KTH Royal Institute of Technology

SF2520

Applied Numerical Methods

Computer Exercise 1

David Ahnlund
20000531-4778

dahnlund@kth.se

Emil Gestsson
20021030-3756

gestsson@kth.se

September 18, 2023

Part 1

In Part 1, we are asked to study the accuracy and stability of an explicit Runge Kutta 3 method. More specific, the following scheme:

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + h, y_n + hk_1) \\
k_3 &= f(t_n + h/2, y_n + \frac{h}{4}k_1 + \frac{h}{4}k_2) \\
y_{n+1} &= y_n + \frac{h}{6}(k_1 + k_2 + 4k_3)
\end{aligned}
$$

The given ODE was a linearized Landau-Lifshitz equation for $\boldsymbol{m}(t) \in \mathbb{R}^3$ as

$$
\frac{d\boldsymbol{m}}{dt} = \boldsymbol{a} \times \boldsymbol{m} + \alpha \boldsymbol{a} \times (\boldsymbol{a} \times \boldsymbol{m}), \qquad \boldsymbol{m}(0) = (0; 0; 1) \tag{1}
$$

where $\boldsymbol{a} = \frac{1}{4}[1; \sqrt{11}; 2]$

a)

When numerically solving eq. 1 using RK3 the following solution was obtained:

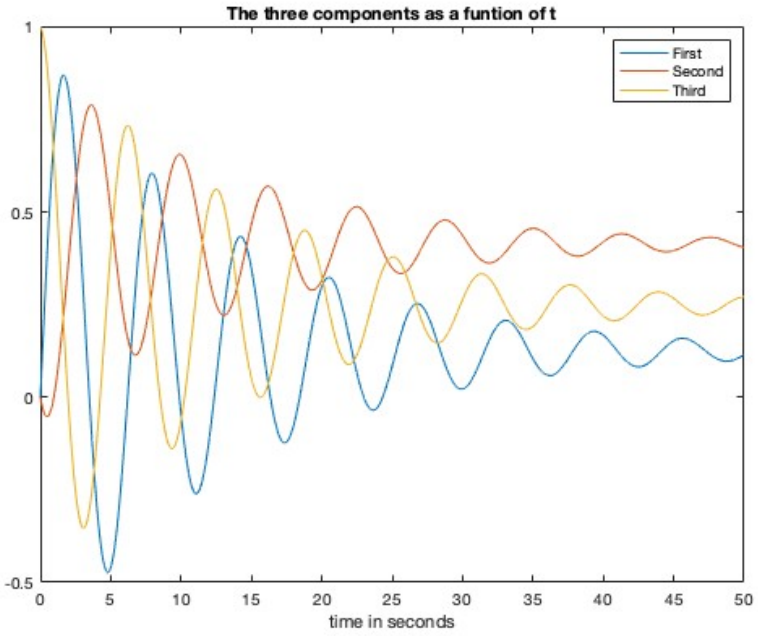


Figure 1: Plot of the three components in $\boldsymbol{m}(t)$

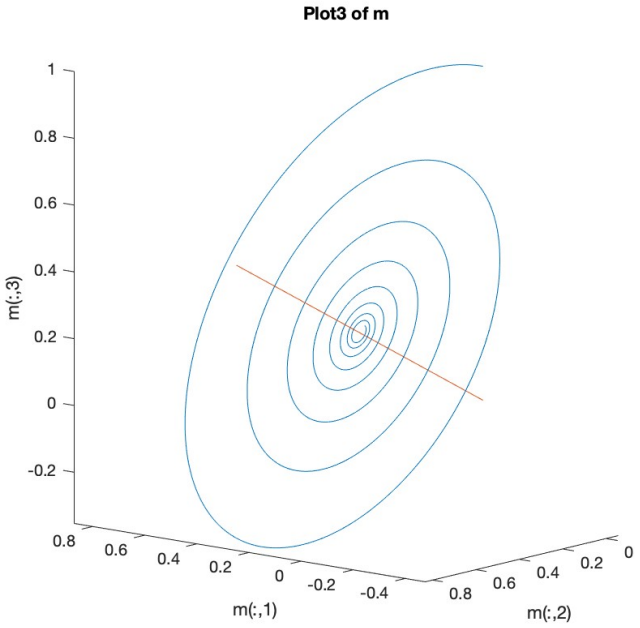


Figure 2: Plot3 of $\boldsymbol{m}(t)$ together with $\boldsymbol{a}$

As seen in Figure 2, the vector $\boldsymbol{a}$ is perpendicular to the plane of $\boldsymbol{m}(t)$.

