Basic plotting and matrix operations (Octave)

June 21, 2017

1 Table of Contents

2 Quick demonstration of GNU Octave

In a Jupyter Notebook

- This Jupyter Notebook was written by Lilian Besson.
- It uses the unofficial Octave kernel for Jupyter by [@Calysto](https://github.com/calysto/).

2.1 Some basic plotting

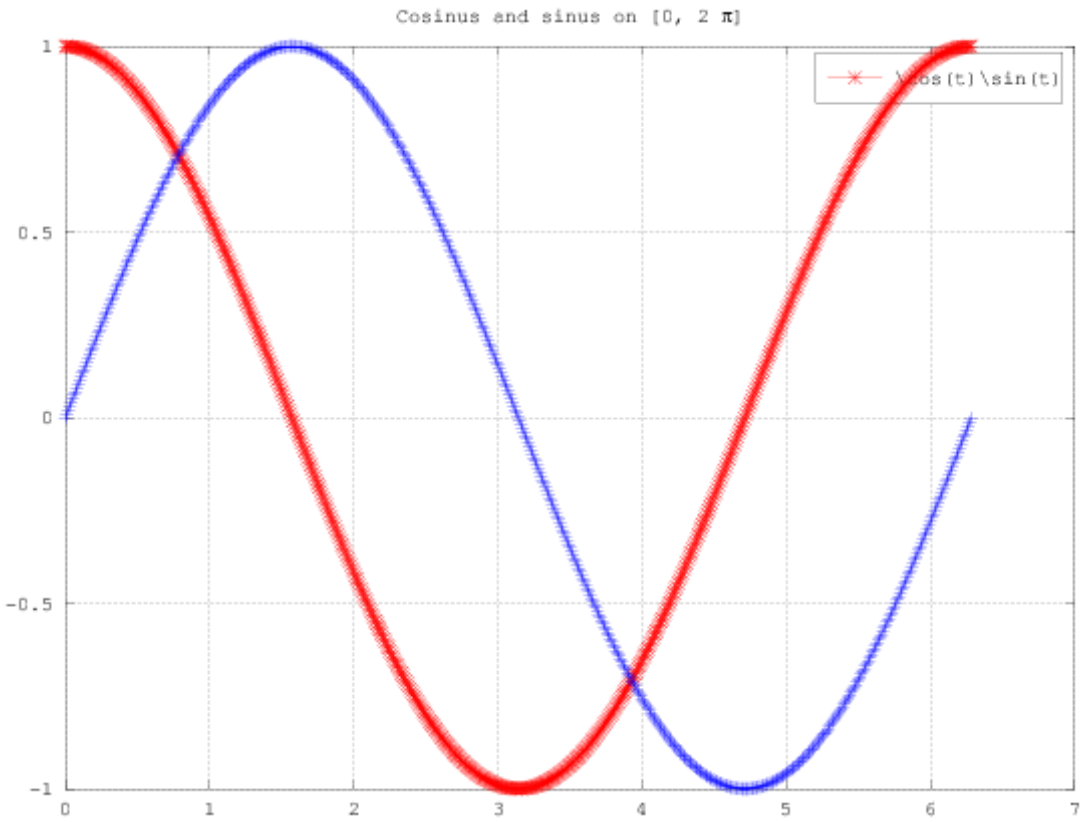
```
In [1]: % Some variables
        N = 100
        disp(['Number of values N =', num2str(N)])
        h = 1 / N
        disp(['Step h = ', num2str(h)])
```

```
N =  100
Number of values N =100
h =  0.010000
Step h = 0.01
```

