Elementary maths

```
syms a b c x y z
foo = expand((a+b)^3);                                  % mat (ans.101,foo)
foo = factor(-2*x+2*x+a*x-x^2+a*x^2-x^3);               % mat (ans.102,foo)
foo = solve(x^2==4, x);                                 % mat (ans.103,foo)
foo = solve([2*a-b == 3, a+b+c == 1,-b+c == 6],[a,b,c]);
tmp = [foo.a, foo.b, foo.c];                            % mat (ans.104,tmp)
foo = vpa(pi,50);                                       % mat (ans.105,foo)
foo = partfrac(1/((1 + x)*(5 + x)));                    % mat (ans.106,foo)
foo = simplifyFraction((1/(1 + x) - 1/(5 + x))/4);      % mat (ans.107,foo)
foo = simplify(tanh(log(x)));                           % mat (ans.108,foo)
foo = simplify(tanh(i*x));                              % mat (ans.109,foo)
foo = simplify(sinh(3*x) - 3*sinh(x) - 4*(sinh(x))^3);  % mat (ans.110,foo)
foo = tanh(log(x));                                     % mat (lhs.108,foo)
foo = tanh(i*x);                                        % mat (lhs.109,foo)
foo = sinh(3*x) - 3*sinh(x) - 4*(sinh(x))^3 ;           % mat (lhs.110,foo)
```

```
\begin{align*}
   &\mat*{ans.101}\\
   &\mat*{ans.102}\\
   &\mat*{ans.103}\\
   &\mat*{ans.104}\\
   &\mat*{ans.105}\\
   &\mat*{ans.106}\\
   &\mat*{ans.107}\\
   \mat{lhs.108} &= \Mat{ans.108}\\
   \mat{lhs.109} &= \Mat{ans.109}\\
   \mat{lhs.110} &= \Mat{ans.110}\\
\end{align*}
```

$$\texttt{ans.101} := a^3 + 3\,a^2\,b + 3\,a\,b^2 + b^3$$

$$\texttt{ans.102} := \begin{pmatrix} x & x+1 & a-x \end{pmatrix}$$

$$\texttt{ans.103} := \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

$$\texttt{ans.104} := \begin{pmatrix} \frac{1}{5} & -\frac{13}{5} & \frac{17}{5} \end{pmatrix}$$

$$\texttt{ans.105} := 3.1415926535897932384626433832795028841971693993751$$

$$\texttt{ans.106} := \frac{1}{4\,(x+1)} - \frac{1}{4\,(x+5)}$$

$$\texttt{ans.107} := \frac{1}{(x+1)\,(x+5)}$$

$$\tanh\left(\ln\left(x\right)\right) = \frac{x^2-1}{x^2+1} \qquad (\texttt{ans.108})$$

$$\tanh\left(x\,1\mathrm{i}\right) = \tan\left(x\right)\,1\mathrm{i} \qquad (\texttt{ans.109})$$

$$-4\sinh\left(x\right)^3 - 3\sinh\left(x\right) + \sinh\left(3\,x\right) = 0 \qquad (\texttt{ans.110})$$

Linear Algebra

```
syms mat lambda
mat  = sym([[2,3]; [5,4]]);              % mat (ans.201,mat)
[vec,lam] = eig(mat);                    % eigenvalues
eig1 = lam(1,1);                         % 1st eigenvalue
eig2 = lam(2,2);                         % 2nd eigenvalue
v1   = vec(:,1);                         % 1st eigenvector
v2   = vec(:,2);                         % 2nd eigenvector
eigennums = sym([eig1,eig2]);            % mat (ans.202,eigennums)
eigenvecs = vec;                         % mat (ans.203,eigenvecs)
charpol   = charpoly(mat,lambda);        % mat (ans.204,charpol)
myrhs = sym([3;7]);                      % mat (ans.205,myrhs)
mysol = mat\myrhs;                       % mat (ans.206,mysol)
```

```
\begin{align*}
   &\mat*{ans.201}\\
   &\mat*{ans.202}\\
   &\mat*{ans.203}\\
   &\mat*{ans.204}\\
   &\mat*{ans.205}\\
   &\mat*{ans.206}
\end{align*}
```

$$\begin{aligned}
\texttt{ans.201} &:= \begin{pmatrix} 2 & 3 \\ 5 & 4 \end{pmatrix} \\
\texttt{ans.202} &:= \begin{pmatrix} -1 & 7 \end{pmatrix} \\
\texttt{ans.203} &:= \begin{pmatrix} -1 & \frac{3}{5} \\ 1 & 1 \end{pmatrix} \\
\texttt{ans.204} &:= \lambda^2 - 6\,\lambda - 7 \\
\texttt{ans.205} &:= \begin{pmatrix} 3 \\ 7 \end{pmatrix} \\
\texttt{ans.206} &:= \begin{pmatrix} \frac{9}{7} \\ \frac{1}{7} \end{pmatrix}
\end{aligned}$$

