MSAS – Assignment #1: Simulation

Fontan Anna, 945648

1 Symbolic manipulation

Exercise 1

Consider a simple mass-spring system whose equation of motion is described by $m\ddot{x} + kx = 0$. Using Matlab's Symbolic Toolbox: 1) Transform the equation of motion in two first order equations introducing $v = \dot{x}$; 2) Save the result of point 1) into a Matlab file using `matlabFunction`; 3) Using $m = 1$ and $k = 100$, integrate the system in $t \in [0, 10]$ with initial conditions $x_0 = 0.2$ and $v_0 = 0$; 4) Add to the equation of motion a damping term $b\dot{x}$; 5) Find the equilibrium point of the system; 6) Put the equation of motion in the second-order canonical form, solve it using `dsolve`. Show the system response in relation to the location of the eigenvalues. Make sure to reproduce over-, under-, and critically damped cases. 7) Plot the solutions using `fplot`.

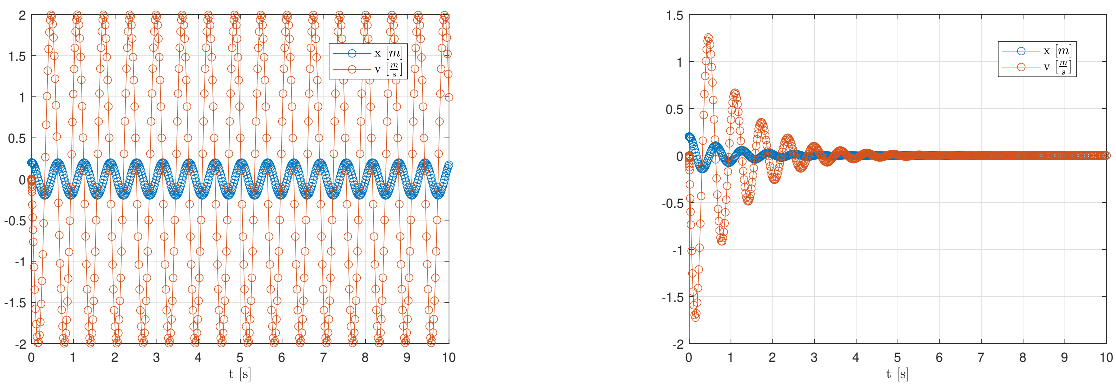
In this exercise the following equation of a simple mass-spring system is examined:

$$m\ddot{x}(t) + kx(t) = 0 \tag{1}$$

Eq. 1 becomes a system of two differential equations if the velocity is considered as well:

$$\begin{cases} \dot{x}(t) = v(t) & (2) \\ m\dot{v}(t) + kx(t) = 0 & (3) \end{cases}$$

where m is the mass and k the stiffness terms.
The results obtained after plugging the data (mass, stiffness, initial velocity and position) are shown in Fig. 1a. Indeed, since the system has not a damping term, both the velocity $v(t)$ and the position $x(t)$ integrated (though MATLAB function ode45) over a time of 10 seconds are behaving as periodic functions.



(a) Undamped ($\xi = 0$) **(b)** Under-damped ($\xi = 0.1$)

Figure 1: Mass - spring system (n.b. ξ: damping ratio)

If the damping term is added the Eq. 3 becomes Eq. 5 and therefore the matrix that links the first derivatives $\dot{x}$(t) and $\dot{v}$(t) with x(t) and v(t) themselves is given by Eq. 7.

$$\begin{cases} \dot{x}(t) = v(t) & (4) \\ m\dot{v}(t) + bv(t) + kx(t) = 0 \rightarrow \dot{v}(t) = -\dfrac{b}{m}v(t) - \dfrac{k}{m}x(t) & (5) \end{cases}$$

$$[\dot{x}(t), \dot{v}(t)]^T = [M][x(t), v(t)]^T \tag{6}$$

$$M = \begin{bmatrix} 0 & 1 \\ {\omega_n}^2 & -2\xi\omega_n \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \tag{7}$$

whether M is expressed in the canonical form (with the natural frequency ω_n and the damping ratio ξ) or not.
Once the damping is considered, x(t) and v(t) asymptotically tend to zero: their amplitude slowly reduces with time (as shown in Fig. 1b). Their behaviours change with the damping ratio, as one can notice from Fig. 3 and Fig. 4, depending on whether the system is under-, critically or over-damped.