b)

For the analysis of the order of the RK3 scheme, the difference, $||\tilde{\boldsymbol{m}}_N(T) - \tilde{\boldsymbol{m}}_{2N}(T)||_2$ when doubling the number of time steps was done for the following N:s:

$$\boldsymbol{N} = [50, 100, 200, 400, 800, 1600, 3200, 6400]$$

When plotting these differences as a function of h in a log-log-plot, together with the function h^3, the two graphs were parallel indicating that the order of the scheme is 3, as expected. See Figure 3

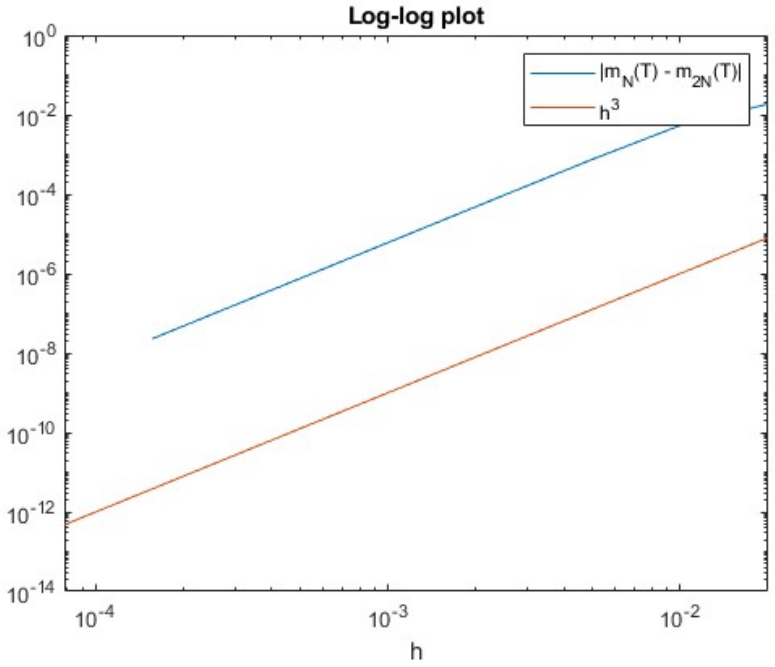


Figure 3

Moreover, the order of accuracy was confirmed by the slope of $E(N) = ||\tilde{\boldsymbol{m}}_N(T) - \tilde{\boldsymbol{m}}_{2N}(T)||_2$. Computing the differences of $\log_2 E(N)$ gave the following values:

$(N, 2N)$	(50,100)	(100,200)	(200,400)	(400,800)	(800,1600)	(1600,3200)
$\log_2$-difference	1.7986	2.8360	2.9873	3.0025	3.0023	3.0013

c)

The ODE was rewritten to standard form by expressing the cross product $\boldsymbol{a} \times \boldsymbol{b}$ as a matrix multiplication $\boldsymbol{A}_c\boldsymbol{b}$, where

$$\boldsymbol{A}_c = \begin{bmatrix} 0 & -\boldsymbol{a}_3 & \boldsymbol{a}_2 \\ \boldsymbol{a}_3 & 0 & -\boldsymbol{a}_1 \\ -\boldsymbol{a}_2 & \boldsymbol{a}_1 & 0 \end{bmatrix}$$

