MATH 225 Linear Algebra and Differential Equations

Fall 2007

Homework 1 Solutions

Problem 1

Exact solution is $y=e^{-t}$

euler.m

```
function [t,y] = euler(diffeq,tn,h,y0)
t=(0:h:tn)';
n=length(t);
y=y0*ones(n,1);

for(j=2:n)
    y(j)=y(j-1)+h*feval(diffeq,t(j-1),y(j-1));
end
```

rhs.m

```
function result = rhs(t,y)
result = -y;
```

midpoint.m

```
function [t,y]=midpoint(diffeq, tn, h, y0)

t=(0:h:tn)';
n=length(t);
y=y0*ones(n,1);
h2=h/2;

for j=2:n
    k1=feval(diffeq,t(j-1),y(j-1));
    k2=feval(diffeq,t(j-1)+h2,y(j-1)+h2*k1);
    y(j)=y(j-1)+h*k2;
end
```

compareEM.m

```
function compareEM

tn=1; y0=1;
```

```
fprintf('\n   h       timeE         errE       timeM         errM\n');
for h=[0.2 0.1 0.05 0.025 0.0125 0.00625]
    tic;
    for i=1:10000
        euler('rhs', tn, h, 1);
    end
    [te,ye] = euler('rhs', tn, h, 1);
    flopse=toc;

    tic;
    for i=1:10000
        midpoint('rhs', tn, h, 1);
    end
    [tm,ym] = midpoint('rhs', tn, h, 1);
    flopsm=toc;

    yex = y0*exp(-te);
    erre = max(abs(ye-yex)); errm = max(abs(ym-yex));
    fprintf('%8.5f %7d %11.2e %7d %11.2e\n',h,flopse,erre,flopsm,errm);
end
```

Output of the compareEm.m is

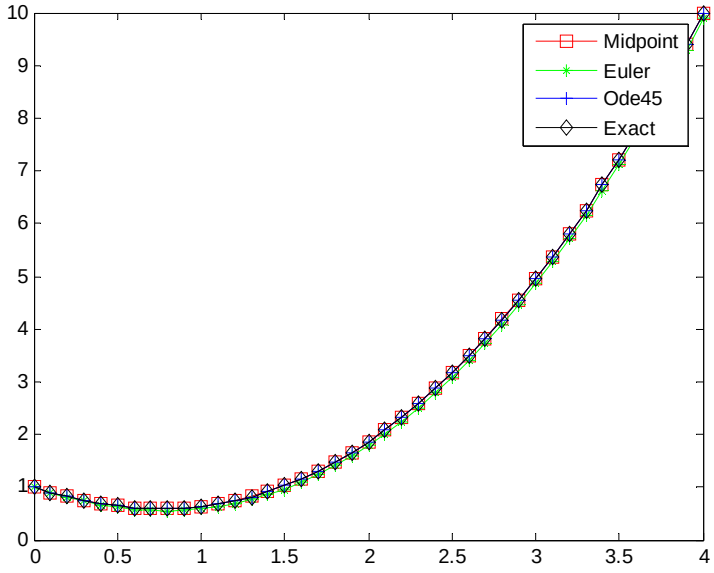
```
  h       timeE            errE       timeM         errM
 0.20000 7.010000e-001   4.02e-002 9.720000e-001   2.86e-003
 0.10000 9.410000e-001   1.92e-002 1.792000e+000   6.62e-004
 0.05000 1.733000e+000   9.39e-003 3.375000e+000   1.59e-004
 0.02500 3.345000e+000   4.65e-003 6.619000e+000   3.90e-005
 0.01250 6.590000e+000   2.31e-003 1.303800e+001   9.67e-006
 0.00625 1.346000e+001   1.15e-003 3.050400e+001   2.41e-006
```

Comments: We gain significant accuracy with the midpoint method. Event though is takes twice computational time compared to simple Euler, it provides 10 to 1000 times less error. Hence, works better.

