Solving Ordinary Differential Equations with Matlab and Maple

October 25, 2000

1 A sample problem

We consider the initial value problem :

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}(t)) \tag{1}$$

$$\mathbf{x}_0 = \mathbf{x}(0) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

where

$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \quad , \quad \mathbf{f}(t, \mathbf{x}(t)) = \begin{pmatrix} -\alpha x(t) + y(t) \\ -x(t) - \alpha y(t) \\ -z^2(t)/(t+e) \end{pmatrix} \tag{3}$$

The exact solution is

$$\mathbf{x}(t) = \begin{pmatrix} e^{-\alpha t} \sin(t) \\ e^{-\alpha t} \cos(t) \\ \ln^{-1}(t+e) \end{pmatrix} \tag{4}$$

2 Solving with Matlab

The main ODE solvers for non-stiff problems are `ode23` and `ode45`. Here we exhibit by means of a simple example the use of the latter, which is a higher order method (`ode23` can be used instead).

First we need a mfile that implements $\mathbf{f}(t, \mathbf{x}(t))$. This should be a Matlab function that receives as input a parameter T and a 3×1 vector X. The output is a 3×1 vector that corresponds to $\mathbf{f}$ evaluated at (T, X).

```
function F=stable(T,X)
% F=stable(T,X)

F = zeros(2,1);  % F must be a column vector
a = 0.1;
F(1) = X(2) - a*X(1);
F(2) = - (X(1) + a*X(2));
F(3) = - (X(3)^2)/(T + exp(1));
```

Alternatively one may write

```
function F=stable(T,X)
% F=stable(T,X)

a = 0.1;
F = [   X(2) - a*X(1)
     - (X(1) + a*X(2))
     - (X(3)^2)/(T + exp(1)) ];
```

For consistency the name of the file matches the name of the function, ie `stable.m`. Then we may call the above function in a Matlab environment, as for example :

```
>> F=stable(2,[1 1 1])
```

Next we make the call to the ODE solver and plot the results. For this purpose we use the following script file `script.m` :

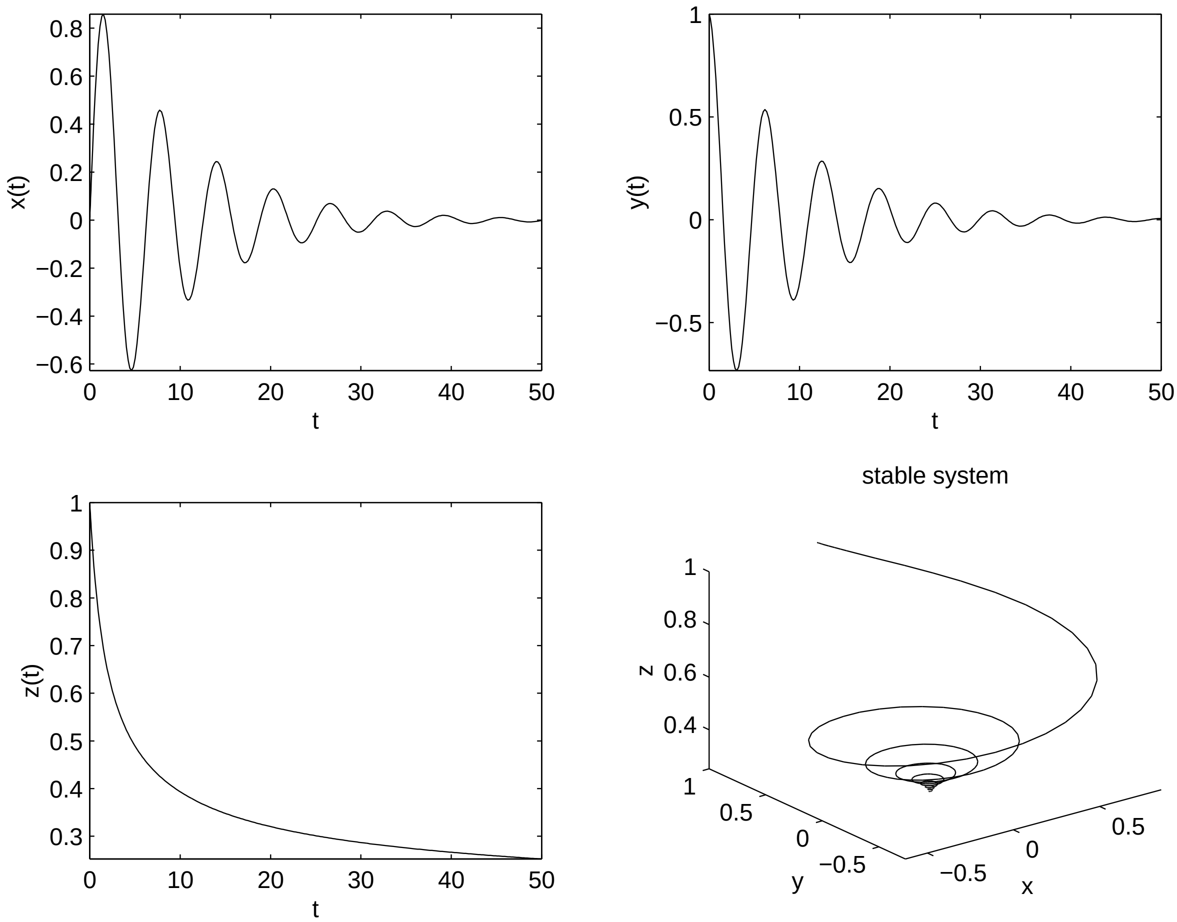
```
X0 = [0; 1; 1]; % initial values
Tspan = [0 50]; % domain of the approximation
[T,X] = ode45('stable',Tspan,X0); % solve using Matlab's default precision
% T = Nx1 vector with the discrete time steps
% X = Nx3 matrix with the corresponding approximations
subplot(2,2,1), plot(T,X(:,1)), xlabel('t'), ylabel('x(t)'), axis tight;
subplot(2,2,2), plot(T,X(:,2)), xlabel('t'), ylabel('y(t)'), axis tight;
subplot(2,2,3), plot(T,X(:,3)), xlabel('t'), ylabel('z(t)'), axis tight;
subplot(2,2,4), plot3(X(:,1),X(:,2),X(:,3));
subplot(2,2,4), xlabel('x'), ylabel('y'), zlabel('z'), axis tight;
title('stable system');
```

