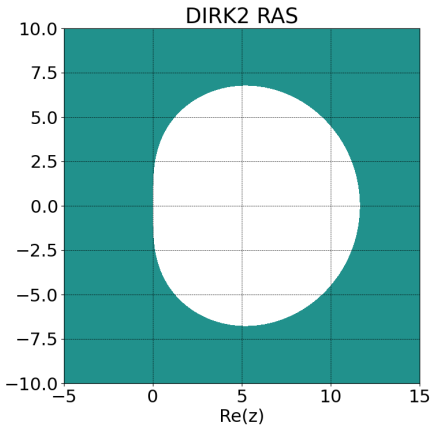
Scientific Computing HW 3

Ryan Chen

February 22, 2024

1. From the Butcher array,
$$A = \begin{bmatrix} \gamma & 0 \\ 1-\gamma & \gamma \end{bmatrix}, \quad b = \begin{bmatrix} 1-\gamma \\ \gamma \end{bmatrix}, \quad c = \begin{bmatrix} \gamma \\ 1 \end{bmatrix}$$
Check the 1st order accuracy condition.
$$\sum_{l=1}^{2} b_l = (1-\gamma) + \gamma = 1$$
Check the 2nd order accuracy condition.
$$\sum_{l=1}^{2} b_l c_l = (1-\gamma)\gamma + \gamma \cdot 1 = \gamma - \gamma^2 + \gamma = 2\gamma - \gamma^2 = 2 - 2^{1/2} - 1 - 2^{-1} + 2^{1/2} = 1 - 2^{-1} = \frac{1}{2}$$
Thus the method is 2nd order accurate. To show it is A–stable, first find the stability function $R(z)$ and let $|z| \to \infty$.
$$I - zA = \begin{bmatrix} 1-\gamma z & 0 \\ -(1-\gamma)z & 1-\gamma z \end{bmatrix} \implies D := \det(I - zA) = (1-\gamma z)^2 = \gamma^2 z^2 - 2\gamma z + 1$$
$$\implies (I - zA)^{-1} = \frac{1}{D}\begin{bmatrix} 1-\gamma z & 0 \\ (1-\gamma)z & 1-\gamma z \end{bmatrix} \implies (I - zA)^{-1} 1_{s\times 1} = \frac{1}{D}\begin{bmatrix} 1-\gamma z \\ (1-\gamma)z + 1 - \gamma z \end{bmatrix} = \frac{1}{D}\begin{bmatrix} 1-\gamma z \\ (1-2\gamma)z + 1 \end{bmatrix}$$
$$R(z) - 1 = z b^T (I - zA)^{-1} 1_{s\times 1} = \frac{z}{D}\left[(1-\gamma)(1-\gamma z) + \gamma((1-2\gamma)z + 1)\right] = \frac{z}{D}\left[1 - \gamma z - \gamma + \gamma^2 z + (\gamma - 2\gamma^2)z + \gamma\right]$$
$$\implies R(z) - 1 = \frac{z}{D}\left[1 - \gamma^2 z\right] = \frac{-\gamma^2 z^2 + z}{\gamma^2 z^2 - 2\gamma z + 1} \implies R(z) = \frac{-\gamma^2 z^2 + z}{\gamma^2 z^2 - 2\gamma z + 1} + 1 \xrightarrow{|z|\to\infty} -1 + 1 = 0$$
To finish showing A–stability, we plot the RAS and see that it contains the left half plane. Code in 2nd cell of:

`https://github.com/RokettoJanpu/scientific-computing-2-redux/blob/main/hw3%20RAS.ipynb`



2. From the Butcher array,

$$A = \begin{bmatrix} \gamma & 0 \\ 1-2\gamma & \gamma \end{bmatrix}, \quad b = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}, \quad c = \begin{bmatrix} \gamma \\ 1-\gamma \end{bmatrix}$$

(a) **Pf.** Check the 1st order accuracy condition.

$$\sum_{l=1}^{2} b_l = \frac{1}{2} + \frac{1}{2} = 1$$

Check the 2nd order accuracy condition.

