CS 370 Winter 2013: Assignment 2

Instructor: Yuying Li	Office: DC3623	e-mail: *yuying@uwaterloo.ca*
Lec 001 :	MWF 8:30-9:20 MC2054	OH (Li):Tues 2-3pm
TAs:		
Nian Ke	knian@uwaterloo.ca	DC3594
Mark Prosser	rmprosse@uwaterloo.ca	DC2302B
Haofan Zhang	h267zhan@uwaterloo.ca	DC3594

Web Site: www.student.cs.uwaterloo.ca/~cs370/

Due Feb 15, 2013, 5:00 PM, in the Assignment Boxes, 3rd Floor MC. Please attach a cover page, which you can download at the course website, to your submitted assignment.

1. **(8 marks)** Consider the IVP $\frac{d^2y}{dt^2} = y$ with the initial value $y(0) = 1$ and $\frac{dy}{dt}(0) = 2$.

   (a) Write this second order ODE as an equivalent system of first order ODEs $\frac{d\mathbf{z}}{dt} = \mathbf{f}(\mathbf{z}, t)$ with appropriate initial conditions.

   (b) Perform *two* steps of the Euler's method for this ODE system from $t = 0$ using a step size of $h = 0.5$.

   (c) Perform *one* step of the *improved* Euler's method for this ODE system from $t = 0$ using a step size of $h = 0.5$.

2. **(14 marks)** Given t_0, α, consider the initial value problem

$$\frac{d^3z}{dt^3} - \frac{dz}{dt} = t, \quad z(0) = \frac{dz(0)}{dt} = \frac{d^2z(0)}{dt^2} = \alpha \tag{1}$$

   (a) Show that $z(t) = (e^t + e^{-t} - t^2)/2 - 1$ is the solution to the above IVP with $t_0 = 0$ and $\alpha = 0$.

   (b) Convert the third order differential equation above to a system of first-order equations.

   (c) Write a matlab function $\tilde{z} = \text{myEuler}(t_0, t_{\text{final}}, N, \alpha)$ for IVP (1), which takes, as input, the initial time t_0, the final time t_{final}, the number of steps N, and constant α specifying initial values. It returns the approximate solution vector z of length $N+1$ such that

$$\tilde{z}(i) \approx z(t_{i-1}), \quad i = 1, 2, \cdots, N+1$$

   where $z(t)$ denotes the exact solution, $t_i = t_0 + ih$, $h = (t_{\text{final}} - t_0)/N$. Note that the function does not return the derivative values here.

   (d) Write a matlab function $z = \text{myImprEuler}(t_0, t_{\text{final}}, N, \alpha)$, which implements a modified Euler method with the same input arguments as in (c).

   (e) Assume $t_{\text{final}} = 4$. Computationally illustrate the order of the Euler method based on the computed value z as follows. Let $\text{err}(h) = |z(t_{\text{final}}) - \tilde{z}(N+1)|$ denote the error when computing with the stepsize h. For a p-order method,

$$\text{err}(h) \approx Ch^p$$

   for some constant $C > 0$. The effect of cutting the stepsize h in half is

$$\frac{\text{err}(h/2)}{\text{err}(h)} \approx \frac{1}{2^p}.$$

   Thus the error is reduced by a factor of 2^p. Illustrate the order of the method by cutting the stepsize in half a few times and calculating the above ratios. Specifically, use $N = 2^i \times 10$, for $i = 1, 2, \cdots 10$, report the error and its associated ratio $\frac{\text{err}(h/2)}{\text{err}(h)}$. What is the order of the Euler method based on your computation?

(f) Repeat (e) for the improved Euler method.