Limits

```
syms a n x dx
foo = limit(sin(4*x)/x,x,0);                % mat (ans.301,foo)
foo = limit(2^x/x,x,Inf);                   % mat (ans.302,foo)
foo = limit(((x+dx)^2 - x^2)/dx, dx,0);     % mat (ans.303,foo)
foo = limit((4*n + 1)/(3*n - 1),n,Inf);     % mat (ans.304,foo)
foo = limit((1+(a/n))^n,n,Inf);             % mat (ans.305,foo)
```

```
\begin{align*}
   &\mat*{ans.301}\\
   &\mat*{ans.302}\\
   &\mat*{ans.303}\\
   &\mat*{ans.304}\\
   &\mat*{ans.305}
\end{align*}
```

$$\begin{aligned}
\texttt{ans.301} &:= 4 \\
\texttt{ans.302} &:= \infty \\
\texttt{ans.303} &:= 2\,x \\
\texttt{ans.304} &:= \frac{4}{3} \\
\texttt{ans.305} &:= \mathrm{e}^{a}
\end{aligned}$$

Series

```
syms n x
foo = taylor((1 + x)^(-2), x, 1, 'Order', 6);   % mat (ans.401,foo)
foo = taylor(exp(x), x, 0, 'Order', 6);         % mat (ans.402,foo)
foo = symsum(1/n^2, n,1,50);                    % mat (ans.403,foo)
foo = symsum(1/n^4, n,1,Inf);                   % mat (ans.404,foo)
```

```
\begin{align*}
   &\mat*{ans.401}\\
   &\mat*{ans.402}\\
   &\mat*{ans.403}\\
   &\mat*{ans.404}
\end{align*}
```

$$\begin{aligned}
\texttt{ans.401} &:= \frac{3\,(x-1)^2}{16} - \frac{x}{4} - \frac{(x-1)^3}{8} + \frac{5\,(x-1)^4}{64} - \frac{3\,(x-1)^5}{64} + \frac{1}{2} \\
\texttt{ans.402} &:= \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1 \\
\texttt{ans.403} &:= \frac{3121579929551692678469635660835626209661709}{1920815367859463099600511526151929560192000} \\
\texttt{ans.404} &:= \frac{\pi^4}{90}
\end{aligned}$$

Calculus

```
syms a b x y
foo = diff(x*sin(x),x);                    % mat (ans.501,foo)
foo = subs(foo,x,pi/4);                    % mat (ans.502,foo)
foo = int(2*sin(x)^2, x,a,b);              % mat (ans.503,foo)
foo = int(2*exp(-x^2),x,0,Inf);            % mat (ans.504,foo)
foo = int(int(x^2 + y^2, y,0,x), x,0,1);   % mat (ans.505,foo)
```

```
\begin{align*}
   &\mat*{ans.501}\\
   &\mat*{ans.502}\\
   &\mat*{ans.503}\\
   &\mat*{ans.504}\\
   &\mat*{ans.505}
\end{align*}
```

$$\texttt{ans.501} := \sin\left(x\right) + x\,\cos\left(x\right)$$

$$\texttt{ans.502} := \frac{\pi\,\sqrt{2}}{8} + \frac{\sqrt{2}}{2}$$

$$\texttt{ans.503} := b - a + \frac{\sin\left(2\,a\right)}{2} - \frac{\sin\left(2\,b\right)}{2}$$

$$\texttt{ans.504} := \sqrt{\pi}$$

$$\texttt{ans.505} := \frac{1}{3}$$

Differential equations

```
syms a x y(x)

ode = diff(y,x) + y(x) == 2*a*sin(x);
sol = dsolve(ode)                                   % mat (ans.601,sol)
sol = dsolve(ode,y(0) == 0)                         % mat (ans.602,sol)

ode = diff(y,x,x) + y(x) == 0;
ddx = diff(y,x)
bcs = [y(0)==0,ddx(0)==1]
sol = dsolve(ode)                                   % mat (ans.602,sol)
sol = dsolve(ode,bcs)                               % mat (ans.603,sol)

ode = diff(y,x,x) + 5*diff(y,x) - 6*y(x) == 0;
sol = dsolve(ode)                                   % mat (ans.604,sol)

syms C10 C11
sol = subs(subs(sol,C10,2),C11,3)                    % mat (ans.605,sol)
```

```
\begin{align*}
   &\mat*{ans.601}\\
   &\mat*{ans.602}\\
   &\mat*{ans.603}\\
   &\mat*{ans.604}\\
   &\mat*{ans.605}
\end{align*}
```

$$\texttt{ans.601} := C_4\,\mathrm{e}^{-x} - a\,\left(\cos\left(x\right) - \sin\left(x\right)\right)$$
$$\texttt{ans.602} := C_6\,\cos\left(x\right) - C_7\,\sin\left(x\right)$$
$$\texttt{ans.603} := \sin\left(x\right)$$
$$\texttt{ans.604} := C_{11}\,\mathrm{e}^{x} + C_{10}\,\mathrm{e}^{-6\,x}$$
$$\texttt{ans.605} := 2\,\mathrm{e}^{-6\,x} + 3\,\mathrm{e}^{x}$$

Numerical integration of coupled ODEs

This example is based that given in the Mathworks web page https://www.mathworks.com/examples/matlab/mw/matlab-ex13677222-solve-nonstiff-odes. It uses the Matlab function `ODE45` to integrate a coupled pair of ordinary differential equations – the van der Pol equation with $\mu = 1$.

```
[t,y] = ode45(@vdp1,[0 20],[2; 0]);

% Plot of the solution
plot(t,y(:,1),'-o',t,y(:,2),'-o')
xlabel('Time t')
ylabel('Solution y')
legend('y_1','y_2')

print(gcf,'example_02_fig.png','-dpng');

dlmwrite ('example_02.txt',[t';y(:,1)';y(:,2)']','delimiter',' ','precision','% .8e');

function dydt = vdp1(t,y)
%VDP1  Evaluate the van der Pol ODEs for mu = 1
%
%   See also ODE113, ODE23, ODE45.

%   Jacek Kierzenka and Lawrence F. Shampine
%   Copyright 1984-2014 The MathWorks, Inc.

dydt = [y(2); (1-y(1)^2)*y(2)-y(1)];
end
```