inserting this into the original expression yields

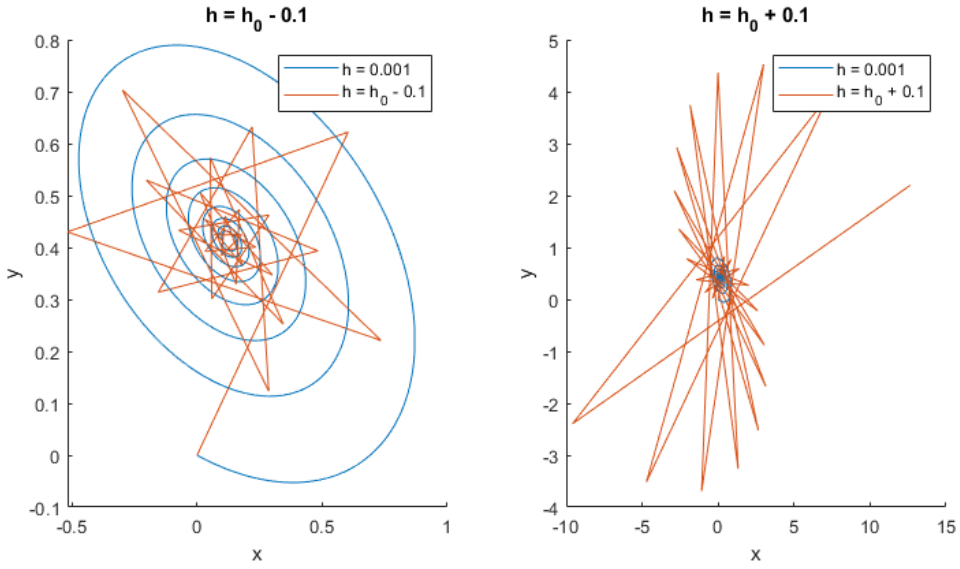
$$\frac{d\boldsymbol{m}}{dt} = \left(\boldsymbol{A}_c + \alpha \boldsymbol{A}_c^2\right) \boldsymbol{m} = \boldsymbol{A}\boldsymbol{m}$$

For finding the stable region, we use $s(z) = |1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3|$, then for each eigenvalue λ of $\boldsymbol{A}_c + \alpha \boldsymbol{A}_c^2$ we find a $h > 0$ such that $s(h\lambda) = 1$ using fsolve. h_0 is then set as the smallest h produced by the eigenvalues, which gave the result $h_0 = 2.055596$.

Note that this approach is only possible because the stable region is convex, as a) only one root exists along the complex line produced by each eigenvalue, and b) the method remains within the stable region even if h is reduced, which means taking the minimum h produced by all eigenvalues still satisfies the stability critera for each eigenvalue.

d)

Plotting results with $h = h_0 - 0.1$ and $h = h_0 + 0.1$ in comparison to a more precise solution, we see that the case where $h = h_0 + 0.1$ diverge drastically - whereas the result for $h = h_0 - 0.1$ is imprecise but but still within the same magnitude as the desired solution.



Part 2

a)

The second order system was rewritten as a first order system by adding the time derivatives of r as additional variables. Using AB4, we got the following result:

