MSAS – Assignment #1: Simulation

Lakrad Adam, 244907

1 Implicit equations

Exercise 1

Consider the physical model of a mechanical system made of four rigid rods $\mathbf{a}_i$ with $i \in [1;4]$ in a kinematic chain, as shown in Fig. 1. The system has only two degrees of freedom. The objective is to determine, fixing the angle β, the corresponding angle of equilibrium α between the rods $\mathbf{a}_1$ and $\mathbf{a}_2$. Writing the kinematic closure, it is possible to obtain the following equation

$$\frac{a_1}{a_2}\cos\beta - \frac{a_1}{a_4}\cos\alpha - \cos(\beta - \alpha) = -\frac{a_1^2 + a_2^2 - a_3^2 + a_4^2}{2a_2a_4} \tag{1}$$

where a_i is the length of the corresponding rod. This equation, called the equation of Freudenstein, can be clearly re-written in the form $f(\alpha) = 0$ and solved, once β is fixed, for a particular value of α. Assume that $a_1 = 10$ cm, $a_2 = 13$ cm, $a_3 = 8$ cm, and $a_4 = 10$ cm.

1) Implement a general-purpose Newton solver (NS).
2) Solve Eq. (1) with NS for $\beta \in [0, \frac{2}{3}\pi]$ with the analytical derivative of $f(\alpha)$ and tolerance set to 10^{-5}. For every β value, use two initial guesses, -0.1 and $\frac{2}{3}\pi$. What do you observe? You can validate your results with the Matlab© built-in function `fzero`.
3) Repeat point 2) using the derivative estimated through finite differences and compare the accuracy of the two methods.
4) Eq. (1) can be solved analytically only for specific values of β. Find at least one and compare the analytical solution with yours.
5) Repeat point 2) extending the β interval up to π. What do you observe?

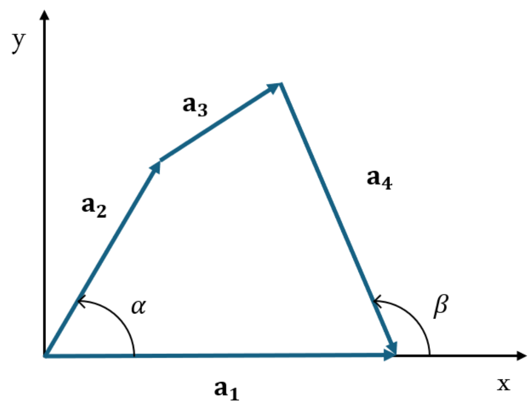

Figure 1: Kinematic chain.

(4 points)

---

1) The Newton method is an iterative numerical method useful for finding approximated solution to the equations of the form $f(x) = 0$. It is particularly suitable to find roots of

non-linear problems. Given a function $f(x)$ and its derivative $\dfrac{df(x)}{dx} = f'(x)$, the Newton method updates the initial guess x_0 to find the equation root, iterating as follows:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2}$$

Each iteration brings ideally closer x_n to the root of $f(x)$, on the condition that the initial guess is sufficiently close to root and that $f'(x_n)$ is non null. The method iterates until a desired tolerance is achieved or a maximum number of iterations is reached.

2) To determine the angle of equilibrium α, fixed the angle $\beta \in [0, \frac{2}{3}\pi]$, as shown in Figure 1, the Newton method is applied, since at first sight the problem seems to be implicit, with a tolerance of 10^{-5} ($|\alpha_i - \alpha_{i-1}|$) and a maximum number of iterations of 10^3. The approximated solution research is rapidly stopped after a maximum of 6 iterations, due to the achievement of the imposed tolerance. The approximated solution $\hat{\alpha}$ is validated with the analytical solution, shown later in point (3). The validation results are reported in Figure 2:

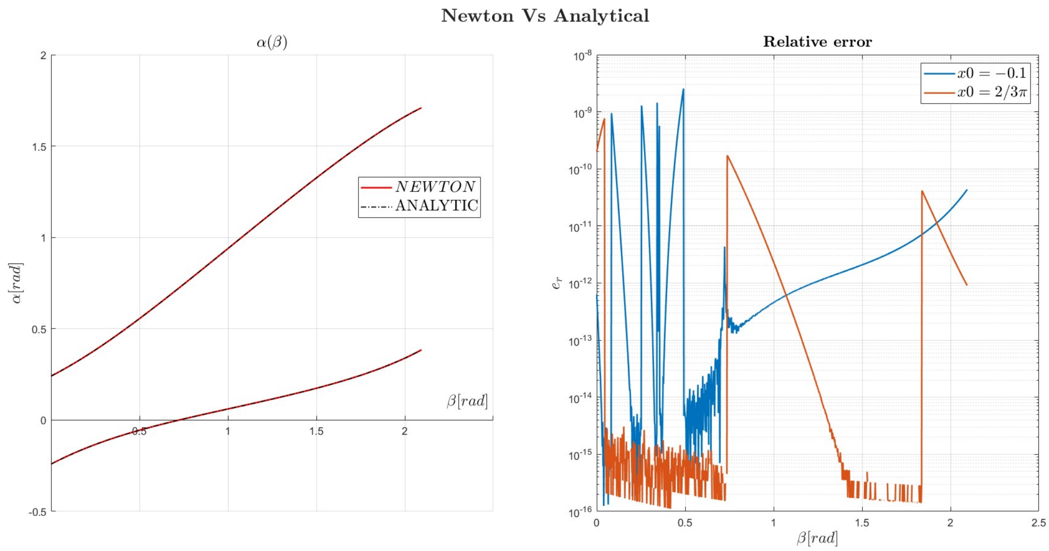

Figure 2: Comparison between Newton method and analytical solution

Having two different initial guesses, $-0.1\ rad$ and $\dfrac{2}{3}\pi\ rad$, is necessary to detect the two distinct possible solutions, corresponding to two different kinematic chain configurations; nevertheless, the Newton method is capable to grasp these two solutions despite the initial guesses are significantly far from the approximated solution. For the β domain considered, the Newton method results are fairly close to the analytical solution, observing a mean relative error in the order of 10^{-11}. It is useful to note that *fzero* delivers similar results, using its default tolerances. Physically speaking, the solution suggests that the for a fixed value of β, the system has two different equilibrium configurations; the initial value of α influences the evolution of the equilibrium configurations.

3) In the previous point the Newton method was based on the analytical derivative:

$$f'(\alpha) = sin(\alpha) + sin(\alpha - \beta) \tag{3}$$

However, it is possible to compute the derivative using the following finite differences schemes:

- forward differences $\rightarrow \quad f'(\alpha) = \dfrac{f(\alpha+h) - f(\alpha)}{h}$
- backward differences $\rightarrow \quad f'(\alpha) = \dfrac{f(\alpha) - f(\alpha-h)}{h}$
- centered differences $\rightarrow \quad f'(\alpha) = \dfrac{f(\alpha+h) - f(\alpha-h)}{2h}$

The centered scheme is second-order accurate whereas the others are first-order accurate. The step size h is chosen carefully to balance between accuracy and numerical stability, as:

$$h = \sqrt{\epsilon} * max(1, |\alpha_0|) \tag{4}$$

where ϵ is the machine epsilon. The accuracy of the Newton method employing the analytical derivative and the ones using the finite difference schemes exposed above is evaluated through the residual, defined as $|f(\hat{\alpha})|$; this definition is preferred over $|f(\hat{\alpha}) - f(\hat{\alpha_{exc}})|$, since it introduces the machine epsilon error, despite analytically α_{exc} is a zero of the function.
As reported in the Table 1, the backward differences scheme appears to be the most accurate, since it turns up to be less sensitive to small oscillations and numerical errors.

	analytical	forward	backward	centered
residual	4.052784e-12	4.062827e-12	4.042597e-12	4.052821e-12

Table 1: Comparison of different derivative computations

4) Equation 1 can be solved analytically, rearranging the equation as follows:

$$[\frac{a_1}{a_4} + cos(\beta)]cos(\alpha) + sin(\beta)sin(\alpha) = \frac{a_1^2 + a_2^2 - a_3^2 + a_4^2}{2a_2a_4} + \frac{a_1}{a_2}cos(\beta) \tag{5}$$

It emerges a non-homogeneous linear trigonometric equation , with the unknown α:

$$A\,cos(\alpha) + B\,sin(\alpha) = C \tag{6}$$

Defining an auxiliary angle $tan(\phi) = \dfrac{B}{A}$, the exact solution can be easily computed:

$$\alpha = \phi \pm acos(cos(\phi)\,\frac{C}{A}) \tag{7}$$

The presence of the $\pm$ grants to determine the two values of α related to the equilibrium configurations for a fixed value of β. The procedure is validated through the comparison with the solution delivered by *fzero*.