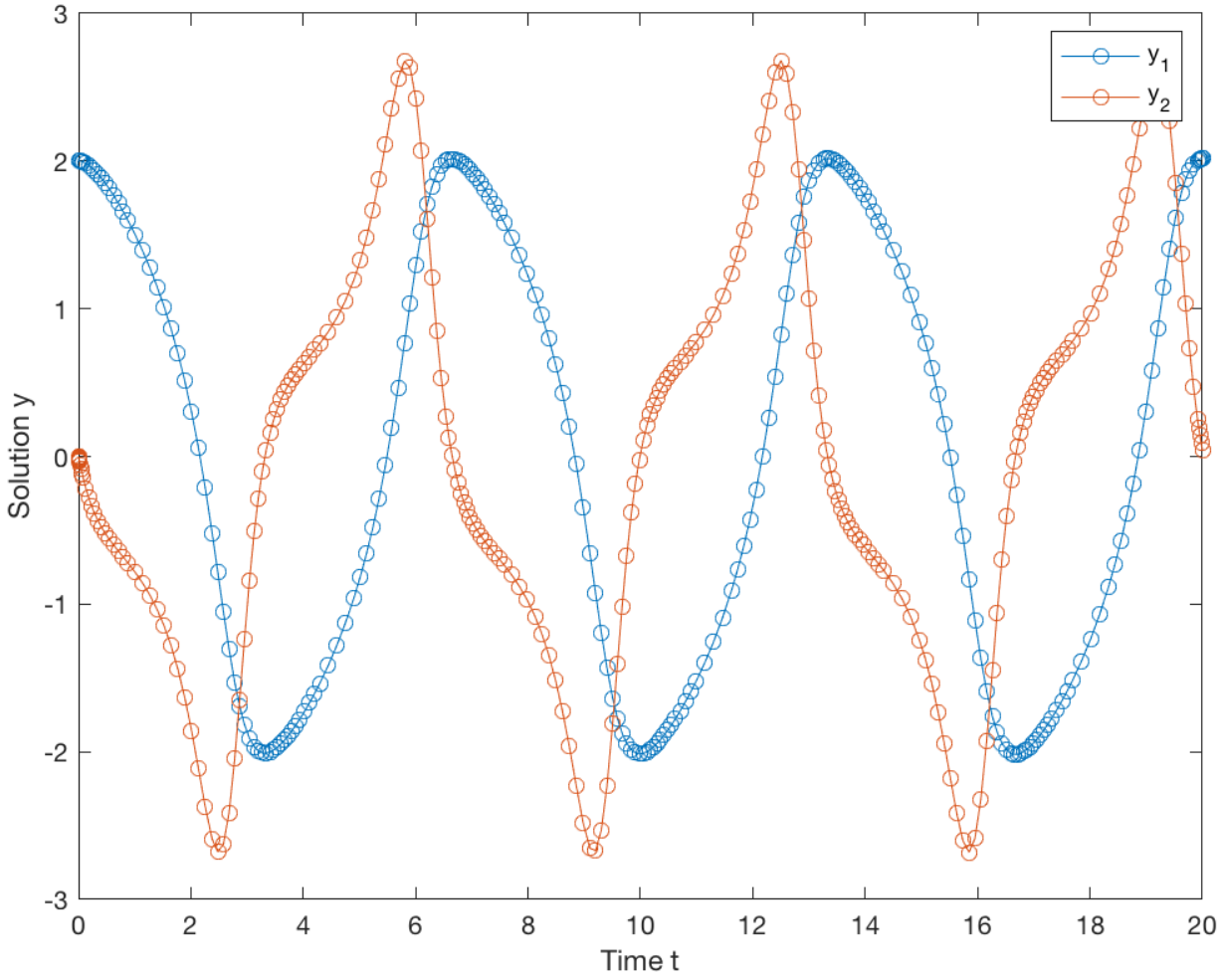

Figure 1: Solution of van der Pol equation ($\mu = 1$) using ODE45.

