MSAS – Assignment #1: Simulation

Matteo Baio, 232805

1 Implicit equations

1.1 Exercise 1

Let $\mathbf{f}$ be a two-dimensional vector-valued function $\mathbf{f}(\mathbf{x}) = (x_2^2 - x_1 - 2,\ -x_1^2 + x_2 + 10)^\top$, where $\mathbf{x} = (x_1, x_2)^\top$. Find the zero(s) of $\mathbf{f}$ by using Newton's method with $\partial\mathbf{f}/\partial\mathbf{x}$ 1) computed analytically, and 2) estimated through finite differences. Which version is more accurate?

(3 points)

Firstly the problem must be plotted to find how many zero(s) must be found and which educated initial guess can be used. The two components $f_1 = (x_2^2 - x_1 - 2)$ and $f_2 = (-x_1^2 + x_2 + 10)$ are plotted in Fig. 1 both in 3-D (on the left) and 2-D (on the right) plane. The chosen educated initial guesses used to solve the problem are: $[3.0, 2.0]$ and $[3.0, -2.0]$

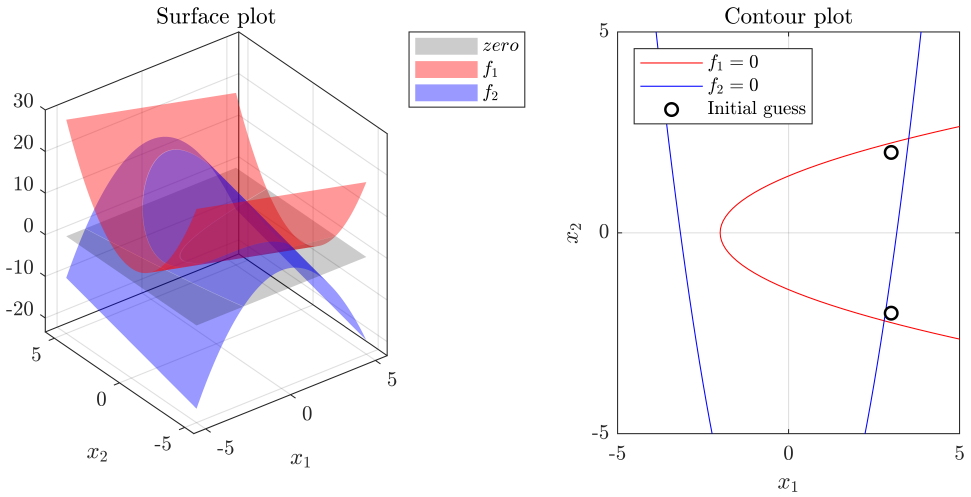

Figure 1: $\mathbf{f}(\mathbf{x})$ zeros and educated initial guess.

The zeros are computed using different implementations of the Newton's method starting from the educated initial guesses just defined. The first implementation relies on the MATLAB Symbolic Math Toolbox™ to compute the exact Jacobian matrix by considering $\mathbf{x}$ as symbolic variable. The second and third implementations estimate the Jacobian matrix respectively by means of the forward difference model Eq. (1) and centered difference model Eq. (2).

$$f'(x) = \frac{f(x+\epsilon) - f(x)}{\epsilon} \tag{1}$$

$$f'(x) = \frac{f(x+\epsilon) - f(x-\epsilon)}{2\epsilon} \tag{2}$$

For both forward and centered difference methods the size of the perturbation is computed as reported in Eq. (3)

$$\epsilon = \max(|x|\sqrt{\varepsilon}, \sqrt{\varepsilon}) \tag{3}$$

Where $\varepsilon = 2.2204 \times 10^{-16}$ is the MATLAB floating point relative accuracy.

All the three implementations converge to the same result in 4 iterations with a similiar level of accuracy as can be observed in Fig. 2. The final zero coordinates are considered founded once the relative error value computed between two consevutive iterations drops behind a user defined tollerance set, in this case, equal to 10^{-6}.

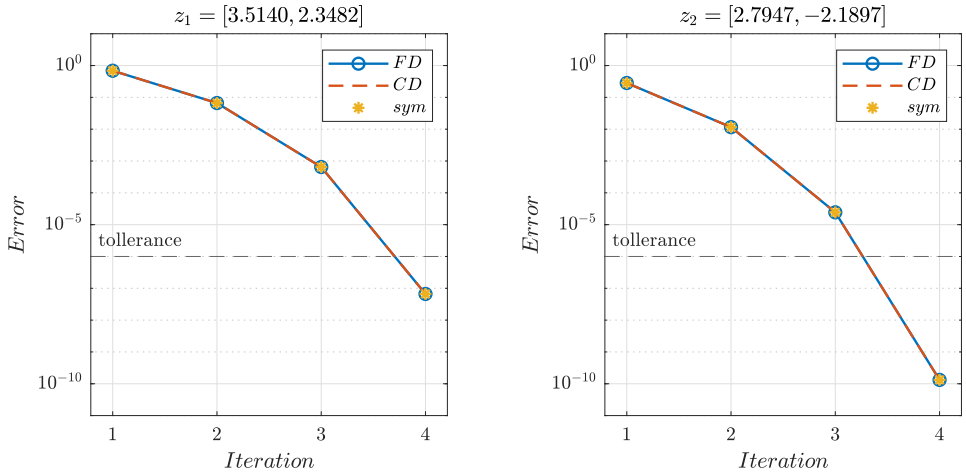

Figure 2: Newton's method relative error convergence

The absolute error on the final result is then obtained by evaluating the norm of the function in the computed zero. Finally, by taking a deeper look to the numerical results obtained and reported in Table 1, it's possibile to define which method is the most accurate.

Method	Error	CPU-time [*sec*]
FD	5.7732×10^{-15}	9.0800×10^{-5}
CD	4.3738×10^{-15}	1.2815×10^{-4}
Sym	2.8436×10^{-15}	2.6300×10^{-2}

(a) Zero 1

Method	Error	CPU-time [*sec*]
FD	1.8310×10^{-15}	9.2350×10^{-5}
CD	1.8310×10^{-15}	1.1976×10^{-4}
Sym	8.8818×10^{-16}	2.5700×10^{-2}

(b) Zero 2

Table 1: Methods compare on zeros computation

All three methods guarantee high accuracy with an absolute error having an order of magnitude of 10^{-15} (near the floating-point relative accuracy). For both zeros the method that, as expected, guarantees the lowest error is the analytical one (Symbolic Math). Must be noted, however, that the price to be paid is a computational cost two orders of magnitude higher than the Finite Difference methods. The Centered Difference method guarantees slightly better result on the first zero while on the second one it performs exactly the same as the Forward Difference, but with a greater computational cost.
In conclusion, there isn't and absolute best method; rather, a choice must be made based on the variable to optimize. If the requirement is to have the maximum precision the most suitable method is the analytical one, while if the requirement is to reduce the computational cost the Finite Forward Differences is the best method.

2 Numerical solution of ODE

2.1 Exercise 2

The Initial Value Problem $\dot{x} = x - 2t^2 + 2$, $x(0) = 1$, has analytic solution $x(t) = 2t^2 + 4t - e^t + 2$. 1) Implement a general-purpose, fixed-step Heun's method (RK2); 2) Solve the IVP in $t \in [0, 2]$ for $h_1 = 0.5$, $h_2 = 0.2$, $h_3 = 0.05$, $h_4 = 0.01$ and compare the numerical vs the analytical solution; 3) Repeat points 1)–2) with RK4; 4) Trade off between CPU time & integration error. (4 points)

The solution of the IVP using the fixed-step Heun's method is computed using four different time step values and compared with the analytical solution in Fig. 3.