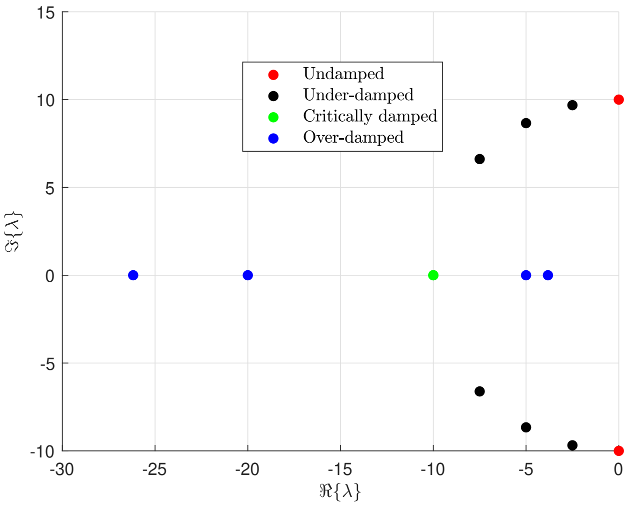


Figure 2: Eigenvalues of matrix in Eq. 7 for different values of ξ

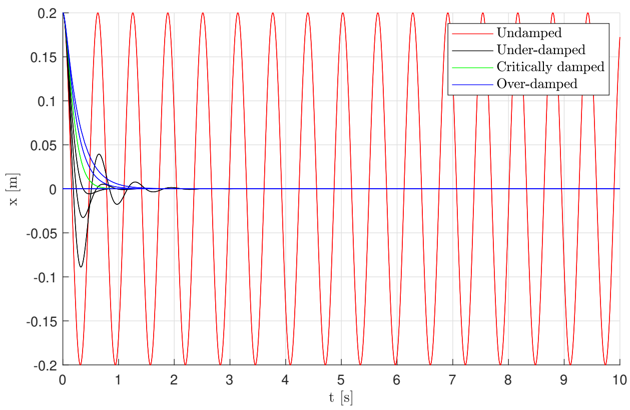


Figure 3: x(t) of the system for the different values of the damping ratio

Moreover the functions x(t) and v(t) are linked to the position of the eigenvalues of the matrix of the system (Eq. 7); Fig. 2 shows the eigenvalues for a variable ξ, while the natural frequency is fixed as $\omega_n = \sqrt{\frac{k}{m}}$.
The following observations can be deduced (see Fig. 3 and Fig. 4):

- for $\xi = 0$ the system is undamped; the two complex and conjugate eigenvalues lie on

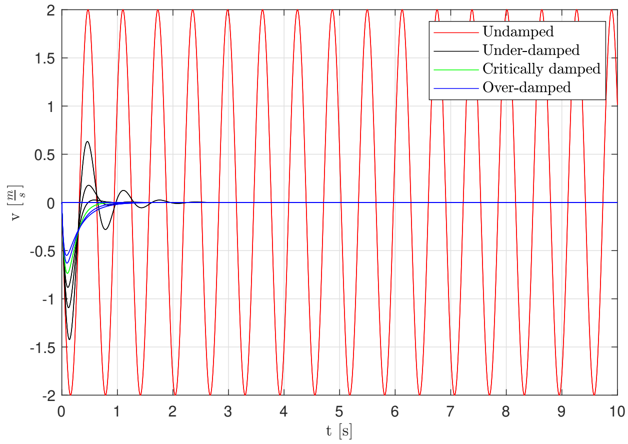


Figure 4: v(t) of the system for the different values of the damping ratio

the imaginary axis. This leads to an unchanging motion during time, which thus results periodic;

- for $\xi \in (0,1)$ the system is under-damped and the eigenvalues are couples of complex conjugates. This leads to a motion that tends to zero; moreover, the higher the value of ξ, the faster the oscillations;
- for $\xi = 1$ the system is critically damped and the two eigenvalues are coincident and real. In this case the system reaches the equilibrium without performing any oscillations;
- for $\xi > 1$ the system is over-damped and it tends to zero without performing any oscillations. The eigenvalues are all real.

