Name _____   Class Time _____

Math 2250 Maple Project 4: Numerical Methods
August 2008

Due date: See the internet due dates. Maple lab 4 has four problems L4.1, L4.2, L4.3, L4.4.

References: Code in `maple` appears in `2250mapleL4-S2009.txt` at URL `http://www.math.utah.edu/~gustafso/`. This document: `2250mapleL4-S2009.pdf`. Other related and required documents are available at the web site:

- Numerical Solution of First Order DE (typeset, 19 pages, 220k pdf). A resource similar to the textbook, with maple examples and deeper detail. It is for a second reading, in case Edwards-Penney left too many questions unanswered.

- Sample Report for 2.4-3 (pdf 3 pages, 350k). This outline might be useful, if you are wondering which details to include.

- Numerical DE coding hints, TEXT Document (1 page, 2k). A modified portion of this document is appended here, for completeness. The web copy `2250mapleL4-S2009.txt` is suited for mouse copying.

- Sample maple code for Euler, Heun, RK4 (maple worksheet). Use `2250mapleL4-S2009-snips.mws` to load maple sample code without mouse copying. Includes error code and exact solution examples.

Problem L4.1. (E & P Exercise 2.4-12, Symbolic Solution)

The exact symbolic solution of $y' = \frac{1}{2}(y-1)^2$, $y(0) = 2$ is $y = \dfrac{x-4}{x-2}$. Using methods from the textbook, Chapter 1, display the details of the derivation for this symbolic solution, plus a full answer check.

Staple this page on top of your hand-written report

Math 2250 Maple Project 4: Numerical Methods
S2009

References: Code in `maple` appears in `2250mapleL4-S2009.txt` at URL `http://www.math.utah.edu/~gustafso/`. This document: `2250mapleL4-S2009.pdf`. Further references appear on the previous page.

Problem L4.2. (E & P Exercise 2.4-12)

Consider the initial value problem $y' = \frac{1}{2}(y-1)^2$, $y(0) = 2$ with symbolic solution $y = \dfrac{x-4}{x-2}$. Apply Euler's method to produce two dot tables, as follows. The first has 101 rows, $h = 0.01$. The second has 201 rows, $h = 0.005$. Do not print the dot tables, just print the computer code that made them. Reproduce, by transcribing computer data, the table below, and fill in missing digits. Follow the sample report for Exercise 2.4-3.

`http://www.math.utah.edu/~gustafso/2250SampleProblem2.4-3.pdf`

Reference L4.1 for the symbolic solution, excluding any derivation or answer check. Don't hand-check the dot tables. Instead, test the answers against the symbolic solution, as suggested in the table below. For the percentage error with $h = 0.005$, use the equation

$$Error(approx, actual) = 100\frac{|approx - actual|}{|actual|}.$$

x	0.0	0.2	0.4	0.6	0.8	1.0
y-approx, $h = 0.01$	2.000000000	2.110466386	2.2482????0	2.4249????1	2.659682???	2.98641????
y-approx, $h = 0.005$	2.000000000	2.110787345	2.249?????1	2.4267????6	2.66314???0	2.9931?????
actual $y(x)$	2.000000000	2.111111111	2.250000000	2.428571429	2.666666667	3.000000000
Error(approx,actual)	0.0000%	0.015?%	0.038?%	0.07??%	0.13??%	0.22??%

Problem L4.3. (E & P Exercise 2.5-12)

Consider the initial value problem $y' = \frac{1}{2}(y-1)^2$, $y(0) = 2$ with symbolic solution $y = \dfrac{x-4}{x-2}$. Apply Heun's method to produce two dot tables, as follows. The first has 101 rows, $h = 0.01$. The second has 201 rows, $h = 0.005$. Do not print the dot tables, just print the computer code that made them. Reproduce, by transcribing computer data, the table below, and fill in missing digits. Follow the sample report for Exercise 2.4-3 as in problem L4.2 above. Don't hand-check the dot tables. Instead, test the tables against the symbolic solution, as suggested in the table below.

x	0.0	0.2	0.4	0.6	0.8	1.0
y-approx, $h = 0.01$	2.000000000	2.11110????	2.24999???2	2.428???558	2.666643???	2.999950375
y-approx, $h = 0.005$	2.000000000	2.111110682	2.249998780	2.428568700	2.666660895	2.999987544
actual $y(x)$	2.000000000	2.111111111	2.250000000	2.428571429	2.666666667	3.000000000
Error(approx,actual)	0.0000?%	0.0000?%	0.0000?%	0.0001?%	0.0002?%	0.0004?%

Problem L4.4. (E & P Exercise 2.6-12)

The actual solution of $y' = \frac{1}{2}(y-1)^2$, $y(0) = 2$ is $y = \dfrac{x-4}{x-2}$. Apply the RK4 method to produce two dot tables, as follows. The first has 101 rows, $h = 0.01$. The second has 201 rows, $h = 0.005$. Do not print the dot tables, just print the computer code that made them. Reproduce, by transcribing computer data, the table below, and fill in missing digits. Follow the sample report for Exercise 2.4-3 as in problem L4.3 above. Don't hand-check the dot tables. Instead, test the tables against the symbolic solution, as suggested in the table below.