$$\sum_{l=1}^{2} b_l c_l = \frac{1}{2}\gamma + \frac{1}{2}(1-\gamma) = \frac{1}{2}(\gamma + 1 - \gamma) = \frac{1}{2}$$

(b) **Pf.** Check the 3rd order accuracy conditions.

$$\sum_{p,q,r} b_p a_{pq} a_{pr} = \frac{1}{2}[\gamma^2 + 2\gamma \cdot 0 + 0^2] + \frac{1}{2}[(1-2\gamma)^2 + 2(1-2\gamma)\gamma + \gamma^2]$$

The quantity in the second bracket is

$$(1-2\gamma)^2 + 2(1-2\gamma)\gamma + \gamma^2 = 1 + 4\gamma^2 - 4\gamma + 2\gamma - 4\gamma^2 + \gamma^2 = \gamma^2 - 2\gamma + 1 = (\gamma - 1)^2$$

giving

$$\sum_{p,q,r} b_p a_{pq} a_{pr} = \frac{1}{2}[\gamma^2 + \gamma^2 - 2\gamma + 1] = \gamma^2 - \gamma + \frac{1}{2}$$

We find

$$\gamma^2 = \frac{1}{2} + \frac{3}{36} + 2\frac{3^{1/2}}{12} = \frac{1}{12}[3 + 1 + 2 \cdot 3^{1/2}] = \frac{1}{12}[4 + 2 \cdot 3^{1/2}] = \frac{1}{6}[2 + 3^{1/2}]$$

so finally,

$$\sum_{p,q,r} b_p a_{pq} a_{pr} = \frac{1}{6}[2 + 3^{1/2} - 3 - 3^{1/2} + 3] = \frac{1}{3}$$

(c) First find the stability function $R(z)$.

$$I - zA = \begin{bmatrix} 1-\gamma z & 0 \\ -(1-2\gamma)z & 1-\gamma z \end{bmatrix} \implies D := \det(I-zA) = (1-\gamma z)^2 = \gamma^2 z^2 - 2\gamma z + 1$$

$$\implies (I-zA)^{-1} = \frac{1}{D}\begin{bmatrix} 1-\gamma z & 0 \\ (1-2\gamma)z & 1-\gamma z \end{bmatrix} \implies (I-zA)^{-1}1_{s\times 1} = \frac{1}{D}\begin{bmatrix} 1-\gamma z \\ (1-2\gamma)z + 1 - \gamma z \end{bmatrix} = \frac{1}{D}\begin{bmatrix} 1-\gamma z \\ (1-3\gamma)z + 1 \end{bmatrix}$$

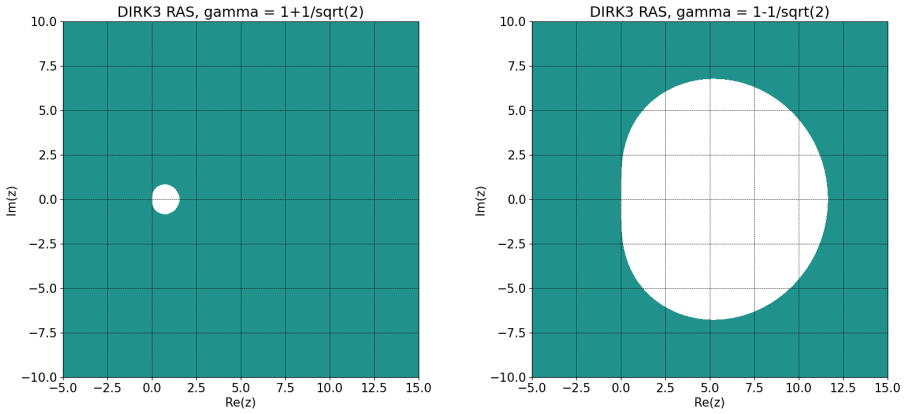
$$R(z) - 1 = zb^T(I-zA)^{-1}1_{s\times 1} = \frac{z}{2D}\left[1-\gamma z + (1-3\gamma)z + 1\right] = \frac{z}{2D}\left[(1-4\gamma)z + 2\right] = \frac{1}{2}\frac{(1-4\gamma)z^2 + 2z}{\gamma^2 z^2 - 2\gamma z + 1}$$