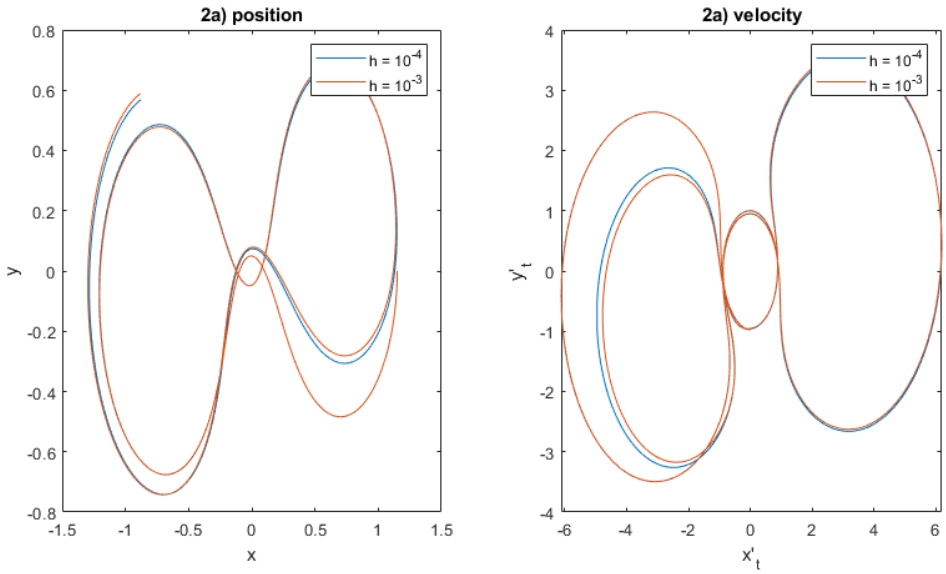


Figure 4: Three body problem solved using AB4, ending at time T = 10. Compares 2 step sizes $h = 10^{-4}$ and $h = 10^{-3}$

b)

N was found using the bisection method, finding the smallest N for which the error was greater than 0.1, assuming the error is a monotonous function of N. An upper limit of 10^6 steps was used.

method	T = 5	T = 20	T = 40
Explicit Euler	188132	exceeded step limit	exceeded step limit
Runge–Kutta 3	698	2002	16175
Adams–Bashforth 4	595	1237	4387

c)

When numerically solving $\tilde{\boldsymbol{r}}$ using Matlab's ode23, i.e. an adaptive method comparing RK3 with RK2, with RelTol $= 10^{-4}$ the number of time steps was $N = 974$. Compared with AB4 for $T = 20$, the number of time steps to achieve an $||\cdot||_2$-error ≤ 0.1, was $N = 24739$, more than 25 times as many time steps.

Comparing the minimun and maximum time steps respectively for ode23, the following was obtained:

$$\text{Minimum time step: } h \approx 0.000161$$
$$\text{Maximum time step: } h \approx 0.1020$$

The varying step size, h, is plotted as a function of t in Figure 5. In Figure 6 the components of $\tilde{\boldsymbol{r}}_{\text{ode23}}$ is shown.

