Homework 3

AM213B

Kevin Silberberg

2025-04-29

Problem 1

Part 1

Derive the stability function $\phi(z)$ for each of the two RK methods below

- predictor-corrector method (Heun's method)
- Classic 4-th order RK4 method

Solution

The **predictor-corrector method (Heun's method)** is as follows,

$$k_1 = f(u_n, t_n) \tag{1}$$
$$k_2 = f(u_n + hk_1, t_n + h) \tag{2}$$
$$u_{n+1} = u_n + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right) \tag{3}$$

The Butcher Array for the method is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{4}$$

We apply the above method to the test equation $\frac{du}{dt} = f(u,t) = \lambda u$ such that $f(u_n, t_n) = \lambda u_n$.

We know that the stability function $\phi(z)$ can be written[1]

$$\phi(z) = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{1} \tag{5}$$

where $\mathbf{1}$ is a column vector of all 1's of length s, and $z = h\lambda$

So the stability function $\phi(z)$ for Heun's method corresponding to the Butcher array (4) is,

[1]"Linear Multistep Methods" (2003)

$$
\begin{aligned}
\phi(z) &= 1 + z \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - z \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} && (6) \\
&= 1 + z \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -z & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} && (7) \\
&= 1 + z \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} && (8) \\
&= 1 + z \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ z+1 \end{bmatrix} && (9) \\
&= 1 + z \left(\frac{1}{2} + \frac{1}{2} (z+1) \right) && (10) \\
&= 1 + z \left(1 + \frac{1}{2} z \right) && (11) \\
&= 1 + z + \frac{1}{2} z^2 && (12)
\end{aligned}
$$

The **RK4 method** has the following butcher array,

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array} \tag{13}
$$

Let us write down the stability function $\phi(z)$ using the formula (5)

$$\phi(z) = 1 + z\begin{bmatrix}\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}\end{bmatrix}\left(\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} - z\begin{bmatrix}0 & 0 & 0 & 0\\ \frac{1}{2} & 0 & 0 & 0\\ 0 & \frac{1}{2} & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\right)^{-1}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \tag{14}$$

$$= 1 + z\begin{bmatrix}\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ \frac{z}{2} & 1 & 0 & 0\\ 0 & \frac{z}{2} & 1 & 0\\ 0 & 0 & z & 1\end{bmatrix}^{-1}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \tag{15}$$

$$= 1 + z\begin{bmatrix}\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ \frac{z}{2} & 1 & 0 & 0\\ \frac{z^2}{4} & \frac{z}{2} & 1 & 0\\ \frac{z^3}{4} & \frac{z^2}{2} & z & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \tag{16}$$

$$= 1 + z\begin{bmatrix}\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}\end{bmatrix}\begin{bmatrix}1\\ \frac{z}{2}+1\\ \frac{z^2}{4}+\frac{z}{2}+1\\ \frac{z^3}{4}+\frac{z^2}{2}+z+1\end{bmatrix} \tag{17}$$

$$= 1 + z\left(\frac{1}{6}\cdot 1 + \frac{1}{3}\cdot(\frac{z}{2}+1) + \frac{1}{3}\cdot(\frac{z^2}{4}+\frac{z}{2}+1) + \frac{1}{6}\cdot(\frac{z^3}{4}+\frac{z^2}{2}+z+1)\right) \tag{18}$$

$$= 1 + z\left(\frac{1}{6}+\frac{1}{3}+\frac{1}{3}+\frac{1}{6}+\frac{z}{6}+\frac{z}{6}+\frac{z}{6}+\frac{z^2}{12}+\frac{z^2}{12}+\frac{z^3}{24}\right) \tag{19}$$

$$= 1 + z\left(1+\frac{z}{2}+\frac{z^2}{6}+\frac{z^3}{24}\right) \tag{20}$$

$$= 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \tag{21}$$

Part 2

Plot the region of absolute stability (S) for each of the two methods.

Solution

The ‘stability region’, defined as $\{z \in \mathbb{C} : |R(z)| \leq 1\}$, is the set of points in the complex plane such that the computed solution remains bounded after many steps of computation.

Let us plot the regions of stability for each method in the complex plane.

Problem 2

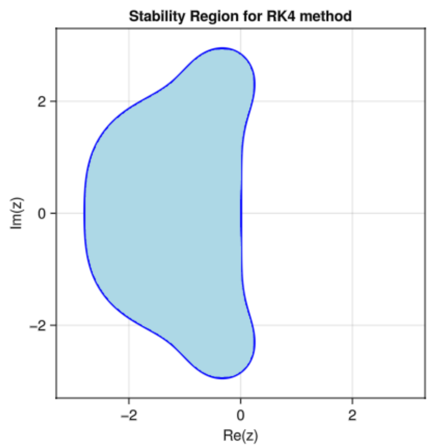
Consider the RK method described by the following Butcher array,

$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array} \tag{22}$$

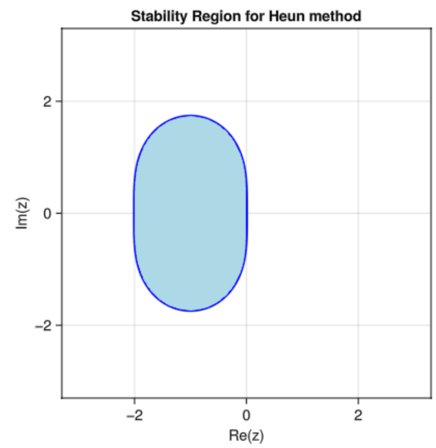
where $\alpha > 0$. This is called a 2s-DIRK (2-Stage Diagonally Implicit Runge-Kutta) method.

Part 1

Show that the method is second order for $\alpha = 1 - \frac{1}{\sqrt{2}}$.



(a)



(b)

Figure 1: Stability regions for numerical methods. Figure 1a The Classic RK4 method and Figure 1b the Heun method, both with stability regions shown in lightblue.

