MSAS – Assignment #1: Simulation

Lorenzo Cucchi, 221732

1 Implicit equations

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)

The problem consists in finding the couple(s) $\{x_1, x_2\}$ which satisfy the relation $f(x_1, x_2) = [0, 0]^T$. In order to solve the problem using Newton's method, the analytical form or an approximation of the inverse Jacobian matrix $\mathbf{J^{-1}}$ is needed. In fact, according to Newton's method, the i-th iteration $\mathbf{x}_i$ is given by Equation 1.

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \mathbf{J^{-1}}(\mathbf{x}_{i-1})\mathbf{f}(\mathbf{x}_{i-1}) \tag{1}$$

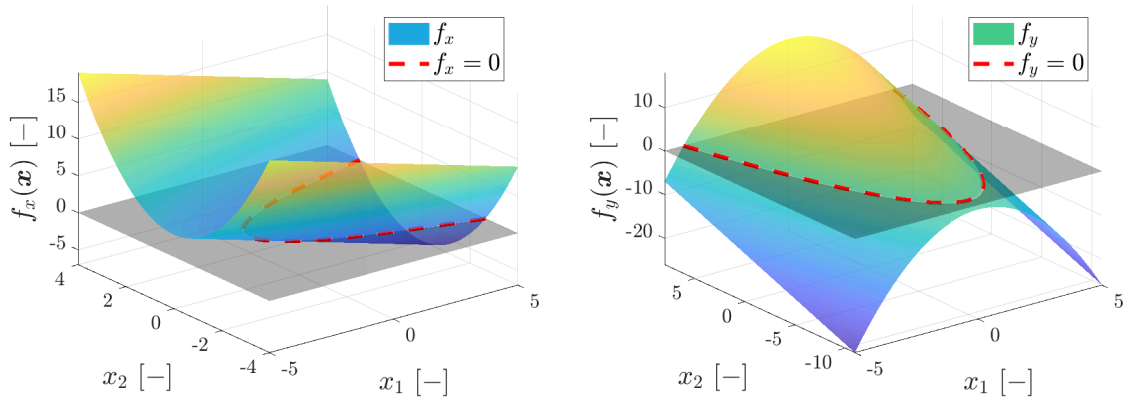

Figure 1: Surf plot of function $f_x(\mathbf{x}) = x_2^2 - x_1 - 2$ and function $f_y(\mathbf{x}) = -x_1^2 + x_2 + 10$.

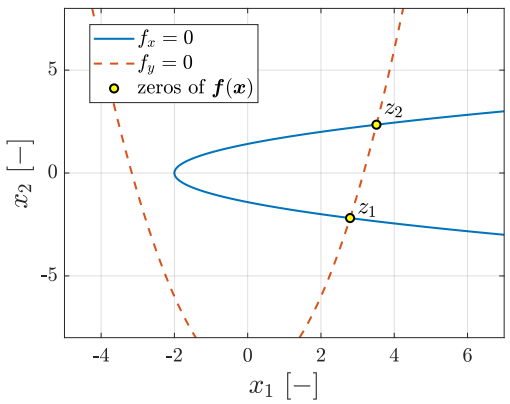

Figure 2: Zeros (z_1 and z_2) of the problem.

The analytical inverse Jacobian matrix can be efficiently obtained using the MATLAB symbolic toolbox through a few simple steps. Notably, for cases with a straightforward structure, like this one, manual calculation proves to be a faster alternative. On the other side, the approximated form of the Jacobian matrix is obtained through finite difference methods. In particular, first order *forward* and *centered* finite difference schemes are implemented. In this sense, the small increment δ which is applied in the finite difference methods follows the *rule*: $\delta = \sqrt{eps} \cdot max(1, |\mathbf{x}|)$, where *eps* is the machine epsilon. Furthermore, to improve code efficiency and accuracy, the Jacobian calculated with finite differences is never inverted; instead, the MATLAB command `mldivide` or '\' is used.

As Figure 2 shows, there are two zeros of the function $\mathbf{f}(\mathbf{x})$. Indeed, two appropriate initial guesses $\mathbf{x}_0$ must be found to make the algorithms converge at the two zeros. The initial guesses are therefore $x_{0,1} = [1, -4]^T$ and $x_{0,2} = [6, 5]^T$. The stopping criteria chosen is the accuracy of the function evaluated in $\mathbf{x}_i$: when both the absolute values of $f_x(\mathbf{x}_i)$ and $f_y(\mathbf{x}_i)$ are lower than a tolerance set to $1e-8$ the algorithm stops. Results reported in Table 1 show that the three algorithms converge at very close values and take the same number of iterations: the error $||err|| = ||\mathbf{f}(\mathbf{x}_{end,analytical}) - \mathbf{f}(\mathbf{x}_{end,method})||$ is very low, suggesting the high accuracy of both the forward and centered differences approximations.

