of the Jacobian matrix in advance, which is given by the used finite element method, this computation can be done in an efficient way so that the linear solver remains the dominant part in terms of the CPU time (see (Turek, 1999), (Turek and Schmachtel, 2002) for more details). A good candidate, at least in 2D, seems to be a direct solver for sparse systems like UMFPACK (see (Davis and Duff, 1999)); while this choice provides very robust linear solvers, its memory and CPU time requirements are too high for larger systems (i.e. more than 20.000 unknowns). Large linear problems can be solved by Krylov-space methods (BiCGStab, GMRes, see (Barrett et al., PA 1994)) with suitable preconditioners. One possibility is the ILU preconditioner with special treatment of the saddle point character of our system, where we allow certain fill-in for the zero diagonal blocks, see (Bramley and Wang, 1997).

As an alternative, we also utilize a standard geometric multigrid approach based on a hierarchy of grids obtained by successive regular refinement of a given coarse mesh. The complete multigrid iteration is performed in the standard defect-correction setup with the V or F-type cycle. While a direct sparse solver (Davis and Duff, 1999) is used for the coarse grid solution, on finer levels a fixed number (2 or 4) of iterations by local MPSC schemes (Vanka-like smoother) (Turek, 1999), (Vanka, 1985), (Hron and Turek, 2006a) is performed. Such iterations can be written as

$$\begin{bmatrix} \mathbf{u}^{l+1} \\ \mathbf{v}^{l+1} \\ p^{l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{l} \\ \mathbf{v}^{l} \\ p^{l} \end{bmatrix} - \omega \sum_{\text{element}\Omega_i} \begin{bmatrix} S_{\mathbf{uu}|\Omega_i} & S_{\mathbf{uv}|\Omega_i} & 0 \\ S_{\mathbf{vu}|\Omega_i} & S_{\mathbf{vv}|\Omega_i} & kB_{|\Omega_i} \\ c_{\mathbf{u}} B^T_{s|\Omega_i} & c_{\mathbf{v}} B^T_{f|\Omega_i} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{def}^l_{\mathbf{u}} \\ \mathbf{def}^l_{\mathbf{v}} \\ def^l_p \end{bmatrix}.$$

The inverse of the local systems (39×39) can be done by hardware optimized direct solvers. The full nodal interpolation is used as the prolongation operator $\mathbf{P}$ with its transposed operator used as the restriction $\mathbf{R} = \mathbf{P}^T$ (see (Hron et al., 2002), (Turek, 1999) for more details).

4 Problem description

In the following, we consider the numerical simulation of special problems encountered in the area of cardiovascular hemodynamics, namely flow interaction with thick-walled deformable material, which can become a useful tool for deeper understanding of the onset of diseases of the human circulatory system, as for example blood cell and intimal damages in stenosis, aneurysm rupture, evaluation of the new surgery techniques of heart, arteries and veins (see (Appanaboyina et al., 2008), (Löhner et al., 2008) (Valencia et al., 2008) and therein cited literature). In this contribution, prototypical studies are performed for brain aneurysm. The word 'aneurysm' comes from the latin word *aneurysma* which means dilatation. Aneurysm is a local dilatation in the wall of a blood vessel, usually an artery, due to a defect, disease or injury. Typically, as the aneurysm enlarges, the arterial wall becomes thinner and eventually leaks or ruptures, causing subarachnoid hemorrhage (SAH) (bleeding into brain fluid) or formation of a blood clot within the brain. In the case of a vessel rupture, there is a hemorrhage, and when an artery ruptures, then the hemorrhage is more rapid and more intense. In arteries the wall thickness can be up to 30% of the diameter and its local thickening can lead to the creation of an aneurysm so that the aim of numerical simulations is to relate the aneurysm state (unrupture or rupture) with wall pressure, wall deformation and effective wall stress. Such a relationship would provide information for the diagnosis and treatment of unrupture and rupture of an aneurysm by elucidating the risk of bleeding or rebleeding, respectively.

In order to use the proposed numerical methods for aneurysm hemodynamics, simplified two-dimensional examples, which however include the interaction of the flow with the deformable material, are considered. Flow through a deformable vein with elastic walls of a brain aneurysm is simulated to analyse qualitatively the described methods; here, the flow is driven by prescribing the flow velocity at the inflow part of the boundary while the elastic part of the boundary is either fixed or stress-free. Both ends of the walls are fixed at the inflow and outflow, and the flow is driven by a periodical change of the inflow at the left end.

4.1 Geometry of the problem

For convenience, the geometry of the fluid domain under consideration is currently based on 2D models (see Fig. 3) which allows us to concentrate on the detailed qualitative evaluation of our approach based on the described monolithic ALE formulation. The underlying construction of the (2D) shape of the aneurysm can be explained as follows:

- The bent blood vessel is approximated by quarter circles around the origin.
- The innermost circle has the radius 6*mm*, the next has 8*mm*, and the last one has 8.25*mm*.
- This results in one rigid inner wall and an elastic wall between 8mm and 8.25*mm* of thickness 0.25*mm*.

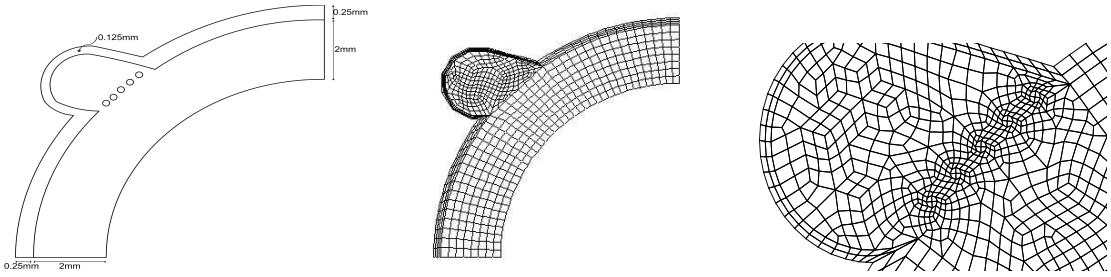


Figure 3: Left: Schematic drawing of the measurement section. Middle: Mesh without stents (776 elements). Right: Mesh with stents (1431 elements) which are part of the simulations.

The aneurysm shape is approximated by two arcs and lines intersecting the arcs tangentially. The midpoints of the arcs are the same (-6.75; 6), they have the radius 1.125*mm* and 1.25*mm*. They are intersected tangentially by lines at angular value 1.3 radians. This results in a wall thickness of 0.125*mm* for the elastic aneurysm walls (see Fig. 3). The examined stents are of circular shape, placed on the neck of the aneurysm, and we use three, resp., five stents (simplified 'circles' in 2D as cutplanes from 3D configurations) of different size and position. The stents also consist of a grid, immersed in the blood flow, which is located at the inlet of the aneurysm so that in future elastic deformations of the stents can be included, too, since in real life, the stent is a medical device which consists of a wire metal wire tube. Stents are typically used to keep arteries open and are located on the vessel wall while this stent is immersed in the blood flow (Fig. 3). The purpose of this device is to reduce the flux into and within the aneurysm in order to occlude it by a clot or rupture. The aneurysm is then intersected with the blood vessel and all missing angular values and intersection points can be determined.

4.2 Boundary and initial conditions

The (steady) velocity profile, to flow from the right to the left part of the channel, is defined as parabolic inflow, namely

$$\mathbf{v}^b(0,y) = \bar{U}(y-6)(y-8). \tag{22}$$

Correspondingly, the pulsatile inflow profile for the nonsteady tests for which peak systole and diastole occur for $\Delta t = 0.25s$ and $\Delta t = 0.75s$ respectively, is prescribed as

$$\mathbf{v}^b(t,0,y) = \mathbf{v}^b(0,y)(1+0.75sin(2\pi t)). \tag{23}$$

The natural outflow condition at the lower left part effectively prescribes some reference value for the pressure variable p, here $p = 0$. While this value could be arbitrarily set in the incompressible case, in the case of a compressible structure this might have influence onto the stress and consequently the deformation of the solid. The *no-slip* condition is prescribed for the fluid on the other boundary parts, i.e. top and bottom wall, stents and fluid-structure interface.

5 Numerical results

The newtonian fluid used in the tests has a density $\rho^b = 1.035 \times 10^{-6} kg/mm^3$ and a kinematic viscosity $\nu^b = 3.38 mm^2/s$ which is similar to the properties of blood. If we prescribe the inflow speed $\bar{U} = -50 mm/s$, this results in a Reynolds number $Re \approx 120$ based on the prescribed peak systole inflow velocity and the width of the veins which is $2mm$ such that the resulting flow is within the laminar region. Parameter values for the elastic vein in the described model are as follows: The density of the upper elastic wall is $\rho^s = 1.12 \times 10^{-6} kg/mm^3$, solid shear modulus is $\mu^s = 42.85 kg/mms^2$, Poisson ratio is $\nu^p = 0.4$, Young modulus is $E = 120 kN/mm^2$. As described before, the constitutive relations used for the materials are the incompressible Newtonian model (2) for the fluid and a hyperelastic neo-Hookean material for the solid. This choice includes most of the typical difficulties the numerical method has to deal with, namely the incompressibility and significant deformations.

From a medical point of view, the use of stents provides an efficient treatment for managing the difficult entity of intracranial aneurysms. Here, the thickness of the aneurysm wall is attenuated and the aneurysm hemodynamics changes significantly. Since the purpose of this device is to control the flux within the aneurysm in order to occlude it by a clot or rupture, the resulting flow behavior into and within the aneurysm is the main objective, particularly in view of the different stent geometries. Therefore, we decided for the 2D studies to locate the (2D parts of the) stents only in direct connection to the aneurysm.

Comparing our studies with the CFD literature (see (Fernandez et al., 2008), (Appanaboyina et al., 2008), (Valencia et al., 2008), (Torri et al., 2007a), (Torri et al., 2007b)), several research groups focus on CFD simulations with realistic 3D geometries, but typically assuming rigid walls. In contrast, we concentrate on the complex interaction between elastic deformations and flow perturbations induced by the stents. At the moment, we are only able to perform these simulations in 2D, however, with these studies we should be able to analyse qualitatively the influence of geometrical details onto the elastic material behavior, particularly in view of more complex blood models and constitutive equations for the structure. Therefore, the aims of our studies can be described as follows:

1. What is the influence of the elasticity of the walls onto the flow behavior inside of the aneurysm, particularly w.r.t. the resulting shape of the aneurysm?
2. What is the influence of the geometrical details of the (2D) stents, that means shape, size, position, onto the flow behavior into and inside of the aneurysm?
3. Do both aspects, small-scale geometrical details as well as elastic fluid-structure interaction, have to be considered simultaneously or is one of them negligible in first order approximation?
4. Are modern numerical methods and corresponding CFD simulations tools able to simulate qualitatively the multiphysics behavior of such biomedical configurations?

In the following, we show some corresponding results for the described prototypical aneurysm geometry, first for the steady state inflow profile, followed by nonsteady tests for the pulsatile inflow, both with rigid and elastic walls, respectively.

5.1 Steady configurations

Due to the given inflow profile, which is not time-dependent, and due to the low Re numbers, the flow behaviour leads to a steady state which only depends on the elasticity and the shape of the stents. Moreover, for the following simulations, we only treat the aneurysm wall as elastic structure. Then, the aneurysm undergoes some slight deformations which can hardly be seen in the following figures. However they result in a different volume of the flow domain (see Fig. 6) and lead to a significantly different local flow behaviour since the spacing between stents and elastic walls may change (see the subsequent color pictures).

Figure 4: Deformed mesh for steady configuration without stents, with elastic wall (left). Mesh for rigid wall (right).

Figure 5: Deformed mesh for steady configuration with stents: 3 stents (left) and 5 stents (right).

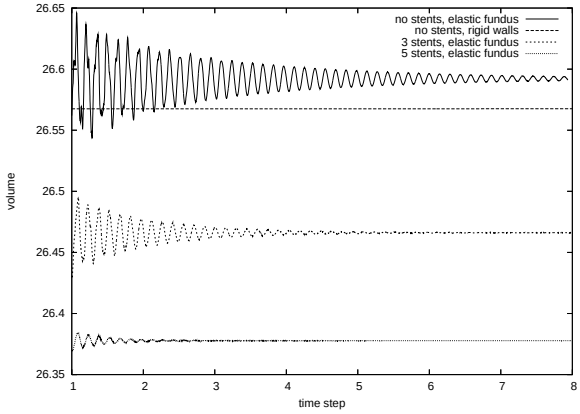


Figure 6: Resulting volume of the fluid domain for different configurations.

In the following pictures, we visualize the different flow behaviour by coloring due the velocity magnitude and by showing corresponding vector plots inside of the aneurysm. Particularly the influence of the number of stents onto the complete fluid flow through the channel including the aneurysm can be clearly seen.

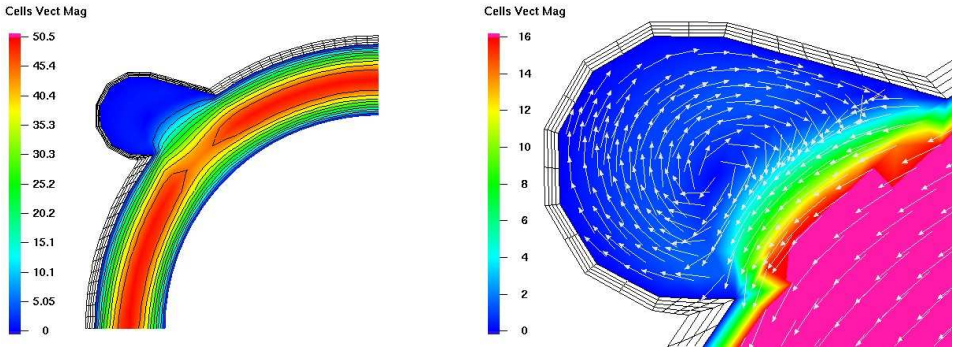


Figure 7: Rigid wall without stents.

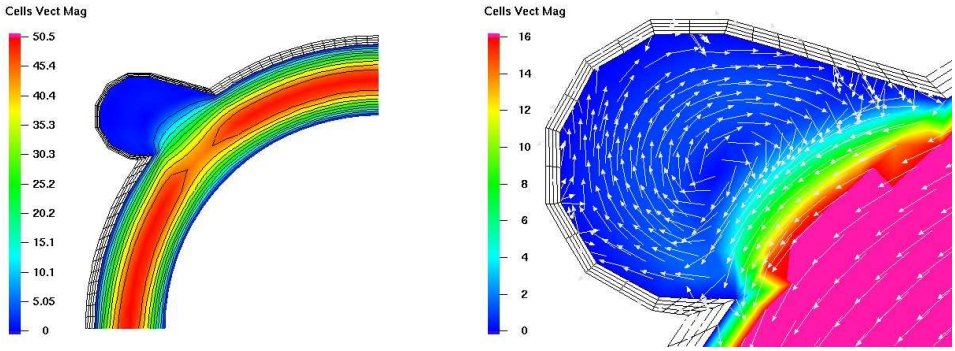


Figure 8: Elastic aneurysm wall without stents.

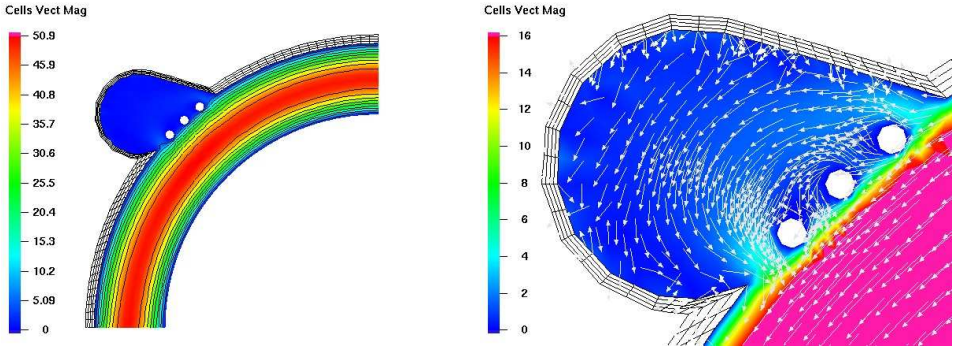


Figure 9: Elastic aneurysm wall with 3 stents.

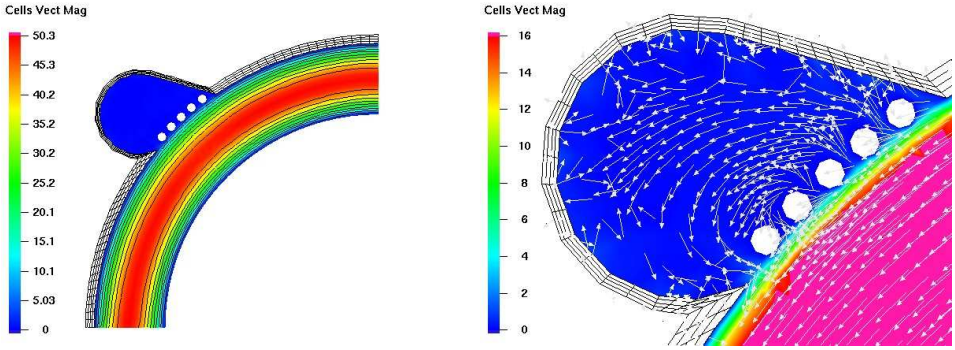


Figure 10: Elastic aneurysm wall with 5 stents.

Summarizing these results for steady inflow, the simulations show that the stent implantation across the neck of the aneurysm prevents blood penetration into the aneurysm fundus. Moreover, the elastic geometrical deformation of the wall is slightly reduced by implanting the stents while the local flow behaviour inside of the aneurysm is more significantly influenced by the elastic properties of the outer wall, particularly due to different width between stents and walls of the aneurysm. In the next section, we will consider the more realistic behaviour of flow configurations with time-dependent pulsatile inflow which will be analyzed for the case of elastic behaviour of the aneurysm walls.

5.2 Pulsatile configurations

For the following pulsatile test case, we have taken again the aneurysm part as elastic while the other parts of the walls belonging to the channel are rigid. First of all, we show again (see Fig. 11) the resulting volume of the flow domain for 5, 3 and no stents. In all cases, the oscillating behaviour due to the pulsative inflow is visible which also leads to different volume sizes. Looking carefully at the resulting flow behaviour, we see global differences w.r.t. the channel flow near the aneurysm, namely due to the different flow rate into the aneurysm, and significant local differences inside of the aneurysm.

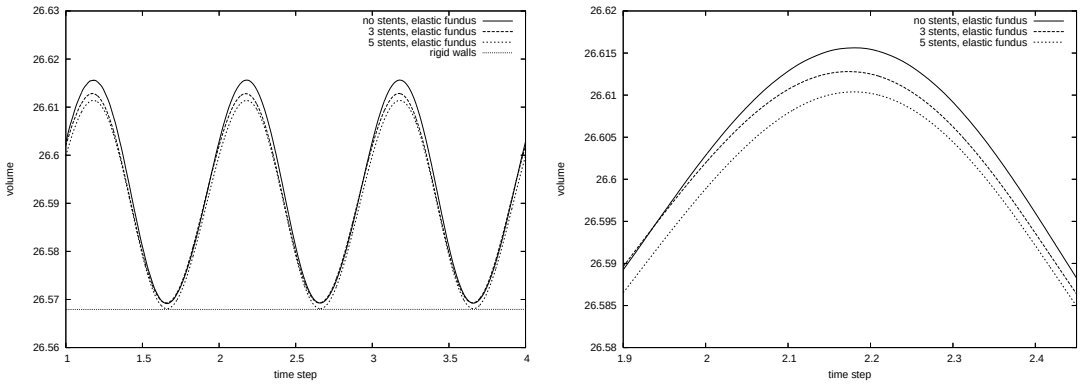


Figure 11: Volume of the domain with rigid and elastic behaviour of the aneurysm wall.

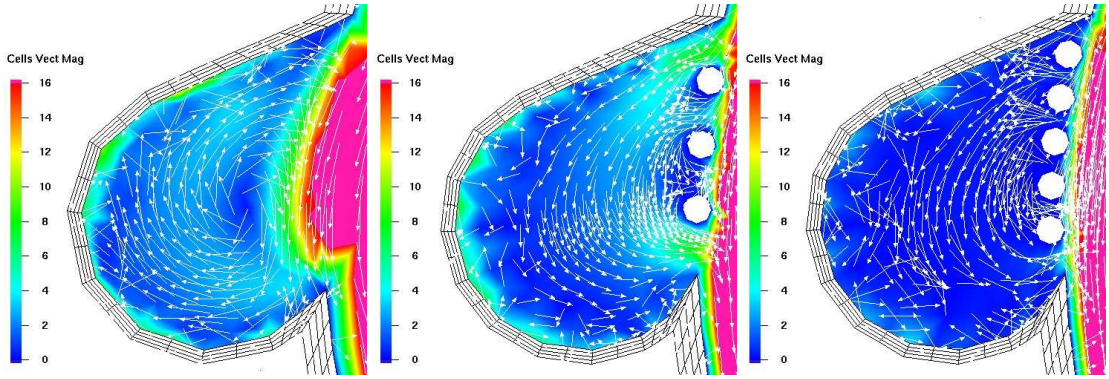


Figure 12: Snapshot for the magnitude of velocity for configurations with no, 3 and 5 stents.

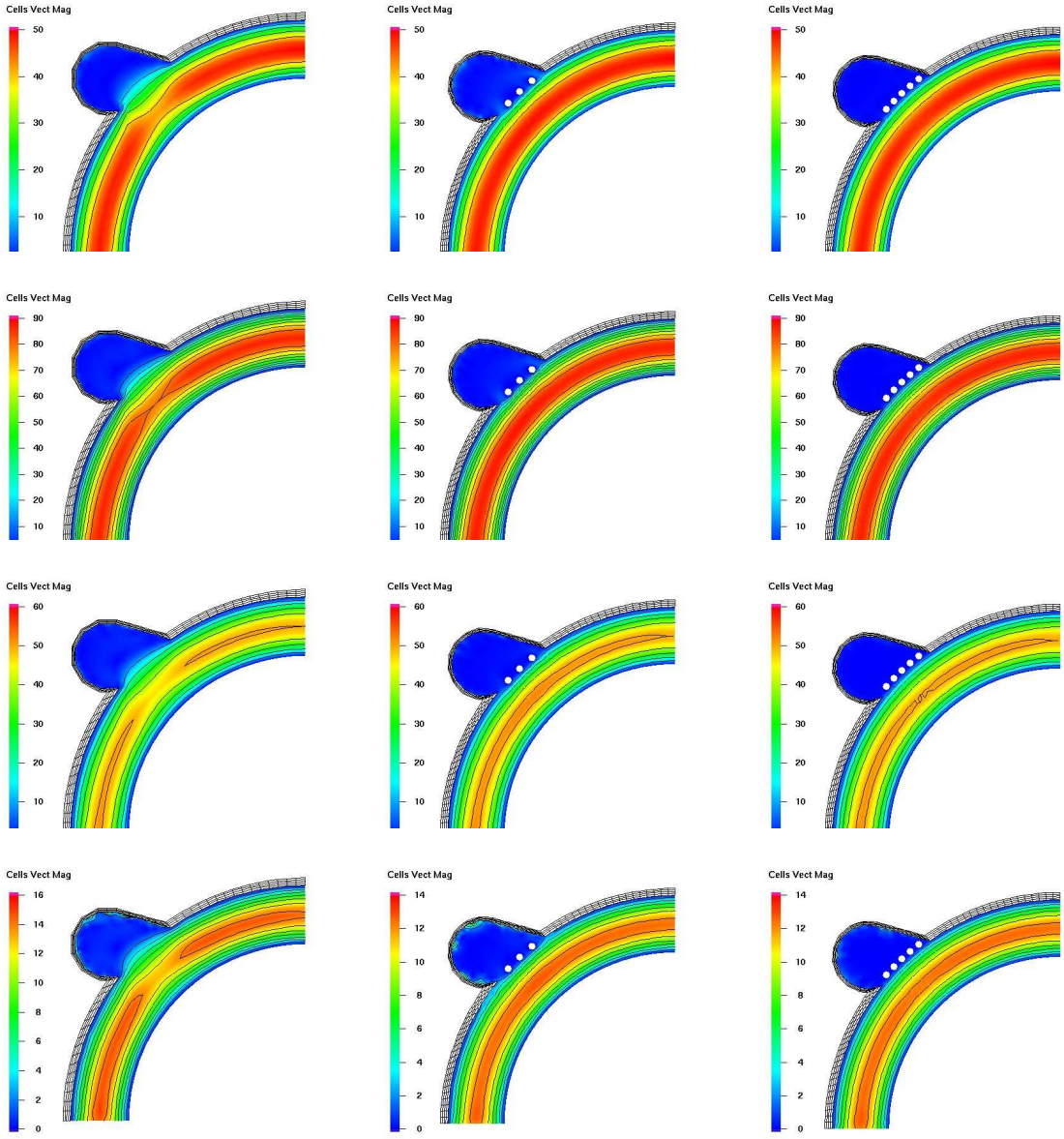


Figure 13: Left column: no stent. Middle column: 3 stents. Right column: 5 stents. Figures demonstrate the global behaviour of the velocity magnitude during one cycle.

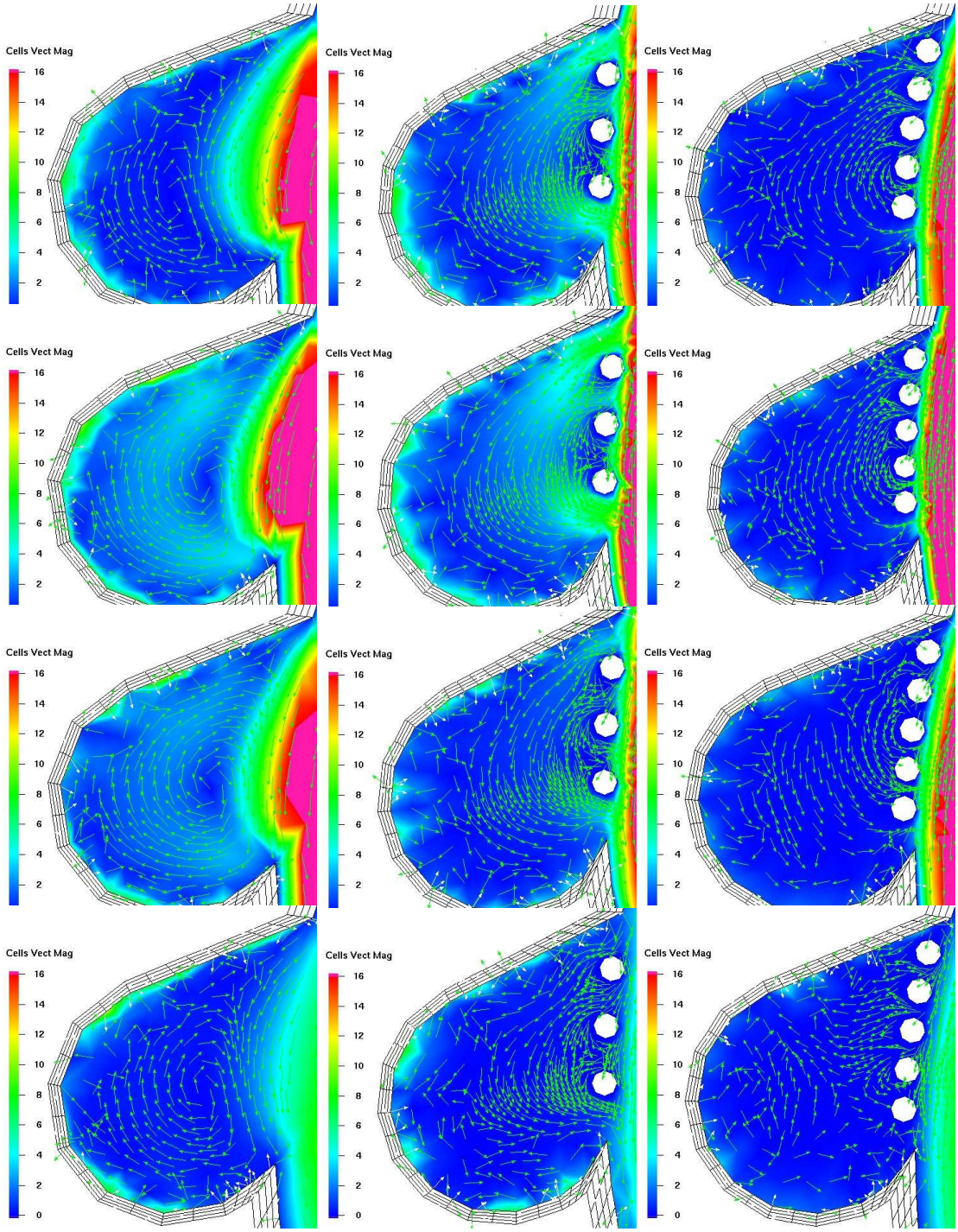


Figure 14: Left column: no stent. Middle column: 3 stents. Right column: 5 stents. Figures demonstrate the local behaviour of the fluid flow inside of the aneurysm during one cycle.