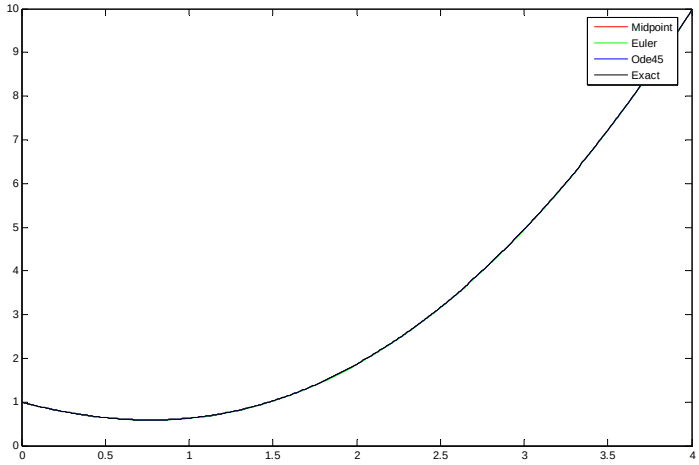
Problem 2

a) Exact solution is $y = x^2 - 2x + 2 - e^{-x}$

For step size 0.1



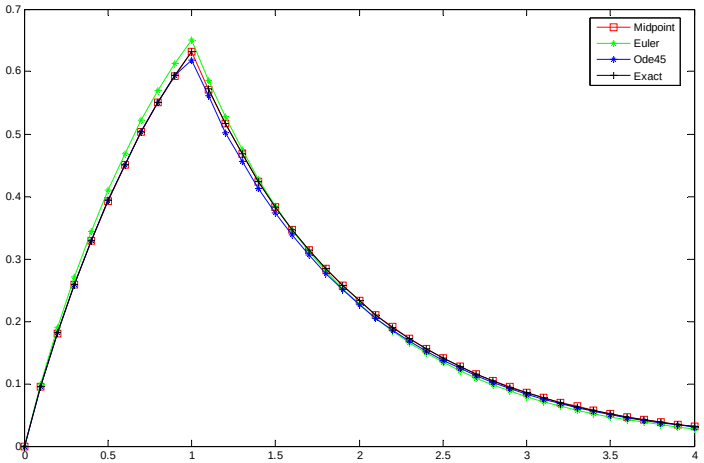
For step size 0.01



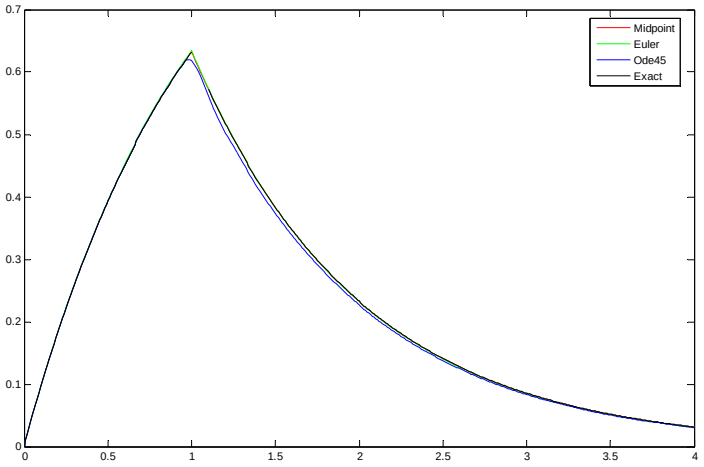
b) Exact solution is

$$y = \begin{cases} (e^{-1}-1)e^{-x} & x \le -1 \\ 1-e^{-x} & -1 \le x \le 1 \\ (e-1)e^{-x} & 1 \le x \end{cases}$$

For step size = 0.1:



For step size 0.01:



Comment: The numerical methods work better for the first differential equation because it has a smoother solution curve than the second one. The solution of the second equation has a discontinuity at x=1 and this makes numerical estimation harder. It is possible to see that problem at that discontinuity by zooming out into that region. The solution of the midpoint method is closest to the exact solution and Euler method performs worst. However, note that the performance benefits of the midpoint method comes with a cost, it takes longer to compute as it can be seen from the first question. Also, using a smaller step size gives better numerical solutions.

