NAME:

MATH 353: Engineering Mathematics III

Spring'13 Midterm exam #2 (solutions) April 17

1. (20 points) Write the output of the following command lines in Matlab:

```
linspace(0,1,5)
```

Five equally spaced points from 0 to 1, including both limits

```
0      0.25      0.5      0.75     1
```

```
10*2.^(0:4)
```

We are computing 10×2^i for $i = 0, \ldots, 4$,

```
10      20     40     80     160
```

```
a=1; b=3;
n=10.^(1:4);
h=(b-a)./n;
```

We first compute the powers $n = 10^i$ for $i = 1, \ldots, 4$, and then $(b-a)/n$. The fact that there's a semicolon does not affect the fact that these quantities are computed. The results are not shown on screen though.

```
n    =    [ 10    1000   1000     10000]
h    =    [ 0.2   0.02     0.002     0.0002]
```

```
f = @(t) t.^2+1;
sum( f(0:0.5:2) );
```

We evaluate the function f in the points $[0\,0.5\,1\,1.5\,2]$ and then add the result:

```
12.5
```

```
doessomething =@(f,x0,h) (f(x0+h)-f(x0-h))./(2*h);
g= @(t) t.^3+3;
doessomething(g,1,[0.1 0.01])
```

$$\frac{g(1.1)-g(0.9)}{0.2}=3.01, \qquad \frac{g(1.01)-g(0.99)}{0.02}=3.0001$$

2. (10 points) Show that

$$\frac{f(x_0+h)-f(x_0-h)}{2h}=f'(x_0)+\mathcal{O}(h^2).$$

Using Taylor expansions

$$\begin{aligned} f(x_0+h) &= f(x_0)+hf'(x_0)+\tfrac{1}{2}h^2f''(x_0)+\tfrac{1}{6}h^3f'''(c), \\ f(x_0-h) &= f(x_0)-hf'(x_0)+\tfrac{1}{2}h^2f''(x_0)z-\tfrac{1}{6}h^3f'''(d) \end{aligned}$$

(c is an unknown point between x_0 and x_0+h, and d is an unknown point between x_0-h and x_0), we can write

$$f(x_0+h)-f(x_0-h)=2hf'(x_0)+\tfrac{1}{6}h^2(f'''(c)+f'''(d)),$$

that is

$$\left|\frac{f(x_0+h)-f(x_0-h)}{2h}-f'(x_0)\right|=\tfrac{1}{6}h^2|f'''(c)+f'''(d)|\le\tfrac{1}{6}h^2\max|f'''(x)|.$$

3. (20 points) Apply the midpoint, trapezoid and Simpson rules (the simple ones) to approximate the following integral

$$\int_0^1(x^3-2x^2)\mathrm{d}x.$$

Is any of the three approximations the exact value of the integral? Why? (**Hint.** You don't need to compute the exact value of the integral to answer this question.)

Note that we need to evaluate the polynomial function $p(x)=x^3-2x^2$ on three points:

$$p(0)=0, \qquad p(\tfrac{1}{2})=(\tfrac{1}{2})^3-2(\tfrac{1}{2})^2=-\tfrac{3}{8}, \qquad p(1)=-1.$$

We then get three possible approximations:

$$\begin{aligned} \text{(Midpoint)}: &\quad \int_0^1(x^3-2x^2)\mathrm{d}x\approx 1\,p(\tfrac{1}{2})=-\tfrac{3}{8} \\ \text{(Trapezoid)}: &\quad \int_0^1(x^3-2x^2)\mathrm{d}x\approx \tfrac{1}{2}\Big(p(0+p(1)\Big)=-\tfrac{1}{2} \\ \text{(Simpson)}: &\quad \int_0^1(x^3-2x^2)\mathrm{d}x\approx \tfrac{1}{6}\Big(p(0)+4p(\tfrac{1}{2})+p(1)\Big)=-\tfrac{5}{12}. \end{aligned}$$

Simpson's rule gives the exact value since $p(x)$ is a ppolynomial of degree three and Simpson's rule has degree of precision three.