Solution

Given the Butcher array and corresponding values for $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, we will demonstrate that the method is second order by verifying the order conditions derived from rooted trees up to order three.

Order conditions are constructed by comparing a polynomial $\Phi(t)$ against the density $\gamma(t)$ for all possible rooted trees t of each order. The order of a tree $|t|$ is the total number of nodes (including all leaf and root nodes).

The order condition is defined as:

$$\Phi(t) = \frac{1}{\gamma(t)}$$

To compute $\gamma(t)$:

1. Label each leaf node with value 1
2. Label each non-leaf node with a value of 1 plus the sum of its child nodes' values
3. Calculate $\gamma(t)$ by taking the product of all labeled values in the tree

To construct $\Phi(t)$:

1. Label the root node with index i
2. Label each subsequent non-leaf node with indices $j, k, l, m, \ldots$
3. Start with factor b_i
4. For each edge not terminating in a leaf node, add factor a_{pq} where:
   - $p =$ parent node label
   - $q =$ child node label
5. For each leaf node, add factor c_p where p is the parent node label
6. Sum the product of all factors across all possible label combinations

Let us proceed with determining the order of the 2s-DIRK method.

Recall from the butcher array (22) we have

$$\mathbf{A} = \begin{bmatrix} \alpha & 0 \\ 1-\alpha & \alpha \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1-\alpha \\ \alpha \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} \alpha \\ 1 \end{bmatrix} \tag{23}$$

For order 1 there is only one possible tree, the root node.

The tree corresponding to order 1 is

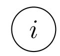


Figure 2: Tree of order 1

the density is $\gamma(t) = 1$ the polynomial is $\Phi(t) = \sum_{i=1}^{2} b_i$, thus the order condition is

$$\begin{aligned} \sum_{i=1}^{2} b_i &= 1 \qquad (24) \\ b_1 + b_2 &= 1 \qquad (25) \\ (1-\alpha) + \alpha &= 1 \qquad (26) \\ 1 &= 1 \qquad (27) \end{aligned}$$

The above is true so let us proceed to order 2.

There is only one possible tree,

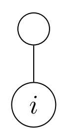


Figure 3: Tree of order 2

the density $\gamma(t) = 2$ and the polynomial is $\Phi(t) = \sum_{i=1}^{2} b_i c_i$, thus the order condition is

$$\begin{aligned} \sum_{i=1}^{2} b_i c_i &= \frac{1}{2} \qquad (28) \\ b_1 c_1 + b_2 c_2 &= \frac{1}{2} \qquad (29) \\ \alpha(1-\alpha) + \alpha &= \frac{1}{2} \qquad (30) \\ \alpha - \alpha^2 + \alpha &= \frac{1}{2} \qquad (31) \\ \alpha\,(2-\alpha) &= \frac{1}{2} \qquad (32) \\ \left(1 - \frac{1}{\sqrt{2}}\right)\left(2 - 1 + \frac{1}{\sqrt{2}}\right) &= \frac{1}{2} \qquad (33) \\ \left(1 - \frac{1}{\sqrt{2}}\right)\left(1 + \frac{1}{\sqrt{2}}\right) &= \frac{1}{2} \qquad (34) \\ 1 + \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} - \frac{1}{2} &= \frac{1}{2} \qquad (35) \\ \frac{1}{2} &= \frac{1}{2} \qquad (36) \end{aligned}$$

The above is true so let us proceed to order 3,

Figure 4: Trees of order 3

For the two trees above we have the following order conditions:

$$\sum_{i,j=0}^{2} b_i a_{ij} c_j = \frac{1}{6} \quad \text{(a)} \tag{37}$$

$$b_i a_{11} c_1 + b_1 a_{12} c_2 + b_2 a_{21} c_1 + b_2 a_{22} c_2 = \frac{1}{6} \tag{38}$$

$$\sum_{i=1}^{2} b_i c_i^2 = \frac{1}{3} \quad \text{(b)} \tag{39}$$

$$b_1 c_1^2 + b_2 c_2^2 = \frac{1}{3} \tag{40}$$

Let us carry out the calculation on the LHS of equation (38) corresponding to tree (a)

$$(1-\alpha)\alpha^2 + \alpha(1-\alpha)^2 + \alpha^2 = \frac{1}{6} \tag{41}$$

$$2\alpha^2 - \alpha^3 + \alpha\left(1 - 2\alpha + \alpha^2\right) = \frac{1}{6} \tag{42}$$

$$2\alpha^2 - \alpha^3 + \alpha - 2\alpha^2 + \alpha^3 = \frac{1}{6} \tag{43}$$

$$\alpha = \frac{1}{6} \tag{44}$$

Since $\alpha = 1 - \frac{1}{\sqrt{2}} \neq \frac{1}{6}$ we know that the 2s-DIRK method is not represented by tree (a).

Let us carry out the calculation on the LHS of equation (40) corresponding to tree (b)

$$(1-\alpha)\,\alpha^2 + \alpha = \frac{1}{3} \tag{45}$$

$$\alpha + \alpha^2 - \alpha^3 = \frac{1}{3} \tag{46}$$

$$\left(1 - \frac{1}{\sqrt{2}}\right) + \left(1 - \frac{1}{\sqrt{2}}\right)^2 - \left(1 - \frac{1}{\sqrt{2}}\right)^3 = \frac{1}{3} \tag{47}$$

$$1 - \frac{1}{\sqrt{(2)}} + \left(1 - \frac{2}{\sqrt{2}} + \frac{1}{2}\right) - \left(1 - \frac{1}{\sqrt{2}} - \frac{2}{\sqrt{2}} + 1 + \frac{1}{2} - \frac{1}{2\sqrt{2}}\right) = \frac{1}{3} \tag{48}$$

$$\frac{1}{2\sqrt{2}} = \frac{1}{3} \tag{49}$$

Thus we can conclude that the 2s-DIRK method with $\alpha = 1 - \frac{1}{\sqrt{2}}$ has the form of the tree in Figure 3, and that it is order 2.