Method	$\boldsymbol{z_1}$	**Iterations**	$\boldsymbol{||err||}$
Analytical	[2.794695112889339, -2.189679226029504]	5	-
Forward differences	[2.794695112889379, -2.189679226029503]	5	3.9563e-14
Centered differences	[2.794695112889374, -2.189679226029529]	5	4.2384e-14
Method	$\boldsymbol{z_2}$	**Iterations**	$\boldsymbol{||err||}$
Analytical	[3.513999235947622, 2.348190630240421]	5	-
Forward differences	[3.513999235947627, 2.348190630240434]	5	5.1789e-15
Centered differences	[3.513999235947622, 2.348190630240424]	5	2.2204e-15

Table 1: Zeros and errors of the used methods.

2 Numerical solution of ODE

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)

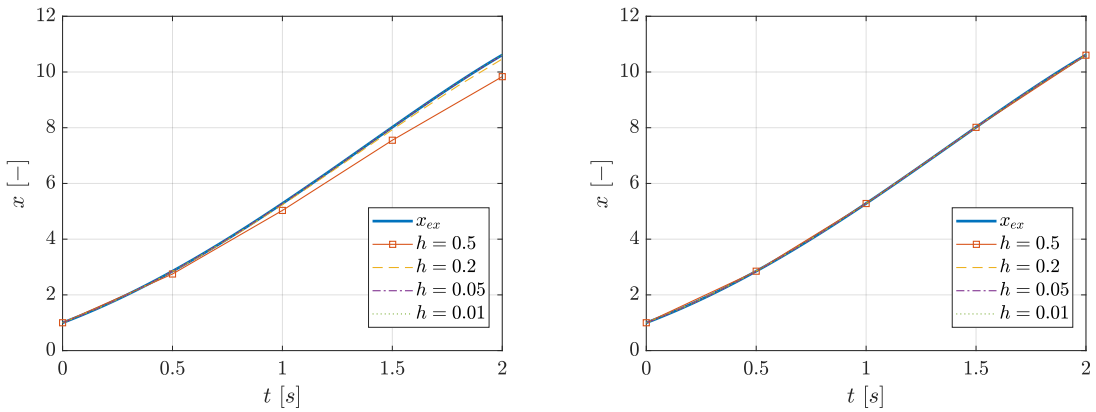

Figure 3: Solutions provided by with $RK2$ (left) and $RK4$ (right) methods by varying h value.

The initial value problem (IVP) over the interval $t \in [0, 2]$ is solved using step sizes $h_1 = 0.5$, $h_2 = 0.2$, $h_3 = 0.05$, and $h_4 = 0.01$ with both RK2 and RK4 methods. The results are illustrated in (Figure 3). Subsequently, the integration errors are compared in Figure 4. As evident from the figures, the RK4 method exhibits superior accuracy compared to RK2, even when using larger step sizes. The higher order of the RK4 method not only leads to increased accuracy but also results in a more substantial reduction in the global integration error, as demonstrated in Figure 5 (left). Considering the balance between computational time and integration error, Figure 5 (right) indicates that, to achieve the same level of accuracy in the final value, the RK4 method requires less time compared to the RK2 method. Therefore, the RK4 method is recommended irrespective of computational time considerations, as it proves to be the most efficient in terms of both accuracy and CPU time.

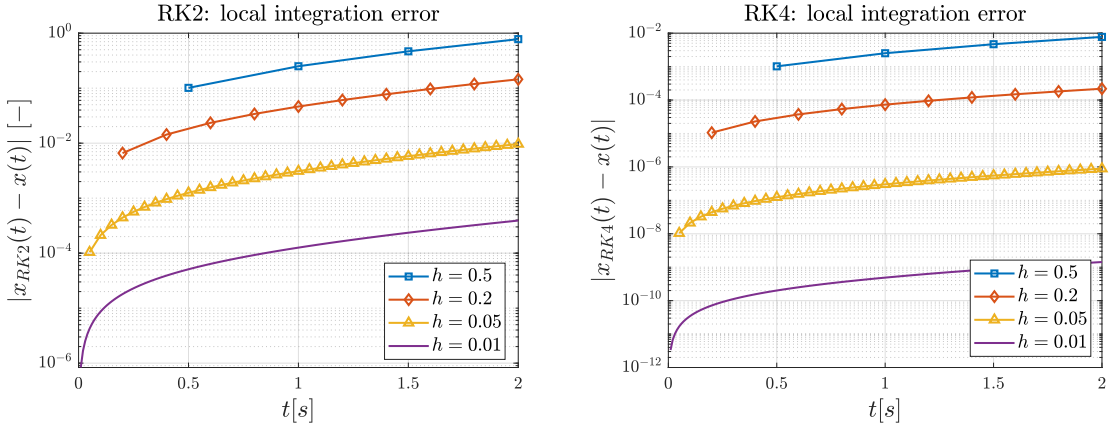

Figure 4: Local integration errors provided by with $RK2$ (left) and $RK4$ (right) methods by varying h value.