5) The extension of β domain up to π allows to catch the physical limit of the system: for certain values of β the rods are not capable of reaching an equilibrium configuration due to the fixed length of the bars and the structural constraints. This behaviour is showed in the following Figure 3:

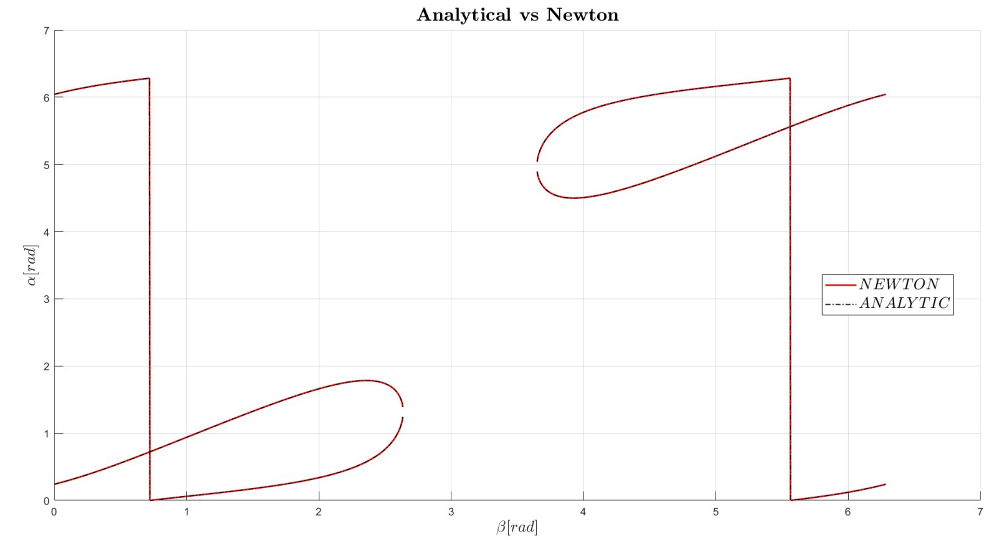

Figure 3: Solution for $\beta \in [0, 2\pi]$

The system shows a symmetric behaviour for $\beta \in [\pi, 2\pi]$. Through a study of the function, by spotting the interval where the analytical method has a complex solution, it possible to define the physical domain of the problem, identified as $\beta \in [0°, 150.92°] \cup [209.09°, 360°]$. Furthermore, the Newton method is able to correctly compute the solution for the physical domain of the problem.
The absence of a physical solution for a certain range of β is also observable through several aspects:

- The argument of arccosine of Equation 7 explodes in correspondence of π, since its neighbourhood do not belong to the physical domain of the problem, as displayed in the plot Figure 4:

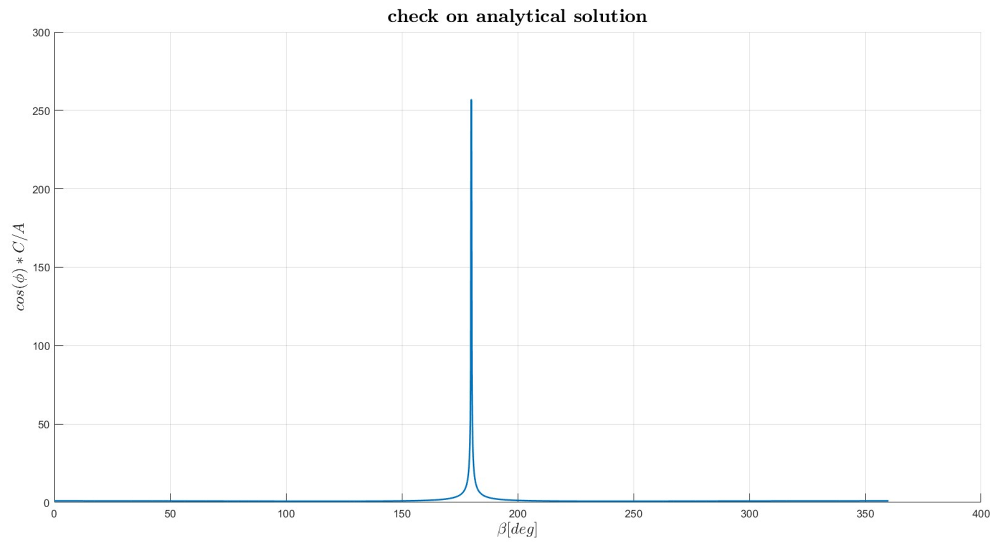

Figure 4: $cos(\phi)\,\dfrac{C}{A}$

- The derivative of the function reaches null values approaching the values of β not belonging to the physical domain, as shown in Figure 5:

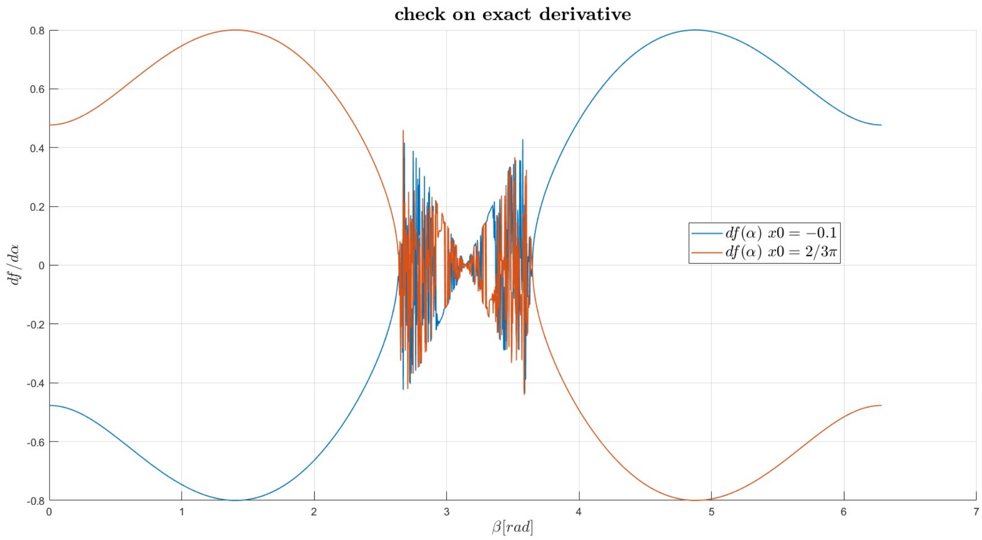

Figure 5: $\dfrac{df}{d\alpha}$

- Both *fzero* and *fsolve* compute accurately the solution, showing, however, a strange behaviour in correspondence of the neighbourhood of π, as shown in Figure 6:

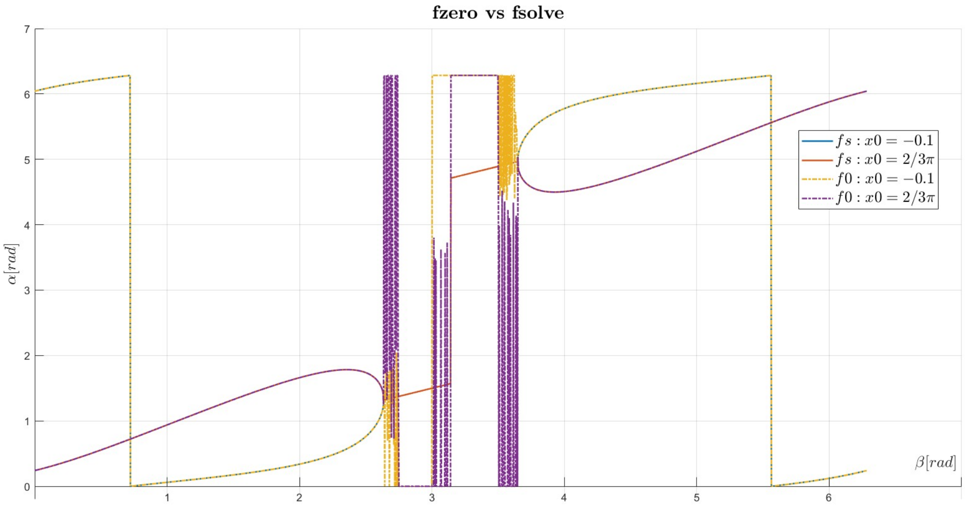

Figure 6: comparison between fzero and fsolve for $\beta \in [0, 2\pi]$

It is reasonable to note that all these aspects are symmetric with respect to π.

2 Numerical solution of ODE

Exercise 2

The physical model of a moving object with decreasing mass is shown in Fig. 10. The mathematical model of the system is expressed by the equations:

$$m(t)\frac{dv}{dt} = \overline{F} + f(t) - \alpha \cdot v \tag{8}$$

$$m(t) = m_0 - c_m \cdot t \tag{9}$$

$$\overline{F} = -\,0.5 \cdot \rho \cdot C_d \cdot A_m \cdot v^2 \tag{10}$$

Assuming that: $v(0) = 0$ [m/s], $m_0 = 20$ [kg], $c_m = 0.1$ [kg/s], $f(t) = 1$ [N], $\alpha = 0.01$ [Ns/m], $\rho = 0$ [kg/m^3], and Eq. (8) has the exact solution:

$$v(t) = \frac{f(t)}{\alpha} - \left[\frac{f(t)}{\alpha} - v(0)\right]\left[1 - \frac{c_m \cdot t}{m_0}\right]^{\frac{\alpha}{c_m}} \tag{11}$$

Answer to the following tasks:

1) Implement a general-purpose, fixed-step Heun's method (RK2);
2) Solve Eq. (8) using RK2 in $t \in [0, 160]$s for $h_1 = 50$, $h_2 = 20$, $h_3 = 10$, $h_4 = 1$ and compare the numerical vs the analytical solution;
3) Repeat points 1)–2) with RK4;
4) Trade off between CPU time & integration error.

(5 points)

---

1) The general purpose Heun method is implemented using the form with predictor and corrector; the solution at each time step relies explicitly on the previous one.

$$\text{predictor:}\quad \begin{aligned}\dot{\mathbf{x}}_k &= f(\mathbf{x}_k, t_k)\\ \mathbf{x}^{\text{P}}_{k+1} &= \mathbf{x}_k + h \cdot \dot{\mathbf{x}}_k\end{aligned}$$

$$\text{corrector:}\quad \begin{aligned}\dot{\mathbf{x}}^{\text{P}}_{k+1} &= f(\mathbf{x}^{\text{P}}_{k+1}, t_{k+1})\\ \mathbf{x}_{k+1} &= \mathbf{x}_k + 0.5 \cdot h \cdot (\dot{\mathbf{x}}_k + \dot{\mathbf{x}}^{\text{P}}_{k+1})\end{aligned}$$

2) The implemented Heun's method ($RK2$) is employed to compute the velocity trend in the given time interval, considering 4 different time steps. The main focus is placed upon the comparison with the exact solution expressed in Equation 11 and on the impact of the variation of the time step, evaluated through the analysis of the relative error, as shown in Figure 7:

