I pledge that the work submitted for this coursework, both the report and the MATLAB code, is my own unassisted work unless stated otherwise.

CID...................... 01730921

Are you a Year 4 student?.. yes

Coursework 3

Fill in your CID and include the problem sheet in the coursework. Before you start working on the coursework, read the coursework guidelines. Any marks received for this coursework are only indicative and may be subject to moderation and scaling. *The mastery component is marked with a star.*

Exercise 1 (Predictor-Corrector Methods) — % of course mark: /6.0

a) Develop a predictor-corrector method based on the explicit and implicit LMMs you developed in Coursework 2.

b) Calculate the local truncation error of the predictor-corrector method.

c) Find the region and interval of absolute stability of the predictor-corrector method.

d) Find the region and interval of absolute stability of the explicit and implicit LMMs developed in Coursework 2 and compare them with those of the predictor-corrector method.

Exercise 2 (Predictor-Corrector and Nonlinear Systems) — % of course mark: /7.0

Solve the initial value problem (1) describing the chemical reaction of Robertson with the predictor-corrector method developed in Exercise 1.

$$\begin{cases} x' = -0.04x + 10^4 yz, \\ y' = 0.04x - 10^4 yz - 3\cdot 10^7 y^2, \\ z' = 3\cdot 10^7 y^2, \end{cases} \tag{1}$$

$$x(0) = 1,\ y(0) = z(0) = 0,\ t = [0, 100].$$

Exercise 3 (Implicit LMM and Nonlinear Systems) — % of course mark: /7.0

a) Solve the initial value problem for the Rabinovich–Fabrikant system (2) with the implicit LMM developed in Coursework 2. Use the Fixed point iteration method and Newton method to solve the nonlinear system of equations.

$$\begin{cases} x' = y(z - 1 + x^2) + \gamma x, \\ y' = x(3z + 1 - x^2) + \gamma y, \\ z' = -2z(\alpha + xy), \end{cases} \tag{2}$$

$$x(0) = -1.0,\ y(0) = 0.0,\ z(0) = 0.5,\ \alpha = 1.1,\ \gamma = 0.87,\ t = [0, 50].$$

b) Compare the number of iterations and execution time of the Fixed point iteration method and Newton method.

Exercise 4 (LMM and Absolute Stability) **% of course mark:** **/4.0★**

Find the coefficients α_2, α_0, β_0 of the LMM

$$x_{n+3} + \alpha_2 x_{n+2} + \alpha_0 x_n = h\beta_0 f_n$$

that give a convergent LMM, with the largest interval of absolute stability, when applied to

$$x' = \lambda x,\ Re(\lambda) < 0\,.$$

What is this largest interval?

Coursework mark: **% of course mark**

Coursework Guidelines

Below is a set of guidelines to help you understand what coursework is and how to improve it.

Coursework

- The coursework requires more than just following what has been done in the lectures, some amount of individual work is expected.
- The coursework report should describe in a concise, clear, and coherent way of what you did, how you did it, and what results you have.
- The report should be understandable to the reader with the mathematical background, but unfamiliar with your current work.
- Do not bloat the report by paraphrasing or presenting the results in different forms.
- Use high-quality and carefully constructed figures with captions and annotated axis, put figures where they belong.
- All numerical solutions should be presented as graphs.
- Use tables only if they are more explanatory than figures. The maximum table length is a half page.
- All figures and tables should be embedded in the report. The report should contain all discussions and explanations of the methods and algorithms, and interpretations of your results and further conclusions.
- The report should be typeset in LaTeX or Word Editor and submitted as a single pdf-file.
- The maximum length of the report is ten A4-pages (additional 3 pages is allowed for Year 4 students); the problem sheet is not included in these ten pages.
- Do not include any codes in the report.
- Marks are not based solely on correctness. The results must be described and interpreted. The presentation and discussion is as important as the correctness of the results.