6 Summary and future developments

We presented a monolithic ALE formulation of fluid-structure interaction problems suitable for applications with finite deformations of the structure and laminar viscous flows, particularly arising in biomechanics. The corresponding discrete nonlinear systems result from the finite element discretization by using the high order Q_2P_1 FEM pair which are solved monolithically via discrete Newton iteration and special Krylov-multigrid approaches. While we restricted in the presented studies to the simplified case of newtonian fluids and small deformations, the used numerical components allow the system to be coupled with additional models of chemical and electric activation of the active response of the biological material as well as power law models used to describe the shear thinning property of blood. Further extension to viscoelastic models and coupling with mixture based models for soft tissues together with chemical and electric processes would allow to perform more realistic simulations for real applications.

In this contribution, we applied the presented numerical techniques to fluid-structure interaction problems which examine prototypically the influence of endovascular stent implantation onto aneurysm hemodynamics. The aim was, first of all, to study the influence of the elasticity of the walls onto the flow behaviour inside of the aneurysm. Moreover, different geometrical configurations of implanted stent structures have been analysed in 2D. These 2D results are far from providing quantitative results for such a complex multiphysics configuration, but they allow a qualitative analysis w.r.t. both considered components, namely the elastic behaviour of the structural parts and the multiscale flow behaviour due to the geometrical details of the stents. We believe that such basic studies may help towards the development of future 'Virtual Flow Laboratories' which individually assist to develop personal medical tools in an individual style.

Acknowledgment: The authors want to express their gratitude to the German Research Association (DFG), funding the project as part of FOR493 and TRR30, the Jindrich Necas Center for Mathematical Modeling, project LC06052 financed by MSMT, and the Higher Education Commission (HEC) of Pakistan for their financial support of the study. The present material is also based upon work kindly supported by the Homburger Forschungsförderungsprogramm (HOMFOR) 2008.

References

Appanaboyina, S.; Mut, F.; Löhner, R.; Scrivano, E.; Miranda, C.; Lylyk, P.; Putman, C.; Cebral, J.: Computational modelling of blood flow in side arterial branches after stenting of cerebral aneurysm. *International Journal of Computational Fluid Dynamics*, 22, (2008), 669–676.

Arnold, D. N.; Boffi, D.; Falk, R. S.: Approximation by quadrilateral finite element. *Math. Comput.*, 71, (2002), 909–922.

Barrett, R.; Berry, M.; Chan, T. F.; Demmel, J.; Donato, J.; Dongarra, J.; Eijkhout, V.; Pozo, R.; Romine, C.; Van der Vorst, H.: *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM, Philadelphia (PA 1994).

Boffi, D.; Gastaldi, L.: On the quadrilateral Q_2P_1 element for the stokes problem. *Int. J. Numer. Meth. Fluids.*, 39, (2002), 1001–1011.

Bramley, R.; Wang, X.: *SPLIB: A library of iterative methods for sparse linear systems*. Department of Computer Science, Indiana University, Bloomington, IN (1997), `http://www.cs.indiana.edu/ftp/bramley/splib.tar.gz`.

Damanik, H.; Hron, J.; Ouazzi, A.; Turek, S.: A monolithic FEM approach for non-isothermal incompressible viscous flows. In: *Journal of Computational Physics*, accepted (2008).

Davis, T. A.; Duff, I. S.: A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *SACM Trans. Math. Software*, 25, (1999), 1–19.

Fernandez, M. A.; Gerbeau, J.-F.; Martin, V.: Numerical simulation of blood flows through a porous interface. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42, (2008), 961–990.

Gresho, P. M.: On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, part 1: Theory. *Int. J. Numer. Meth. Fluids.*, 11, (1990), 587–620.

Hron, J.; Ouazzi, A.; Turek, S.: A computational comparison of two fem solvers for nonlinear incompressible flow. In: E. Bänsch, ed., *Challenges in Scientific Computing*, LNCSE, pages 87–109, Springer (2002).

Hron, J.; Turek, S.: *Lecture Notes in Computational Science and Engineering, Fluid –Structure Interaction – Modelling, Simulation, Optimization*. Springer, ISBN 3–540–34595–7 (2006a).

Hron, J.; Turek, S.: A monolithic FEM/multigrid solver for ALE formulation of fluid structure interaction with application in biomechanics. In: H.-J. Bungartz; M. Schäfer, eds., *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, LNCSE, Springer (2006b).

Löhner, R.; Cebral, J.; Appanaboyina, S.: Parabolic recovery of boundary gradients. *Communications in Numerical Methods in Engineering*, 24, (2008), 1611–1615.

Rannacher, R.; Turek, S.: A simple nonconforming quadrilateral stokes element. *Numer. Methods Partial Differential Equations.*, 8, (1992), 97–111.

Razzaq, M.: *Numerical techniques for solving fluid-structure interaction problems with applications to bioengineering*. PhD Thesis, TU Dortmund, to appear (2009).

Razzaq, M.; Hron, J.; Turek, S.: Numerical simulation of laminar incompressible fluid-structure interaction for elastic material with point constraintst. In: *Advances in Mathematical Fluid Mechancis-Dedicated to Giovanni paolo Galdi on the Occasion of his 60th Birthday*, Springer, submitted (2008).

Tezduyar, T.; Sathe, S.; Cragin, T.; Nanna, B.; Conklin, B.; Pausewang, J.; Schwaab, M.: Modeling of fluid structure interactions with the space time finite elements: Arterial fluid mechanics. *International Journal for Numerical Methods in Fluids*, 54, (2007), 901–922.

Tezduyar, T.; Sathe, S.; Schwaab, M.; Conklin, B.: Arterial fluid mechanics modeling with the stabilized space time fluid structure interaction technique. *International Journal for Numerical Methods in Fluids*, 57, (2008), 601–629.

Torri, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Influence of wall elasticity in patient-specific hemodynamic simulations. *Computers and Fluids*, 36, (2007a), 160–168.

Torri, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Numerical investigation of the effect of hypertensive blood pressure on cerebral aneurysm dependence of the effect on the aneurysm shape. *International Journal for Numerical Methods in Fluids*, 54, (2007b), 995–1009.

Turek, S.: *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer-Verlag (1999).

Turek, S.; Rivkind, L.; Hron, J.; Glowinski, R.: Numerical study of a modified time-steeping theta-scheme for incompressible flow simulations. *Journal of Scientific Computing*, 28, (2006), 533–547.

Turek, S.; Schmachtel, R.: Fully coupled and operator-splitting approaches for natural convection flows in enclosures. *International Journal for Numerical Methods in Fluids*, 40, (2002), 1109–1119.

Valencia, A.; Ladermann, D.; Rivera, R.; Bravo, E.; Galvez, M.: Blood flow dynamics and fluid–structure interaction in patient -specific bifurcating cerebral aneurysms. *International Journal for Numerical Methods in Fluids*, 58, (2008), 1081–1100.

Vanka, S.: Implicit multigrid solutions of Navier-Stokes equations in primitive variables. *J. of Comp. Phys.*, 65, (1985), 138–158.

Address: M. Razzaq (corresponding author), S. Turek, and J. F. Acker, Angewandte Mathematik und Numerik, (LS III), TU Dortmund, Vogelpothsweg 87, D-44227, Dortmund, Germany. (mrazzaq@math.tu-dortmund.de)
J. Hron, Institute of Mathematics, Charles University, Prague, Czech Republic.
F. Weichert, Department of Computer Graphics, TU Dortmund, Germany.
I. Q. Grunwald, C. Roth, Department of Neuroradiology, M. Wagner, Department of Pathology,
Saar State University Medical School, Homburg Saar, Germany.
B. F. Romeike, Department of Neuropathology, Friedrich-Schiller University Jena, Germany.

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 231– 252
Herrsching am Ammersee, 29.9.-1.10.2008

Multiscale Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) Technique

Tayfun E. Tezduyar, Kenji Takizawa and Jason Christopher

We present the multiscale versions and extensions of the Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique. The SCAFSI technique was introduced as an approximate FSI approach in arterial fluid mechanics. It is based on the assumption that the arterial deformation during a cardiac cycle is driven mostly by the blood pressure. First we compute a "reference" arterial deformation as a function of time, driven only by the blood pressure profile of the cardiac cycle. Then we compute a sequence of updates involving mesh motion, fluid dynamics calculations, and recomputing the arterial deformation. Although the SCAFSI technique was developed and tested in conjunction with the stabilized space–time FSI (SSTFSI) technique, it can also be used in conjunction with other moving-mesh FSI modeling approaches. The SSTFSI technique is based on the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation and is supplemented with a number of special techniques developed for arterial FSI. These include a recipe for pre-FSI computations that improve the convergence of the FSI computations, using an estimated zero-pressure arterial geometry, layers of refined fluid mechanics mesh near the arterial walls, and a special mapping technique for specifying the velocity profile at an inflow boundary with non-circular shape. The SCAFSI technique provides a computationally more economical alternative to the fully coupled FSI approach in arterial fluid mechanics. It also brings additional flexibility, such as being able to carry out the computations in a spatially or temporally multiscale fashion. In the test computations we report here for the spatially multiscale versions and extensions of the SCAFSI technique, we focus on a patient-specific middle cerebral artery segment with aneurysm, where the arterial geometry is based on computed tomography images. The arterial structure is modeled with the continuum element made of hyperelastic (Fung) material.

1 Introduction

We are seeing more research emphasis on computation of arterial fluid–structure interaction (FSI) problems (see, for example, Torii et al. (2004); Gerbeau et al. (2005); Torii et al. (2006a,b); Bazilevs et al. (2006); Torii et al. (2007a); Tezduyar et al. (2007b); Torii et al. (2007b); Tezduyar et al. (2008); Bazilevs et al. (2008); Torii et al. (2008); Tezduyar et al. (July 2008); Torii et al. (July 2008)). While an increased emphasis on biomechanics research is part of the reason, it is also because there has been much emphasis on FSI modeling in general (see, for example, Tezduyar et al. (1993, 1994); Mittal and Tezduyar (1994, 1995); Johnson and Tezduyar (1999); Kalro and Tezduyar (2000); Stein et al. (2000); Tezduyar and Osawa (2001); Ohayon (2001); Tezduyar et al. (2004); Torii et al. (2004); van Brummelen and de Borst (2005); Michler et al. (2005); Gerbeau et al. (2005); Tezduyar et al. (2006a,b); Torii et al. (2006a); Tezduyar et al. (2006c); Torii et al. (2006b); Dettmer and Peric (2006); Bazilevs et al. (2006); Khurram and Masud (2006); Kuttler et al. (2006); Lohner et al. (2006); Bletzinger et al. (2006); Torii et al. (2007a); Masud et al. (2007); Sawada and Hisada (2007); Wall et al. (2007); Tezduyar and Sathe (2007); Tezduyar et al. (2007b); Torii et al. (2007b); Tezduyar et al. (2008); Bazilevs et al. (2008); Kuttler and Wall (2008); Dettmer and Peric (2008)). The Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation Tezduyar (1992); Tezduyar et al. (1992a,c); Tezduyar (2003) was introduced in 1991 as a general-purpose interface-tracking (i.e. moving-mesh) technique for simulation of flow problems with moving boundaries or interfaces, including FSI. The stabilization components used are the Streamline-Upwind/Petrov-Galerkin (SUPG) Hughes and Brooks (1979); Brooks and Hughes (1982) and Pressure-Stabilizing/Petrov-Galerkin (PSPG) Tezduyar (1992); Tezduyar et al. (1992d) methods. An earlier version of the pressure stabilization, for Stokes flows, was introduced in Hughes et al. (1986). The DSD/SST formulation, together with the mesh update methods Tezduyar et al. (1992b); Johnson and Tezduyar (1994); Tezduyar (2001) developed in conjunction with the DSD/SST formulation and block-iterative coupling Tezduyar (2004) (see Tezduyar et al. (2006a,b); Tezduyar and Sathe (2007) for the

terminology), has been the core technology used in the arterial FSI computations reported by Torii et al. Torii et al. (2004, 2006a,b, 2007a,b, 2008, July 2008) for patient-specific image-based geometries. The cases studied in these articles by Torii et al. were almost all for middle cerebral artery segments with aneurysm, and the geometries were constructed from computed tomography images.

The stabilized space–time FSI (SSTFSI) technique was introduced recently in Tezduyar and Sathe (2007). It is based on the new-generation DSD/SST formulations, which were also introduced in Tezduyar and Sathe (2007). The SSTFSI technique was extended in Tezduyar et al. (2007b, 2008, July 2008) to arterial FSI, with emphasis on arteries with aneurysm. The arterial geometries were approximations to patient-specific image-based geometries, mainly to those reported by Torii et al. A number of special techniques for arterial FSI were developed in conjunction with the SSTFSI technique. These include using an estimated zero-pressure arterial geometry Tezduyar et al. (2007a, 2008), a special mapping technique for specifying the velocity profile at an inflow boundary with non-circular shape, using layers of refined fluid mechanics mesh near the arterial walls Tezduyar et al. (July 2008), a recipe for pre-FSI computations that improve the convergence of the FSI computations Tezduyar et al. (2007b, 2008), and the Sequentially-Coupled Arterial FSI (SCAFSI) technique Tezduyar et al. (2007c, 2008, July 2008) The need for an estimated zero-pressure arterial geometry is based on recognizing that the patient-specific image-based geometries correspond to time-averaged blood pressure values. With the special mapping technique for inflow boundaries with non-circular shapes, we can specify the velocity profiles in a way that is somewhat consistent with a preferred profile, such as a parabolic profile or Womersley solution Womersley (1955). With the explicitly-contolled mesh refinement near the arterial walls, we can increase the accuracy in computing the wall shear stress. The recipe for pre-FSI computations is based on the assumption that the arterial deformation during a cardiac cycle is driven mostly by the blood pressure. The SCAFSI technique, which is an approximate FSI approach in arterial fluid mechanics, is also based on that assumption.

The SCAFSI technique was introduced in Tezduyar et al. (2007c, 2008), where the technique was in early stages of its development, the description was rather cursory, and the test computations were limited. A more extensive description of the SCAFSI technique was provided in Tezduyar et al. (July 2008), together with a wider set of test computations. The test computations were for the abdominal aortic and cerebral aneurysms, and the arterial geometries used in the computations were close approximations to patient-specific image-based data. The multiscale versions and extensions of the SCAFSI technique were introduced in Tezduyar et al. (July 2008), and the test computations were presented for the temporally multiscale version, using different time step sizes for the structural and fluid mechanics parts. In the spatially multiscale versions and extensions proposed in Tezduyar et al. (July 2008), fluid mechanics meshes with different refinement levels are used at different stages of the FSI computation. We use a relatively coarser mesh at the early stages and reserve the highly-refined mesh for the stage where we plan to do the high-accuracy fluid mechanics computations, such as the wall shear stress computation. In this paper we present the test computations we have recently carried out with the spatially multiscale versions and extensions. We use actual patient-specific image-based data. Specifically, we focus on the bifurcating middle cerebral artery segment of a 67 year-old female with aneurysm, which was reported in Torii et al. (2007b, 2008, July 2008). The structural modeling for the arteries is based on the continuum element made of hyperelastic (Fung) material. In this paper we also evaluate the performances of different types of structural mechanics meshes.

For the governing equations, the SSTFSI technique, the concept of estimated zero-pressure arterial geometry, and the recipe for pre-FSI computation, we refer the reader to Tezduyar and Sathe (2007); Tezduyar et al. (2007b, 2008). The SCAFSI algorithm is described in Section 2, and its multiscale versions and extensions in Section 3. The special mapping technique is described in Section 4. General conditions for the test computations are given in Section 5, and the test results are presented in Section 6. In Section 7, we extend the multiscale sequentially-coupled FSI technique from arterial fluid mechanics to other classes of applications. The concluding remarks are given in Section 8.

2 SCAFSI Algorithm

The SCAFSI stage of the computations is preceded by a set of pre-FSI computation steps., which can be found in Tezduyar et al. (2007b, 2008). In the SCAFSI technique, first we compute a “reference” (i.e. “base”) arterial deformation as a function of time, driven only by the blood pressure, which is given as a function of time by specifying the pressure profile in a cardiac cycle. Then we compute a sequence of updates involving mesh motion, fluid dynamics calculations, and recomputing the arterial deformation. The SCAFSI steps are described below.

Step 1

Compute the "reference" arterial displacement:

$(\mathbf{Y}_R)_n \quad n = 1, 2, \cdots n_{ts}$

Driven only by the blood pressure: $p_R(t)$

Predictor options in moving from time level n to $n+1$:

$$\begin{aligned}
((\mathbf{Y}_R)_{n+1})^0 &= (\mathbf{Y}_R)_n && (1)\\
((\mathbf{Y}_R)_{n+1})^0 &= 2(\mathbf{Y}_R)_n - (\mathbf{Y}_R)_{n-1} && (2)\\
((\mathbf{Y}_R)_{n+1})^0 &= 3(\mathbf{Y}_R)_n - 3(\mathbf{Y}_R)_{n-1} + (\mathbf{Y}_R)_{n-2} && (3)\\
((\mathbf{Y}_R)_{n+1})^0 &= (\mathbf{Y}_R)_n + \frac{(\mathbf{Y}_R)_n - (\mathbf{Y}_R)_{n-1}}{p_R(t_n) - p_R(t_{n-1})} \left(p_R(t_{n+1}) - p_R(t_n)\right) && (4)
\end{aligned}$$

Nodal values of $p_R(t_n)$: $(\mathbf{P}_R)_n$

Nodal values of the interface stress: $(\mathbf{H}_R)_n$

Step 2

Compute the "reference" mesh motion:

$(\mathbf{V}_R)_n \quad n = 1, 2, \cdots n_{ts}$

Predictor options:

$$\begin{aligned}
((\mathbf{V}_R)_{n+1})^0 &= \mathbf{0} && (5)\\
((\mathbf{V}_R)_{n+1})^0 &= (\mathbf{V}_R)_n && (6)\\
((\mathbf{V}_R)_{n+1})^0 &= 2(\mathbf{V}_R)_n - (\mathbf{V}_R)_{n-1} && (7)\\
((\mathbf{V}_R)_{n+1})^0 &= 3(\mathbf{V}_R)_n - 3(\mathbf{V}_R)_{n-1} + (\mathbf{V}_R)_{n-2} && (8)\\
((\mathbf{V}_R)_{n+1})^0 &= (\mathbf{V}_R)_n + \frac{(\mathbf{V}_R)_n - (\mathbf{V}_R)_{n-1}}{p_R(t_n) - p_R(t_{n-1})} \left(p_R(t_{n+1}) - p_R(t_n)\right) && (9)
\end{aligned}$$

Step 3

For zero-stress conditions at the outflow boundaries, compute the time-dependent flow field and the corresponding interface stress: $(\mathbf{H}_1)_n \quad n = 1, 2, \cdots n_{ts}$

Predictor options:

$$\begin{aligned}
((\mathbf{P}_1)_{n+1})^0 &= (\mathbf{P}_1)_n && (10)\\
((\mathbf{P}_1)_{n+1})^0 &= 2(\mathbf{P}_1)_n - (\mathbf{P}_1)_{n-1} && (11)\\
((\mathbf{P}_1)_{n+1})^0 &= 3(\mathbf{P}_1)_n - 3(\mathbf{P}_1)_{n-1} + (\mathbf{P}_1)_{n-2} && (12)\\
((\mathbf{P}_1)_{n+1})^0 &= (\mathbf{P}_1)_n + \frac{(\mathbf{P}_1)_n - (\mathbf{P}_1)_{n-1}}{U(t_n) - U(t_{n-1})} \left(U(t_{n+1}) - U(t_n)\right) && (13)
\end{aligned}$$

Here $U(t)$ is the cross-sectional average of the inflow velocity. To enhance the stability of the computation in Step 4, smoothen $(\mathbf{H}_1)_n$ by time averaging:

$$\begin{aligned}(\mathbf{H}_1)_n \leftarrow \omega_0(\mathbf{H}_1)_n &+ \omega_{\pm1}\left((\mathbf{H}_1)_{n+1} + (\mathbf{H}_1)_{n-1}\right) + \omega_{\pm2}\left((\mathbf{H}_1)_{n+2} + (\mathbf{H}_1)_{n-2}\right) \\ &+ \omega_{\pm3}\left((\mathbf{H}_1)_{n+3} + (\mathbf{H}_1)_{n-3}\right) + \omega_{\pm4}\left((\mathbf{H}_1)_{n+4} + (\mathbf{H}_1)_{n-4}\right) \end{aligned} \tag{14}$$

Options for time-averaging weights:

$$(\omega_0, \omega_{\pm1}, \omega_{\pm2}, \omega_{\pm3}, \omega_{\pm4}) = \frac{1}{9}(3, 2, 1, 0, 0) \tag{15}$$

$$(\omega_0, \omega_{\pm1}, \omega_{\pm2}, \omega_{\pm3}, \omega_{\pm4}) = \frac{1}{16}(4, 3, 2, 1, 0) \tag{16}$$

$$(\omega_0, \omega_{\pm1}, \omega_{\pm2}, \omega_{\pm3}, \omega_{\pm4}) = \frac{1}{25}(5, 4, 3, 2, 1) \tag{17}$$

Now the total interface stress: $(\mathbf{H}_R)_n + (\mathbf{H}_1)_n$

Step 4

Compute the updated arterial displacement:

$\mathbf{Y}_n \quad n = 1, 2, \cdots n_{ts}$

Predictor options:

$$(\mathbf{Y}_{n+1})^0 = 2\mathbf{Y}_n - \mathbf{Y}_{n-1} \tag{18}$$

$$(\mathbf{Y}_{n+1})^0 = (\mathbf{Y}_R)_{n+1} + ((\mathbf{Y}_1)_{n+1})^0 \tag{19}$$

Displacement increment: $(\mathbf{Y}_1)_n = \mathbf{Y}_n - (\mathbf{Y}_R)_n$

Predictor options for the displacement increment:

$$((\mathbf{Y}_1)_{n+1})^0 = (\mathbf{Y}_1)_n \tag{20}$$

$$((\mathbf{Y}_1)_{n+1})^0 = 2(\mathbf{Y}_1)_n - (\mathbf{Y}_1)_{n-1} \tag{21}$$

$$((\mathbf{Y}_1)_{n+1})^0 = 3(\mathbf{Y}_1)_n - 3(\mathbf{Y}_1)_{n-1} + (\mathbf{Y}_1)_{n-2} \tag{22}$$

Step 5

Compute the updated mesh motion:

$\mathbf{V}_n \quad n = 1, 2, \cdots n_{ts}$

Predictor options:

$$(\mathbf{V}_{n+1})^0 = \mathbf{0} \tag{23}$$

$$(\mathbf{V}_{n+1})^0 = (\mathbf{V}_R)_{n+1} + ((\mathbf{V}_1)_{n+1})^0 \tag{24}$$

Mesh-motion increment: $(\mathbf{V}_1)_n = \mathbf{V}_n - (\mathbf{V}_R)_n$

Predictor options for the mesh-motion increment:

$$\left((\mathbf{V}_1)_{n+1}\right)^0 = \mathbf{0} \tag{25}$$
$$\left((\mathbf{V}_1)_{n+1}\right)^0 = (\mathbf{V}_1)_n \tag{26}$$
$$\left((\mathbf{V}_1)_{n+1}\right)^0 = 2(\mathbf{V}_1)_n - (\mathbf{V}_1)_{n-1} \tag{27}$$
$$\left((\mathbf{V}_1)_{n+1}\right)^0 = 3(\mathbf{V}_1)_n - 3(\mathbf{V}_1)_{n-1} + (\mathbf{V}_1)_{n-2} \tag{28}$$

Step 6

For zero-stress conditions at the outflow boundaries, compute the time-dependent flow field and the corresponding interface stress: $(\mathbf{H}_2)_n \quad n = 1, 2, \cdots n_{ts}$

Predictor options:

$$\left((\mathbf{P}_2)_{n+1}\right)^0 = (\mathbf{P}_2)_n \tag{29}$$
$$\left((\mathbf{P}_2)_{n+1}\right)^0 = 2(\mathbf{P}_2)_n - (\mathbf{P}_2)_{n-1} \tag{30}$$
$$\left((\mathbf{P}_2)_{n+1}\right)^0 = 3(\mathbf{P}_2)_n - 3(\mathbf{P}_2)_{n-1} + (\mathbf{P}_2)_{n-2} \tag{31}$$
$$\left((\mathbf{P}_2)_{n+1}\right)^0 = (\mathbf{P}_2)_n + \frac{(\mathbf{P}_2)_n - (\mathbf{P}_2)_{n-1}}{U(t_n) - U(t_{n-1})} \left(U(t_{n+1}) - U(t_n)\right) \tag{32}$$
$$\left((\mathbf{P}_2)_{n+1}\right)^0 = (\mathbf{P}_1)_{n+1} \tag{33}$$

Now the total interface stress: $(\mathbf{H}_R)_n \; + \; (\mathbf{H}_2)_n$

The SCAFSI algorithm described above is based on the assumption that in computations with more than one outflow boundary, we specify the same traction condition for all. Versions of the SCAFSI technique that do not rely on that assumption were proposed in Tezduyar et al. (July 2008). These versions are applicable even if the outflow traction conditions are not specified explicitly but are modeled as a function of the flow rate at each outflow boundary. We refer the interested reader to Tezduyar et al. (July 2008).

Remark 1 *Clearly, the SCAFSI technique result in savings in computer time compared to the (fully) coupled arterial FSI (CAFSI) technique. These savings come from various aspects of SCAFSI, which can be found in Tezduyar et al. (July 2008).*

Remark 2 *The predictors given by Eqs. (30) and (31) were written in Tezduyar et al. (July 2008) with a subscript typo in each equation. Although the typos were obvious, we wanted to point that out here.*

Remark 3 *Due to a combination of publisher's typesetting errors and misinterpretation of what has been used in the computations, the predictor options identified in Tezduyar et al. (July 2008) as those used in the test computations were not the ones that were actually used. The predictor options used, in reference to the equation numbers of this paper, were those given by Eqs. (2), (5), (10), (18), (23) and (33). We wanted to correct that here.*

3 Multiscale Versions and Extensions of the SCAFSI Technique

Temporally multiscale. A temporally multiscale version of the SCAFSI technique was proposed in Tezduyar et al. (July 2008), where different time step sizes are used for the structural and fluid mechanics parts. This version was tested in Tezduyar et al. (July 2008) on FSI modeling of a middle cerebral artery segment with aneurysm. The arterial geometry was a close approximation to the patient-specific image-based geometry used in Torii et al. (2007a). The geometry used in Torii et al. (2007a) was extracted from the computed tomography model of an artery

segment from a 57 year-old male. The arterial wall was modeled with the continuum element made of hyperelastic (Fung) material. The mesh for the artery had four-node tetrahedral elements, with two elements across the arterial wall. The time step size for the structural mechanics part was twice that of the fluid mechanics part. The multiscale SCAFSI computation resulted in good mass balance, and the flow field obtained looked essentially the same as the flow field from the CAFSI computation. Time histories of the arterial volume and (spatially-averaged) interface stress obtained with the multiscale SCAFSI technique were also very close to those obtained with the CAFSI technique.