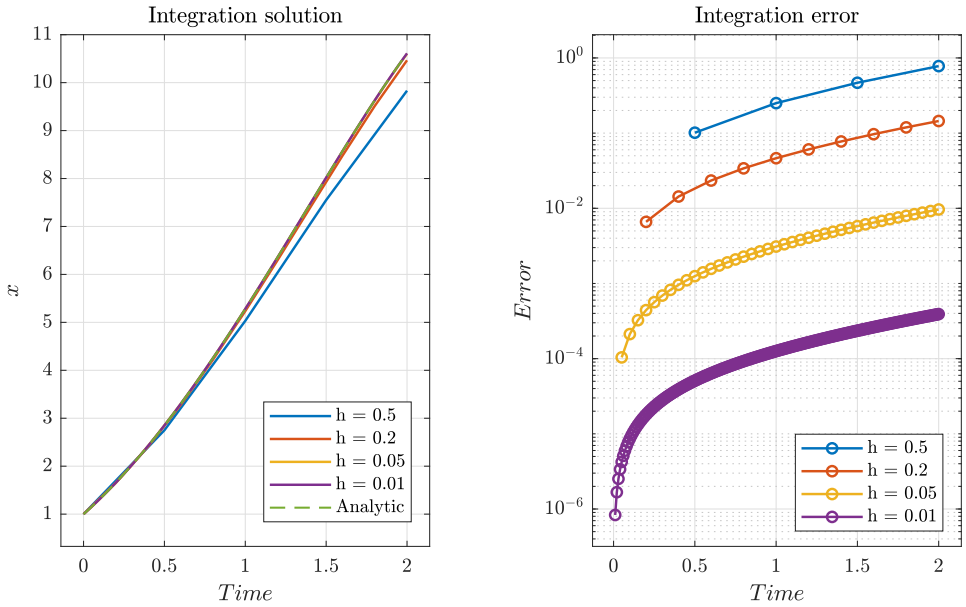

Figure 3: Heun's method solution and error over time

The method results stable with every time step and the error rises as the time goes on. By decreasing the step size, as expected, the precision increases and the error reduces so much that there is no longer a visible difference between the analytical and numerical solution for $h = 0.05$ and $h = 0.01$.
Following the same procedure using the fourth order Runge-Kutta method the results presented in Fig. 4 are obtained.
Even in this case the method is stable with every time step and the error incresese as the simulation time rises. Unlike the previous case, the error is so small (always less than 10^{-2}) that it is not visible in the solution graph (left figure) for all time steps at every time. Once again, as expected, by observing the integration error plot (right figure) can be noticed that the precision grows up as the time step dimension goes down.
Finally the trade off between CPU time and integration error is considered. For each method and time step a *for* cycle of 1000 runs is exectued, the time needed to complete these tasks are measured and post processed by means of an arithmetic mean. By following this procedure a valid estimation of the time cost of each cases is retrived and presented in Fig. 5 and Table 2
In conclusion for both methods the choice of a smaller step size produces higher precision but with higher computational cost. As it's possible to notice from Fig. 5 the best solutions in terms of trade off are fourth order Runge-Kutta method with 0.5 and 0.2 as time step. In alternative, the best step size for the Heun's method (again in terms of trade off) is 0.05.

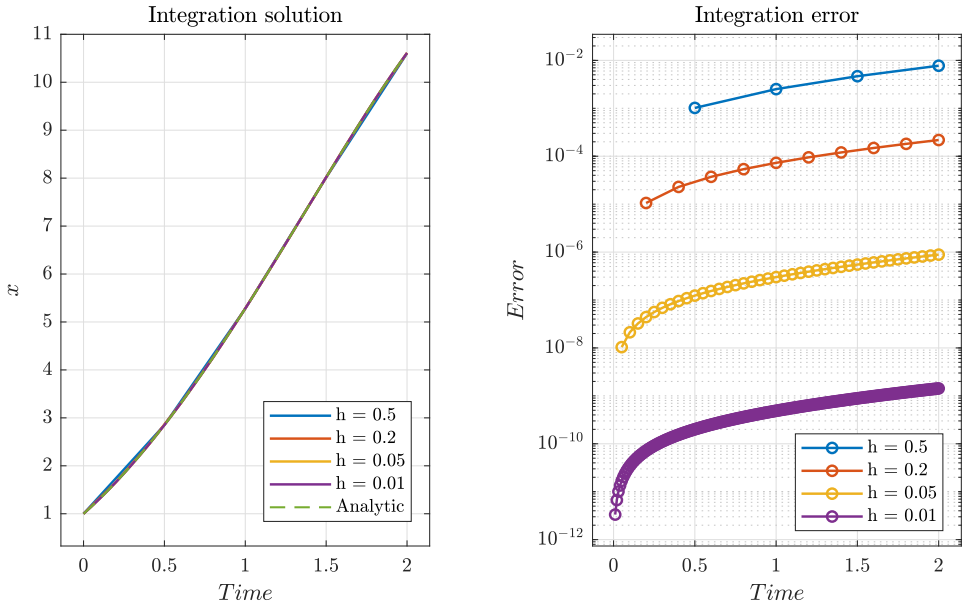

Figure 4: Runge-Kutta 4 method solution and error over time

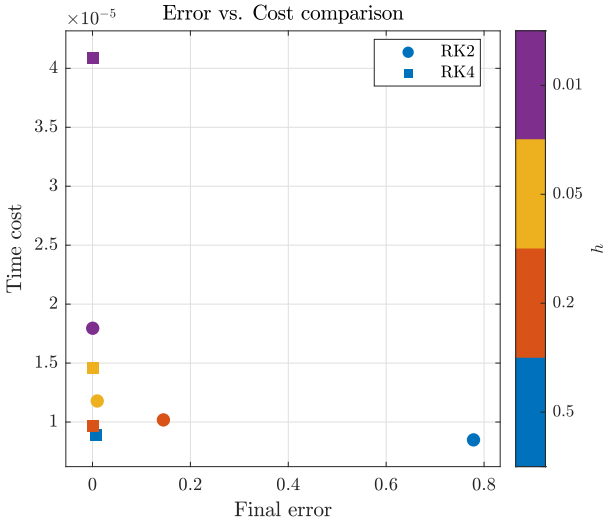

Figure 5: Comparison between final error and CPU time for Heun's and Runge-Kutta methods

h	Error	CPU-time $[sec]$
0.50	7.7842×10^{-1}	8.4865×10^{-6}
0.20	1.4483×10^{-1}	1.0180×10^{-5}
0.05	9.6397×10^{-3}	1.1786×10^{-5}
0.01	3.9124×10^{-4}	1.7955×10^{-5}

(a) Heun's method

h	Error	CPU-time $[sec]$
0.50	7.7334×10^{-3}	8.9280×10^{-6}
0.20	2.1790×10^{-4}	9.6987×10^{-6}
0.05	8.8427×10^{-7}	1.4548×10^{-5}
0.01	1.4275×10^{-9}	4.0908×10^{-5}

(b) Runge-Kutta 4 method

Table 2: Methods error and CPU time results

2.2 Exercise 3

Let $\dot{\mathbf{x}} = A(\alpha)\mathbf{x}$ be a two-dimensional system with $A(\alpha) = [0, 1; -1, 2\cos\alpha]$. Notice that $A(\alpha)$ has a pair of complex conjugate eigenvalues on the unit circle; α denotes the angle from the $\mathrm{Re}\{\lambda\}$-axis. 1) Write the operator $F_{\mathrm{RK2}}(h, \alpha)$ that maps $\mathbf{x}_k$ into $\mathbf{x}_{k+1}$, namely $\mathbf{x}_{k+1} = F_{\mathrm{RK2}}(h, \alpha)\,\mathbf{x}_k$. 2)With $\alpha = \pi$, solve the problem "Find $h \geq 0$ s.t.max $(|\mathrm{eig}(F(h, \alpha))|) = 1$". 3) Repeat point 2) for $\alpha \in [0, \pi]$ and draw the solutions in the $(h\lambda)$-plane. 4) Repeat points 1)–3) with RK4.