We find γ by imposing $\lim_{|z|\to\infty} R(z) = 0$.

$$\lim_{|z|\to\infty} R(z) = 0 \iff -1 = \frac{1}{2}\lim_{|z|\to\infty}\frac{(1-4\gamma)z^2+2z}{\gamma^2 z^2 - 2\gamma z + 1} \iff \lim_{|z|\to\infty}\frac{(1-4\gamma)z^2+2z}{\gamma^2 z^2 - 2\gamma z + 1} = -2 \iff \frac{1-4\gamma}{\gamma^2} = -2$$

$$\iff -2\gamma^2 = 1 - 4\gamma \iff 2\gamma^2 - 4\gamma + 1 = 0 \iff \gamma = \frac{4}{4} \pm \frac{(16-8)^{1/2}}{4} = 1 \pm \frac{2\cdot 2^{1/2}}{4} = 1 \pm 2^{-1/2}$$

We check that the method for $\gamma = 1 \pm 2^{-1/2}$ is A–stable, hence L–stable, by plotting the RASes and seeing that they contain the left half plane. Code in 3rd cell of:
`https://github.com/RokettoJanpu/scientific-computing-2-redux/blob/main/hw3%20RAS.ipynb`



3. Code for all parts of this problem:
`https://github.com/RokettoJanpu/scientific-computing-2-redux/blob/main/hw3.ipynb`

(a) Set $f(t,y) := -L(y-\phi(t)) + \phi'(t)$. To implement DIRK2, we first obtain explicit formulas for k_1, k_2, u_{n+1}.

$$k_1 = f(t_n + \gamma h, u_n + h\gamma k_1) = -L\left[u_n + h\gamma k_1 - \phi(t_n + \gamma h)\right] + \phi'(t_n + \gamma h)$$

$$\implies k_1 = -Lh\gamma k_1 - L\left[u_n - \phi(t_n + \gamma h)\right] + \phi'(t_n + \gamma h)$$

$$\implies (1+Lh\gamma)k_1 = -L\left[u_n - \phi(t_n+\gamma h)\right] + \phi'(t_n+\gamma h) \implies k_1 = \frac{-L\left[u_n - \phi(t_n+\gamma h)\right] + \phi'(t_n+\gamma h)}{1+Lh\gamma}$$

$$k_2 = f(t_n + h, u_n + h(1-\gamma)k_1 + h\gamma k_2) = -L\left[u_n + h(1-\gamma)k_1 + h\gamma k_2 - \phi(t_n + h)\right] + \phi'(t_n + h)$$

$$\implies k_2 = -Lh\gamma k_2 - L\left[u_n + h(1-\gamma)k_1 - \phi(t_n + h)\right] + \phi'(t_n + h)$$

$$\implies (1 + Lh\gamma)k_2 = -L\left[u_n + h(1-\gamma)k_1 - \phi(t_n + h)\right] + \phi'(t_n + h)$$

$$\implies k_2 = \frac{-L\left[u_n + h(1-\gamma)k_1 - \phi(t_n + h)\right] + \phi'(t_n + h)}{1 + Lh\gamma}$$

$$u_{n+1} = u_n + h\left[(1-\gamma)k_1 + \gamma k_2\right]$$

We do the same for DIRK3 (abuse of language for DIRK of order 3).

$$k_1 = f(t_n + \gamma h, u_n + h\gamma k_1)$$

so the explicit formula for k_1 is the same as in DIRK2.