Codes

- You cannot use third party numerical software in the coursework.
- The code you developed should be well-structured and organised, as well as properly commented to allow the reader to understand what the code does and how it works.
- All codes should run out of the box and require no modification to generate the results presented in the report.

Submission

- The coursework submission must be made via Turnitin on your Blackboard page. You must complete and submit the coursework anonymously, **the deadline is 1pm on the date of submission** (unless stated otherwise). The coursework should be submitted via two separate Turnitin drop boxes as a pdf-file of the report and a zip-file containing MATLAB (m-files only) or Python (py-files only) code. The code should be in the directory named `CID_Coursework#`. The report and the zip-file should be named as `CID_Coursework#.pdf` and `CID_Coursework#.zip` , respectively. The executable MATLAB (or Python) scripts for the exercises should be named as follows: exercise1.m, exercise2.m, etc.

Numerical odes - Coursework 3

CID: 01730921

October 2022

Intro

The explicit method in coursework 2 was given by:

$$x_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{h}{24}\left(41f_n - 40f_{n-1} + 11f_{n-2}\right) \tag{1}$$

The implicit method in coursework 2 was given by:

$$x_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{h}{72}\left(40f_{n+1} + 3f_n - 7f_{n-2}\right) \tag{2}$$

1 Question 1

1.1 PECE method

The basic principle underlaying predictor-corrector methods is to predict the numerical solution with the predictor, and then correct it with the corrector. We are going to use the explict method as predictor and implicit method as corrector. Therefore, for $n = 0, 1, 2, \ldots$:

$$\begin{aligned}
\textbf{P}\text{redict} \quad & \hat{x}_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{h}{24}\left(41f(t_n, x_n) - 40f(t_{n-1}, x_{n-1}) + 11f(t_{n-2}, x_{n-2})\right) && (3)\\
\textbf{E}\text{valuate} \quad & f(t_{n+1}, \hat{x}_{n+1}) \\
\textbf{C}\text{orrect} \quad & x_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{h}{72}\left(40f(t_{n+1}, \hat{x}_{n+1}) + 3f(t_n, x_n) - 7f(t_{n-2}, x_{n-2})\right) && (4)\\
\textbf{E}\text{valuate} \quad & f(t_{n+1}, x_{n+1})
\end{aligned}$$

1.2 Local truncation error

In order to find the local truncation error for our PECE method we take the continuous form of the predictor

$$\hat{x}(t_{n+1}) = \frac{3}{2}x(t_n) - \frac{1}{2}x(t_{n-1}) + \frac{h}{24}\left(41f(t_n, x(t_n)) - 40f(t_{n-1}, x(t_{n-1})) + 11f(t_{n-2}, x(t_{n-2}))\right) \tag{5}$$

and use the difference equation $x' = f(t, x)$ to replace the terms of the type $f(t_n, x(t_n))$:

$$\hat{x}(t_{n+1}) = \frac{3}{2}x(t_n) - \frac{1}{2}x(t_{n-1}) + \frac{h}{24}\left(41x'(t_n) - 40x'(t_{n-1}) + 11x'(t_{n-2})\right) \tag{6}$$

Then, we take the continuous form of the corrector

$$x(t_{n+1}) = \frac{3}{2}x(t_n) - \frac{1}{2}x(t_{n-1}) + \frac{h}{72}\left(40f(t_{n+1}, \hat{x}(t_{n+1})) + 3f(t_n, x(t_n)) - 7f(t_{n-2}, x(t_{n-2}))\right) \tag{7}$$

and replace again the terms of the type $f(t_n, x(t_n))$:

$$x(t_{n+1}) = \frac{3}{2}x(t_n) - \frac{1}{2}x(t_{n-1}) + \frac{h}{72}\left(40\hat{x}'(t_{n+1}) + 3x'(t_n) - 7x'(t_{n-2})\right) \tag{8}$$

