AM213B Assignment #4

Problem 1 (Theoretical)

Consider BDF2 method

$$\frac{3}{2}u_{n+2}-2u_{n+1}+\frac{1}{2}u_n=hf(u_{n+2},t_{n+2})$$

Part 1:

Use Taylor expansion to show that the local truncation error is $O(h^3)$.

Part 2:

Find the stability polynomial and use the quadratic formula to write out the two roots.

Use the expressions obtained to show that the two roots satisfy

$$\lim_{z\to\infty}\xi_1(z)=0 \text{ and } \lim_{z\to\infty}\xi_2(z)=0.$$

Problem 2 (Theoretical)

Consider the implicit 2-step method below

$$u_{n+2}-u_n=h\left[\frac{1}{3}f(u_{n+2},t_{n+2})+\frac{4}{3}f(u_{n+1},t_{n+1})+\frac{1}{3}f(u_n,t_n)\right]$$

Part 1:

Use Taylor expansion to show that the local truncation error is $O(h^5)$.

Hint:

Use $f(u(t_{n+j}), t_{n+j}) = u'(t_{n+j})$ and expand every term around t_{n+1}.

Part 2:

The stability polynomial is

$$\pi(\xi,z)=\left(\xi^2-1\right)-z\left(\frac{1}{3}\xi^2+\frac{4}{3}\xi+\frac{1}{3}\right)$$

Let $z = -\varepsilon$ with small $\varepsilon > 0$, we examine the two roots of $\pi(\xi, z)$.

Show that the two roots $\xi_1(\varepsilon)$ and $\xi_2(\varepsilon)$ satisfy

$$\xi_1(\varepsilon)=1-\varepsilon+O(\varepsilon^2), \qquad \xi_2(\varepsilon)=-\left(1+\frac{\varepsilon}{3}\right)+O(\varepsilon^2)$$

Hint:

Using the quadratic formula and then expand in terms of ε.

Remarks:

- This method demonstrates that the Dahlquist barrier on accuracy of implicit LMM: $p \le r + 2$ is actually attainable.
- Although this method has the 4th order accuracy, it is not practically useful (similar to the situation with the 2-step midpoint method).

Problem 3 (Computational)

Plot the region of absolute stability for each of the methods below.

- BDF2
- BDF3
- BDF4

Hint:

- We learned that all BDF methods satisfy $\lim_{z\to\infty} \xi_j(z) = 0$, which implies that the region of stability is the exterior of its boundary, instead of the interior.
- See sample code on how to shade the exterior of a closed loop.

Problem 4 (Computational)

Implement the finite difference method to solve the two-point BVP.

$$\begin{cases} u'' - q\cdot\left(1+\cos^2 x\right)u = -\exp(4\sin x) \\ u(0)=1, \quad u(3)=1.5 \end{cases}$$

Solve the BVP for q = 10 with h = 0.01 (see hint below).

Part 1:

Plot $u(x)$ vs x.

Part 2:

Solve the BVP with $h_2 = h/2$ and do error estimation. Plot the estimated error vs x.

Use linear scales for both x and error. Do not take the absolute value of error.

Part 3:

Repeat Part 1 & Part 2 above for q = 100 and compare the results.

Hint:

Note that $h = 0.01$ corresponds to $N1 = N + 1 = 3/0.01 = 300$ where $N1$ is # of sub-intervals and N is # of internal points (# of unknown u's).

Problem 5 (Computational)

Consider the linear first order ODE system below

$$\frac{d\vec{u}(t)}{dt} = A\vec{u}(t) + \vec{b}(t) \qquad \text{(E01)}$$

$$\vec{u}(t) = \begin{pmatrix} u(1,t) \\ u(2,t) \\ \vdots \\ u(m,t) \end{pmatrix}_{m\times 1}, \quad A = a\begin{pmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix}_{m\times m}, \quad \vec{b}(t) = a\begin{pmatrix} \sin(\pi t) \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{m\times 1}$$

where A is an $m\times m$ tridiagonal matrix given above.

Implement the 2S-DIRK method to solve the ODE system with initial value $\vec{u}(0) = \vec{0}$.

Part 1:

Use m = 399, a = 3×10^4, h = 0.02, and solve to T = 10.

Plot $u(5, t)$ vs t, $u(80, t)$ vs t, and $u(200, t)$ vs t in one figure.

Part 2:

Solve the ODE system with $h_2 = h/2$ and estimate the errors of $u(5, t)$, $u(80, t)$ and $u(200, t)$. Plot the three errors vs t in one figure. Use linear scales for both t and error. Do not take the absolute value of error. Near t = 0, you will have relatively large error. That is expected.

Hint:

- In Matlab, use "diag" to construct matrix A.
- The implementation of the 2S-DIRK is described below.

The 2S-DIRK is specified by its Butcher tableau.

$$\text{Butcher tableau:} \quad \begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array}, \qquad \alpha = 1 - \frac{1}{\sqrt{2}}$$

We apply it to (E01). For simplicity, we drop the vector notation: $\vec{u} \to u, \vec{b} \to b$. We first write it in terms of $f(u, t)$ and then replace $f(u, t)$ with $Au+b(t)$.

$$k_1 = \Delta t\, f(u_n + \alpha k_1, t_n + \alpha\Delta t)$$

$$\text{==>} \qquad k_1 = \Delta t\, A(u_n + \alpha k_1) + \Delta t\, b(t_n + \alpha\Delta t)$$

$$\text{==>} \qquad (I - \Delta t\alpha A)k_1 = \Delta t\, A u_n + \Delta t\, b(t_n + \alpha\Delta t)$$

To calculate vector k_1, we only need to solve the linear system above!

In Matlab, this linear system is solved using

$$k_1 = (I - \Delta t \alpha A) \backslash \left(\Delta t\, A u_n + \Delta t\, b(t_n + \alpha \Delta t) \right)$$

For vector k_2, we have the linear system

$$(I - \Delta t \alpha A) k_2 = \Delta t\, A(u_n + (1-\alpha) k_1) + \Delta t\, b(t_n + \Delta t)$$

In Matlab, this linear system is solved using

$$k_2 = (I - \Delta t \alpha A) \backslash \left(\Delta t\, A(u_n + (1-\alpha) k_1) + \Delta t\, b(t_n + \Delta t) \right)$$

Once we obtain vectors k_1 and k_2, we calculate u_{n+1}.

$$u_{n+1} = u_n + (1-\alpha) k_1 + \alpha k_2$$

<u>In summary</u>, when implementing the 2S-DIRK to solving $\vec{u}'(t) = A\vec{u} + \vec{b}(t)$, we do not need to solve any non-linear system.