4. (10 points) Check the degree of precision of the numerical integration formula

$$\int_0^1 f(x)\mathrm{d}x = \tfrac{1}{4}f(0) + \tfrac{3}{4}f(\tfrac{2}{3}).$$

We need to check how far we can go with $f(x) = 1, x, x^2, ...$ until the numerical integration formula differs from the exac value:

$$\int_0^1 1\mathrm{d}x = 1 = \tfrac{1}{4}\,1 + \tfrac{3}{4}\,1,$$
$$\int_0^1 x\mathrm{d}x = \tfrac{1}{2} = \tfrac{1}{2}\,0 + \tfrac{3}{4}\,\tfrac{2}{3},$$
$$\int_0^1 x^2\mathrm{d}x = \tfrac{1}{3} = \tfrac{1}{4}\,0^2 + \tfrac{3}{4}\,(\tfrac{2}{3})^2,$$
$$\int_0^1 x^3\mathrm{d}x = \tfrac{1}{4} \neq \tfrac{1}{4}\,0^3 + \tfrac{3}{4}(\tfrac{2}{3})^3$$

The degree of precision is therefore two.

5. (20 points) Here's the code of the composite Simpson rule.

```
function integral=simpsonrule(f,interval,m)

% Shf=simpsonrule(f,[a b],m)
%
% Input:
%     f     : vectorized function of one variable
%     [a b] : vector with limits of integration interval
%     m     : number of subdivisions
% Output
%     Shf   : approx of \int_a^b f(x) dx with composite Simpson rule

x=linspace(interval(1),interval(2),m+1);
h=(interval(2)-interval(1))/m;
mdpt=0.5*(x(2:end)+x(1:end-1));
integral=(h/6)*(f(x(1))+2*sum(f(x(2:end-1)))+f(x(end))+4*sum(f(mdpt)));

return
```

(a) Modify it to be the composite midpoint rule.

Substitute only one line:

```
integral=h*sum(f(mdpt))
```

(b) Modify it to be the composite trapezoid rule. Substitute only one line:

```
integral=(h/2)*( f(x(1))+2*sum(f(x(2:end-1)))+f(x(end)) );
```

The line where we compute `mdpt` can be eliminated.

6. (10 points) Consider the differential equation

$$(y+1)\,y' + 3t\,y = 0, \qquad t \geq 0, \qquad y(0) = 1.$$

Compute two steps of Heun's method (the explicit trapezoidal method) using time step $h = 0.1$. Give all the results with five significant digits.

We write the equation in standard form

$$y' = -\frac{3ty}{y+1} = f(t,y) \qquad y(0) = 1.$$

The $w_0 = 1$. To compute w_1, we compute the internal stages

$$\begin{aligned} k_1 &= f(t_0, w_0) = f(0,1) = 0, \\ k_2 &= f(t_0 + h, w_0 + h\,k_1) = f(0.1, 1) = -0.15, \end{aligned}$$

and then do the time-stepping process

$$w_1 = w_0 + \tfrac{h}{2}(k_1 + k_2) = 0.9925.$$

We repeat the process starting at $(t_1, w_1) = (0.1, 0.9925)$

$$\begin{aligned} k_1 &= f(t_1, w_1) = f(0.1, 0.9925) \approx -0.14944, \\ k_2 &= f(t_1 + h, w_1 + h\,k_1) \approx f(0.2, 0.9925 + 0.1 \times (-0.14944)) \\ &\approx f(0.2, 0.97756) \approx -0.29660, \end{aligned}$$

and do the time-step

$$w_2 = w_1 + 0.05 \times (k_1 + k_2) \approx 0.97020$$

7. (20 points) Here is the code of Heun's method.

```
function [w,t]=heun(f,tinit,tfinal,yinit,n)

% [w,t]=heun(f,tinit,tfinal,yinit,n)
% Input:
%     f      : function of two variables
%     tinit  : initial time
%     tfinal : final time
%     yinit  : initial value (at t=tinit)
%     n      : number of time-steps
% Output:
%     w      : vector with n+1 components w_i \approx y(t_i)
%              (Heun's method for y'=f(t,y), y(tinit)=yinit
```

```
%     t      : vector with n+1 components (times)

t=linspace(tinit,tfinal,n+1);
h=(tfinal-tinit)/n;
w=zeros(1,n+1);       % room to store the solution
w(1)=yinit;           % initial value
for i=1:n
    k1=f(t(i),w(i));
    k2=f(t(i)+h,w(i)+h*k1);
    w(i+1)=w(i)+h*0.5*k1+h*0.5*k2;
end
return
```

(a) Modify it for the midpoint method.

