

## Matlab : Numerical Solution of Ordinary Differential Equations

Matlab has facilities for the numerical solution of ordinary differential equations (ODEs) of any order. In this document we first consider the solution of a first order ODE. Higher order ODEs can be solved using the same methods, with the higher order equations first having to be reformulated as a system of first order equations. Techniques for solving the first order and second order equations are demonstrated by considering a couple of typical examples.

### First Order ODEs

In this section an example of a first order ODE is solved in Matlab. The results are compared with the exact solution.

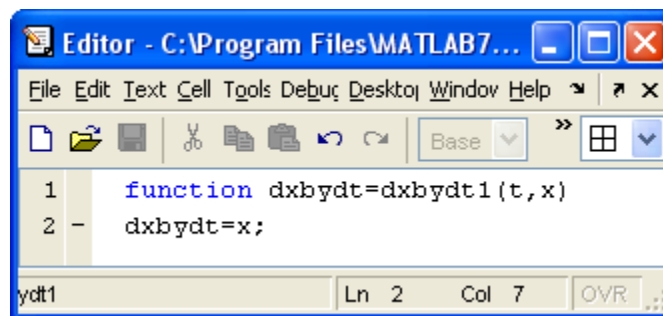
#### Example 1

As the first example let us solve the first order ODE:

$$\frac{dx}{dt} = x$$

with  $x(0) = 2$  in the domain  $0 \leq t \leq 2$ .

Firstly the  $\frac{dx}{dt}$  function can be most neatly defined in a separate “.m” file. In the main Matlab window click on File → New → M-File. In the M-File editor below type in the function as follows.



```
Editor - C:\Program Files\MATLAB7...
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Base >> [Grid] [Dropdown]
1 function dxbydt=dxbydt1(t,x)
2 - dxbydt=x;
ydt1 Ln 2 Col 7 OVR
```

In general we are solving the differential equation  $\frac{dx}{dt} = f(t, x)$ . In this file dxbydt is a general identifier for  $\frac{dx}{dt}$ . The identifier dxbydt1 is the particular identifier for this example (example 1). The filename must follow this particular identifier; save the file as dxbydt1.m. In this example  $f(t, x) = x$  so we write dxbydt=x; .

Going back to the main Matlab window type the following and press return.  
`[T,X]=ode23(@dxbydt1,[0,2],2)`

The meaning of each part of this command is explained here:

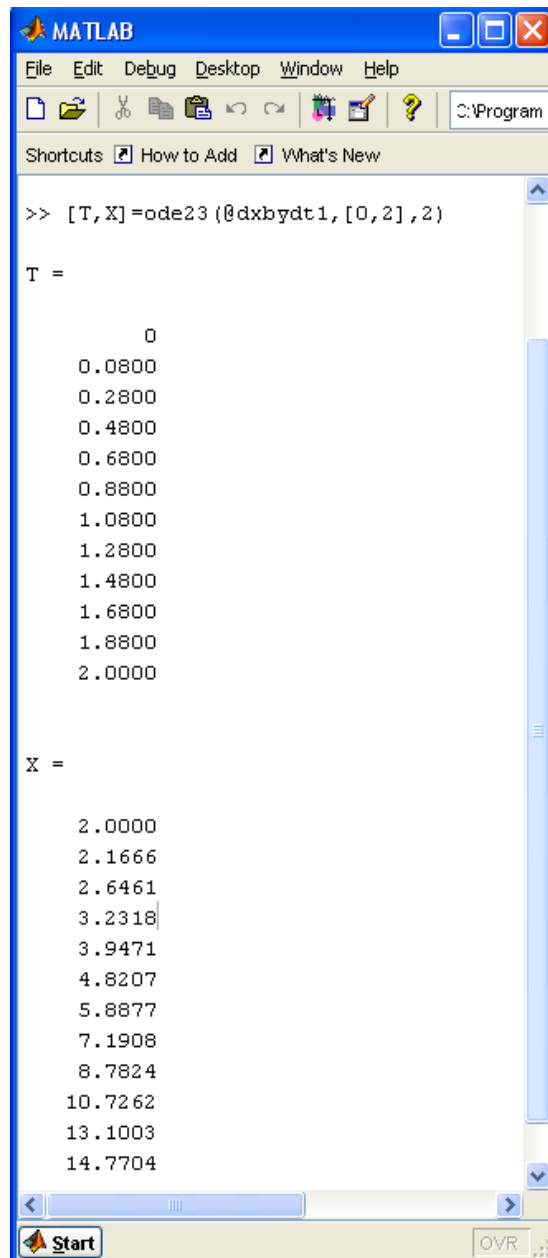
ode23 – the ordinary differential equation solving routine

@dxbydt1 – the handle to the m-file that contains the function for  $\frac{dx}{dt}$ , defined earlier

[0,2] – the range of t ;  $0 \leq t \leq 2$

2 – the value of x when t=0.