Spatially multiscale. Spatially multiscale versions and extensions of SCAFSI techniques were also proposed in Tezduyar et al. (July 2008), where the fluid mechanics meshes with different refinement levels are used at different stages of the SCAFSI computation. In the version called SCAFSI M1SC, a more refined fluid mechanics mesh is used at SCAFSI Steps 5 and 6 than the mesh used at Steps 2 and 3. With this approach, we can increase the accuracy of the fluid mechanics solution at the final stage, just before we calculate the fluid mechanics quantities that we are interested in, such as the wall shear stress. By using a relatively coarser fluid mechanics mesh at Steps 2 and 3, we avoid incurring high computational cost at stages where a highly-refined fluid mechanics mesh is not needed for accurately computing the arterial shape as a function of time. In the extension called SCAFSI M1C, we first compute the arterial shape with the CAFSI technique and a relatively coarser fluid mechanics mesh, followed by mesh motion and fluid mechanics computations with a more refined mesh. Again, by using a relatively coarser mesh at the stage where a highly-refined fluid mechanics mesh is not needed, we reserve our computational effort for the final stage, where we do need a highly-refined fluid mechanics mesh to calculate the fluid mechanics quantities such as the wall shear stress. We present the test computations we have recently carried with the SCAFSI M1SC and SCAFSI M1C techniques in Section 6.

4 Special Techniques

Time-integration point for the incompressibility constraint. For integration of the incompressibility-constraint term over each space–time slab, we propose to use only one integration point in time, shifted to the upper time level of the slab. All other terms in the space–time finite element formulation would be integrated by using Gaussian quadrature points in time, with the number of points set to whatever we intended to have for the overall formulation. With this technique, for any time level n, as we move to level $n+1$, the incompressibility constraint equation focuses on the velocity field $(\mathbf{u}^h)^-_{n+1}$.

Special mapping. Some inflow profiles require the inlet to be circular, however, the inlets in many of the geometries we encounter are not circular. Furthermore, as the artery deforms, the inlet shape changes. Thus, even if the inlet is initially circular, it will not remain so once deformed. To address this requirement, we have developed a special mapping technique to map the inflow boundaries from non-circular shapes to circular shapes. Here the actual inflow profile $U(\boldsymbol{z},t)$, where $\boldsymbol{z}$ is the coordinate vector in the inflow plane, is obtained by mapping from a preferred inflow profile $U^{\mathrm{P}}(r,t)$, where r is the circular coordinate. The variable r is defined at $0 \leq r \leq r_{\mathrm{B}}$, where r_{B} is the boundary radius. The method involves the following two steps:

1. Map $\boldsymbol{z}$ to r:

$$r(\boldsymbol{z}) = \frac{||\boldsymbol{z}-\boldsymbol{z}_{\mathrm{C}}||}{||\boldsymbol{z}-\boldsymbol{z}_{\mathrm{B}}|| + ||\boldsymbol{z}-\boldsymbol{z}_{\mathrm{C}}||}\, r_{\mathrm{B}}, \tag{34}$$

   where subscripts "C" and "B" indicate the centroid and the closest boundary point Using this mapping, we first calculate a "trial" velocity:

$$U^{\mathrm{T}}(\boldsymbol{z},t) = U^{\mathrm{P}}(r,t), \tag{35}$$

   where the superscript "T" stands for "trial".

2. Adjust the velocity:

$$U(\boldsymbol{z},t) \;=\; \frac{Q(t)}{\int_{\Gamma_{\mathrm{INFL}}} U^{\mathrm{T}}(\boldsymbol{z},t)\,d\Gamma}\, U^{\mathrm{T}}(\boldsymbol{z},t), \tag{36}$$

   where Q is the flow rate and Γ_{INFL} is the discretized inflow area; i.e. the integration area in the finite element space.

This technique is applicable to any preferred inflow profile, including the Womersley profile Womersley (1955), as described in the next section.

5 General Conditions for the Test Computations

All computations were carried out in a parallel computing environment. All computations were completed without any remeshing. The fully-discretized, coupled fluid and structural mechanics and mesh-moving equations are solved with the quasi-direct coupling technique (see Section 5.2 in Tezduyar and Sathe (2007)). In solving the linear equation systems involved at every nonlinear iteration, the GMRES search technique Saad and Schultz (1986) is used with a diagonal preconditioner. The S→F→S→FSI sequence is used in the computations (see Section 6.2 in Tezduyar et al. (2008)). This is slightly different from the S→F→FSI sequence described in Tezduyar et al. (2008) in that it includes an extra structural mechanics step which incorporates the traction obtained from the fluid computation. This helps to match the structure mesh to the fluid solution and provides a better starting point for the FSI computations. In the spatially multiscale SCAFSI computations, Step 1–6 predictor options used are those given by Eqs. (2), (5), (10), (18), (23) and (29). We note that the pressure predictor option used at Step 6 is different than the one used in the SCAFSI computations reported in Tezduyar et al. (July 2008). This is because in the spatially multiscale SCAFSI computations the fluid mechanics mesh at (Step 5 and) Step 6 is different than the fluid mechanics mesh used at the earlier stages, and therefore the pressure values obtained at the earlier stages cannot be directly used as pressure predictors at Step 6. The time-averaging weights are those given by Eq. (17). The time-averaging, however, is not used in the SCAFSI M1C computation. The predictor options we are using are relatively simple ones among those proposed. We do not expect that using more sophisticated predictors would change the results that much, since the SCAFSI results are already quite close to the CAFSI results. Among the options proposed for the time-averaging weights, we are using the one with the largest spread. We have not experimented with the options with narrower spread.

Fluid and structure properties. As it was done for the computations reported in Torii et al. (2004, 2006a,b, 2007a,b), the blood is assumed to behave like a Newtonian fluid (see Section 2.1 in Tezduyar et al. (2008)). The density and kinematic viscosity are set to 1,000 kg/m^3 and 4.0×10^{-6} m^2/s. The material density of the arterial wall is known to be close to that of the blood and therefore set to 1,000 kg/m^3. The arterial wall is modeled with the continuum element made of hyperelastic (Fung) material. The Fung material constants D_1 and D_2 (from Huang et al. (2001)) are 2.6447×10^3 N/m^2 and 8.365, and the penalty Poisson's ratio is 0.45. Arteries are surrounded by tissues, and we expect those tissues to have a damping effect on the structural dynamics of the arteries. Therefore we add a mass-proportional damping, which also helps in removing the high-frequency modes of the structural deformation. The damping coefficient η is chosen in such a way that the structural mechanics computations remain stable at the time-step size used. The value of η used in the test computations reported in this paper will be given in the section where we describe those test computations.

Boundary conditions. At the inflow boundary we specify the velocity profile as a function of time. We use a velocity waveform which represents the cross-sectional maximum velocity as a function of time, where $A_k \in \mathbb{C}$ are the Fourier coefficients of the waveform. Assuming that the maximum velocity occurs at $r = 0$ and that the artery is rigid and the cross-sectional shape is a perfect circle, we can apply the Womersley Womersley (1955) solution as follows:

$$U^{\mathrm{P}}(r,t) = A_0\left(1-\left(\frac{r}{r_{\mathrm{B}}}\right)^2\right) + \sum_{k=1}^{N} A_k \frac{J_0(\Upsilon\sqrt{k}\imath^{\frac{3}{2}}) - J_0(\Upsilon\sqrt{k}\left(\frac{r}{r_{\mathrm{B}}}\right)\imath^{\frac{3}{2}})}{J_0(\Upsilon\sqrt{k}\imath^{\frac{3}{2}}) - 1}\exp\left(\imath 2\pi k\frac{t}{T}\right). \tag{37}$$

Here N is the number of Fourier coefficients (we use $N = 20$), T is the period of the cardiac cycle, J_0 is the Bessel functions of the first kind of order 0, $\imath$ is the imaginary number, and Υ is the Womersley parameter: $\Upsilon = r_{\mathrm{B}}\sqrt{(2\pi)/(\nu T)}$, and ν is the kinematic viscosity. We use the special mapping technique described in Section 4 for non-circular shapes. At the two outflow boundaries, we specify the same traction boundary condition. The traction condition is based on a pressure profile, which, as a function of time, is determined based on the flow

rate using the Windkessel model Otto (1899). From Eq. (37), we obtain the flow rate:

$$
\begin{aligned}
Q(t) &= \int_0^{r_{\mathrm{B}}} 2\pi r U^{\mathrm{P}}(r,t)dr \\
&= \pi r_{\mathrm{B}}^2 \frac{A_0}{2} + \pi r_{\mathrm{B}}^2 \sum_{n=1}^{N} A_k \frac{J_0(\Upsilon\sqrt{k}\imath^{\frac{3}{2}}) - 2\left(\Upsilon\sqrt{k}\imath^{\frac{3}{2}}\right)^{-1} J_1(\Upsilon\sqrt{k}\imath^{\frac{3}{2}})}{J_0(\Upsilon\sqrt{k}\imath^{\frac{3}{2}}) - 1} \exp\left(\imath 2\pi k \frac{t}{T}\right), \qquad (38) \\
&= \sum_{k=0}^{N} B_k \exp\left(\imath 2\pi k \frac{t}{T}\right), \qquad (39)
\end{aligned}
$$

where J_1 is the Bessel function of the first kind of order 1, and for notational convenience we introduce another set of coefficients, $B_k \in \mathbb{C}$. The pressure, based on the Windkessel model, can be written as

$$
p(t) = \exp\left(-\frac{t}{RC}\right) \int_0^t \frac{1}{C} Q(\tau) \exp\left(\frac{\tau}{RC}\right) d\tau + p_0, \qquad (40)
$$

where C and R are the compliance and resistance of the distal arterial networks, and p_0 is a constant of integration. Substituting Eq. (39) into Eq. (40), we obtain:

$$
p(t) = \sum_{k=0}^{N} \frac{B_k}{\imath 2\pi k \frac{C}{T} + \frac{1}{R}} \left[\exp\left(\imath 2\pi k \frac{t}{T}\right) - \exp\left(-\frac{t}{RC}\right)\right] + p_0. \qquad (41)
$$

After a sufficient number of periods, the $\exp\left(-\frac{t}{RC}\right)$ term in Eq. (41) goes to 0:

$$
p(t) = \frac{T}{C} \sum_{k=0}^{N} \frac{B_k}{\imath 2\pi k + \frac{T}{RC}} \exp\left(\imath 2\pi k \frac{t}{T}\right) + p_0. \qquad (42)
$$

Here $\frac{T}{RC}$ is only a profile factor, because it is a parameter which only acts on each Fourier coefficient. We set $\frac{T}{RC}$ to 18.2 and the other parameters, $\frac{T}{C}$ and p_0, are set in such a way that the range for the pressure profile is from 80 to 120 mm Hg for normal blood pressure. Figure 1 shows the volumetric flow rate and the outflow pressure profile. On the arterial walls, we specify no-slip boundary conditions for the flow. In the structural mechanics part, as boundary condition at the ends of the arteries, we set the normal component of the displacement to zero, and for one of those nodes we also set to zero the tangential displacement component that needs to be specified to preclude rigid-body motion.

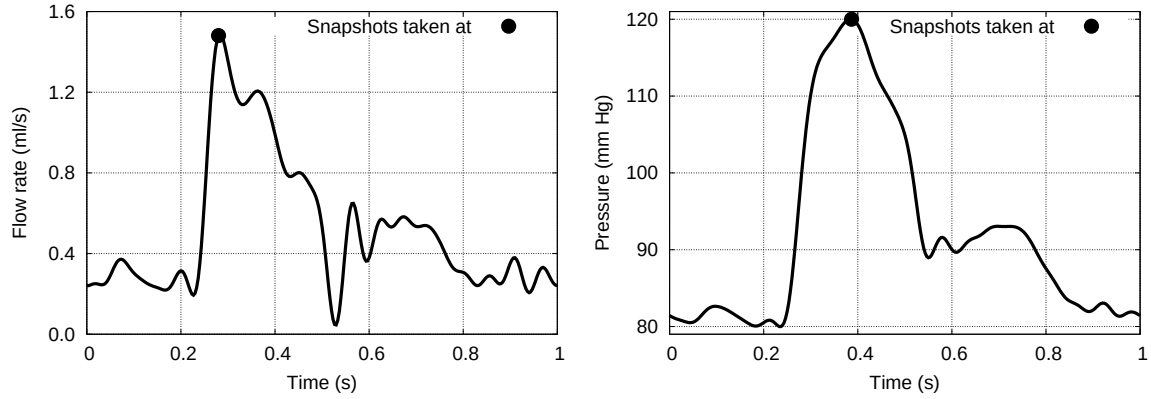


Figure 1: Volumetric flow rate and outflow pressure profile, with the maximum value marked.

Preconditioning technique. In computations with hyperelastic materials, we do not compute the diagonal of the tangent stiffness matrix. Therefore, as proposed in Tezduyar et al. (July 2008), we use a diagonal preconditioner based on the assembly of only the element-level lumped mass matrices $\mathbf{m}^e_{\text{LUMP}}$, but after being multiplied by a factor that, to some extent, takes into account the material stiffness. The expression for that multiplication factor can be found in Tezduyar et al. (July 2008). We use the "Selective Scaling" technique (see Remark 14 in Tezduyar and Sathe (2007)) to dynamically shift the emphasis between the fluid and structure parts.

6 Test Computations

The geometry of the arterial lumen is from Torii et al. (2007b, 2008, July 2008), which was extracted from the computed tomography model of a bifurcating segment of the middle cerebral artery of a 67 year-old female with aneurysm. The diameter of the arterial lumen is 2.39 mm at the inflow, and 1.53 and 1.73 mm at the two outflow ends. The Womersley parameter Υ and the period of the cardiac cycle T, described in Section 5, are 1.5 and 1.0 s, respectively. We use the "estimated zero-pressure arterial geometry", as described in Tezduyar et al. (2007a, 2008). In estimating that geometry, the time-averaged value of the blood pressure, obtained by averaging over a cardiac cycle, is 92.2 mm Hg. As the zero-pressure shape, we use a scaled down version of the geometry used in Torii et al. (2007b, 2008, July 2008). We try different wall-thickness ratios with the zero-pressure shape until we obtain, approximately, a 10% wall-thickness ratio (relative to the diameter of the arterial lumen) at the inflow. At each iteration, the trial wall-thickness ratio is globally uniform, but the base length scales for the four "patches" are defined individually, with a smooth transition between the patches. The length scales for the inflow and two outflow patches are the lumen diameters at those ends. The length scale for the aneurysm patch is $0.67\times$(lumen diameter at the inflow). Figure 2 shows, for the zero-pressure configuration, the wall thickness normalized by the wall thickness at the inflow. The structural mechanics mesh for the artery consists of 8,067 nodes and 5,316 eight-node

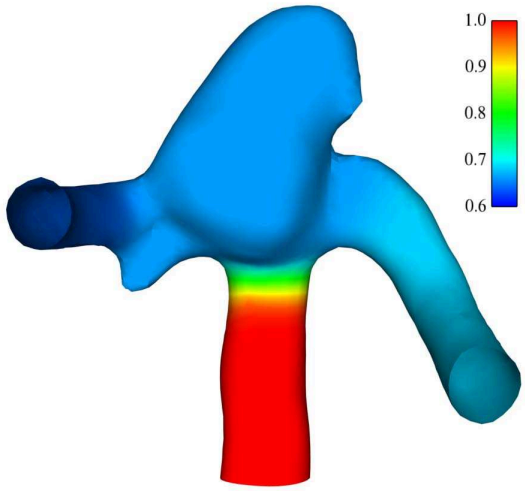


Figure 2: Zero-pressure surface configuration colored with normalized wall thickness.

hexahedral elements, with 2,689 nodes and 2,658 four-node quadrilateral elements on the fluid–structure interface. Figure 3 shows the structural mechanics mesh. We have two layers of elements across the arterial wall, which we believe to be sufficient based on our earlier numerical tests involving the inflation of a thick-walled cylinder slice. Those tests were carried out with 3D elements, under plane-strain conditions, and with material properties and length and force scales similar to those we are using here for the arterial wall. The results were accurate even with a single element across the arterial wall. In addition, we report here some recent tests we carried out with the actual arterial geometry we are focusing on in this paper.

To evaluate the performance of the structural mechanics mesh with two layers of elements across the arterial wall, we tested three additional structural mechanics meshes under a prescribed traction over a cardiac cycle. The prescribed traction comes from the "fine" mesh FSI computation described in the later parts of this section. The structural mechanics mesh properties are shown in Table 1. The only difference among the hexahedral meshes is the number of element layers across the arterial wall. The tetrahedral mesh is based on a triangular surface mesh which is the same as the fluid interface mesh in the FSI computations. All meshes have the same number of nodes at the inlet and each outlet boundary. The results for all hexahedral meshes are geometrically almost identical during the cardiac cycle, while the tetrahedral mesh results in a slightly different geometry. When we rotate and translate the deformed tetrahedral mesh, the geometries are very similar (see Figure 4). The least-squares projection of the traction from the triangular surface mesh to the quadrilateral surface mesh is the likely reason behind the small differences observed in the structure mesh deformations. The differences are insignificant when solving for the flow field within the artery. Since the results are geometrically almost identical, we provide lumen volume as a quantitative measurement of the differences. Each mesh volume, as a percentage of the volume of the hexahedral mesh with four layers, is shown in Table 1. These volume ratios remain almost constant throughout a cardiac cycle.

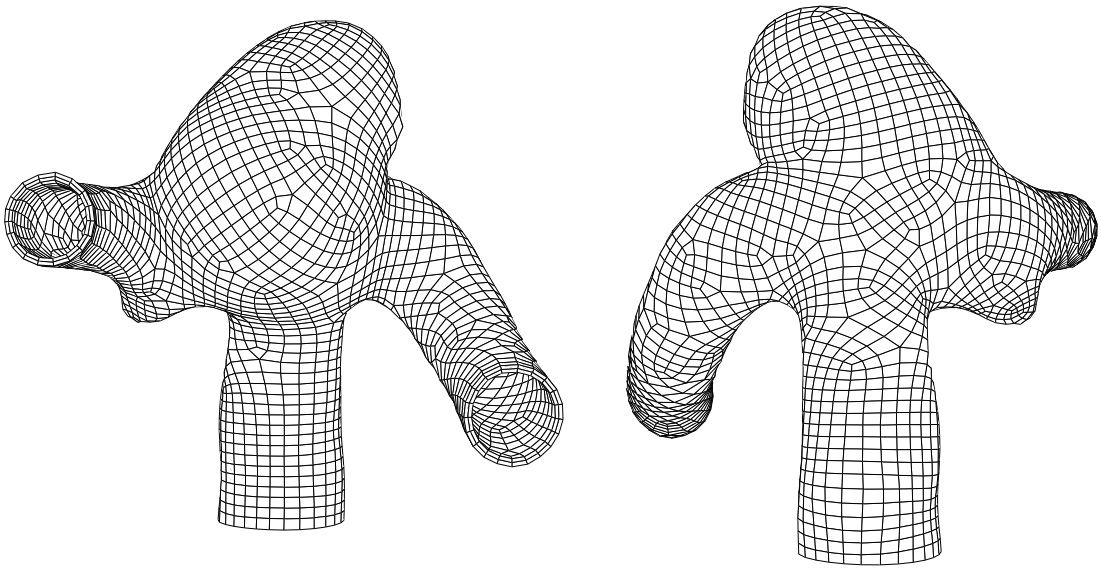


Figure 3: Structural mechanics mesh.

	Hexahedral			Tetrahedral
Number of layers	1	2	4	2
Number of nodes	5,378	8,067	13,445	9,171
Number of elements	2,658	5,316	10,632	36,312
Lumen volume	97.6%	99.4%	100%	97.1%

Table 1: Volume ratios for different structural mechanics meshes.

We use two different fluid mechanics meshes. A "coarse" mesh with 15,850 nodes and 88,573 four-node tetrahedral elements, and a "fine" mesh with 22,775 nodes and 128,813 four-node tetrahedral elements. The fine mesh has four layers of elements with higher refinement near the arterial wall. The thickness of the first layer is approximately 0.02 mm. The coarse mesh has one layer of elements with uniform thickness, and the thickness is approximately 0.2 mm. The coarse and fine meshes have the same number of nodes and elements at the fluid–structure interface: 3,057 nodes and 6,052 three-node triangular elements. Figure 5 shows the mesh at the fluid–structure interface and the inflow plane for the coarse and fine meshes.

The computations are carried out with the SSTFSI-TIP1 technique (see Remarks 4 and 7 in Tezduyar et al. (2008)) and the SUPG test function option WTSA (see Remark 1 in Tezduyar et al. (2008)). The stabilization parameters used are those given by Eqs. (12)–(18) in Tezduyar et al. (2008). The damping coefficient η is set to 1.5×10^4 s^{-1}. The time-step size is 3.333×10^{-3} s. In the CAFSI computations, the number of nonlinear iterations per time step is 6, and the number of GMRES iterations per nonlinear iteration is 300 for the fluid and structural mechanics parts, and 30 for the mesh moving part. For all six nonlinear iterations the fluid scale is set to 1.0 and the structure scale to 50. In the SCAFSI M1SC computation, we use at Step 2 and Step 3 the coarse mesh, which reduces the cost of the computations, and at Step 5 and Step 6 the fine mesh, which increases the accuracy of the flow field computed. The number of nonlinear iterations per time step is 5 for the fluid mechanics part and 4 for the structural mechanics and mesh moving parts. The number of GMRES iterations per nonlinear iteration is 150, 50 and 30 for the fluid mechanics, structural mechanics and mesh moving parts, respectively. In the SCAFSI M1C computation, Steps 1–4 are replaced with a CAFSI computation with the coarse mesh. The arterial shape obtained from the CAFSI computation is used at Step 5 and Step 6 with the fine mesh. In other words, Step 4 arterial shape comes from the reduced-cost CAFSI computation with the coarse mesh, and the fine mesh used at Step 5 and Step 6 increases the accuracy of the flow field computed. The number of nonlinear iterations per time step is 5 for the fluid mechanics part and 4 for the structural mechanics and mesh moving parts. The number of GMRES iterations per nonlinear iteration is 150, 50 and 30 for the fluid mechanics, structural mechanics and mesh moving parts, respectively.

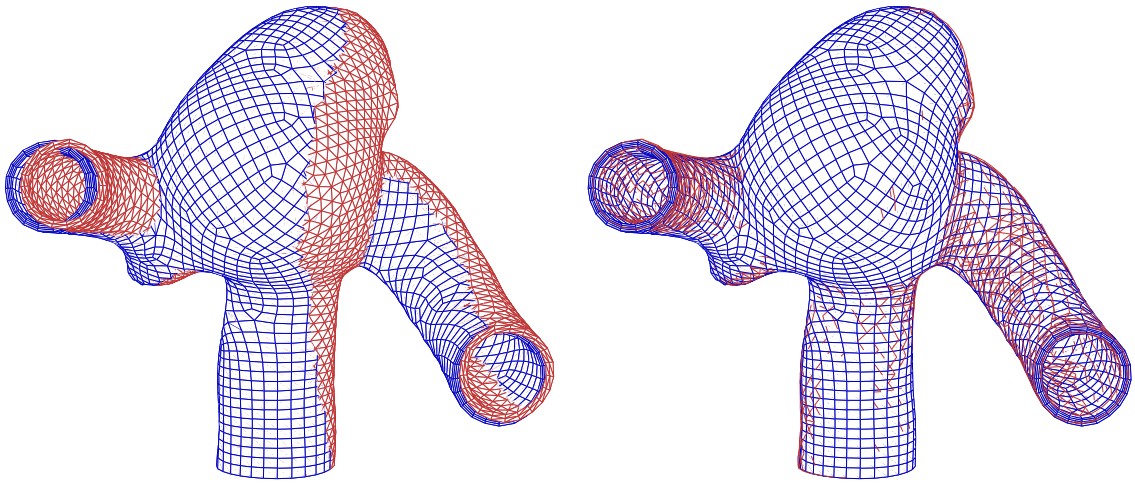


Figure 4: Structural mechanics computations with prescribed traction at the maximum outflow pressure. Blue represents the hexahedral mesh with four layers and red indicates the tetrahedral mesh with two layers. The left picture shows the original deformed geometries. The right picture shows the geometries after the tetrahedral mesh is rotated by 0.75° clockwise around an axis parallel to the inflow direction and translated to align the inlets.

Remark 4 *In our current implementation, the nonlinear iterations for the structural mechanics and mesh moving parts are not separable. Therefore the mesh motion can only be computed while computing the arterial deformation. Because of that, Step 4 arterial geometry does not actually come directly from the CAFSI computation with the coarse mesh, but it is recomputed with the interface stresses obtained from that CAFSI computation. The differences are very minor.*

Remark 5 *The fluid mechanics computation at Step 6 requires an initial flow field. For SCAFSI versions that are not spatially multiscale, this initial flow field comes directly from Step 3. For the spatially multiscale versions and extensions, we propose to use an initial flow field obtained from Step 3 by projection. In our current implementation, however, we do not have that projection capability. Instead, we carry out a very brief fluid mechanics computation to produce a divergence-free flow field. The inflow velocity for this brief computation is the velocity at the beginning of Step 6. The initial condition consists of an essentially-zero velocity field.*

We achieve good mass balance in all computations. We verify that by comparing the rate of change for the artery volume and the difference between the volumetric inflow and outflow rates. Figure 6 shows the mass balance for the CAFSI computations with the coarse and fine meshes. Figure 7 shows the mass balance for the SCAFSI M1SC and SCAFSI M1C computations. Figure 8 shows, in terms of the arterial volume, the comparisons between the CAFSI computations with the coarse and fine meshes and the SCAFSI M1SC and SCAFSI M1C computations. Figures 9 and 10 illustrate the flow field for the CAFSI computations with the coarse and fine meshes. The flow fields for the SCAFSI M1SC and SCAFSI M1C computations are essentially indistinguishable from the flow field for the CAFSI computation with the fine mesh. They are not shown here. Figures 11 and 12 illustrate the wall shear stress for the CAFSI computations with the coarse and fine meshes. Figures 13 and 14 illustrate the wall shear stress for the SCAFSI M1SC and SCAFSI M1C computations. Figure 15 shows the time-averaged wall shear stress for the CAFSI computations with the coarse and fine meshes. Figure 16 shows the time-averaged wall shear stress for the SCAFSI M1SC and SCAFSI M1C computations. Table 2 shows the maximum, mean and minimum values of the wall shear stress for the CAFSI computations with the coarse and fine meshes. Table 3 shows the

	Peak Systole		Time Average		
Mesh	Maximum	Mean	Maximum	Mean	Minimum
Coarse	127	47	39	15	0.50
Fine	227	55	58	17	0.27

Table 2: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress $\left[dyn/cm^2\right]$ for the CAFSI computations with the coarse and fine meshes. Spatial maximum and mean at the peak systole, and the spatial maximum, mean and minimum of the time-averaged values.

maximum, mean and minimum values of the wall shear stress for the CAFSI computation with the fine mesh and

the SCAFSI M1SC and SCAFSI M1C computations.

	Peak Systole		Time Average		
Computation	Maximum	Mean	Maximum	Mean	Minimum
CAFSI Fine Mesh	227	55	58	17	0.27
SCAFSI M1SC	227	55	60	17	0.29
SCAFSI M1C	225	55	60	17	0.30

Table 3: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress $\left[dyn/cm^2\right]$ for the CAFSI computation with the fine mesh and the SCAFSI M1SC and SCAFSI M1C computations. Spatial maximum and mean at the peak systole, and the spatial maximum, mean and minimum of the time-averaged values.

7 Multiscale Sequentially Coupled FSI (SCFSI) Techniques

In this section we propose to extend the multiscale sequentially-coupled FSI technique from arterial fluid mechanics to other classes of applications. The underlying concepts will still essentially be the same as those described in Section 3. The name, however, will be Multiscale Sequentially-Coupled FSI (SCFSI) Technique, which no longer implies a functionality limited to arterial fluid mechanics. Specifically, we are proposing the SCFSI M1C technique as a way of reducing the FSI computational effort where we do not need it and increasing the accuracy of the fluid mechanics computations where we need accurate, detailed flow computation. We propose to first compute the structural deformation with the (fully) coupled FSI (CFSI) technique and a relatively coarser fluid mechanics mesh, followed by mesh motion and fluid mechanics computations with a more refined mesh. We also propose a time-integration version of this, where we first compute the structural deformation with the CFSI technique and a time-step size as small as it is need in that computation, followed by mesh motion and fluid mechanics computations with a smaller time-step size that might be needed for more accurate, detailed flow computation.

To illustrate how the spatially multiscale SCFSI M1C technique works, we carry out FSI computations for 2D flow past a flexible beam. The problem set up is shown in Figure 17. The length and thickness of the beam are 2.0 m and 10.0 cm. Its density, modulus of elasticity and Poisson's ratio are 1.1358×10^4 kg/m^3, 1.2×10^7 N/m^2 and 0.3. At the midpoint cross-section of the beam all displacements are set to zero. The fluid density and kinematic viscosity are 1,000 kg/m^3 and 1.0×10^{-2} m^2/s. The flow boundary conditions in the 2D plane are shown in Figure 17. The inflow velocity is 1.0 m/s. Because we do the computations with 3D FSI solvers, at the lateral computational boundaries perpendicular to the third direction we use slip conditions. The structure is modeled with the geometrically nonlinear continuum element made of linearly elastic material (see Section 2.2.1 in Tezduyar et al. (2007b)).