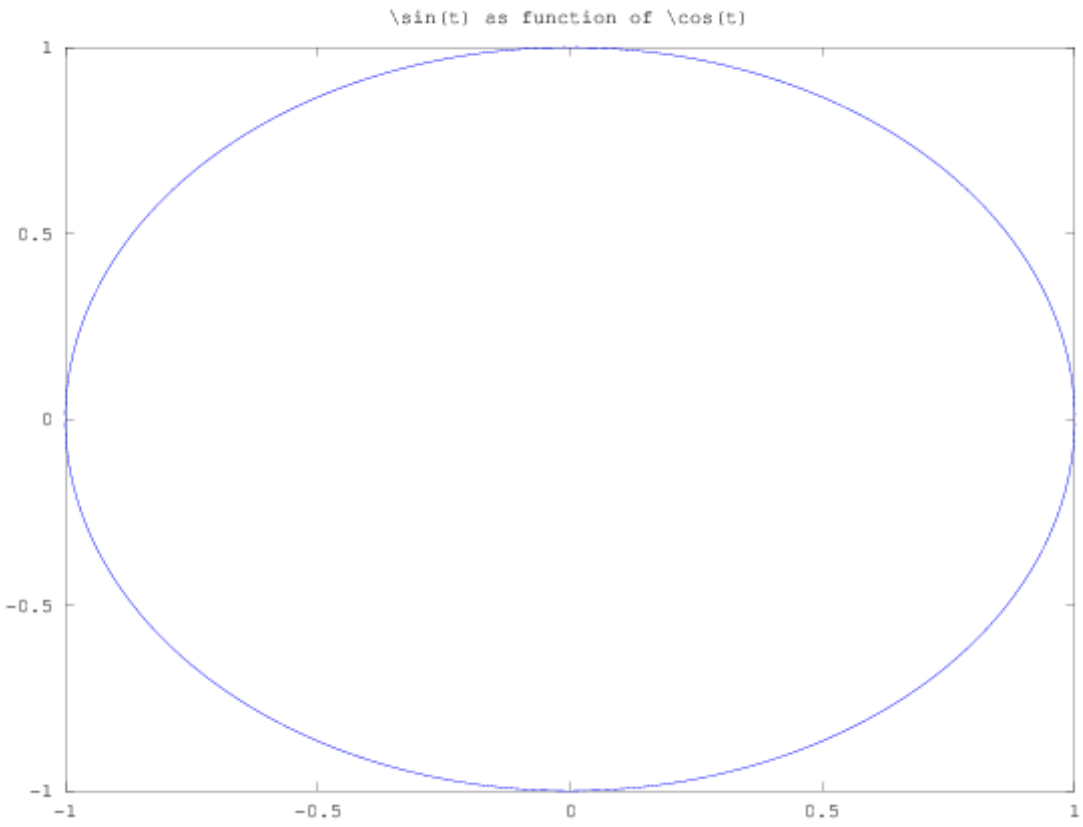
```
In [2]: % Some arrays
        t = 0 : h : 2*pi;
        x = cos(t);
        y = sin(t);
        length(t)
        length(x)
        length(y)
```

```
ans =  629
ans =  629
ans =  629
```

In [3]:
```
fig = figure();
plot(t, x, 'r*-')
grid on
hold on
plot(t, y, 'b+-')
legend(['\cos(t)', '\sin(t)'])
title('Cosinus and sinus on [0, 2 \pi]')
% whitebg(fig);
```



In [4]:
```
fig = figure();
grid on
plot(x, y)
title('\sin(t) as function of \cos(t)')
% whitebg(fig);
```



2.2 Some basic linear algebra

```
In [5]: a = [[1 0 1]; [0 1 1]; [1 1 0]]
```

```
a =

   1   0   1
   0   1   1
   1   1   0
```

```
In [6]: a'
```

```
ans =

   1   0   1
   0   1   1
   1   1   0
```

```
In [7]: b = 1 + a^5

b =

   12   11   12
   11   12   12
   12   12   11
```

eig gives the eigen values:

```
In [8]: l_a = eig(a)
        l_b = eig(b)

l_a =

  -1.00000
   1.00000
   2.00000

l_b =

   -1.00000
    1.00000
   35.00000
```