2 Implicit equations

Exercise 2

Given the function $f(x) = \cos x - x$, guess a and b such that $f(a)f(b) < 0$. Find the zero(s) of f in $[a, b]$ with 1) the bisection method, 2) the secant method, 3) the regula falsi method. All solutions must be calculated with 8-digit accuracy. Consider the solution given by MATLAB's `fsolve` as the exact result. Which method requires the least number of function evaluations? Report the computational time required by each method.

Fig. 5 is the plot of the function $f(x)$. It is plotted in order to understand graphically how many zeros of the function $f(x)$ lie within a certain window $[a, b]$.
Since the first derivative of $f(x)$, given by Eq. 8, is always non-positive, the function has only one zero that can be found in particular between the values a = 0.5 and b = 0.8. Thus in this exercise these are the values that will be considered as the extremes for all the methods.

$$\frac{df(x)}{dx} = -\sin x - 1 \tag{8}$$

The results are reported in Tab. 1. In the last column the average values of the computational times are written; indeed, since all methods to be performed request computational times that slightly change for each iteration, they have been computed for an adequate amount of times (1000) in order to obtain reasonable mean values. Anyhow, the first fifty values have been discharged from the evaluation of the average ones because the more times the functions are

Figure 5: Function $f(x)$

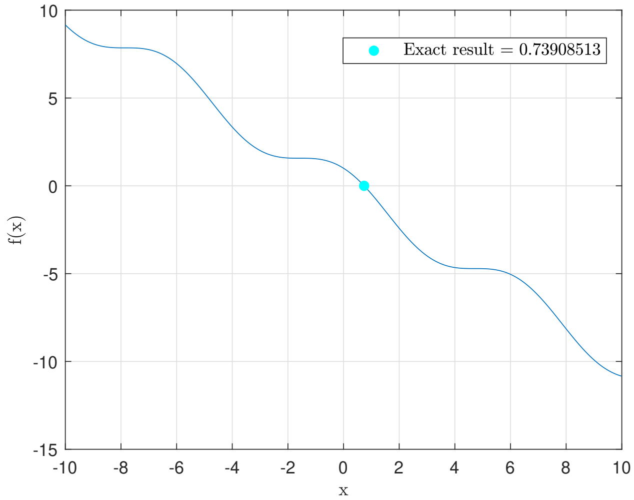


Table 1: Results of the bisection, secant and regula falsi methods

Method	zero	iterations	time [s]
Bisection	0.739085138	25	0.000037
Secant	0.739085133	5	0.000024
Regula Falsi	0.739085133	5	0.000026

run, the faster they get.
The best method according to both the number of iterations and the computational time is the secant one, followed by the regula falsi. Moreover, the bisection method has the slowest convergence rate, it requests the highest number of iterations and also it is not able to reach the right value for the zero when the accuracy is set on eight digits (if the solution is rounded). Also, in order to solve each method loop cycles are used. In all of them, however, only one function evaluation is needed for a single iteration.

Exercise 3

Let $\mathbf{f}$ be a two-dimensional vector-valued function $\mathbf{f}(\mathbf{x}) = (x_1^2 - x_1 - x_2,\, x_1^2/16 + x_2^2 - 1)^\top$, 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?

The results obtained with the different methods are reported in Tab. 2.

