Instructor: Emre Mengi

Math 304 (Spring 2012)

Study Guide for Weeks 11-14
This homework concerns the following topics.

- Composite Newton-Cotes
- Gaussian quadrature
- Euler's Method
- Trapezoidal method
- Order of a numerical method for the solution of ordinary differential equations.
  - Order of Euler's method is one.
  - Order of trapezoidal method is two.
- Global convergence and error of a numerical method for the solution of ordinary differential equations.
- Explicit Runge-Kutta methods, especially midpoint rule
- Systems of ordinary differential equations
- Higher-order differential equations

Homework 5 (due on May 30th, Wednesday by 5pm)
Pick any five questions out of the twelve questions given below. Turn in the solutions only to these questions. Questions 3, 4 and 12 require Matlab. All other questions must be solved by hand.

For questions **3** and **4** you will need to use the implementations of the Euler's method (`Euler.m`) and midpoint rule (`midpoint.m`) provided on the course webpage for the numerical solution of the ordinary differential equation (ODE)

$$y'(t) = f(t, y(t)) \;\; t > a \;\text{ and }\; y(a) = y_0$$

on the interval $[a, b]$. Both of the m-files take five input parameters;

- fun - the name of the m-file that evaluates $f(t, y)$ at a given t and y
- a, b - the left and right end-points of the interval on which the solution of the ODE is sought
- h - the step size
- y0 - the initial condition.

They return two output parameters;

- yvec - the approximate solution vector at the discrete points
- tvec - vector of t values at which the approximate y values are computed.

For instance to solve the ODE

$$y' = 3t^2 y \quad t > 0 \quad \text{and} \quad y(0) = 1$$

on the unit interval $[0, 1]$ using Euler's method with the step-size $h = 0.01$ type

```
[yvec, tvec] = Euler('fun_ODE',0,1,0.01,1);
```

Make sure to download the m-file `fun_ODE.m` that evaluates $f(t, y) = 3t^2 y$ before typing the Matlab command above. Also open and go through the m-file `fun_ODE.m` to understand its input and output parameters. You can plot the approximate solution by typing

```
plot(tvec,yvec,'b-*');
```

For questions **3** and **4** attach your Matlab outputs/figures as well as the print-outs of the m-files that you implemented.

1. Solve each of the following ordinary differential equations using

(i) Euler's method with $h = 0.5$ and **(ii)** midpoint rule with $h = 0.5$.

Additionally for each of the ODEs write down the nonlinear system resulting from an application of the trapezoidal method with $h = 0.5$. You must perform all calculations by hand.

(ODE 1) $y' = y^3 \quad t > 0, \quad y(0) = -1 \quad \text{on} \quad [0, 1]$
(ODE 2) $y' = -2y\cos(t) \quad t > 0, \quad y(0) = \mathrm{e} \quad \text{on} \quad [0, 1]$

2. Show that

(i) the order of backward Euler's method with the update rule

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1})$$

is one,

(ii) the order of the Runge-Kutta method with the update rule

$$y_{k+1} = y_k + \frac{h}{4} f(t_k, y_k) + \frac{3h}{4} f\left(t_k + \frac{2h}{3}, y_k + \frac{2h}{3} f(t_k, y_k)\right)$$

is two.

3. The velocity v of an object subject to the gravitational force and air resistance (proportional to v) satisfies the ODE

$$\frac{dv}{dt} = g - \frac{kv}{m}$$

where $g = 9.81$ m/s^2 is the gravitational acceleration, k is a constant for the air-resistance and m is the mass of the object.

(a) Suppose that the initial velocity $v(0) = 100$ m/s and $k/m = 0.05\ \ 1/\text{s}$. Find the approximate solution on the time interval $[0, 3]$ using Euler's method (in particular use the m-file `Euler.m` on the course webpage) with the step size $h = 0.001$. Plot the approximate solution.

(b) Repeat part **(a)** but with the initial velocity $v(0) = 0$ m/s and $k/m = 1.5\ \ 1/\text{s}$.

4. The purpose of this question is to compare the accuracy of Euler's method and midpoint rule. Consider the ODE

$$y' = -y \sin t \quad t > 0, \quad \text{and} \quad y(0) = e$$

with the exact solution $y(t) = e^{\cos t}$.

(a) Solve this ODE using Euler's method with $h = 0.1, 0.01, 0.001, 0.0001$ on the interval $[0, 2]$.

(i) Download the routines `generate_fun.m` and `fun_real_trig.m` from the course webpage. Plot the actual solution $y(t) = e^{\cos t}$ on the interval $[0, 2]$ by typing

```
>> [yreal, treal] = generate_fun('fun_real_trig',0,2,0.001);
>> plot(yreal, treal, 'r-*');
```

(ii) Plot the computed solutions for each of the h values above on the same figure. You should type

```
>> figure(1)
>> hold on
```

to keep the previous curves plotted and superimpose

(iii) Calculate the error at time $t = 2$ of the computed solution for each of the h values. Does the global error decay as expected in theory?