The structural mechanics mesh consists of 410 nodes and 160 eight-node hexahedral elements, with 40 elements along the beam, 4 elements across the thickness, and one element in the third direction. We use two different fluid mechanics meshes. A "coarse" mesh with 8,476 nodes and 24,882 four-node tetrahedral elements, and a "fine" mesh with 25,536 nodes and 75,936 four-node tetrahedral elements. The fine mesh has 8 layers of elements with higher refinement near the beam, starting with the first-layer thickness of 3 cm, and increasing with a factor of 1.1 from one layer to the next. The coarse mesh has 4 layers of elements with higher refinement near the beam, and all four layers have the same thickness of 10 cm. At the fluid–structure interface, the fine mesh nodes match the structure nodes fully along the beam, and with intervals of two structure elements at the top and bottom of the beam. The coarse mesh nodes match the structure nodes only along the beam, with intervals of two structure elements.

The computations are carried out using SSTFSI-TIP1 technique (see Remarks 4 and 7 in Tezduyar et al. (2008)), with the SUPG test function option WTSA (see Remark 1 in Tezduyar et al. (2008)). The stabilization parameters used are those given in Tezduyar et al. (2008) by Eqs. (14)–(18) and (19)–(20), with the τ_{SUGN2} term dropped from Eq. (19). The time-step size is 5.0×10^{-2} s. In the CFSI computation with the coarse mesh, the number of nonlinear iterations per time step is 4, and the number of GMRES iterations per nonlinear iteration is 120 for the fluid and structural mechanics parts, and 60 for the mesh moving part. For the first nonlinear iteration the structure scale is set to 1.0×10^{-8}, and back to 1.0 for the remaining iterations. In the CFSI computation with the fine mesh, the number of nonlinear iterations per time step is 4, and the number of GMRES iterations per nonlinear iteration is 240 for the fluid and structural mechanics parts, and 60 for the mesh moving part. For the first nonlinear iteration the structure scale is set to 1.0×10^{-8}, and back to 1.0 for the remaining iterations. In the SCFSI M1C computation, the number of nonlinear iterations per time step is 3 for the fluid mechanics part and 1 for the mesh moving part. The number

of GMRES iterations per nonlinear iteration is 80 for the fluid mechanics part and 60 for the mesh moving part. In the CFSI computation with the fine mesh and SCFSI M1C computation, the starting (velocity and pressure) values for the fluid mechanics part come by a least-squares projection from the values obtained in the CFSI computation with the coarse mesh, followed by a one time-step computation with a very small time-step size (5.0×10^{-6} s) just to recover the divergence-free condition, followed by "pressure clipping" (see Tezduyar and Sathe (2007) for a brief description of "pressure clipping"). Figure 18 shows, in terms of the (upper-) tip displacement of the beam, the comparison between the CFSI computations with the coarse and fine meshes. Figure 19 shows the vorticity field for the CFSI computation with the coarse mesh. Figure 20 shows the vorticity field for the CFSI computation with the fine mesh and SCFSI M1C computation.

8 Concluding Remarks

We presented the multiscale versions and extensions of the Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique, with emphasis on the spatially multiscale versions and extensions. The original version of the SCAFSI technique was introduced as an approximate FSI approach in arterial fluid mechanics and is based on the assumption that the arterial deformation during a cardiac cycle is driven mostly by the blood pressure. In that original version, first we compute a "reference" arterial deformation as a function of time, driven only by the blood pressure profile of the cardiac cycle. Then we compute a sequence of updates involving mesh motion, fluid dynamics calculations, and recomputing the arterial deformation. Although they can also be used in conjunction with other moving-mesh FSI modeling approaches, the SCAFSI technique and its multiscale versions and extensions were developed and tested in conjunction with the stabilized space–time FSI (SSTFSI) technique. The SSTFSI technique is based on the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation and is supplemented with a number of special techniques developed for arterial FSI. These special techniques include using an estimated zero-pressure arterial geometry, a special mapping technique for specifying the velocity profile at an inflow boundary with non-circular shape, using layers of refined fluid mechanics mesh near the arterial walls, and a recipe for pre-FSI computations that improve the convergence of the FSI computations. The SCAFSI technique provides a computationally more economical alternative to the fully coupled FSI approach in arterial fluid mechanics. It also brings additional flexibility, such as being able to carry out the computations in a spatially or temporally multiscale fashion. In the temporally multiscale version, we can use different time step sizes for the structural and fluid mechanics parts. In the spatially multiscale versions and extensions, fluid mechanics meshes with different refinement levels are used at different stages of the FSI computation. A relatively coarser mesh is used at the early stages, and a more refined mesh is used at the stage where we do the high-accuracy fluid mechanics computations, such as the wall shear stress computation. We presented the test computations we have recently carried out with the spatially multiscale versions and extensions of the SCAFSI technique. We used actual patient-specific image-based data, where the arterial geometry is based on computed tomography images. Specifically, we focused on the bifurcating middle cerebral artery segment of a 67 year-old female with aneurysm. We modeled the arterial wall with the continuum element made of hyperelastic (Fung) material. Our test computations show that the spatially multiscale SCAFSI technique is a very effective way of reducing the FSI computational effort where it is not needed and increasing the accuracy of the fluid mechanics computations where it is needed. We also extended the multiscale sequentially-coupled FSI technique from arterial fluid mechanics to other classes of applications, calling it Multiscale Sequentially-Coupled FSI (SCFSI) Technique, so that the name no longer implies a functionality limited to arterial fluid mechanics. In the context of FSI computations for 2D flow past a flexible beam, we demonstrated how the spatially multiscale SCFSI technique works.

ACKNOWLEDGMENT

This work was supported in part by a Seed Grant from the Gulf Coast Center for Computational Cancer Research funded by John & Ann Doerr Fund for Computational Biomedicine. It was also supported in part by the Rice Computational Research Cluster funded by NSF under Grant CNS-0421109, and a partnership between Rice University, AMD and Cray. We are grateful to Dr. Ryo Torii (Imperial College) and Professor Marie Oshima (University of Tokyo) for providing the arterial geometry and inflow velocity data used in the computations.

References

Bazilevs, Y.; Calo, V.; Hughes, T.; Zhang, Y.: Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43, (2008), 3–37.

Bazilevs, Y.; Calo, V.; Zhang, Y.; Hughes, T.: Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38, (2006), 310–322.

Bletzinger, K.-U.; Wuchner, R.; Kupzok, A.: Algorithmic treatment of shells and free form-membranes in FSI. In: H.-J. Bungartz; M. Schafer, eds., *Fluid–Structure Interaction*, vol. 53 of *Lecture Notes in Computational Science and Engineering*, pages 336–355, Springer (2006).

Brooks, A.; Hughes, T.: Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32, (1982), 199–259.

Dettmer, W.; Peric, D.: A computational framework for fluid-structure interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006), 5754–5779.

Dettmer, W.; Peric, D.: On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Computational Mechanics*, 43, (2008), 81–90.

Gerbeau, J.-F.; Vidrascu, M.; Frey, P.: Fluid–structure interaction in blood flow on geometries based on medical images. *Computers and Structures*, 83, (2005), 155–165.

Huang, H.; Virmani, R.; Younis, H.; Burke, A.; Kamm, R.; Lee, R.: The impact of calcification on the biomechanical stability of atherosclerotic plaques. *Circulation*, 103, (2001), 1051–1056.

Hughes, T.; Brooks, A.: A multi-dimensional upwind scheme with no crosswind diffusion. In: T. Hughes, ed., *Finite Element Methods for Convection Dominated Flows*, AMD-Vol.34, pages 19–35, ASME, New York (1979).

Hughes, T.; Franca, L.; Balestra, M.: A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59, (1986), 85–99.

Johnson, A.; Tezduyar, T.: Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119, (1994), 73–94.

Johnson, A.; Tezduyar, T.: Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics*, 23, (1999), 130–143.

Kalro, V.; Tezduyar, T.: A parallel 3D computational method for fluid–structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190, (2000), 321–332.

Khurram, R.; Masud, A.: A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction. *Computational Mechanics*, 38, (2006), 403–416.

Kuttler, U.; Forster, C.; Wall, W.: A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics*, 38, (2006), 417–429.

Kuttler, U.; Wall, W.: Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43, (2008), 61–72.

Lohner, R.; Cebral, J.; Yang, C.; Baum, J.; E.L.Mestreau; Soto, O.: Extending the range of applicability of the loose coupling approach for FSI simulations. In: H.-J. Bungartz; M. Schafer, eds., *Fluid–Structure Interaction*, vol. 53 of *Lecture Notes in Computational Science and Engineering*, pages 82–100, Springer (2006).

Masud, A.; Bhanabhagvanwala, M.; Khurram, R.: An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction. *Computers & Fluids*, 36, (2007), 77–91.

Michler, C.; van Brummelen, E.; de Borst, R.: An interface Newton–Krylov solver for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 47, (2005), 1189–1195.

Mittal, S.; Tezduyar, T.: Massively parallel finite element computation of incompressible flows involving fluid-body interactions. *Computer Methods in Applied Mechanics and Engineering*, 112, (1994), 253–282.

Mittal, S.; Tezduyar, T.: Parallel finite element simulation of 3D incompressible flows – Fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, 21, (1995), 933–953.

Ohayon, R.: Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems. *Computer Methods in Applied Mechanics and Engineering*, 190, (2001), 3009–3019.

Otto, F.: Die grundform des arteriellen pulses. *Zeitung fur Biologie*, 37, (1899), 483–586.

Saad, Y.; Schultz, M.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7, (1986), 856–869.

Sawada, T.; Hisada, T.: Fuid–structure interaction analysis of the two dimensional flag-in-wind problem by an interface tracking ALE finite element method. *Computers & Fluids*, 36, (2007), 136–146.

Stein, K.; Benney, R.; Kalro, V.; Tezduyar, T.; Leonard, J.; Accorsi, M.: Parachute fluid–structure interactions: 3-D Computation. *Computer Methods in Applied Mechanics and Engineering*, 190, (2000), 373–386.

Tezduyar, T.: Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28, (1992), 1–44.

Tezduyar, T.: Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8, (2001), 83–130.

Tezduyar, T.: Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43, (2003), 555–575.

Tezduyar, T.: Finite element methods for fluid dynamics with moving boundaries and interfaces. In: E. Stein; R. D. Borst; T. Hughes, eds., *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chap. 17, John Wiley & Sons (2004).

Tezduyar, T.; Aliabadi, S.; Behr, M.; Johnson, A.; Mittal, S.: Parallel finite-element computation of 3D flows. *Computer*, 26, 10, (1993), 27–36.

Tezduyar, T.; Aliabadi, S.; Behr, M.; Mittal, S.: Massively parallel finite element simulation of compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 119, (1994), 157–177.

Tezduyar, T.; Behr, M.; Liou, J.: A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94, 3, (1992a), 339–351.

Tezduyar, T.; Behr, M.; Mittal, S.; Johnson, A.: Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations. In: *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, pages 7–24, ASME, New York (1992b).

Tezduyar, T.; Behr, M.; Mittal, S.; Liou, J.: A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94, 3, (1992c), 353–371.

Tezduyar, T.; Cragin, T.; Sathe, S.; Nanna, B.: FSI computations in arterial fluid mechanics with estimated zero-pressure arterial geometry. In: E. Onate; J. Garcia; P. Bergan; T. Kvamsdal, eds., *Marine 2007*, CIMNE, Barcelona, Spain (2007a).

Tezduyar, T.; Mittal, S.; Ray, S.; Shih, R.: Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95, (1992d), 221–242.

Tezduyar, T.; Osawa, Y.: Fluid–structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191, (2001), 717–726.

Tezduyar, T.; Sathe, S.: Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54, (2007), 855–900.

Tezduyar, T.; Sathe, S.; Cragin, T.; Nanna, B.; Conklin, B.; Pausewang, J.; Schwaab, M.: Modeling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics. *International Journal for Numerical Methods in Fluids*, 54, (2007b), 901–922.

Tezduyar, T.; Sathe, S.; Keedy, R.; Stein, K.: Space–time techniques for finite element computation of flows with moving boundaries and interfaces. In: S. Gallegos; I. Herrera; S. Botello; F. Zarate; G. Ayala, eds., *Proceedings of the III International Congress on Numerical Methods in Engineering and Applied Science*, CD-ROM, Monterrey, Mexico (2004).

Tezduyar, T.; Sathe, S.; Keedy, R.; Stein, K.: Space–time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006a), 2002–2027.

Tezduyar, T.; Sathe, S.; Schwaab, M.; Conklin, B.: Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique. *International Journal for Numerical Methods in Fluids*, 57, (2008), 601–629.

Tezduyar, T.; Sathe, S.; Stein, K.: Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006b), 5743–5753.

Tezduyar, T.; Sathe, S.; Stein, K.; Aureli, L.: Modeling of fluid–structure interactions with the space–time techniques. In: H.-J. Bungartz; M. Schafer, eds., *Fluid–Structure Interaction*, vol. 53 of *Lecture Notes in Computational Science and Engineering*, pages 50–81, Springer (2006c).

Tezduyar, T.; Schwaab, M.; Sathe, S.: Arterial fluid mechanics with the sequentially-coupled arterial FSI technique. In: E. Onate; M. Papadrakakis; B. Schrefler, eds., *Coupled Problems 2007*, CIMNE, Barcelona, Spain (2007c).

Tezduyar, T.; Schwaab, M.; Sathe, S.: Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique (July 2008), *Computer Methods in Applied Mechanics and Engineering*, published online, DOI: 10.1016/j.cma.2008.05.024.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Influence of wall elasticity on image-based blood flow simulation. *Japan Society of Mechanical Engineers Journal Series A*, 70, (2004), 1224–1231, in Japanese.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Computer modeling of cardiovascular fluid–structure interactions with the Deforming-Spatial-Domain/Stabilized Space–Time formulation. *Computer Methods in Applied Mechanics and Engineering*, 195, (2006a), 1885–1895.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Fluid–structure interaction modeling of aneurysmal conditions with high and normal blood pressures. *Computational Mechanics*, 38, (2006b), 482–490.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Influence of wall elasticity in patient-specific hemodynamic simulations. *Computers & Fluids*, 36, (2007a), 160–168.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Numerical investigation of the effect of hypertensive blood pressure on cerebral aneurysm — Dependence of the effect on the aneurysm shape. *International Journal for Numerical Methods in Fluids*, 54, (2007b), 995–1009.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Fluid–structure interaction modeling of a patient-specific cerebral aneurysm: Influence of structural modeling. *Computational Mechanics*, 43, (2008), 151–159.

Torii, R.; Oshima, M.; Kobayashi, T.; Takagi, K.; Tezduyar, T.: Fluid–structure interaction modeling of blood flow and cerebral aneurysm: Significance of artery and aneurysm shapes (July 2008), *Computer Methods in Applied Mechanics and Engineering*, published online, DOI: 10.1016/j.cma.2008.08.020.

van Brummelen, E.; de Borst, R.: On the nonnormality of subiteration for a fluid-structure interaction problem. *SIAM Journal on Scientific Computing*, 27, (2005), 599–621.

Wall, W.; Genkinger, S.; Ramm, E.: A strong coupling partitioned approach for fluid–structure interaction with free surfaces. *Computers & Fluids*, 36, (2007), 169–183.

Womersley, J.: Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *Journal of Physiology*, 127, (1955), 553–563.

Address: T.E. Tezduyar (corresponding author), K. Takizawa and J. Christopher, Mechanical Engineering, Rice University – MS 321, 6100 Main Street, Houston, Texas 77005, USA
email: `tezduyar@rice.edu`

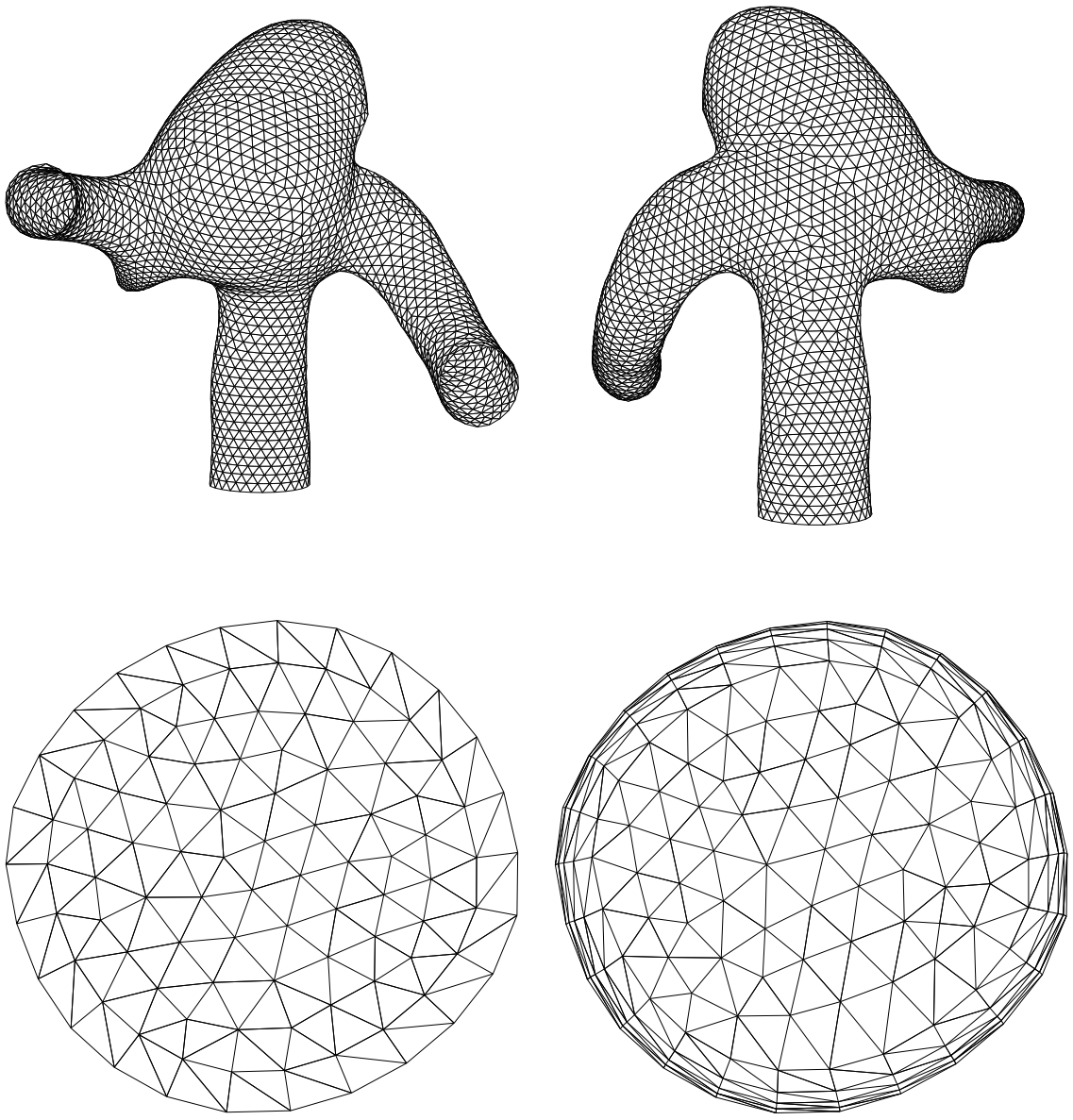


Figure 5: A bifurcating middle cerebral artery segment with aneurysm. Fluid mechanics mesh at the fluid–structure interface and the inflow plane for the coarse and fine meshes.

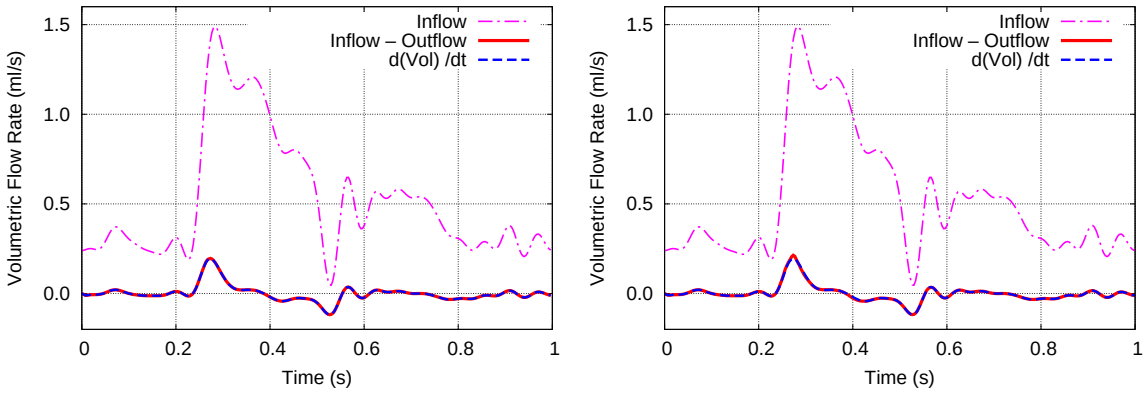


Figure 6: A bifurcating middle cerebral artery segment with aneurysm. Verification of mass balance for the CAFSI computation with the coarse (left) and fine (right) meshes. Volumetric inflow rate, difference between the volumetric inflow and outflow rates, and rate of change for the artery volume.

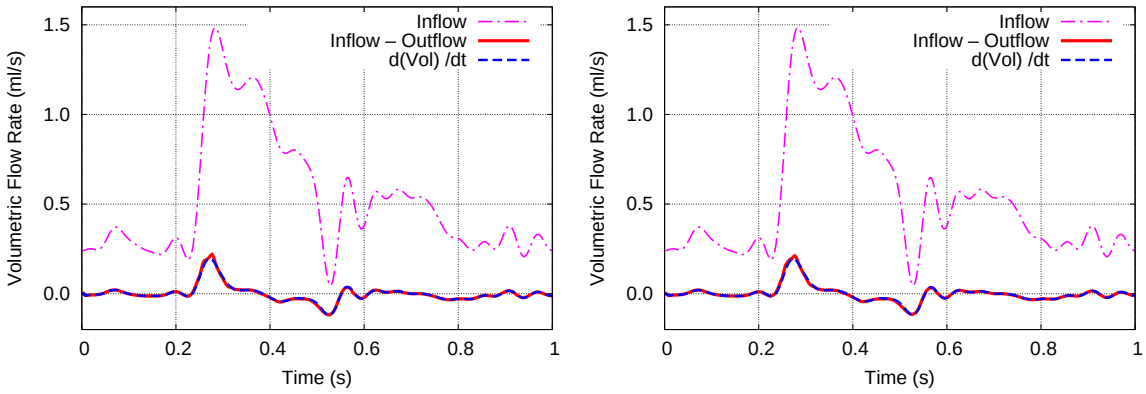


Figure 7: A bifurcating middle cerebral artery segment with aneurysm. Verification of mass balance for the SCAFSI M1SC (left) and SCAFSI M1C (right) computations. Volumetric inflow rate, difference between the volumetric inflow and outflow rates, and rate of change for the artery volume.

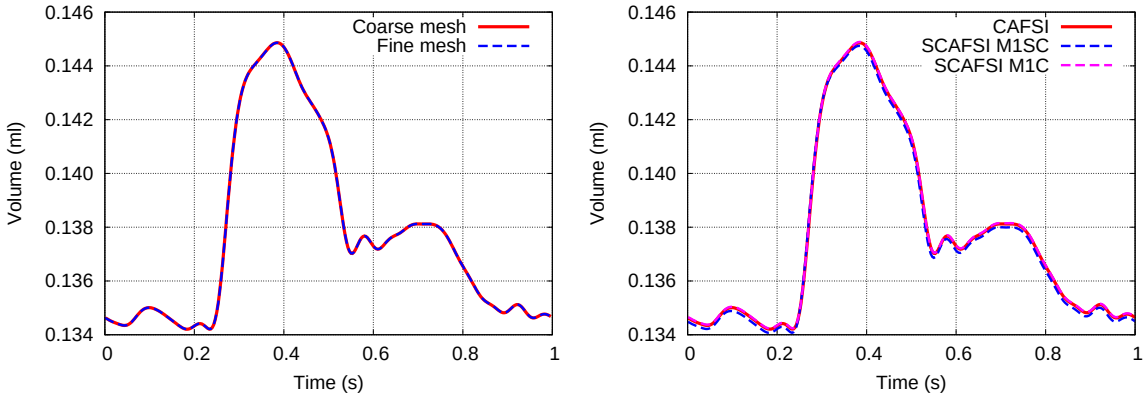


Figure 8: A bifurcating middle cerebral artery segment with aneurysm. Arterial volume. Left: comparison between the CAFSI computations with the coarse and fine meshes. Right: Comparison between the CAFSI computation with the fine mesh and the SCAFSI M1SC and SCAFSI M1C computations.

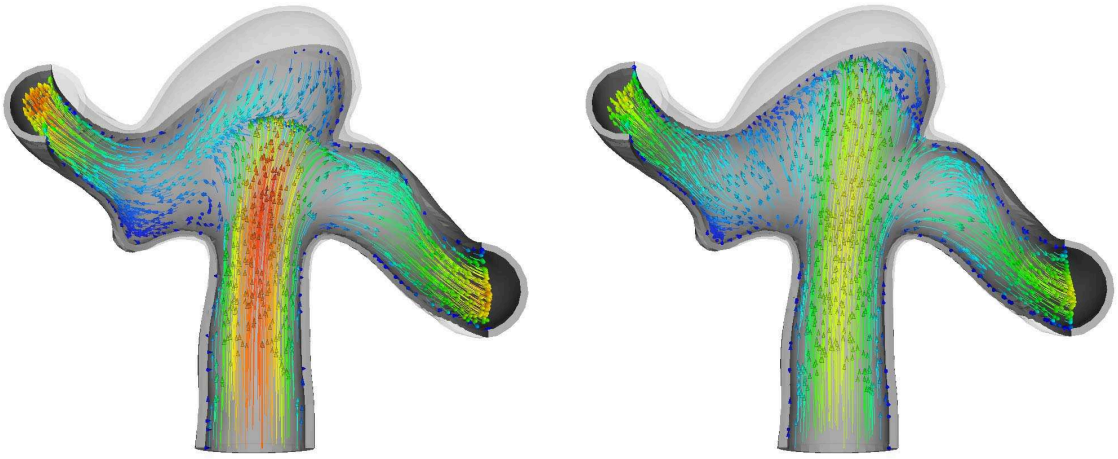


Figure 9: A bifurcating middle cerebral artery segment with aneurysm. Flow field for the CAFSI computation with the coarse mesh when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right). Velocity vectors colored by magnitude.

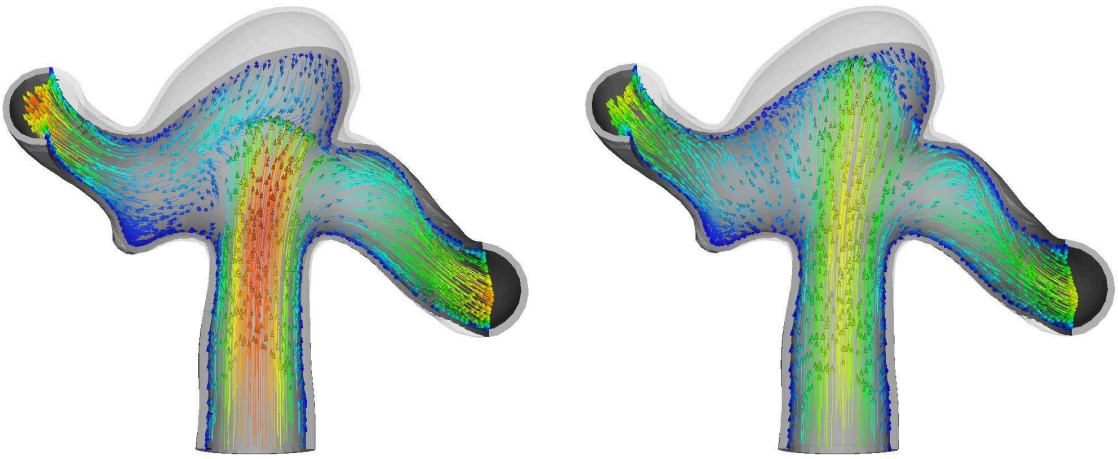


Figure 10: A bifurcating middle cerebral artery segment with aneurysm. Flow field for the CAFSI computation with the fine mesh when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right). Velocity vectors colored by magnitude.