(5 points)

The operator $F_{\mathrm{RK2}}(h, \alpha)$ can be easily computed by considering the second order Runge-Kutta method integration scheme:

$$\mathbf{x^p} = \mathbf{x}_k + \beta_{11}\, h\, \mathbf{f}\left(\mathbf{x}_k, t_k\right) \tag{4a}$$

$$\mathbf{x^c}_{k+1} = \mathbf{x}_k + \alpha_{21}\, h\, \left[\beta_{21}\, \mathbf{f}\left(\mathbf{x}_k, t_k\right) + \beta_{22}\, \mathbf{f}\left(\mathbf{x^p}, t_k + \alpha_{11}\, h\right)\right] \tag{4b}$$

The function $\mathbf{f}$ can be rewritten as function of the matrix $\mathbf{A}(\alpha)$ and $\mathbf{x}_k$ as follows:

$$\mathbf{A}(\alpha)\, \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_k, t_k\right) \tag{5}$$

Finally the operator can be retrived by substituting a set of α and β parameters valid for the second order Runge-Kutta method:

$$F_{RK2}(h, \alpha) = \frac{\mathbf{x^c}_{k+1}}{\mathbf{x}_k} = \mathbf{I} + h\, \mathbf{A}(\alpha) + \frac{h^2}{2}\mathbf{A}^2(\alpha) \tag{6}$$

The problem "Find $h \geq 0$ s.t.max $(|\mathrm{eig}(F(h, \alpha))|) = 1$" is then rewritten as a root finding problem and solved for $\alpha \in [0, \pi]$.

$$S(h, \alpha) = \max\left(|\mathrm{eig}(F(h, \alpha))|\right) - 1 \tag{7}$$

Firstly the function $S(h, \pi)$ is plotted for different h values to pick a good educated initial guess to find the zero of the function through $fzero$ MATLAB function. Once the initial guess is defined the the problem is solved for $\alpha = \pi$ and returns the solution $h = 2.0000$. Iteratively, the problem is solved for all other α using the solution found in the previous step as initial guess for $fzero$. The solution in the $(h\lambda)$-plane for $\alpha \in [0, \pi]$ is finally reported in Fig. 6.

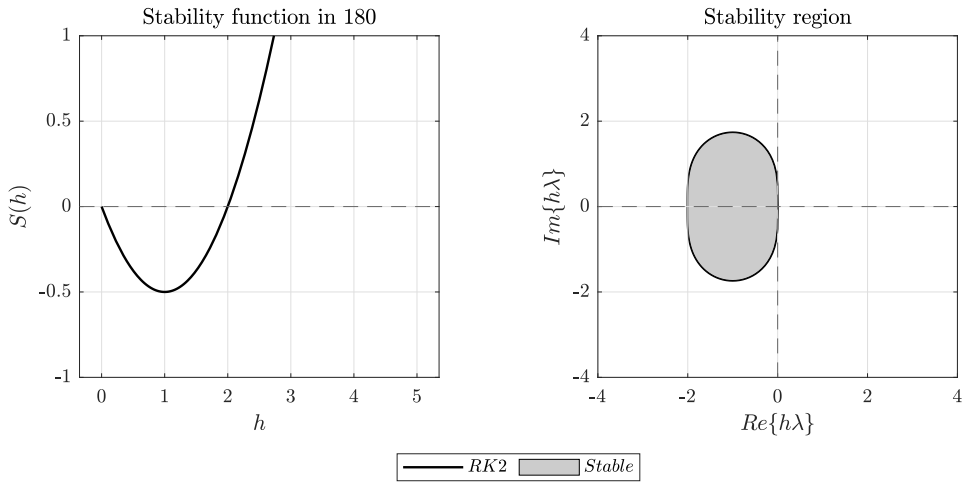

Figure 6: Solution of the problem with RK2

The exact same process is then applied with the fourth order Runge-Kutta method where the operator $F_{RK4}(h, \alpha)$ is:

$$F_{RK4}(h, \alpha) = \mathbf{I} + h\,\mathbf{A}(\alpha) + \frac{h^2}{2}\mathbf{A}^2(\alpha) + \frac{h^3}{6}\mathbf{A}^3(\alpha) + \frac{h^4}{24}\mathbf{A}^4(\alpha) \tag{8}$$

The solution of the problem "Find $h \geq 0$ s.t.max $(|\text{eig}(F(h, \pi))|) = 1$" with $F(h, \pi) = F_{RK4}(h, \pi)$ is $h = 2.7853$. Once the problem is solved for $\alpha \in [0, \pi]$ is possible, once again, to plot the results in the $(h\lambda)$-plane (Fig. 7).

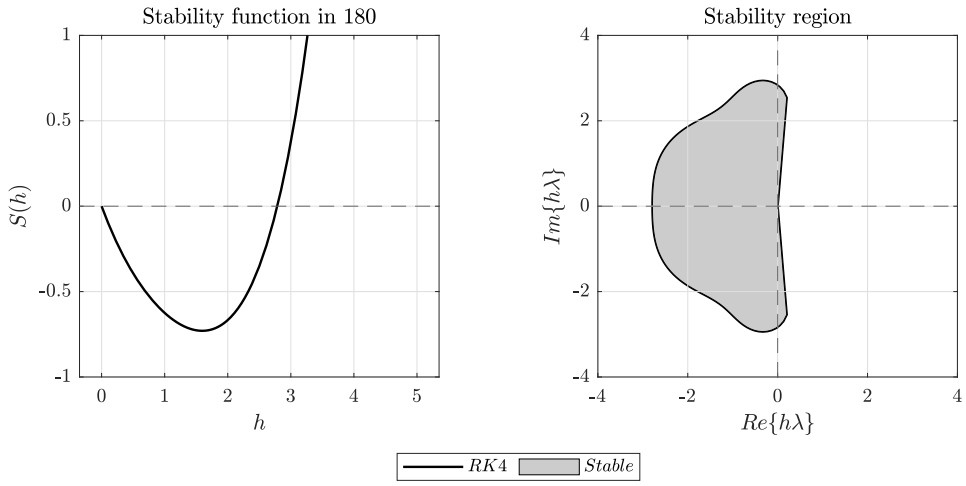

Figure 7: Solution of the problem with RK4

In conclusion the stability domain for second and fourth order Runge-Kutta methods are obtained and plotted in the same graph in Fig. 8.

