Numerical Methods for Differential Equations
Homework 3: DRAFT

DRAFT VERSION, PLEASE CONSULT FINAL VERSION.
The answers you submit should be attractive, brief, complete, and should include program listings and plots where appropriate. The use of "publish" in Matlab/Octave is one possible approach.

Problem 1 (RK Stability). What is the amplification factor for "*the* 4th-order Runge–Kutta method"? (That is, the one covered in the notes). Plot its region of absolute stability in the complex plane. Note that the stability region includes part of the imaginary axis; write a code to determine (to 10 decimal places) the largest value of $y \in \mathbb{R}$ such that $z = iy$ is included in the linear stability region.

Problem 2 (BDF-2 Stability Analysis).

(a) Consider the BDF-2 linear multistep method. Complete a zero-stability analysis of the method.

(b) Now perform an absolute stability analysis of the scheme. Hint: while we didn't look at this in lecture, the idea is quite similar to the Runge–Kutta case: try to solve the Dahlquist test problem, but now *two* quantities will need to each be bounded by 1.

(c) Plot the BDF-2 region of absolute stability in the complex domain $[-4, 8] \times [-6, 6]$.

(d) Show that the BDF-2 scheme is L-stable. Hint: consider the limit as $z = k\lambda \to -\infty$.

(e) Now consider a fixed λ and consider the $k \to 0$: comment on any similarity or difference to your results in (a).

Problem 3 (Forward Euler and IVP versus BVP). This problem looks at relating convergence in the initial value problem back to the example we did for the BVP $u_{xx} = f$.

Consider the Forward Euler method applied to the Dahlquist test problem. Assume we start at $t = 0$ and compute until $t = T_f = Nk$. Instead of "telescoping" the recursion (as we did in lecture when looking at absolute stability), we will write *all* the steps of the method (from $n = 1$ up to $n = N$) as one $N \times N$ matrix. It should have a diagonal and a subdiagonal only. Call this matrix A.

(a) Write down a system $AU = F$ where $U = \begin{bmatrix} U^1 \approx u(t_1) \\ U^2 \approx u(t_2) \\ \vdots \\ U^N \approx u(T_f) \end{bmatrix}$ is the vector of numerical solutions at each time-step t_n and F is mostly zeros (except the first entry). You can leave some "$\ddots$" but you should otherwise give the matrix A explicitly. Hint: if you get stuck, you can find something very similar in text of LeVeque 2007, Chapter 6.

(b) Now let $\hat{U}$ be the exact solution, sampled at the time-steps t_n. Following our approach in lecture for the BVP, find an expression for the global error E (a vector of values for each time-step t_n), in terms of the local truncation error τ (also a vector of values for each time-step t_n).

(c) Compute (in Octave/Matlab or otherwise) the inverse of this matrix, and measure its size in the matrix max norm $\|A^{-1}\|_\infty$ and the matrix Frobenius norm $\|A^{-1}\|_F$. Report these numbers for the pairs $(N, k) = (10, 0.1)$, $(50, 0.02)$, $(100, 0.01)$ (that is give 6 numbers). For a fixed final time T_f, does it seem like the inverse is bounded independent of k?

(d) Explain how this shows convergence of the Forward Euler method for fixed T_f (at least for this problem, or more generally, at least for linear problems).

Problem 4 (Symmetry and Simpson's Rule). Explain, by application of one of the error theorems, why the Simpson's Rule quadrature scheme is exact for polynomials of degree 3. (Recall this result is counter-intuitive because the method can be viewed as fitting a merely quadratic polynomial proxy and integrating exactly)

Next give a geometric/algebraic argument for why this happens. You should assume (without loss of generality) that $x_0 = -h, x_1 = 0, x_2 = h$. Hint: consider the polynomial in monomial form.

Problem 5: (Trapezoidal Rule). To follow.