MATH 8600 (FALL 2018) HOMEWORK 8

Assigned 11/29/18, preferably due 12/07/18 by 5pm, recommended due by 12/12/18 (final exam), accepted until 5pm 12/14/18.

Instructor: Dr. Fei Xue, Martin O-203, fxue@clemson.edu.

1. **[Q1-Q8]** Textbook Chapter 16, problems 2(b)(c*), 4*, 6(b), 11, 12, 17(b), 18, 16* (change the problem to "Analyze the absolute stability of the one-step implicit midpoint method and the 2-step explicit midpoint method $y_{k+1} = y_{k-1} + 2hf(t_n, y_n)$")

2. **[Q9]** (a*) Consider the single-step explicit method

$$y_{k+1} = y_k + h\left[sf(t_k, y_k) + (1-s)f(t_k + \alpha h, y_k + \beta h f(t_k, y_k))\right].$$

Determine the relations among s, α, and β such that the local truncation error of this method is $\mathcal{O}(h^2)$ (Hint: let $y'(t) = f(t, y(t))$, find the expression of $y''(t)$ and $y'''(t)$ that involves partial derivatives of f, and the 2nd order Taylor expansion of $f(t_k + \alpha h, y_k + \beta h f(t_k, y_k))$ at (t_k, y_k). Cancel the constant and linear terms of h)

(b) Explore the order of local truncation error and zero stability of

$$\begin{aligned} &y_{k+1} + 4y_k - 5y_{k-1} = h(4f_k + 2f_{k-1}) \\ &y_{k+1} - \tfrac{18}{11}y_k + \tfrac{9}{11}y_{k-1} - \tfrac{2}{11}y_{k-2} = \tfrac{6}{11}hf_{k+1} \end{aligned}$$

For zero-stable method(s), can you show, either by mathematics or numerical evidence, whether the region of absolute stability contains the entire negative real axis? (The region of absolute stability for multistep methods is defined as all values of λh such that $\sum_{k=0}^{s}(\alpha_k - \lambda h \beta_k)z^{s-k}$ has all roots strictly inside the unit circle)

3. **[Q10]** (An implementation/experimental exercise to compare different methods) Consider the ODE initial value problem

$$\begin{cases} y' = -m[y^2 - e^{-2t}(t^2+1)^2] - y + 2te^{-t} \\ y(0) = 1 \end{cases}$$

whose true solution is $y(t) = e^{-t}(t^2 + 1)$ for *all values of* m.

(a) Let $m = 1$ and step size $h = 0.2$. By hand, with the help of a calculator, evaluate y_1 by the forward Euler's method and the implicit midpoint method.

(b) Implement forward/backward Euler, implicit midpoint, RK4, AB4-AM4 predictor/corrector and BDF4, for a single ODE, in MATLAB. The first few y_k of multistep methods can be computed by MATLAB's `ode15s` with machine precision tolerance. Have them ready to numerically solve this problem on time interval $[0, 5]$.

(c) Let $m = 10$, with $N = 25, 35, 45, 55$ uniform steps with step size $h = \frac{5-0}{N}$. Plot the true solution and the computed solution at $\{t_k\}_{k=0}^{N}$, for forward Euler, RK4, and AB4-AM4 ($N = 25, 35, 45$), and backward Euler, implicit midpoint ($N = 25$) and BDF4 (12 plots in total). Explain these plots qualatatively.

(d) With $m = 10$, use forward Euler, implicit midpoint, and RK4, take $N = 100, 200, 400, 800, 1600$ steps, find $\|[y(t_k) - y_k]\|_\infty$. Are the errors consistent with the order?

(e) Let $m = 10^4$. By trial and error, find the smallest N, in a multiple of 1000, such that $\|[y(t_k) - y_k]\|_\infty \le 10^{-4}$, for forward Euler, RK4, and AB4-AM4. Try again for backward Euler, implicit midpoint, and BDF4, but use N in a multiple of 10.

(f) With $m = 10^4$, try MATLAB's `ode45` and `ode15s` with default setting. Report the number of time steps and $\|[y(t_k) - y_k]\|_\infty$.

4. [**Q**11*] (Another practical problem for high order/systems of ODEs, just need a bit work to solve the nonlinear systems for implicit methods)

   Consider the initial value problem for $1 \le t \le 6$

   $$t^3 y''' + 1001 t^2 y'' - 951 t y' + 50000 y = -49049 t, \quad \text{with} \quad y(1) = 1, y'(1) = y''(1) = 0.$$

   The true solution is $y(t) = -t + \frac{2004001}{1002050} t \cos(7 \ln t) - \frac{128993}{1002050} t \sin(7 \ln t) + \frac{99}{1002050 t^{1000}}$.

   (a) Formulate this third order linear ODE as a system of first order ODEs. Explore the eigenvalues of the Jacobian J_f of $f(t,u)$ at $t_0 = 1$ and comment on the stiffness.

   (b) Extend your implicit midpoint, RK4 and BDF4 code to solve systems of 1st order ODEs. Please use Newton's method for solving y_{k+1} in the implicit methods.

   (c) Solve this problem by RK4 with appropriate number of steps $N = 2^p$ (p is an integer). What is the smallest N needed to avoid instabilities?

   (d) Solve this problem by BDF4 with $N = 32, 64, \ldots, 16384$ (multiplied by 2) steps. Evaluate $\|[y(t_k) - y_k]\|_\infty$ for each N and report the run time for $N = 16384$. Compare your results with `ode45` and `ode15s` with default setting and with proper tolerances to achieve $\|[y(t_k) - y_k]\|_\infty$ similar to BDF4 with $N = 16384$.