$$k_1 = \frac{-L\left[u_n - \phi(t_n + \gamma h)\right] + \phi'(t_n + \gamma h)}{1 + Lh\gamma}$$

$$k_2 = f(t_n + (1-\gamma)h, u_n + h(1-2\gamma)k_2 + h\gamma k_2)$$

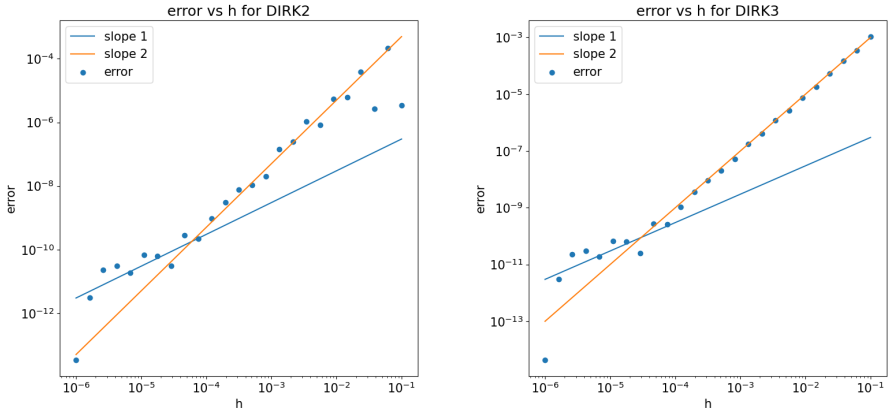
$$= -L\left[u_n + h(1-2\gamma)k_1 + h\gamma k_2 - \phi(t_n + (1-\gamma)h)\right] + \phi'(t_n + (1-\gamma)h)$$

$$= -Lh\gamma k_2 - L\left[u_n + h(1-2\gamma)k_1 - \phi(t_n + (1-\gamma)h)\right] + \phi'(t_n + (1-\gamma)h)$$

$$\implies (1 + Lh\gamma)k_2 = -L\left[u_n + h(1-2\gamma)k_1 - \phi(t_n + (1-\gamma)h)\right] + \phi'(t_n + (1-\gamma)h)$$

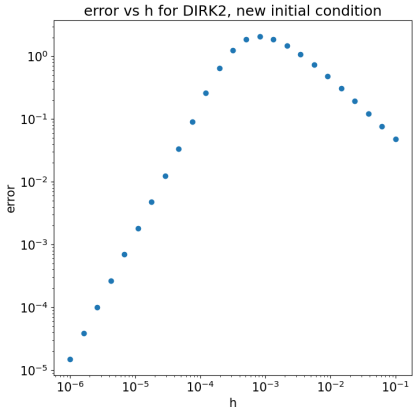
$$\implies k_2 = \frac{-L\left[u_n + h(1-2\gamma)k_1 - \phi(t_n + (1-\gamma)h)\right] + \phi'(t_n + (1-\gamma)h)}{1 + Lh\gamma}$$

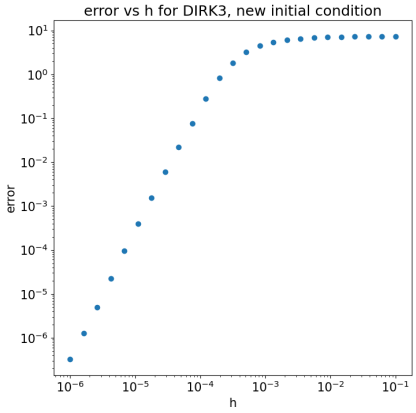
$$u_{n+1} = u_n + \frac{h}{2}[k_1 + k_2]$$



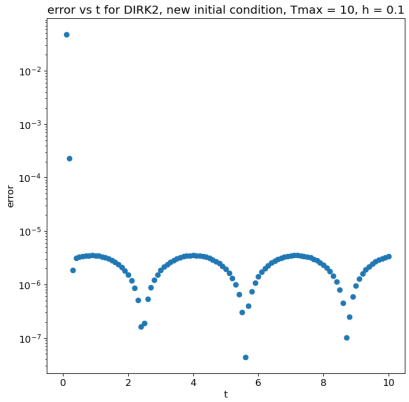
For $h > 10^{-4}$, the log–log graph roughly has a slope of 2, ie the error is on the order of h. For $10^{-5} < h < 10^{-4}$, the log–log graph roughly has a slope of 1, ie the error is on the order of h^2. For $h < 10^{-5}$, the slope is considerably steeper, ie the error is on the order of a high power of h.