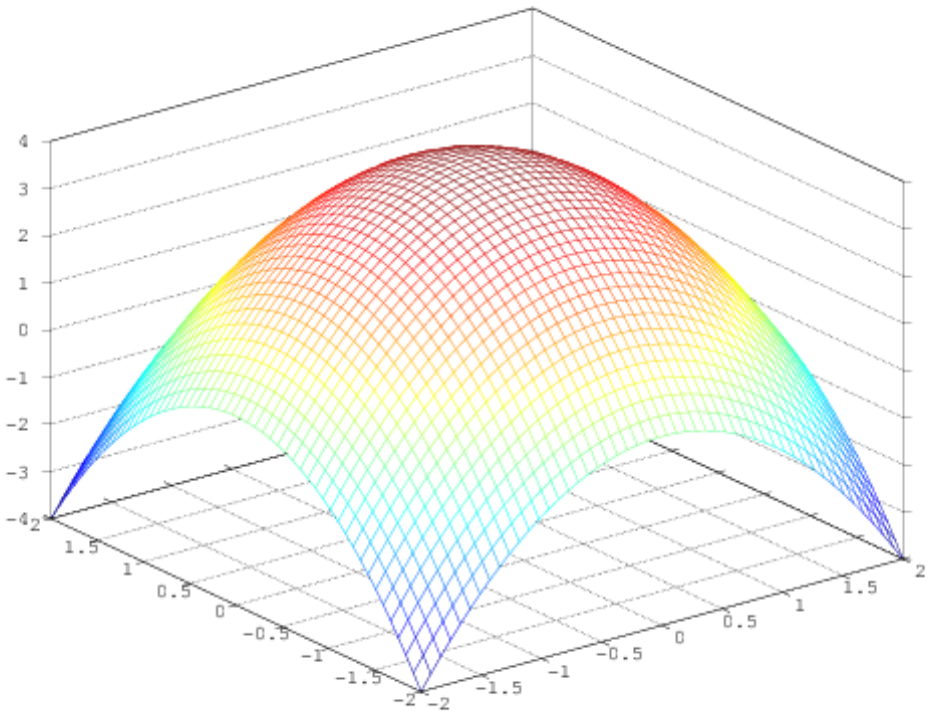
```
In [9]: [Ua, Sa, Va] = svd(a)
        [Ub, Sb, Vb] = svd(b)

Ua =

  -5.7735e-01   4.0825e-01  -7.0711e-01
  -5.7735e-01  -8.1650e-01  -7.8505e-17
  -5.7735e-01   4.0825e-01   7.0711e-01

Sa =

Diagonal Matrix

   2.00000         0         0
         0   1.00000         0
         0         0   1.00000

Va =

  -0.57735   0.81650  -0.00000
```

```
  -0.57735  -0.40825   0.70711
  -0.57735  -0.40825  -0.70711

Ub =

  -5.7735e-01   4.0825e-01  -7.0711e-01
  -5.7735e-01  -8.1650e-01  -1.6098e-15
  -5.7735e-01   4.0825e-01   7.0711e-01

Sb =

Diagonal Matrix

   35.00000          0          0
          0    1.00000          0
          0          0    1.00000

Vb =

  -0.57735   0.81650  -0.00000
  -0.57735  -0.40825   0.70711
  -0.57735  -0.40825  -0.70711
```