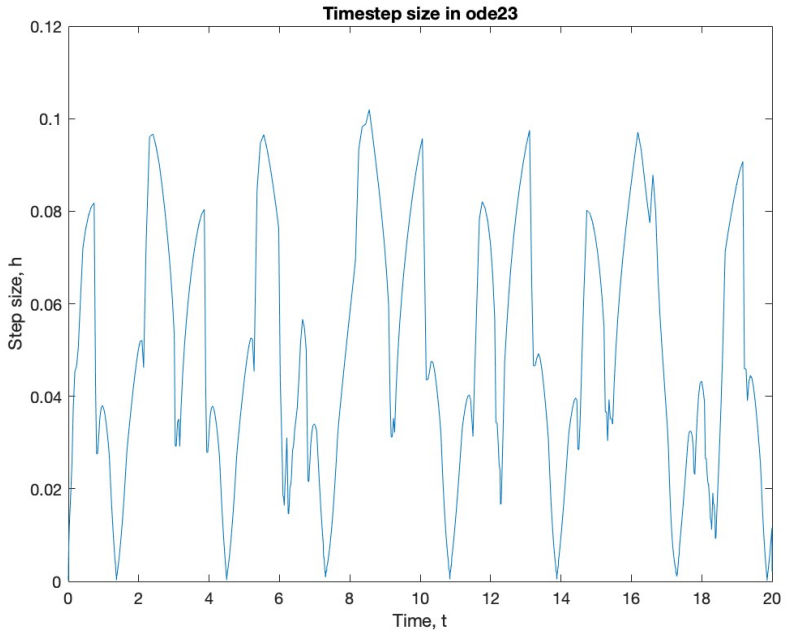


Figure 5: $h(t)$ using ode23

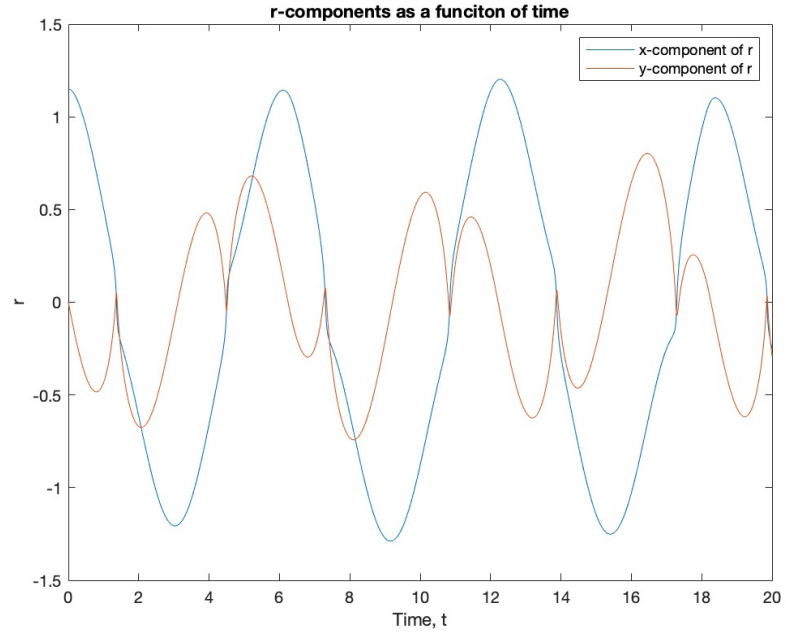


Figure 6: $\tilde{r}$-components as a function of time

When comparing Figure 5 and 6, one can notice that when the slope of the components of $\tilde{\boldsymbol{r}}$,

changes rapidly, that's also where the time steps are the smallest (e.g $t \approx 4.5$).

Part 3

The stiff system of ODE:s in this problem were given as the following:

$$\begin{aligned}\frac{dx_A}{dt} &= -r_1 x_A + r_2 x_B x_C \\ \frac{dx_B}{dt} &= r_1 x_A - r_2 x_B x_C - r_3 x_B^2 \\ \frac{dx_C}{dt} &= r_3 x_B^2\end{aligned}$$

with given constants r_i and initial values as $\vec{x}_0 = [1; 0; 0]$

a)

When finding the largest allowed step size $h = \frac{T}{N}$ for $t \in [0, 10]$ the procedure was done by, for an interval of h values, test if a small perturbation of $\vec{x}_0$, defined as $\delta = 0.01$, acting on the first element of $\vec{x}$ increased the relative error significantly during the scheme, $e_{Rel} = \frac{|\vec{x}_{0:T} - \tilde{x}_{0:T}|}{\vec{x}_{0:T}}$. Here the condition was defined to be that the solution was unstable if the maximum relative error surpassed $\delta \times 10$

When letting h vary in steps of 10^{-6} the $h_{\max}$ was found to be $h_{\max} = 7.52 \cdot 10^{-4}$ and was unstable for $h = 7.53 \cdot 10^{-4}$. This is shown in Figure 7

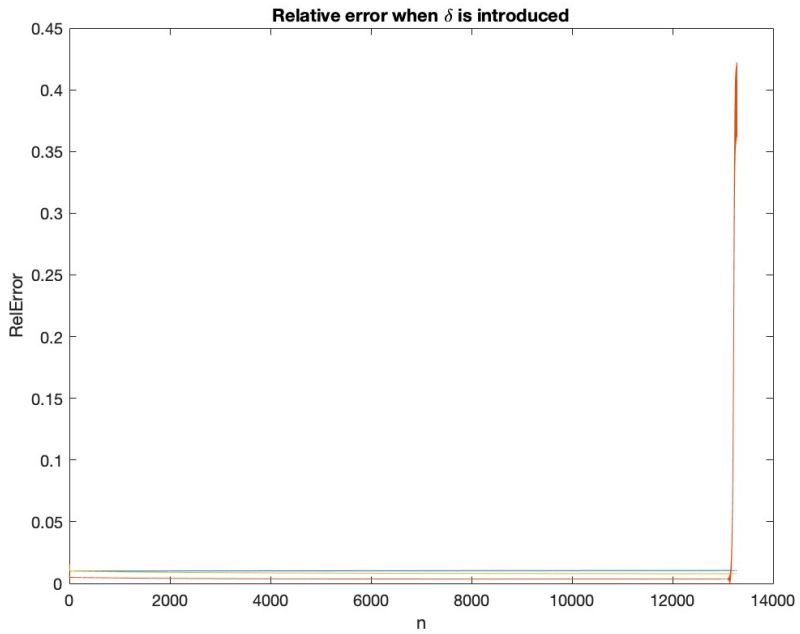


Figure 7: Relative error when $h = 7.53 \cdot 10^{-4}$ (unstable)

