

# A coupling strategy and its software implementation for waves propagation and their impact on coast



Prospects



Christophe Kassiotis

SPH Meeting – Feb 9, 2010



# Introduction

Who I am ? What did I do ?...



Ecole des Ponts ParisTech  
TU-Braunschweig

ENS Cachan  
University of Manchester

EDF R&D

# Introduction

## Ph. D.



Ecole des Ponts ParisTech  
TU-Braunshweig

ENS Cachan  
University of Manchester

EDF R&D

# Introduction

## Now



Ecole des Ponts ParisTech  
TU-Braunschweig

ENS Cachan  
University of Manchester

EDF R&D



# Introduction

## Problem description

Waves modeling from open sea to coast:

- ↪ Various physical models and associated software
- ↪ Countless applications
- ↪ Among important issues: extreme waves or tsunami impacts



# Introduction

## Problem description

Waves modeling from open sea to coast:

- ↪ Various physical models and associated software
- ↪ Countless applications
- ↪ Among important issues: extreme waves or tsunami impacts

Tsunami modeling:

- ↪ Generation
- ↪ Propagation
- ↪ Run-up

Coupling key issues

- ↪ Amplitude of the flood
- ↪ Resistance of buildings

Source: CMLA-Cachan [Dutykh,

09]



# Introduction

## Problem description

Waves modeling from open sea to coast:

- ↪ Various physical models and associated software
- ↪ Countless applications
- ↪ Among important issues: extreme waves or tsunami impacts

Tsunami modeling:

- ↪ Generation
- ↪ Propagation
- ↪ Run-up

Coupling key issues

- ↪ Amplitude of the flood
- ↪ Resistance of buildings

[Kassiotis, 07]



# Introduction

## Problem description

Waves modeling from open sea to coast:

- ↪ Various physical models and associated software
- ↪ Countless applications
- ↪ Among important issues: extreme waves or tsunami impacts

Tsunami modeling:

- ↪ Generation
- ↪ Propagation
- ↪ Run-up

Coupling key issues

- ↪ Amplitude of the flood
- ↪ Resistance of buildings

FBI, American Samoa, Sep. 09





# Introduction

## Problem description

Waves modeling from open sea to coast:

- ↪ Various physical models and associated software
- ↪ Countless applications
- ↪ Among important issues: extreme waves or tsunami impacts

Tsunami modeling:

- ↪ Generation
- ↪ Propagation
- ↪ Run-up

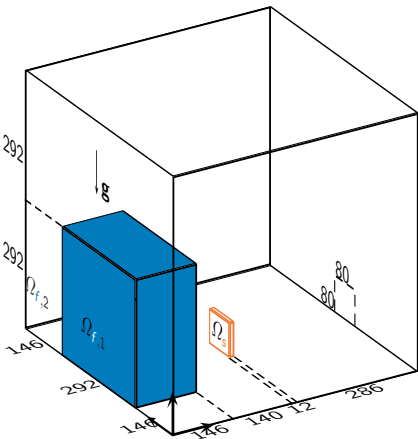
Coupling key issues

- ↪ Amplitude of the flood
- ↪ Resistance of buildings

FBI, American Samoa, Sep. 09

# Introduction

Ph. D. results: fluid structure interaction



- Partitioned strategy
- Strong coupling
- Coupling FEM and FVM
- Free-surface flow VOF

Material properties

- Neo-Hookean structure  
 $E_s = 1 \times 10^6 Pa$ ,  $\nu_s = 0$ ,  
 $\rho_s = 2500 kg \cdot m^{-3}$ .
- Two-phase flow  $\rho_{f,1} = 1000 kg/m^3$ ,  
 $\nu_{f,1} = 1 \times 10^6 m/s$ ,  $\rho_{f,2} = 1 kg/m^3$ ,  
 $\nu_{f,2} = 1 \times 10^5 m/s$ .