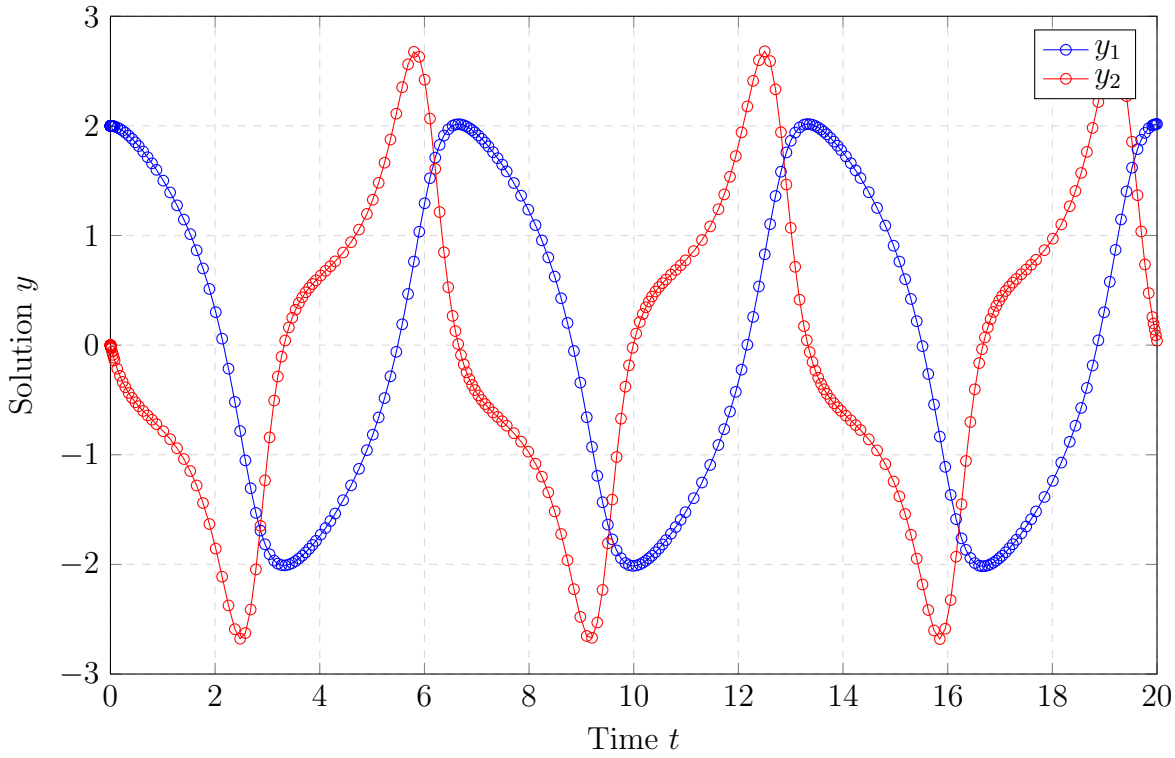

Figure 2: Solution of van der Pol equation ($\mu = 1$) using ODE45.

```
\begin{tikzpicture} % requires \usepackage{pgfplots}
   \begin{axis}
      [xmin= 0.0, xmax=20.0,
       ymin=-3.0, ymax=3.0,
       xlabel=$\text{Time }t$, ylabel=$\text{Solution }y$,
       grid=major, grid style={dashed,gray!30},
       legend entries = {$y_1$, $y_2$}]
       \addplot[blue, mark=o]   table [x index=0, y index=1]{example_02.txt};
       \addplot[red, mark=o]    table [x index=0, y index=2]{example_02.txt};
   \end{axis}
\end{tikzpicture}
\captionof{figure}{Solution of van der Pol equation ($\mu = 1$) using ODE45.} % requires \usepackage{caption}
```

Surface plot

This and the following exampes were lifted from the collection at https://www.mathworks.com/examples/symbolic/mw/symbolic-ex98670373-computational-mathematics-in-symbolic-math-toolbox. Other nice examples can also be found here https://www.mathworks.com/examples/symbolic/mw/symbolic-ex98670382-analytical-plotting-with-symbolic-math-toolbox.

```
syms x y
fsurf(sin(x)+cos(y));
print(gcf,'example_03_fig1.png','-dpng');
```