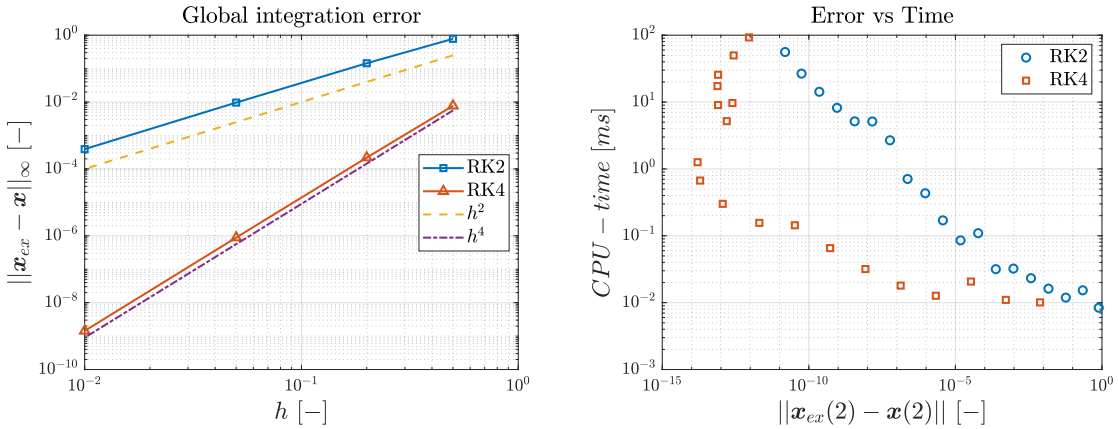

Figure 5: Global integration errors (left) and comparison between error and computational time of RK2 and RK4 methods (right).

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)

In order to retrieve the expression of the linear operator $F_{RK2}(h, \alpha)$ a generic RK2 iteration with step h is derived:

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

where, in our case, $\mathbf{f}(\mathbf{x}_k, t_k) = \mathbf{A}(\alpha)\mathbf{x}_k$. By substituting the first equation of Equation 2 in the second one it is obtained:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{2}\mathbf{A}(\alpha)\mathbf{x}_k + \frac{h}{2}\mathbf{A}(\alpha)\mathbf{x}_k + \frac{h}{2}h\mathbf{A}(\alpha)^2\mathbf{x}_k \tag{3}$$

By rearranging:

$$\mathbf{x}_{k+1} = (\mathbf{I} + h\mathbf{A}(\alpha) + \frac{h^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_k = \mathbf{F}_{RK2}(h, \alpha)\mathbf{x}_k \tag{4}$$

The same procedure can be performed to find F_{RK4}:

$$\mathbf{F}_{RK4}(h, \alpha) = \mathbf{I} + h\mathbf{A}(\alpha) + \frac{h^2}{2}\mathbf{A}^2(\alpha) + \frac{h^3}{6}\mathbf{A}^3(\alpha) + \frac{h^4}{24}\mathbf{A}^4(\alpha) \tag{5}$$

Table 2 shows the solutions of the statement " Find $h \geq 0$ s.t.max $(|\mathrm{eig}(F(h, \alpha))|) = 1$" imposing $\alpha = \pi$ with both F_{RK2} and F_{RK4}.

	RK2	**RK4**
h	2.0000000	2.7852935

Table 2: Results of statement "Find $h \geq 0$ s.t.max $(|\mathrm{eig}(F(h, \alpha))|) = 1$" where $\alpha = \pi$ with F_{RK2} and F_{RK4} functions.

Solving the problem for $\alpha \in [0, \pi]$ allows us to ascertain and visualize the numerical stability domains for both RK2 and RK4 methods, as depicted in Figure 6. As illustrated, the stability domains expand with higher approximation orders, necessitating larger step sizes for higher-order algorithms. The stability domains noticeably expand with higher approximation orders.