Part 2

Show that the stability function $\phi(z)$ is

$$\phi(z) = \frac{1 + (1 - 2\alpha)z}{(1 - \alpha z)^2} \tag{50}$$

Solution

Given $\mathbf{c}, \mathbf{b}, \mathbf{A}$ the stability function $\phi(z)$ of an RK method can be computed by the following[2],

$$\phi(z) = \frac{\det\left(\mathbf{I} + z\left(\mathbf{1}\mathbf{b}^T - \mathbf{A}\right)\right)}{\det\left(\mathbf{I} - z\mathbf{A}\right)} \tag{51}$$

from the butcher array (22) we compute the stability function as follows,

$$\phi(z) = \frac{\det\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + z\left(\begin{bmatrix}1\\ 1\end{bmatrix}\begin{bmatrix}1-\alpha & \alpha\end{bmatrix} - \begin{bmatrix}\alpha & 0\\ 1-\alpha & \alpha\end{bmatrix}\right)\right)}{\det\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - z\begin{bmatrix}\alpha & 0\\ 1-\alpha & \alpha\end{bmatrix}\right)} \tag{52}$$

$$= \frac{\det\left(\begin{bmatrix}1 + z(1-2\alpha) & z\alpha\\ 0 & 1\end{bmatrix}\right)}{\det\left(\begin{bmatrix}1 - z\alpha & 0\\ -z(1-\alpha) & 1 - z\alpha\end{bmatrix}\right)} \tag{53}$$

$$= \frac{1 + z(1 - 2\alpha)}{(1 - \alpha z)^2} \tag{54}$$

Part 3

Plot the region of absolute stability (S) for the 2s-DIRK with $\alpha = 1 - \frac{1}{\sqrt{2}}$.

Numerically, conclude that the 2s-DIRK with $\alpha = 1 - \frac{1}{\sqrt{2}}$ is A-stable.

Solution

Let us plot the region of absolute stability

According to the book of Butcher[3], we have the following definition for A-stability of RK methods.

A Runge-Kutta method is A-stable if its stability function satisfies

$$|\phi(z)| \leq 1, \quad \text{whenever } Re(z) \leq 0 \tag{55}$$

where $Re(z)$ denotes the real part of the complex number z.

Clearly from the region of stability in Figure 5 we can see that the condition for A-stability is satisfied because for all complex values with zero or negative real part the stability function is less than or equal to 1.

[2] "Linear Multistep Methods" (2003)
[3] "Linear Multistep Methods" (2003)

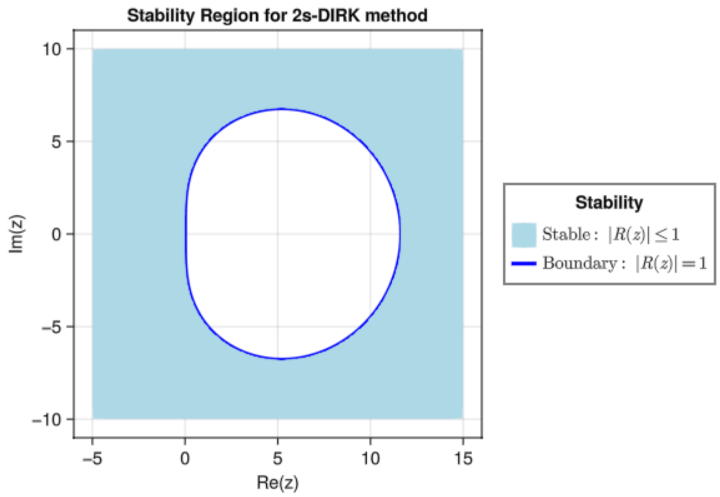


Figure 5: The Region of absolute stability for the 2s-DIRK method with $\alpha = 1 - \frac{1}{\sqrt{2}}$

Problem 3

Part 1

Study the zero-stability for each of the three LMMs below

$$u_{n+2} - 2u_{n+1} + u_n = h\left(f(u_{n+1}, t_{n+1}) - f(u_n, t_n)\right) \tag{56}$$

$$u_{n+2} - u_n = \frac{h}{3}\left(f(u_{n+2}, t_{n+2}) + 4f(u_{n+1}, t_{n+1}) + f(u_n, t_n)\right) \tag{57}$$

$$u_{n+2} - \frac{4}{3}u_{n+1} + \frac{1}{3}u_n = \frac{2}{3}hf(u_{n+2}, t_{n+2}) \tag{58}$$

Solution

An LMM satisfies the 'root condition' if the zeros of the polynomial

$$\rho(z) = \sum_{j=0}^{q} \alpha_j z^j \tag{59}$$

are within the unit circle, and those of modulus one are simple.

A numercial method of the form

$$\sum_{j=0}^{q} \alpha_j u_{n+j} = h \sum_{j=0}^{q} \beta_j f(u_{n+j}, t_{n+j}) \tag{60}$$

is zero-stable if and only if it satisfies the root condition.

Let us compute the characteristic polynomial for each numerical method in Part 1.

For (56) we have

$$\alpha_0 = 1 \qquad \beta_0 = -1 \tag{61}$$
$$\alpha_1 = -2 \qquad \beta_1 = 1 \tag{62}$$
$$\alpha_2 = 1 \qquad \beta_2 = 0 \tag{63}$$

the characteristic polynomials are

$$\rho(z) = z^2 - 2z + 1 \tag{64}$$
$$\sigma(z) = z - 1 \tag{65}$$

Let us find the roots of the characteristic polynomial $\rho(z)$

$$z^2 - 2z + 1 = 0 \tag{66}$$
$$z = \frac{2 \pm \sqrt{(-2)^2 - 4}}{2} \tag{67}$$
$$z = 1 \text{ with algebraic multiplicity } 2 \tag{68}$$

thus (56) does not satisfy the root condition for zero-stability.