# Introduction

Ph. D. results: three-dimensional dam-break problem

# Introduction

## Waves model (Boussinesq, Saint-Venant)

- ↪ Unable to represent complex free surface (multi-connected domains)
- ↪ Can represent sloshing with damping [Benoit 02, Yu 99]
- ↪ Studies shows the necessity of more physical models [Duthyk 10]

## Sloshing representation (VOF, SPH)

- ↪ Waves damping (can be handled by *ad-hoc* treatment)
- ↪ Computational coast (3D computations un-reachable)

Coupling is a natural choice



# Outline

- 1 Sub-problems
  - Waves propagation
  - Waves sloshing on coast
- 2 Coupling algorithm
- 3 Software implementation – Component architecture

## Some existing software

	name	licence	language	2d	3d	val
waves	BSQ_V2P3 (EDF R&D)		Fortran	✓		✓
	funwave	GPL	Fortran	✓	✓	✓
	shallowWaterFoam	GPL	C++	✓	✓	
SPH	Spartacus		Fortran	✓	✓	✓
	SPHysics	GPL	Fortran	✓	✓	✓



# Wave propagation

## Boussinesq model

BSQ\_V2P3 implements:

### Physical description

- wave height  $\eta$
- horizontal velocity  $u$
- reference height  $z_\alpha$

### Finite difference discretization ( $\mathcal{O}(\Delta x^4)$ )

### Explicit Predictor ( $\mathcal{O}(\Delta t^3)$ ) – Corrector ( $\mathcal{O}(\Delta t^4)$ ) time integration scheme

### Stabilising filters

### Waves damping (sloshing)



# Wave propagation

## Boussinesq model – results





# Wave sloshing

- Using SPH
- A lot of open questions ... (for me)



# Outline

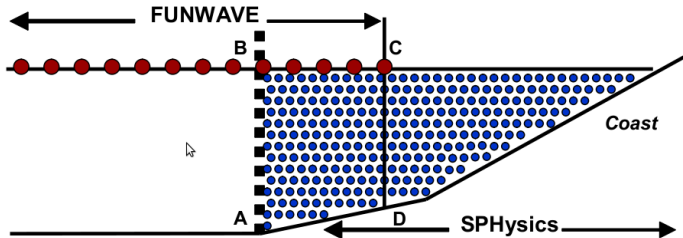
- 1 Sub-problems
  - Waves propagation
  - Waves sloshing on coast
- 2 Coupling algorithm
- 3 Software implementation – Component architecture

# Coupling strategy

[Narayanaswamy 09]

Proposed coupling algorithm:

- Explicit and staggered in time
- Overlapping regions in space



Figure

from [Narayanaswamy *et al* 10]

# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:  $t = n\Delta t_{\text{Bsq}}$
- 3: impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4: solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5: get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6: solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7: **for**  $k = 1 \dots k_{\max}$  **do**
- 8:  $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9: interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10: solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11: **end for**
- 12: get wave height and velocity at the reference height
- 13: **end for**

# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:  $t = n\Delta t_{\text{Bsq}}$
- 3: impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4: solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5: get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6: solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7: **for**  $k = 1 \dots k_{\max}$  **do**
- 8:  $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9: interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10: solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11: **end for**
- 12: get wave height and velocity at the reference height
- 13: **end for**

# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:  $t = n\Delta t_{\text{Bsq}}$
- 3: impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4: solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5: get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6: solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7: **for**  $k = 1 \dots k_{\max}$  **do**
- 8:  $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9: interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10: solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11: **end for**
- 12: get wave height and velocity at the reference height
- 13: **end for**

# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:  $t = n\Delta t_{\text{Bsq}}$
- 3: impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4: solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5: get velocity on the SPH boundary:  $u(x_i, z_i)$   
$$= u_\alpha(x_i) + \partial_x z_\alpha \partial_x h u_\alpha + (z_\alpha - z_i) \partial_x^2 h u_\alpha + [\dots]$$