We only need to modify two lines:

```
k2=f(t(i)+h/2, w(i)+h*k1/2);
w(i+1)=w(i)+h*k2;
```

(b) Modify it for Euler's method.

We only need to modify one line:

```
w(i+1)=w(i)+h*k1;
```

The line where we define `k2` can be eliminated.

8. (10 points) We have run three different methods for a differential equation in the interval $0 \leq t \leq 1$. We have computed the maximum error as a function of $h = 1/n$, where n is the number of time steps. We run experiments with $n = 1, 2, 4, 8, 16$. Each of the three columns contains all the experiments for a method and we are expecting

$$E_h = \max_{0 \leq i \leq n} |w_i - y(t_i)| = \mathcal{O}(h^p)$$

($y(t)$ is the exact solution, $t_i = 0 + i\,h$, and $w_i \approx y(t_i)$ is the approximated solution). What is p for each column/method?

```
0.569560557758917   0.000579323417547   0.140859085770477
0.267769111808837   0.000037013462701   0.035649264005780
0.129466548687958   0.000002326240851   0.008940076098471
0.063618163463359   0.000000145592846   0.002236763705256
0.031529636406385   0.000000009102726   0.000559300120949
```

In each column we divide one number but the number below. This is the table that we get...

```
2.1271   15.6517    3.9512
2.0682   15.9113    3.9876
2.0351   15.9777    3.9969
2.0177   15.9944    3.9992
```

The process $h \mapsto \frac{h}{2}$ transforms errors $E_h \approx C\,h^p \mapsto E_{h/2} \approx C\,h^p 2^{-p}$, so $E_h/E_{h/2} \approx 2^p$. This means that the first column is order one ($p = 1$), the second column is order four ($p = 4$) and the third column order two ($p = 2$). Only a few of these divisions have to be computed to see the decreasing pattern of the error on each column.

9. (20 points) You are next given the code for the Runge-Kutta method of order four, but the help lines have been deleted, so you need to read the code to figure out what kind of input and output this code needs.

```
function [w,t]=rk4(f,tinit,tfinal,yinit,n)

% Help lines deleted

t=linspace(tinit,tfinal,n+1);
h=(tfinal-tinit)/n;
w=zeros(n+1,length(yinit));       % create room to store the solution
w(1,:)=yinit;                     % initial value
for i=1:n
    k1=f(t(i),w(i,:));
    k2=f(t(i)+h/2,w(i,:)+h/2*k1);
    k3=f(t(i)+h/2,w(i,:)+h/2*k2);
    k4=f(t(i)+h,w(i,:)+h*k3);
    w(i+1,:)=w(i,:)+(h/6)*(k1+2*k2+2*k3+k4);
end
return
```

We want to solve the system of ODE:

$$\begin{aligned} y_1' &= y_1\,y_2 + t^2, \qquad 2 \le t \le 5, \\ y_2' &= y_1^2 + \sin t, \qquad 2 \le t \le 5, \\ y_1(2) &= 3.1, \\ y_2(2) &= -1. \end{aligned}$$

(a) Write down (in Matlab) what would be the input (all necessary data) if we want to use the code `rk4` with time steps of length $h = 0.1$.

```
f = @(t,y) [y(1)*y(2)+t^2,  y(1)^2+sin(t)];
n=30;        % tfinal-tinit=3, h=0.1 implies n=30 steps
rk4(f,2,5,[3.1 -1],30)
```

(b) What kind of output do we have?

`t` is a (row) vector with 31 components. `w` is a 31×2 matrix: its rows contain approximations in the different times (including t_0) and its columns contain the different components of the solution.

(c) If we want to find a approximation of $y_1(t_3)$, where should we look for it?

Since it is the third time-step it's stored in the fourth row. Since it's y_1, it's stored in the first column. Therefore, we have to look for in in `w(4,1)`.

10. (10 points) Write the second order differential equation with initial conditions

$$\theta'' + \frac{7}{2}\theta + 2\theta' = \sin t, \qquad \theta(0) = 1, \qquad \theta'(0) = -1,$$

as an initial value problem for a system of first order equations.

We use the variables

$$y_1 = \theta, \qquad y_2 = \theta',$$

so that

$$y_1' = y_2, \qquad y_2' = -\tfrac{7}{2}y_1 - 2y_2 + \sin t$$

and the initial values are

$$y_1(0) = 1, \qquad y_2(0) = -1.$$