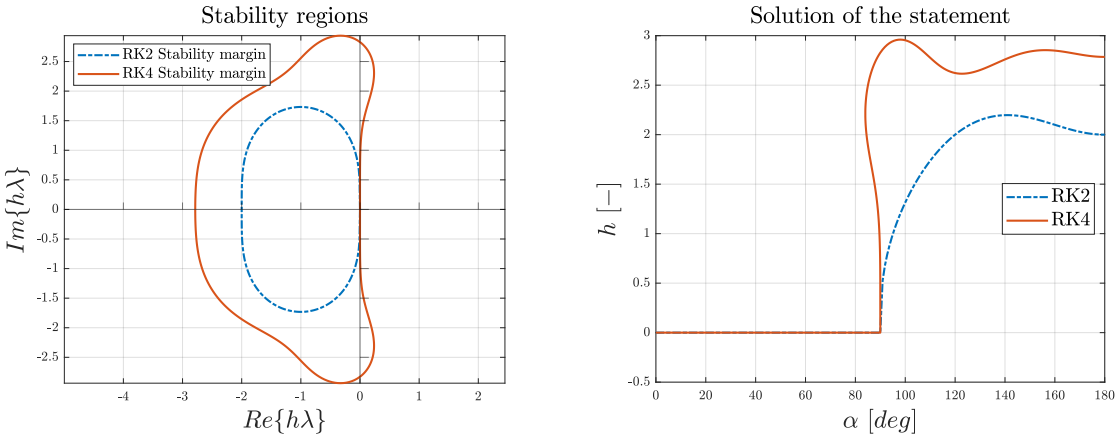

Figure 6: RK2 and RK4 method stability domain (left) and solution of the statement "Find $h \geq 0$ s.t.max $(|\text{eig}(F(h, \alpha))|) = 1$" with RK2 and RK4 (right).

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)

The outcomes are illustrated in Figure 7 and Figure 8. As observed in the figures, the use of higher-order approximation methods allows for larger permissible values of the step size h. Consequently, to maintain the same error tolerance $|\mathbf{x}_{\text{an}}(1) - \mathbf{x}_{\text{RK1}}(1)|_\infty$, higher-order methods can accommodate larger h, resulting in fewer steps at which the function needs to be evaluated, this observation is further exemplified in Figure 8.

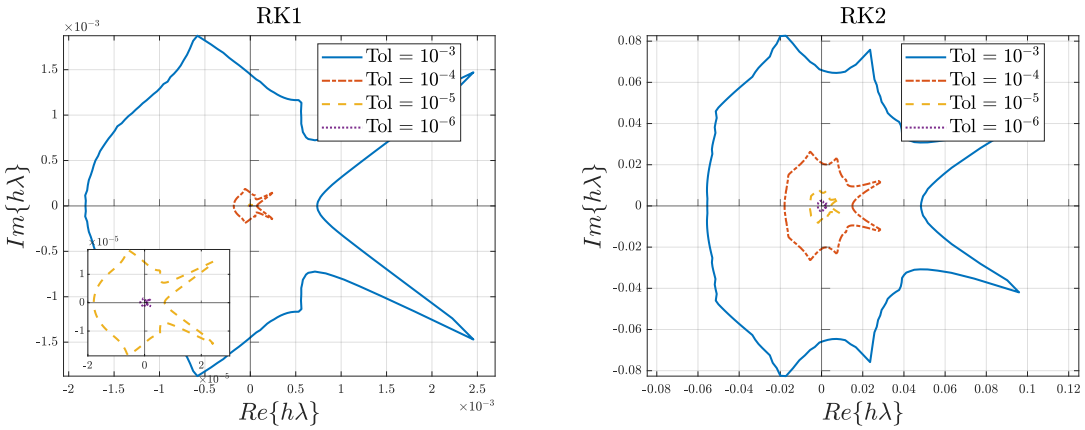

Figure 7: Five locus of solutions for RK1 (left) and RK2 (right) methods.

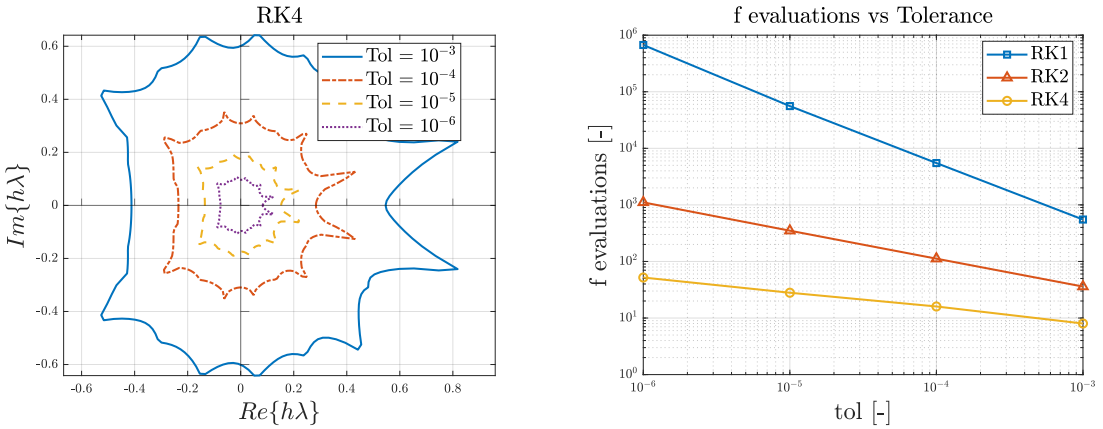

Figure 8: RK4 locus of solutions (left) and function evaluations vs. tolerance (right).