- 6: solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7: **for**  $k = 1 \dots k_{\max}$  **do**
- 8:  $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9: interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10: solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11: **end for**

# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:  $t = n\Delta t_{\text{Bsq}}$
- 3: impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4: solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5: get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6: solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7: **for**  $k = 1 \dots k_{\max}$  **do**
- 8:  $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9: interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10: solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11: **end for**
- 12: get wave height and velocity at the reference height
- 13: **end for**



# Coupling strategy

[Narayanaswamy 09]

- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:    $t = n\Delta t_{\text{Bsq}}$
- 3:   impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4:   solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5:   get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6:   solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7:   **for**  $k = 1 \dots k_{\max}$  **do**
- 8:      $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9:     interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}}u_i^k$
- 10:    solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11:   **end for**
- 12:   get wave height and velocity at the reference height
- 13: **end for**

# Coupling strategy

[Narayanaswamy 09]

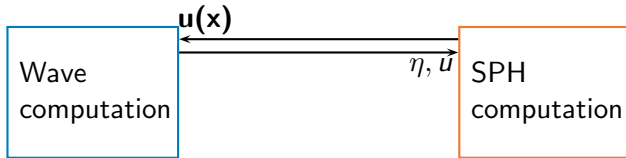
- 1: **for**  $n = 1 \dots n_{\max}$  **do**
- 2:    $t = n\Delta t_{\text{Bsq}}$
- 3:   impose wave height and velocity ( $\eta_b, v_{\alpha,b}$ )
- 4:   solve wave problem with  $\Delta t_{\text{Bsq}}$
- 5:   get velocity on the SPH boundary:  $u(x_i, z_i)$
- 6:   solve SPH problem from  $t - \frac{\Delta t_{\text{Bsq}}}{2}$  to  $t + \frac{\Delta t_{\text{Bsq}}}{2}$
- 7:   **for**  $k = 1 \dots k_{\max}$  **do**
- 8:      $t = (n - \frac{1}{2})\Delta t_{\text{Bsq}} + k\Delta t_{\text{SPH}}$
- 9:     interpolate  $u_i^k$  and impose  $x_i^{k+1} = x_i^k + \Delta t_{\text{SPH}} u_i^k$
- 10:    solve SPH problem with time step  $\Delta t_{\text{SPH}}$
- 11:   **end for**
- 12:   get wave height and velocity at the reference height
- 13: **end for**



# Outline

- 1 Sub-problems
  - Waves propagation
  - Waves sloshing on coast
- 2 Coupling algorithm
- 3 Software implementation – Component architecture

# Programming context for partitioned solution

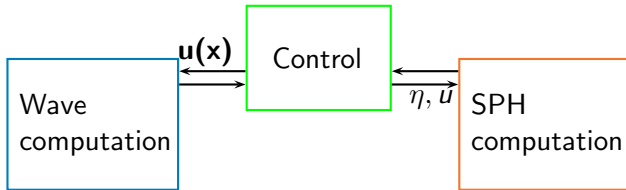


## Software implementation

- Data exchange between wave propagation and near coast
- Implementation of a master code
- Non matching meshes/time handled by the Interpolator (?)
- Re-using existing waves and SPH codes

Minimum requirement: a communication protocol

# Programming context for partitioned solution

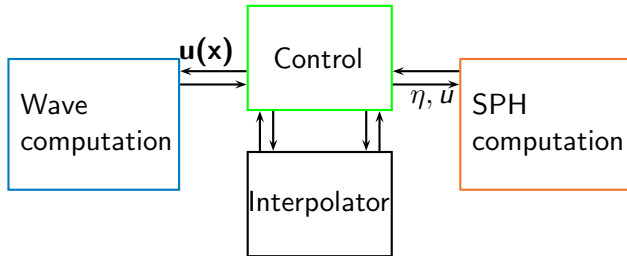


## Software implementation

- 🌊 Data exchange between wave propagation and near coast
- 🌊 Implementation of a master code
- 🌊 Non matching meshes/time handled by the Interpolator (?)
- 🌊 Re-using existing waves and SPH codes