Taylor expanding (6) we get:

$$\begin{aligned}
\hat{x}(t_{n+1}) &= \frac{3}{2}x(t_n) - \frac{1}{2}\left(x(t_n) - hx'(t_n) + \frac{h^2}{2}x''(t_n) - \frac{h^3}{6}x'''(t_n) + \frac{h^4}{24}x''''(t_n) + \mathcal{O}(h^5)\right) \\
&+ \frac{h}{24}\left(41x'(t_n) - 40\left(x'(t_n) - hx''(t_n) + \frac{h^2}{2}x'''(t_n) - \frac{h^3}{6}x''''(t_n) + \mathcal{O}(h^4)\right)\right. \\
&\left.+ 11\left(x'(t_n) - 2hx''(t_n) + 2h^2x'''(t_n) - \frac{4h^3}{3}x''''(t_n) + \mathcal{O}(h^4)\right)\right) \\
&= x(t_n) + hx'(t_n) + \frac{h^2}{2}x''(t_n) + \frac{h^3}{6}x'''(t_n) - \frac{17h^4}{48}x''''(t_n) + \mathcal{O}(h^5)
\end{aligned} \tag{9}$$

whereas Taylor expanding (8) and plugging in (9) we get:

$$\begin{aligned}
x(t_{n+1}) &= \frac{3}{2}x(t_n) - \frac{1}{2}\left(x(t_n) - hx'(t_n) + \frac{h^2}{2}x''(t_n) - \frac{h^3}{6}x'''(t_n) + \frac{h^4}{24}x''''(t_n) + \mathcal{O}(h^5)\right) \\
&+ \frac{h}{72}\left(40\left(x'(t_n) + hx''(t_n) + \frac{h^2}{2}x'''(t_n) + \frac{h^3}{6}x''''(t_n) + \mathcal{O}(h^4)\right)\right. \\
&\left.+ 3x'(t_n) - 7\left(x'(t_n) - 2hx''(t_n) + 2h^2x'''(t_n) - \frac{4h^3}{3}x''''(t_n) + \mathcal{O}(h^4)\right)\right) \\
&= x(t_n) + hx'(t_n) + \frac{h^2}{2}x''(t_n) + \frac{h^3}{6}x'''(t_n) + \frac{29h^4}{144}x''''(t_n) + \mathcal{O}(h^5)
\end{aligned} \tag{10}$$

Expanding the exact solution we have

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2}x''(t) + \frac{h^3}{6}x'''(t) + \frac{h^4}{24}x''''(t) + \mathcal{O}(h^5) \tag{11}$$

$$\implies \text{LTE} := x(t+h) - x(t_{n+1}) = \mathcal{O}(h^4) \qquad \implies \text{order} = 4 \tag{12}$$

1.3 Interval and region of stability of explicit and implicit methods

To find the region of absolute stability we make use of the Boundary Locus Method; i.e. we look for the boundary of the region of absolute stability. The roots of the stability polynomial on the boundary satisfy the equation $|r| = 1$. The solution of this equation is $r = e^{is}$, where $i = \sqrt{-1}$ and $s \in [0, 2\pi)$. Thus, to find the boundary of the region of absolute stability one has to plug $r = e^{is}$ into the stability polynomial and solve it for $\hat{h} = \hat{h}(s)$. By varying s, we can plot a closed curve in the complex $\hat{h}$-plane. This curve divides the plane into different subregions some of which (where $\hat{h}$ such that $|r| < 1$) form the region of absolute stability.

1.3.1 Explicit method

The stability polynomial for our explicit method applied to $x' = \lambda x$ is

$$p(r) = r^3 - \frac{3}{2}r^2 + \frac{1}{2}r - \frac{\hat{h}}{24}(41r^2 - 40r + 11), \quad \hat{h} = \lambda h \tag{13}$$

Substitution of $r = e^{is}$ into $p(r) = 0$, given by (13), leads to:

$$e^{3is} - \frac{3}{2}e^{2is} + \frac{1}{2}e^{is} - \frac{\hat{h}}{24}(41e^{2is} - 40e^{is} + 11) = 0 \implies \hat{h}(s) = \frac{24(e^{3is} - \frac{3}{2}e^{2is} + \frac{1}{2}e^{is})}{41e^{2is} - 40e^{is} + 11}, \quad s \in [0, 2\pi) \tag{14}$$

The locus of points for which $|r| = 1$, i.e. the curve $\hat{h}(s)$ in the complex plane (left plot of Figure (1)), divides the plane into two subdomains: red, and white. We thus take one value of $\hat{h}$ in each subregion (respectively $\hat{h} = -0.5$ and $\hat{h} = -1$), and as explained before, we find that the red region is the region of absolute stability.

