MATH661 Homework 5 - Approximate solution of differential systems

Posted: Oct 27, Due: 11:55PM, Nov 10

1 Problem statement

Solution of systems of differential equations is a bedrock of scientific computing. Our current understanding of physics is stated as conservation laws: some change occurs due to certain effects. A familiar example is Newton's second law of dynamics, commonly stated as $\boldsymbol{F} = m\boldsymbol{a}$, but more properly formulated as

$$\frac{\mathrm{d}\boldsymbol{h}}{\mathrm{d}t} = \frac{\mathrm{d}(m\boldsymbol{v})}{\mathrm{d}t} = \boldsymbol{F},$$

or that changes in momentum $\mathrm{d}(m\boldsymbol{v})$ occur due to the action of force over a time interval, $\boldsymbol{F}\,\mathrm{d}t$.

This homework guides you through the basic theoretical concepts, and then applies these to solve a system of ordinary differential equations (ODEs) that arise from the semi-discretization of a partial differential equation (PDE).

2 Theoretical exercises

1. K&C, 8.1.12, 8.1.13, 8.1.19 pp. 528-529.

   Solution. (8.1.12) Use Theorem 1, p.525 with $f(t, x) = 1 + x + x^2 \cos t$, $f \in C(\mathbb{R} \times \mathbb{R})$, i.e., f continuous everywhere. Consider rectangle $R = \left\{(t, x)\colon |t| \leqslant \alpha = \frac{1}{3},\, |x| \leqslant \beta = 1\right\}$. Over R, $M = \|f\|_\infty = \max_{(t,x)\in R} |f(t, x)| = 3$. Theorem 1 states that solution exists for $|t| \leqslant \min\left(\alpha = \frac{1}{3}, \frac{\beta}{M} = \frac{1}{3}\right) = \frac{1}{3}$, q.e.d.

   (8.1.13) Both $x_1(t) = 0$, and $x_2(t) = \frac{t^2}{4}$ are solutions that satisfy the ODE and initial condition. Note that $f(x) = \sqrt{|x|}$ is continuous, but $\partial f / \partial x$ is not continuous at $t = 0$, hence

conditions of Theorem 2 are not satisfied.

(8.1.19) The function $f(x) = 1 + x^2$, is a polynomial in x, and continuous everywhere $f \in C(\mathbb{R})$. The derivative $f'(x) = 2x$ is not bounded in $\mathbb{R}$, and neither is f Lipschitz continuous since $|f(x_1) - f(x_2)| = |x_1 - x_2|\,|x_1 + x_2|$, and there is no finite $L \in \mathbb{R}$ that ensures $|f(x_1) - f(x_2)| < L\,|x_1 - x_2|$ everywhere. The conditions of Theorem 3 are therefore not met. However, by Theorem 2, a unique solution does exist for $|t| \leqslant \min(\alpha, \beta / M)$ if f and $\partial f / \partial x$ are both continuous in the rectangle $R = \{(t, x)\colon |t| \leqslant 1 = \alpha, |x| \leqslant \beta\}$, with $M = \|f\|_\infty = \max_{|x| \leqslant \beta} |f(x)|$. The solution to the problem is $x(t) = \tan(t)$. Choose $\alpha = \arctan(1) \cong 0.79$, $\beta = 1$, $M = 2$, $\beta / M = \frac{1}{2}$, so the solution is unique for $|t| \leqslant \frac{1}{2}$, over which interval $|\tan(t)| \leqslant 1$. The solution an be constructed for $|t| < \frac{\pi}{2}$, by patching together several intervals over which to apply Theorem 2. (1 Bonus point for this observation)

2. K&C, 8.2.9, p. 538. (Use a symbolic computation package to compute the derivatives needed in the Taylor-series method)

