

A mixed Matlab-Python example

This example demonstrates a cooperative effort where Matlab is used to do the analytic computations while Python is used to plot the data.

The example chosen here is to find and plot the solution to the boundary value problem defined by

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} + 10y = 0 \quad \text{with } y(0) = 3, y'(0) = 0$$

This example requires two passes, once for Matlab and once for Python (and in that order). This example can be run using

```
matlateg.sh -x -i mixed
pylateg.sh  -x -i mixed
pdflatex      mixed
```

Note that the last pair of commands could also be combined as `pylateg.sh -i mixed`.

The Matlab code

Here Matlab is used to first find the general solution of the differential equation. The boundary conditions are then imposed and finally a uniform sampling of the solution is written to a file for later use by Python and Matplotlib.

```
syms x y(x)
% a second order ode
ode = diff(y(x), x, x) + 2*diff(y(x),x) + 10*y(x) == 0; % mat (ans.101,ode)
ddx = diff(y,x);

% find the general solution
sol = dsolve(ode); % mat (ans.102,sol)

% set initial conditions
ics = [y(0) == 3, ddx(0) == 0];
tmp = ics'; % mat (ans.103,tmp)

% find the particular solution
sol = dsolve(ode,ics); % mat (ans.104,sol)
ddx = diff(sol,x); % mat (ans.105,ddx)

xvals = linspace (0.0,2.0*pi,300);
f = vpa(subs(sol,x,xvals),10);
df = vpa(subs(ddx,x,xvals),10);

dlmwrite ('mixed.txt',[xvals;f;df]','delimiter',' ','precision','% .8e');
```

The general solution of the differential equation is

$$y(x) = C_1 \cos(3x) e^{-x} - C_2 \sin(3x) e^{-x}$$

while the particular solution satisfying the boundary conditions is given by

$$y(x) = 3 \cos(3x) e^{-x} + \sin(3x) e^{-x}$$

The Python code

This is a straightforward use of Matplotlib to plot two functions. The code reads the datafile created previously by Matlab and then calls Matplotlib to plot that data.

```
import numpy as np
import matplotlib.pyplot as plt

plt.matplotlib.rc('text', usetex = True)
plt.matplotlib.rc('grid', linestyle = 'dotted')
plt.matplotlib.rc('figure', figsize = (5.5,4.1)) # (width,height) inches

x, y, dy = np.loadtxt('mixed.txt', unpack=True)

plt.plot(x,y)
plt.plot(x,dy)

plt.xlim(0.0,4.0)

plt.legend(('y(x)', 'dy(x)/dx'), loc = 0)
plt.xlabel('$x$')
plt.ylabel('$y(x), \> dy/dx$')
plt.grid(True)
plt.tight_layout(0.5)

plt.savefig('mixed_fig.pdf')
```

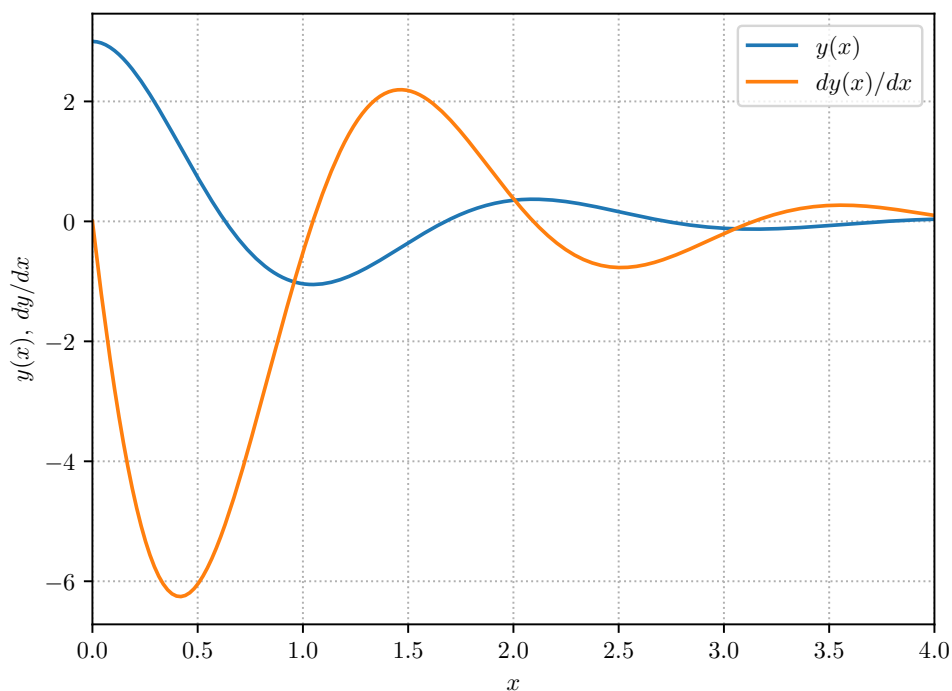


Figure 1: The function and its derivative.