x	0.0	0.2	0.4	0.6	0.8	1.0
y-approx, $h = 0.01$	2.000000000	2.111111???	2.250000001	2.428571428	2.66666????	2.999999996
y-approx, $h = 0.005$	2.000000000	2.111111112	2.2500000??	2.4285714??	2.6666666??	2.999999982
actual $y(x)$	2.000000000	2.111111111	2.250000000	2.428571429	2.666666667	3.000000000
Error(approx,actual)	0.000000000%	0.00000004?%	0.00000004?%	0.000000123%	0.000000262%	0.000000600%

Staple this page on top of your hand-written and maple worksheet report

```
# Warning: These snips of code made for y'=1-x-y, y(0)=3.
#          Code computes approx values for y(0.1) to y(1).
# 'Dots' is the list of dots for connect-the-dots graphics.
# ======================================
# Euler. Group 1, initialize.
 f:=(x,y)->1-x-y:
 x0:=0:y0:=3:h:=0.1:Dots:=[x0,y0]:n:=200:
# Group 2, repeat n times. Euler's method
for i from 1 to n do
 Y:=y0+h*f(x0,y0);
 x0:=x0+h:y0:=Y:Dots:=Dots,[x0,y0];
od:
# Group 3, display relevant dots and plot.
 Exact:=x->2-x+exp(-x);
 P:=unapply(evalf(100*abs(exact-approx)/abs(exact)),(exact,approx)):
 m:=40:X:=[seq(1+m*j,j=0..n/m)]: # List of relevant indices
 print("Dots"),seq(Dots[k],k=X);
 print("Exact"),seq(Exact(Dots[k][1]),k=X);
 print("Error"),seq(P(Exact(Dots[k][1]),Dots[k][2]),k=X);
 plot([Dots]);
# ======================================
# Heun. Group 1, initialize.
 f:=(x,y)->1-x-y:
 x0:=0:y0:=3:h:=0.1:Dots:=[x0,y0]:n:=200:
# Group 2, repeat n times. Heun method.
for i from 1 to n do
 Y1:=y0+h*f(x0,y0);
 Y:=y0+h*(f(x0,y0)+f(x0+h,Y1))/2:
 x0:=x0+h:y0:=Y:Dots:=Dots,[x0,y0];
od:
# Group 3, display relevant dots and plot.
 Dots[1],Dots[2],seq(Dots[1+40*j],j=1..n/40);
 plot([Dots]);


# ======================================
# RK4. Group 1, initialize.
 f:=(x,y)->1-x-y:
 x0:=0:y0:=3:h:=0.1:Dots:=[x0,y0]:n:=100:
# Group 2, repeat n times. RK4 method.
for i from 1 to n do
 k1:=h*f(x0,y0):
 k2:=h*f(x0+h/2,y0+k1/2):
 k3:=h*f(x0+h/2,y0+k2/2):
 k4:=h*f(x0+h,y0+k3):
 Y:=y0+(k1+2*k2+2*k3+k4)/6:
 x0:=x0+h:y0:=Y:Dots:=Dots,[x0,y0];
od:
# Group 3, display some dots and plot.
 Dots[1],Dots[2],Dots[101];
 plot([Dots]);

# Code snips for exact/error reports
# ======================================
# Making multiple curves on one plot
# ======================================
  Exact:=x->2-x+exp(-x);               # An exact solution
  plot({Exact(x),[Dots]},x=0..1/2);    # plot exact and approx solutions


# ======================================
```

```
# How to create a Dots table for the exact solution
# ======================================
Exact:= x -> 2-x+exp(-x):n:=10:
ExactDots:=seq([Dots[j][1],Exact(Dots[j][1])],j=1..n+1);


# ======================================
# How to define and print percentage relative error:
# ======================================
  P:=unapply(evalf(100*abs(exact-approx)/abs(exact)),(exact,approx));
  ExactVal:=Exact(Dots[11][1]): # Compute exact y-value for x=0.5
  ApproxVal:=Dots[11][2]:         # Get Euler approx y-value for x=0.5
  P(ExactVal,ApproxVal);          # print percent relative error


# ======================================
# How to create a Dots table for percentage error
# ======================================
  P:=unapply(evalf(100*abs(exact-approx)/abs(exact)),(exact,approx));
  Pdots:=seq([Dots[j][1],P(Exact(Dots[j][1]),Dots[j][2])],j=1..11);


# ======================================
# Printing results and tables
# Make tables with a pencil, it saves time.
# ======================================
# To extract and print items 1,101,201,1001 from a list:
Dots1:=Dots[1],Dots[101],Dots[201],Dots[1001];
```

End of Maple Lab 4: Numerical Methods.