Solution. Differentiate $x(t) = \int_0^t (1 - k \sin^2\theta)^{1/2}\, d\theta$, to obtain the ODE

$$x' = (1 - k \sin^2 t)^{1/2} = f(t).$$

Taylor series method of order 3 is

$$x(t+h) = x(t) + x'(t)h + \frac{1}{2}x''(t)h^2 + \frac{1}{6}x'''(t)h^3$$

Mathematica

```
In[3]:= f[t_,k_]=Sqrt[1 - k Sin[t]^2];
        fp[t_,k_]=Simplify[D[f[t,k],t]/2]
```

$$-\frac{k \sin(t) \cos(t)}{2\sqrt{1 - k \sin^2(t)}}$$

```
In[4]:= fpp[t_,k_]=Simplify[D[f[t,k],{t,2}]/2]
```

$$-\frac{\sqrt{2}\, k\,(k \sin^4(t) - \sin^2(t) + \cos^2(t))}{(k \cos(2t) - k + 2)^{3/2}}$$

```
In[7]:= {f[t,1/2],fp[t,1/2],fpp[t,1/2]} // TableForm
```

$$\sqrt{1 - \frac{\sin^2(t)}{2}}$$

$$-\frac{\sin(t)\cos(t)}{4\sqrt{1 - \frac{\sin^2(t)}{2}}}$$

$$-\frac{\frac{\sin^4(t)}{2} - \sin^2(t) + \cos^2(t)}{\sqrt{2}\left(\frac{1}{2}\cos(2t) + \frac{3}{2}\right)^{3/2}}$$

```
In[18]:= h=0.01; table={}; x=0;
         For[t=0, t<=Pi/2., t += h,
           AppendTo[table,{t,x,EllipticE[t,1/2]}];
           x += h (f[t,1/2] + h (fp[t,1/2]/2 + h fpp[t,1/2]/6)) ];
         TableForm[table]
```

0	0	0
0.01	0.00999996	0.00999992
0.02	0.0199995	0.0199993
0.03	0.0299983	0.0299978
0.04	0.0399956	0.0399947
0.05	0.049991	0.0499896
0.06	0.0599841	0.059982
0.07	0.0699743	0.0699714
0.08	0.0799612	0.0799574
0.09	0.0899442	0.0899393
0.1	0.0999228	0.0999168
0.11	0.109897	0.109889
0.12	0.119865	0.119856
0.13	0.129828	0.129817
0.14	0.139784	0.139772
0.15	0.149733	0.14972
0.16	0.159675	0.15966
0.17	0.16961	0.169592
0.18	0.179536	0.179516
0.19	0.189453	0.189431
0.2	0.199361	0.199337
0.21	0.209259	0.209233
0.22	0.219148	0.219118
0.23	0.229025	0.228993
0.24	0.238891	0.238856
0.25	0.248746	0.248708
0.26	0.258589	0.258548
0.27	0.268419	0.268375
0.28	0.278236	0.278189
0.29	0.28804	0.287989
0.3	0.29783	0.297775
0.31	0.307606	0.307547
0.32	0.317366	0.317304
0.33	0.327112	0.327046
0.34	0.336842	0.336772
0.35	0.346556	0.346482
0.36	0.356253	0.356175
0.37	0.365934	0.365851
0.38	0.375597	0.37551
0.39	0.385242	0.385151
0.4	0.39487	0.394774
0.41	0.404478	0.404378
0.42	0.414068	0.413963
0.43	0.423639	0.423528
0.44	0.433189	0.433074
0.45	0.44272	0.442599
0.46	0.45223	0.452104
0.47	0.461719	0.461588
0.48	0.471187	0.471051
0.49	0.480634	0.480492
0.5	0.490058	0.489911
0.51	0.49946	0.499308
0.52	0.50884	0.508681
0.53	0.518197	0.518032
0.54	0.52753	0.52736
0.55	0.53684	0.536664
0.56	0.546126	0.545944
0.57	0.555387	0.555199
0.58	0.564625	0.56443
0.59	0.573837	0.573636
0.6	0.583024	0.582817
0.61	0.592186	0.591973
0.62	0.601323	0.601103

3. K&C, 8.3.4, p. 546

   Solution. Euler's method is

$$x(t+h) = x(t) + h\,f(t, x(t)) + C_1\,h^2. \tag{1}$$

   For step size $h/2$

$$x\left(t + \frac{h}{2}\right) = x(t) + \frac{h}{2}\,f(t, x(t)) + C_2\left(\frac{h}{2}\right)^2 \tag{2}$$

$$x(t+h) = x\left(t + \frac{h}{2}\right) + \frac{h}{2}\,f\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}\right)\right) + C_3\left(\frac{h}{2}\right)^2 \tag{3}$$

   Assume $C_1 = C_2 = C_3 = C$, to obtain

$$x(t+h) = x(t) + \frac{h}{2}\,f(t, x(t)) + \frac{h}{2}\,f\left(t + \frac{h}{2}, x(t) + \frac{h}{2}\,f(t, x(t))\right) + C\,\frac{h^2}{2}. \tag{4}$$

   Eliminate C using equations (1),(4) to obtain

$$x(t+h) = x(t) + h\,f\left(t + \frac{h}{2}, x(t) + \frac{h}{2}\,f(t, x(t))\right).$$

4. K&C, 8.3.5, p. 546

   Solution. See solution to August 2016 SciComp exam.