Let us study the zero-stabilty for (57).

The coefficients are

$$\alpha_0 = -1 \qquad \beta_0 = \frac{1}{3} \tag{69}$$
$$\alpha_1 = 0 \qquad \beta_1 = \frac{4}{3} \tag{70}$$
$$\alpha_2 = 1 \qquad \beta_2 = \frac{1}{3} \tag{71}$$

the characteristic polynomials are

$$\rho(z) = z^2 - 1 \tag{72}$$
$$\sigma(z) = \frac{1}{3}\left(z^2 + 4z + 1\right) \tag{73}$$

the roots of the polynomial $\rho(z)$ are,

$$z^2 - 1 = 0 \tag{74}$$
$$z^2 = 1 \tag{75}$$
$$z = 1 \text{ with multiplicity } 2 \tag{76}$$

thus (57) also does not satisfy the root condition for zero-stability.

let us study the zero-stability for (58)

the coefficients are

$$\alpha_0 = \frac{1}{3} \qquad \beta_0 = 0 \tag{77}$$
$$\alpha_1 = -\frac{4}{3} \qquad \beta_1 = 0 \tag{78}$$
$$\alpha_2 = 1 \qquad \beta_2 = \frac{2}{3} \tag{79}$$

the characteristic equations are

$$\rho(z) = z^2 - \frac{4}{3}z + \frac{1}{3} \tag{80}$$
$$\sigma(z) = \frac{2}{3}z^2 \tag{81}$$

the roots of the polynomial $\rho(z)$ are,

$$z^2 - \frac{4}{3}z + \frac{1}{3} = 0 \tag{82}$$
$$z = \frac{4}{3} \cdot \frac{1}{2} \pm \frac{1}{2} \cdot \sqrt{(-\frac{4}{3})^2 - 4 \cdot \frac{1}{3}} \tag{83}$$
$$= \frac{2}{3} \pm \frac{1}{3} \tag{84}$$
$$z_1 = 1 \tag{85}$$
$$z_2 = \frac{1}{3} \tag{86}$$

thus (58) satisfies the root condition for zero-stability.

Part 2

For (57), use the Taylor series expansion to show that the local truncation error $e_n(h) = O(h^5)$.

Solution

We find the local truncation error by considering the difference between the exact solution and the approximate solution for one time step.

Let us write down the numerical method.

$$u_{n+2} - u_n = \frac{h}{3}\left[f(u_{n+2}, t_{n+2}) + 4f(u_{n+1}, t_{n+1}) + f(u_n, t_n)\right] \tag{87}$$

Let us write down the local truncation error formula for this method.

$$e_n(h) = \underbrace{u(t_{n+2}) - u(t_n)}_{exact} - \underbrace{(u_{n+2} - u_n)}_{approximate} \tag{88}$$

Notice that we can center all the terms about the point t_{n+1} by offsetting by the timestep h. So t_{n+2} becomes $t_{n+1} + h$ and t_n becomes $t_{n+1} - h$.

Let us taylor series expand the exact solution about the point t_{n+1}.

For the following we will use the short hand notation

$$f(u_{n+1}, t_{n+1}) = f(u(t_{n+1}), t_{n+1}) = f \tag{89}$$

$$u(t_{n+1}) = u \tag{90}$$

$$\frac{d^m}{dt^m}(u(t_{n+1})) = u^{(m)} \tag{91}$$

$$u(t_{n+1} + h) = u + hu^{(1)} + \frac{h^2}{2!}u^{(2)} + \frac{h^3}{3!}u^{(3)} + \frac{h^4}{4!}u^{(4)} + O(h^5) \tag{92}$$

$$u(t_{n+1} - h) = u - hu^{(1)} + \frac{h^2}{2!}u^{(2)} - \frac{h^3}{3!}u^{(3)} + \frac{h^4}{4!}u^{(4)} + O(h^5) \tag{93}$$

$$u(t_{n+2}) - u(t_n) = 2hu^{(1)} + \frac{h^3}{3}u^{(3)} + O(h^5) \tag{94}$$

Now let us taylor series expand the RHS of (87) centered at the time step t_{n+1}

$$f(u_n, t_n) = f(u(t_{n+1} - h), t_{n+1} - h) = f - hf^{(1)} + \frac{h^2}{2!}f^{(2)} - \frac{h^3}{3!}f^{(3)} + \frac{h^4}{4!}f^{(4)} + O(h^5) \tag{95}$$

$$f(u_{n+1}, t_{n+1}) = f(u(t_{n+1}), t_{n+1}) = f \tag{96}$$

$$f(u_{n+2}, t_{n+2}) = f(u(t_{n+1} + h), t_{n+1} + h) = f + hf^{(1)} + \frac{h^2}{2!}f^{(2)} + \frac{h^3}{3!}f^{(3)} + \frac{h^4}{4!}f^{(4)} + O(h^5) \tag{97}$$

Let us plug this into the RHS of (87)

$$\frac{h}{3}\left[6f + h^2 f^{(2)} + \frac{h^4}{12}f^{(4)} + O(h^6)\right] \tag{98}$$

Let us plug all this back into (88), while recalling that for the general ODE we have $u^{(m)} = f^{(m-1)}$

$$e_n(h) = 2hu^{(1)} + \frac{h^3}{3}u^{(3)} - 2hf - \frac{h^3}{3}f^{(2)} - \frac{h^5}{36}f^{(4)} + O(h^7) \tag{99}$$

$$= 2hf + \frac{h^3}{3}f^{(2)} - 2hf - \frac{h^3}{3}f^{(2)} - \frac{h^5}{36}f^{(4)} + O(h^7) \tag{100}$$

$$= O(h^5) \tag{101}$$

Part 3

Plot the region of absolute stability for (57) and (58).