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)

BI_i backinterpolation methods are a special implicit Runge-Kutta (IRK) methods. In order to derive the expression of the linear operator $B_{\text{BI2}_{0.4}}(h,\alpha)$ such that $\mathbf{x}_{k+1} = B_{\text{BI2}_{0.4}}(h,\alpha)\mathbf{x}_k$, a generic iteration of RK2 with step hθ is derived:

$$\begin{cases} \mathbf{x}^P_{k+h\theta} = \mathbf{x}_k + \theta h \mathbf{f}(\mathbf{x}_k, t_k) \\ \mathbf{x}^C_{k+h\theta} = \mathbf{x}_k + \frac{\theta h}{2}[\mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}(\mathbf{x}^P_{k+h\theta}, t_{k+h\theta})] \end{cases} \tag{6}$$

where, in our case, $\mathbf{f}(\mathbf{x}_k, t_k) = \mathbf{A}(\alpha)\mathbf{x}_k$, by substituting this equation and the first equation of Equation 6 in the second one, we obtain:

$$\mathbf{x}_{k+h\theta} = \mathbf{x}_k + \frac{\theta h}{2}\mathbf{A}(\alpha)\mathbf{x}_k + \frac{\theta h}{2}\mathbf{A}(\alpha)\mathbf{x}_k + \frac{\theta h}{2}\theta h\mathbf{A}^2(\alpha) \tag{7}$$

Which can be rewritten as:

$$\mathbf{x}_{k+h\theta} = (\mathbf{I} + h\theta\mathbf{A}(\alpha) + \frac{\theta^2 h^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_k \tag{8}$$

The same procedure is performed to find the relation between $\mathbf{x}_{k+h\theta}$ and $\mathbf{x}_{k+1}$. In order to achieve that, the generic RK2 iteration with $-h(1-\theta)$ step is considered:

$$\begin{cases} \mathbf{x}^P_{k+h\theta} = \mathbf{x}_k - h(1-\theta)\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) \\ \mathbf{x}^C_{k+h\theta} = \mathbf{x}_k - \frac{h(1-\theta)}{2}[\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{f}(\mathbf{x}^P_{k+h\theta}, t_{k+h\theta})] \end{cases} \tag{9}$$

Following the procedure that allowed to find Equation 8, Equation 9 becomes:

$$\mathbf{x}_{k+h\theta} = (\mathbf{I} - h(1-\theta)\mathbf{A}(\alpha) + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_{k+1} \tag{10}$$

By comparing Equation 8 and Equation 9:

$$(\mathbf{I} + h\theta\mathbf{A}(\alpha) + \frac{\theta^2 h^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_k = (\mathbf{I} - h(1-\theta)\mathbf{A}(\alpha) + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_{k+1} \tag{11}$$

By isolating $\mathbf{x}_{k+1}$:

$$\mathbf{x}_{k+1} = (\mathbf{I} - h(1-\theta)\mathbf{A}(\alpha) + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2(\alpha))^{-1}(\mathbf{I} + h\theta\mathbf{A}(\alpha) + \frac{\theta^2 h^2}{2}\mathbf{A}^2(\alpha))\mathbf{x}_k \tag{12}$$

The linear operator $B_{\mathrm{BI2}_\theta}(h, \alpha)$ is obtained:

$$B_{\mathrm{BI2}_{0.4}}(h, \alpha) = (\mathbf{I} - h(1-\theta)\mathbf{A}(\alpha) + \frac{h^2(1-\theta)^2}{2}\mathbf{A}^2(\alpha))^{-1}(\mathbf{I} + h\theta\mathbf{A}(\alpha) + \frac{\theta^2 h^2}{2}\mathbf{A}^2(\alpha)) \tag{13}$$

By imposing $\theta = 0.4$ the linear operator $B_{\mathrm{BI2}_{0.4}}(h, \alpha)$ is obtained. Given the expression of the linear operator $B_{\mathrm{BI2}_\theta}(h, \alpha)$, it is possible to derive the domain of numerical stability of $BI2_\theta$ method with different θ values by solving "Find $h \geq 0$ s.t.max $(|\mathrm{eig}(BI2_\theta(h, \alpha))|) = 1$". Results are shown in Figure 9, where the stability domains are depicted in the $(h\lambda)$-plane.