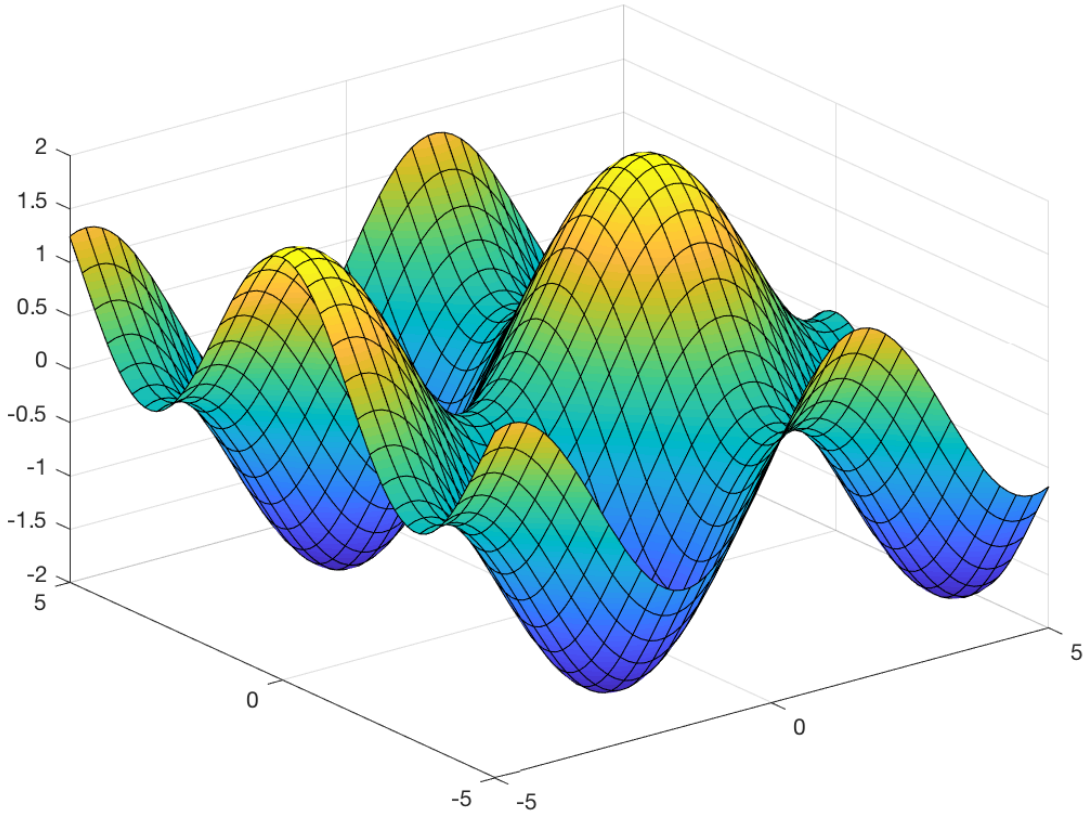

Figure 1: A 2d surface plot.

Contour plot

```
syms x y
fcontour(sin(x)+cos(y));
print(gcf,'example_03_fig2.png','-dpng');
```

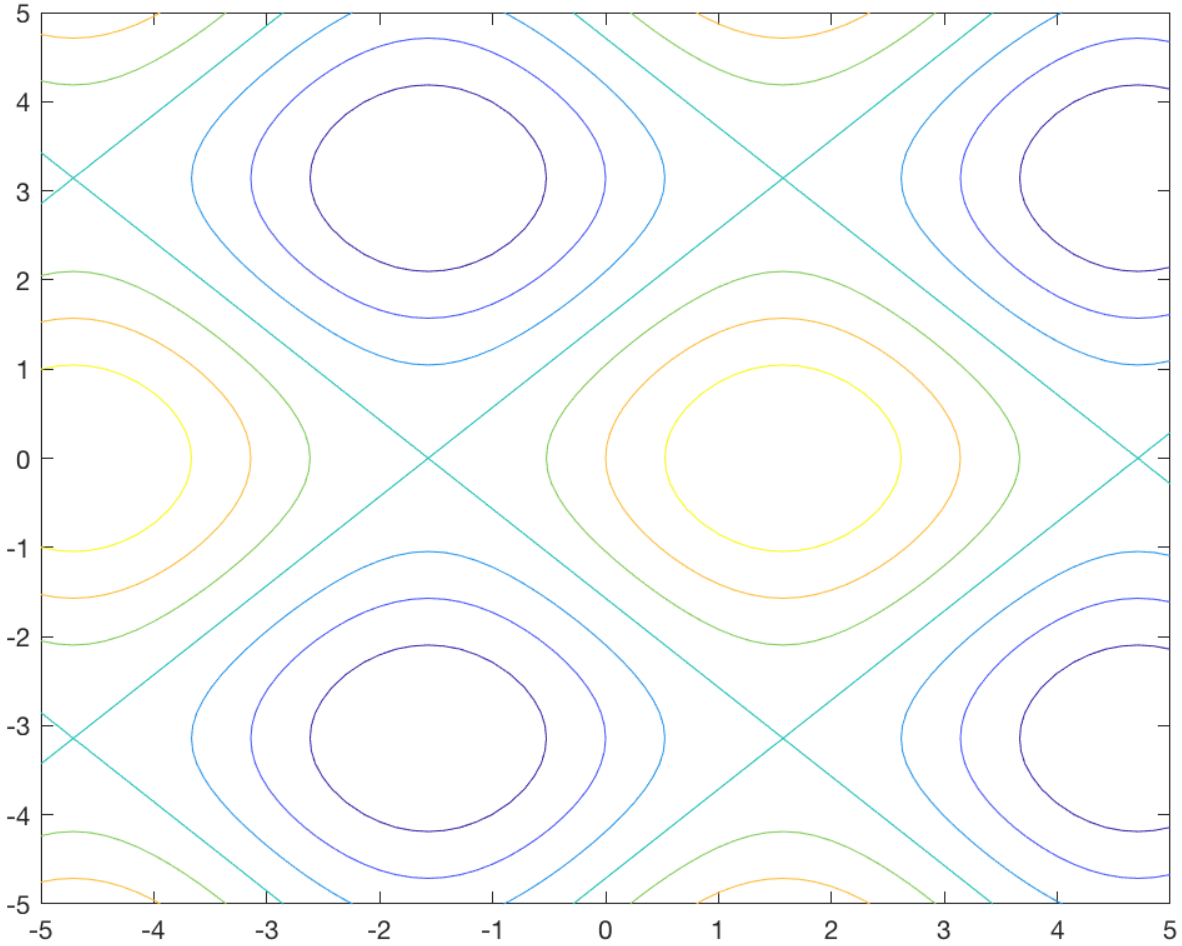

Figure 2: A contour plot

Parameteric plot

```
syms t
xt = exp(abs(t)/10).*sin(5*abs(t));
yt = exp(abs(t)/10).*cos(5*abs(t));
zt = t;
h = fplot3(xt,yt,zt,[-10,10],'r');
view([-45,20]);
print(gcf,'example_03_fig3.png','-dpng');
```