Minimum requirement: a communication protocol

# Programming context for partitioned solution

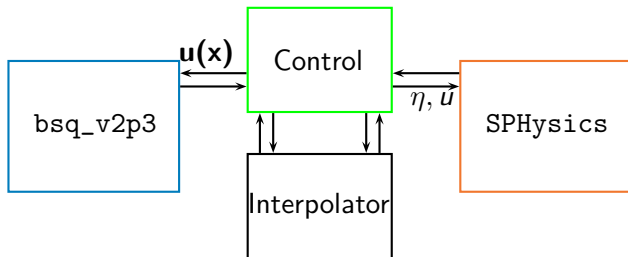


## Software implementation

- Data exchange between wave propagation and near coast
- Implementation of a master code
- Non matching meshes/time handled by the Interpolator (?)
- Re-using existing waves and SPH codes

Minimum requirement: a communication protocol

## Programming context for partitioned solution

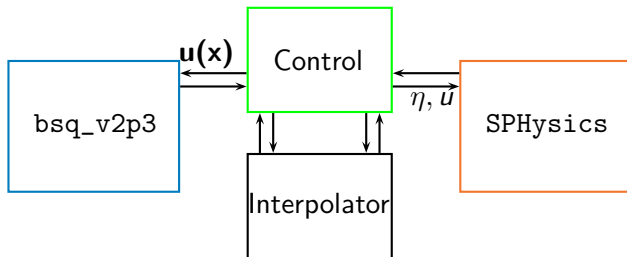


### Software implementation

- Data exchange between wave propagation and near coast
- Implementation of a master code
- Non matching meshes/time handled by the Interpolator (?)
- Re-using existing waves and SPH codes

Minimum requirement: a communication protocol

## Programming context for partitioned solution



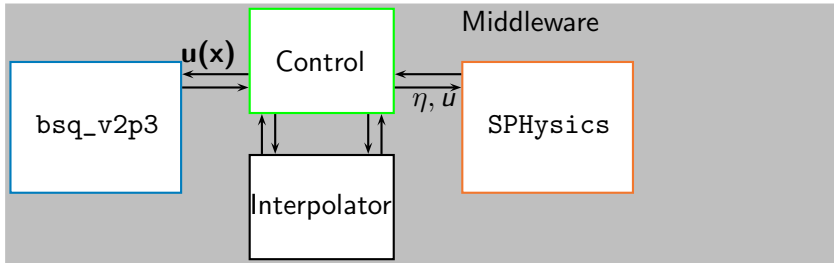
### Software implementation

- Data exchange between wave propagation and near coast
- Implementation of a master code
- Non matching meshes/time handled by the Interpolator (?)
- Re-using existing waves and SPH codes

Minimum requirement: a communication protocol



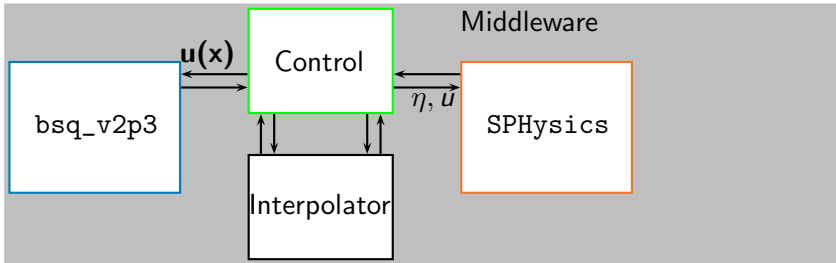
# Programming context for partitioned solution



Middleware – Software component technology

- ~ “Between” software and hardware
- ~ Computer science [Mac Ilroy 68, Szyperski & Meeserschmitt 98]
- ~ Each software: a **component**
- ~ Generalization of OOP to software: encapsulated / interface
- ~ Middleware in charge of communication and data types