Numerically, conclude that (58) is A-stable.

Solution

For a general linear multistep method (LMM) of the form

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = k \sum_{j=0}^{r} \beta_j f(u_{n+j}, t_{n+j}) \tag{102}$$

the region of absolute stability is found by applying the method to the test equation

$$\frac{du}{dt} = \lambda u \tag{103}$$

obtaining

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = k \sum_{j=0}^{r} \beta_j \lambda u_{n+j} \tag{104}$$

which can be rewritten

$$\sum_{j=0}^{r} (\alpha_j - z\beta_j) u_{n+j} = 0 \tag{105}$$

the solution has the general form

$$u_n = c_1 \xi_1^n + c_2 \xi_2^n + \cdots + c_r \xi_r^n \tag{106}$$

where ξ_j are now the roots of the characteristic polynomial

$$\sum_{j=0}^{r} (\alpha_j - z\beta_j) \xi^j \tag{107}$$

this is the stability polynomial $\pi(\xi; z)$. It is a polynomial in ξ but its coefficients depend on the value $z = \lambda h$. The polynomial can be expressed in terms of the characteristic polynomials for the LMM computed above in part 2.

$$\pi(\xi; z) = \rho(\xi) - z\sigma(\xi) \tag{108}$$

Let us compute the stability polynomial for (57) and (58).

For (57) we have

$$\pi(\xi; z) = \rho(\xi) - z\sigma(\xi) \tag{109}$$

$$= \xi^2 - 1 - \frac{z}{3}(\xi^2 + 4\xi + 1) \tag{110}$$

for (58) we have

$$\pi(\xi; z) = \rho(\xi) - z\sigma(\xi) \tag{111}$$

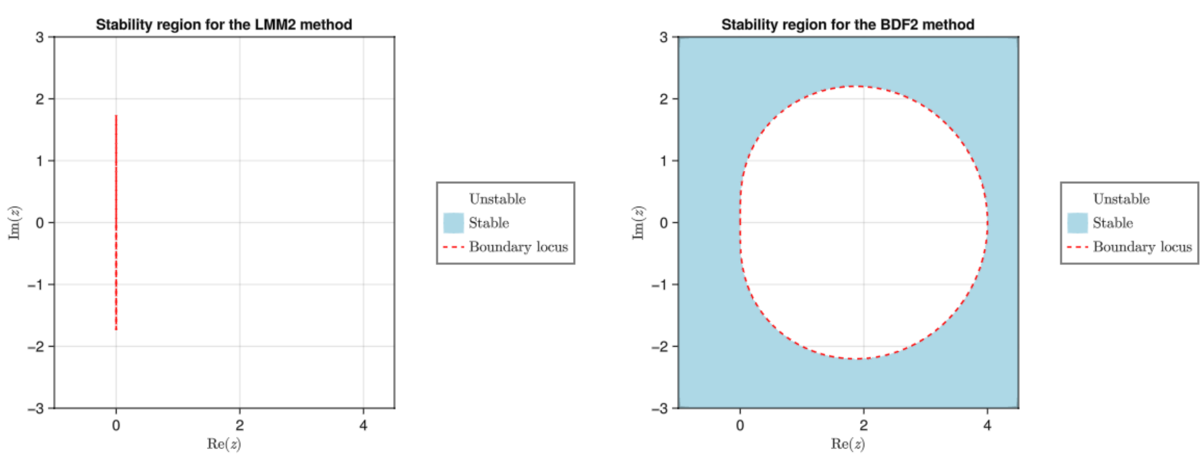
$$= \xi^2 - \frac{4}{3}\xi + \frac{1}{3} - \frac{2z}{3}\xi^2 \tag{112}$$

we can simply plot the parametrized curve

$$\tilde{z}(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} \tag{113}$$

for $0 \leq \theta \leq 2\pi$ to find the locus of all points which are potentially on the boundary, then to find which side of the curve is the *interior* of $\mathcal{S}$, we need only evaluate the roots of $\pi(\xi; z)$ at some random point z on one side or the other and see if the polynomial satisfies the root condition.

Let us implement the above algorithm and plot the stability regions.



(a) (b)

Figure 6: Stability Regions for (57) (a) and (58) (b). We can immediately conclude that the method (58) (the BDF2 method) is A-stable because the left-half complex plane is entirely stable. Additionally, for the LMM2 method, the region collapses to the interval [-i, i] on the imaginary axis. Meaning the method is unconditionally absolutely unstable and there is no hope to simulate accurately a linear dynamical system that has an attractor at the origin.

Problem 4

Read the sample code on implementing a 3-stage DIRK method. Write your code to implement the 2s-DIRK method (22) in problem 2 with $\alpha = 1 - \frac{1}{\sqrt{2}}$ to solve the following IVP.

$$\begin{cases} \frac{du}{dt} & = -\left(\frac{1}{2} + e^{20\cos(1.3t)}\right)\sinh(u - \cos(t)) \\ u(0) & = 0 \end{cases} \tag{114}$$

In the time domain $t \in [0, 30]$

Part 1

Plot the numerical solution $u(t)$ vs t for $h = 2^{-6}$.

Plot $\cos(t)$ vs t in the same figure for comparison.

Does the solution $u(t)$ always follow the function $\cos(t)$ very closely?

Solution

Let us plot the numerical solution over the function $\cos(t)$ for the time domain $t \in [0, 30]$.