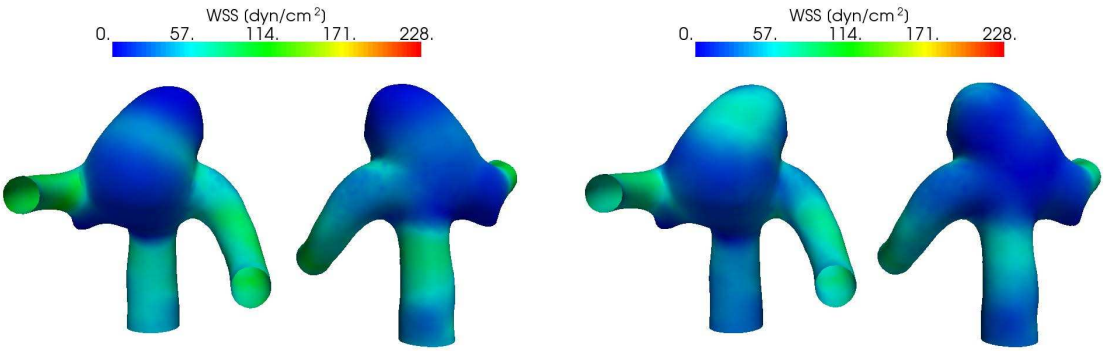


Figure 11: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress for the CAFSI computation with the coarse mesh when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right).

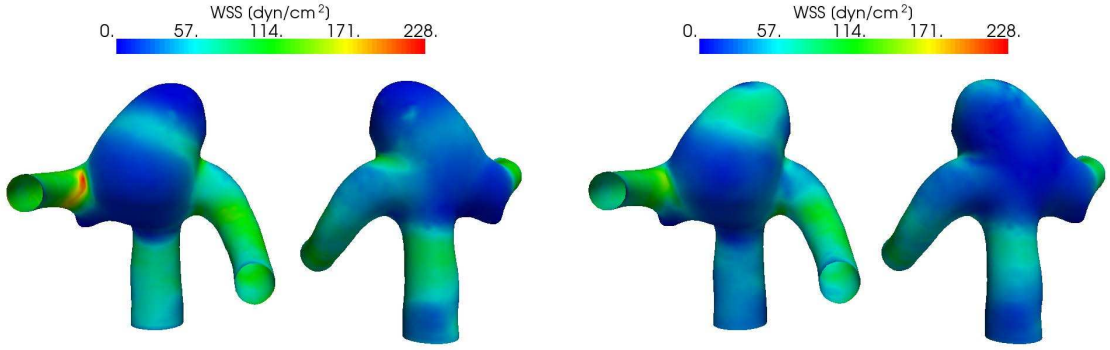


Figure 12: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress for the CAFSI computation with the fine mesh when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right).

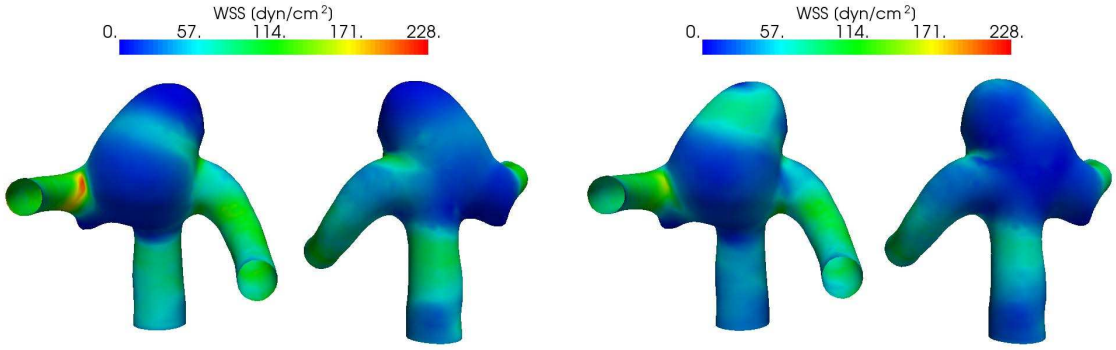


Figure 13: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress for the SCAFSI M1SC computation when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right).

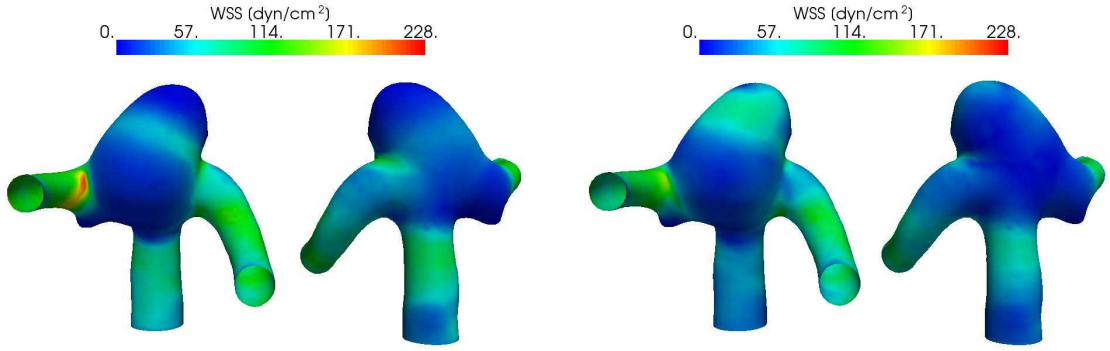


Figure 14: A bifurcating middle cerebral artery segment with aneurysm. Wall shear stress for the SCAFSI M1C computation when the volumetric flow rate is maximum (left) and when the outflow pressure is maximum (right).

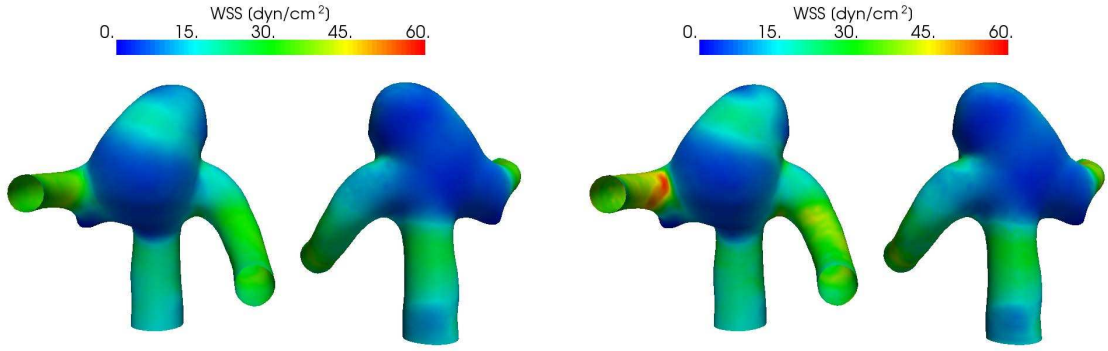


Figure 15: A bifurcating middle cerebral artery segment with aneurysm. Time-averaged wall shear stress for the CAFSI computations with the coarse (left) and fine (right) meshes.

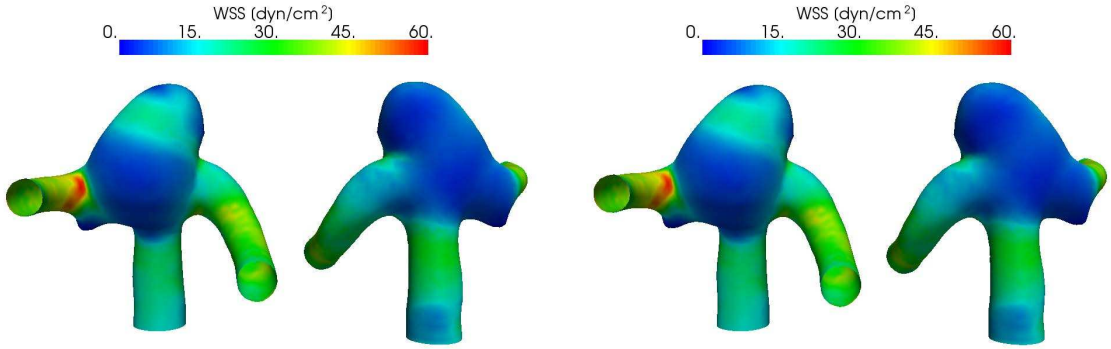


Figure 16: A bifurcating middle cerebral artery segment with aneurysm. Time-averaged wall shear stress for the SCAFSI M1SC (left) and SCAFSI M1C (right) computations.

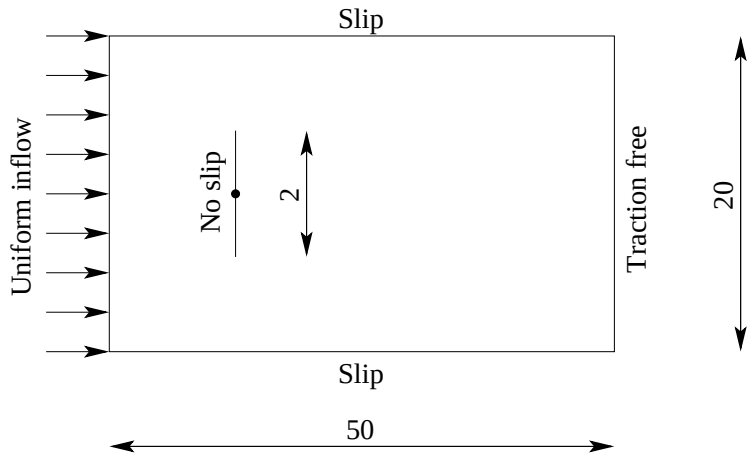


Figure 17: 2D flow past a flexible beam. Problem set up. The dimensions indicated are in meters. The length and thickness of the beam are 2.0 m and 10.0 cm. At the midpoint cross-section of the beam all displacements are set to zero. The flow boundary conditions in the 2D plane are indicated next to each boundary. The inflow velocity is 1.0 m/s.

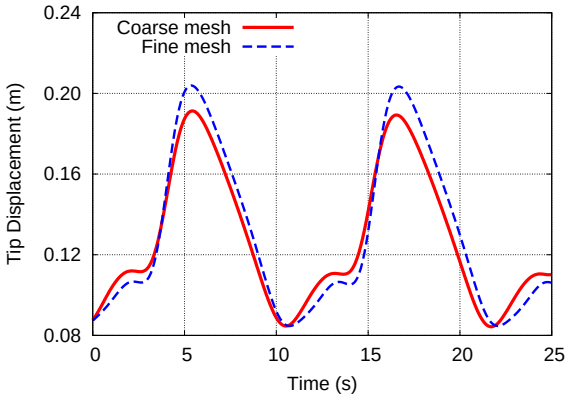


Figure 18: 2D flow past a flexible beam. (Upper-) Tip displacement. Comparison between the CFSI computations with the coarse and fine meshes.

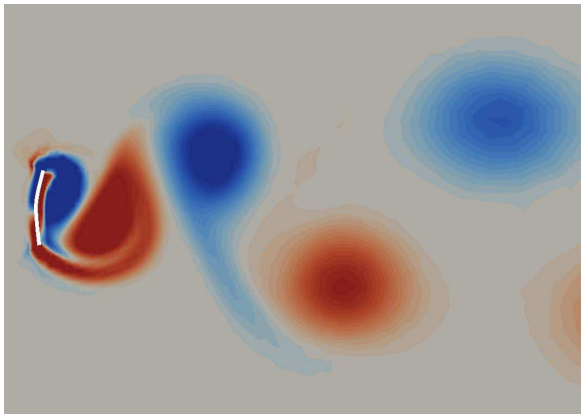


Figure 19: 2D flow past a flexible beam. Vorticity field for the CFSI computation with the coarse mesh, at the instant corresponding to t = 5 s in Figure 18.

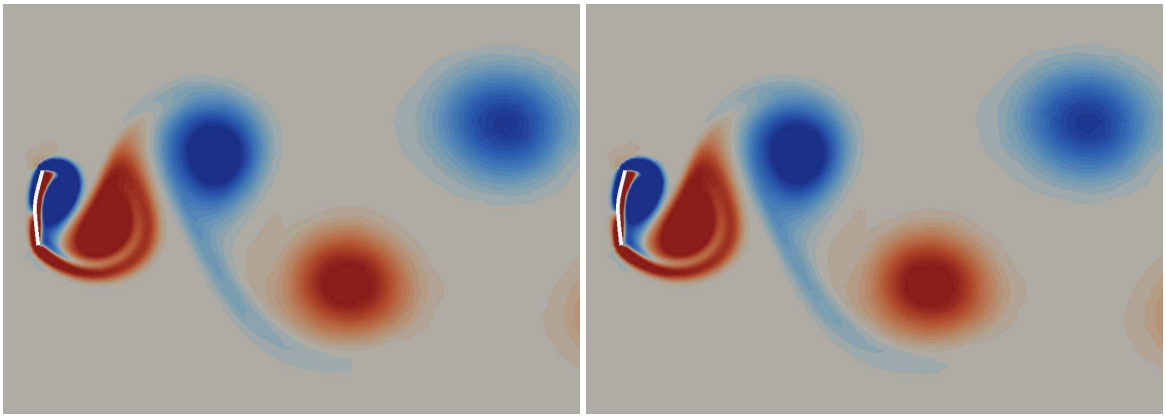


Figure 20: 2D flow past a flexible beam. Vorticity field for the CFSI computation with the fine mesh (left) and SCFSI M1C computation (right), at the instant corresponding to t = 5 s in Figure 18.

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 253– 264
Herrsching am Ammersee, 29.9.-1.10.2008

Fluid-Structure-Acoustic Interaction of a Thin, Flexible Plate in the Wake of a Wall-Mounted Square Cylinder

T. Uffinger, F. Schäfer, S. Becker, J. Grabinger, M. Kaltenbacher

A coupled simulation approach for fluid-structure-acoustic interactions is presented, which considers both the flow-induced sound and the sound due to structural vibrations. The proposed computation scheme is based on a partitioned approach that uses different simulation codes for the flow simulation and the structural and acoustic computations. The flow simulation is done by an in-house finite-volume code solving the incompressible Navier-Stokes equations in space and time. For the structural mechanics computations, an in-house finite-element multi-physics code is applied, which is also able to solve the acoustic wave equation describing the wave propagation to the far field. An implicit coupling scheme with sub-iterations in each time step is employed for fluid-structure interaction. The computation of the aeroacoustic sound is based on Lighthill's theory. The structural-acoustic coupling is realized by applying appropriate boundary conditions to the acoustic wave equation.

The above-described methodology is applied to a simplified model of a car underbody. The acoustic field due to the interaction of a thin, flexible plate representing coverings of the car underbody and a turbulent flow is investigated. The flexible plate is part of an otherwise rigid wall that is overflown by air. To model obstacles typically found at real car underbodies, a square cylinder is placed upstream of the flexible plate. The Reynolds number of the flow based on the free stream veloctiy and the obstacle height is $26\,000$.

Nomenclature

δV	Swept volume	Φ	Transport variable
ε	Convergence criterion	c	Speed of sound
η	Local coordinate	e	Finite element
φ	Acoustic velocity potential	i	Grid node/center
ρ	Fluid density	l	FSI iteration counter
ρ_0	Ambient density	n	Iteration counter
ρ_s	Density of mechanical structure	$\mathbf{n}$	Normal vector
τ_{ij}	Shear stress tensor	p	Pressure
τ_w	Wall shear stress	p_0	Ambient pressure
ξ	Local coordinate	p'	Acoustic pressure fluctuations
Δw	Grid displacement	$\mathbf{t}$	Tangential vector
Γ_Φ	Diffusion coefficient	u_i	Velocity
$u_{g,i}$	Grid velocity	T_{ij}	Lighthill tensor
$\mathbf{w}$	Structural displacement	U_∞	Free-stream velocity
y^+	Dimensionless wall scale	V	Volume
D	Edge length of square cylinder	CFD	Computational fluid dynamics
E	Modulus of elasticity	CV	Control volume
K_r	Grid node r	FSI	Fluid-structure interaction
$\mathbf{L}$	Load vector	LES	Large-eddy simulation
N	Interpolation function	RANS	Reynolds averaged Navier-Stokes
Q_Φ	Source/sink	Re	Reynolds number
S_i	Surface i of a CV		

1 Introduction

In many technical applications, the interaction of a fluid flow with a flexible structure leads to structural vibrations and thereby to the generation of vibrational sound. Additionally, aeroacoustic sound is generated by turbulent fluctuations within the fluid. In such flows, sound is a result of the complex interaction between fluid mechanics, structural mechanics and acoustics. As the generated sound is often considered as noise, its reduction is of major interest. For this purpose, a deeper understanding of the physical phenomena taking place in flow configurations with fluid-structure-acoustic interactions is needed.

Vibrating structures, and in particular vibrating plates, are of special interest for applications in aviation and automotive engineering, hence the topic has received much attention in the literature. One of the early investigations in this direction was the work of Davies (1971), where the excitation of a thin flexible panel by wall-pressure fluctuations of a turbulent boundary layer was studied using modal analysis. The boundary layer excitation of flexible plates and the resulting emission of noise have also been investigated by, e.g., Graham (1996), Howe and Shah (1996), Frampton and Clark (1997), and Mazzoni and Kristiansen (1999). In most of the studies, the influence of the fluid on the flexible plate is modeled based on spectra of the turbulent wall-pressure fluctuations in the boundary layer. The flow above the plate is not resolved in detail. A more explicit treatment of the flow over the plate was performed by Zolotarev (1997) and Tang et al. (2005), where the fluid flow was modeled using potential theory. Tang et al. (2005) estimated the sound resulting from the interaction between a single model vortex and a flexible wall in order to investigate the basic mechanisms of the fluid-structure-acoustic interaction. With respect to the treatment of the fluid flow, studies considering a highly resolved flow based on the Navier-Stokes equations are rare. There has also been very little work towards a fully coupled treatment of fluid flow, structural mechanics and acoustics at a realistic level of detail [see, e.g., Visbal and Gordnier (2004) and Vergne et al. (2007)]. However, such a methodology would be very helpful for improving the understanding of the mechanisms leading to the emission of sound and for taking measures towards a reduction in flow-induced and vibrational noise.

In this paper, a numerical simulation approach for the coupled treatment of fluid-structure-acoustic interactions is presented. The method considers acoustic pressure fluctuations which are low compared with the overall fluid pressure, so that the acoustic feedback on the fluid and on the structure can be neglected. This assumption is valid for many technical applications. By this simplification, a reduction in complexity can be achieved, resulting in three subproblems which can be treated and computed separately: fluid-structure interaction, fluid-acoustic coupling and structure-acoustic coupling. The proposed software architecture is based on a partitioned approach, i.e., different codes are used for the computation of fluid dynamics, structural mechanics and acoustics. For the flow simulation, an in-house finite-volume code is applied that solves the incompressible Navier-Stokes equations in space and time. For structural mechanics, an in-house finite-element code is used. The coupling between fluid and structure is realized by a code coupling interface. The computation of flow-induced sound relies on Lighthill's analogy and a finite-element discretization of the acoustic wave equation describing the wave propagation to the far field. Vibrational sound is computed from the structural movement by applying appropriate boundary conditions to the wave equation. Details of the simulation approach are given in Section 2.

Our computational methodology is applied to a simplified model of a car underbody. Basically, the flow over a flat plate is considered, where most of the plate is rigid, but part of it is made of a thin, flexible blank representing coverings, which are often installed to improve the aerodynamic properties of cars. Additionally, a square cylinder is placed upstream of the flexible structure to model obstacles typically found with real car underbodies.

2 Numerical Method

The applied numerical computation scheme is based on a partitioned simulation approach. Two different codes are used for the computation of fluid-structure-acoustic interactions. For the fluid simulation, the finite-volume CFD solver FASTEST-3D [see Durst and Schäfer (1996)] is applied, which has been developed at the Institute of Fluid Mechanics, University of Erlangen-Nuremberg. The structural mechanics problem is solved with the multiphysics finite-element code CFS++ [see Kaltenbacher et al. (2007)], developed at the Department of Sensor Technology, University of Erlangen-Nuremberg. The wave equation of the acoustic problem is also solved with CFS++. For the coupling between FASTEST-3D and CFS++, realizing the fluid-structure interaction, the commercial code coupling interface MpCCI [see Ahrem et al. (2003)] is used. A schematic of the software architecture is shown in Figure 1.

The CFD code FASTEST-3D is capable of computing incompressible flow fields with heat and mass transfer of

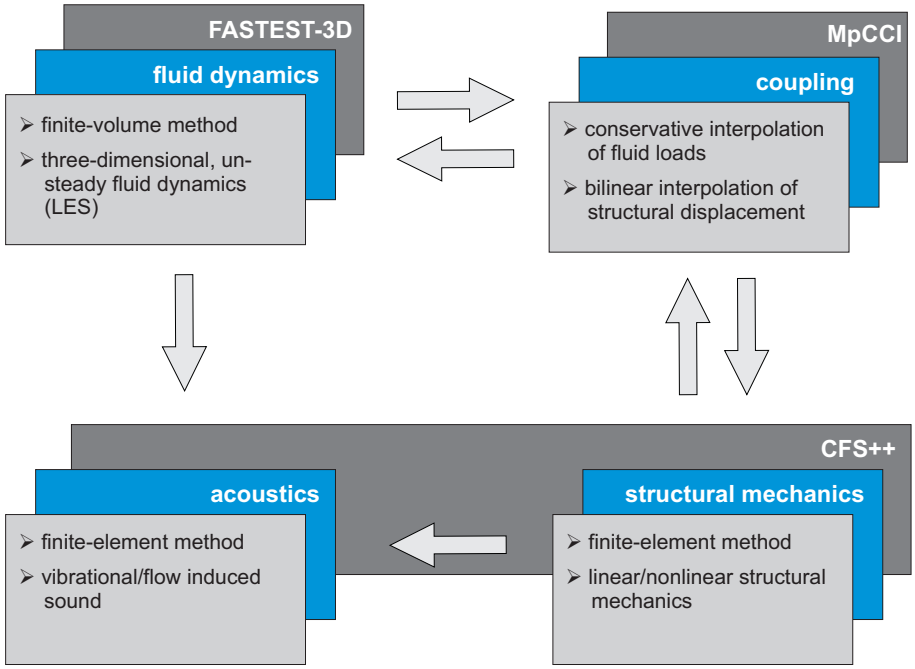


Figure 1: Coupled computation scheme for the simulation of fluid-structure-acoustic interactions

Newtonian fluids. It is based on the finite-volume method. The discretization of the Navier-Stokes equations is of second-order accuracy in space and time. Both Reynolds averaged Navier-Stokes (RANS) and large-eddy simulations (LES) can be carried out with the software. For problems with fluid-structure-acoustic interactions, usually the unsteady and three-dimensional flow field is needed. Thus, LES is the turbulence model mainly used for this kind of simulations. In FASTEST-3D, a Smagorinsky subgrid scale model is implemented for LES. The code works on block-structured grids, which allows the treatment of complex geometries. The implementation of multigrid schemes and the parallelization and vectorization of all important code sections ensure efficient computations on high-performance computing platforms.

CFS++ is a simulation environment for multiphysics simulations. Currently, discretized forms of differential equations of structural mechanics, acoustics and electromagnetics are implemented. The discretization is done using the finite-element method. In the present work, CFS++ is used to solve the structural mechanics problem and the acoustic wave equation, which describes the sound propagation to the far field.

Provided that the acoustic pressure fluctuations are low compared with the fluid pressure, no feedback of acoustics on the fluid or on the structure has to be considered. Under this assumption, the fluid-structure interaction splits into three subproblems, which can be treated and computed separately: fluid-structure interaction, fluid-acoustic coupling and structure-acoustic coupling. Details of the coupling algorithms between the different physical domains are given in the following sections.

2.1 Fluid-structure interaction

The fluid-structure interaction is realized by an implicit coupling between FASTEST-3D and CFS++. As a coupling interface, MpCCI is used, which provides the interpolation of data between the fluid and the structural grid. A schematic of the implicit coupling for fluid-structure interaction is shown in Figure 2. At the beginning of the fluid-structure iteration loop, the flow solution is determined by FASTEST-3D. The resulting fluid loads (pressure forces and shear stress forces) on the structure are calculated and transmitted to CFS++ via MpCCI. Taking the fluid loads into account, the structural mechanics problem is solved. The displacement of the structure is then sent to FASTEST-3D. Based on the structural displacement and the resulting deformation of the fluid domain, the fluid grid is adapted. In most cases, no dynamic equilibrium between the flow and the structural mechanics solution is achieved by cycling the fluid-structure iteration loop only once. Therefore, the procedure is repeated within the same time step until equilibrium is reached. The convergence criterion is based on the change of the mechanical displacement $\mathbf{w}$ between two subsequent iterations:

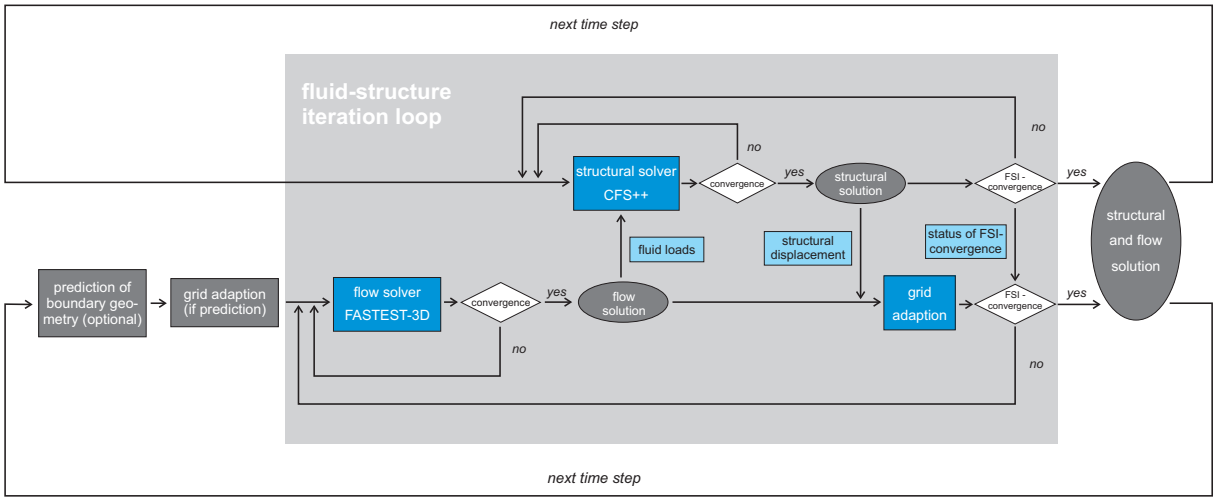


Figure 2: Schematic of the implicit algorithm for the fluid-structure interaction

$$\frac{|| \mathbf{w}_{l+1} - \mathbf{w}_l ||_2}{|| \mathbf{w}_{l+1} ||_2} \leq \varepsilon \tag{1}$$

where l denotes the iteration counter of the fluid-structure iteration loop, $||\ \ ||_2$ the L^2-norm and ε the desired accuracy. At the beginning of each time step, an optional prediction of the structural deformation can be carried out. The prediction extrapolates the position of the boundary using its positions at previous time steps. In many cases, this procedure can significantly reduce the number of CFD iterations needed until convergence is reached.

Since the fluid and structural grids are typically non-matching, an interpolation is necessary for the above-mentioned transfers of data between the flow and the structural mechanics solver. Depending on the type of data to be exchanged, different interpolation techniques are used. For the interpolation of the structural displacement, bilinear interpolation is empolyed. On the fluid-structure interface, for each grid node i of the fluid grid the according surface of the finite element e of the structural mechanics grid, in which i is located, is determined. Then the position ξ_i and η_i of the node i referring to the local coordinate system of the finite element (ξ, η) is calculated. The displacement $\mathbf{w}^e(\xi_i, \eta_i)$ of the node i is computed with the equation

$$\mathbf{w}^e(\xi_i, \eta_i) = \sum_{r=1}^{4} N^e_{K_r}(\xi_i, \eta_i) \mathbf{w}_{K_r} \tag{2}$$

where w_{K_r} are the displacements of the nodes K_1, K_2, K_3 and K_4 of the surface of the finite element e and $N^e_{K_r}(\xi, \eta)$ are the bilinear interpolation functions, defined as follows:

$$\begin{aligned} N^e_{K_1}(\xi, \eta) &= (1 - \xi)(1 - \eta) \\ N^e_{K_2}(\xi, \eta) &= \xi(1 - \eta) \\ N^e_{K_3}(\xi, \eta) &= \xi\eta \\ N^e_{K_4}(\xi, \eta) &= (1 - \xi)\eta \end{aligned} \tag{3}$$

An illustration of bilinear interpolation of the displacement from the structural mechanics to the fluid grid is given in Figure 3.