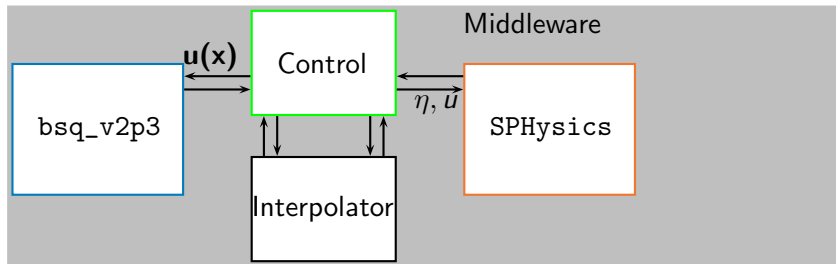
# Programming context for partitioned solution



Middleware – for scientific computing

- Available middleware: Corba, Java-RMI, MS.net ...
- Communication Template Library (CTL): C++ [Niekamp, 02]
- Scientific computing: requires good performances [Niekamp, 05]
- Salomé platform (ÉDF R&D)
- Software development made by non-programmers

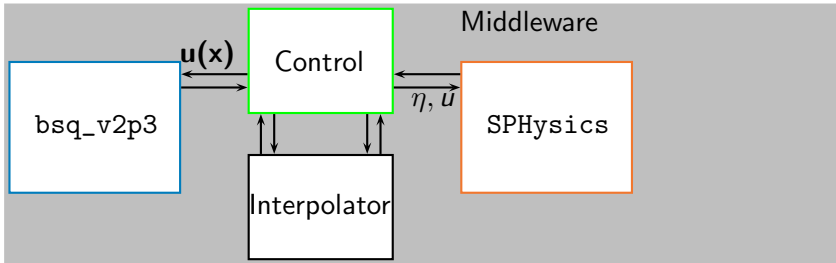
## Programming context for partitioned solution



Middleware – for scientific computing

- Available middleware: Corba, Java-RMI, MS.net ...
- Communication Template Library (CTL): C++ [Niekamp, 02]
- Scientific computing: requires good performances [Niekamp, 05]
- Salomé platform (ÉDF R&D)
- Software development made by non-programmers

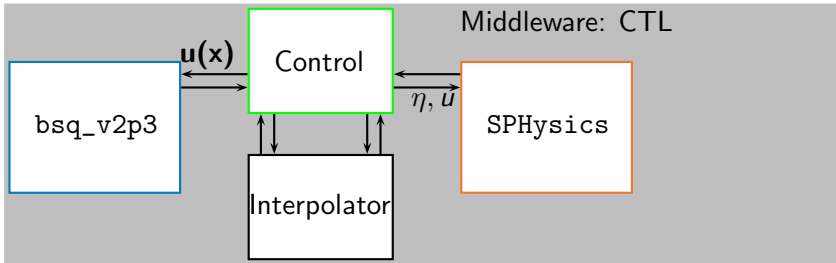
# Programming context for partitioned solution



Middleware – for scientific computing

- Available middleware: Corba, Java-RMI, MS.net ...
- Communication Template Library (CTL): C++ [Niekamp, 02]
- Scientific computing: requires good performances [Niekamp, 05]
- Salomé platform (ÉDF R&D)
- Software development made by non-programmers

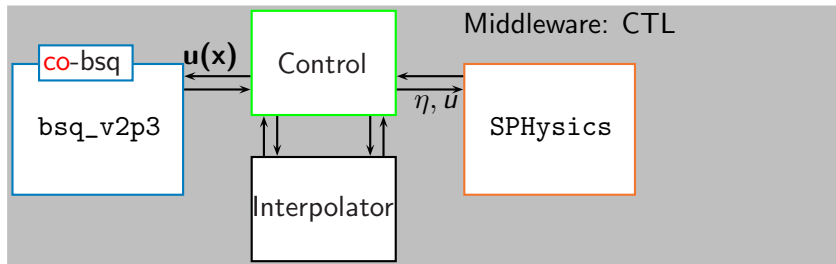
# Programming context for partitioned solution