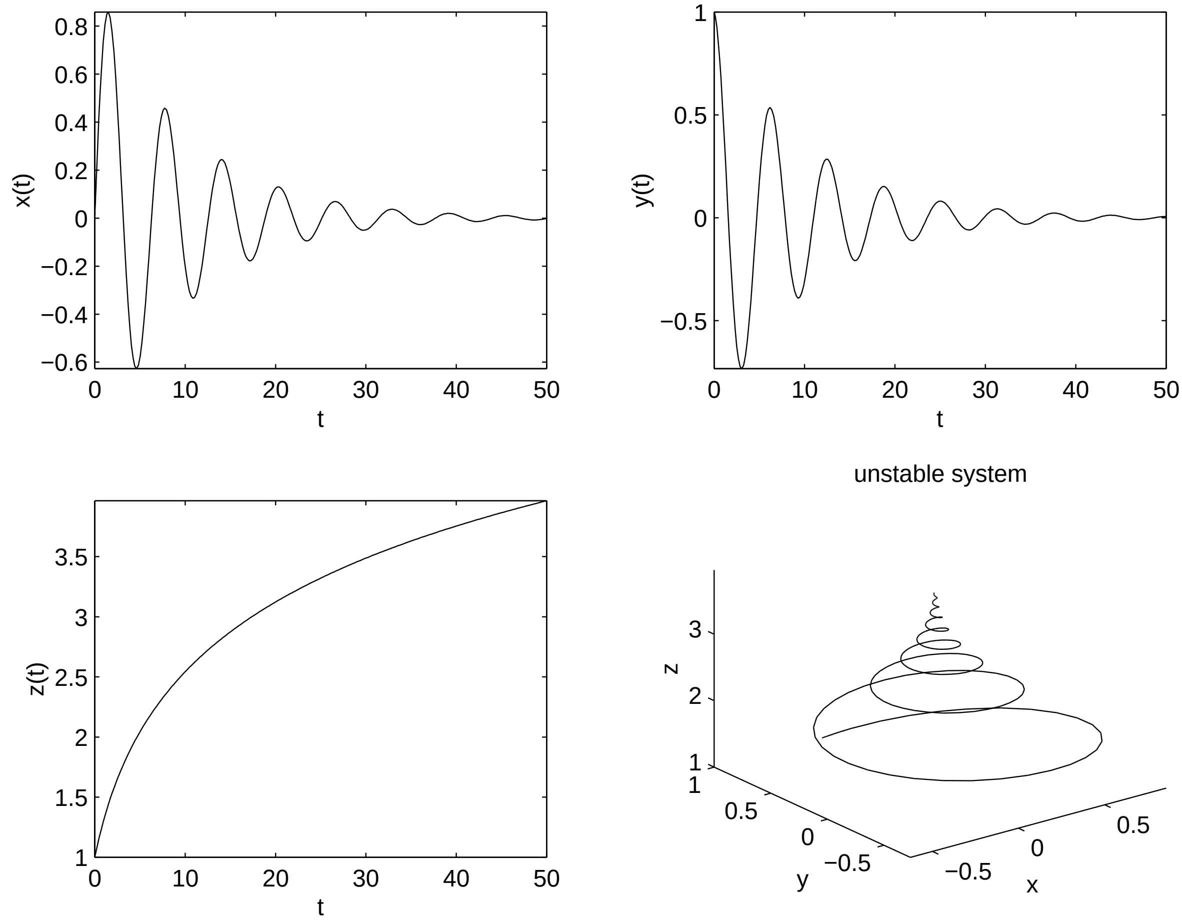
We execute the script by typing :

```
>> script
>> print -deps stable.eps
```

The last command saves the figure as an eps file, that can be included in a LaTeX file [1] :

[1] of course there are other options - see `help print`


x(t)
t
y(t)
t
z(t)
t
stable system
z
y
x



Above a small example of instability :

$$\mathbf{f}(t, \mathbf{x}(t)) = \begin{pmatrix} -\alpha x(t) + y(t) \\ -x(t) - \alpha y(t) \\ 1/(t+e) \end{pmatrix} \tag{5}$$

This was just an overview. For the necessary details see the online manual's of Matlab for `ode23, ode45, plot, plot3, print`.

3 Solving with Maple

The first step is to assign to some variables the expressions that describe the differential equations. This is done in a interactive Maple environment as follows (Maple's responses are also shown) :

```
> ode1 := diff(x(t),t)=-a*x(t)+y(t);
                                        d
                               ode1 := -- x(t) = -a x(t) + y(t)
                                        dt

> ode2 := diff(y(t),t)=-x(t)-a*y(t);
                                        d
                               ode2 := -- y(t) = -x(t) - a y(t)
                                        dt

> ode3 := diff(z(t),t)=-(z(t))^2/(t+exp(1));
                                                        2
                                         d           z(t)
                                ode3 := -- z(t) = - ----------
                                         dt         t + exp(1)
```

We may assign the whole system to another variable :

```
> ODE := {ode1,ode2,ode3};
                                                                                      2
             d                          d                          d               z(t)
      ODE := {-- x(t) = -a x(t) + y(t), -- y(t) = -x(t) - a y(t), -- z(t) = - ----------}
             dt                         dt                         dt         t + exp(1)
```

or together with the initial conditions :

```
> InitVal := {ode1,ode2,ode3,x(0)=0,y(0)=1,z(0)=1};
                                                                                     2
            d                          d                          d               z(t)
InitVal := {-- x(t) = -a x(t) + y(t), -- y(t) = -x(t) - a y(t), -- z(t) = - ----------, y(0) = 1,
            dt                         dt                         dt         t + exp(1)

    z(0) = 1, x(0) = 0}
```

Now everything is set and we may use Maple's `dsolve` routine both for analytic and numerical solutions.

We seek the general solution of our system with no initial conditions specified :

```
> dsolve(ODE,{x(t),y(t),z(t)});
{x(t) = _C2 exp(-t a) sin(t) - _C1 exp(-t a) cos(t),
```

```
                                                                                1
 y(t) = _C1 exp(-t a) sin(t) + _C2 exp(-t a) cos(t)}, {z(t) = --------------------}
                                                              ln(t + exp(1)) + _C3
```

Let's try a numerical solution

```
> dsolve(ODE,{x(t),y(t),z(t)},type=numeric);
Error, (in dsolve/numeric/error_conditions) initial conditions missing
```

We forgot something! Another attempt

```
> X := dsolve(InitVal,{x(t),y(t),z(t)},type=numeric);
                                   X := proc(rkf45_x)  ...  end
```

Maple's responses can be weird sometimes! But now things are not so complicated; Maple has returned a list of equations describing the value of t and the values of $x(t)$, $y(t)$ and $z(t)$ at that points :

```
> X(0);
                             [t = 0, x(t) = 0, y(t) = 1., z(t) = 1.]
> X(1);
Error, (in dsolve/numeric/rkf45) cannot evaluate boolean
```

In the second example we got a negative response because we didn't specify a value for the parameter `a`. Correction :

```
> a := 0.1;
                                                a := .1

> X(1);
     [t = 1, x(t) = .7613944391615205, y(t) = .4888857436828576, z(t) = .7614628597625739]

> X(50);
   [t = 50, x(t) = -.001767859749209587, y(t) = .006501934611122264, z(t) = .2522092077102135]
```

Here is how we convert the numeric solution to functions, one for each coordinate :

```
> x := t -> rhs( op(2,X(t)) );
                                    x := t -> rhs(op(2, X(t)))

> y := t -> rhs( op(3,X(t)) );
                                    y := t -> rhs(op(3, X(t)))

> z := t -> rhs( op(4,X(t)) );
                                    z := t -> rhs(op(4, X(t)))
```

Now we can plot each individual component :

```
> plot(x, 0..50);
```

For more information check the online manuals of Maple (usage : ?< topic >) for `dsolve`, `diff`, `int`, `plot`, `eval`, `evalf`.

References

[1] A. Heck, Introduction to Maple (2nd edition), Springer-Verlag 1996.

[2] W. Gander, J. Hřebíček Solving Problems in Scientific Computing using Maple and Matlab (3rd edition), Springer-Verlag 1997.