For the transfer of integral fluid loads, conservative interpolation is applied to ensure that the sum of the loads over the whole fluid-structure interface is the same on the fluid and the structural mechanics grid. As for bilinear interpolation, the coordinates ξ_i and η_i of a fluid grid node i in the local coordinate system of the surface of the finite element e of the structural mechanics grid are computed. The load contribution $\mathbf{L}^{e,i}_{K_r}$ to the nodes K_r of e due to the fluid load $\mathbf{L}_i$ at the grid node i is calculated using the following equation:

$$\mathbf{L}^{e,i}_{K_r} = N^e_{K_r}(\xi_i, \eta_i) \mathbf{L}_i \qquad \text{for} \qquad r = 1,\ 2,\ 3,\ 4 \tag{4}$$

The interpolation functions $N^e_{K_r}(\xi, \eta)$ are the same as in equation (3). Note that $\sum_{n=1}^{4} N^e_{K_r} = 1$, so that the

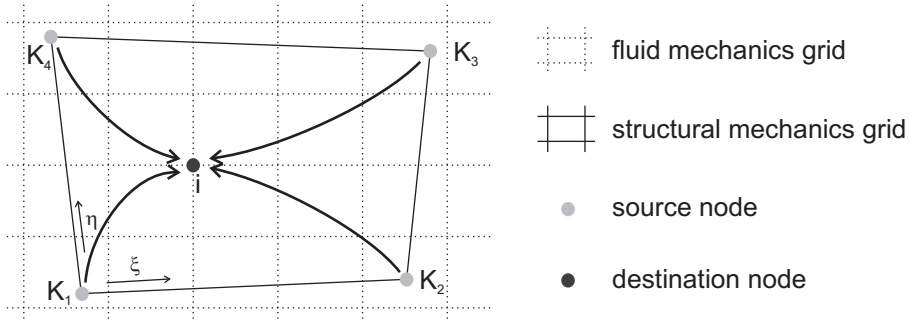


Figure 3: Bilinear interpolation of a variable from the structural mechanics to the fluid mechanics grid

interpolation is really conservative. The load vector $\mathbf{L}_i$ is defined as

$$\mathbf{L}_i = p \cdot \mathbf{n} \cdot S_i + \tau_w \cdot \mathbf{t} \cdot S_i \tag{5}$$

where p is the pressure, $\mathbf{n}$ the wall-normal vector at grid node i, S_i the area of the grid cell, τ_w the wall shear stress and $\mathbf{t}$ the tangential vector at i parallel to the local flow direction. Conservative interpolation of the fluid loads from the fluid to the structural mechanics grid is illustrated in Figure 4.

Due to the structural displacement and the resulting deformation of the flow boundaries, the fluid grid in the interior of the flow domain has to be adapted accordingly. For this purpose, robust and fast algebraic methods are used. If the deformation of the boundary is known on two opposite faces of a grid block, linear interpolation is employed to determine the grid deformation in between. Linear interpolation of the node displacement $\Delta\mathbf{w}$ along a grid line is visualized in Figure 5, following the equation

$$\Delta\mathbf{w}(\xi) = \xi\Delta\mathbf{w}(0) + (1-\xi)\Delta\mathbf{w}(1) \tag{6}$$

Linear interpolation cannot be used if two adjacent or more than two block faces of the grid have moved due to fluid-structure interaction. In this case, transfinite interpolation is applied. Transfinite interpolation is based on a superposition of two shear transformations and one tensor product transformation. The grid displacement $\Delta\mathbf{w}$ at the position (ξ, η) can therefore be computed with the following equation:

$$\Delta\mathbf{w}(\xi,\eta) = A(\xi,\eta) + B(\xi,\eta) - T(\xi,\eta) \tag{7}$$

with

$$\left.\begin{aligned} A(\xi,\eta) &= \xi\Delta\mathbf{w}(0,\eta) + (1-\xi)\Delta\mathbf{w}(1,\eta), \\ B(\xi,\eta) &= \eta\Delta\mathbf{w}(\xi,0) + (1-\eta)\Delta\mathbf{w}(\xi,1), \end{aligned}\right\} \text{ shear transformations} \tag{8}$$

$$T(\xi,\eta) = B(A(\xi,\eta)) = \eta A(\xi,0) + (1-\eta)A(\xi,1) \quad \text{tensor product transformation}$$

As a consequence of grid movement, the control volumes of the fluid grid change with time, which has to be taken into account in the finite-volume formulation of the Navier-Stokes equations. The integral form of the general

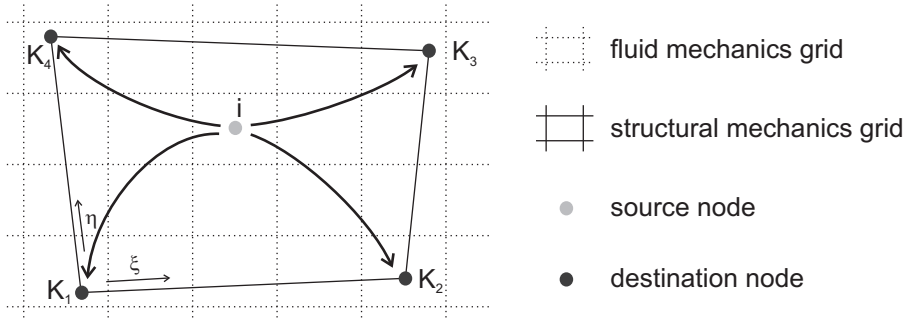


Figure 4: Conservative interpolation of a variable from the fluid mechanics to the structural mechanics grid

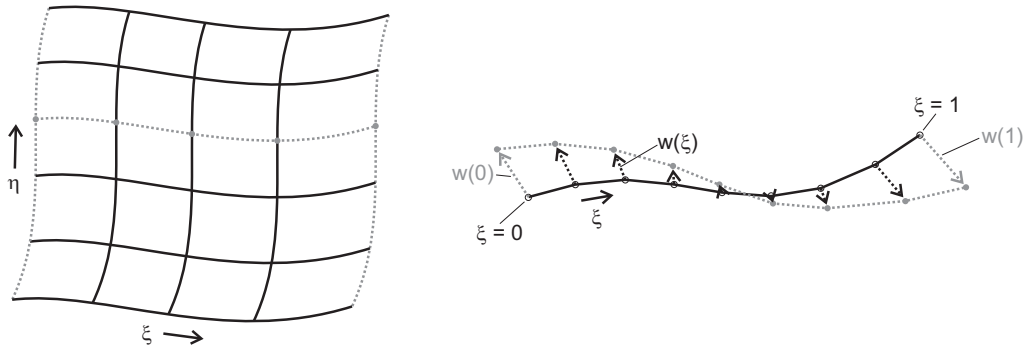


Figure 5: Linear interpolation of the node displacement along a grid line

conservation law for a variable Φ is given by

$$\int_V \frac{\partial(\rho\Phi)}{\partial t}\mathrm{d}V + \int_S \rho u_j \Phi \mathrm{d}S_j - \int_S \Gamma_\Phi \frac{\partial \Phi}{\partial x_j}\mathrm{d}S_j = \int_V Q_\Phi \mathrm{d}V \tag{9}$$

For moving grids, the volume V and the surface S of a control volume are functions of time, $V = V(t)$ and $S = S(t)$. By means of the three-dimensional Leibniz equation, the first term containing the time derivative can be transformed as follows:

$$\int_{V(t)} \frac{\partial(\rho\Phi)}{\partial t}\mathrm{d}V = \frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho\Phi \mathrm{d}V - \int_{S(t)} (\rho u_{g,j}\Phi)\mathrm{d}S_j \tag{10}$$

In this relationship, $u_{g,j}$ refers to the velocity of the grid movement. Inserting equation (10) in equation (9), one obtains:

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\Phi\mathrm{d}V}_{\substack{\text{rate of change in}\\ \text{a moving CV}}} + \underbrace{\int_S \rho u_j \Phi \mathrm{d}S_j}_{\substack{\text{convective fluxes over}\\ \text{a CV surface}}} - \underbrace{\int_S \rho u_{g,j}\Phi\mathrm{d}S_j}_{\substack{\text{convective grid}\\ \text{fluxes}}} - \underbrace{\int_S \Gamma_\Phi \frac{\partial\Phi}{\partial x_j}\mathrm{d}S_j}_{\substack{\text{diffusive fluxes over a}\\ \text{CV surface}}} = \underbrace{\int_V Q_\Phi \mathrm{d}V}_{\substack{\text{sources/sinks}\\ \text{in CV}}} \tag{11}$$

Summing the convective terms leads to the arbitrary Lagrangian-Eulerian formulation (ALE) of the conservation law:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\Phi\mathrm{d}V + \int_S \rho\left(u_j - u_{g,j}\right)\Phi\mathrm{d}S_j - \int_S \Gamma_\Phi \frac{\partial\Phi}{\partial x_j}\mathrm{d}S_j = \int_V Q_\Phi \mathrm{d}V \tag{12}$$

The convective flux over the control volume surface depends only on the differences between flow velocity u_j and grid velocity $u_{g,j}$. Special cases are $u_{g,j} = 0$ and $u_{g,j} = u_j$, which lead to an Eulerian or a Lagrangian formulation, respectively.

The discretization of the grid fluxes is based on the space conservation law, which is obtained by inserting $\Phi = 1$, $\rho =$ constant and $Q_\Phi = 0$ into equation (12):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \mathrm{d}V - \int_{S(t)} u_{g,j}\mathrm{d}S_j = 0 \tag{13}$$

This relationship must be fulfilled for each control volume. Its discretized form using an implicit three-point scheme of second order is

$$\int_{S(t)} u_{g,j}\mathrm{d}S_j = \sum_k \left(\frac{3\delta V_k^{n+1} - \delta V_k^n}{2\Delta t}\right), \qquad k = w,\ s,\ b,\ t,\ n,\ e \tag{14}$$

In this equation, the so-called swept volumes V_k^n and V_k^{n+1} are used. They represent the volume that is spanned by the movement of surface k of a control volume between the time steps $n-1$ and n or n and $n+1$, respectively. An illustration of swept volumes can be found in Figure 6. Equation (14) can be used to deduce a discretized form of the convective grid fluxes. Based on an implicit three-point scheme, this leads to the following discretization:

$$\int_{S(t)} (\rho u_{g,j}\Phi)\mathrm{d}S_j = \sum_k \left(\rho\Phi_k^{n+1}\frac{3\delta V_k^{n+1} - \delta V_k^n}{2\Delta t}\right) \tag{15}$$

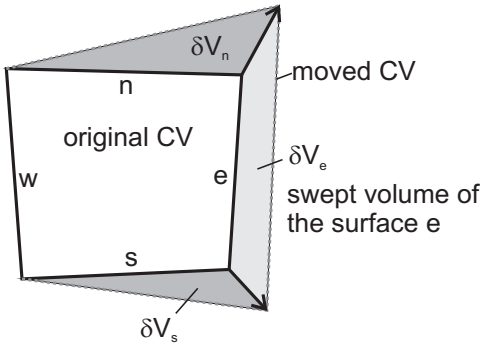


Figure 6: Illustration of swept volumes (CV = control volumes)

2.2 Fluid-acoustic coupling

The computation of flow-induced sound is based on Lighthill's analogy [see Lighthill (1952, 1954)]. Lighthill derived the following inhomogeneous wave equation, which describes the acoustic wave propagation to the far field, from the momentum equation of fluid mechanics:

$$\frac{1}{c^2}\frac{\partial^2 p'}{\partial t^2} - \frac{\partial^2 p'}{\partial x_j^2} = \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \tag{16}$$

The acoustic pressure fluctuations are denoted by p'. The so-called Lighthill tensor T_{ij} is defined as follows:

$$T_{ij} = \rho u_i u_j + \left[(p - p_0) - c^2(\rho - \rho_0)\right]\delta_{ij} + \tau_{ij} \tag{17}$$

In equation (17) ρ denotes the fluid density, p the fluid pressure, ρ_0 and p_0 the ambient density and pressure, c the speed of sound and τ_{ij} the viscous stress tensor. According to Lighthill's analogy, T_{ij} formulates the flow-induced sources of the inhomogeneous wave equation as acoustic quadrupole sources. In many applications, sound due to viscous effects and to entropy changes can be neglected, so that the second and the third terms on the right-hand side of equation (17) can be skipped. Only the turbulent source term $\rho u_i u_j$ is considered in the present work. Based on this approximation, the acoustic source term is calculated from the velocty field. Finally, the propagation of sound to the far field is computed by solving the inhomogeneous wave equation (16) using a finite-element formulation.

2.3 Structure-acoustic coupling

The vibrational sound is computed by imposing the structural movement as a boundary condition to the acoustic wave equation. The wall-normal component of the acoustic particle velocity u_i' must be equal to the wall-normal component of the structural velocity $u_{i\,\mathrm{struct}}$:

$$n_i u_i' = n_i u_{i\,\mathrm{struct}} \tag{18}$$

Using the formulation of the acoustic velocity potential φ for the acoustic wave equation that has to be solved, the left-hand side of equation (18) can be rewritten in a more convenient way ($u_i' = -\partial\varphi/\partial x_i$). Furthermore, the structural velocity $u_{i\,\mathrm{struct}}$ can be expressed as a time derivative of the structural displacement w_i. This leads to a von Neumann boundary condition for the acoustic velocity potential φ:

$$-n_i \frac{\partial \varphi}{\partial x_i} = n_i \frac{\partial w_i}{\partial t} \tag{19}$$

Based on this boundary condition, the propagation of vibrational noise to the far field is computed by solving the wave equation for φ using the finite-element method. The acoustic pressure p' is then obtained by $p' = \rho \cdot \partial\varphi/\partial t$.

Further information about the coupling algorithms can be found in the work of Schäfer et al. (2006, 2008). A benefit of the computation scheme presented for fluid-structure-acoustic interactions is the optimization of each code for its special field of application, which is due to the partitioned simulation approach. Additionally, FASTEST-3D and CFS++ can be adapted if necessary, because they are both in-house codes. Therefore, flexiblility is increased compared with commercial solutions. Moreover, an advantage over experimental methods is the possibility of computing separately the sound caused by the fluid flow on the one hand and by the structural movement on the other.

3 Application of the computational methodology

The presented computational methodology is applied to a simplified model of a car underbody. The setup consists of a thin, flexible plate that is part of an otherwise rigid wall. To model obstacles often found with real car underbodies, a square cylinder is mounted on the wall upstream of the flexible plate and perpendicular to the free stream (see Figure 7).

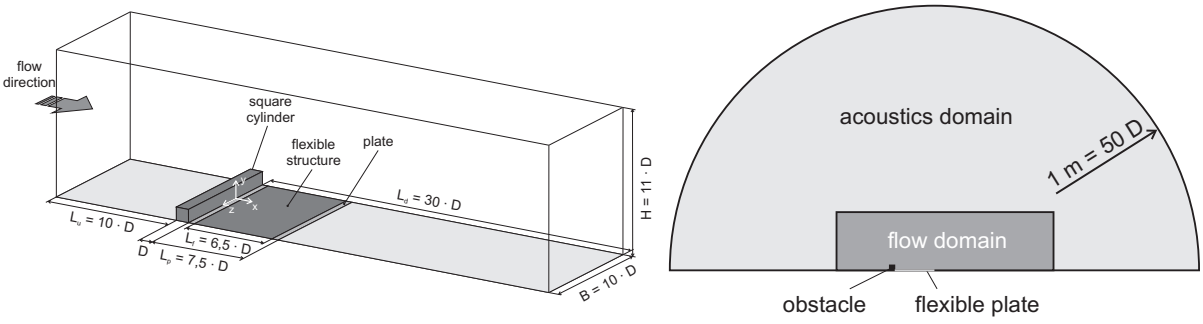
The flexible plate is made of stainless steel and has a thickness of 40 µm. The density is $\rho_\mathrm{s} = 7850\ \mathrm{kg/m^3}$, the modulus of elasticity $E = 2 \times 10^{11}\ \mathrm{kg/m/s^2}$, and the Poisson number is 0.3. The edge length of the square cylinder is $D = 0.02\ \mathrm{m}$ and the free-stream velocity U_∞ was set to 20 m/s. Considering a flow of air at ambient conditions, this corresponds to a Reynolds number $\mathrm{Re} = 26\,000$ based on U_∞ and D. The dimensions of the flow domain referring to D are shown in Figure 8(a). In dimensional notation they are $0.82 \times 0.2 \times 0.22\ \mathrm{m}^3$ (length $\times$ width $\times$ height). The flexible structure is clamped at both its upstream and downstream edges. The area including the clamping of the flexible structure is just called 'plate' in the following. The plate is pre-stressed in the main flow direction at a value of $7 \times 10^6\ \mathrm{N/m^2}$. The domain of aeroacoustic and vibrational sound is visualized in Figure 8(b). The computational domain of the acoustic calculations is much larger than the flow domain because we are interested in the radiation of noise to the far field.

In the flow simulation, for all walls no-slip boundary conditions are chosen. At the upper boundary, symmetry is assumed, whereas in the spanwise direction a periodic boundary condition is applied. To model the inflow, a measured velocity profile of the time-averaged main flow component is used. The other two velocity components at the inflow are set to zero. At the outlet, a convective outflow condition is imposed. The spatial discretization of the flow domain is realized by a block-structured grid. Although the geometry is fairly simple, a very high number of control volumes is necessary to resolve the boundary layers. The fluid grid consists of about six million control volumes, so that all boundaries can be resolved properly, which means that the dimensionless wall scale y^+ does not exceed a value of 0.4 based on the time-averaged velocity field. The flow simulation is carried out as LES using a Smagorinsky subgrid scale model. The spatial discretization utilizes a central differencing scheme and the temporal discretization is based on an implicit three-point scheme. Both are of second order of accuracy. The time step is chosen 2.5×10^{-5} s. Due to the large number of control volumes, use of high-performance computers is necessary. The computations run on four computing nodes, each equipped with two dualcore processors (Intel® Xeon® 5160) and 8 GB of main memory. First, the unsteady development of the flow field is computed until a fully developed turbulent state is reached, before finally fluid-structure interaction is activated.

For the structural mechanics model, periodic boundary conditions are applied in the spanwise direction. The spatial discretization of the computational domain is done by hexahedral elements. The number of nodes is about $13\,000$. In the finite-element computation of structural mechanics, bilinear basis functions with an incompatible mode approach to account for shear locking effects are used. Moreover, an implicit second-order time discretization scheme of the Newmark type is applied [see Kaltenbacher (2007)]. The time step size of the structural mechanics simulation is the same as that for the flow simulation.

For the free borders of the acoustic domain, absorbing boundary conditions are chosen, while the rigid walls are acoustically reflective. At the flexible wall, we apply the inhomogeneous von Neumann boundary condition of equation (19). As for the structural mechanics grid, the spatial discretization for the acoustic domain is made up of hexahedral elements. The number of nodes is approximately $450\,000$.

Figure 7: Setup of the test case

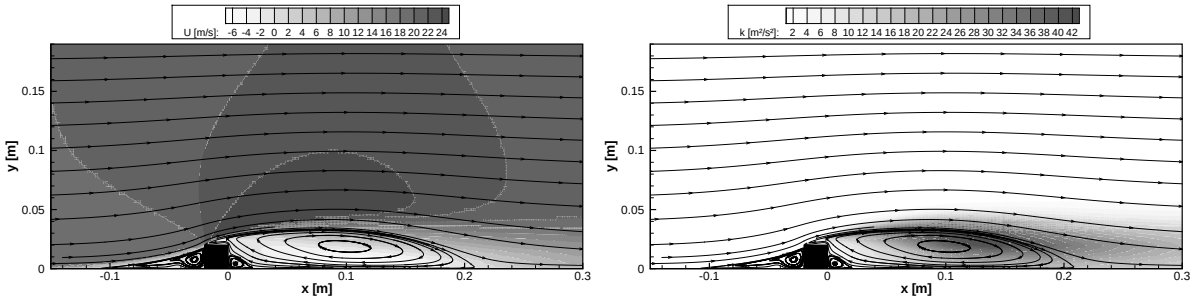


(a) Flow domain (b) Acoustics domain

Figure 8: Computational domains

4 Results

The computed flow field averaged in time and in the spanwise direction is depicted in Figure 9. Figure 9(a) shows the average velocity in the main flow direction and Figure 9(b) the distribution of the turbulent kinetic energy. Behind the square cylinder obstacle, a recirculation region with a length of about $11\ D$ can be seen. Additionally,



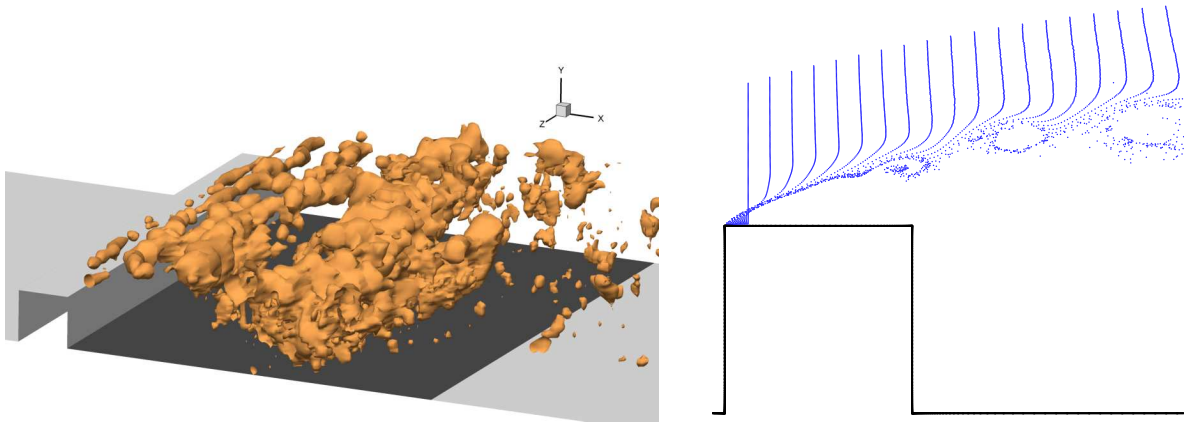
(a) Average velocity in the main flow direction (b) Distribution of the turbulent kinetic energy

Figure 9: Flow field averaged in time and in the spanwise direction

in front of the cylinder and also directly behind it, smaller secondary vortices appear. In front of the obstacle no significant turbulence is found. The laminar to turbulent transition is taking place in the shear layer above the cylinder. Instabilities in this region lead to the formation of unsteady vortex structures and therefore to an increased amount of turbulent kinetic energy.

The vortex formation behind the square cylinder is visualized in Figure 10. Figure 10(a) shows a snapshot of pressure isosurfaces and Figure 10(b) timelines based on particle tracing techniques, both at an arbitrary instant of time. It can be seen very well how instabilities in the shear layer between the main flow and the recirculation region evolve to vortices, which become larger while they are convectively transported downstream. The darker part of the wall in Figure 10(a) represents the flexible plate. Especially in the downstream part of the plate the vortices come close to the plate, which makes an interaction between the flow and the structure likely.

The bending of the plate at an arbitrary instant of time is shown in Figure 11. The structural displacement is clearly dominated by the first eigenmode of the plate. Additionally, a higher eigenmode with a lower amplitude of displacement can be seen. Both modes are uniform in the spanwise direction. The temporal evolution of the displacement of the point in the middle of the plate is shown in Figure 12(a). The first eigenmode of the plate clearly dominates the amplitude of the oscillation. The amplitudes of the higher frequency is considerably smaller. The amplitudes of both the first and of the higher eigenmode are increasing over time. Possibly, the geometric linear model used in the structural mechanics computations is not suited for the present problem. For future investigations, the use of a non-linear model should be considered. The frequency spectra of the displacement and the velocity of the point in the middle of the plate are plotted in Figure 12(b). The frequencies corresponding to the two mentioned eigenmodes can be seen as peaks around 140 and 1380 Hz. The other peaks found in the spectra correspond to further eigenmodes, which are all homogeneous in the spanwise direction. No inhomogeneous eigenmodes are excited in the present case, although such inhomogeneous modes are observed in a modal analysis



(a) Pressure isosurfaces (b) Timelines

Figure 10: Vortex formation behind the square cylinder obstacle (snapshots at an arbitrary instant of time)

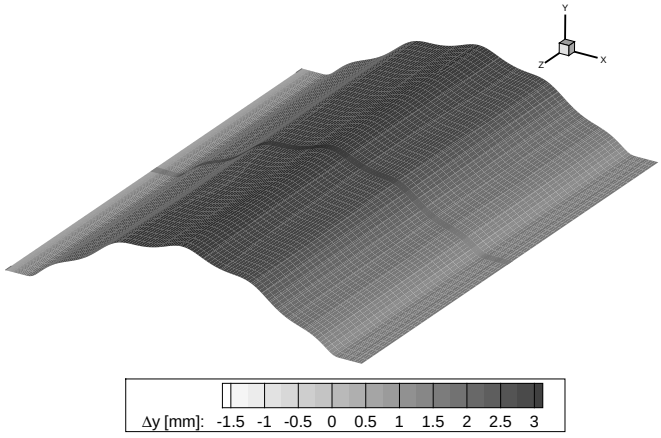


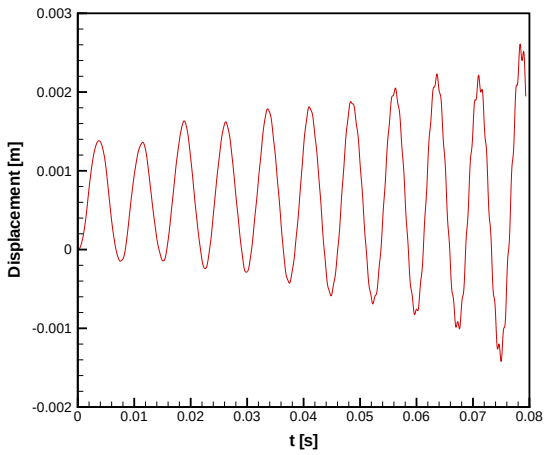
Figure 11: Snapshot of the plate displacement (visualization of displacement magnified by a factor of 20)

of the plate.

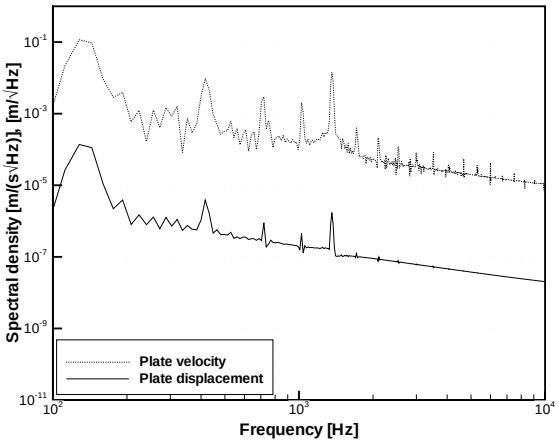
The coupled computation scheme described above allows for the separate determination of flow-induced and vibrational sound. The spectral density of the sound pressure level for the two sound components is depicted in Figure 13. The evaluation considers three angular positions on the boundary of the acoustic computation domain [see Figure 8(b)]. The aeroacoustic sound predominantly shows broadband noise. Nevertheless, the broad peak around 140 Hz in Figure 13(a) indicates an influence of the structural movement on the flow-induced sound. As the structural movement acts directly as a source for the computation of vibrational sound, the same peaks as in Figure 12(b) can be found in the spectrum of the vibrational sound [Figure 13(b)].