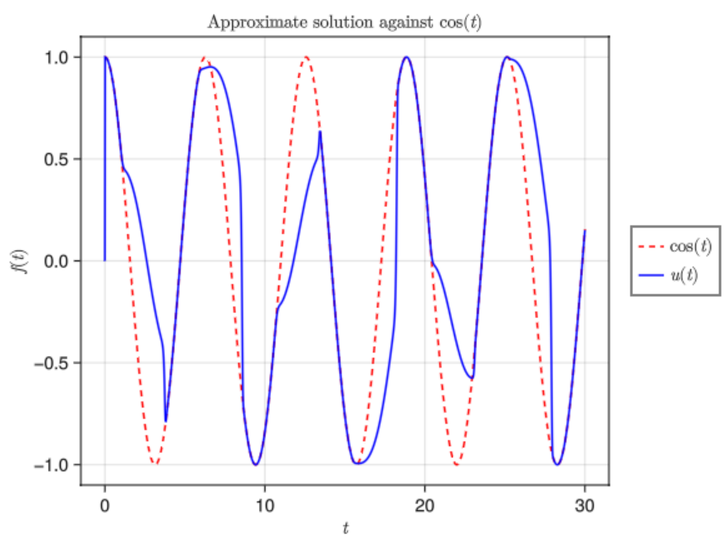


Figure 7: Comparison of the numerical solution to the ODE (114) in the time domain $t \in [0, 30]$ for $h = 2^{-6}$ against $\cos(t)$

Yes the numerical solution closely follows the function $\cos(t)$ even for later times.

Part 2

Use log-log to plot $|u(t) - \cos(t)|$ vs $(\frac{1}{2} + e^{20\cos(1.3t)})$ for $t \in [0, 30]$.

Solution

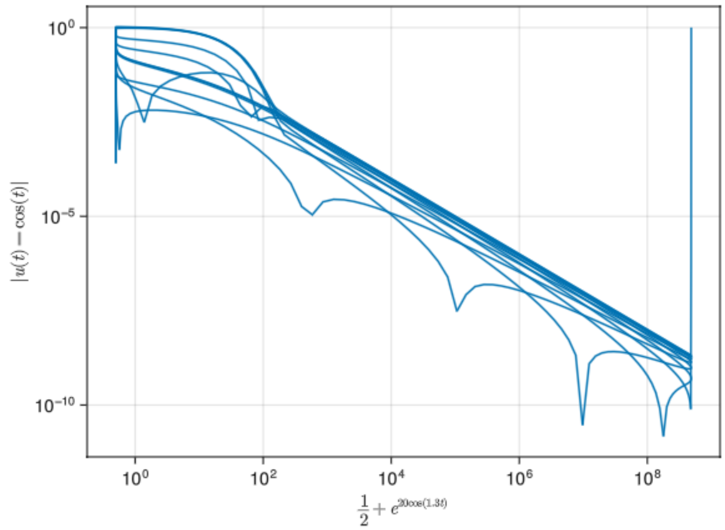


Figure 8: Log-log plot of $|u(t) - \cos(t)|$ vs $(\frac{1}{2} + e^{20\cos(1.3t)})$ with $h = 2^{-6}$ for $t \in [0, 30]$

Problem 5

Implement the backward Euler and the 2s-DIRK method with $\alpha = 1 - \frac{1}{\sqrt{2}}$. Use each of these two methods to solve (114) in $t \in [0, 30]$. Try time steps $h = 2^{-i} \quad \forall i \in [4, 5, \cdots, 9]$.

For each numerical method, carry out error estimation.

Part 1

For each method, plot the |estimated error| vs t for $h = 2^{-4}$. Plot the two curves in ONE figure for comparison. **Use log-scale for errors**.

Solution

Given that the 2-stage DIRK method is 2nd order, we carry out the error estimation using the Richardsom extrapolation,

$$E_n(h) = \frac{1}{1 - (0.5)^2} \left(u_n(h) - u_{2n}(h/2)\right) \tag{115}$$

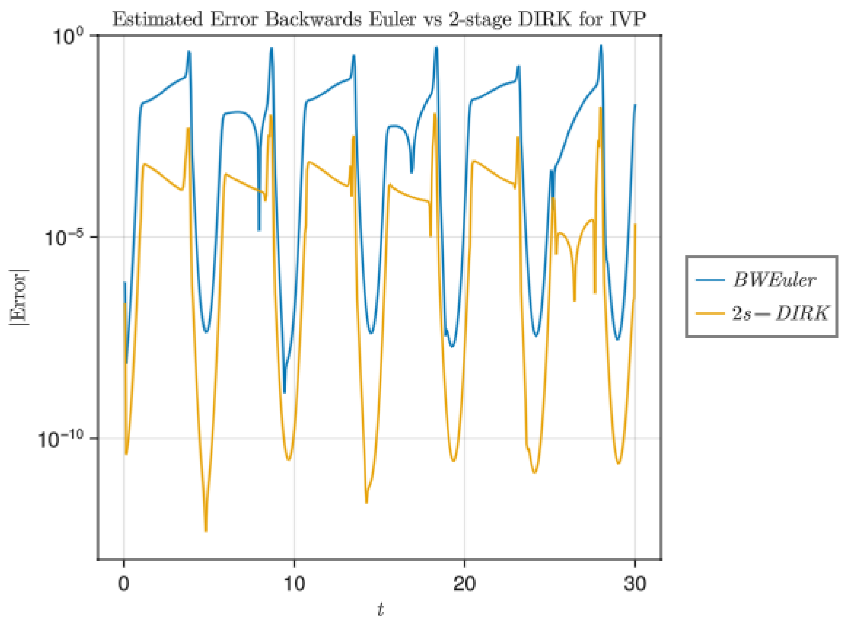


Figure 9: Plot of the estimated error for the numerical solution of IVP (114) using the backward euler method (blue) and the 2-stage DIRK method (yellow) for $h = 2^{-4}$ over time.

Part 2

In a seperate figure, plot the two curves of estimated errors vs t for $h = 2^{-8}$.