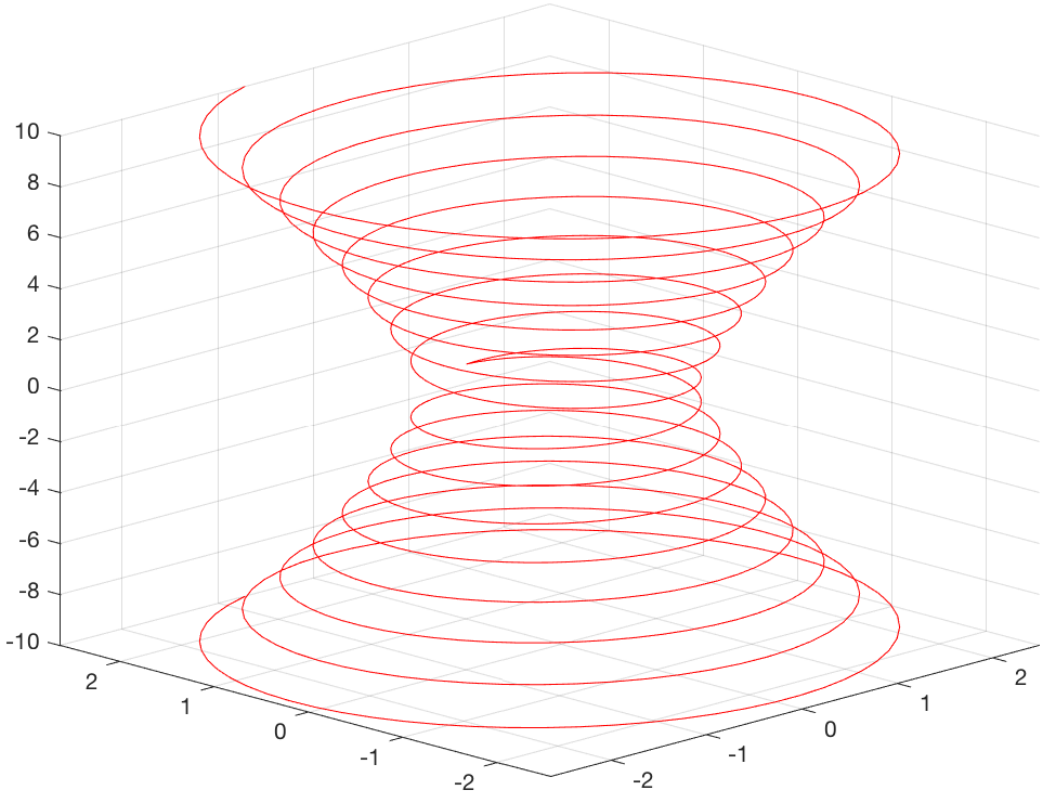

Figure 3: A 3d parametric plot.

Plotting Bessel functions

This simple example uses Matlab to produce a plot of the first six Bessel functions. Two plots are shown, one created by Matlab and a second created by LaTeX using the plotting package `pgfplots` and the data exported from Matlab.

This example is based upon the Mathworks example at https://au.mathworks.com/matlabcentral/fileexchange/35229-matlab-plot-gallery-standard-line-colors.

```
x  = 0:0.1:15;
y0 = besselj(0,x);
y1 = besselj(1,x);
y2 = besselj(2,x);
y3 = besselj(3,x);
y4 = besselj(4,x);
y5 = besselj(5,x);

plot(x, y0, 'r', x, y1, 'g', x, y2, 'b',  ...
     x, y3, 'c', x, y4, 'm', x, y5, 'y');
legend('J_0','J_1','J_2','J_3','J_4','J_5');

print(gcf,'example_04_fig.png','-dpng');

% Note: using ' on [x;y0...]' ensures the six functions are written as columns of example_01.txt

dlmwrite ('example_04.txt',[x;y0;y1;y2;y3;y4;y5]','delimiter',' ','precision','% .8e');
```