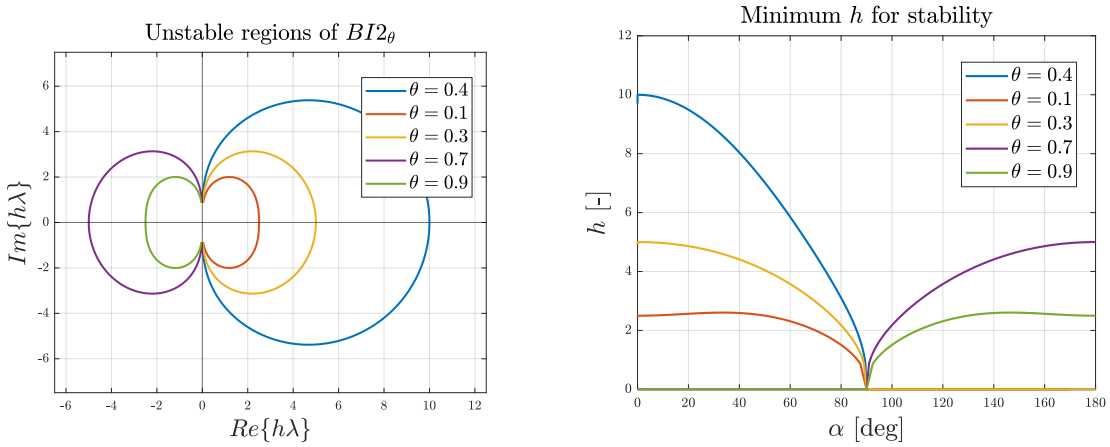

Figure 9: Unstable domains of $\mathrm{BI2}_\theta$ for different θ values and inimum step h for stabilit.

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.

By setting $h = 0.1$, the solution to the differential equation $\dot{\mathbf{x}} = B\mathbf{x}$ is obtained using both the Runge-Kutta fourth order (RK4) and Implicit Extrapolation fourth order (IEX4) methods. The achieved solutions are then compared against the analytical solution $x_{\mathrm{analytical}} = e^{\mathbf{B}t}$, as depicted in Figure 11. Notably, the RK4 method exhibits divergence in integration errors, attributed to the eigenvalues of the matrix $\mathbf{B}$, specifically $\lambda_i = [-1, -400]$. The instability arises from the fact that the eigenvalue λ_2, when multiplied by the step size $h = 0.1$, falls into the unstable region of the RK4 method. Consequently, the associated component of the solution deviates significantly from the analytical trajectory. Despite the second eigenvalue lying within the stable domain of RK4, the coupled nature of the state equations amplifies the divergence. In contrast, the IEX4 method displays a more favorable behavior. As illustrated in Figure 10, both $h\lambda_i$ values for the eigenvalues lie within the stable domain of the method. This characteristic enables the solution to converge towards the analytical solution with minimal integration errors. The efficacy of IEX4 in handling the coupled nature of the state equations stands out, presenting a marked improvement over the RK4 method.

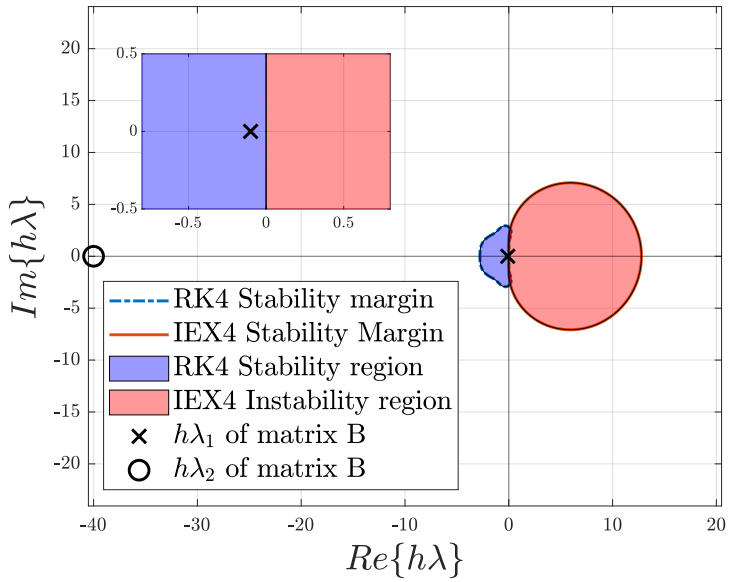

Figure 10: Stable and unstable domains of RK4 and IEX4 methods.