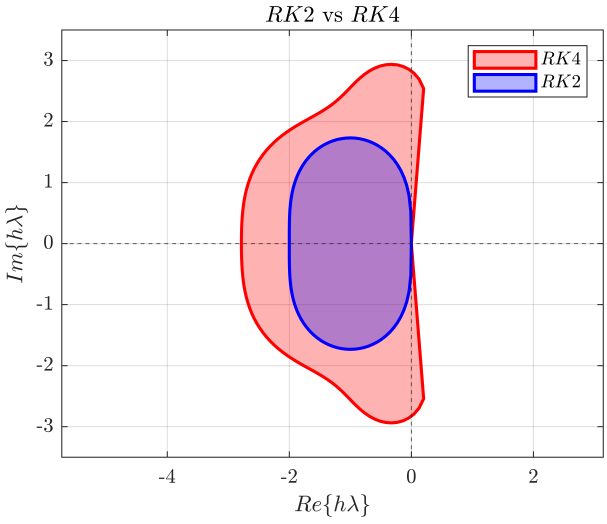

Figure 8: Stability region of RK2 and RK4

2.3 Exercise 4

Consider the IVP $\dot{\mathbf{x}} = A(\alpha)\mathbf{x}$, $\mathbf{x}(0) = [1,1]^T$, to be integrated in $t \in [0,1]$. 1) Take $\alpha \in [0,\pi]$ and solve the problem "Find $h \geq 0$ s.t. $\|\mathbf{x}_{\text{an}}(1) - \mathbf{x}_{\text{RK1}}(1)\|_\infty = \text{tol}$", where $\mathbf{x}_{\text{an}}(1)$ and $\mathbf{x}_{\text{RK1}}(1)$ are the analytical and the numerical solution (with RK1) at the final time, respectively, and $\text{tol} = \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. 2) Plot the four locus of solutions in the $(h\lambda)$-plane; plot also the function evaluations vs tol for $\alpha = \pi$. 3) Repeat points 1)–2) for RK2 and RK4.

(4 points)

Following the same procedure of exercise 3, the operator $F(h,\alpha)$ of the numerical method RK1 is retrived (Eq. (9)) and the problem is rewritten as a root finding problem (Eq. (10)).

$$F_{RK1}(h,\alpha) = \mathbf{I} + h\,\mathbf{A}(\alpha) \tag{9}$$

$$S_{RK1}(h,\alpha) = \|\mathbf{x}_{\text{an}}(1) - \mathbf{x}_{\text{RK1}}(1)\|_\infty - \text{tol} \tag{10}$$

The problem is firstly solved for every tollerance values in $\alpha = \pi$ starting from an educated initial guess obtained by the plot of the function $S_{RK1}(h,\pi)$. As done in the previous exercise the solution for all the other α are obtained by using as initial guess the result of the problem at the previous step. In Fig. 9 and Fig. 10 the function is plotted to find the initial guess (left figure) and the locus of solution for each tollerance values (right figure) are shown.

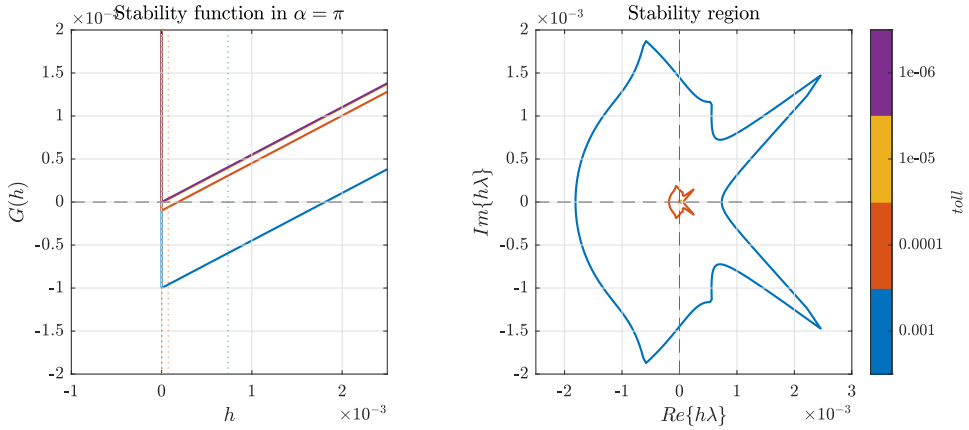

Figure 9: Solution of the problem with RK1

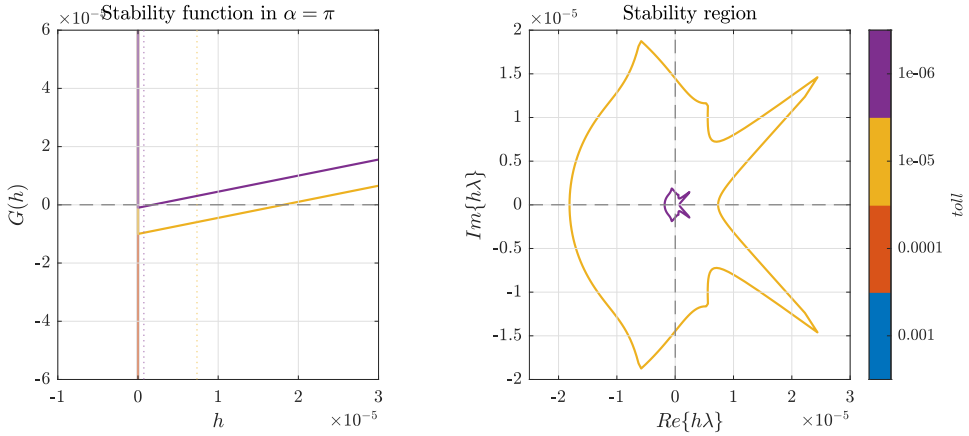

Figure 10: Zoom of the solution obtained with RK1

By repeating the exact same procedure using the operators RK2 (Eq. (6)) and RK4 (Eq. (8)) the solutions in Fig. 11 and Fig. 12 are obtained.