Solution

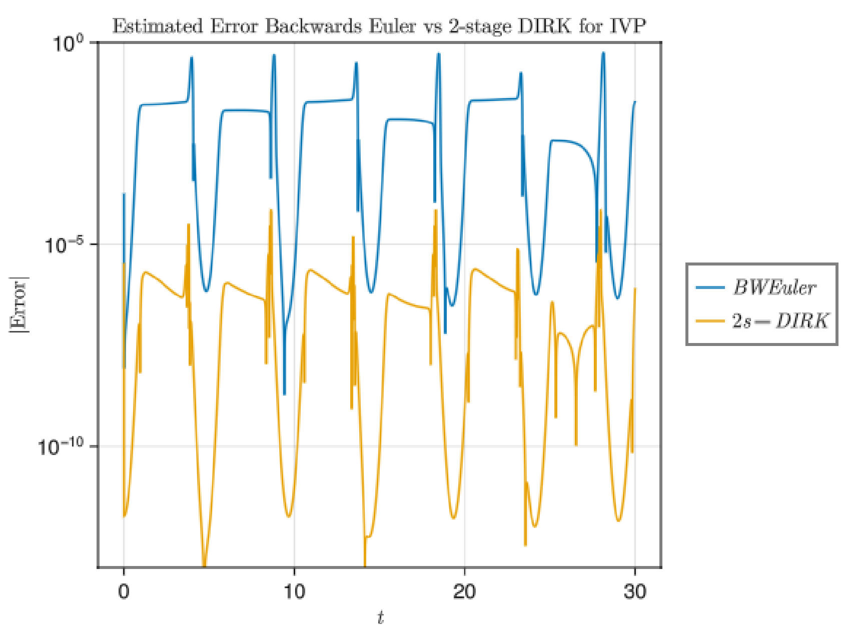


Figure 10: Plot of the estimated error for the numerical solution of IVP (114) using the backward euler method (blue) and the 2-stage DIRK method (yellow) for $h = 2^{-8}$ over time.

Appendix: Source Code

Problem 1 Part 2

```
using GLMakie
function p1p2(method::Symbol)
    local R
    if isequal(method, :RK4)
        R = z -> 1 + z + 0.5*z^2 + (1.0/6.0)*z^3 + (1.0/24.0)*z^4
    elseif isequal(method, :Heun)
        R = z -> 1 + z + 0.5*z^2
    else
        println("wrong symbol use :RK4 or :Heun")
        return nothing
    end
    res = 1000
    xs = [i for i in LinRange(-3, 3, res)]
    ys = [i for i in LinRange(-3, 3, res)]
    S = zeros(res, res)
    for idx in CartesianIndices(S)
        i, j = idx.I
        S[idx] = abs(R(complex(xs[i], ys[j])))
    end
    fig = Figure()
    ax = Axis(
        fig[1, 1],
        title = "Stability Region for $(method) method",
        xlabel = "Re(z)",
        ylabel = "Im(z)",
        aspect = DataAspect()
    )
    contour!(ax, xs, ys, S, levels = [1.0], color = :blue, linewidth = 3)
    contourf!(ax, xs, ys, S, levels = [0, 1], colormap = [:lightblue, :white])
    save("$(method)_stability.png", fig)
end
```

Problem 2 Part 3

```
using GLMakie
function p2p3()
    local R(z, α) = (1 + z*(1 - 2*α))/((1 - α*z)^2)
    α::Float64 = 0.29289321881345254
    res::Int = 1000
    xs = [i for i in LinRange(-5, 15, res)]
    ys = [i for i in LinRange(-10, 10, res)]
    S = zeros(res, res)
    for idx in CartesianIndices(S)
        i, j = idx.I
        S[idx] = abs(R(complex(xs[i], ys[j]), α))
    end
    fig = Figure()
    ax = Axis(fig[1, 1],
        title = "Stability Region for 2s-DIRK method",
        xlabel = "Re(z)",
```

```
        ylabel = "Im(z)",
        aspect = DataAspect()
    )
    contour!(ax, xs, ys, S, levels = [1.0], color = :blue, linewidth = 3)
    contourf!(ax, xs, ys, S, levels = [0, 1], colormap = [:lightblue, :white])
    elements = [PolyElement(color = :lightblue), LineElement(color = :blue, linewidth = 3)]
    labels = [L"Stable: $|R(z)| \leq 1$", L"Boundary: $|R(z)| = 1$"]
    Legend(fig[1, 2], elements, labels, "Stability", halign = :left)
    save("2s-DIRK_stability.png", fig)
end
```

Problem 3 Part 3

```
using GLMakie
using PolynomialRoots
using LinearAlgebra

# LLM1
## characteristic polynomials
ρ_1(ξ) = ξ^2.0 - 1.0
σ_1(ξ) = (1.0/3.0)*(ξ^2.0 + 4.0*ξ + 1.0)
## stability polynomial
Π_1(z) = [-z/3.0 - 1.0, -4.0*z/3.0, 1.0 - z/3.0]
## parametrized curve
z_1(θ) = ρ_1(exp(im*θ)) / σ_1(exp(im*θ))

# BDF1
## characteristic polynomials
ρ_2(ξ) = ξ^2.0 - (4.0/3.0)*ξ + (1.0/3.0)
σ_2(ξ) = (2.0/3.0)*ξ^2
## stability polynomial
Π_2(z) = [1.0/3.0, -4.0/3.0, 1.0 - 2.0*z/3.0]
## parameterized curve
z_2(θ) = ρ_2(exp(im*θ)) / σ_2(exp(im*θ))

# function to plot stability region
function make_plot(z::Function, Π::Function, method::AbstractString)
    lims = ((-1, 4.5), (-3, 3))
    res = 1000
    xs = collect(LinRange(lims[1][1], lims[1][2], res))
    ys = collect(LinRange(lims[2][1], lims[2][2], res))
    S = zeros(res, res)
    for idx in CartesianIndices(S)
        i, j = idx.I
        k = complex(xs[i], ys[j])
        S[idx] = maximum(abs.(roots(Π(k))))
    end
    fig = Figure()
    ax = Axis(
        fig[1, 1],
        xlabel = L"\text{Re}(z)",
        ylabel = L"\text{Im}(z)",
        aspect = DataAspect(),
```

```
        limits = lims
    )
    ax.title = "Stability region for the $(method) method"
    # plot the boundary locus
    θ = LinRange(0, 2π, 1000)
    z_vals = z.(θ)
    z_real = real.(z_vals)
    z_imag = imag.(z_vals)
    contourf!(ax, xs, ys, S, levels = [0, 1], colormap = [:lightblue, :white])
    lines!(ax, z_real, z_imag, color = :red, linestyle = :dash)
    elements = [PolyElement(color = :white), PolyElement(color = :lightblue), LineElement(color = :red, linestyle =
    labels = [L"\text{Unstable}", L"\text{Stable}", L"\text{Boundary locus}"]
    Legend(fig[1, 2], elements, labels, halign = :left)
    save("h3prob3part3$(method).png", fig)
end
```