Middleware – for scientific computing

- Available middleware: Corba, Java-RMI, MS.net ...
- Communication Template Library (CTL): C++ [Niekamp, 02]
- Scientific computing: requires good performances [Niekamp, 05]
- Salomé platform (ÉDF R&D)
- Software development made by non-programmers

# Components implementation and use



Wave component:

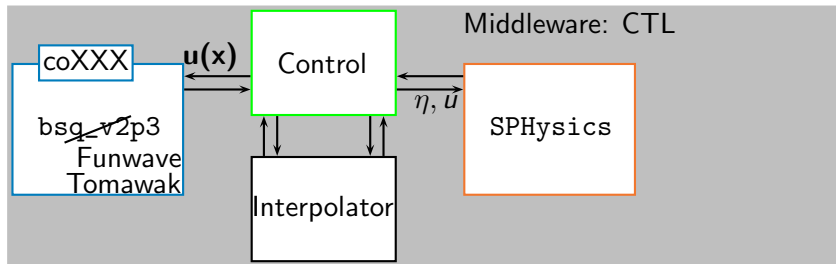
Interface definition `SimuWave.ci` (Genericity)

Methods declaration

```
#define CTL_Method3 void, get, ( const string /*fieldName*/,  
    array<scalar1> /*nodes*/, array<scalar1> /*field*/ ), 3
```

Methods implementation in Fortran

# Components implementation and use



Wave component:

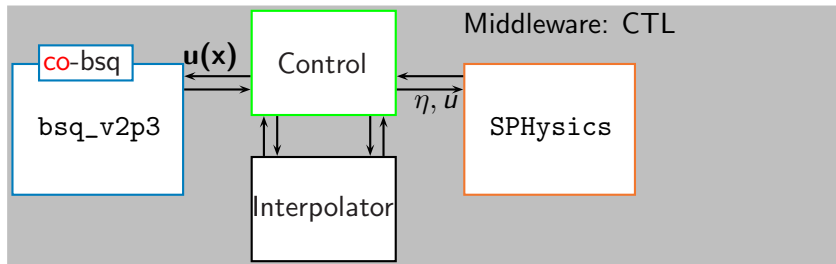
Interface definition `SimuWave.ci` (Genericity)

Methods declaration

```
#define CTL_Method3 void, get, ( const string /*fieldName*/,  
    array<scalar1> /*nodes*/, array<scalar1> /*field*/ ), 3
```

Methods implementation in Fortran

# Components implementation and use



Wave component:

➤ Interface definition `SimuWave.ci` (Genericity)

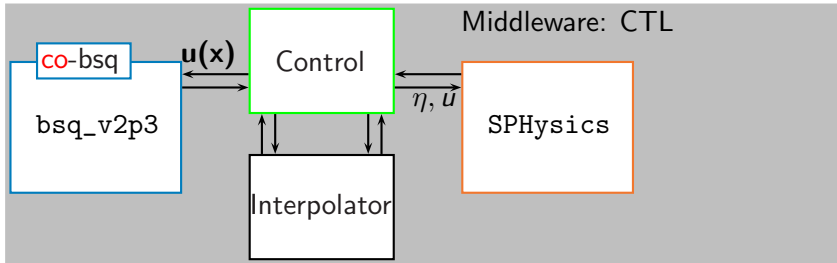
➤ Methods declaration

```
#define CTL_Method3 void, get, ( const string /*fieldName*/,  
    array<scalar1> /*nodes*/, array<scalar1> /*field*/ ), 3
```