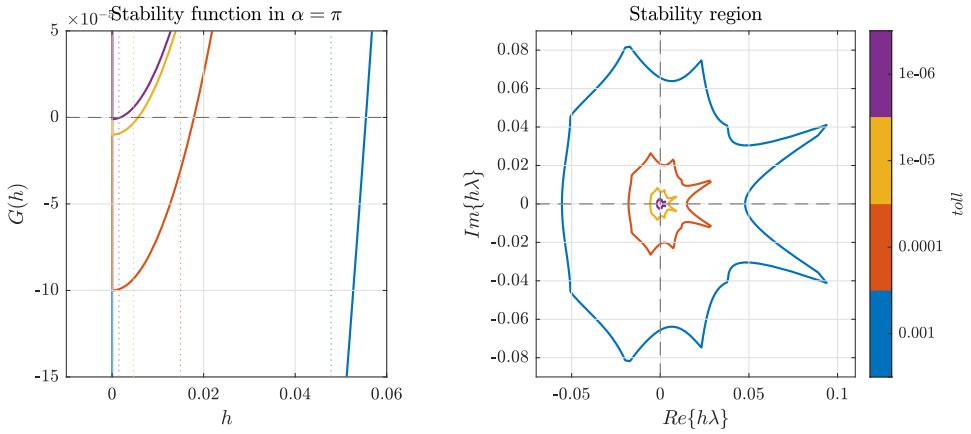

Figure 11: Solution of the problem with RK2

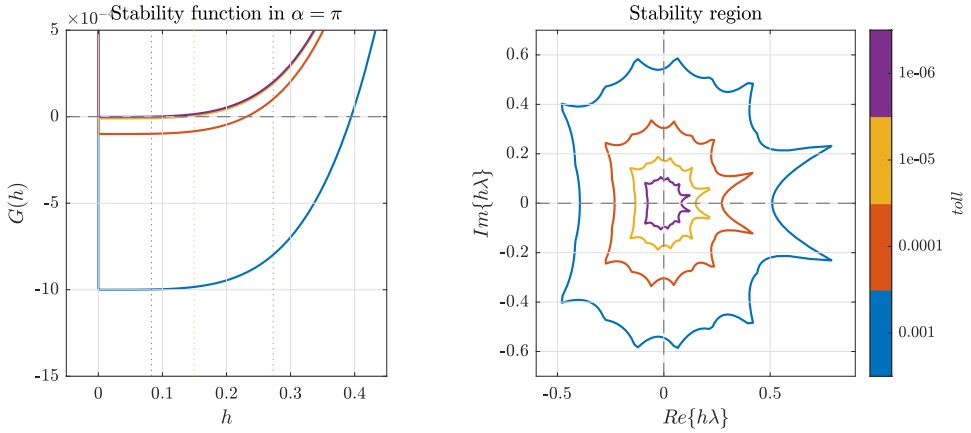

Figure 12: Solution of the problem with RK4

Finally in Fig. 13 the number of function evaluations for each methods and tollerances are plotted in $\alpha = \pi$. As expected, a bigger tollerance value requires less function evaluations.
It's intersting to notice how RK1 is the method that requires always the highest number of function evaluations and RK2 requires more function evaluations than RK4. This is because since they are methods of lower order a smaller step size, and so a higher number of steps, is needed to satisfy the imposed tollerance value.

Table 3: Solutions of root finding problem

Toll	RK1	RK2	RK4
10^{-3}	1.81×10^{-3}	5.54×10^{-2}	3.95×10^{-1}
10^{-4}	1.81×10^{-4}	1.79×10^{-2}	2.31×10^{-1}
10^{-5}	1.81×10^{-5}	5.69×10^{-3}	1.33×10^{-1}
10^{-6}	1.81×10^{-6}	1.80×10^{-3}	7.61×10^{-2}

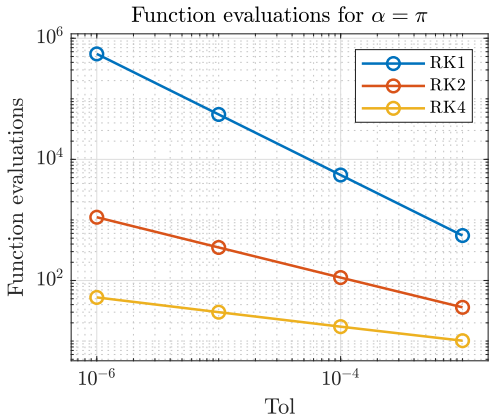

Figure 13: Methods function evaluation vs tollerance

2.4 Exercise 5

Consider the backinterpolation method $\text{BI2}_{0.4}$. 1) Derive the expression of the linear operator $B_{\text{BI2}_{0.4}}(h,\alpha)$ such that $\mathbf{x}_{k+1} = B_{\text{BI2}_{0.4}}(h,\alpha)\mathbf{x}_k$. 2) Following the approach of point 3) in Exercise 3, draw the stability domain of $\text{BI2}_{0.4}$ in the $(h\lambda)$-plane. 3) Derive the domain of numerical stability of BI2_θ for the values of $\theta = [0.1, 0.3, 0.7, 0.9]$.

(5 points)

The operator $B_{BI2_{0.4}}(h,\alpha)$ can be computed starting from the second order Runge-Kutta method operator. The Eq. (6) can be rewritten considering a time step equal to θh starting from $\mathbf{x}_k$ (Eq. (11a)) and equal to $-(1-\theta)h$ starting from $\mathbf{x}_{k+1}$ (Eq. (11b)):

$$\mathbf{x}_{k+\theta h} = \left[\mathbf{I} + h\theta\,\mathbf{A} + \frac{h^2\theta^2}{2}\mathbf{A}^2\right]\mathbf{x}_k \tag{11a}$$

$$\mathbf{x}_{k+\theta h} = \left[\mathbf{I} - (1-\theta)h\,\mathbf{A} + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2\right]\mathbf{x}_{k+1} \tag{11b}$$

Equalising the two equations just written it's easy to retrive the $B_{BI2_\theta}(h,\alpha)$ operator:

$$B_{BI2_\theta}(h,\alpha) = \frac{\mathbf{x}_{k+1}}{\mathbf{x}_k} = \left[\mathbf{I} + h\theta\,\mathbf{A} + \frac{h^2\theta^2}{2}\mathbf{A}^2\right]\left[\mathbf{I} - (1-\theta)h\,\mathbf{A} + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2\right]^{-1} \tag{12}$$

Finally by substituting $\theta = 0.4$ in Eq. (12) the $B_{BI2_{0.4}}(h,\alpha)$ is computed. Following the same approach of exercise 3, the stability region for $\theta = [0.1, 0.3, 0.4, 0.7, 0.9]$ is computed and plotted in the $(h\lambda)$-plane (Fig. 14).

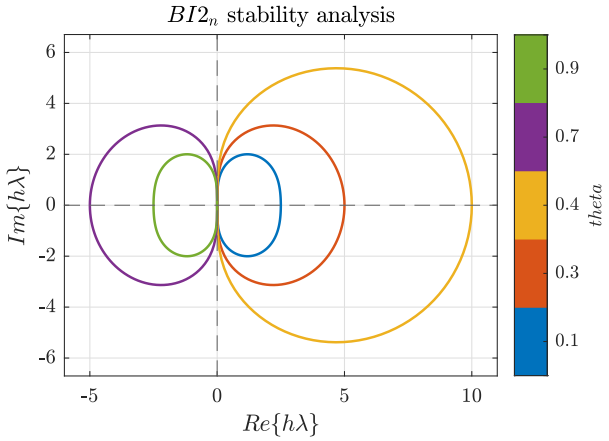

Figure 14: Stability region of $BI2_\theta$ for different θ

It's important to notice that the $BI2_\theta$ method is "A-stable" for $\theta = [0.1, 0.3, 0.4]$ while simply stable for the remaning θ values tested. This particular characteristic isn't visible from Fig. 14 but can be observerd in Fig. 17, Fig. 18 and Fig. 19. Because of the different position of the stability region boundary the starting point of the stability problem is placed in $\alpha = 0$.

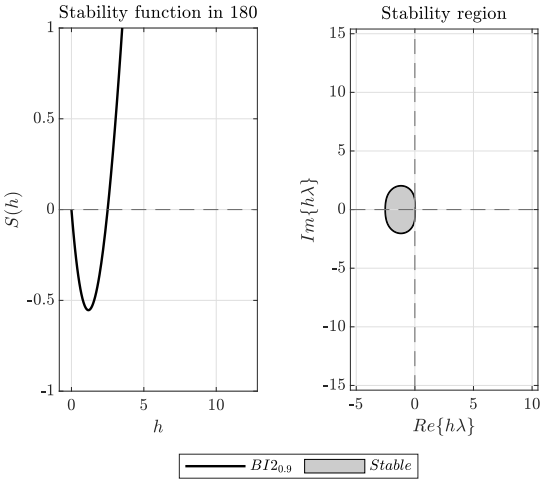

Figure 15: Solution of the problem with $BI2_{0.9}$

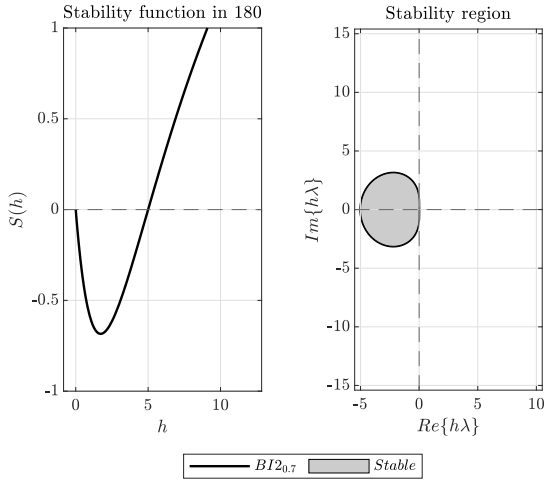

Figure 16: Solution of the problem with $BI2_{0.7}$

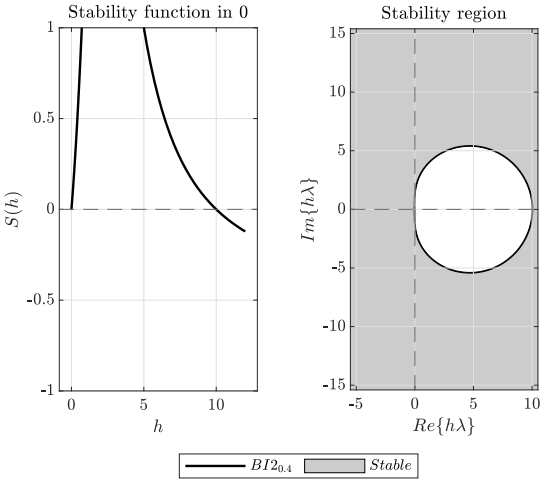

Figure 17: Solution of the problem with $BI2_{0.4}$

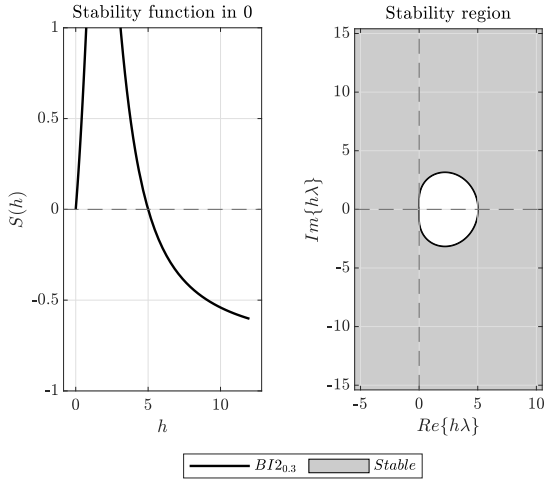

Figure 18: Solution of the problem with $BI2_{0.3}$

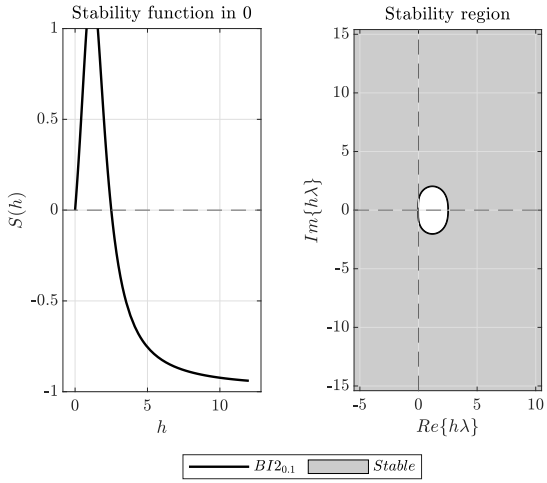

Figure 19: Solution of the problem with $BI2_{0.1}$

2.5 Exercise 6

Consider the IVP $\dot{\mathbf{x}} = B\mathbf{x}$ with $B = [-180.5, 219.5; 179.5, -220.5]$ and $\mathbf{x}(0) = [1, 1]^T$ to be integrated in $t \in [0, 5]$. Notice that $\mathbf{x}(t) = e^{Bt}\mathbf{x}(0)$. 1) Solve the IVP using RK4 with $h = 0.1$; 2) Repeat point 1) using implicit extrapolation technique IEX4; 3) Compare the numerical results in points 1) and 2) against the analytic solution; 4) Compute the eigenvalues associated to the IVP and represent them on the $(h\lambda)$-plane both for RK4 and IEX4; 5) Discuss the results. (4 points)

The IVP is solved using the operators F_{RK4} (Eq. (8)) and F_{IEX4} (Eq. (13)) with a time step equal to $h = 0.1$.

$$F_{IEX4}(h, \alpha) = -\frac{1}{6}(\mathbf{I} - h\mathbf{A})^{-1} + 4\left(\mathbf{I} - \frac{h\mathbf{A}}{2}\right)^{-2} - \frac{27}{2}\left(\mathbf{I} - \frac{h\mathbf{A}}{3}\right)^{-3} + \frac{32}{3}\left(\mathbf{I} - \frac{h\mathbf{A}}{4}\right)^{-4} \tag{13}$$

The two solutions obtained from the numerical methods are compared with the analytical solution Eq. (14) in Fig. 20.

$$\mathbf{x}(t) = e^{Bt}\mathbf{x}(0) \tag{14}$$

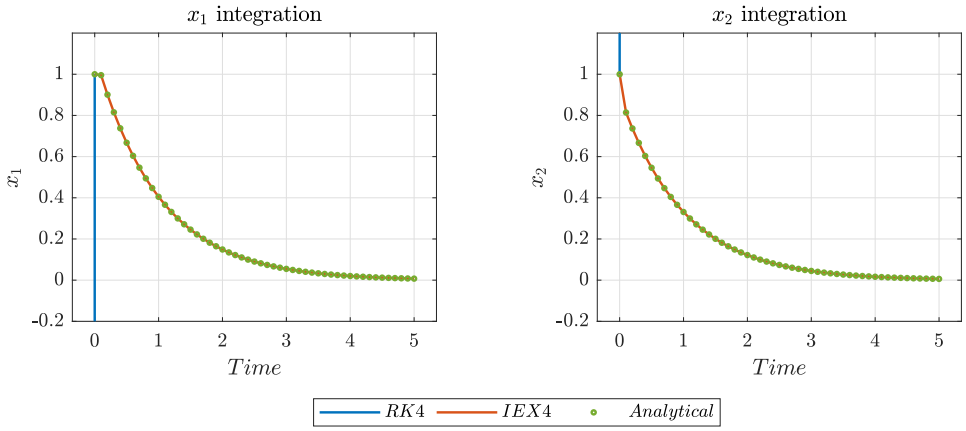

Figure 20: Integration results of the IVP

The numerical solutions obtained through RK4 diverge for both variables, while the ones obtained with IEX4 method follows almost perfectly the analytical solution. Due to this behavior it's possibile to hypotize that the IVP just analyzed is a "Stiff problem" with the two negative eigenvalues having very different magnitude. The stability region of the two methods are, once again, computed (Fig. 22 and Fig. 23) and the eigenvalues of the system, multiplied for the time step $h = 0.1$, are rappresented in the same $(h\lambda)$-plane (Fig. 21).

