MA 512 **Homework 6** **March 2, 2017**

1. (5 points) In class, we outlined a *predictor-corrector* framework for implementing an implicit Adams–Moulton method of a given order, as follows: At the nth step, "predict" a next solution value y_{n+1}^{AB} using an explicit Adams–Bashforth method *of the same order*, then "correct" to obtain y_{n+1}^{AM} specified by the implicit Adams–Moulton method.[1] As noted in class, this framework allows estimating local error using the difference between the "predicted" and "corrected" values y_{n+1}^{AB} and y_{n+1}^{AM}, specifically

$$\text{local error} \approx \frac{C_{\mathrm{AM}}}{C_{\mathrm{AM}} - C_{\mathrm{AB}}} \left(y_{n+1}^{\mathrm{AM}} - y_{n+1}^{\mathrm{AB}} \right),$$

where C_{AB} and C_{AM} are the constants in the leading local error terms for the Adams–Bashforth and Adams–Moulton methods, respectively. Suppose you use fourth-order Adams–Bashforth "predictor" and Adams–Moulton "corrector" methods. What is $C_{\mathrm{AM}}/(C_{\mathrm{AM}} - C_{\mathrm{AB}})$ in this case? You can find expressions for the local errors on the PDF file "Adams Methods Formulas" posted on the "Handouts" page.[2]

2. (10 points) Recall from Homeworks 1 and 3 the IVP for the damped mechanical oscillator:

$$x'' + Dx' + x = 0, \quad x(0) = x'(0) = 1.$$

As before, this can be recast as a first-order system $y' = f(t, y)$, $y(0) = y_0$, by setting

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \equiv \begin{pmatrix} x \\ x' \end{pmatrix}, \quad f(t, y) = \begin{pmatrix} y_2 \\ -y_1 - Dy_2 \end{pmatrix}, \quad y_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Use MATLAB's `ode113`, which implements an Adams–Bashforth–Moulton predictor-corrector algorithm, to solve this IVP from $t = 0$ to $t = 20$ with $D = .1,\ 1,\ 10,\ 100$. Use `odeset` to set the `RelTol` and `AbsTol` error tolerances to 10^{-8} and also to set the

[1] If the ODE is linear, then the equation determining y_{n+1}^{AM} is a linear system. If this system can be solved using a direct (i.e., non-iterative) linear-algebra method, then the "prediction" step isn't necessary. In the general nonlinear case, as well as in the linear case when iterative linear-algebra methods are appropriate, the "predicted" value y_{n+1}^{AB} is used as an initial approximate solution in an iterative procedure, which is applied to determine an acceptable approximation of y_{n+1}^{AM}.

[2] These expressions involve derivatives $y^{(k)}(\xi_n)$. The point ξ_n differs in different formulas. However, the difference is usually very small in practice and, for error estimation, can be regarded as negligible.

`Jacobian` option to the system Jacobian $\partial f/\partial y$. Print out (or write up) and hand in a table showing the values of D in the first column and the corresponding numbers of steps in the second column. Compare your results with those obtained in Homework 3.[3]

[3] In Homework 3, I asked you to include numbers of f-evaluations, based on `ode45` using six f-evaluations per step. This isn't strictly true, since adjustments in the step-size may require additional f-evaluations. Things are even murkier for `ode113`, since the corrector iterations as well as possible step-size adjustments may require an uncertain number of f-evaluations. The only sure way to count f-evaluations is to put a counter in the routine for evaluating f. In MATLAB, this can be done conveniently using global variables, which allow passing evaluation counts back to the main program.