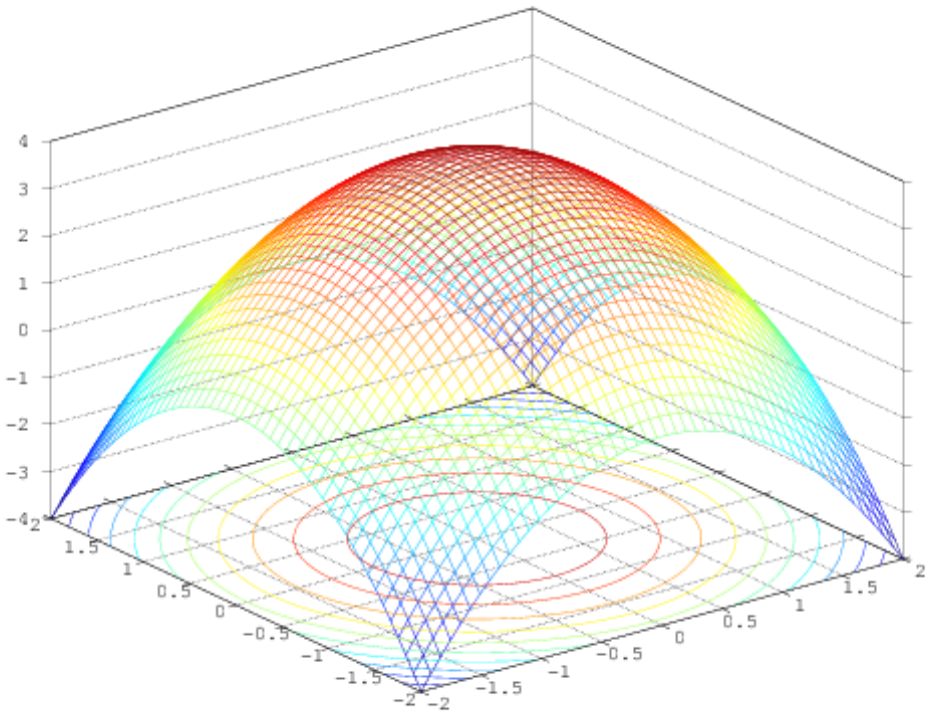
2.3 3D plotting

Some basic 3D plotting:

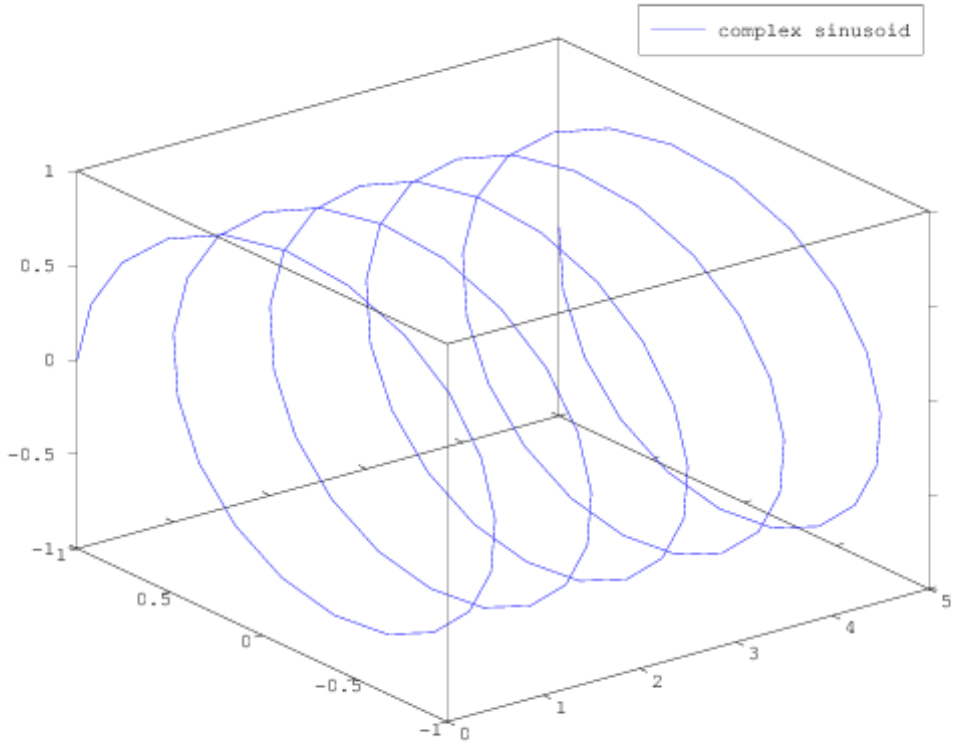
```
In [10]: x = linspace(-2, 2, 50);
         y = linspace(-2, 2, 50);
         [xx,yy] = meshgrid(x, y);
         mesh(xx, yy, 4 - (xx.^2 + yy.^2))
```