Problem 3

Solution is $\frac{1}{1+100x^2}$ for both I.C.s. 2 pts

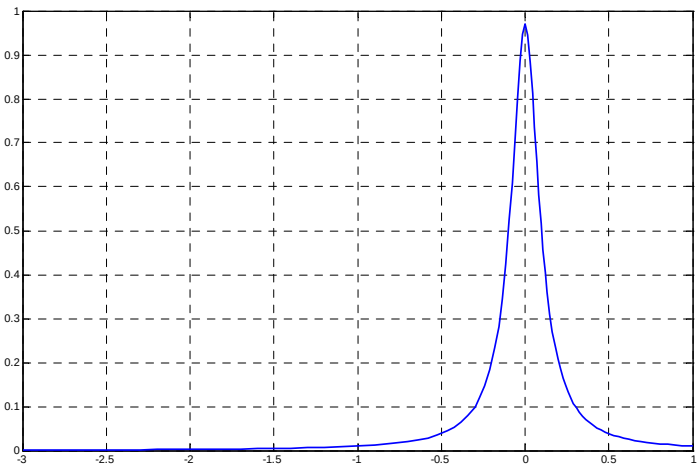
Codes

rhs.m

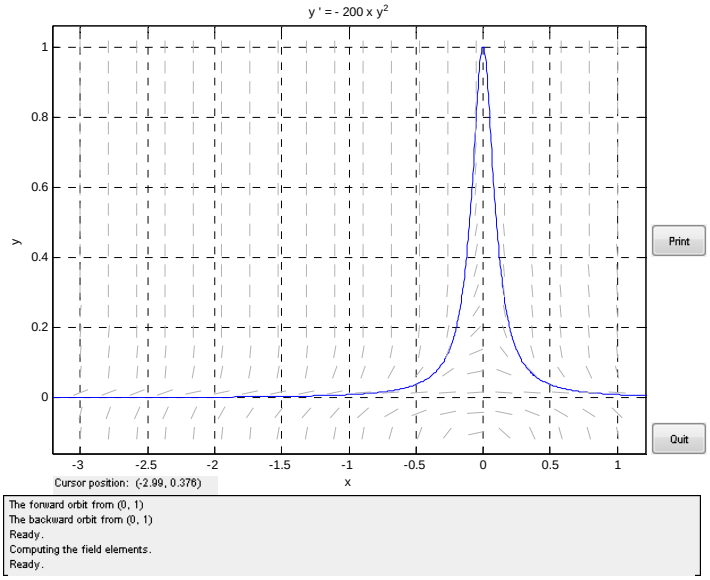
```
function result = rhs(x,y)
result = -200*x*y^2;
```

```
>> [x,y]=ode45(@rhs,[-3 1],1/901);
>> plot(x,y)
>> plot(x,y,'LineWidth',2)
>> grid
```

