**APMA 0160 (A. Yew) Spring 2011**

# Homework #8

Deadline: April 24 (Sunday) 4:00am EST

---

**Problem 1**

Write a Matlab program that computes an approximation of the solution to a **first-order scalar** ODE using the **Crank–Nicolson (trapezoidal)** method with a fixed step size $h$.
*Note*: Use `fzero` rather than `fsolve` to find the root of the implicit relation for $u_{n+1}$.

(a) Use your code to approximate the solution of

$$\frac{du}{dt} + u^2 + 5e^{-t}\sin 5t = 0, \qquad u(0) = 1$$

Compute the value of $u$ at the times $t = 0.02,\ 0.04,\ 0.06, \ldots, 3$, and save your results in `A11.dat` as a column vector whose first entry is the value $u(0)$, second entry is the value $u(0.02)$, third entry is the value $u(0.04)$, and so on.

(b) Test your code on the simple differential equation $u' = u$ with initial condition $u(0) = 0.5$. Taking $h = 2^{-m}$ for $m = 3, 4, \ldots, 8$, compute the absolute error $E$ at $t = 2$. By plotting $\log_2(E)$ against $m$ and calculating the gradient $\big(\min \log_2(E) - \max \log_2(E)\big)/5$, estimate the **order** of the Crank–Nicolson method, i.e. the power $q$ such that the error is proportional to $h^q$. Save your estimate of $q$ in `A12.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.

---

**Problem 2**

Write a Matlab program that computes an approximation of the solution to an ODE system using the **three-stage Runge–Kutta** method described by the Butcher tableau on the right, with a fixed step size $h$.

| | | | |
|---|---|---|---|
| 0 | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | |
| 1 | $-1$ | 2 | |
| | $\frac{1}{6}$ | $\frac{2}{3}$ | $\frac{1}{6}$ |

(a) The following differential equation describes the motion of a simple pendulum:

$$\frac{d^2\theta}{dt^2} + \sin\theta = 0$$

Use your code to compute the solution of this equation that satisfies the initial conditions $\theta(0) = -0.5$ and $\theta'(0) = 1.2$. Compute the values of $\theta$ and $\theta'$ at the times $t = 0.1,\ 0.2,\ 0.3, \ldots, 10$, and save your results in `A21.dat` as a matrix whose first row contains the values $\theta(0)$ and $\theta'(0)$, second row contains the values $\theta(0.1)$ and $\theta'(0.1)$, third row contains the values $\theta(0.2)$ and $\theta'(0.2)$, and so on.

(b) Use your code to approximate the solution of the two-body problem

$$\frac{d^2x}{dt^2} + \frac{x}{(\sqrt{x^2+y^2}\,)^3} = 0, \qquad \frac{d^2y}{dt^2} + \frac{y}{(\sqrt{x^2+y^2}\,)^3} = 0$$

that satisfies the initial conditions $x(0) = 1$, $y(0) = 0$, $x'(0) = 0$ and $y'(0) = 0.8$.
Compute the values of $x$ and $y$ at the times $t = 0.05,\ 0.1,\ 0.15,\ 0.2, \ldots, 7$, and save your results in `A22.dat` as a matrix whose first row contains the values $x(0)$ and $y(0)$, second row contains the values $x(0.05)$ and $y(0.05)$, third row contains the values $x(0.1)$ and $y(0.1)$, and so on.

(PLEASE TURN OVER)

(c) Estimate the order of this Runge–Kutta method by testing your code on the differential equation $u' = u$ with initial condition $u(0) = 0.5$. Taking $h = 2^{-m}$ for $m = 3, 4, \ldots, 12$, compute the absolute error $E$ at $t = 2$; then plot $\log_2(E)$ against $m$ and calculate the gradient $\big(\min \log_2(E) - \max \log_2(E)\big)/9$. Save your estimate of the order $q$ in `A23.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.

---

**Problem 3**

(a) Write a Matlab program that implements the **three-step Adams–Bashforth** method with a fixed step size $h$. Use Euler's method to initialize the time-stepping loop. Estimate the order of this algorithm by testing your code on the simple differential equation $u' = u$ with initial condition $u(0) = 0.5$. Taking $h = 2^{-m}$ for $m = 3, 4, \ldots, 11$, compute the absolute error $E$ at $t = 2$; then plot $\log_2(E)$ against $m$ and calculate the gradient $\big(\min \log_2(E) - \max \log_2(E)\big)/8$. Save your estimate of the order $q$ in `A31.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.

(b) Repeat part (a), but now use the standard four-stage Runge–Kutta method to initialize the time-stepping loop. Save your unrounded and rounded estimates of the order in `A32.dat`.

(c) Use your code from part (b) to approximate the solution of the predator–prey model

$$\frac{dV}{dt} = rV - a\,VP$$
$$\frac{dP}{dt} = -sP + ba\,VP$$

with parameter values $r = 20$, $s = 10$, $a = 1$, $b = 1$ and initial conditions $V(0) = 10$, $P(0) = 10$. Compute the values of $V$ and $P$ at the times $t = 0.01,\ 0.02,\ 0.03, \ldots, 3$, and save your results in `A33.dat` as a matrix whose first row contains the values $V(0)$ and $P(0)$, second row contains the values $V(0.01)$ and $P(0.01)$, third row contains the values $V(0.02)$ and $P(0.02)$, and so on.

(d) Write a Matlab program that implements the **predictor–corrector** method combining the three-step Adams–Bashforth method with the three-step Adams–Moulton method. Use the standard four-stage Runge–Kutta method to initialize the time-stepping loop. Estimate the order of this algorithm with the procedure described in part (a), and save your unrounded and rounded estimates of the order in `A34.dat`.
Also, use your predictor–corrector code to compute the solution of the predator–prey system given in part (c) (with the same parameter values and initial conditions); save the approximated $V(t), P(t)$ values for $t = 0,\ 0.01,\ 0.02,\ 0.03, \ldots, 3$ in `A35.dat`.