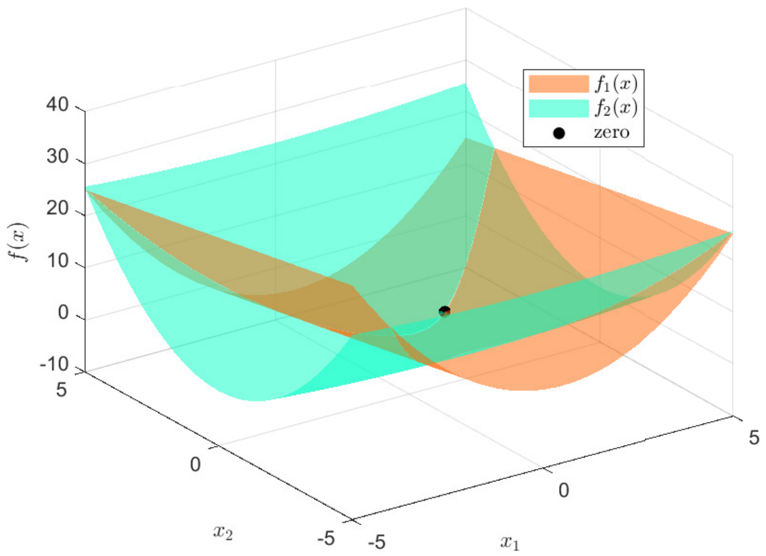
Table 2: Results of the Newton, forward and centered finite differences methods

Method	zero	iterations	time [s]
Analytical computation	[1.5810055; 0.91857299]	4	0.000046821
Forward finite differences	[1.5810055; 0.91857299]	4	0.000070135
Centered finite differences	[1.5810055; 0.91857299]	4	0.000064344

As one can notice, all methods reach the same zero and they request the same number of iter-

ations to find it. However, the computation times are different: the fastest method is achieved through the analytical computation, while the longest with the forward finite differences.
Since the analytical computation is the only one that uses the first derivative of the function in order to achieve the result, while the other two only approximations of it, this first case is the most accurate.

Figure 6: Function $f(x)$

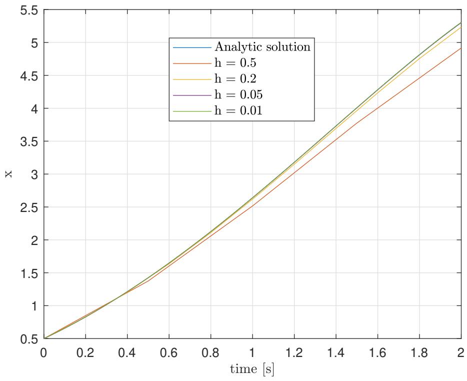


3 Numerical solution of ODE

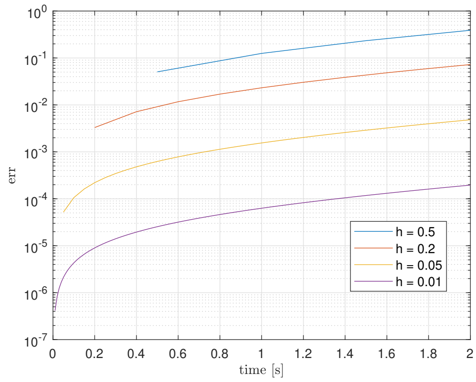
Exercise 4

The Initial Value Problem $\dot{x} = x - t^2 + 1$, $x(0) = \frac{1}{2}$, has analytic solution $x(t) = t^2 + 2t + 1 - \frac{1}{2}e^t$. 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.

In order to solve the exercise two main functions are built, one for the Heun's and the other for the forth order Runge Kutta methods. The graphs shown in Fig. 7a and in Fig. 8a were built by dividing the time window into different intervals for each value of the step size h. Indeed, one can notice that in both methods an increase of h (hence an increase of the number of intervals the time window is divided in) corresponds to a better approximation of the function.
Instead, the graphs related to the errors (Fig. 7b and Fig. 8b) are built by subtracting from the analytic solution $x(t)$ the results obtained respectively with Heun's and 4^{th} order Runge-Kutta methods in the corresponding times (n.b. which number varies with h as already specified).
As shown in Fig. 7b and in Fig. 8b the method that converges with the lowest integration error is 4^{th} order Runge Kutta. Fig. 7b and Fig. 8b are graphs with the vertical axis in logarithmic scale; hence the first points of the functions are not represented since the errors in these cases are equal to zero.
Fig. 9 shows which method should be employed in order to minimise the integration error or

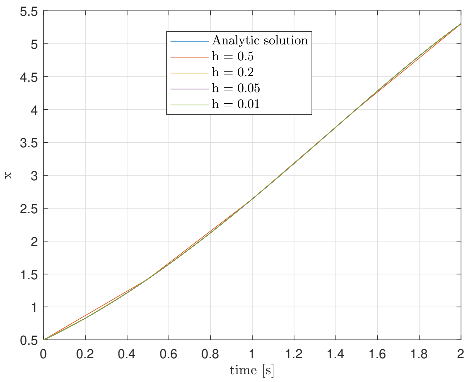


(a) Different values of step size h

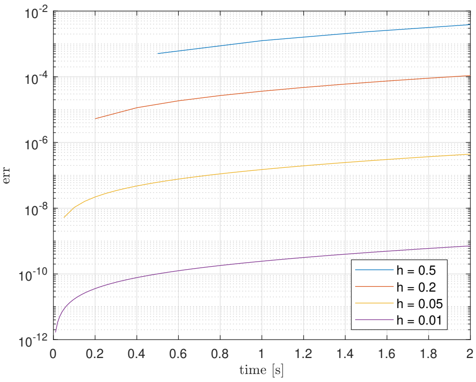


(b) Error

Figure 7: Heun's method



(a) Different values of step size h



(b) Error

Figure 8: Runge Kutta 4^{th} order method

the computational time for each value of the step size h.
In order to minimise the error the 4^{th} order Runge Kutta is better for all values of the step size (see also Tab. 4). However, to minimise the integration time Heun's method should be chosen (see Tab. 3).
Moreover, as in Exercise 2, the functions have been run an adequate number of times (1000 in this case too) in order to obtain the average values of the computational times. In this exercise the first fifty values are discharged as well.
For both methods as h decreases the integration times increase and the errors get lower.

Table 3: Results of Heun's method

h	error	time [s]
0.5	0.3892121849	0.00002819
0.2	0.0724173203	0.00003277
0.05	0.0048198649	0.00006440
0.01	0.0001956223	0.00035111

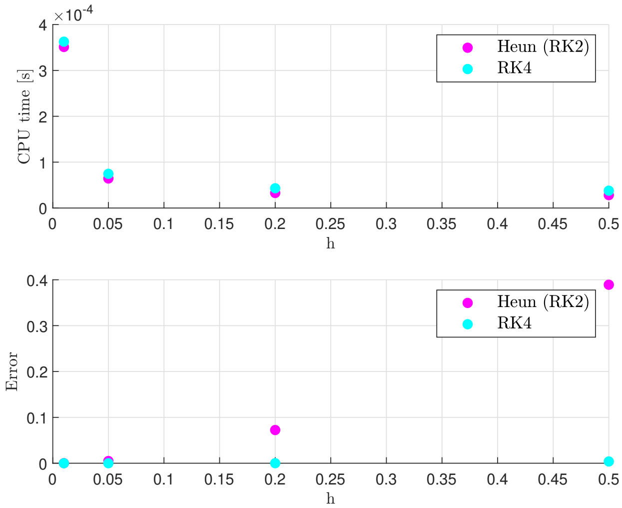


Figure 9: Comparison between RK4 and Heun's methods

Table 4: Results of RK4 method

h	error	time [s]
0.5	0.0038667212	0.00003782
0.2	0.0001089498	0.00004308
0.05	0.0000004421	0.00007460
0.01	0.0000000007	0.00036269

Exercise 5

Let $\dot{\mathbf{x}} = A(\alpha)\mathbf{x}$ be a two-dimensional dynamical 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 counted from the Re$\{\lambda\}$-axis. 1) Write the operator $F_{\text{RK2}}(h, \alpha)$ that maps $\mathbf{x}_k$ into $\mathbf{x}_{k+1}$, namely $\mathbf{x}_{k+1} = F_{\text{RK2}}(h, \alpha)\,\mathbf{x}_k$. 2) Take $\alpha = \pi$ and solve the problem "Find $h \geq 0$ s.t. $|\text{eig}(F(h, \alpha))| = 1$". 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 and represent the points $\{h_i\lambda\}$ of Exercise 4 with $t = 0$. What can you say?

The operators of respectively Heun's and 4^{th} order Runge Kutta are:

$$\text{F}_{\text{RK2}}(h, \alpha) = I + A(\alpha)h + \frac{[A(\alpha)h]^2}{2} \tag{9}$$

$$\text{F}_{\text{RK4}}(h, \alpha) = I + A(\alpha)h + \frac{[A(\alpha)h]^2}{2!} + \frac{[A(\alpha)h]^3}{3!} + \frac{[A(\alpha)h]^4}{4!} \tag{10}$$

where I is the identity matrix.
To analyse the values of h that solve the problems $|\text{eig}(F(h, \alpha))| = 1$ for the two methods the operators in Eq. 9 and Eq. 10 have been exploited.
Firstly, $\alpha = \pi$ is used as requested, obtaining the following results of the step size:

$$\text{h}_{\text{RK2}} = 2.0000147 \tag{11}$$

$$\text{h}_{\text{RK4}} = 2.9067165 \tag{12}$$

Then though the substitution of $\alpha = [0, \pi]$ the graphs of the stability regions of the two methods are drawn; they are reported respectively in Fig. 10 and Fig. 11.

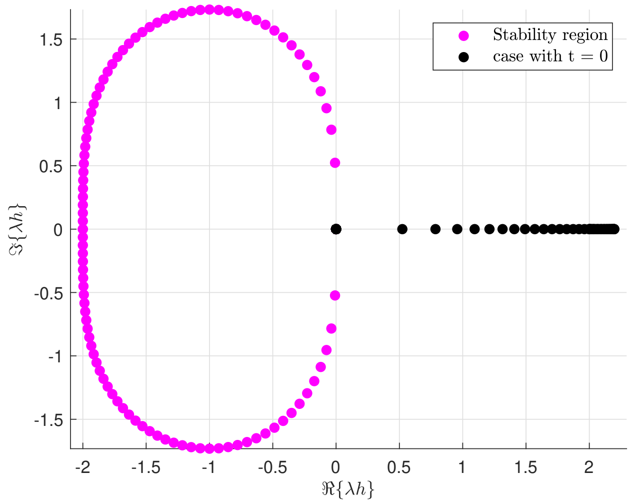


Figure 10: Stability region: Heun's method

For t = 0, the eigenvalues are all unitary and real numbers; they are represented in Fig. 10 and Fig. 11. Since they are all outside the stability regions, they are all unstable.

Exercise 6

Consider the backinterpolation method $\mathrm{BI2}_{0.4}$. 1) Derive the expression of the linear operator $B_{\mathrm{BI2}_{0.4}}(h,\alpha)$ such that $\mathbf{x}_{k+1} = B_{\mathrm{BI2}_{0.4}}(h,\alpha)\mathbf{x}_k$. 2) Following the approach of point 3) in Exercise 5, draw the stability domain of $\mathrm{BI2}_{0.4}$ in the $(h\lambda)$-plane. 3) Derive the domain of numerical stability of $\mathrm{BI2}_\theta$ for the values of $\theta = [0.1,\ 0.3,\ 0.7,\ 0.9]$.

The operator of the second order backinterpolation method is:

$$B_{\mathrm{BI2}_{0.4}}(h,\alpha) = [I - (1-\theta)hA]^{-1}(I + hA\theta) \tag{13}$$

where I is the identity matrix.
This exercise was solved as the previous one, by making the angle α to vary from 0 to π in order to draw the stability regions of the analysed methods. As shown in Fig. 13 as θ increases, the size of the stability region of the relative method shrinks.

Exercise 7

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 $\mathrm{BI2}_{0.1}$; 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 $\mathrm{BI2}_{0.1}$; 5) Discuss the results.

In order to solve the IVP through RK4 with h = 0.1 the same function of the previous exercises is exploited, as well as the one for $\mathrm{BI2}_{0.1}$. Here, the results of the two methods identify two different behaviors.
As shown in Fig. 14, the integration though RK4 does not lead to the solution of the problem; indeed, the function rises exponentially. Instead, with the backinterpolation method the solution can be found, as shown in Fig. 15.
To understand why they behave differently one can notice the different positions of the eigenvalues of the matrix B with respect to the two stability regions of the methods.

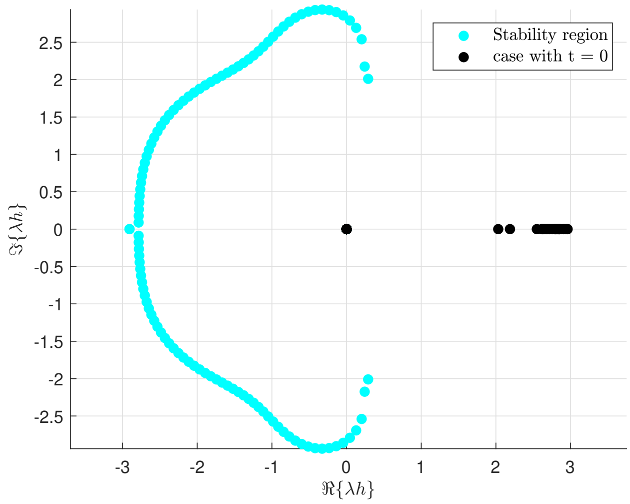


Figure 11: Stability region: 4^{th} order Runge Kutta

For RK4 one eigenvalue of the matrix B is inside the stability region (see Fig. 16), while the other is outside. Instead, for $BI2_{0.1}$ both eigenvalues are outside the circle, hence they are both stable (see Fig. 17).

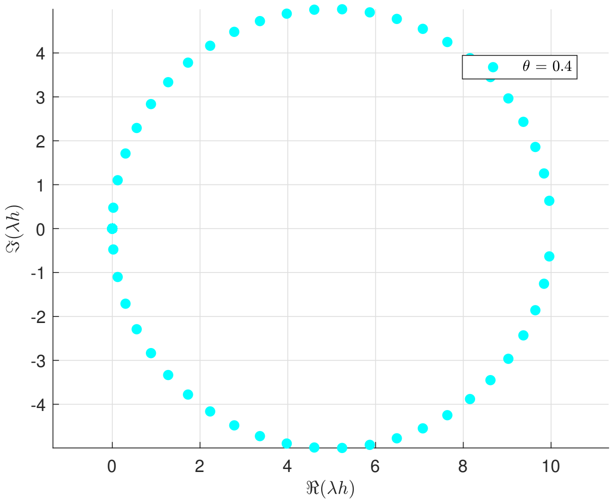


Figure 12: Stability regions for $\mathrm{BI2}_{0.4}$.

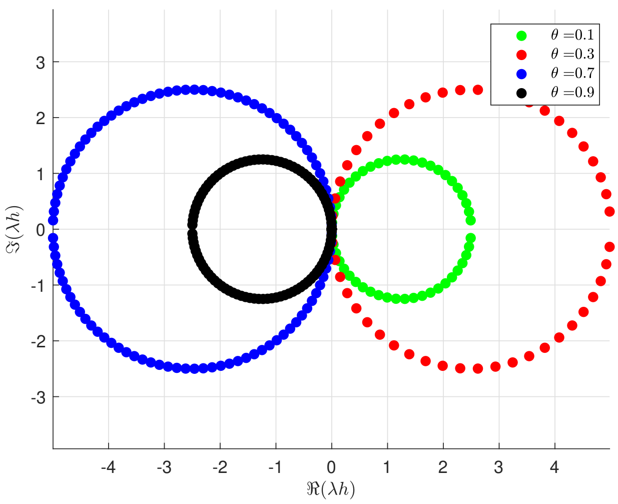


Figure 13: Stability regions for $\mathrm{BI2}_{\theta}$.