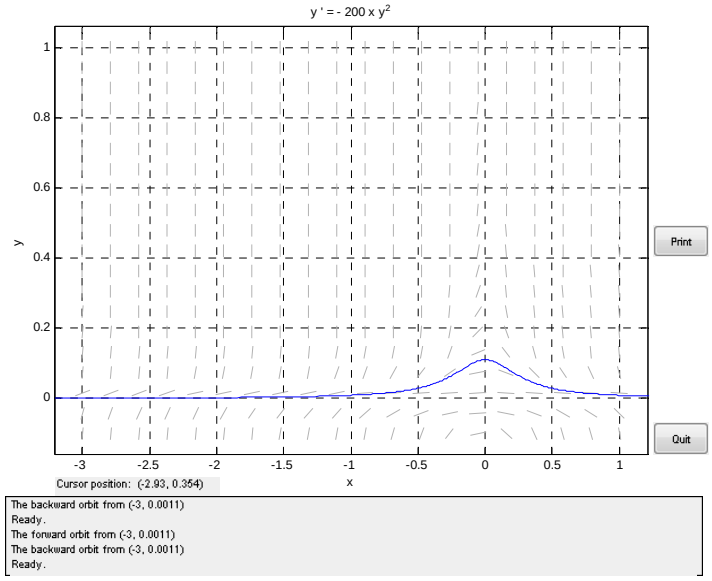
From ode45



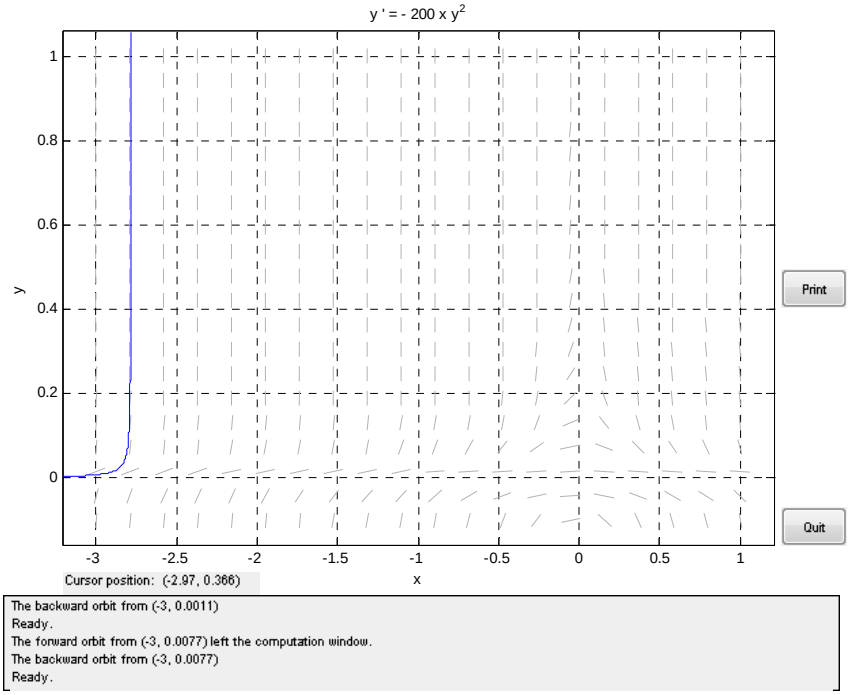
Dfield with y(0)=1



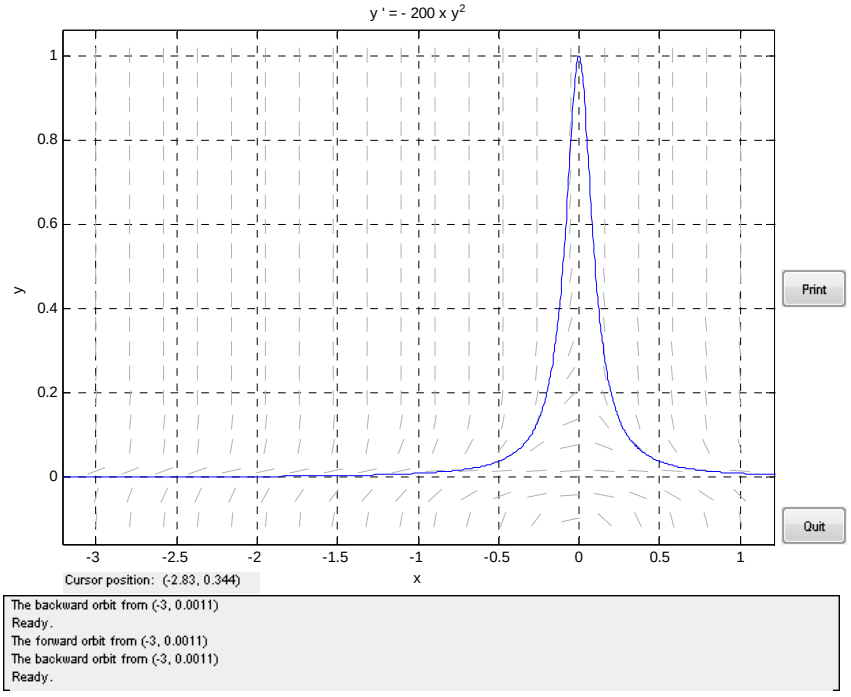
Dfield with y(-1)=0.0011≈1/901 (keyboard input)



Dfield with a mouse click



Dfield with y(-1)=1/901 (keyboard input)



Comment: Dfield can take the initial values either by keyboard inputs (Options -> Keyboard Input) or by mouse click on the plot. It is very difficult to give exact initial condition by mouse click. The difficulty here is that a very slight change in the initial condition changes the solution dramatically. This is because of the fact that direction field changes very rapidly around y=0.

Problem 4

Solution is $y = \frac{x^2}{4}$.

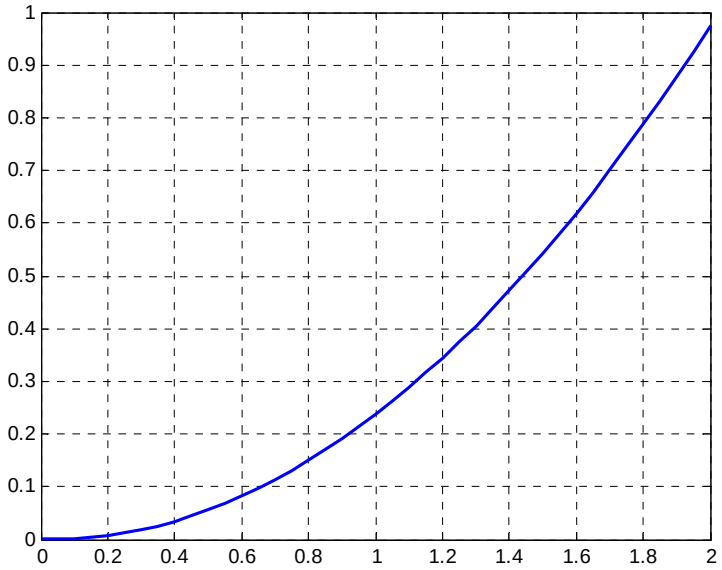
The ode45 routine finds y=0 solution. If you set the initial condition exactly y(0)=0, dfield also gives the 0 solution.

Comment: Note that y(x)=0 is a singular solution. Since the direction fields at y=0 have 0 slope, ODE45 and dfield finds the singular solution.

But if you set y(0) a small number, you get the $y = \frac{x^2}{4}$ solution.

```
[x,y]=ode45(@rhs,[0 2],0.0000001);
plot(x,y,'LineWidth',2);
```

yields



Dfield with a mouse click: