CISC271
Fall 2006
Homework for week 11
in preparation for quiz 5
Solutions

The following questions are from Recktenwald Chapter 12.

12-10 This is my m-file for Heun's method.

```
function [t,y] = odeHeunDR(diffeq,tn,h,y0)
% odeEuler  Heun's method for integration of a single, first order ODE
%
% Synopsis:   [t,y] = HeunDR(diffeq,tn,h,y0)
%
% Input:      diffeq = (string) name of the m-file that evaluates the right
%                      hand side of the ODE written in standard form
%             tn  = stopping value of the independent variable
%             h   = stepsize for advancing the independent variable
%             y0  = initial condition for the dependent variable
%
% Output:     t = vector of independent variable values:  t(j) = (j-1)*h
%             y = vector of numerical solution values at the t(j)

t = (0:h:tn)';           %  Column vector of elements with spacing h
n = length(t);           %  Number of elements in the t vector
y = y0*ones(n,1);        %  Preallocate y for speed

%   Begin Heun's algorithm; j=1 is the initial condition
for j=2:n
    K1 = feval(diffeq,t(j-1),y(j-1));
    K2 = feval(diffeq,t(j),y(j-1) + h*K1);
```

```
   y(j) = y(j-1) + h/2*(K1+K2);
end
```

I ran it on equation (12.11) with varying h values using the following m-file:

```
function M12Q10
eq1211 = inline('t-2*y','t','y');
h = [ 0.2 0.1 0.05 0.025];
maxabserr = [0 0 0 0 ];

for i = 1:4
    [t,y] = odeHeunDR(eq1211,2,h(i),1);
    yex = 1/4*(2.*t - 1 + 5*exp(-2.*t));
    maxabserr(i) = norm(y-yex,inf);
end
fprintf('h values           %10g %10g %10g %10g \n',h);
fprintf('max absolute errors  %g %g %g %g \n',maxabserr)
```

and got the following output:

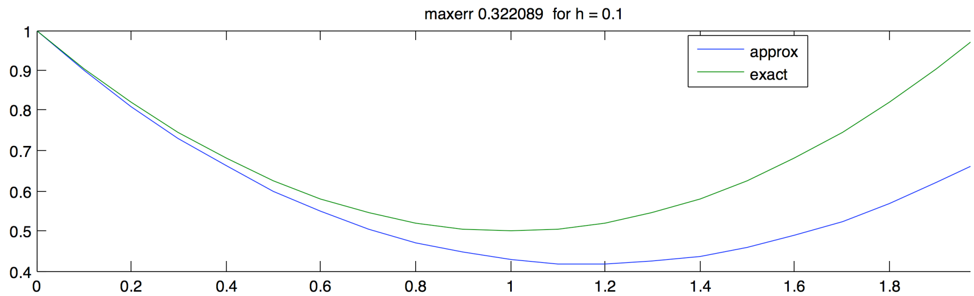
```
h values                0.2           0.1          0.05       0.025
max absolute errors  0.0165472 0.0035755 0.00082693 0.000198976
```

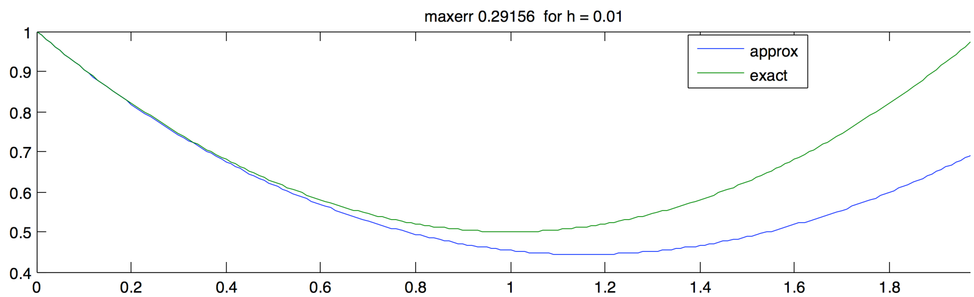
The ratio between successive h values is 1/2, Whereas the ratio between successive errors is 1/4. Heun's method is a $O(h^2)$ method, and this appears to be confirmed empirically with this example.

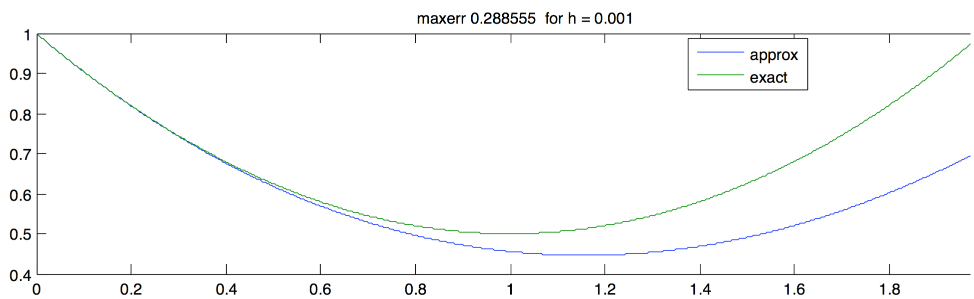
12-12 a) Evaluating $t^2/\alpha - y(t)$ for $y(t) = 1/\alpha(t^2 - 2t + 2 + (\alpha - 2)e^{-t)})$ is equal to the first derivative of $y(t)$ which is $1/\alpha(2t - 2 - (\alpha - 2)e^{-t)})$.

12-13 a) I used the following Matlab m-file.

```
function Q13a
h = [.1 .01 .001]
diffy = inline('t^2/3 - y','t','y')
for i = 1:3
  [t,y] = odeEuler(diffy,2,h(i),1);
  yex = 1/3.*(t.^2 - 2.*t + 2+ exp(-t));
  maxabserr = norm(y-yex,inf);
  subplot(3,1,i);   plot(t,y, t, yex); legend('approx','exact','Location', 'Best');
  title(sprintf('maxerr %g  for h = %g',maxabserr,h(i)));
end
```

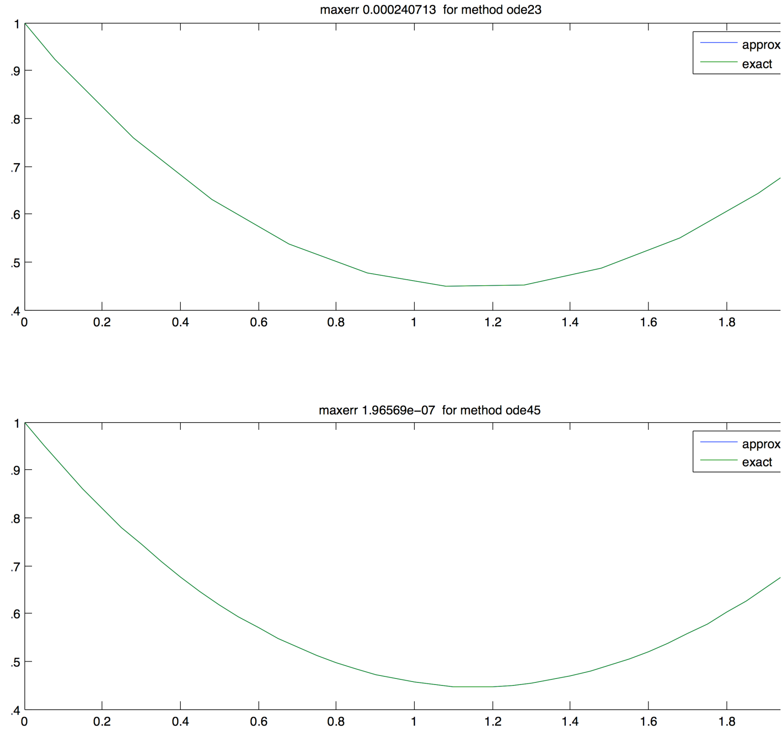






12-15 a) I used the following Matlab m-file. On the plots both ode23 and ode45 return very accurate solutions. However, the printed values of the errors show that ode45 is much more accurate.

```
function Q15a
diffy = inline('t^2/3 - y','t','y')
s = [ 'ode23'; 'ode45'];
for i = 1:2
    if i == 1
        [t,y] = ode23(diffy,2,1);
    else
        [t,y] = ode45(diffy,2,1);
    end
    yex = 1/3.*(t.^2 - 2.*t + 2 + exp(-t));
    maxabserr = norm(y-yex,inf);
    subplot(2,1,i);   plot(t,y, t, yex); legend('approx','exact',
                                                'Location', 'Best');
    title(sprintf('maxerr %g  for method %s',maxabserr,s(i,:)));
end
```



12-18 a) I ran the demoPredPrey program. No matter what the initial conditions, this pair of differential equations will always yield period functions.