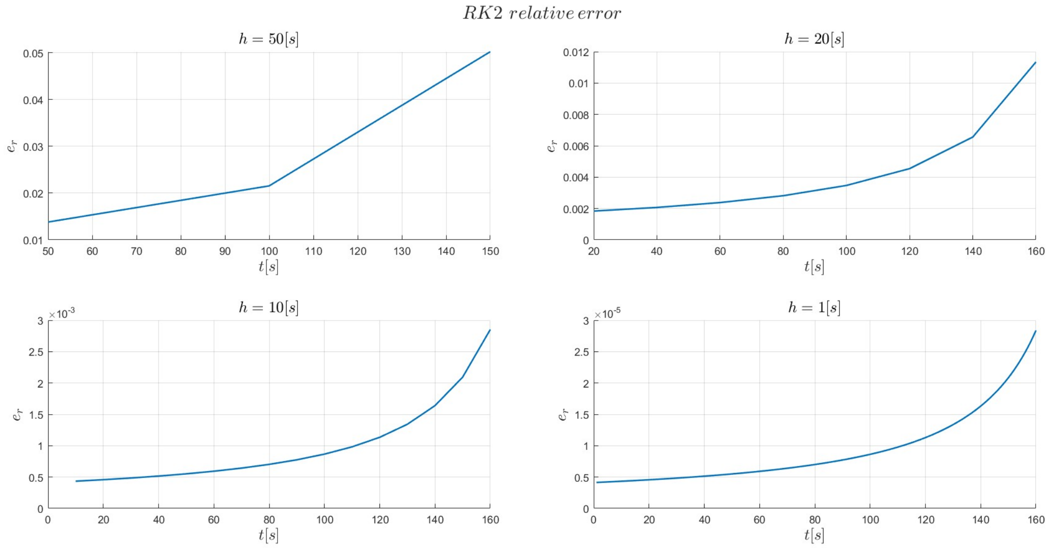

Figure 7: RK2 relative error

As expected, the relative error is larger for increasing time step. Correctly the method using the smaller time step is the most accurate; nevertheless, also the other methods are able to catch the stable behaviour of the system. In addition, the problem is solvable only for a finite time interval ($200\,s$), explainable through the decrease of the mass and the consequent increase of the velocity until it diverges of null mass.

3) The problem is solved through RK4, implemented as follows:

$$\begin{aligned}
k_1 &= f(\mathbf{x}_k, t_k) \\
k_2 &= f\left(\mathbf{x}_k + \frac{h}{2}k_1, t_k + \frac{h}{2}\right) \\
k_3 &= f\left(\mathbf{x}_k + \frac{h}{2}k_2, t_k + \frac{h}{2}\right) \\
k_4 &= f(\mathbf{x}_k + hk_3, t_k + h) \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

The comparison with the exact solution is shown in Figure 8:

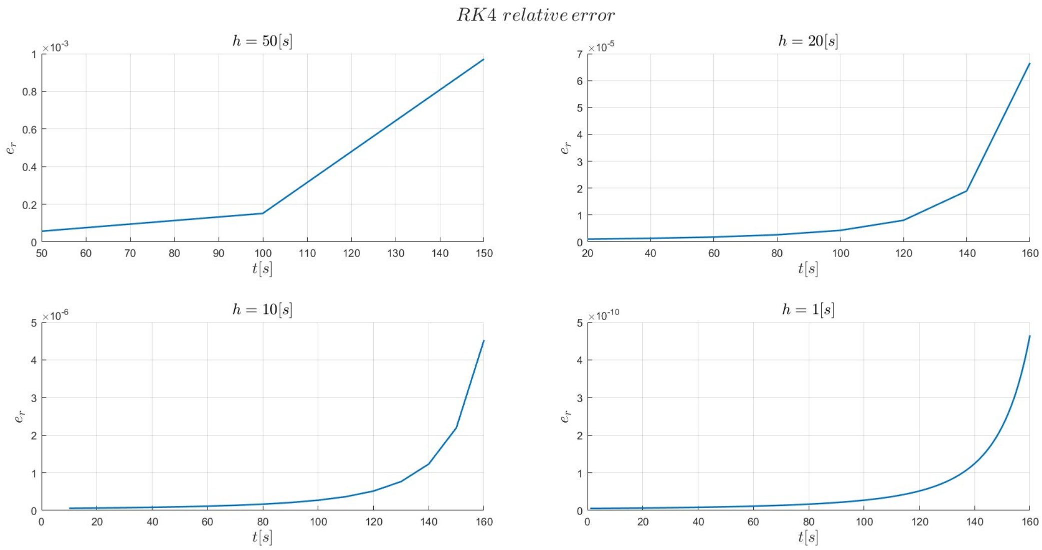

Figure 8: RK4 relative error

Due to its 4^{th} order, RK4 is more accurate than RK2 with a relative error approximately equal to the square of the one of RK2. Both methods clearly grasp the problem's solution as shown in Figure 9, where the velocity monotonically grows, due to the constant reduction of the mass and the null density.

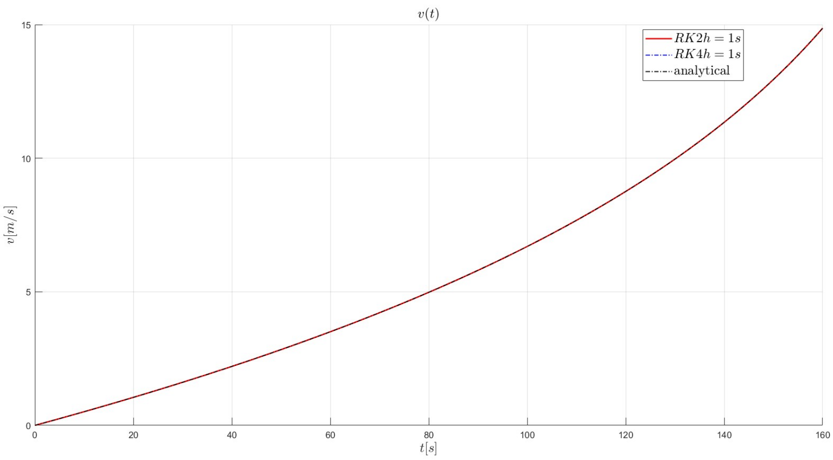

Figure 9: v(t) solution comparison between RK2, RK4 and exact solution

4) To realize a fair trade off between RK2 and RK4, the CPU time has to be analyzed:

	RK2				**RK4**			
h[s]	1	10	20	50	1	10	20	50
$\bar{t}_{CPU}$ [s]	5.14e-05	8.17e-06	5.59e-06	4.19e-06	7.03e-05	9.97e-06	6.86e-06	4.86e-06
σ [s]	5.95e-06	1.71e-06	1.33e-06	1.14e-06	7.99e-06	1.72e-06	1.74e-06	1.69e-06

Table 2: $RK2$ & $RK4$ CPU time

The CPU time is computed through the use of Matlab "tic-toc" function, running each

RK code for 10^5 times,in order to dump the main oscillations. Despite the higher accuracy, RK4 is more computational demanding since it requires 4 function evaluations, compared to the 2 required by RK2.

Now, assuming that: the fluid density is $\rho = 900$ [kg/m^3], $C_d = 2.05$, and $A_m = 1$ [m^2], answer to the following tasks:

1) Solve Eq. (8) using RK2 in $t \in [0, 160]$s for $h_1 = 1$;
2) From the results of the previous point, use a proper ode of Matlab to solve Eq. (8) ;
3) Discuss the results.

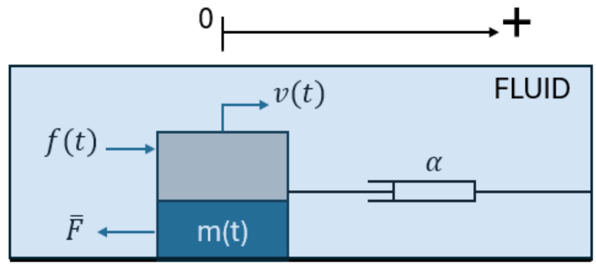

Figure 10: Moving object with decreasing mass.

(5 points)

---

1) Considering a non null fluid density, the non linearity of the problem emerges, posing particular focus on the choice of the time step. The results are shown in Figure 11:

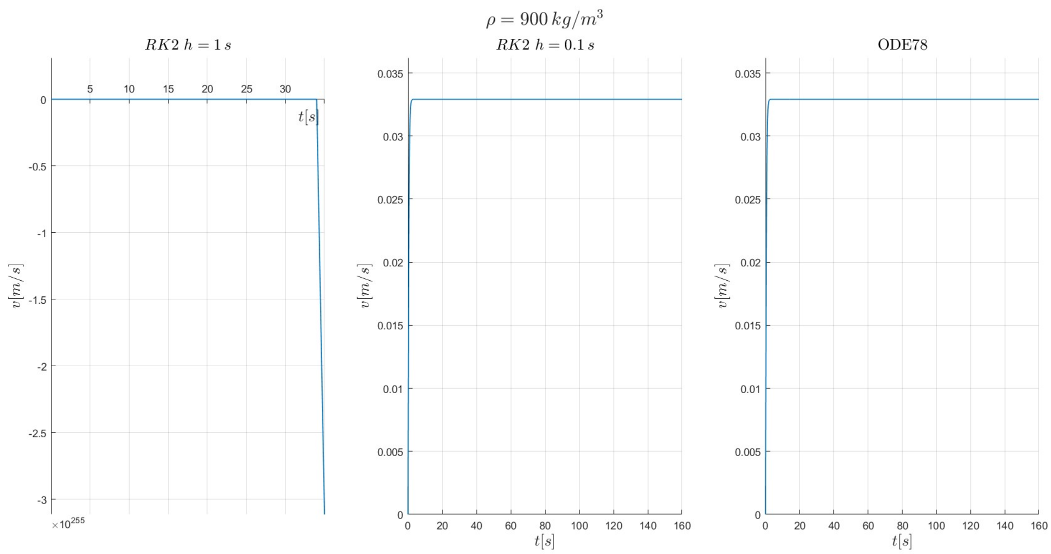

Figure 11: $v(t)$ with RK2 h=0.1 , h=1 and ODE78

Clearly the choice of $h = 1\,s$ leads to an unstable discrete mapping of the continuous system; indeed, due to the explicit nature of RK2, the chosen time step requires a reduction in order to map correctly the stable continuous time system into a stable discrete one.

2) The system can be simply solved using ODE78, which offers for non stiff systems the perfect compromise between computational efficiency and accuracy. A fixed time step of $0.1\,s$ is chosen for ODE78 to properly compare the results with RK2. As expected, ODE78 does not suffer from numerical instability problems and obtains a velocity evolution coherent with the system, where the solution suggests that after a rapid transient, a state of constant velocity is reached, where the forces are dynamically balanced. The velocity is considered constant because the average difference between successive iterations is around 10^{-5} m/s, compared to a constant velocity of 3.3×10^{-2} m/s.

3) As previously reported in point 1), the time span chosen is not enough small to properly describe the system; as shown in Figure 11 a time step of $0.1\,s$ is enough to map correctly the system in discrete time system and get a fairly accurate solution, with a mean absolute error with respect to ODE78 in the order of $10^{-6}\,m/s$. The maximum time step strictly depends on the time span considered, since the mass of the object decreases in time; for $t \in [0, 160]\,s$, the maximum time step is around $0.19\,s$. The maximum time step can be slightly increased up to $0.25\,s$, using RK4, being an higher order scheme that expands RK2 stability region .

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.

a) Runge-Kutta methods

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)

---

1) The Runge-Kutta methods are a family of iterative techniques used for numerically solving ordinary differential equations. They are based on evaluating the derivative at multiple points within each step and combining these evaluations to improve the accuracy of the approximation. The methods vary in order and complexity, with higher-order methods generally providing more accurate results with larger step sizes. Runge-Kutta methods offer a valid balance between computational efficiency and accuracy. The special case of 2^{nd} order Runge-Kutta method is Heun's method [2]:

$$\mathbf{x}_{k+1}^{P} = \mathbf{x}_k + h\,\mathbf{f}(\mathbf{x}_k, t_k) \tag{12}$$

$$\mathbf{x}_{k+1}^{C} = \mathbf{x}_k + \frac{h}{2}[\mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}(\mathbf{x^P}_{k+1}, t_{k+1})] \tag{13}$$

The Heun's operator, for linear systems, is obtained by substituting the predictor in the corrector:

$$\mathbf{F}_{RK2} = \mathbf{I} + \mathbf{A} \cdot h + \frac{(\mathbf{A} \cdot h)^2}{2!} \tag{14}$$

2) The given $\mathbf{A}$ is a matrix with a pair of conjugate eigenvalues on the unit circle in function of α. The given request is based on the research of the time step associated to marginal stability for the eigenvalue $\lambda = -1 + 0i$. Using the symbolic computation, the largest time step is equal to $h = 2$, as stated by the theory.

3) Extending this procedure to $\alpha \in [0, \pi]$, it is possible to compute the stability domain of RK2. Computationally this is done by using *fzero*, using as guess the previous solution. The stability domain is reported in Figure 12:

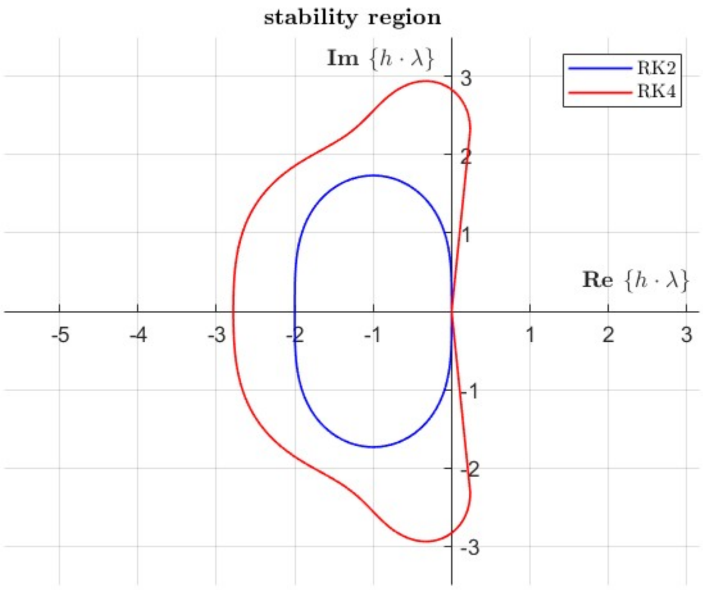

Figure 12: RK2 and RK4 stability domain

The stability domain is within the blue curve, indeed as usual for explicit methods the stability domain is within the curve found with method reported, in the $(h \cdot \lambda)$-plane.

4) RK4 is the most popular Runge-Kutta method, based on the following steps:

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_k, t_k) \tag{15}$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{x}_k + \frac{h}{2} \cdot \mathbf{k}_1,\, t_k + \frac{h}{2}) \tag{16}$$

$$\mathbf{k}_3 = \mathbf{f}(\mathbf{x}_k + \frac{h}{2} \cdot \mathbf{k}_2,\, t_k + \frac{h}{2}) \tag{17}$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{x}_k + h \cdot \mathbf{k}_3,\, t_k + h) \tag{18}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{6} \cdot [\mathbf{k}_1 + 2 \cdot \mathbf{k}_2 + 2 \cdot \mathbf{k}_3 + \mathbf{k}_4] \tag{19}$$

RK4 operator is obtained by enlarging the RK2 operator, adding terms up to the 4^{th} order:

$$\mathbf{F}_{RK4} = \mathbf{I} + \mathbf{A} \cdot h + \frac{(\mathbf{A} \cdot h)^2}{2!} + \frac{(\mathbf{A} \cdot h)^3}{3!} + \frac{(\mathbf{A} \cdot h)^4}{4!} \tag{20}$$

Using the procedure showed previously, it is possible to determine RK4 stability domain, depicted in Figure 12. Being an higher order method, RK4 enlarges the stability domain compared to RK2, following better and better the imaginary axis; however it presents a portion of the positive right semi-plane, which is an issue since it could lead to the mapping of an unstable time continuous system into a time discrete stable one. In addition, the curves defining these two lobes seem straight, even tough they should not, due to lack of solutions when approaching the origin.

---

b) Backinterpolation methods

Consider the backinterpolation method $\mathrm{BI2}_{0.3}$.
1) Derive the expression of the linear operator $B_{\mathrm{BI2}_{0.3}}(h, \alpha)$ such that $\mathbf{x}_{k+1} = B_{\mathrm{BI2}_{0.3}}(h, \alpha)\mathbf{x}_k$.
2) Using the same approach of a), draw the stability domain of $\mathrm{BI2}_{0.3}$ in the $(h\lambda)$-plane.
3) Derive the domain of numerical stability of $\mathrm{BI2}_\theta$ for the values of $\theta = [0.2, 0.4, 0.6, 0.8]$.

(5 points)

---

1) Backinterpolation method is a cyclic method based on the division of the time step in two fractions ($\theta\, h$ and $(1-\theta\, h)$), where the first one is solved through an explicit method whereas the second with an implicit one, the solution $\mathbf{x}_{k+1}$ allows to get the same solution approaching with both the explicit and implicit method. Such technique is called θ-method [1]. For backinterpolation methods it is possible to choose the order and the value of θ. Following this rationale, it is possible to derive the expression of the linear operator:

$$\mathbf{F}_{BI2} = [\mathbf{I} - \mathbf{A}(1-\theta)\cdot h + \frac{(\mathbf{A}(1-\theta)\cdot h)^2}{2!}]^{-1} \cdot [\mathbf{I} + \mathbf{A}\theta \cdot h + \frac{(\mathbf{A}\theta \cdot h)^2}{2!}] \tag{21}$$

2) Using the same approach used for Runge-Kutta methods, it is possible to draw the stability domain of $\mathrm{BI2}_{0.3}$ in the $(h\lambda)$-plane.

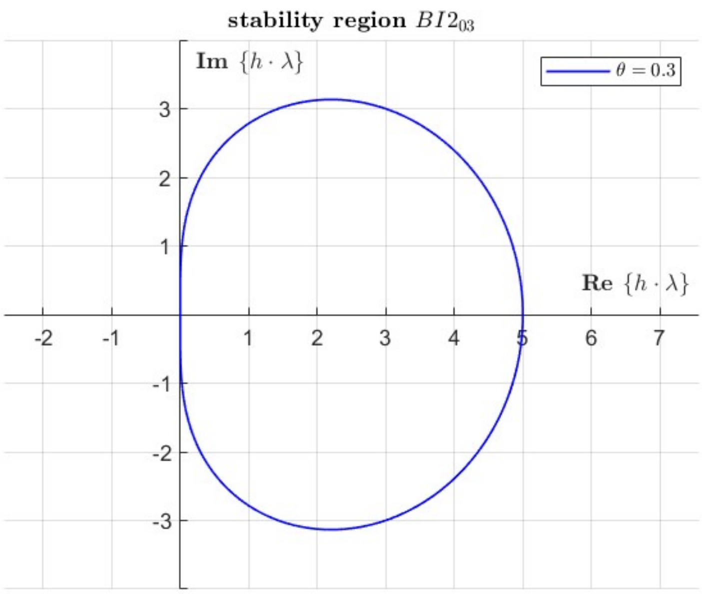

Figure 13: $\mathrm{BI2}_{0.3}$ stability domain

As shown in Figure 13, for $\theta = 0.3$, the method is A-stable and the unstable domain is inside the blue curve; as consequence, the method seems a suitable choice for stiff systems.

3) Following the same procedure, the stability domain of BI2 can be computed for different values of θ, in order to visualize a particular behaviour of the method.

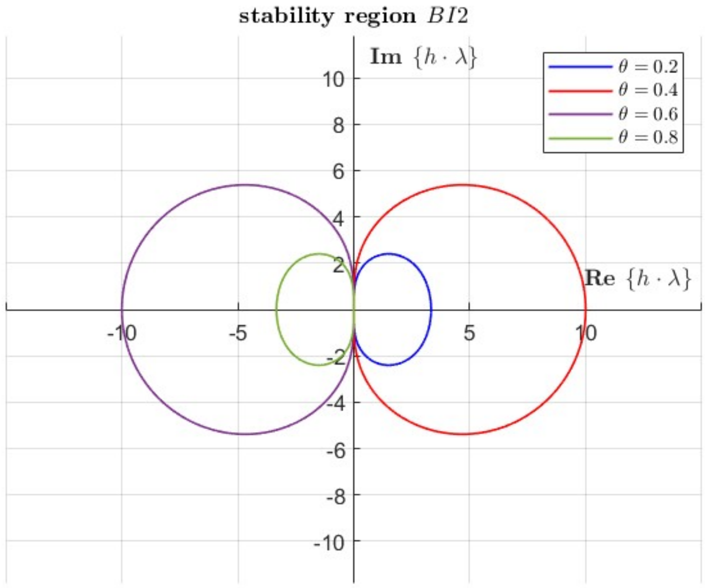

Figure 14: BI2 stability domain for $\theta = [0.2; 0.4; 0.6; 0.8]$

- $\theta < 0.5$, the backinterpolation method has a predominant implicit shade, indeed the method is A-stable and method is *stable outside* the relative curve, with increasing unstable domain as θ grows; for $\theta = 0$ BI2 coincides with Backward Runge-Kutta 2.

- $\theta > 0.5$, the explicit character rules, leading the method to be *stable inside* the relative curve, with decreasing stable domain as θ grows, to the extent where for $\theta = 1$ BI2 coincides with Runge-Kutta 2.

- $\theta = 0.5$, the curve coincides with the imaginary $(h\,\lambda)$-axis and the method is F-stable, since it contains the entire left half $(h\,\lambda)$–plane and nothing but the left half $(h\,\lambda)$–plane and as the negative real axis is approached toward negative infinity, stability progressively decreases until stability is entirely lost at negative infinity [1].

Exercise 4

The cooling problem of a high-temperature mass is shown in Fig. 15. Assuming that: K_c= 0.0042 [J/(s K)] and K_r = 6.15 x 10^{-11} [J/(s K^4)] are the convective and radiation heat loss coefficients respectively, $\widetilde{T}$ is the mass temperature in time, T_a = 277 [K] is the surrounding air temperature, C= 45 [J/K] is the mass thermal capacity, and $\widetilde{T}(0)$ = 555 [K] is the initial mass temperature.

1) Write the mathematical model of the system.

2) Solve the mathematical model using: RK2 with h= 720, and RK4 with h= 1440.

3) Compare the solution of point 2) with the "exact" one. Discuss the results. (Hint: use an ODE solver of Matlab).

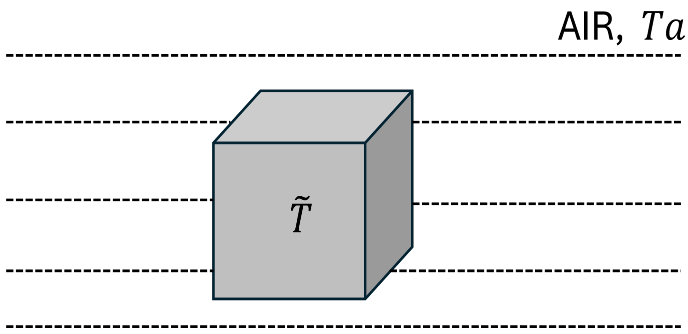

Figure 15: The cooling of high-temperature mass.

(5 points)

1) Considering thermal equilibrium within the mass, the conductive heat transfer is neglected, focusing therefore on the radiative and convective heat transfer with the surrounding air. The physical model can be observed through the following electric resistance analogy:

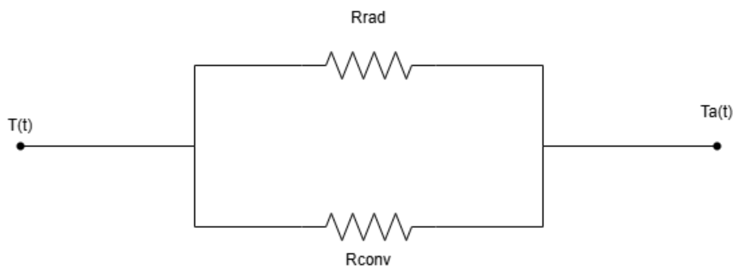

Figure 16: Electrical analogy scheme

Assuming constant heat loss coefficients, mass thermal properties and a constant air temperature, the mathematical model can be extrapolated, starting from an energy balance balance:

$$\frac{\partial U}{\partial t} = -\sum \dot{Q}^{\rightarrow} \tag{22}$$

$$\frac{\partial T}{\partial t} = \frac{K_c}{C}(T_a - T) + \frac{K_r}{C}(T_a^4 - T^4) \tag{23}$$

The non linearity of the problem caused by the presence of the 4^{th} power of the unknown temperature, does not consent to have a close form solution, therefore a numerical method has to be employed to solve the problem and to track the mass temperature variation in time.

2) The mathematical model is solved using both Runge-Kutta 2 and 4 methods, with the time step for RK4 being twice that of RK2. The time span considered is $t_{span} = 100\, h_{RK2} = 72000\, s$, helding 100 points for RK2 and 50 points for RK4. The results are reported in the following Figure 17:

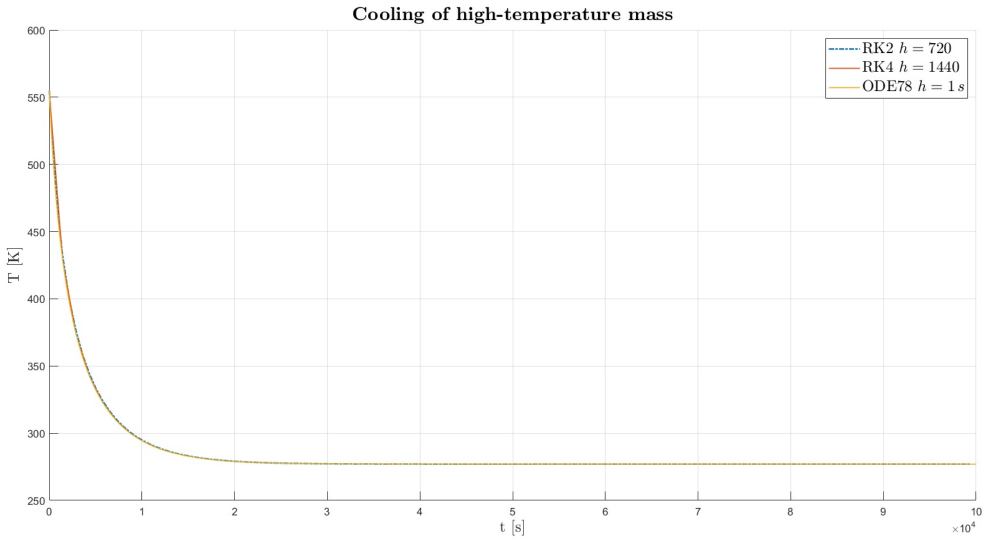

Figure 17: Mass temperature trend, using RK2,RK4 and ODE78

As the physical model suggests, the mass temperature tends to reach the thermal equilibrium with the surrounding air, indeed both the methods reaches from above $T_a = 277\ K$, up to the 4^{th} decimal.

3) The RK solutions are compared with Matlab built-in function ODE78, with $h_{ODE} = h_{RK2} = 720\,s$; the temperature trend computed with ODE78 is reported in Figure 17. The RK error, taking ODE78 as 'exact' solution, is reported in Figure 18:

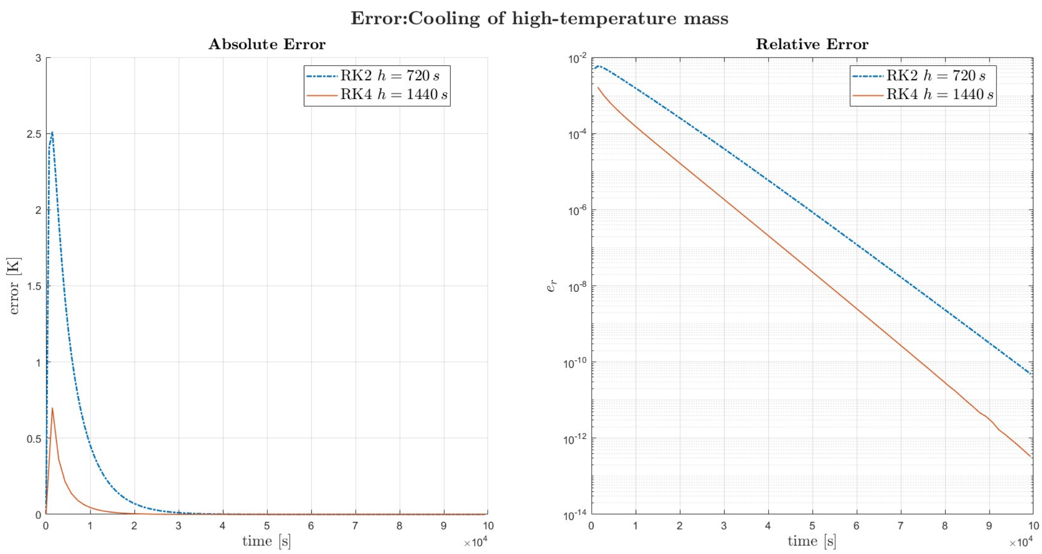

Figure 18: Absolute and relative error of RK2 and RK4, with respect to ODE78

As anticipated, RK4 is more accurate than RK2 due to its higher-order scheme; however, no numerical instability issues were observed. For a broader comparison of these two methods applied to this problem, Table 3 presents the mean relative error and CPU time for both RK2 and RK4, calculated using Matlab's `tic-toc`.

	RK2	RK4
e_{rel}	6.048e-04	9.203e-05
$t_{CPU}[s]$	2.762e-03	4.292e-03

Table 3: RK2 and RK4 mean relative error and CPU time

As theory confirms, RK4 is more accurate than RK2, at the price of a greater computational effort, even with different time steps. These differences would be more pronounced if both methods used the same time step.

Exercise 5

Consider the electrical circuit in Fig. 19. At time t>=0 the switch disconnects the battery and the capacitor discharges its stored energy to the circuit. Answer to the following tasks:

1) Find the state variable form of the circuit with x_1=q and x_2=$\frac{dq}{dt}$, where q is the charge on capacitor.

2) Show that the system is stiff when the circuit parameters are R=25 [Ω], L = 20 [mH], C = 200 [mF], and v_0 = 12 [V]. Represent the eigenvalues on the ($h\lambda$)-plane both for RK2 and IEX4 stability domain.

3) Use IEX4 to simulate the transient response of the system with the parameters of point 2), and determine the largest step size such that RK2 yields a stable and accurate solution.

4) Discuss the results.

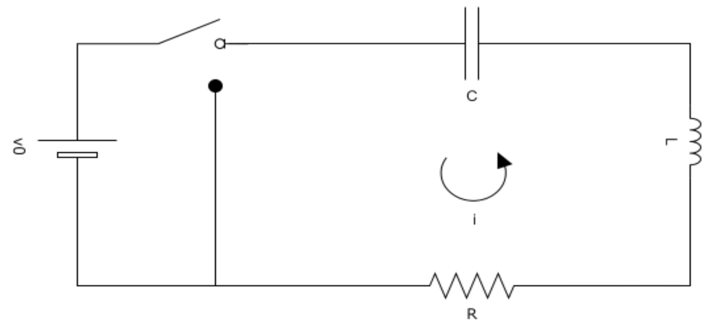

Figure 19: Electric circuit model.

(5 points)

---

1) To derive the state variable form of the RLC circuit, a voltage balance must be applied within the circuit during condenser discharge, where the battery is disconnected and the circuit is driven by the condenser. The voltage drops across the elements must satisfy Kirchhoff's Voltage Law (KVL):

$$v_C + v_R + v_L = 0 \tag{24}$$

where:

- v_C=$\dfrac{q}{C}$, voltage across the capacitor.

- $v_R = R\,i$, voltage across the resistor.
- $v_L = L\dfrac{di}{dt}$, voltage across the inductor.

Being the current the time derivative of the charge ($i = \dfrac{dq}{dt}$), the state space form of the system is:

$$\begin{bmatrix} \dot{q} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} q \\ i \end{bmatrix} \tag{25}$$

Written in matrix form as $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$.
The initial condition of the system is $q(t=0) = v(t=0)\,C$, $i(t=0) = 0$, since at the last instant before the capacitor discharge, no charge is circulating and the charge stored in the capacitor is simply related to the voltage across it and to the capacitance.

2) According to [1], stiff systems are characterized by eigenvalues with negative real part and large stiffness ratio, different characteristic time scales and step-size constraint dictated by stability requirements rather than accuracy. To solve the problem it is suggested to use RK2, explicit method, and IEX4, implicit A-stable extrapolation technique of order 4. Inserting the given circuit parameters, it is possible to represent the eigenvalues of the system on the (hλ) with RK2 and IEX4 stability domains. To compute the domain of a numerical model, using the procedure shown in Exercise 3 (2), it is necessary to derive the numerical method operator $F(h,\alpha)$ such that $\mathbf{x}_{k+1} = F(h,\alpha)\,\mathbf{x}_k$. The novelty lies on the extrapolation of IEX4 operator $F_{IEX4}(h,\alpha)$ and on the representation of its stability domain on the $(h\lambda)$-plane. Through the substitution of the 4 predictors in the corrector, for a linear system, it is possible to derive:

$$\mathbf{F} = -\frac{1}{6}\cdot[\mathbf{I} - \mathbf{A}\cdot h]^{-1} + 4\cdot[\mathbf{I} - \frac{\mathbf{A}\cdot h}{2}]^{-2} - \frac{27}{2}\cdot[\mathbf{I} - \frac{\mathbf{A}\cdot h}{3}]^{-3} + \frac{32}{3}\cdot[\mathbf{I} - \frac{\mathbf{A}\cdot h}{4}]^{-4} \tag{26}$$

The eigenvalues are then plotted on the $(h\lambda)$-plane for both RK2 and IEX4 stability domain:

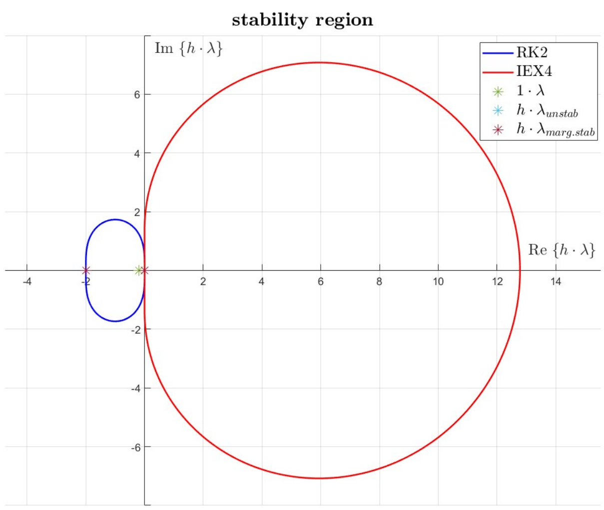

Figure 20: IEX4 and RK2 stability domain

The RLC matrix eigenvalues are $\lambda_1 = -0.2$ and $\lambda_2 = -1250$, where the latter is not represented in Figure 20, due to its larger order of magnitude, underlying the stiffness of the system. As previously mentioned, RK2 is an explicit method with a stability region inside the blue curve, while IEX4 is an A-stable implicit method with a stability region outside the red curve. It is interesting to point out that for the representation of IEX4 stability region, the initial guess taken, corresponding to $\alpha = \pi$, is $h = -12$, since the minus is necessary to collocate the point, associated to $\lambda_A(\pi) = -1$, in the positive real axis.

3) Due to the linearity of the system, IEX4 method can be implemented exploiting the numerical method operator F_{IEX4} and thanks to the A-stability of the method, the stability of the method can be caught for every positive time step. In order to yield a stable and accurate solution, RK2 largest time step is computed as maximum value of h such that the largest magnitude eigenvalue is placed on the left extreme of the stability region:

$$h_{max} > 0 \,|\; h_{max}\lambda_{max} = -0.2 \rightarrow h_{max} < 0.0016 = h_{marg.stab} \tag{27}$$

However, the largest step size should be lower than 0.0016, since this value should ensure marginal stability and so with a slightly lower stepsize, as $h_{stab} = 99\%\, h_{marg.stab}$, a stable and accurate solution is computed using RK2. The solutions using IEX4($h = h_{stab}$) and RK2($h = h_{stab}$,$h = h_{marg.stab}$) are reported in Figure 21:

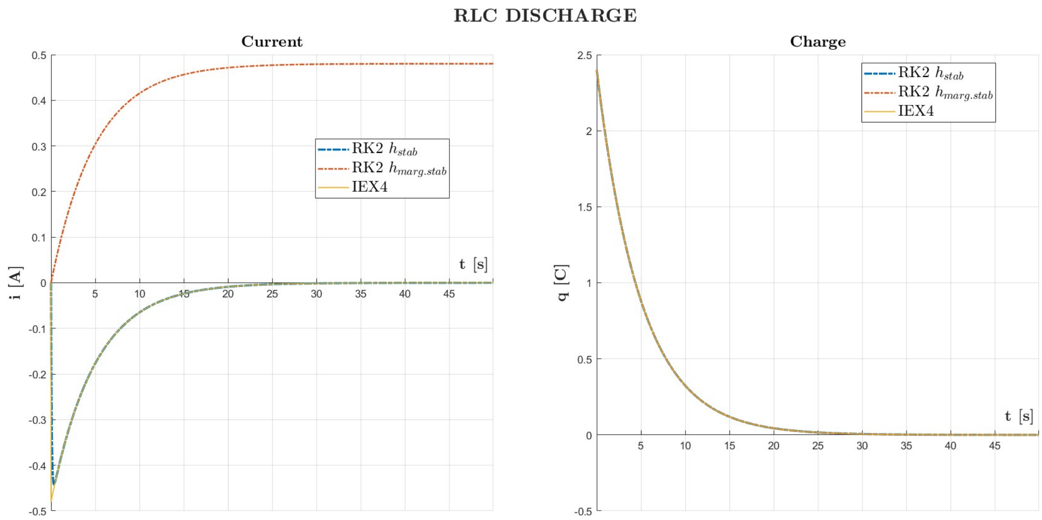

Figure 21: RLC circuit transient analysis using RK2,IEX4 and ODE23s

4) No issues are evident in the charge trend, but significant differences appear in the current plot, noting that the current is the time derivative of the charge. Although theoretically $h_{\text{marg.stab}}$ should ensure marginal stability, it is clear that this is not the case, as the current eventually reaches $\frac{v_0}{R} = 0.48\,A$, after the transient.

Using a time step of $h_{\text{stab}} = 99\%\,h_{\text{marg.stab}}$, RK2 solves the problem more stably and accurately. By decreasing the step size, RK2 more precisely captures the initial current spike to $-\frac{v_0}{R} = -0.48\,A$.

The system's stiffness is also evident in the current discharge transient, where two distinct characteristic time scales are observed: the first relates to reaching steady-state conditions, initially viewing the condenser as a constant voltage generator, while the second relates to the actual discharge of the condenser, leading to zero current and charge in the circuit. Instead IEX4 is capable to compute properly the solutions, regardless of the positive time step chosen, thanks to its A-stability; the time step should be chosen to have a proper balance between accuracy and computational time. On the following Figure 22 the relative error of RK2, using $h = h_{stab}$, with respect to IEX4, with the same time step, is reported:

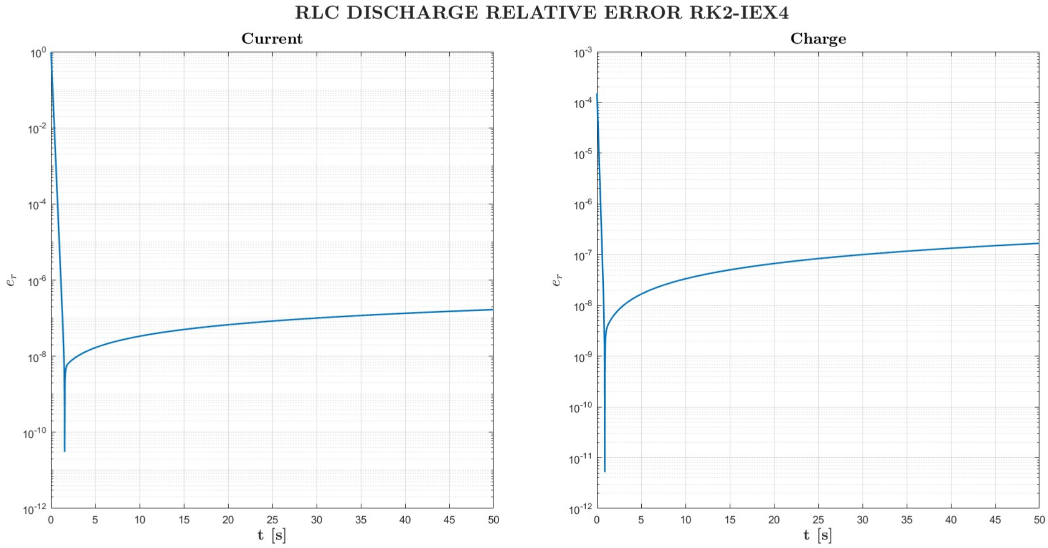

Figure 22: RK2 relative error, with IEX4 as reference

The primary source of error is due to RK2's difficulty in accurately capturing the initial current peak. This can be mitigated by reducing the time step, though it increases computational effort.

Exercise 6

Consider the bouncing ball physical model in Fig. 23. The mathematical model of the system is represented by the following equations:

$$\frac{dv}{dt} = -g + \frac{F_D}{m_b} + \frac{F_A}{m_b} \tag{28}$$

$$\frac{dx}{dt} = v \tag{29}$$

$$v^+ = -k \cdot v^- \tag{30}$$

$$F_D = -\ 0.5 \cdot \rho \cdot C_d \cdot A_b \cdot v \cdot |v| \tag{31}$$

$$F_A = \rho \cdot V_b \cdot g \tag{32}$$

where the initial conditions are: $v(0) = 0$ [m/s], and $x(0) = 10$ [m]. Assuming that: the ball velocity before (v^-) and after (v^+) the ground impact is governed by the attenuation factor $k = 0.9$, the air density is $\rho = 1.225$ [kg/m^3], the ball mass is $m_b = 1$ [kg], the ball area is $A_b = 0.07$ [m^2], the ball volume is $V_b = 0.014$ [m^3], and the drag coefficient is C_d=1.17, answer to the following tasks:

1) Solve the mathematical model in the time interval $t \in [0, 10)$s using an ode integrator of Matlab.

2) Repeat point 1) for ρ=15 [kg/m^3]. Which is the difference with respect to point 1)?

3) Now solve the problem when the fluid density is ρ=60 [kg/m^3] in $t \in [0, 10)$s. Discuss the result.

4) Repeat point 3) using RK4 for $h = 3$. Discuss the result.

(5 points)

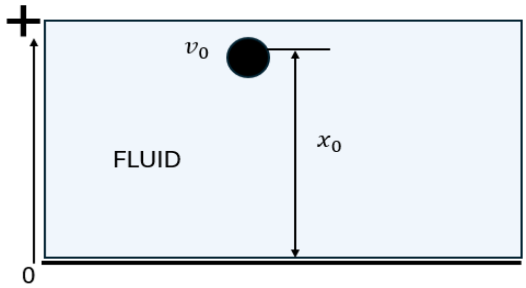

Figure 23: Bouncing ball system.

1) The mathematical model represents a bouncing ball, where the motion is influenced by gravity, aerodynamic drag, and buoyant force. A dissipation of kinetic energy is included to model the impact with the ground. To solve the model, ODE78 is chosen due to its excellent balance between accuracy and efficiency. To properly represent and track the ground impact, an Event Function is added as *option* to the ODE78 caller, so that whenever the ball reaches the ground, the integration is stopped and then performed with updated initial conditions; this cycle is executed until either the end time is reached or the velocity falls under a tolerance threshold of $10^{-3}\,\frac{m}{s}$. The results are shown in Figure 24 and Figure 25:

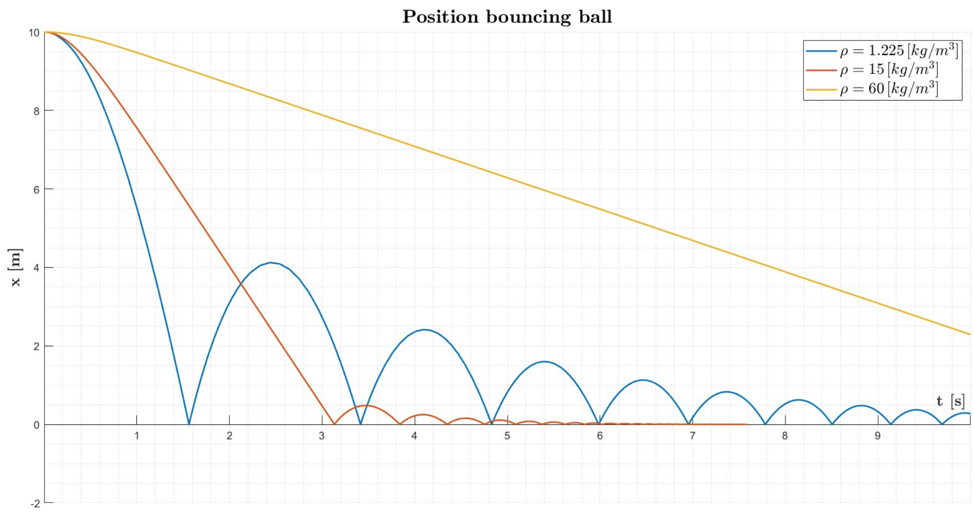

Figure 24: Bouncing ball position

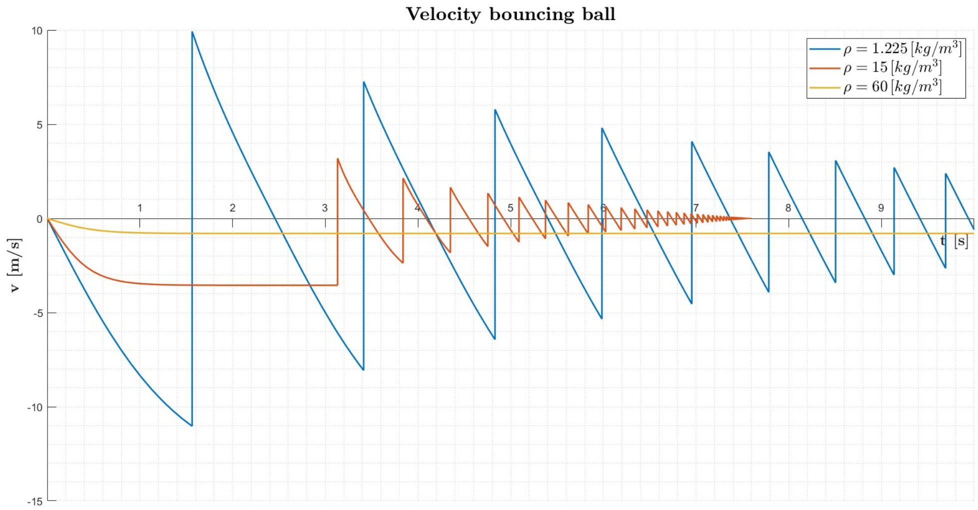

Figure 25: Bouncing ball velocity

For $\rho = 1.225\,\dfrac{\mathrm{kg}}{\mathrm{m}^3}$, density of air according to the International Standard Atmosphere (ISA), the density is insufficient to stop the ball from bouncing. However, the position amplitude decreases less than in cases with higher density.

The velocity plot shows the damping effect due to impact. In this scenario, aerodynamic drag is the dominant force driving the ball's bounce, as the density does not significantly enhance the Archimedes force, while drag becomes crucial due to the resulting higher velocity, as demonstrated below in Figure 26.

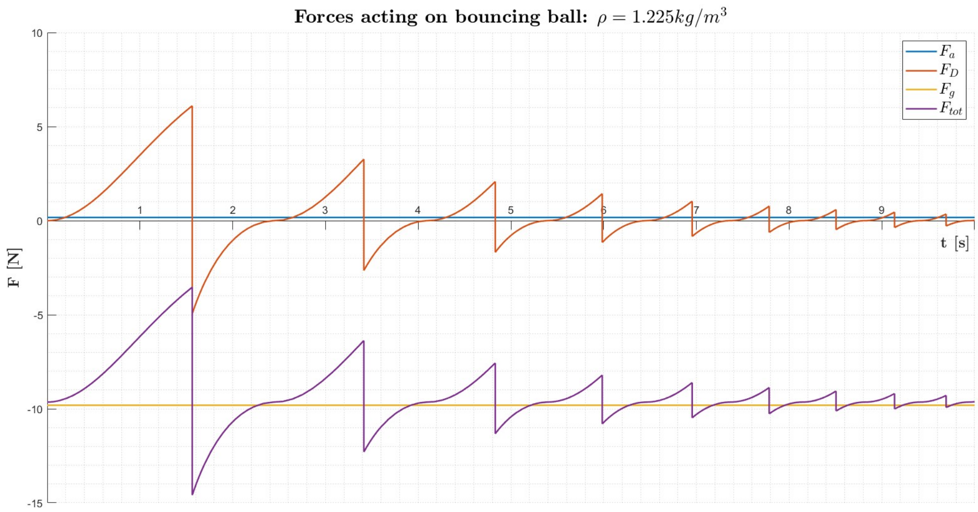

Figure 26: Forces acting on the ball

2) For $\rho = 15\,\dfrac{\mathrm{kg}}{\mathrm{m}^3}$, the ball first reaches the ground after the case with $\rho = 1.225\,\dfrac{\mathrm{kg}}{\mathrm{m}^3}$; however, the bouncing amplitude quickly diminishes to zero. This behaviour is due to the significant buoyant force, which slows the initial descent. As a result, subsequent bounces are minimal due to the lack of kinetic energy, causing the cycle to end before reaching the maximum time. The acting forces are shown in Figure 27:

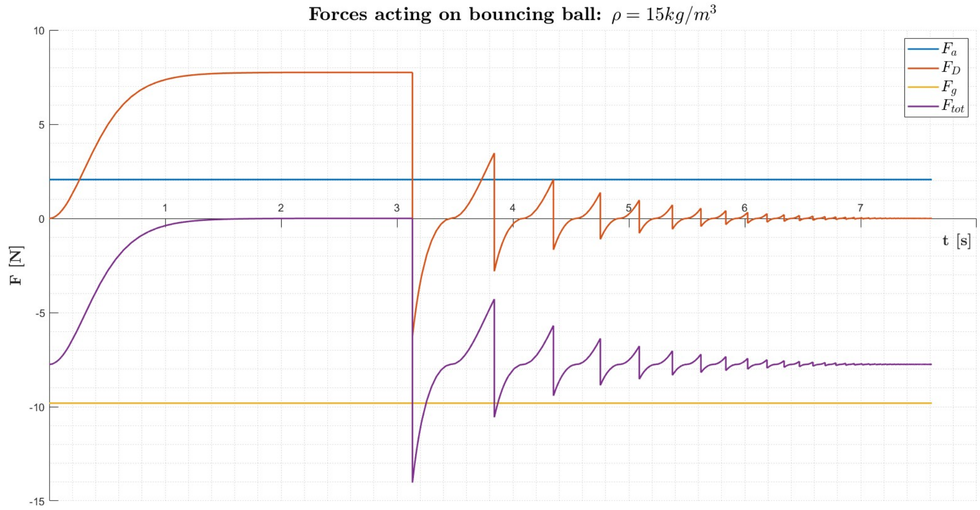

Figure 27: Forces acting on the ball

Analysing this case, it can be appreciated the role and correct functioning of the Event Function.