5. K&C, 8.4.4 and 8.4.5, p. 555

   Solution. Consider

$$x_{n+1} = x_n + h(a\,f_n + b\,f_{n-1} + c\,f_{n-2} + d\,f_{n-3})$$

and choose $h=1,\ f\in\{1,t,t(t+1),t(t+1)(t+2)\}$ to obtain method coefficients

6. K&C, 8.4.12, p. 556

   Solution. Choose $h=1$, and replace $x(t)\in\{p_0(t)=1,\ p_1(t)=t-1,\ p_2(t)=(t-1)t,\ p_3(t)=(t-1)t(t+1), p_4(t)=p_3(t)(t+2)\}$ successively in formula

$$x_{n+1}=(1-A)x_n+Ax_{n-1}+\frac{h}{12}[(5-A)x'_{n+1}+8(1+A)x'_n+(5A-1)x'_{n-1}], \tag{5}$$

   with $x_n=x(n)$, at $n=0$ to obtain

$$\begin{aligned} p_0:&\ 1=1 && \checkmark\\ p_1:&\ 0=A-1-2A+\frac{1}{12}[5-A+8+8A+5A-1] && \checkmark\\ &\ \ldots \end{aligned}$$

The tedium of the calculations is alleviated by symbolic computation. Define a basis set, and define functions `e[p,a],d[p,a]` that evaluate $p(a)$, and the derivative $p'(t=a)$ (these functions work by delaying evaluation until a specific argument is given for `p`, computing the derivative and replacing the argument `a`).

Mathematica

```
In[8]:= p0[t_]=1; p1[t_]=t-1; p2[t_]=(t-1)t; p3[t_]=p2[t](t+1);
        p4[t_]=p3[t](t+2);
        e[p_,a_] := p /. t->a;
        d[p_,a_] := D[p,t] /. t->a;
        {d[p2[t],1],d[p2[t],2]}
```

$\{1,3\}$

```
In[9]:=
```

Define a function to evaluate (5) for general $x(t)$, and evaluate for $x\in\{p_0,p_1,p_3,\ldots\}$

```
In[9]:= eqn[x_] := FullSimplify[e[x,1] == (1-A) e[x,0] + A e[x,-1] + (1/12)((5-
        A)d[x,1]+8(1+A)d[x,0]+(5A-1)d[x,-1])];
        eqn[p0[t]]
```

True

```
In[10]:= eqn[p1[t]]
```

True

```
In[11]:= eqn[p2[t]]
```

True

```
In[12]:= eqn[p3[t]]
```

True

```
In[13]:= eqn[p4[t]]
```

$A=1$

From the above deduce $A=1$, $m=4$.

3 Implementation and analysis

Consider the following PDE, initial and boundary conditions that define $u(t,\,x)$, u: $[0,\,T]\times[0,\,\pi]\to\mathbb{R}$,

$$\begin{aligned}&\frac{\partial u}{\partial t}=\nabla^2 u=\frac{\partial^2 u}{\partial x^2},\\&u(t,x=0)=0, u(t,x=\pi)=0,\\&u(t=0,x)=\sin(x)\exp\Big[-2\Big(x-\frac{\pi}{2}\Big)^2\Big]=f(x).\end{aligned}\tag{6}$$

Solution. Problem can be solved analytically by separation of variables (see class discussion), leading to

$$u(t,x)=\sum_{k=1}^{\infty}a_k\sin(kx)e^{-k^2t}$$

with coefficients a_k determined from initial condition

$$u(t=0,x)=\sum_{k=1}^{\infty}a_k\sin(kx)=f(x)$$

1. Obtain a system of ODEs from (1) by introducing the functions $U_k(t)=u(t,x_k)$, $x_k=kh$, $h=\pi/m$, and replacing the x-derivative by a centered finite difference approximation

$$\frac{\partial^2 u}{\partial x^2}(t,x_k)\cong\frac{U_{k+1}(t)-2U_k(t)+U_{k-1}(t)}{h^2}.$$

Write the resulting system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}=\boldsymbol{A}\boldsymbol{U},\boldsymbol{U}(t)=(\ U_1\ \ U_1\ \ \ldots\ \ U_{m-1}\ ).\tag{7}$$

Solution. Details were presented in class. The result is $\boldsymbol{A}=\frac{1}{h^2}\mathrm{diag}([\ 1\ \ -2\ \ 1\ ])\in\mathbb{R}^{(m-1)\times(m-1)}$.

2. Write the code to solve (2) using the forward Euler method. Use the code to solve the system for $m=128,256$. Experiment with various choices of the time step.

Solution. Define the discretization

```
octave> m=128; h=pi/m; dt=0.0001; x=(0:m)*h; n=10000; T=n*dt; xi=(1:m-1)*h;
octave>
```

Define the initial condition

```
octave> function y=f(x)
          y=sin(x).*exp(-(x-pi/2).^2);
        end;
        U0=f(xi); U0b=[0 U0 0]; sig=dt/h^2;
octave>
```

Carry out the time steps. Organize the Euler method as a Runge-Kutta method for later computations

$$\begin{aligned}&\boldsymbol{U}^{n+1}=\boldsymbol{U}^n+\boldsymbol{K}_1\\&\boldsymbol{K}_1=\delta_t\,\boldsymbol{A}\boldsymbol{U}^n\Rightarrow(K_1)_i=\frac{\delta_t}{h^2}(U_{i+1}^n-2U_i^n+U_{i-1}^n)\end{aligned}$$

```
octave> for k=1:n
          U0b=[0 U0 0]; U0l(1:m-1)=U0b(3:m+1); U0r(1:m-1)=U0b(1:m-1);
          K1=sig*(U0l-2*U0+U0r);
          U1=U0+0.5*K1;
          U1b=[0 U1 0]; U1l(1:m-1)=U1b(3:m+1); U1r(1:m-1)=U1b(1:m-1);
          U0=U1;
        end;
octave>
```

Plot the result

```
octave> plot(xi,U1); xlabel('x'); ylabel('u');
octave> cd /home/student/courses/MATH661/homework;
        print -dpng HW5CP52plot1.png;
octave>
```

Organize the computations into a single function call to enable experiments with the time step

```
octave> function [xi,U1]=Euler(m,dt,n)
          h=pi/m; x=(0:m)*h; T=n*dt; xi=(1:m-1)*h;
          U0=f(xi); U0b=[0 U0 0]; sig=dt/h^2;
          for k=1:n
            U0b=[0 U0 0]; U0l(1:m-1)=U0b(3:m+1); U0r(1:m-1)=U0b(1:m-1);
            K1=sig*(U0l-2*U0+U0r);
            U1=U0+0.5*K1;
            U1b=[0 U1 0]; U1l(1:m-1)=U1b(3:m+1); U1r(1:m-1)=U1b(1:m-1);
            U0=U1;
          end;
        end;
octave>
```

Carry out computations for $\delta_t = 2^{-13}, 2^{-14}, 2^{-15}$, to $T = 1$, for $m = 128, 256$