When setting h to be the empirically determined limit, the following solution for $\vec{\boldsymbol{x}}$ is shown in Figure 8 and 9.

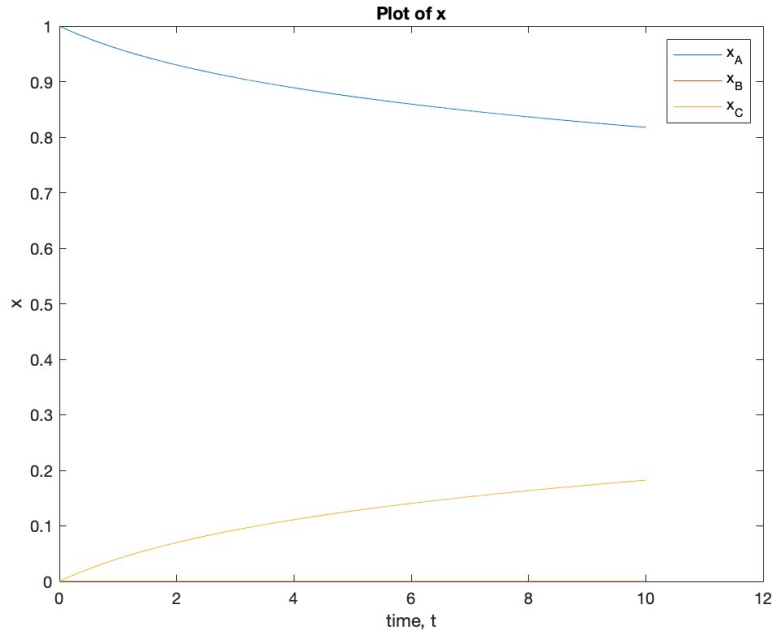


Figure 8: $\vec{x}(t)$, Linear Scale

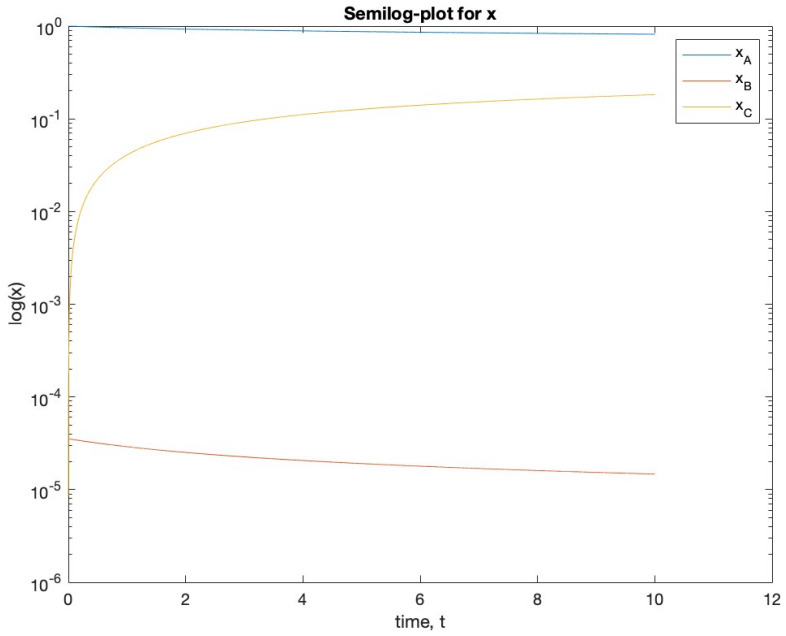


Figure 9: $\vec{x}(t)$, Semi-log Scale

b)