Problem 4

```
module problem4
using GLMakie
using LinearAlgebra
export DIRK2, dudt, newton
function newton(f::Function; x_init::Float64 = 1.0, maxiter = 100, h = 1e-8)
    x_current = x_init
    x_next = x_current
    y = f(x_current)
    Δx = Inf
    k = 1
    while (abs(Δx) > 10*eps()) && (abs(y) > 10*eps()) && (k < maxiter)
        df = (f(x_current + h) - f(x_current - h)) / (2*h)
        Δx = -y/df
        x_next = x_current + Δx
        k += 1
        y = f(x_next)
        x_current = x_next
    end
    x_current
end
function DIRK2(f::Function, u0::Float64, tspan::Tuple, h::Float64;
    α = 1 - (1/sqrt(2)))
    p = 2
    A = [α 0; 1-α α]; b = [1-α, α]; c = [α, 1]
    t0, tf = tspan
    N = Int(floor((tf - t0)/h))
    t = Vector{Float64}(undef, N+1)
    u = Vector{Float64}(undef, N+1)
    k = zeros(p, 1)
    u[1] = u0
    t[1] = t0
    for n = 1:N
        for i = 1:p
            G = x -> h*f(u[n]+dot(A[i, 1:i-1],k[1:i-1])+A[i, i]*x,t[n]+c[i]*h)-x
            k[i] = newton(G, x_init = 0.0)
```

```
        end
        u[n + 1] = u[n] + dot(b, k)
        t[n + 1] = t[n] + h
    end
    u, t
end
dudt(u, t) = -(0.5 + exp(20*cos(1.3*t)))*sinh(u - cos(t))
function part1()
    fig = Figure()
    ax = Axis(fig[1, 1],
        title = L"Approximate solution against $\cos(t)$",
        xlabel = L"$t$", ylabel = L"f(t)")
    u, t = DIRK2(dudt, 0.0, (0.0, 30.0), 2^(-6))
    lines!(ax, t, cos.(t), color = :red, linestyle = :dash, label = L"$\cos{(t)}$")
    lines!(ax, t, u, color = :blue, label = L"$u(t)$")
    Legend(fig[1, 2], ax)
    fig
end
function part2()
    fig = Figure()
    ax = Axis(
        fig[1, 1],
        xlabel = L"$\frac{1}{2} + e^{20\cos(1.3t)}$",
        ylabel = L"$|u(t) - \cos(t)|$",
        xscale = log10,
        yscale = log10
    )
    u, t = DIRK2(dudt, 0.0, (0.0, 30.0), 2^(-6))
    x = cos.(t)
    k = abs.(u .- x)
    g = t -> 0.5 + exp(20*cos(1.3*t))
    z = g.(t)
    lines!(ax, z, k)
    fig
end
end
```

Problem 5

```
if !@isdefined problem4
    include("hw3problem4.jl")
    using .problem4
end
using GLMakie
using LinearAlgebra

function backwardEuler(f::Function, u0::Float64, tspan::Tuple, h::Float64)
    t0, tf = tspan
    N = Int(floor((tf - t0)/h))
    t = Vector{Float64}(undef, N+1)
    u = Vector{Float64}(undef, N+1)
    u[1] = u0
    t[1] = t0
```

```
    for n = 1:N
        G = k -> h*f(u[n] + h*k, t[n] + h) - k
        k = newton(G, x_init = 0.0)
        u[n + 1] = u[n] + h*k
        t[n + 1] = t[n] + h
    end
    return u, t
end

function error(u::Vector{Float64}, u2::Vector{Float64})
    a = 1/(1 - 0.5^2)
    E = Vector{Float64}(undef, length(u))
    for n in eachindex(E)
        E[n] = norm(a*(u[n] - u2[2*n-1]))
    end
    E
end

function part12(h::Float64)
    fig = Figure()
    ax = Axis(fig[1, 1],
        title = L"Estimated Error Backwards Euler vs $2$-stage DIRK for IVP",
        xlabel = L"t",
        ylabel = L"|\text{Error}|",
        yscale = log10,
        limits = (nothing, (1e-13, 1))
    )
    u1, t1 = backwardEuler(dudt, 0.0, (0.0, 30.0), h)
    u2, t2 = backwardEuler(dudt, 0.0, (0.0, 30.0), (h/2))
    e = error(u1, u2)
    lines!(ax, t1, e, label = L"BW Euler")
    u1, t1 = DIRK2(dudt, 0.0, (0.0, 30.0), h)
    u2, t2 = DIRK2(dudt, 0.0, (0.0, 30.0), (h/2))
    e = error(u1, u2)
    lines!(ax, t1, e, label = L"2s-DIRK")
    Legend(fig[1, 2], ax)
    fig
end
```

References

"Linear Multistep Methods." 2003. In *Numerical Methods for Ordinary Differential Equations*, 301–55. John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/0470868279.ch4.