➤ Methods implementation in Fortran



# Components implementation and use



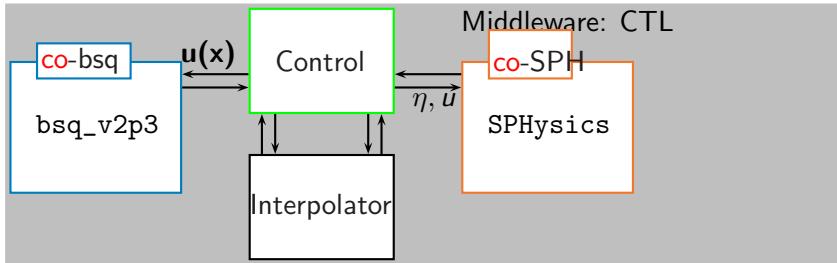
Wave component:

🌊 **Compilation gives:**

- 🌊 A library: call like a `lib`, `thread` (asynchronous calls)
- 🌊 An executable: remote call with `tcp`, `pipe`, `MPI`...

🌊 **Use:** *Multiscale* [Hautefeuille 09], *EFEM* [Benkemoun 09], *Stochastic* [Krosche 09], *Thermomechanics* [Kassiotis 06], *Mass transfer* [De Sa 08], *FSI* [Kassiotis 09] ...

# Components implementation and use

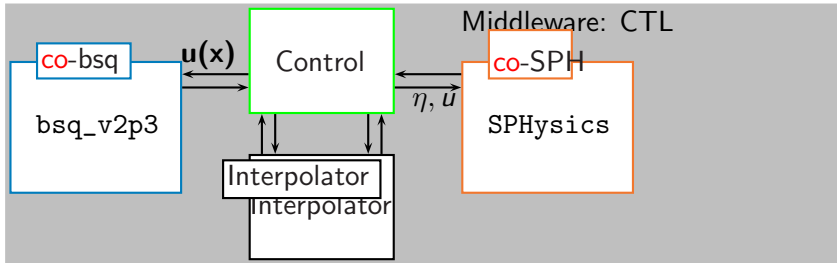


SPH component:

- Interface definition: derived from existing `CFDsimu.ci`:
- Implemented `foam` [Krosche 07, Kassiotis 09] in C++
- Methods declaration

```
#define CTL_Method2 void, get,  
    ( const string /*name*/, array<real8> /*v*/ ) const, 2
```

# Components implementation and use



Interpolation component: Interpolator [Jürgens 09]

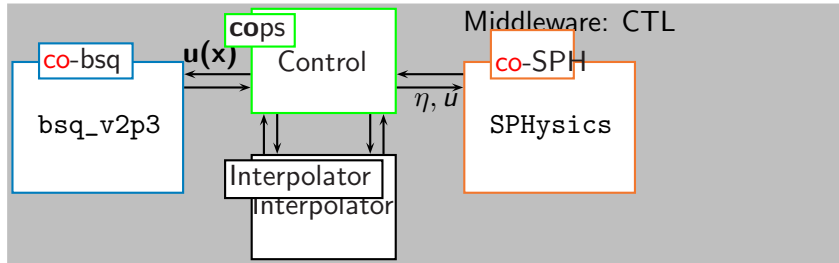
- ~ C++ component

- ~ Interpolation with radial basis functions [Beckert & Wendland 01]

- ~ Full matrices

- ~ Solve: coupled with the Lapack library

## Components implementation and use



COupling COmponents by a Partitioned Strategy: **cops**

- ~ Coupling components as templates
- ~ Explicit coupling: collocated and non-collocated
- ~ Implicit coupling: BGS
- ~ Predictors (order 0–2), fixed and dynamic Aitken's relaxation

# Conclusions and Outlooks

## Conclusion:

- Generic implementation re-using existing code
- Efficiency using CTL
- Simple implementation and cheap computation outside existing codes

## Outlooks:

- Use wave solver as BC in SPH computation

## Open questions:

- Which solver for the waves ? Saint-Venant ? Boussinesq ?
- Which size for overlapping area ?
- Which BC : input for SPH ? output in wave solver ?



## Conclusions and Outlooks

Thank you for attention

Questions ? Remarks ? Help for SPH ?