5 Conclusion

A coupled simulation approach for fluid-structure-acoustic interactions was presented, which assumes acoustic pressures that are low compared with the fluid pressure. This simplification is valid for many of the technically relevant problems of fluid-structure-acoustic interaction. In this case the problem splits into three subproblems, which can be treated and computed separately: fluid-structure interaction, fluid-acoustic coupling and structure-acoustic coupling. A partitioned simulation approach with a finite-volume CFD code for the flow computation and a multiphysics finite-element code for structural mechanics and acoustics was used. Both are inhouse codes, which increases flexibility and adaptability. An implicit coupling between fluid mechanics and structural mechanics was applied for the simulation of fluid-structure interactions. Computations of flow-induced sound were based on Lighthill's analogy. The acoustic sources for the inhomogeneous wave equation were determined from the velocity field of the fluid simulation. The evaluated structural displacement was used as a boundary condition for the computation of the vibrational sound. The flow-induced and the vibrational sound were computed separately,

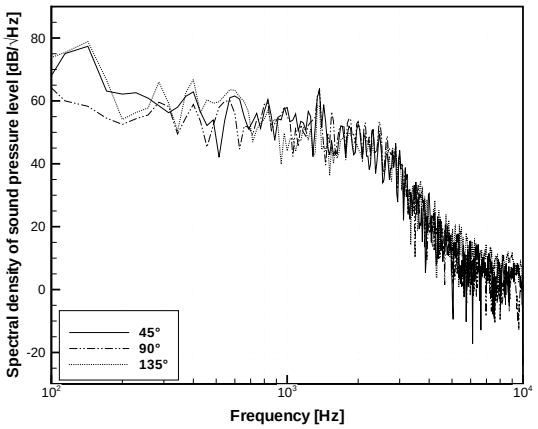


(a) Temporal evolution of the displacement

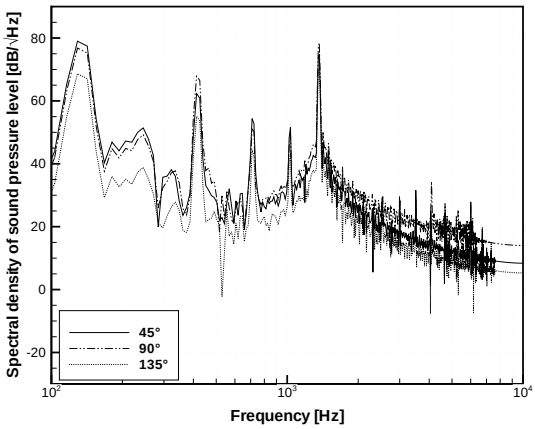


(b) Frequency spectra of the displacement and the velocity of the plate

Figure 12: Displacement of a point in the middle of the plate



(a) Flow-induced sound



(b) Vibrational sound

Figure 13: Spectral density of the sound pressure level at three angular positions on the boundary of the acoustic domains (xy plane, 0° corresponds to the main flow direction)

which is a great advantage over experimental methods.

The simulation approach presented was applied to a simplified model of a car underbody. The acoustic field due to the interaction between a thin, flexible plate in the wake of a wall-mounted square cylinder and a turbulent flow was investigated. Detailed analysis of the obtained data gave reasonable results, which shows the capability of the simulation approach to treat fluid-structure-acoustic interactions. To validate the numerical results, comparisons with experimental data have to be made in the future.

Acknowledgment

Financial support from the Bavarian Research Foundation (Bayerische Forschungsstiftung, BFS) is gratefully acknowledged.

References

Ahrem, R.; Hackenberg, M. G.; Redler, R.; Roggenbunck, J.: MpCCI – Mesh-Based Parallel Code Coupling Interface. Tech. rep., Institute of Algorithms and Scientific Computing (SCAI), GMD, http://www.mpcci.org (2003).

Davies, H. G.: Sound from Turbulent-Boundary-Layer-Excited Panels. *Journal of the Acoustical Society of America*, 49, 3 (Part 2), (1971), 878–889.

Durst, F.; Schäfer, M.: A Parallel Block-Structured Multigrid Method for the Prediction of Incompressible Flows. *International Journal of Numerical Methods in Fluids*, 22, (1996), 549–565.

Frampton, K. D.; Clark, R. L.: Power Flow in an Aeroelastic Plate Backed by a Reverberant Cavity. *Journal of the Acoustical Society of America*, 102, 3, (1997), 1620–1627.

Graham, W. R.: Boundary Layer Induced Noise in Aircraft, Part I: The Flat Plate Model. *Journal of Sound and Vibration*, 192, 1, (1996), 101–120.

Howe, M. S.; Shah, P. L.: Influence of Mean Flow on Boundary Layer Generated Interior Noise. *Journal of the Acoustical Society of America*, 99, 6, (1996), 3401–3411.

Kaltenbacher, M.: *Numerical Simulation of Mechatronic Sensors and Actuators*. Springer, 2nd edn. (2007).

Kaltenbacher, M.; Hauck, A.; Triebenbacher, S.; Link, G.; Bahr, L.: CFS++: Coupled Field Simulation. Tech. rep., Department of Sensor Technology, University of Erlangen-Nuremberg (2007).

Lighthill, M. J.: On Sound Generated Aerodynamically – I. General Theory. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 211, (1952), 564–587.

Lighthill, M. J.: On Sound Generated Aerodynamically – II. Turbulence as a Source of Sound. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 222, (1954), 1–32.

Mazzoni, D.; Kristiansen, U.: Finite Difference Method for the Acoustic Radiation of an Elastic Plate Excited by a Turbulent Boundary layer: a Spectral Domain Solution. *Flow, Turbulence and Combustion*, 61, (1999), 133–159.

Schäfer, F.; Ali, I.; Becker, S.; Kaltenbacher, M.; Escobar, M.; Link, G.: Computational Aeroacoustics using MpCCI as Coupling Interface between Fluid Mechanics, Structural Mechanics and Acoustics. In: *Proceedings of the 7th MpCCI User Forum*, pages 98–111, St. Augustin, Germany (February 21–22 2006).

Schäfer, F.; Kniesburges, S.; Uffinger, T.; Becker, S.; Grabinger, J.; Link, G.; Kaltenbacher, M.: Numerical Simulation of Fluid-Structure- and Fluid-Structure-Acoustic Interaction based on a Partitioned Coupling Scheme. In: *High Performance Computing in Science and Engineering*, Springer, Munich (2008).

Tang, S. K.; Leung, R. C. K.; So, R. M. C.; Lam, K. M.: Acoustic Radiation by Vortex Induced Flexible Wall Vibration. *Journal of the Acoustical Society of America*, 118, 4, (2005), 2182–2189.

Vergne, S.; Auger, J.-M.; Périé, F.; Jacques, A.; Nicolopoulos, D.: Aeroelastic Noise. In: C. Wagner; T. Hüttl; P. Sagaut, eds., *Large-Eddy Simulation for Acoustics*, pages 272–293, Cambridge University Press, Cambridge (2007).

Visbal, M. R.; Gordnier, R. E.: Numerical simulation of the interaction of a transitional boundary layer with a 2-D flexible panel in the subsonic regime. *Journal of Fluids and Structures*, 19, (2004), 881–903.

Zolotarev, I.: Fluid-Structural-Acoustical Interactions of a Thin Plate in a Channel with Flowing Fluid. *American Society of Mechanical Engineers*, 1, (1997), 403–410.

Address: Dipl.-Ing. T. Uffinger, Dr. F. Schäfer, Dr. S. Becker, Institute of Fluid Mechanics (LSTM), University of Erlangen-Nuremberg, Cauerstraße 4, 91058 Erlangen, Germany
email: thomas.uffinger@lstm.uni-erlangen.de,
frank.schaefer@lstm.uni-erlangen.de, stefan.becker@lstm.uni-erlangen.de
M.Sc. J. Grabinger, Department of Sensor Technology (LSE), University of Erlangen-Nuremberg, Paul-Gordan-Straße 3/5, 91052 Erlangen, Germany
email: jens.grabinger@lse.eei.uni-erlangen.de
Prof. M. Kaltenbacher, Chair of Applied Mechanics, University of Klagenfurt, Universitätsstraße 67–69, 9020 Klagenfurt, Austria
email: manfred.kaltenbacher@uni-klu.ac.at

Fluid-Structure Interaction.
Theory, Numerics and Applications
pp. 265– 276
Herrsching am Ammersee, 29.9.-1.10.2008

Projection-based reduced-order model of strongly-coupled fluid-structure interaction by monolithic space-time modes

A. Zilian, A. Vehre, D. Dinkler

The focus of this work is the development of reduced models for engineering applications in complex bidirectional fluid-structure interaction. In the simultaneous solution procedure, velocity variables are used for both fluid and solid, and the whole set of model equations is discretized by a stabilized time-discontinuous space-time finite element method. Flexible structures are modeled using a three-dimensional continuum approach in a total Lagrangian setting considering large displacements and rotations. In the flow domain the incompressible Navier-Stokes equations describe the Newtonian fluid. A continuous finite element mesh is applied to the entire spatial domain, and the discretized model equations are assembled in a single set of algebraic equations, considering the two-field problem as a whole. The continuous fluid-structure mesh with identical orders of approximation for both solid and fluid in space and time automatically yields conservation of mass, momentum and energy at the fluid-structure interface. A mesh-moving scheme is used to adapt the nodal coordinates of the fluid space-time finite element mesh to the structural deformation. The computational approach for strongly coupled fluid-structure interaction is used to create suitable reduced models of generic nonlinear problems. Reduction is performed with monolithic projection-based space-time modes, ensuring strong coupling of fluid and structure in the reduced model. The contribution discusses results using proper orthogonal decomposition (POD) for determination of monolithic space-time modes in the reduction of fluid-structure systems.

Address: A. Zilian (corresponding author), Institute for Structural Analysis, Technische Universität Braunschweig, Beethovenstr. 51, Braunschweig (GERMANY).
email: a.zilian@tu-bs.de

1 Introduction

Numerical simulations of fluid-structure interaction phenomena are still computationally expensive. In this contribution, a methodology for a strongly coupled model with space-time-discretization is described, so that simulations of fluid-structure interaction problems can be carried out using a reduced coupled model. The reduction of the original model, which is discretized by the space-time finite element method for both continua, is achieved by projection with space-time modes, i.e. the reduction modes contain a discretization both in space and time, see Vehre (2006). Numerical examples show that complex fluid-structure interaction problems can be simulated rather accurately using the reduced model with only few generalized coordinates.

2 Governing equations and space-time weak formulation

The following sections give the mathematical description of the boundary-coupled fluid-structure problem, involving an elastic structure at large deformations and a viscous incompressible fluid in terms of the strong form equations. The weak form is derived in context of the space-time finite element method and a monolithic solution approach. The final weak formulation is constructed in terms of velocities for the involved continua, simplifying the realization of coupling conditions considerably.

2.1 Strong form of fluid and structure

Structure. The conservation of momentum of a solid body at small strains is described on the material configuration

$$\rho_0 \dot{\mathbf{v}} - \nabla_0 \cdot (\mathbf{F}\mathbf{S}) - \mathbf{f}_0 = \mathbf{0} \quad \text{on} \quad Q_0 \tag{1}$$

with density ρ_0 and volume force $\mathbf{f}_0$. The domain $Q_0 = \Omega_0 \times I$ refers to the space-time continuum under consideration with the spatial reference domain Ω_0 and time interval $I = [t_a, t_e]$. A linear elastic material behavior is assumed and given in its rate form

$$\underline{\mathbf{C}}^{-1} : \dot{\mathbf{S}} - \dot{\mathbf{E}} = \mathbf{0}, \tag{2}$$

where $\mathbf{S}$, $\dot{\mathbf{E}}$ and $\underline{\mathbf{C}}$ are 2nd Piola-Kirchhoff stress, Green-Lagrange strain rate and fourth order elasticity tensor, respectively. The rate of strain at a material point is a function of deformation and velocity state

$$\dot{\mathbf{E}}(\mathbf{v}, \mathbf{u}) = \frac{1}{2}\left(\nabla_0 \mathbf{v} + (\nabla_0 \mathbf{v})^T + (\nabla_0 \mathbf{u})^T \nabla_0 \mathbf{v} + (\nabla_0 \mathbf{v})^T \nabla_0 \mathbf{u}\right). \tag{3}$$

Dirichlet and Neumann boundary conditions are defined on the outer space-time boundary $P_0 = \Gamma_0 \times I$ of the solid body

$$\mathbf{v} - \bar{\mathbf{v}} = \mathbf{0} \quad \text{on} \quad P_0^v \qquad \text{and} \qquad \mathbf{t}_0 - \bar{\mathbf{t}}_0 = \mathbf{0} \quad \text{on} \quad P_0^t, \tag{4}$$

where $\bar{\mathbf{v}}$ and $\bar{\mathbf{t}}_0$ are imposed boundary velocities and tractions, respectively. Moreover, the dynamic problem at hand requires the definition of initial values for the velocity state

$$\mathbf{v}(\mathbf{x}, t = 0) - \mathbf{v}_{t_a}(\mathbf{x}) = \mathbf{0} \quad \text{on} \quad Q_0 \tag{5}$$

in the spatial domain.

Fluid. The incompressible Navier-Stokes equations are used to describe viscous flow at moderate speeds. Momentum balance is defined on the current configuration

$$\rho(\mathbf{v}_{,t} + \mathbf{v} \cdot \nabla \mathbf{v}) - \nabla \cdot \mathbf{T} - \mathbf{f} = \mathbf{0} \quad \text{on} \quad Q \tag{6}$$

and accompanied by the continuity equation or incompressibility contraint on the velocity field

$$\nabla \cdot \mathbf{v} = 0 \quad \text{on} \quad Q. \tag{7}$$

Assuming a Newtonian fluid, the constitutive relation

$$\mathbf{T} = 2\mu \mathbf{D}(\mathbf{v}) - p\mathbf{I} \tag{8}$$

between Cauchy stress state $\mathbf{T}$ and the rate of strain

$$\mathbf{D}(\mathbf{v}) = \frac{1}{2}\left(\nabla_0 \mathbf{v} + (\nabla_0 \mathbf{v})^T\right) \tag{9}$$

introduces hydrostatic pressure p and viscosity μ of the fluid. Again, Dirichlet and Neumann boundary conditions are defined on the space-time boundary $P = \Gamma \times I$ of the fluid body

$$\mathbf{v} - \bar{\mathbf{v}} = \mathbf{0} \quad \text{on} \quad P^v \qquad \text{and} \qquad \mathbf{t} - \bar{\mathbf{t}} = \mathbf{0} \quad \text{on} \quad P^t, \tag{10}$$

where $\bar{\mathbf{v}}$ and $\bar{\mathbf{t}}$ are imposed boundary velocities and tractions, respectively. Compatible (divergence-free) initial values for the velocity state

$$\mathbf{v}(\mathbf{x}, t = 0) - \mathbf{v}_{t_a}(\mathbf{x}) = \mathbf{0} \quad \text{on} \quad Q_0 \tag{11}$$

are required in the whole spatial domain.

Coupling conditions. Flow domain and structural domain are coupled along the common space-time boundary R representing the fluid-structure interface. At the interface no-slip conditions are applied, requesting continuity of fluid and structural velocities

$$\mathbf{v}^f - \mathbf{v}^s = \mathbf{0} \quad \text{on} \quad R \tag{12}$$

ensuring herewith geometrical conservation. Further, in order to fulfill momentum balance at the interface, fluid and solid interfacial tractions have to be of the same magnitude and opposite direction

$$\mathbf{t}^f + \frac{d\Gamma_0}{d\Gamma}\mathbf{t}^s = \mathbf{0} \quad \text{on} \quad R. \tag{13}$$

The referential solid traction $\mathbf{t}^s$ is projected to the current frame.

2.2 Weak form

The weighted residual method is applied to the strong form equations of solid, fluid and coupling conditions presented in the previous section. The resulting weak form of the whole coupled system and the space-time domain is then discretized using the space-time finite element method, see Argyris and Scharpf (1969), and a time-discontinuous Galerkin method for integration in time. The basic idea of a space-time discretization is to include the temporal axis in the finite element discretization. For numerical efficiency the space-time domain Q is divided into a sequence of N time slabs $Q^n = \Omega^n \times [t_n, t_{n+1}]$, as shown in Figure 1, which are solved successively.

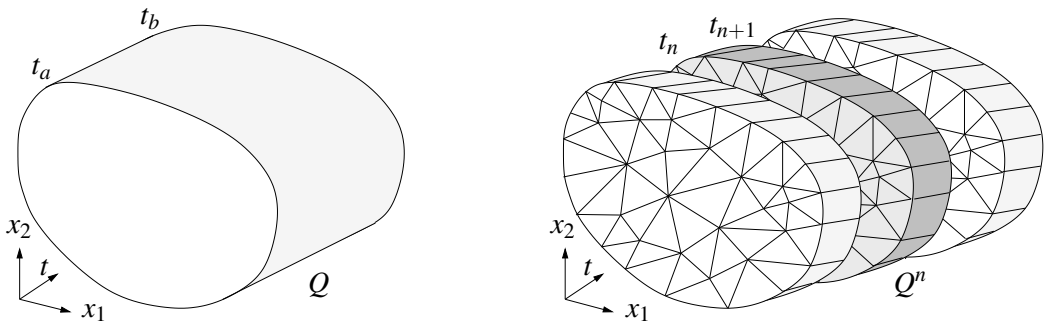


Figure 1: Discretization of the continuous space-time domain using space-time finite elements.

At time instant t_n the energy of the discretized system at the end of the previous time slab t_n^- must be equal to the energy at the beginning of the next time step t_n^+. For time-discontinuous approximations of field unknowns this leads to additional jump terms in the weak form. Moreover, spatial discretizations from t_n^- and t_n^+ do not need to be conforming. For first order ordinary differential equations the resulting time integration scheme is A-stable and third-order accurate for linear temporal interpolation.

Solid. The stabilized space-time finite element formulation of the structural part (1)-(5) within the time slab $Q_{0,n}$ including boundaries $P_{0,n}$ reads

$$\int_{Q_{0,n}} \delta\mathbf{v} \cdot \rho_0\, \dot{\mathbf{v}}\; dQ_0 \;+ \int_{Q_{0,n}} \dot{\mathbf{E}}(\delta\mathbf{v}, \mathbf{u}) : \mathbf{S}\; dQ_0 \;- \int_{Q_{0,n}} \delta\mathbf{v} \cdot \mathbf{f}_0\; dQ_0 \tag{14a}$$

$$+ \sum_e \int_{Q^e_{0,n}} \delta\mathbf{S} : \big(\underline{\mathbf{C}}^{-1} : \dot{\mathbf{S}} - \dot{\mathbf{E}}(\mathbf{v}, \mathbf{u})\big)\; dQ_0 \tag{14b}$$

$$+ \int_{\Omega_0} \delta\mathbf{v}(t_n^+) \cdot \rho_0 \big(\mathbf{v}(t_n^+) - \mathbf{v}(t_n^-)\big)\; d\Omega_0 \tag{14c}$$

$$+ \sum_e \int_{\Omega_0^e} \delta\mathbf{S}(t_n^+) : \underline{\mathbf{C}}^{-1} : \big(\mathbf{S}(t_n^+) - \mathbf{S}(t_n^-)\big)\; d\Omega_0 \tag{14d}$$

$$+ \sum_e \int_{Q^e_{0,n}} \tau_M^S\, \delta\dot{\mathbf{v}} \cdot \big(\rho_0\, \dot{\mathbf{v}} - \mathrm{div}_0(\mathbf{F}\mathbf{S}) - \mathbf{f}_0\big)\; dQ_0 \tag{14e}$$

$$- \int_{P^t_{0,n}} \delta\mathbf{v} \cdot \bar{\mathbf{t}}_0\; dP_0 \;=\; 0 \qquad \forall\; \delta\mathbf{v},\, \delta\mathbf{S}\;. \tag{14f}$$

In equation (14) line (a) represents the weak form of the momentum conservation and line (b) fulfills the constitutive law on element level, leading to the mixed-hybrid formulation of (Knippers and Harbord, 1994), where only the velocities are global degrees of freedom. The jump terms for velocities (c) and stresses (d) satisfy the initial conditions of the time slab in integral form. The stabilization term, which is needed for wave propagation problems, is shown in line (e). For the parameter τ_M^S the definition of (Hughes and Hulbert, 1988) is used. Interpolation functions for velocities are chosen to be bilinear in space and discontinuous linear in time, while the stress interpolation is discontinuous and incompletely linear in space and discontinuous linear in time, see (Hübner, 2003). Time integration of the velocities leads to the displacement field $\mathbf{u}$, which is used for computation of the rate of the Green-Lagrange strain tensor and for specifying the current position of the fluid-structure interface.

Fluid. The weighted residual formulation of the strong forms of the incompressible viscous fluid (6)-(11) in a space-time slab Q_t^n using the Galerkin method reads as

$$\int_{Q_t^n} \delta\mathbf{v} \cdot \rho(\mathbf{v}_{,t} + \mathbf{v} \cdot \nabla\mathbf{v}) \,\mathrm{d}Q_t + \int_{Q_t^n} \mathbf{D}(\delta\mathbf{v}) \,:\, 2\mu\mathbf{D} \,\mathrm{d}Q_t - \int_{Q_t^n} \nabla \cdot (\delta\mathbf{v})\, p \,\mathrm{d}Q_t - \int_{Q_t^n} \delta\mathbf{v} \cdot \rho\mathbf{b} \,\mathrm{d}Q_t \tag{15a}$$

$$+ \int_{Q_t^n} \delta p \,\nabla \cdot \mathbf{v} \,\mathrm{d}Q_t \tag{15b}$$

$$+ \int_{\Omega_t^n} \delta\mathbf{v}(t_n^+) \cdot \rho(\mathbf{v}(t_n^+) - \mathbf{v}(t_n^-)) \,\mathrm{d}\Omega_t \tag{15c}$$

$$+ \sum_e \int_{\overset{e}{Q_t^n}} (\rho\delta\mathbf{v}_{,t} + \rho\mathbf{v} \cdot \nabla(\delta\mathbf{v}) - \nabla \cdot \mathbf{T}(\delta\mathbf{v}, \delta p) \cdot \tau \frac{1}{\rho} (\rho\mathbf{v}_{,t} + \rho\mathbf{v} \cdot \nabla\mathbf{v} - \nabla \cdot \mathbf{T} - \rho\mathbf{b}) \,\mathrm{d}Q_t \tag{15d}$$

$$- \int_{P_t^{n,h}} \delta\mathbf{v} \cdot \overline{\mathbf{t}} \,\mathrm{d}P_t = 0 \qquad \forall \delta\mathbf{v}, \delta p. \tag{15e}$$

Line (15a) represents the weak form of conservation of momentum, fulfilling the constitutive relation for the Newtonian fluid and the kinematics exactly. The incompressibility constraint is weighted with the variation of the pressure in line (15b). Line (15c) ensures the consistent transfer of kinetic energy from the previous time slab end at t_n^- to the current time slab at t_n^+. The weighted residual form is stabilized by a Galerkin/least squares term, see Masud and Hughes (1997), of the momentum balance in line (15d). The Galerkin/least squares stabilization suppresses numerical oscillations in solutions to hyperbolic differential equations by the introduction of additional numerical diffusion, allowing herewith the application of equal order approximations of velocities and pressure for the incompressible flow field. The stabilization parameter τ is determined for each element e using the definition given by Tezduyar et al. (1992). Neumann boundary conditions can be imposed in a weak sense by (15e), while (10) is treated a priori as an essential boundary condition.

Fluid-structure coupling. The space-time finite element formulation of the coupling conditions (12)-(13) uses boundary traction variables on the interface in order to fulfill momentum conservation and geometrical continuity between structure and fluid in integral form, see Hübner et al. (2004). In particular, the velocity values of the Dirichlet boundary conditions of the fluid are substituted by structural velocities and the fluid boundary tractions act as surface loads onto the structure

$$\int_{P^c} \delta\mathbf{t}^F \cdot \left(\mathbf{v}^F - \mathbf{v}^S\right) \,\mathrm{d}P \;-\; \int_{P^c} \delta\mathbf{v}^F \cdot \mathbf{t}^F \,\mathrm{d}P \;-\; \int_{P_0^c} \delta\mathbf{v}^S \cdot \left(-\frac{d\Gamma_t}{\mathrm{d}\Gamma_0}\,\mathbf{t}^F\right) dP_0 \;. \tag{16}$$

The tractions are projected onto the reference configuration and change their sign, since the unit outward normal vectors of fluid and structure are of opposite directions.

As an alternative, one may use direct coupling of fluid and structural velocity degree of freedom in case of fitting mesh discretizations along the common fluid-structure interface. This enables the feature of an unconstrained formulation in terms of the coupling conditions and improves efficiency in the numerical solution.

Monolithic system of equations. The monolithic discrete form of fluid, structure, and coupling conditions of one space-time slab results in a single equation system, shown in Figure 2 (depicted without pressure) and therefore allows the analysis of the strongly coupled system.

The resulting highly nonlinear system – describing both fluid and structure – is solved by a Picard iteration scheme in terms of secants.

$$\mathbf{A}(\mathbf{x}_{i-1}^n)\mathbf{x}_i^n = \mathbf{b}(\mathbf{x}^{n-1}) \tag{17}$$

Nonlinearities in the coefficient matrix $\mathbf{A}$ are due to nonlinear kinematics of the structure, the convection and stabilization terms of the fluid and the motion of the fluid space-time mesh. In the considered applications, the iteration scheme reaches the coupled monolithic solution to a time slab within 3 to 4 steps. Within each nonlinear iteration step i the monolithic linear system of equations is solved by an ILU(k)-preconditioned GMRES solver with restart 50.

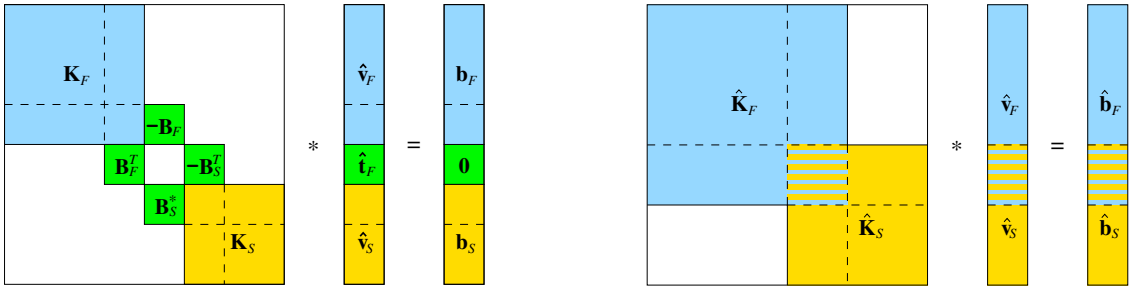


Figure 2: Structure of algebraic system: use of Lagrange multipliers (left) and native velocity coupling (right).

For an in-depth description of the design and implementation of the space-time finite element method for fluid-structure interaction, the interested reader is referred to the following publications: Walhorn (2002); Hübner (2003); Hübner et al. (2004); Walhorn et al. (2005); Kölke (2005); Hübner and Dinkler (2005); Zilian and Legay (2008), where the eligibility of the monolithic space-time finite element method for strongly coupled fluid-structure interaction problems is demonstrated and validated on a number of applications in aero- and hydro-elasticity.

3 Modal reduction of strongly-coupled fluid-structure interaction

Due to requirements of resolving boundary layers, detachment and vortex shedding sufficiently correct, the discretization of fluid-structure interaction problems results in large-scale algebraic systems. As a side effect of refined meshes around flow-immersed structures, one increases the band of higher frequencies in the solution spectrum of the coupled problem. Concerning efficient and stable solution procedures this behavior is not desirable and in terms of the physical phenomena also not necessary for a number of fluid-structure problems.

The key of modal reduction methods is to filter out less important information in the solution space to a given problem. For fluid-structure interaction this relates to identification of major interaction effects for a nonlinear problem at hand and to build an efficient reduced problem-dependent model of predefined accuracy to the unreduced physics. Projection-based system reduction is focused on lowering the need to computational resources (memory and computation time) by making the transition from a large number of (nodal) natural degrees of freedom to a small number of generalized degrees of freedom. Together with domain-wide specified basis functions (modes) the generalized coordinates should be able to approximate the system behavior. This approach is especially advantageous if a system has to be investigated numerically for a number of different modes of excitation resulting in qualitative similar responses.