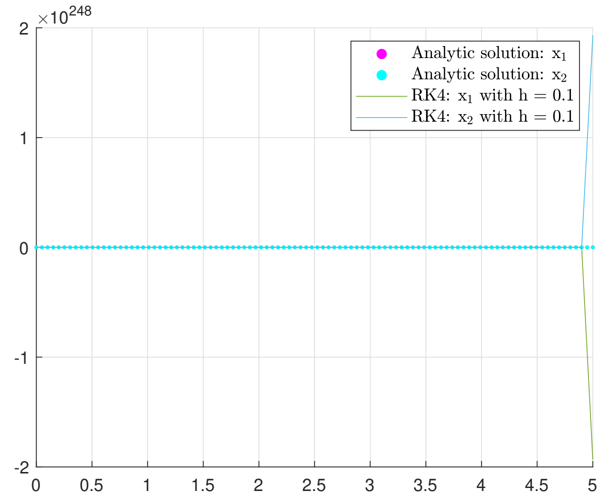


Figure 14: Solution obtained with RK4

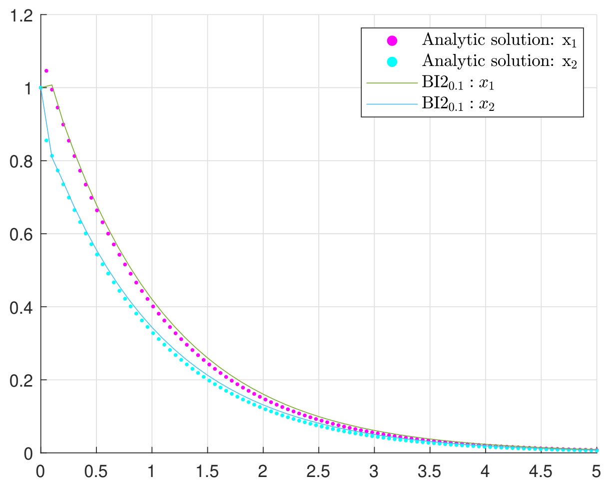


Figure 15: Solution obtained with $BI2_{0.1}$

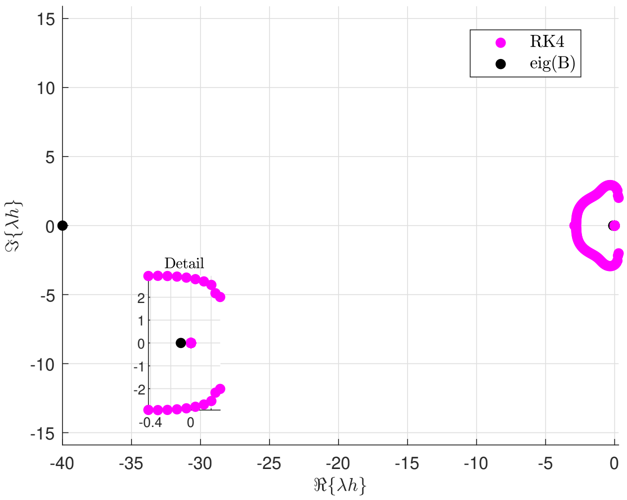


Figure 16: Stability region RK4 and eigenvalues of B

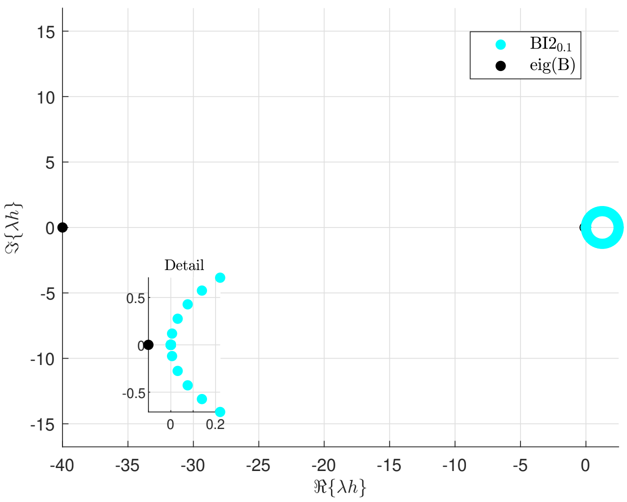


Figure 17: Stability region BI2 and eigenvalues of B