3) In case of $\rho = 60\,\dfrac{kg}{m^3}$, the buoyant force becomes so large that the ball is not able to reach the ground within $10\,s$. After a rapid transient, the height is enough to allow reaching a constant velocity, suggesting that the overall force acting on the mass is almost null and the aerodynamic limit velocity is reached, as Figure 28 suggests.

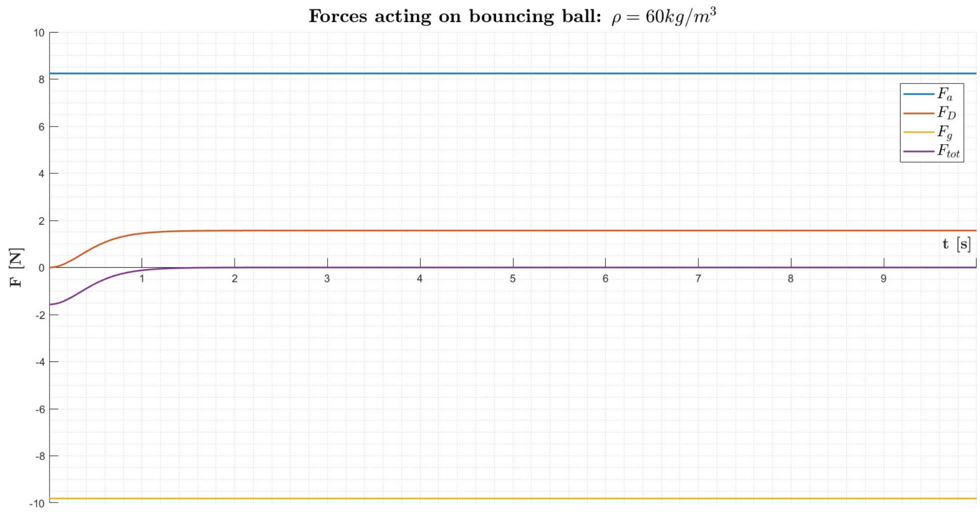

Figure 28: Forces acting on the ball

4) The resolution of the model with $\rho = 60\,\dfrac{kg}{m^3}$ is handled to RK4. The results are shown in Figure 29 :

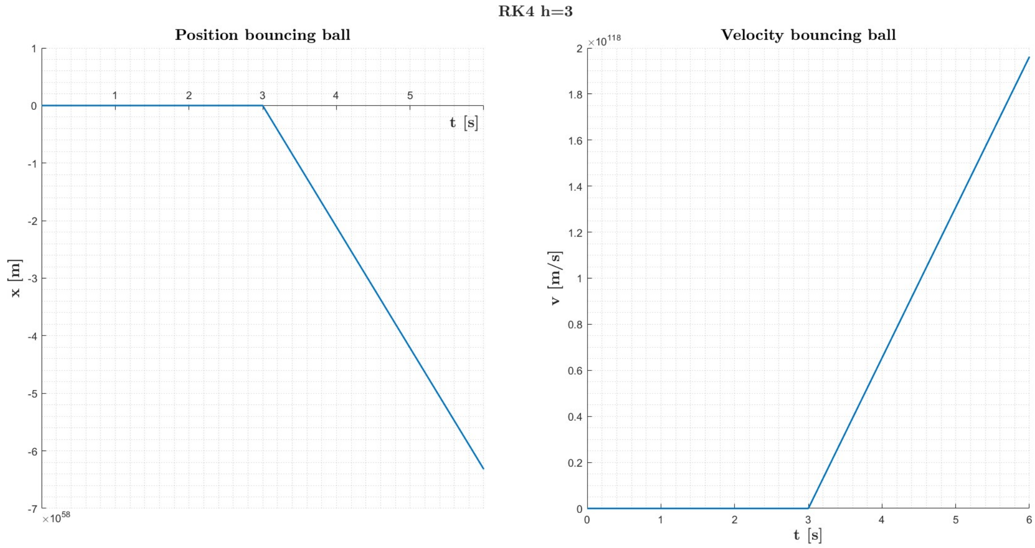

Figure 29: Bouncing ball $\rho = 60\,\dfrac{kg}{m^3}$, using RK4 with h=3

It is clear that the time step used is not suitable for the method, as it exceeds the stability domain. By decreasing the time step, one can observe a transition from instability to stability, passing through marginal stability. The results for $h = [1; 0.8925; 0.1]$ shown in Figure 30 and Figure 31 :

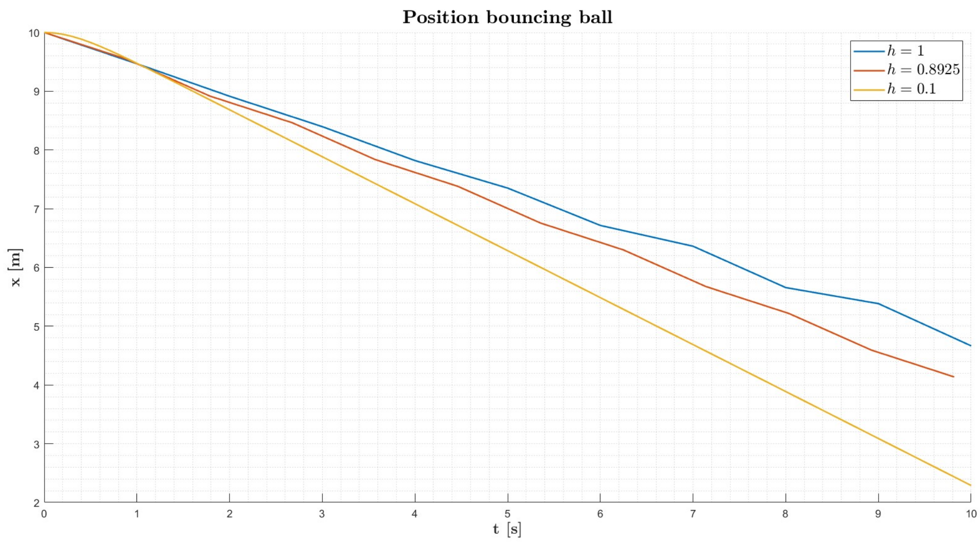

Figure 30: Bouncing ball position $\rho = 60\,\dfrac{kg}{m^3}$, using RK4 with $h = [1; 0.8925; 0.1]$

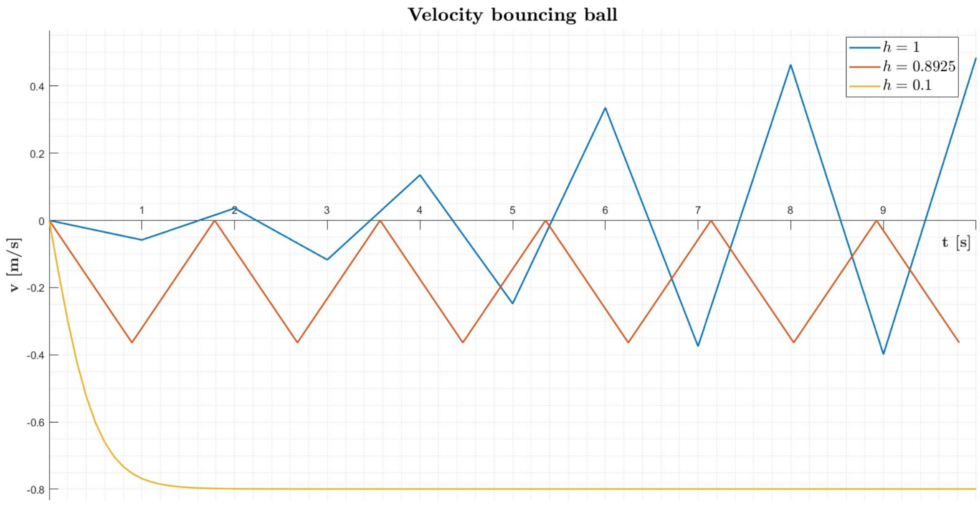

Figure 31: Bouncing ball velocity $\rho = 60\,\dfrac{kg}{m^3}$, using RK4 with $h = [1; 0.8925; 0.1]$

Despite the position might seem following a similar trend for the three values of h, the real behaviour can be appreciated through the velocity plot, where for these different values of h, the solution exhibits different stability behaviours. On this basis it can be stated that the system does not show a stiff-behaviour, since the time step, to have a stable mapping of the solution, is closely under the unity, suggesting that the eigenvalues of the linearized problem are not particularly large; furthermore, considering the maximum time of the problem, a time step larger than the unity would not be significant since it would lead to a solution based only on 10 points. The solution obtained by applying RK4, with h=0.1, is reasonably accurate if compared with ODE78 solution. [2]

References

[1] François E. Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

[2] Topputo F. *Notes of Modeling and Simulation of Aerospace Systems*. Politecnico di Milano, a.a. 2024/2025.