Following the execution of this line of code, T contains a list of t-values and X contains the corresponding list of values for x for each t-value. The results are shown in the following screen.



```
MATLAB
File Edit Debug Desktop Window Help
C:\Program F
Shortcuts How to Add What's New
>> [T,X]=ode23(@dxbydt1,[0,2],2)

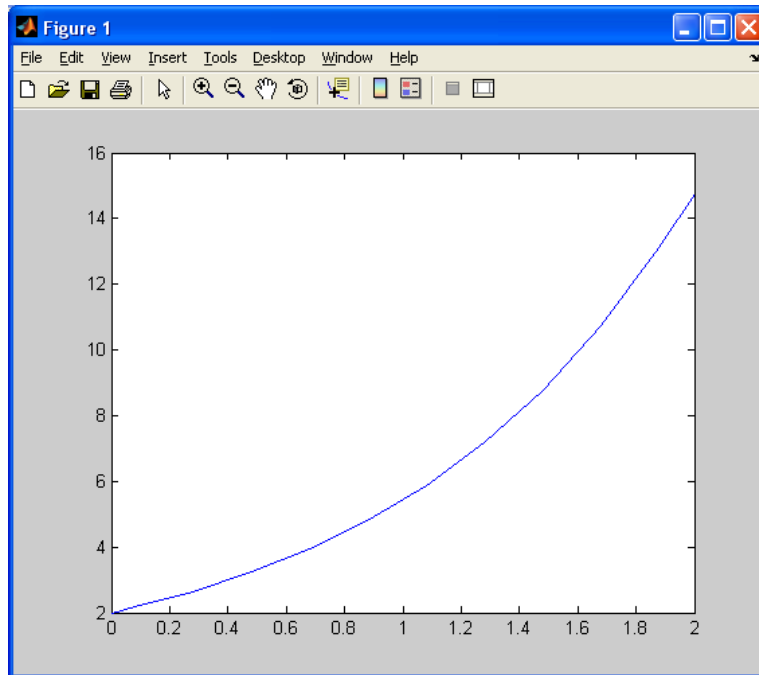
T =

    0
  0.0800
  0.2800
  0.4800
  0.6800
  0.8800
  1.0800
  1.2800
  1.4800
  1.6800
  1.8800
  2.0000

X =

  2.0000
  2.1666
  2.6461
  3.2318
  3.9471
  4.8207
  5.8877
  7.1908
  8.7824
  10.7262
  13.1003
  14.7704
```

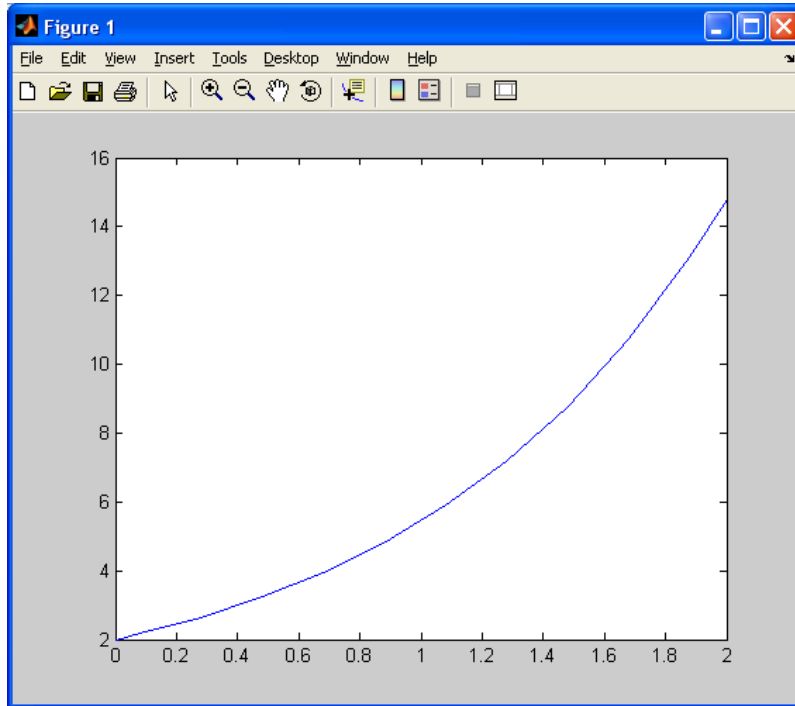
The command `plot(T,X)` gives a graph of the computed solution.