The Jacobian of the equation system given above is as follows:

$$\boldsymbol{J}(x_A, x_B, x_C) = \begin{bmatrix} -r_1 & r_2 x_C & r_2 x_B \\ r_1 & -r_2 x_C - 2r_3 x_B & -r_2 x_B \\ 0 & 2r_3 x_B & 0 \end{bmatrix} \tag{2}$$

In Figure 10 and 11 the two non-zero eigenvalues of $\boldsymbol{J}$ is plotted as a function of time. One can notice that, judging by the plots, the magnitude of the eigenvalue 1 get worse as t increases.

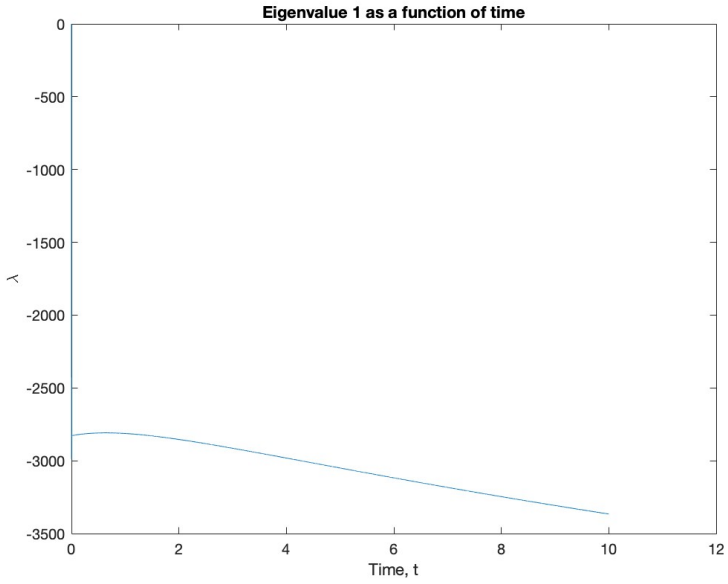


Figure 10: Eigenvalue 1

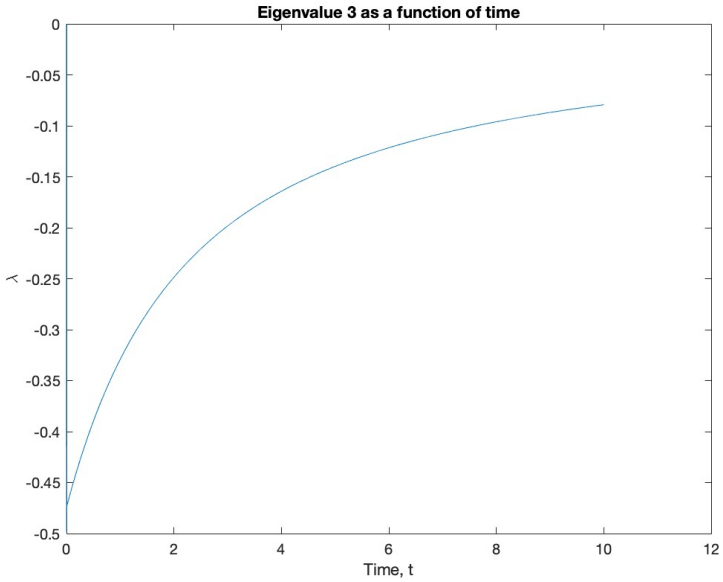


Figure 11: Eigenvalue 3

As given in the exercise description, the stability condition for RK3 in linear ODE:s is

$$|1 + z + \frac{x^2}{2} + \frac{z^3}{6}| \leq 1 \tag{3}$$

but in our case, all eigenvalues were real, $\lambda_k \in \mathbb{R}$ (see Figure 10 and 11), and the one with greatest magnitude were ≤ 0, $\forall t_n$ (Figure 10). This simplified the condition to the following:

$$2 + z + \frac{x^2}{2} + \frac{z^3}{6} \geq 0 \tag{4}$$

This could then be solved via Matlab's fzero function, which yield a $h_{\max} = 7.47 \cdot 10^{-4}$, surprisingly close to the empirically determined value ($h = 7.52 \cdot 10^{-4}$).

