MATH 335: Numerical Analysis

Problem Set 20, Final version

Due Date: Thurs., May 7, 2009

Read Sections 11.10–11.11 from Kharab & Guenther, along with your in-class notes, including my handout through the bottom of p. 17.

$\star$42 (a) Write a routine to carry out the Adams-Bashforth-Moulton predictor-corrector method (I believe the text omits Bashforth's name, as if there were no separate Adams-Moulton method which is purely implicit) for a system

$$\begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_m' \end{bmatrix} = \begin{bmatrix} f_1(t, y_1, \ldots, y_m) \\ f_2(t, y_1, \ldots, y_m) \\ \vdots \\ f_m(t, y_1, \ldots, y_m) \end{bmatrix}, \quad t > 0\,, \qquad \text{subject to} \qquad \begin{bmatrix} y_1(0) \\ y_2(0) \\ \vdots \\ y_m(0) \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha \end{bmatrix},$$

or, in vector form,

$$\mathbf{y}' \;=\; \mathbf{f}(t, \mathbf{y})\,, \quad t > 0\,, \qquad \text{subject to} \qquad \mathbf{y}(0) = \boldsymbol{\alpha}\,.$$

Hint: As in the scalar case, we have to determine the first few states via another method (click here for my RK4 method for systems). Now, say we are implementing a 3$^{\text{rd}}$-order ABM method. Once we have progressed far enough to implement actual ABM steps, we get a predicted **y** from 3rd-order Adams-Bashforth:

$$\mathbf{y}_{\text{pred}} \;=\; \mathbf{y}^{(j)} + \frac{h}{12}\Big[23\,\mathbf{f}(t_j, \mathbf{y}^{(j)}) - 16\,\mathbf{f}(t_{j-1}, \mathbf{y}^{(j-1)}) + 5\,\mathbf{f}(t_{j-2}, \mathbf{y}^{(j-2)})\Big]\,,$$

and we *correct* it using Adams-Moulton:

$$\mathbf{y}^{(j+1)} \;=\; \mathbf{y}_{\text{corr}} \;=\; \mathbf{y}^{(j)} + \frac{h}{12}\Big[5\,\mathbf{f}(t_{j+1}, \mathbf{y}_{\text{pred}}) + 8\,\mathbf{f}(t_j, \mathbf{y}^{(j)}) - \mathbf{f}(t_{j-1}, \mathbf{y}^{(j-1)})\Big]\,.$$

(b) Use your routine (use a 4$^{\text{th}}$-order method, and $h = 0.1$) to solve (numerically) the problem of Exercise 11.10.1 (c):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t^2 - 3x^2 + y \\ 2t + x - y \end{bmatrix}, \quad t \in [0,1]\,, \qquad \text{subject to} \qquad \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

An answer is in the back, though the numbers come from an RK4 routine for systems, so it will only tell you if your routine is producing numbers in the right ballpark.

★43 When solving a 1st-order *linear* system of DEs $\mathbf{y}' = \mathbf{A}\mathbf{y}$, the implicit Euler method can be manipulated algebraically so as to eliminate the need for a root-finder/fixed-point iteration. The details look like this:

$$\begin{aligned} \mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} + h\mathbf{f}(t_{j+1}, \mathbf{y}^{(j+1)}) \quad &\Rightarrow \quad \mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} + h\mathbf{A}\mathbf{y}^{(j+1)} \\ &\Rightarrow \quad \mathbf{y}^{(j+1)} - h\mathbf{A}\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} \\ &\Rightarrow \quad (\mathbf{I} - h\mathbf{A})\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} \\ &\Rightarrow \quad \mathbf{y}^{(j+1)} = (\mathbf{I} - h\mathbf{A})^{-1}\mathbf{y}^{(j)} . \end{aligned}$$

(a) Write a routine which accepts the matrix $\mathbf{A}$, an interval $[t_0, t_{final}]$ over which to calculate a solution, an initial value (vector) $\boldsymbol{\alpha}$ (I suggest you make sure it is a column vector!), and a number of time steps to calculate. Your routine should return values like those returned by the routine for Problem ★42. (In particular, the *state matrix* should store the j^{th} state in its j^{th} column).

(b) On p. 17 of the class handout there is an example of a stiff linear system being solved using Euler's method. The theory, confirmed by graphs, suggests we will be pleased only with explicit Euler solutions with $h \leq 1$. (I suggest you run through the commands I have there, perhaps zooming in on specific regions of the graphs, or even trying out some smaller timesteps. Here is a link to my eulerSys.m routine.)

Use your implicit Euler method for linear systems to solve the same problem on the interval $[0, 100]$ using several h-values *no smaller* than 1. Zoom in on specific regions again. Does the implicit Euler method appear to be unconditionally stable on this problem? Is there any stepsize h_0 at which $h > h_0$ appears to result in a degraded solution? Write your comments.