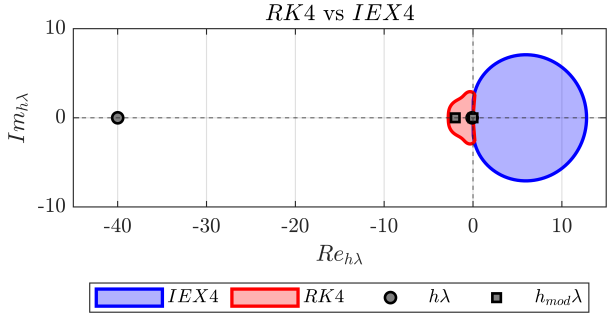

Figure 21: IVP eigenvalues and stability region of RK4 and IEX4

It must be noticed that in (Fig. 21) the red colored area rappresents the stability region for RK4 method, while the blu colored area rappresents the instability region for IEX4 method.

Table 4: Eigenvalues of the system multiplied for the different time steps

step	$h\lambda_1$	$h\lambda_2$
0.100	-0.100	-40
0.005	-0.005	-2

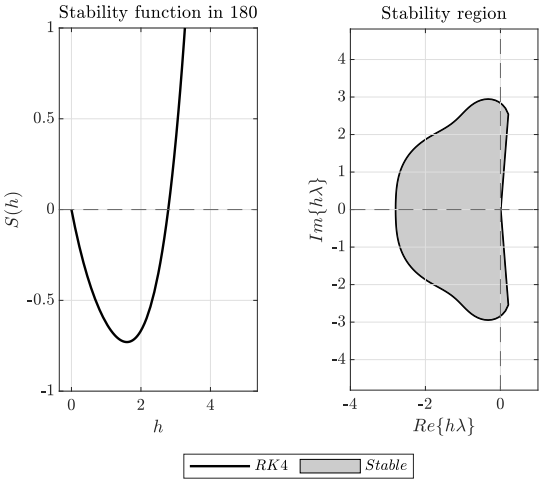

Figure 22: Stability problem for RK4

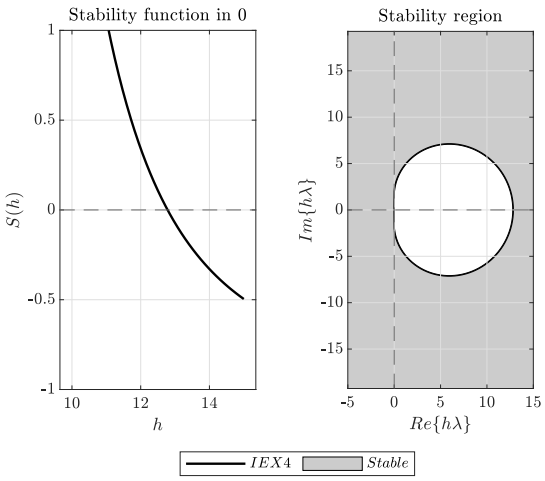

Figure 23: Stability problem for IEX4

The two eigenvalues, as expected, are both stable but has two very different magnitudes. For this reason the problem can be classified as "stiff" and the IEX4, since is an "A-stable" method, is the ideal choiche. An alternative solution to integrate this IVP with RK4 could be to reduce the time step utill both eigenvalues enters in the stability region. This has been done by reducing the time step to $h = 0.005$ as visible in (Fig. 21). The results of the new integration are reported in (Fig. 24) as *RK4mod* and, as expected, is consistent with the analytical solution and the IEX4 solution.

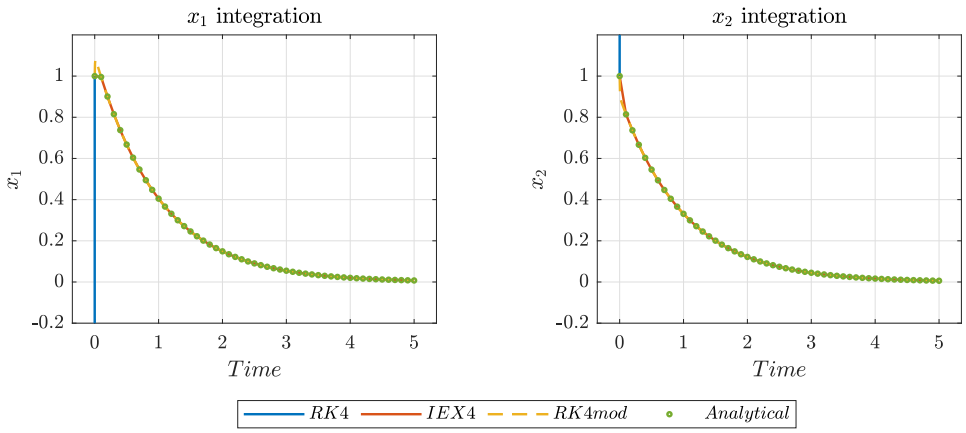

Figure 24: Compare integration results of the IVP with *RK4mod*

2.6 Exercise 7

Consider the two-dimensional IVP

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{5}{2}\left[1+8\sin(t)\right]x_1\\ (1-x_1)x_2+x_1\end{bmatrix}, \qquad \begin{bmatrix}x_1(t_0)\\ x_2(t_0)\end{bmatrix} = \begin{bmatrix}1\\ 1\end{bmatrix}$$

1) Solve the IVP using AB3 in $t \in [0,3]$ for $h = 0.1$; 2) Repeat point 1) using AM3, ABM3, and BDF3; 3) Discuss the results.

(5 points)

The IVP is solved using AB3, AM3, ABM3, and BDF3 methods; the results are reported in Fig. 25, Fig. 26, Fig. 27, Fig. 28. Since they are all multistep methods of order 3 and the initial value is known only in t_0 two additional starting point must be defined in $t_1 = t_0 + h$ and $t_2 = t_0 + 2h$. The "Startup" problem is solved in two different ways: 1) Using a single step method of equivalent order (in this case RK3) to find $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$; 2) Using the same multistep method and increasing the order until the one desired is reached (in this case third order).