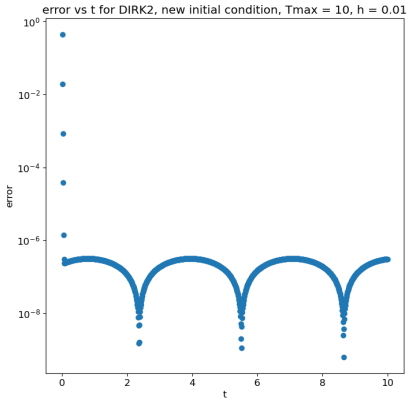
(b) Repeating (a) with $y(0) = \sin\left(\frac{\pi}{4}\right) + 10$, we notably get a region for DIRK2 where error increases as h decreases, and a region for DIRK3 of very small slope.

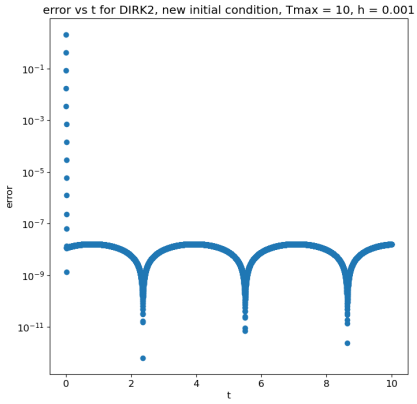


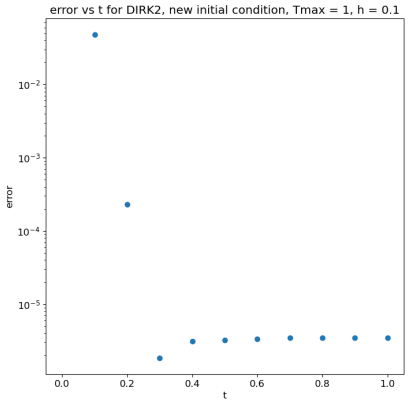


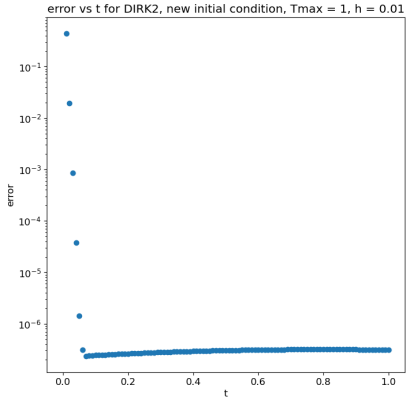
For each method, we plot error vs time graphs for $h = 10^{-1}, 10^{-2}, 10^{-3}$ and $T_{\max} = 10, 1$.