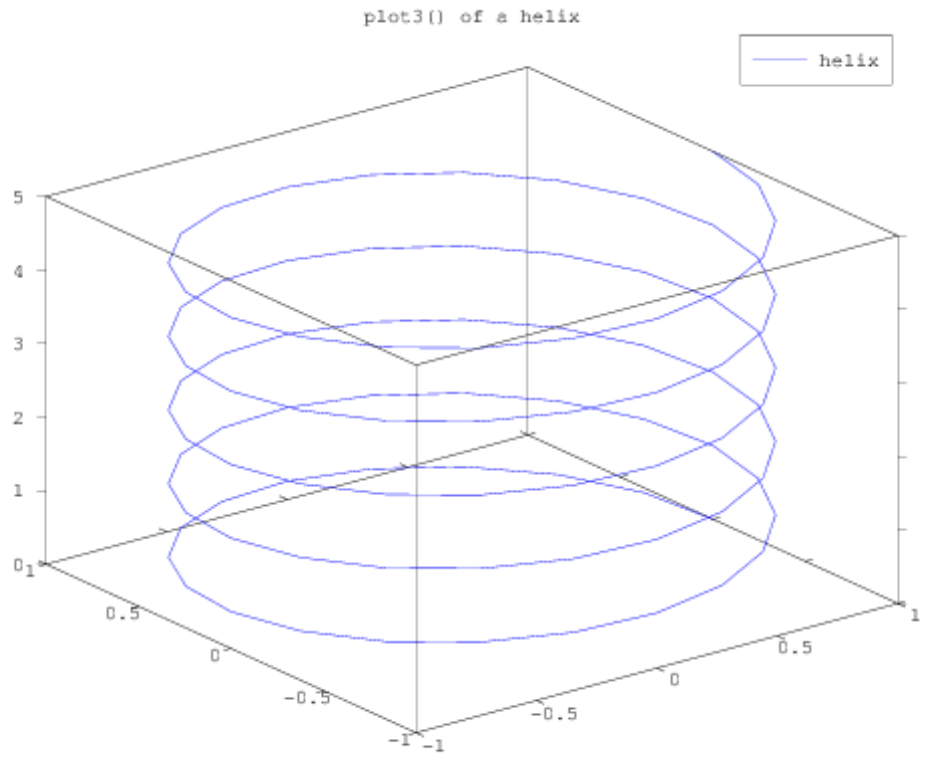

```
In [11]: x = linspace(-2, 2, 50);
         y = linspace(-2, 2, 50);
         [xx,yy] = meshgrid(x, y);
         meshc(xx, yy, 4 - (xx.^2 + yy.^2))
```



In [13]:
```
# From https://octave.sourceforge.io/octave/function/plot3.html
z = [0:0.05:5];
plot3(cos (2*pi*z), sin (2*pi*z), z, ";helix;");
plot3(z, exp (2i*pi*z), ";complex sinusoid;");
```




```
In [14]: clf;
         z = [0:0.05:5];
         plot3 (cos (2*pi*z), sin (2*pi*z), z);
         legend ("helix");
         title ("plot3() of a helix");
```



In [15]:
```
clf;
colormap ("default");
[X, Y] = meshgrid (linspace (-3, 3, 40));
Z = sqrt (abs (X .* Y)) ./ (1 + X.^2 + Y.^2);
meshc (X, Y, Z);
title ("meshc() combines mesh/contour plots");
```