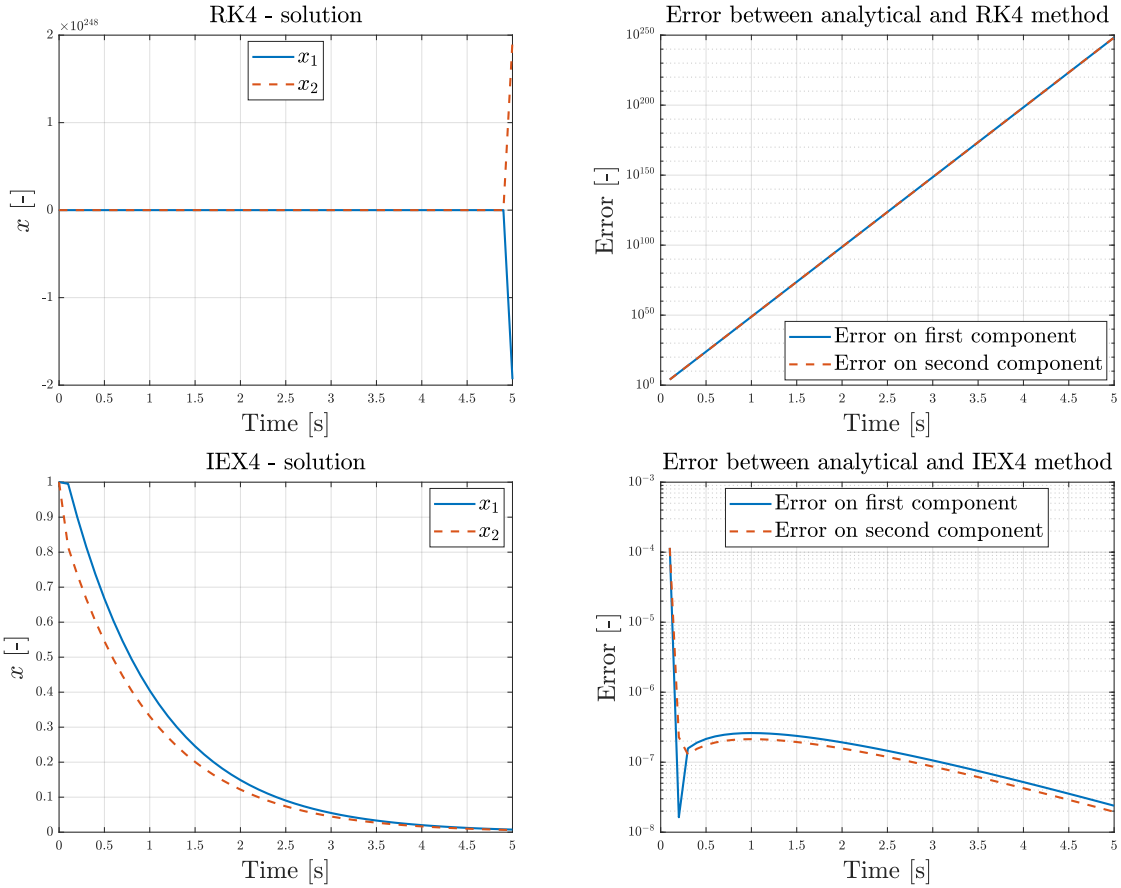

Figure 11: Solutions obtained with RK4 and IEX4 methods and their respective errors compared with analytic solution.

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)

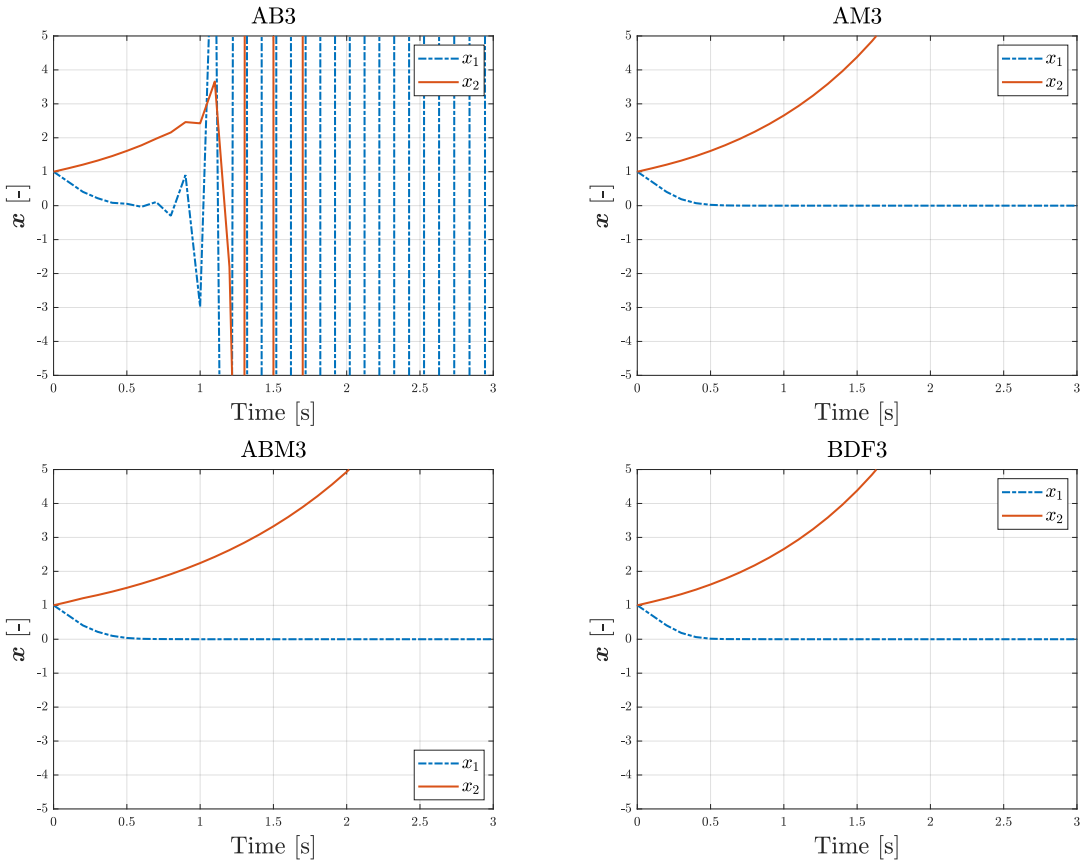

Figure 12: Solutions obtained with AB3, AM3, ABM3 and BDF3 methods.