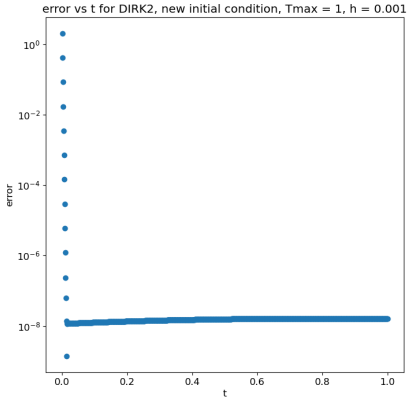


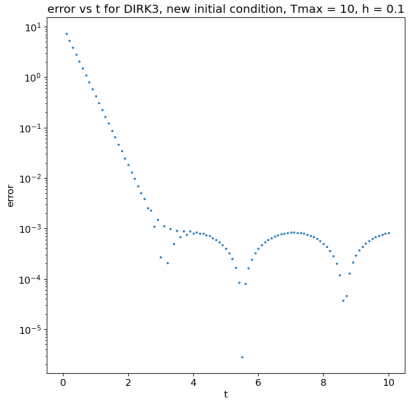


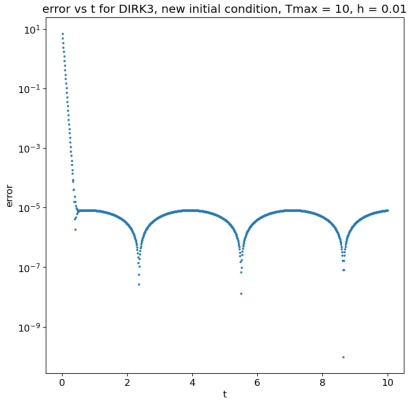


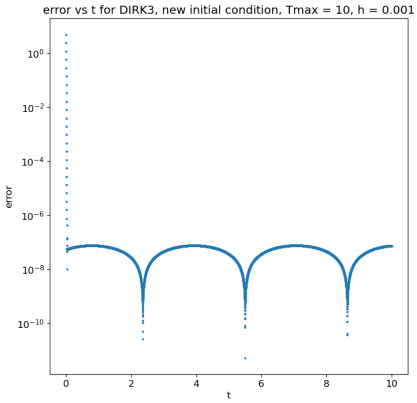


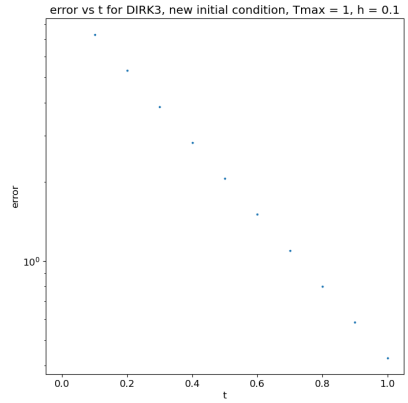

error vs t for DIRK2, new initial condition, Tmax = 1, h = 0.01
error
t

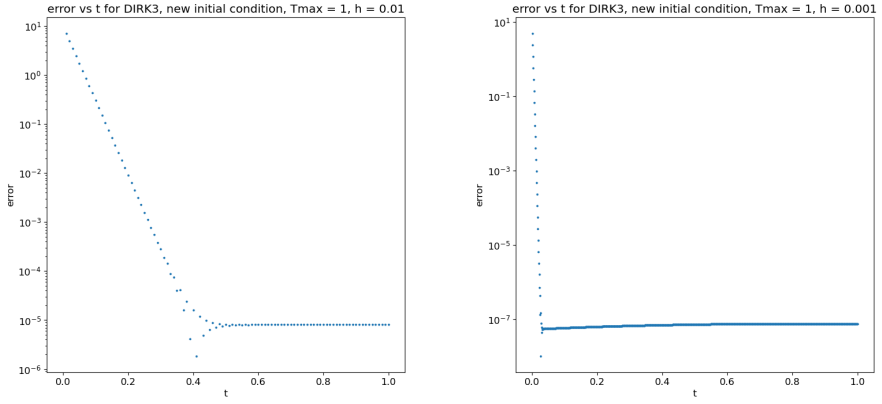

error vs t for DIRK2, new initial condition, Tmax = 1, h = 0.001
error
t


error vs t for DIRK3, new initial condition, Tmax = 10, h = 0.1
error
t


error vs t for DIRK3, new initial condition, Tmax = 10, h = 0.01
error
t


error vs t for DIRK3, new initial condition, Tmax = 10, h = 0.001
error
t


error vs t for DIRK3, new initial condition, Tmax = 1, h = 0.1
error
t



(c) When solving the problem with the first initial condition, different orders of convergence were observed in each method. For DIRK3, note that it is 3rd order accurate, yet the graph for its error shows a regime in which error decayed as h, an instance of order reduction. When the initial condition is far from the forced response $\phi(t)$, the issue is worse. DIRK2 has a regime where error actually increases as h gets smaller, and DIRK3 has a regime of error decaying as a very small power of h in spite of its 3rd order accuracy. The error likely comes from the fact that we are using RK methods to solve a stiff problem. To try to minimize the effect of order reduction, we may try higher order stage methods.