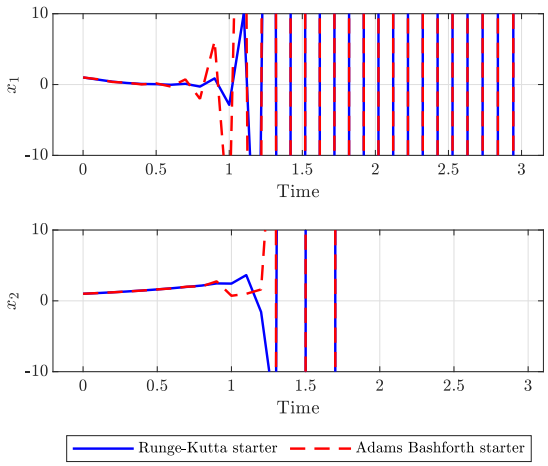

Figure 25: IVP integration with AB3

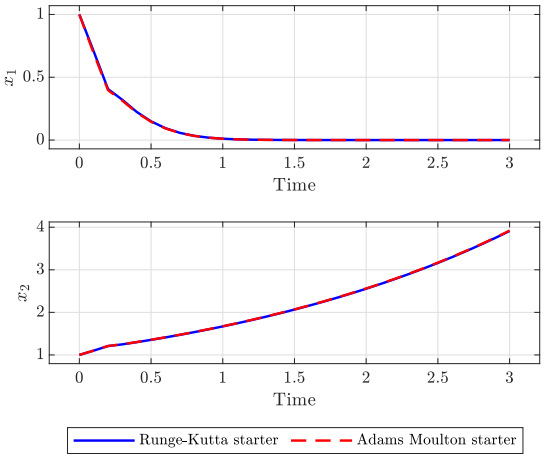

Figure 26: IVP integration with AM3

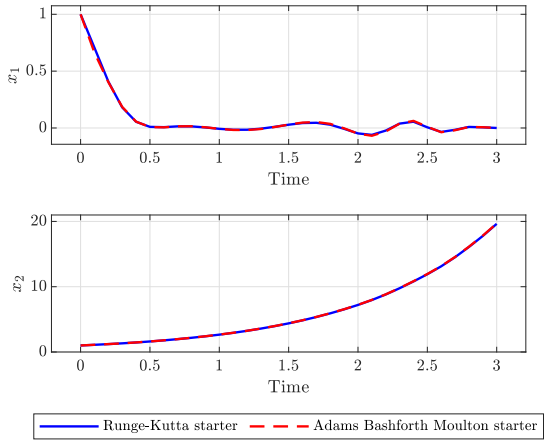

Figure 27: IVP integration with ABM3

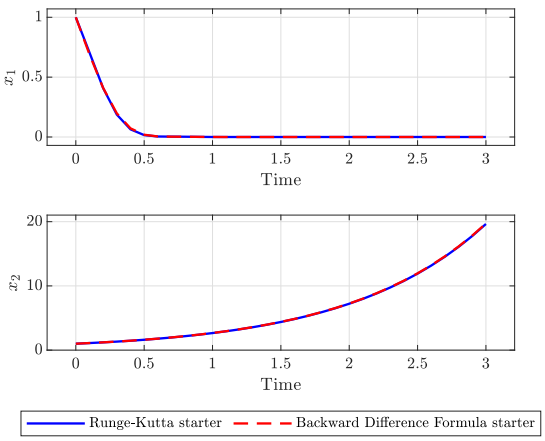

Figure 28: IVP integration with BDF3

Both the startup approaches produce the same results, besides the one obtained with Adam Bashforth methods which diverges earlier by using the increasing order approach. To have a better overview, all the solutions obtained with the different multistep methods are plotted together in Fig. 29.

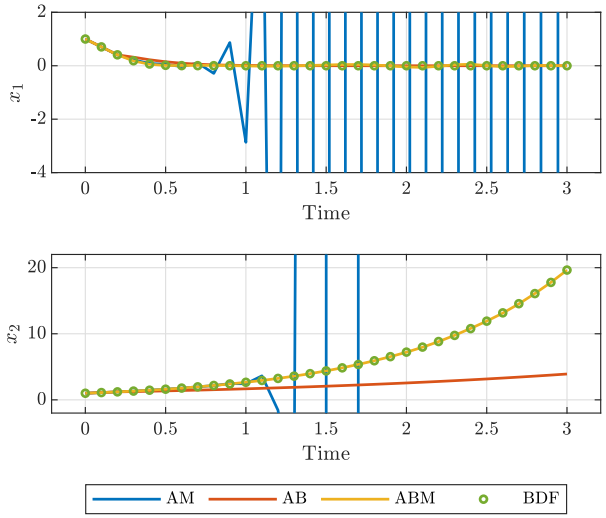

Figure 29: Compare between different multistep integration methods

The solution associated to the x_2 variable diverges with every method implemented, while, the x_1 variable is totally unstable only when integrated with Adams Bashforth method. A particular oscillatory behavior can also be observed on the x_1 variable when integrated with the Adam Bashforth Moulton method. To better understand the reason behind these trends the eigenvalues of the system are computed and compared with the stability region boundary of the different methods.
The eigenvalues are computed starting form the analytical solution of the first equation Eq. (15) which is reported in Eq. (16).

$$\dot{x}_1 = -\frac{5}{2}\left[1 + 8\sin(t)\right] x_1 \tag{15}$$

$$x_1 = e^{-\frac{5}{2}(t-t_0)+20(\cos(t)-\cos(t_0))} \tag{16}$$

The Eq. (16) can now be substited in the second equation of the system and finally the two eigenvalues are retrived as function of time t:

$$\lambda_1(t) = -\frac{5}{2}\left[1 + 8\sin(t)\right] \tag{17}$$

$$\lambda_2(t) = 1 - e^{20\cos(t)-2.5t-20} \tag{18}$$

The eigenvalues have no immaginary part, so some observations about their stability over time can be done by considering the interceptions of the stability domain of each method with the real axis. In particular, as reported in Fig. 30, the boundary of AB3, ABM3 and AM3 stability regions are shown on the graph as dashed lines. No line is drawn for BDF3 method since it is "A stable" and so all the negative part of the complex plain is included in its stability region. In Table 5 the limits of the analyzed multistep methods stability region are reported.

	AB3	AM3	ABM3	BDF3
$\min Re(h\lambda)$	-0.6	-6.0	-1.7	$-\infty$
$\max Re(h\lambda)$	0.0	0.0	0.0	0.0

Table 5: TBD

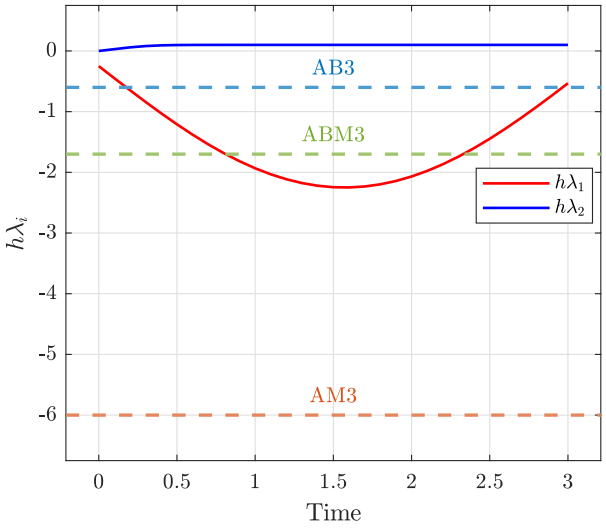

Figure 30: TBD

The product $h\lambda_2$ results always greater or equal than zero, so it's unstable for every $t \in (0, 3]$ and it's out of the stability region of every multistep method applied.
The product $h\lambda_1$ is always negative in the time interval considered and has a parabolic behavior. For this reason only AM3 and BDF3 methods completly contains in their domain of numerical stability the eigenvalue envelop. It's so explained the oscillatory behavior observed with the ABM3 method which doesn't include in its domain the eignevalue when its values drops behind -1.7. The situation is even worse when AB3 method is considered; In this case $h\lambda_1$ is outside its stabiliy region as soon as it's value drops behind -0.6, which turns out to be the case for the major part of the studied time interval.
In conclusion the two methods that can be used to solve the IVP with the time step $h = 0.1$ are AM3 and BDF3. ABM3 and AB3 can be used only with a smaller timestep which guarantee the eigenvalues to fall within the stability region in every time instant.