(b) Repeat part **(a)** but with the midpoint rule instead of Euler's method.

5. A simple predator-prey relationship is described by the Lotka-Volterra model provided below, which is written in terms of

- a fox population $f(t)$ with birth rate b_f and death rate d_f, and
- a geese population $g(t)$ with birth rate b_g and death rate d_g

$$\begin{aligned}\frac{df}{dt} &= f(t)(b_f g(t) - d_f)\\ \frac{dg}{dt} &= g(t)(b_g - d_g f(t))\end{aligned}$$

Find the populations on $[0, 2]$ using Euler's method with step-size $h = 1$ for the following parameter values and initial conditions

(a) $b_f = d_f = b_g = d_g = 1,\ f(0) = g(0) = 2.$

(b) $b_f = b_g = 1,\ d_f = d_g = 0.5,\ f(0) = 2$ and $g(0) = 10.$

6. Convert the second-order ODE

$$y'' + 3y' + y^4 = 2t \quad t > 0 \quad \text{and} \quad y(0) = 2,\ y'(0) = 0$$

into a system first order ODEs. Then solve the resulting ODE on the interval $[0, 6]$ using Euler's method with the step-size $h = 2$.

7. Estimate the integral

$$\int_{-1}^{1} e^{-x^3}\, dx$$

using the composite Simpson's rule with $m = 2$ subintervals and the Gaussian quadrature with three nodes x_0, x_1, x_2.

8. Derive the Gaussian quadrature formula for the integral

$$\int_0^3 f(x)\, dx$$

with three nodes x_0, x_1, x_2.

9. Derive the Gaussian quadrature with the weight function $W(x) = \sqrt{\frac{1}{1-x^2}}$ and with two nodes x_0, x_1, that is derive the quadrature formula

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\, dx \approx w_0 f(x_0) + w_1 f(x_1)$$

with the maximal degree-of-exactness.

10. Show that all of the n roots of the nth degree orthogonal polynomial $q_n(x)$ satisfying

$$\int_a^b q_n(x) p(x)\, dx = 0 \quad \text{for all } p(x) \in \mathbb{P}_{n-1}$$

is contained in the interval $[a, b]$.

11. Consider a quadrature formula

$$\int_a^b W(x)f(x) \approx \sum_{k=0}^{n} w_k f(x_k)$$

where $x_0, \ldots, x_n \in [a, b]$ are distinct.

Suppose that $x_0, \ldots, x_n$ are roots of a polynomial $q_{n+1}(x)$ of degree $n+1$ such that

$$\int_a^b W(x) q_{n+1}(x) p(x)\, dx = 0$$

for all polynomials $p(x)$ of degree $\ell - 1$ or smaller where $\ell \leq (n+1)$. Show that the degree of the exactness of the quadrature formula is $n + \ell$.

12.

(a) Implement the composite Simpson's rule to calculate the integral

$$\int_a^b f(x)\, dx$$

numerically to a specified accuracy.

Your routine must take five input parameters.

- a and b
- *fname*: The name of a matlab m-file evaluating $f(x)$ at a given point
- *tol*: The computed value must not differ from the exact integral by more than *tol*.
- *ubound*: An upper bound for the fourth derivative on $[a, b]$ must be supplied by the user. This is needed to estimate the error, in particular to determine how many subintervals in $[a, b]$ are needed to reach the specified accuracy.

It must return two output parameters.

- *intval*: computed value of the integral
- m: number of subintervals of $[a, b]$ used to reach the specified accuracy.

Your routine should be called as follows.

```
>> [intval, m] = composite_simpson(a,b,'fname',tol,ubound);
```

It must exploit the fact that when $[a, b]$ is split into m subintervals of equal length, the error of the Simpson's method is given by

$$\left| \frac{b-a}{180} (H/2)^4 f^{(4)}(\varepsilon) \right| \tag{1}$$

for some $\varepsilon \in (a, b)$ where $H = (b-a)/m$.

(b) Consider the ellipse

$$\frac{x^2}{\beta^2} + \frac{y^2}{\alpha^2} = 1.$$

The surface area of the resulting ellipsoid obtained when the ellipse above is rotated about the x-axis is given by the integral

$$4\pi\alpha \int_0^\beta \sqrt{1 - K^2 x^2}\, dx$$

where $K^2 = \frac{1}{\beta^2}\sqrt{1 - \frac{\alpha^2}{\beta^2}}$. Calculate the surface area of the ellipsoid for

$$\alpha = \sqrt{(3 - 2\sqrt{2})/100}, \quad \beta = 10$$

within an accuracy of 10^{-4} and 10^{-6} using your routine `composite_simpson`.