3.1 Projection-based modal reduction

The monolithic nonlinear discrete system in Eq. (17) describes the coupled behavior of the full fluid-structure system in terms of the natural degrees of freedom $\mathbf{x}^T = [\hat{\mathbf{v}}_\mathbf{s}^T \;\; \hat{\mathbf{v}}_\mathbf{f}^T \;\; \hat{\mathbf{p}}^T]$ (velocities and pressure). In a projection-based reduction approach the approximation space $\mathcal{V}^h$ of the natural solution of discrete dimension M is projected to the lower-dimensional space of the generalized basis $\mathcal{P}^h$

$$\mathbf{x} = \mathbf{P}\mathbf{y}, \tag{18}$$

where $\mathbf{y}$ is the vector of N generalized degrees of freedom and $\mathbf{P} = [\mathbf{p}_1 \;\; \mathbf{p}_2 \;\; \ldots \;\; \mathbf{p}_N]$ contains the associated reductions vectors (projection modes) column-wise. The minimum requirement to the set of reduction vectors is linear independency. If the behavior of the fluid-structure model is mainly governed by large coarse-scale coherent structures in fluid and solid, one usually choses $N \ll M$. While within the fluid energy is dissipated on the small scales, large-scale effects contain the majority of kinetic energy, allowing good representation of dynamic fluid-structure phenomena.

The residual of the discrete natural weak form to the monolithic fluid-structure formulation is transformed in terms of a Bubnov-Galerkin projection

$$\mathbf{P}^T \, \mathbf{A}(\mathbf{P}\mathbf{y}) \, \mathbf{P}\mathbf{y} = \mathbf{P}^T \, \mathbf{b}(\mathbf{P}\mathbf{y}) \tag{19}$$

leading to the much smaller set of only N discrete equations

$$\tilde{\mathbf{A}}(\mathbf{y})\mathbf{y} = \tilde{\mathbf{b}}(\mathbf{y}). \tag{20}$$

The choice of the reduced basis is of great influence on the properties of the reduced system. If the modes are orthogonal with respect to the unreduced coupled coefficient matrix $\mathbf{A}$, the reduced system will be de-coupled (diagonal) and each equation could be integrated in time independently as long as the system behavior is linear. In case of nonlinear equations as for the fluid-structure problem at hand, the reduced system is fully coupled. The changing characteristics of the system before and after modal reduction are summarized in Figure 3.

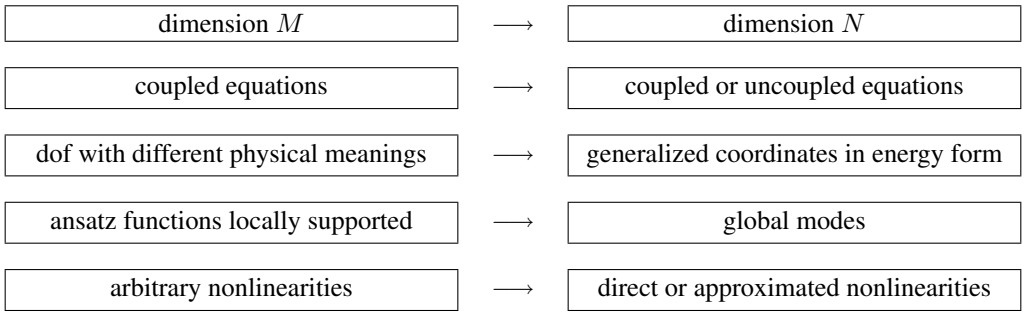


Figure 3: Comparison of characteristics of the discrete equations of the original (left) and reduced model (right).

3.2 Monolithic space-time modal reduction of coupled problems

With traditional modal reduction one discretizes the weak form of the model equations in space first and then performs the reduction step. The resulting system in terms of the vector of generalized coordinates is then evolved in time, see Clough and Penzien (1975). Using the proposed space-time finite element discretization of the fluid-structure problem, the reduced basis is defined in space-time and together with the generalized coordinates already covers time-dependent effects. This space-time view of modal reduction to fluid-structure interaction may incorporate changes of the fluid domain due to structural response in terms of mesh-motion techniques more easily.

The non-reduced monolithic space-time system of equations can be written for time slab n as

$$\mathbf{A}(\mathbf{x}^n, \mathbf{g}^n)\mathbf{x}^n = \mathbf{b}(\mathbf{x}^{n-1}, \mathbf{g}^n), \tag{21}$$

where $\mathbf{g}$ denotes dependency on the current geometry of the space-time mesh. Considering a block Gauss-Seidel iteration scheme to resolve mesh deformation and quadratic nonlinearities introduced by the convection term in the weak form of the Navier-Stokes equations, one obtains

$$\mathbf{A}_0\mathbf{x}^n + \mathbf{A}_1\mathbf{x}^n = \mathbf{b}_0 + \mathbf{B}_0\mathbf{x}^{n-1}. \tag{22}$$

Introducing equal reduced test and trial spaces $\mathcal{P}^h$ with the Bubnov-Galerkin approach in Eq. (18) together with a vector of generalized unknowns $\mathbf{y}$, the reduced discrete weak form transforms to

$$\delta\mathbf{y}^T \left(\mathbf{P}^T\mathbf{A}_0\mathbf{P}\ \mathbf{y}^n + \mathbf{P}^T\mathbf{A}_1\mathbf{P}\ \ \mathbf{y}^n - \mathbf{P}^T\mathbf{b}_0 - \mathbf{P}^T\mathbf{B}_0\mathbf{P}\ \mathbf{y}^{n-1}\right) = 0. \tag{23}$$

Each entry y_i in the vector of generalized coordinates $\mathbf{y}$ describes the participation factor of the associated global space-time mode $\mathbf{p}_i$ and therefore all influences to the combined solid and fluid state (velocities and pressure) simultaneously. Monolithic reduction modes inherently ensure strong coupling of fluid pressure and velocity as well as structural velocity. The matrices $\mathbf{A}_0 = \mathbf{A}_0(\mathbf{x}^{n-1})$, $\mathbf{A}_1 = \mathbf{A}_1(\mathbf{x}^n)$ and $\mathbf{B}_0 = \mathbf{B}_0(\mathbf{x}^{n-1})$ follow directly by projection of the weak form and are stored for computation. Reduction and global assembly are performed matrix-free on the element level. Thus, the total reduced fluid-structure equation system is $N \times N$ and can be solved nearly in real-time without large storage requirements.

3.3 Generation of the reduced space-time basis: Proper Orthogonal Decomposition (POD)

The choice of the reduced basis impairs the quality of the reduced solution to the fluid-structure interaction problem considerably. Therefore, the projection matrix $\mathbf{P}$ has to be filled with appropriate basis vectors, where a good method of determination strongly depends on the character of the underlying set of partial differential equations, involved nonlinearities, the reduction approach itself, and the type of excitation. Within the presented framework the focus is on the determination of basis vectors taking advantage of the monolithic space-time formulation of the problem and enabling herewith a uniform approach to modal reduction of fluid-structure interaction.

The method of Proper Orthogonal Decomposition (POD), which has been successfully applied to fluid and solid body dynamics, enables efficient extraction of coherent structures from experimental and simulation data. The method of snapshots in the POD, see Sirovich (1987), allows the generation of a reduced basis for nonlinear problems by the use of existing representative computational results. Therefore, the snapshot approach is well suited for quasi-steady and slightly transient response of a coupled system.

The non-orthogonal data snapshots of a representative time interval t_{SnPh} are collected in a matrix of dimension $M \times K$

$$\mathbf{R} = [\mathbf{r}_1 \;\; \mathbf{r}_2 \;\; \mathbf{r}_3 \;\; \ldots \;\; \mathbf{r}_K], \tag{24}$$

where $\mathbf{r}_i$ is a space-time solution vector of the monolithic fluid-structure system in Eq. (21) and $K \ll M$. The temporal distance between the snapshots has to be large enough to ensure a set of uncorrelated snapshots. For orthogonalization, the eigenproblem

$$(\frac{1}{K}\mathbf{R}^T\mathbf{R} - \lambda_i\mathbf{I})\mathbf{q}_i = \mathbf{0} \qquad i = 1, \ldots, K. \tag{25}$$

is solved and the resulting K eigenvectors are used to generate the set of basis vectors for model reduction

$$\mathbf{p}_i = \mathbf{R}\mathbf{q}_i. \tag{26}$$

A common mode selection criterium is the relative magnitude of the associated eigenvalues, determining the energy contained in that mode during the snapshot phase. In the examples presented in the next section, only a fraction of the K snapshot-generated projection vectors is used for efficiency. In contrast to other POD-based reduction methods, the mean flow is not to be subtracted from the snapshot input data, such that the sum of the eigenvalues is not zero but equal to the total energy contained in the mean flow. The POD therefore delivers the mean flow basis vector as the mode of largest energy.

4 Examples

4.1 Cavity flow with compliant bottom

An incompressible fluid in a quadratic cavity is set in motion by a harmonic excitation at the top as specified in Fig. 4. No-slip boundary conditions apply at the side walls and along the interface to the compliant structure at the bottom. The fluid develops a harmonic cavity flow with changing sign of the velocity components. With $\Omega = 10$ rad/s the excitation period, and therefore the period of the flow motion, is $T = \pi/5$ s. The fluid has density $\rho_f = 1\,\mathrm{kg/m^3}$ and dynamic viscosity $\mu = 1\,\mathrm{kg/m/s}$ leading to a Reynolds number of $Re = 1$. The elastic structure is of Youngs modulus $E = 1.5 \cdot 10^{-4}\ \mathrm{N/m^2}$, Poisson ratio $\nu = 0.35$ and density $\rho_s = 10^{-3}\ \mathrm{kg/m^3}$. Point A $(2/3, 0)$ is used to compare full and reduced solutions to the coupled fluid-structure system in terms of the motion of the membrane structure. The fluid domain is coarsely discretized by 12×12 elements. A matching fluid-structure interface mesh is obtained by using 12×2 solid elements for the membrane-like structure. The time step size $\Delta t = 0.01$ s is chosen according to the period T of the excitation.

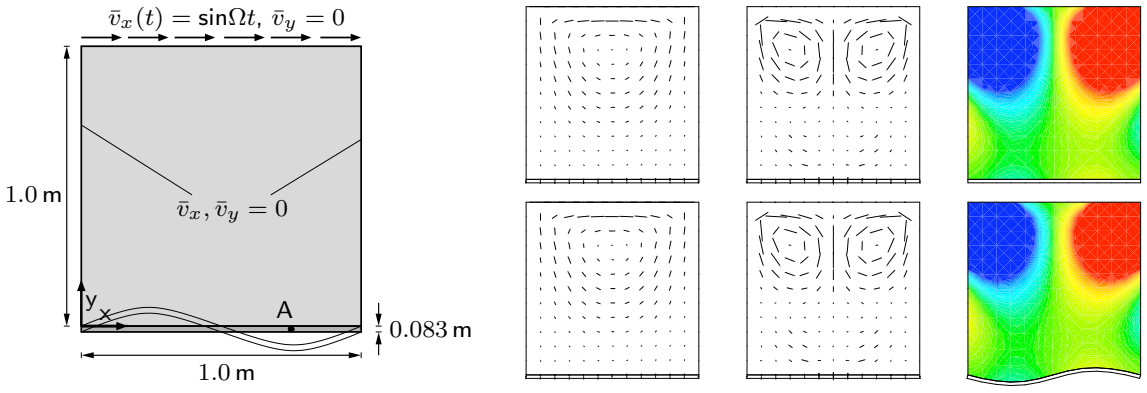


Figure 4: Driven cavity with compliant bottom: system definition, resultant reduced velocity and pressure fields.

In a time interval of $t_{SnPh} = 1.6$ s 10 snapshots are taken at each 16th time step and the POD is performed. The snapshot phase (SnPh) is identical to 2.5 times the period of excitation. The choice of a non-integer multiple

excludes unwanted special situations affecting the modal reduction in space-time. The mean flow, averaged over one period, is zero due to the symmetry of the system. All generalized coordinates of the reduced coupled system are free, since boundary conditions are incorporated by the set of reduced basis vectors. Inspection of the eigenvalue spectrum of the POD underlines, that a pair of modes is dominating but not the mean flow mode. Figure 4 shows at the right the velocity fields of the first and third space-time mode column-wise and the pressure field of the first mode is depicted in the third column. The upper row gives the t_n^+-state at the beginning of the space-time slab, while the bottom row shows the state at t_{n+1}^-. The membrane displacements are magnified for illustration. In the fluid domain and close to the membrane structure the interaction effect of fluid and structure is reflected by increased pressure differences compared to the situation with a fixed bottom.

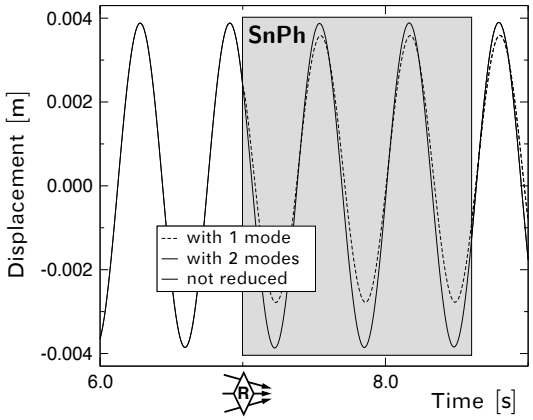


Figure 5: Driven cavity with compliant bottom: motion of point A, comparison of full and reduced computation.

Modal reduction is performed according to Eq. (23). Figure 5 shows a comparison of the results obtained for membrane vertical displacement at point A using one and two space-time modes. Using only one mode leads to an almost phase-correct but a non-symmetric diffusive reduced solution. If two space-time modes are used for the modal reduction, the reduced solution of just 2 degrees of freedom meets the non-reduced solution to the full system consisting of 5,766 unknowns.

The example shows that the strong coupling of both continua is well represented in the reduced model. While the structural displacements remain small and most of the energy is related to the fluid, the overall structural behavior is very good approximated. Without monolithic modes for the modal reduction this result is hardly to achieve.

4.2 Vortex-induced vibration of a plate

The following example discusses the application of the proposed reduction method to the fluid-structure benchmark situation proposed by Wall (1999). This coupled problem involves strong interactions of an incompressible fluid and a plate-like structure, where depending on initial tip displacements different and not always quasi-steady solutions are obtained. The reduction method must be able to project the complex system behavior onto the reduced model.

set	E	ν
(a)	$2.5 \cdot 10^7$ N/m^2	0.35
(b)	$2.5 \cdot 10^6$ N/m^2	0.35

Table 1: Parameter sets for the elastic plate structure.

A thin and slender structure is located in the wake of quadratic bluff body in such a way, that it is excited due to continuous vortex shedding. Figure 6 shows the geometrical system setup and its discretization by finite elements. The spatial mesh consists of 8,338 fluid elements and 136 structural elements which leads to a total of 52,766 equations of the full monolithic space-time discretization. Based on this discretization, a numerical reference solution is available in Walhorn (2002) and allows the determination of the quality of the reduced solution. The inflow velocity is of $\bar{v}_x = 31.5$ cm/s. With the dynamic viscosity $\mu = 1.82 \cdot 10^{-4}$ g/cm/s and a fluid density $\rho_f = 1.18 \cdot 10^{-3}$ g/cm^3 the Reynolds number of the flow problem is $Re = 204$. Two different parameter settings, see Table 1, are investigated for the elastic structure at plane stress and of constant density $\rho_s = 0.1$ g/cm^3.

The time step size $\Delta t = 0.005$ s is chosen according to the Strouhal number associated with the vortex shedding phenomenon.

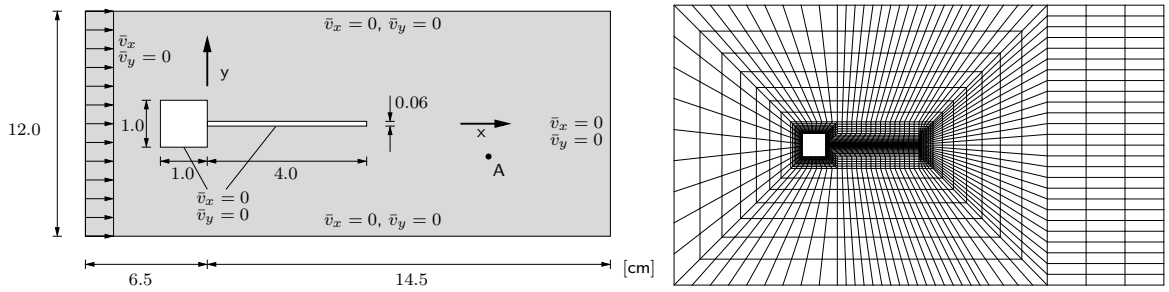


Figure 6: Vortex-induced vibration of plate: system definition and mesh of full discretization.

Parameter set (a). Due to the stiff structure in parameter set (a) small structural displacements and small mesh deformations indicate limited influence of nonlinear behavior in the system. Figure 7 shows the tip displacement in the time interval and the dominance of 2 frequencies in the oscillation. The interval of the snapshot phase is chosen $t_{SnPh} = 1.0$ s and every 10 time steps a total of 20 snapshots is taken for generation of the reduced basis, such that one period of the low-frequency range is covered. In Figure 8 the first three t_n^+-modes are shown in terms of the velocity state (left) and pressure state (right). The mean flow is not subtracted from the snapshots.

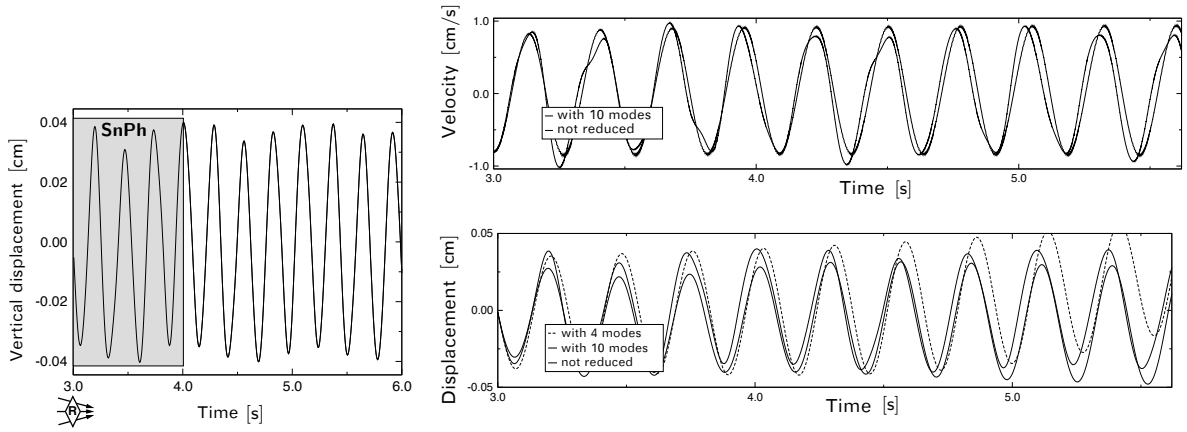


Figure 7: Vortex-induced vibration of plate (a): tip displacement and choice of snapshot phase / small deformations.

As in the previous example the method of reduction given by Eq. (23) is used with monolithic modes and reduced coupled system matrix. In the right of Fig. 7 the tip velocity of the not reduced and the reduced solution using 10 modes is compared. The amplitude for both computations is nearly equivalent and the reduced solution is governed by only a few lower frequencies. The reduction method is able to capture the phenomena of vortex shedding and plate vibration. The displacement solution underlines accumulation of errors in the reduced solution to the structural velocity state. The deviation can be lowered by increasing the number of space-time modes in the basis used for reduction.

Parameter set (b). The transient tip displacement is much more complex in this setting than for parameter set (a), see left in Figure 9. An analysis of the frequency spectrum shows that more than two frequencies dominate the structural response due to interaction with the fluid. Therefore, the POD snapshot phase is chosen larger with $t_{SnPh} = 2$ s and a total of 20 snapshots is taken every 20th time step. Figure 9 (middle) compares the tip velocity of the plate for the not reduced system to the reduced solution using the first 13 space-time modes. The reduction method is able to capture the system behavior until $t = 3.3$ s quite well, but afterwards deviates from the reference solution. The same holds for the horizontal velocity in point A given in Figure 9 (right). For the system governed by various dominant frequencies the complex behavior is not well projected to the reduced model by the chosen POD snapshot method and parameters due to higher-order nonlinearities introduced by large mesh deformations, and these issues are in the focus of ongoing research.

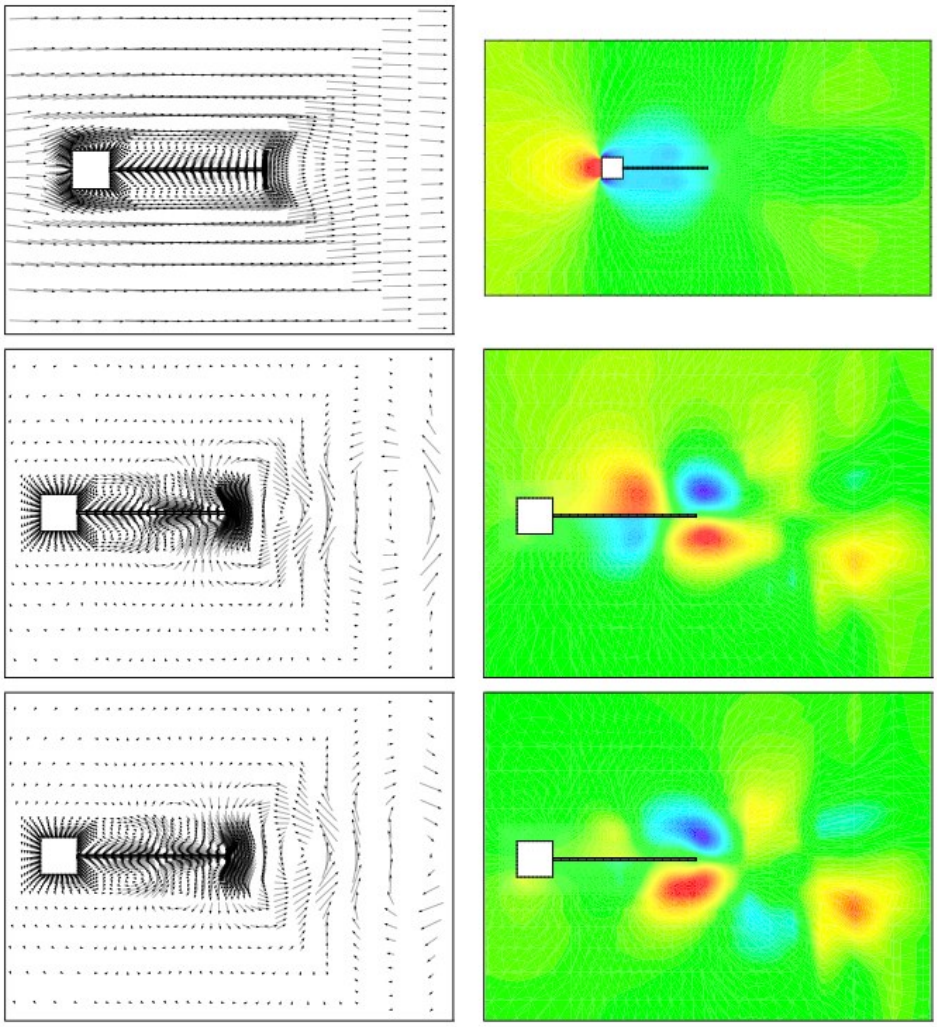


Figure 8: Vortex-induced vibration of plate (a): first 3 modes (velocity, pressure) of reduced model.

5 Conclusions

In this work a modal reduction method using space-time modes for strongly coupled fluid-structure interaction problems is investigated. Taking a verified and validated monolithic space-time finite element model as a basis, projection-based reduction is performed on all terms of the weak form of governing equations up to second order using monolithic space-time modes of a time slab. Evolution of the reduced solution in time is not necessary, since transient effects are already incorporated by the chosen space-time reduction basis. The snapshot method of the Proper Orthogonal Decomposition is used for generation of the reduction basis. For convection-dominated fluid dynamics pairwise conjugated space-time modes are generated, representing phase-shifted coherent structures that positively affect the projection of the convective terms in the weak form. The single mean flow mode is part of the POD-generated basis. Application of the reduction method to fluid-structure interaction problems involving vortex-shedding at small structural displacements shows good agreement with the full solution even if only a small number of modes are used for reduction.

References

Argyris, J.; Scharpf, D.: Finite elements in space and time. *Journal of Royal Aeronautical Society*, 73, (1969), 1041–1044.

Clough, R. W.; Penzien, J.: *Dynamics of Structures*. McGraw-Hill, New York (1975).

Hübner, B.: *Simultane Analyse von Bauwerk-Wind-Wechselwirkungen*. Dissertation, Technische Universität Braunschweig, Braunschweig (2003).

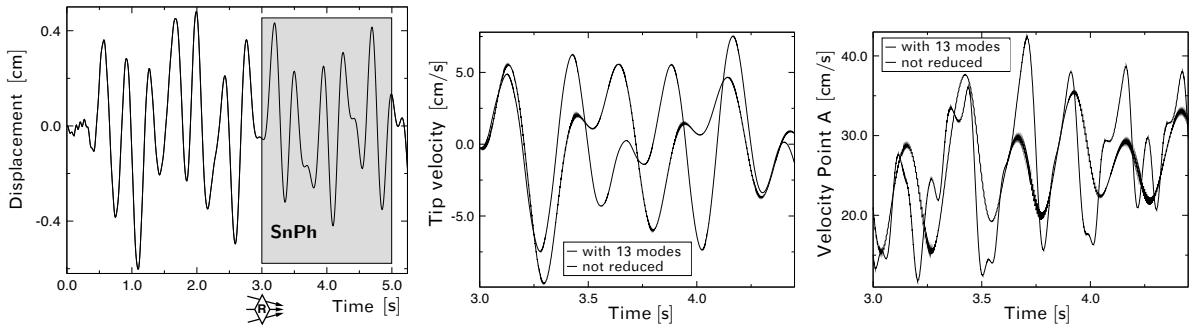


Figure 9: Vortex-induced vibration of plate (b): tip displacement and choice of snapshot phase / large deformations.

Hübner, B.; Dinkler, D.: A simultaneous solution procedure for strong interactions of generalized Newtonian fluids and viscoelastic solids at large strains. *International Journal for Numerical Methods in Engineering*, 64, (2005), 920–939.

Hübner, B.; Walhorn, E.; Dinkler, D.: A Monolithic Approach to Fluid-Sructure Interaction using Space-Time Finite Elements. *Computer Methods in Applied Mechanics and Engineering*, 193, 23–26, (2004), 2069–2086.

Hughes, T.; Hulbert, G.: Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66, 3, (1988), 339–363.

Knippers, J.; Harbord, R.: A Mixed Hybrid FE-Formulation for Solution of Elasto-Viscoplastic Problems. *Computational Mechanics*, 13, (1994), 231–240.

Kölke, A.: *Modellierung und Diskretisierung bewegter Diskontinuitäten in randgekoppelten Mehrfeldsystemen*. Phd thesis, Technische Universität Braunschweig (2005).

Masud, A.; Hughes, T.: A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. *Computer Methods in Applied Mechanics and Engineering*, 146, 1-2, (1997), 91–126.

Sirovich, L.: Turbulence and the dynamics of coherent structures, Part I-III: Coherent structures; symmetries and transformations; dynamics and scaling. *Quarterly of Applied Mathematics*, 45, (1987), 561–590.

Tezduyar, T. E.; Behr, M.; Liou, J.: A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces - The Deforming-Spatial-Domain/Space-Time Procedure: I. The Concept and the Preliminary Numerical Tests. *Computer Methods in Applied Mechanics and Engineering*, 94, (1992), 339–351.

Vehre, A.: *Ein Reduktionsverfahren für Fluid-Struktur-Wechselwirkungen mit Finiten Raum-Zeit-Elementen*. Ph.D. thesis, Technische Universität Braunschweig (2006).

Walhorn, E.: *Ein simultanes Berechnungsverfahren für Fluid-Struktur-Wechselwirkungen mit finiten Raum-Zeit-Elementen*. Dissertation, Technische Universität Braunschweig, Braunschweig (2002).

Walhorn, E.; Kölke, A.; Hübner, B.; Dinkler, D.: Fluid-structure coupling within a monolithic model involving free surface flows. *Computers and Structures*, 83, 25-26, (2005), 2100–2111.

Wall, W. A.: *Fluid-Struktur-Interaktion mit stabiliserten Finiten Elementen*. Dissertation, Universität Stuttgart (1999).

Zilian, A.; Legay, A.: The enriched space-time finite element method (EST) for simultaneous solution of fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 75, (2008), 305–334.