Homework #4 • Numerical Analysis II (math 416) • Initial Value Problems

- due Wednesday 11 October.
- you may hand in the matlab part of **B)** on Friday, but only if you have participated in the class e-mail discussion.
- please indicate any collaborations, or acknowledge any useful e-mails.
- have a good Thanksgiving.

A) (1-1/2 pages) Show that the 2^{nd}-order Runge-Kutta scheme has a local discretization error that scales as $O(\Delta t^2)$. Remember, the local error is obtained by substituting the true solution $y(t)$ into the discrete operator

$$\bar{\mathcal{L}}_{\Delta t}[\bar{y}, t] = \frac{\bar{y}_{j+1} - \bar{y}_j}{\Delta t} - \frac{1}{\Delta t}\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)$$

where k_1, k_2 are as defined in class. (Hint: use Taylor expansions for small Δt, and cancel the y'' using the derivative of the ODE.)

B) (1 page) Produce a one-page pseudocode for Problem 9.1 from Heath, page 297. Begin with a complete list of equations and variable definitions.

$$\vec{Y}(t) \equiv \begin{pmatrix} y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} b\,y & -\,c\,y\,z \\ -\,d\,z & +\,c\,y\,z \end{pmatrix} \equiv \begin{pmatrix} f(y,z) \\ g(y,z) \end{pmatrix} \equiv \vec{F}(t)$$

I suggest using the above notation, as it is more subscript friendly.

(2 pages) Download the script *hw04.m* for a 2^{nd}-order Runge-Kutta implementation, and modify it to implement a 3^{rd}-order Adams-Bashforth method instead. You should use the existing 2^{nd}-order Runge-Kutta iterations to get the *jumpstart* values for AB3, then just add one extra loop! Don't forget to verify the convergence. State what you learned from doing this problem.

* Consider problems **A)** and the pseudocode of **B)** to be typical midterm-style problems (open notes & texts – Heath, Burden/Faires, or others with my advance permission).