Figure 12 displays the results of the numerical solution obtained using a step size $h = 0.1$ is adopted with AB3, AM3, ABM3 and BDF3 methods. The figure reveals that the AB3 method suffers from integration instability on both the components x_1 and x_2. On the other side, AM3 and BDF3 methods can provide a reliable solution for the x_1 component but not for x_2. Notably, the ABM3 method seems to suffer from instability in the x_1 components during the time interval $t \simeq 1.5$ s to $t \simeq= 2.5$ s and the same problem encountered with the other methods for the x_2 component. In order to study such a behaviour, the linearized system is studied. Regarded as $\mathbf{x}_0$ the equilibrium solution, the problem is formulated as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},t) \simeq \mathbf{f}(\mathbf{x}_0,t) + \left.\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x}-\mathbf{x}_0) \tag{14}$$

By definition, $\mathbf{f}(\mathbf{x}_0,t) = \mathbf{0}$. Furthermore, single $\mathbf{x}_0$ value is obtained by imposing $\dot{\mathbf{x}} = \mathbf{0}$: $\mathbf{x}_0 = [0,0]^T$. As a result, Equation 14 becomes:

$$\mathbf{f}(\mathbf{x},t) \simeq \left.\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} \mathbf{x} \tag{15}$$

The partial derivative term in Equation 15 is the Jacobian matrix of function $\mathbf{f}(\mathbf{x}, t)$:

$$\mathbf{J} = \mathbf{J}(\mathbf{x}, t) = \begin{bmatrix} \frac{5}{2}\left[1 + 8\sin(t)\right] & 0 \\ 1 - y & 1 - x \end{bmatrix} \tag{16}$$

the linearized problem in $\mathbf{x}_0$ is:

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{5}{2}\left[1 + 8\sin(t)\right] & 0 \\ 1 & 1 \end{bmatrix} \mathbf{x} \tag{17}$$

and, thus, the eigenvalues can be evaluated at each instant of time t. Left plot of Figure 13 shows the evolution in time of the $h\lambda_i$ values associated to the linearized problem of Equation 17. Looking at right plot of Figure 13, the behaviour of the four proposed methods is characterized:

- **AB3**: $h\lambda_{x_2}$ is never inside the method stability domain, so the x_2 component diverges. With the exception of little intervals at the beginning and at the end of the time-span, the same situation is encountered for $h\lambda_x$;
- **AM3**: $h\lambda_{x_2}$ is never inside the method stability domain, so the x_2 component diverges. On the other side, the $h\lambda_x$ is always in the stable domain, resulting in the convergence of the x component;
- **ABM3**: the situation is the same encountered with AM3 method, except for the fact that for a small interval between $t \simeq 1.0$ s and $t \simeq= 2.0$ s the $h\lambda_x$ values are outside the stability domain;
- **BDF3**: the instability of the x_2 component is encountered for the whole time-span, while the x_1 component is stable for all the time-span. This is due to the instability region being almost entirely limited to the positive real axis.

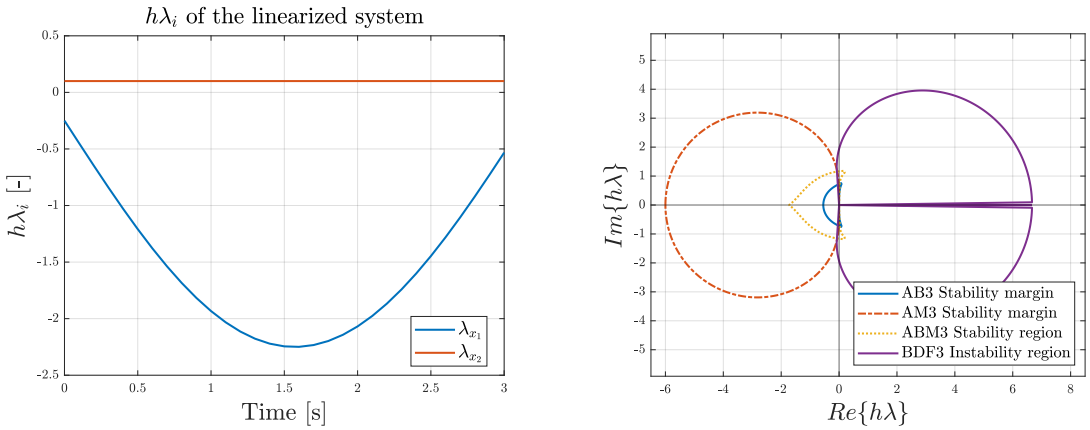

Figure 13: $h\lambda_i$ values of the linearized system (left). Stability and Instability domains of the proposed methods (right).