c)

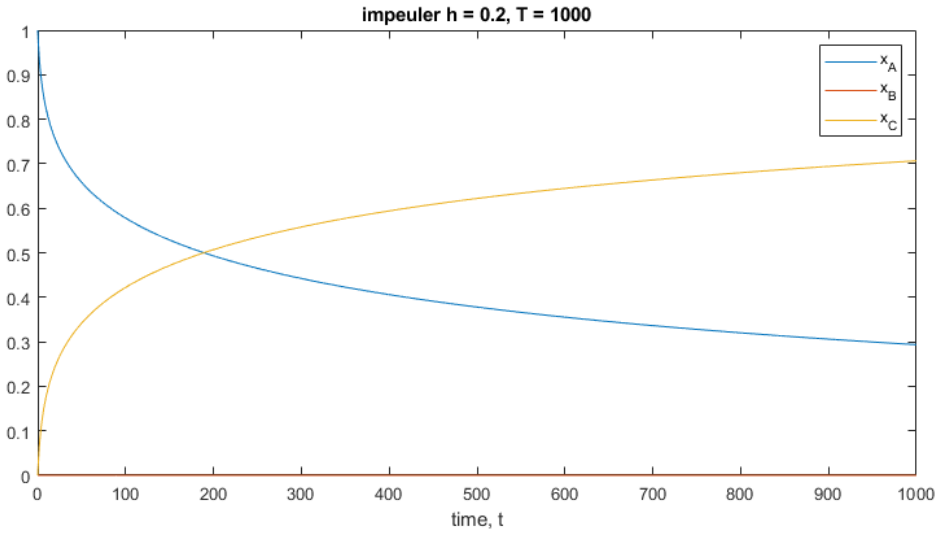
When solving the stiff problem for $t \in [0, 1000]$, a step size of $h = 2 \cdot 10^{-4}$ was stable according to the same condition as in 3a. Using that h, the time to run the scheme took on average 7.79 seconds. The final values for $\vec{x}$ was

$$x_A(T) = 2.934 \cdot 10^{-1}$$
$$x_B(T) = 1.716 \cdot 10^{-6}$$
$$x_C(T) = 7.066 \cdot 10^{-1}$$

For comparison of $h_{\max}$ to the one in 3b; here the theoretical $h_{\max}$ was $h = 2.916 \cdot 10^{-4}$.

3d)

We tested and compared implicit euler using stepsizes 5, 2, 1, 0.5 and 0.2, all of which were stable, and at 0.2 the results were visually equal the rk3 solution (without zooming in). This gave the result:



3e)

method	h	error	computation time
RK	0.000290	0.0000000066659	3.395381
RK	0.000200	0.0000000000004	4.794924
IE	1.000000	0.0004934366807	0.011219
IE	0.100000	0.0000502906203	0.063768
IE	0.010000	0.0000050401267	0.624139
IE	0.002000	0.0000010082254	3.368941
IE	0.001000	0.0000005041253	6.439206

- The primary factor in computation time is the number of steps, and is directly proportional to T/h. There are also minor factors introduced by JIT compilation, branch prediction and caching, but this only introduces timing issues when the total number of steps are low.
- RK3 is a higher order method, so reaching small errors are possible with relatively large values of h. However, the stability criteria on h also introduces a minimal computation time, which can be too large for its intended purpose (and a more stable method is needed).
- In the table we see that IE using $h = 0.002$ is roughly as fast as RK3 near the $h_{\max}$ limit, and using its error as a limit we can conclude that RK3 is will be faster when seeking an error smaller than 10^{-5} (even though there is a large jump in precision per computation time).
- The underlying process for finding the threshold is the same, where we find after which error the computation time of rk3 becomes faster - in part 1 the max value of h was 2.055596, at which RK3 likely performs worse than IE, and thus the graph of optimal precision per computation time would be continuous.