APMA 0160 (A. Yew) Spring 2011

Homework #9

Deadline: May 1 (Sunday) 4:00am EST

Problem 1

Consider the Lorenz system

$$\frac{dx}{dt} = \sigma(-x+y)$$
$$\frac{dy}{dt} = \rho x - y - xz$$
$$\frac{dz}{dt} = -\beta z + xy$$

With the parameter values $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$, a solution will trace out a shape in (x, y, z) space known as the "Lorenz attractor".

(a) Compute an approximation of the Lorenz attractor by numerically solving the Lorenz equations from $t = 0$ to $t = 100$, starting from the initial conditions $x(0) = y(0) = z(0) = 1$.

(i) Do the computation using the predictor–corrector method you coded for Homework #8 Problem 3 part (d) (combining the three-step Adams–Bashforth and three-step Adams–Moulton methods and initializing with the standard four-stage Runge–Kutta method—don't forget to upload this code to Scorelator). Take the step size to be 0.02. Save the last 700 computed solution points in `A11.dat` as a three-column matrix whose last row contains the approximate values of $x(100), y(100), z(100)$, penultimate row contains the approximate values of $x(99.98), y(99.98), z(99.98)$, 698th row contains the approximate values of $x(99.96), y(99.96), z(99.96)$, and so on.

(ii) Do the computation using Matlab's built-in predictor–corrector ODE solver `ode113` with absolute tolerance 10^{-5} and relative tolerance 10^{-4}. Ask Matlab to record the approximate solution values at $t = 0 : 0.02 : 100$, and save the last 700 recorded solution points as a three-column matrix in `A12.dat`.

(b) Use `ode113` with absolute tolerance 10^{-5} and relative tolerance 10^{-4} to approximate the solution of the forced Lorenz system

$$\frac{dx}{dt} = \sigma(-x+y) + \gamma \cos t$$
$$\frac{dy}{dt} = \rho x - y - xz$$
$$\frac{dz}{dt} = -\beta z + xy$$

where $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ and $\gamma = 80$. Compute the solution from $t = 0$ to $t = 100$, starting from the initial conditions $x(0) = y(0) = z(0) = 1$. Ask Matlab to record the approximate solution values at $t = 0 : 0.02 : 100$, and save the last 700 recorded solution points as a three-column matrix in `A13.dat`.

Note: Although the code you submit to Scorelator should not output any graphs, after each of the above computations it is recommended that you plot the solution trajectory in (x, y, z) phase space and also view its projection onto the (x, y) plane, to get an idea of what the famous Lorenz attractor looks like.

(PLEASE TURN OVER)

Problem 2

The Van der Pol differential equation is

$$\frac{d^2y}{dt^2} + \mu(y^2 - 1)\frac{dy}{dt} + y = 0$$

(a) Use `ode45` with default tolerances to compute the solution of the Van der Pol equation from $t = 0$ to $t = 40$, starting from the initial conditions $y(0) = 0.1$, $y'(0) = -1$. Do this for the parameter values $\mu = 0.1$, 2, 5 and 10, and ask Matlab to record the approximate solution values at $t = 0 : 0.05 : 40$. Save these values of $y(t)$ and $y'(t)$ as two-column matrices in `A21.dat`, `A22.dat`, `A23.dat` and `A24.dat` (for $\mu = 0.1$, 2, 5 and 10, respectively).

(b) When μ is large, the Van der Pol equation is "stiff". With $\mu = 100$, use the stiff solver `ode15s` with default tolerances to compute the solution from $t = 0$ to $t = 500$, starting from the initial conditions $y(0) = 0.1$, $y'(0) = -1$, and save the computed solution as a two-column matrix in `A25.dat`

(c) In this part you will use the Van der Pol equation with $\mu = 1$ and initial conditions $y(0) = 5$, $y'(0) = 0$ to estimate the **order** of the methods `ode45`, `ode113` and `ode15s`.
With each of `ode45`, `ode113` and `ode15s`, do the following:

- For $p = 4, 5, 6, \ldots, 10$, set both the absolute tolerance and the relative tolerance to 10^{-p}.
- Compute the solution from $t = 0$ to $t = 30$ and calculate the *average* step size h used.
- Plot $\log_{10}(h)$ against p and calculate the gradient $\dfrac{\min\log_{10}(h) - \max\log_{10}(h)}{10-4}$. Use this gradient to obtain an estimate of the order of the method, i.e. the power q such that the error is proportional to h^q.

Save your estimate of the order of `ode45` in `A26.dat` as a row vector whose first entry is the unrounded value of q with at least four decimal places and whose second entry is the rounded (integer) value. Save your unrounded and rounded estimates of the order of `ode113` in `A27.dat`, and save your unrounded and rounded estimates of the order of `ode15s` in `A28.dat`.

(d) Use `ode45` with absolute tolerance = relative tolerance = 10^{-7} to compute the solution of the Van der Pol equation with $\mu = 10$ from $t = 0$ to $t = 40$, starting from the initial conditions $y(0) = -2$, $y'(0) = 0.07$. If you plot this computed solution, you will see that it seems periodic. Estimate the period by asking Matlab to detect the minimum "event" time τ such that $(y(\tau), y'(\tau))$ gets within distance 0.01 of $(y(0), y'(0))$. Save the period you found in `A29.dat`.

Note: Although the code you submit to Scorelator should not output any graphs, after each of the computations in parts (a) and (b) it is recommended that you plot at least the $y(t)$ graph of the solution. For part (d) it would also be informative to plot the solution's trajectory in the (y, y') phase plane. In part (c), plot the $\big(p, \log_{10}(h)\big)$ graphs and check that they do resemble straight lines.