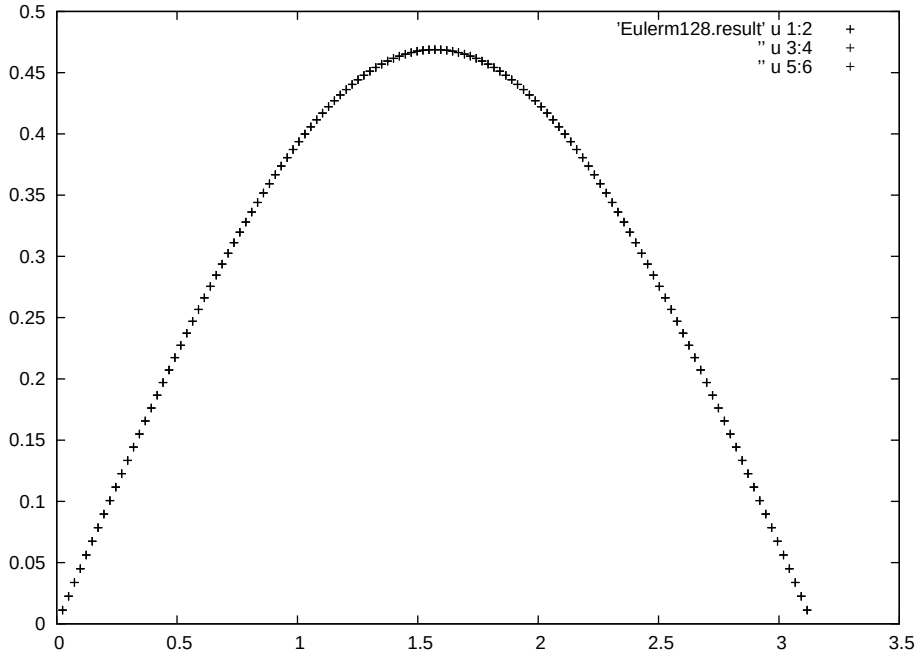
```
octave> [xd3m128,Ud3m128]=Euler(128,2**(-13),2**13);
octave> [xd3m256,Ud3m256]=Euler(256,2**(-13),2**13);
octave> [xd4m128,Ud4m128]=Euler(128,2**(-14),2**14);
octave> [xd4m256,Ud4m256]=Euler(256,2**(-14),2**14);
octave> [xd5m128,Ud5m128]=Euler(128,2**(-15),2**15);
octave> [xd5m256,Ud5m256]=Euler(256,2**(-15),2**15);
octave>
```

Save the data to a file for plotting

```
octave> out128=[xd3m128' Ud3m128' xd4m128' Ud4m128' xd5m128' Ud5m128'];
octave> save('Eulerm128.result','out128','-ascii');
octave> out256=[xd3m256' Ud3m256' xd4m256' Ud4m256' xd5m256' Ud5m256'];
octave> save('Eulerm256.result','out256','-ascii');
octave>
```

Plot in Gnuplot

```
GNUplot] cd '/home/student/courses/MATH661/homework';
         plot 'Eulerm128.result' u 1:2, '' u 3:4, '' u 5:6
```



GNUplot]

The results agree quite well. As discussed in class for large time steps, the method can become unstable.

3. Repeat using the fourth-order Runge Kutta method.

   Solution. Here's an implementation of RK4 for this case

```
octave> m=128; h=pi/m; dt=0.0001; x=(0:m)*h; n=10000; T=n*dt; xi=(1:m-1)*h;
octave> function y=f(x)
          y=sin(x).*exp(-(x-pi/2).^2);
        end;
octave> U0=f(xi); U0b=[0 U0 0]; sig=dt/h^2;
octave> for k=1:n
          U0b=[0 U0 0]; U0l(1:m-1)=U0b(3:m+1); U0r(1:m-1)=U0b(1:m-1);
          K1=sig*(U0l-2*U0+U0r);
          U1=U0+0.5*K1;
          U1b=[0 U1 0]; U1l(1:m-1)=U1b(3:m+1); U1r(1:m-1)=U1b(1:m-1);
          K2=sig*(U1l-2*U1+U1r);
          U1=U0+0.5*K2;
          U1b=[0 U1 0]; U1l(1:m-1)=U1b(3:m+1); U1r(1:m-1)=U1b(1:m-1);
          K3=sig*(U1l-2*U1+U1r);
          U1=U0+K3;
          U1b=[0 U1 0]; U1l(1:m-1)=U1b(3:m+1); U1r(1:m-1)=U1b(1:m-1);
          K4=sig*(U1l-2*U1+U1r);
          U1=U0+(K1+2*K2+2*K3+K4)/6.;
          U0=U1;
        end;
octave> plot(xi,U1);
octave> print -deps t.eps
```

```
texmacs
```

4. Repeat using the fourth-order Runge-Kutta-Gill method.

   Solution. As above, with different weights in the Runge-Kutta method.

Extra credits:

EC3: Repeat using the adaptive Runge-Kutta-Fehlberg method

EC4: Compute the analytical solution using separation of variables. Construct convergence plots, i.e., plots of the log of the norm of the error $\lg\|\boldsymbol{e}\|$ as a function of $\lg(h)$ for all of the methods considered (forward Euler, 3 Runge-Kutta methods).