If we know the exact solution, we can find the exact solution for the same t-values using a command as follows. In this case the exact solution is  $2e^t$ . To make a comparison between the approximate and exact solutions, we first obtain the values of x at the same values of T. This can be achieved by the command `Xexact=2*exp(T)`, giving the following result.

```
MATLAB
File Edit Debug Desktop Window Help
C:\Program F
Shortcuts How to Add What's New
>> Xexact=2*exp(T)
Xexact =
    2.0000
    2.1666
    2.6463
    3.2321
    3.9478
    4.8218
    5.8894
    7.1933
    8.7859
   10.7311
   13.1070
   14.7781
```

The results can be plotted using the command `plot(T,Xexact)`, giving the following graph, demonstrating the accuracy of the computed solution.



## Second order ODEs

Second or higher order ODEs can be transformed into a system of 1<sup>st</sup> order ODEs<sup>1</sup>. Once the technique is understood for second order systems, the same technique can easily be extended to ODEs of any order. In this section, the technique for solving second order ODEs with Matlab is demonstrated through using a typical example.

### Example 2

Let us solve the second order ODE:

$$\frac{d^2x}{dt^2} - 7\frac{dx}{dt} + 12x = 2$$

with  $x = 1$  and  $\frac{dx}{dt} = 5$  when  $t = 0$  for  $t$  in the domain  $0 \leq t \leq 2$ .

First we need to write the ODE as a system of two first order ODEs. The most straightforward and common technique for doing this is to introduce a new variable

---

<sup>1</sup> [Ordinary Differential Equations](#)

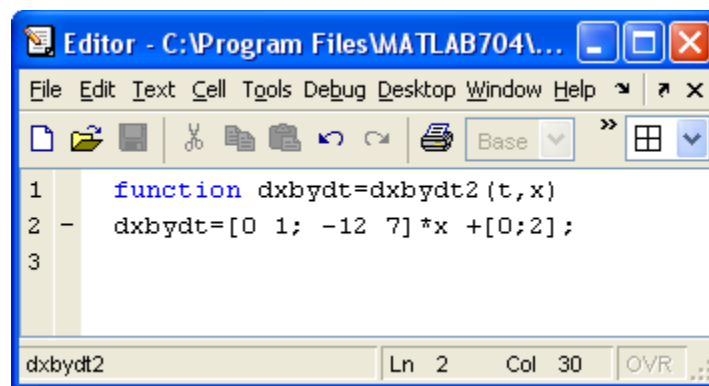
$y$  and setting  $y = \frac{dx}{dt}$ , giving us the first equation. Substituting this into the second order ODE above gives the second equation:

$$\frac{dy}{dt} - 7y + 12x = 2$$

Putting the two equations together and using matrix-vector<sup>2</sup> notation gives the following first order system of equations:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -12 & 7 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Using Matlab's matrix-vector operations<sup>3</sup>, the system of equations can be written as follows. Note that  $x$  is a 2-vector,  $x(1)$  represents  $x$  and  $x(2)$  represents  $y$ . The following function is stored in a file named `dxbydt2.m`.



```
Editor - C:\Program Files\MATLAB704\...
File Edit Text Cell Tools Debug Desktop Window Help
Base
1 function dxbydt=dxbydt2(t,x)
2 - dxbydt=[0 1; -12 7]*x+[0;2];
3
dxbydt2 Ln 2 Col 30 OVR
```

In the main program, the command `[T,X]=ode23(@dxbydt2,[0,2],[1;5]);` forms a set of  $t$ -values in  $T$  and a set of  $x$ -values in  $X$  that is the approximate solution. Since  $x$  is a 2-vector  $X$  is a matrix with two columns; the first column represents  $x(t)$  at the values of  $t$  listed in  $T$  and the second column similarly represents  $y(t)$  ( or  $dx/dt$ ). The exact solution to the ODE can be found analytically<sup>4</sup> and is as follows:

$$x(t) = \frac{1}{6} - \frac{5}{3}e^{3t} + \frac{5}{2}e^{4t}.$$

<sup>2</sup> [Matrix Arithmetic](#)

<sup>3</sup> [Matlab/Freemat: Array Operations - Matrix and Vector Arithmetic](#)

<sup>4</sup> [Solving ODEs by Laplace Transforms](#)

The command  $X_{\text{exact}}=(1/6)-(5/3)*\exp(3*T)+(5/2)*\exp(4*T)$ ; sets the exact solution at the same values of  $t$ .

The computed solution can now be plotted using the command `plot(T,X(:,1))` and the exact solution may be plotted using the command `plot(T,Xexact)`. The approximate solution graph is given on the left, the exact solution is given on the right, below.

