Advanced Numerical Analysis, MATH 269A
E. K. Ryu
Fall 2024

Homework 4
Due on Wednesday, November 27, 2024.

Problem 1: *Stability function adjoint methods.* Consider an s-stage RK method with stability function $\psi(z)$. Show that its adjoint method has stability function $1/\psi(-z)$.

Problem 2: *Implementing implicit midpoint.* Consider the Runge–Kutta method expressed by the Butcher Tableau

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

(a) Show that the method can be equivalently expressed as

$$y_{n+1} = y_n + hf\big(t_n + \tfrac{h}{2}, \tfrac{1}{2}(y_n + y_{n+1})\big)$$

(b) Let

$$G(y) = y - y_n - hf\big(t_n + \tfrac{h}{2}, \tfrac{1}{2}(y_n + y)\big).$$

Show that the Newton iteration for finding a root of G is

$$y = y - \Big(I - \tfrac{h}{2} f_y\big(t_n + \tfrac{h}{2}, \tfrac{1}{2}(y_n + y)\big)\Big)^{-1} G(y),$$

where

$$f_y(t, y) = \frac{\partial f}{\partial y}(t, y).$$

(c) Consider the ODE

$$y' = -100(y - \sin(t)), \qquad y(0) = 1$$

for $t \in [0, 3]$. Implement the midpoint method in Python.

Problem 3: *Implementing Gauss–Legendre.* Consider the Runge–Kutta method expressed by the Butcher Tableau

$$
\begin{array}{c|cc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
$$

(a) Show that the method can be equivalently expressed as

$$
\begin{aligned}
k_{n+1/3} &= f\big(t_n + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h, y_n + \tfrac{1}{4}hk_{n+1/3} + (\tfrac{1}{4} - \tfrac{\sqrt{3}}{6})hk_{n+2/3})\big) \\
k_{n+2/3} &= f\big(t_n + (\tfrac{1}{2} + \tfrac{\sqrt{3}}{6})h, y_n + (\tfrac{1}{4} + \tfrac{\sqrt{3}}{6})hk_{n+1/3} + \tfrac{1}{4}hk_{n+2/3})\big) \\
y_{n+1} &= y_n + \tfrac{h}{2}(k_{n+1/3} + k_{n+2/3})
\end{aligned}
$$

(b) Show that the method can be equivalently expressed as

$$
\begin{aligned}
\underbrace{\begin{bmatrix} y_{n+1/3} \\ y_{n+2/3} \end{bmatrix}}_{=\vec{y}} &= y_n\mathbf{1} + h \underbrace{\begin{bmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{bmatrix}}_{=M} \underbrace{\begin{bmatrix} f\big(t_n + (\frac{1}{2} - \frac{\sqrt{3}}{6})h, y_{n+1/3}\big) \\ f\big(t_n + (\frac{1}{2} + \frac{\sqrt{3}}{6})h, y_{n+2/3}\big) \end{bmatrix}}_{=F(\vec{y})} \\
y_{n+1} &= y_n + \tfrac{h}{2}\mathbf{1}^\intercal F(\vec{y}).
\end{aligned}
$$

(c) Let

$$G(\vec{y}) = \vec{y} - y_n\mathbf{1} - hMF(\vec{y}).$$

Show that the Newton iteration for finding a root of G is

$$\vec{y} = \vec{y} - \left(I - hM \begin{bmatrix} f_y\big(t_n + (\frac{1}{2} - \frac{\sqrt{3}}{6})h, y_{n+1/3}\big) & 0 \\ 0 & f_y\big(t_n + (\frac{1}{2} + \frac{\sqrt{3}}{6})h, y_{n+2/3}\big) \end{bmatrix}\right)^{-1} G(\vec{y}),$$

where

$$f_y(t, y) = \frac{\partial f}{\partial y}(t, y).$$

(d) Consider the ODE

$$y' = -100(y - \sin(t)), \qquad y(0) = 1$$

for $t \in [0, 3]$. Implement the Gauss–Legendre method in Python.

Problem 4: *Cost of Newton for implicit RK.* Consider a fully implicit s-stage RK method with Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

applied to $y' = f(t, y)$, where $y \in \mathbb{R}^d$.

(a) Show that $y_n \mapsto y_{n+1}$ can be obtained by solving a (sd)-dimensional root-finding problem.

(b) Show that Newton's iteration for finding such a root requires inverting $(sd) \times (sd)$ matrices. (Therefore, if K Newton steps are required, the cost is $\mathcal{O}(Ks^3d^3)$ flops.)

Problem 5: *Diagonally implicit RK method.* A Diagonally Implicit RK (DIRK) method is an RK method with a lower triangular Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \dots & 0 \\ c_2 & a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

(Recall that explicit RK methods have *strictly* lower triangular Butcher tableaus.) Consider the DIRK method applied to $y' = f(t, y)$, where $y \in \mathbb{R}^d$.

(a) Show that $y_n \mapsto y_{n+1}$ can be obtained by solving s separate d-dimensional root-finding problems.

(b) Show that Newton's iteration for finding such roots requires inverting $d \times d$ matrices. (Therefore, if K Netwton steps are required for each of the s stages, the cost is $\mathcal{O}(Ksd^3)$ flops.)

(c) Consider the ODE

$$y' = -100(y - \sin(t)), \qquad y(0) = 1$$

for $t \in [0, 3]$. Implement the DIRK method

$$\begin{array}{c|ccc} x & x & 0 & 0 \\ \frac{1+x}{2} & \frac{1-x}{2} & x & 0 \\ 1 & -3x^2/2 + 4x - 1/4 & 3x^2/2 - 5x + 5/4 & x \\ \hline & -3x^2/2 + 4x - 1/4 & 3x^2/2 - 5x + 5/4 & x \end{array}$$

with $x = 0.4358665215$. (The $x = 0.4358665215$ is a root of the polynomial $6x^3 - 18x^2 + 9x - 1$.)

Remark. It can be shown that DIRK methods have an order barrier of $p \leq s + 1$.

Problem 6: *Splitting method for a simple pendulum.* Consider the ODE

$$\begin{aligned} q' &= \frac{p}{mL^2} & \qquad q(0) &= 0 \in \mathbb{R} \\ p' &= -mgL\sin q & \qquad p(0) &= 1 \in \mathbb{R} \end{aligned}$$

for $t \in [0, 10]$. This is the Hamiltonian dynamics of a simple pendulum with angle q from the vertical (in radians), angular momentum $p = mL^2 q'$, mass m, (rigid) string length L, and gravitational acceleration $g = 9.81$. For simplicity, set $m = 1$ and $L = 1$. Consider the splitting

$$\frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} \frac{p}{mL^2} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -mgL\sin q \end{bmatrix}.$$

Implement the Lie–Trotter and Strang splitting methods.

Problem 7: *Baker–Campbell–Hausdorff formula.* Consider the linear ODE

$$y' = \underbrace{(B + C)}_{=A}\, y, \qquad y(0) = y_0.$$

Consider using the Lie–Trotter splitting method, and consider the local error $e^{hA}y_0 - e^{hC}e^{hB}y_0$. Show

$$e^{hA} - e^{hC}e^{hB} = \frac{h^2}{2}[B, C] + \mathcal{O}(h^3),$$

where $[B, C] = BC - CB$ is the *commutator* of the matrices B and C.