3. (10 marks) Show that the local truncation error of the midpoint method

$$y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}hf(t_n, y_n)\right)$$

is $O(h^3)$ where h is the time stepsize. Recall that

$$f(x+\delta x, y+\delta y) = f(x,y) + \frac{\partial f(x,y)}{\partial x}\delta x + \frac{\partial f(x,y)}{\partial y}\delta y + O((\delta x)^2) + O((\delta y)^2) + O(\delta x \delta y)$$

$$\frac{d^2 y(t)}{dt^2} = \frac{df(t,y(t))}{dt} = \frac{\partial f(t,y(t))}{\partial t} + f(t,y(t))\frac{\partial f(t,y(t))}{\partial y}$$

4. **(18 marks)** (Variation of NCM 7.19, 12 marks) In the 1968 Olympic games in Mexico City, Bob Beamon established a world record with a long jump of 8.90m. This was 0.8m longer than the previous world record. Since 1968, Beamon's jump has been exceeded only once in competition, by Mike Powell's jump of 8.95m in Tokyo in 1991. After Beamon's remarkable jump, some people suggested that the lower air resistance at Mexico City's 2250m altitude was a contributing factor. This problem examines that possibility.

The mathematical model for the jump can be described as follows. The fixed Cartesian coordinate system has a horizontal x-axis, a vertical y-axis, and an origin at the takeoff board. The jumper's initial velocity has magnitude v_0 and makes an angle with respect to the x-axis of θ_0 radians. The only forces acting after takeoff are gravity and aerodynamic drag, D, which is proportional to the square of the magnitude of the velocity. There is no wind. The equations describing the jumper's motion are

$$\frac{dx}{dt} = v\cos(\theta), \quad \frac{dy}{dt} = v\sin(\theta),$$

$$\frac{d\theta}{dt} = -\frac{g}{v}\cos(\theta), \quad \frac{dv}{dt} = -\frac{D}{m} - g\sin(\theta).$$

The drag is

$$D = \frac{c\rho s}{2}\left(\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2\right).$$

Assume the following constants for this problem are the acceleration of gravity, $g = 9.81$ meter/sec^2, the mass, $m = 90$ kg, the drag coefficient, $c = 0.72$, the jumper's cross-sectional area, $s = 0.50$ meter^2, and the takeoff angle, $\theta_0 = \pi/8$ radians (22.5°).

Compute four different jumps, with different values for initial velocity, v_0, and air density, ρ. The length of each jump is $x(t_f)$, where the air time, t_f, is determined by the condition $y(t_f) = 0$.

(a) Normal jump at high altitude: $v_0 = 10$ meter/sec and $\rho = 0.94$ kg/meter^3.

(b) Normal jump at see level: $v_0 = 10$ meter/sec and $\rho = 1.29$ kg/meter^3.

(c) Sprinter's approach at high altitude: $\rho = 0.94$ kg/meter^3. Determine v_0 so that the length of the jump is Beamon's record, 8.90 meter. This is a root finding problem: determining the initial v_0 such $f(v_0) = 0$ where $f(v_0)$ is the length of the jump with the initial speed v_0 subtracted by 8.90. Matlab function `fzerotx.m`, which can be downloaded from the course web, can be used to solve this problem. You need to write a function to evaluate the function $f(v_0)$ and pass the function name to `fzerotx.m`.

(d) Sprinter's approach at sea level: $\rho = 1.29$ kg/meter^3 and v_0 is the value determined in (c) above.

Use the Matlab ODE solver `ode45()` as follows

$$[t, X] = ode45(\text{ODEFUN}, [0:0.05:5], [x0; y0; theta0; v0], options, rho);$$

for all your submitted computations and use an event function to determine the length of jumps (a), (b) and (d).

Write a simple animation subroutine to plot the animation of each jump above. You can use the Matlab `pause()` command to simulate the animation. Submit the final animation plot for the jumps in (a) and (b) listed above.

In addition, present your results by completing the following table. Provide numerical results with 4 digits of accuracy.

v_0	θ_0	ρ	distance
10.0000	22.5000	0.9400	???
10.0000	22.5000	1.2900	???
???	22.5000	0.9400	8.9000
???	22.5000	1.2900	???

Comment on the results obtained. Which is more important, the air density or the jumper's initial velocity?

Submit all your Matlab code (excluding the `fzerotx.m` function from the course web page). Submit the completed results table as well as the final animation plot for jumps in (a) and (b) specified above. Also, submit your comments on the results obtained.