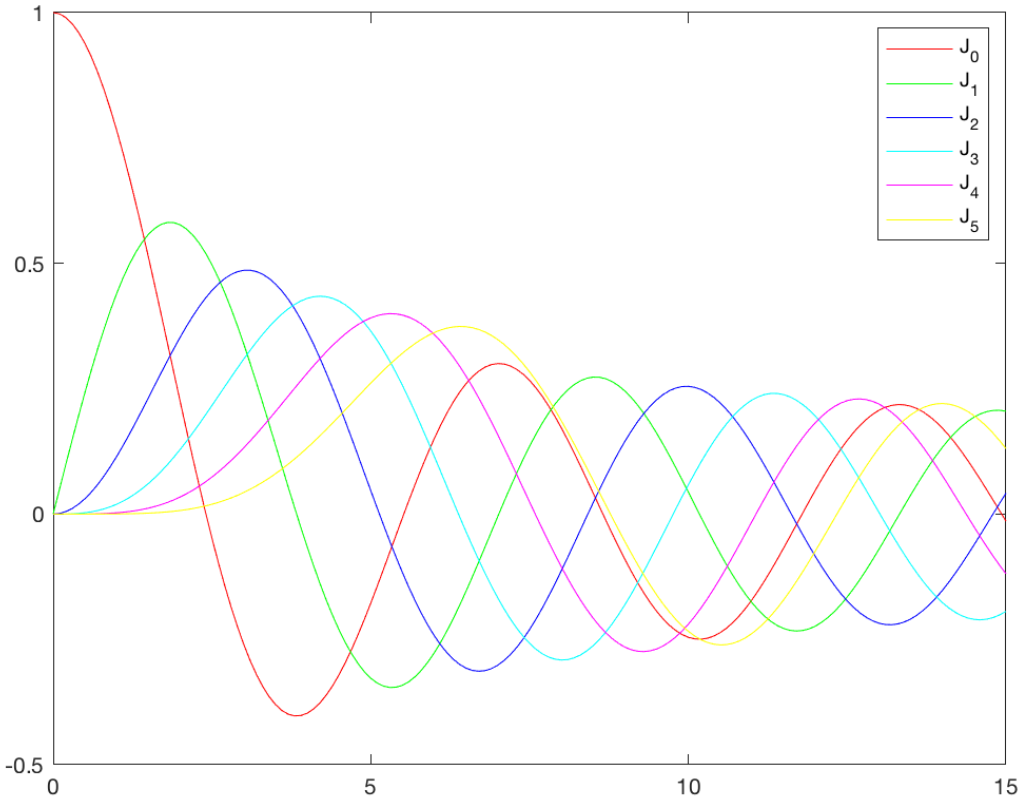

Figure 1: The first six Bessel functions.

Using pgfplots

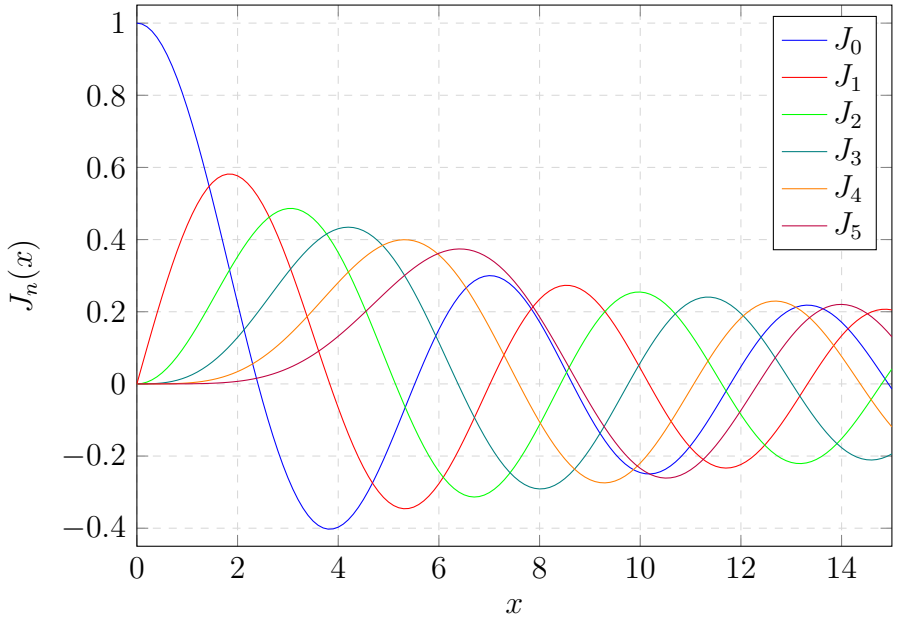

Figure 2: The first six Bessel functions.

```
\begin{tikzpicture} % requires \usepackage{pgfplots}
   \begin{axis}
      [xmin= 0.0,  xmax=15.0,
       ymin=-0.45, ymax=1.05,
       xlabel=$x$, ylabel=$J_n(x)$,
       grid=major, grid style={dashed,gray!30},
       legend entries = {$J_0$, $J_1$, $J_2$, $J_3$, $J_4$, $J_5$}]
       \addplot[blue]   table [x index=0, y index=1]{example_04.txt};
       \addplot[red]    table [x index=0, y index=2]{example_04.txt};
       \addplot[green]  table [x index=0, y index=3]{example_04.txt};
       \addplot[teal]   table [x index=0, y index=4]{example_04.txt};
       \addplot[orange] table [x index=0, y index=5]{example_04.txt};
       \addplot[purple] table [x index=0, y index=6]{example_04.txt};
   \end{axis}
\end{tikzpicture}
\captionof{figure}{The first six Bessel functions.} % requires \usepackage{caption}
```

Quadratic convergence of Newton-Raphson iterations

This is a simple example that uses the Matlab Symbolic Toolbox to demonstrate the quadratic converegnce of Newton-Raphson iterations to the exact root of a non-linear equation.

```
syms x f(x) df(x) Step(x)

f(x)    = x-exp(-x);
df(x)   = diff(f,x);
Step(x) = x- f(x)/df(x);

digits(200);

x_new = vpa (1/2);
f_new = vpa (f(x_new));

% matBeg (table)

fprintf ('\\RuleA %2d & % .25f & % .10e &\\\\\n',[0,x_new,f_new]);

for n = 1:6
   x_old = x_new;
   x_new = vpa (Step(x_new));
   f_old = vpa (f(x_old));
   f_new = vpa (f(x_new));
   ratio = vpa (f_new / f_old^2);
   fprintf ('\\RuleA %2d & % .25f & % .10e & % .5f\\\\\n',[n,x_new,f_new,ratio]);
end

% matEnd (table)
```

Note the clear quadratic convergence in the iterations – the last column settles to approximately -0.11546 independent of the number of iterations. This behaviour would not be seen using normal floating point computations as they are normally limited to no more than 18 decimal digits. This computation used 200 decimal digits.

Newton-Raphson iterations $x_{n+1} = x_n - f_n/f'_n\ , \quad f(x) = x - e^{-x}$			
n	x_n	$\epsilon_n = x_n - e^{-x_n}$	$\epsilon_n/\epsilon_{n-1}^2$
0	0.50000000000000000000000000	-1.0653065971e-01	
1	0.56631100319721816571671980	-1.3045098060e-03	-0.11495
2	0.56714316503486217335705530	-1.9648047172e-07	-0.11546
3	0.56714329040978106455384020	-4.4574262753e-15	-0.11546
4	0.56714329040978384011140180	-2.2941072910e-30	-0.11546
5	0.56714329040978384011140180	-6.0767705445e-61	-0.11546
6	0.56714329040978384011140180	-4.2637434326e-122	-0.11546

```
\def\RuleA{\vrule depth0pt  width0pt height14pt}
\def\RuleB{\vrule depth8pt  width0pt height14pt}
\def\RuleC{\vrule depth10pt width0pt height16pt}

\setlength{\tabcolsep}{0.025\textwidth}%

\begin{center}
   \begin{tabular}{cccc}%
      \noalign{\hrule height 1pt}
      \multicolumn{4}{c}{\RuleC\rmfamily\bfseries%
      Newton-Raphson iterations \quad%
      $x_{n+1} = x_n - f_n/f'_n\ ,\quad f(x) = x-e^{-x}$}\\
      \noalign{\hrule height 1pt}
      \RuleB$n$&$x_n$&$\epsilon_{n} = x_{n} - e^{-x_{n}}$&$\epsilon_{n}/\epsilon_{n-1}^2$\\
      \noalign{\hrule height 0.5pt}
      \mat{table}
      \noalign{\hrule height 1pt}
   \end{tabular}
\end{center}
```

Using tagged blocks

The following Matlab code block contains a matched `matBeg/matEnd` pair, with the tag name `info`, to capture the output from the formatted Python `print` statements.

```
% matBeg(info)
prodinfo=ver('matlab');
fprintf('Date :         &%s\\\\\n',datestr(now));
fprintf('Name :         &%s\\\\\n',prodinfo.Name);
fprintf('Version :      &%s\\\\\n',prodinfo.Version);
fprintf('Release :      &%s\\\\\n',prodinfo.Release);
fprintf('Release date : &%s\n',    prodinfo.Date);
% matEnd(info)
```

```
\bgroup\tt
\begin{tabular}{rl}
   \mat{info}
\end{tabular}
\egroup
```

Here is the output caught from the above block.

```
         Date :  27-Aug-2018 10:18:21
         Name :  MATLAB
      Version :  9.4
      Release :  (R2018a)
 Release date :  06-Feb-2018
```

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

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

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

```
matlatex.sh -x -i mixed
pylatex.sh  -x -i mixed
pdflatex          mixed
```

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

The Matlab code

Here Matlab is used to first find the general solution of th differential equation. The boundary conitions 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

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

while the particular solution satifying the boundary conditions is given by

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

The Python code

This is a straighforward 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')
```

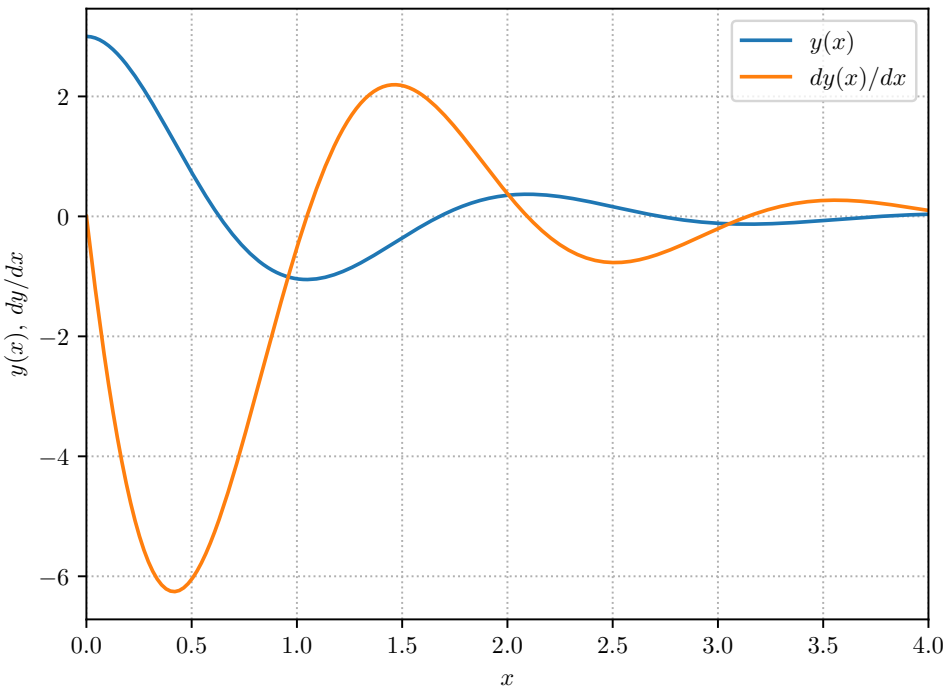

Figure 1: The function and its derivative.