1.3.2 Implicit method

The stability polynomial for our explicit method applied to $x' = \lambda x$ is

$$p(r) = r^3 - \frac{3}{2}r^2 - \frac{1}{2}r - \frac{\hat{h}}{72}(40r^3 + 3r^2 - 7), \quad \hat{h} = \lambda h \tag{15}$$

Substitution of $r = e^{is}$ into $p(r) = 0$, given by (15), leads to

$$e^{3is} - \frac{3}{2}e^{2is} + \frac{1}{2}e^{is} - \frac{\hat{h}}{72}(40e^{3is} + 3e^{2is} - 7) \implies \hat{h}(s) = \frac{72(e^{3is} - \frac{3}{2}e^{2is} + \frac{1}{2}e^{is})}{40e^{3is} + 3e^{2is} - 7}, \quad s \in [0, 2\pi) \tag{16}$$

The shape of $\hat{h}(s)$ is shown on the right plot of Figure (1). Plugging in two values of $\hat{h}$ for each subregion (e.g. respectively $\hat{h}=2$ and $\hat{h}=-1$), we find that the red region is the region of absolute stability.

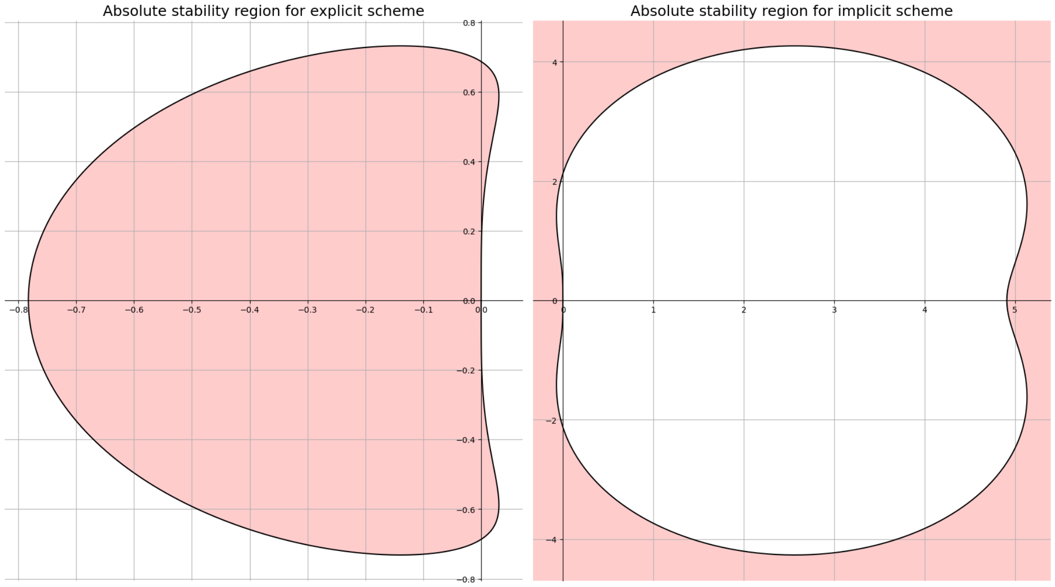

Figure 1: In the figure above the regions of absolute stability for both the explicit and implicit method are shown highlighted in red. This stability of each subregion has been analysed substituting one value within that subregion in the stability polynomial and then computing the magnitude of the roots (according to the boundary locus method). Eventually the interval of absolute stability is given by the intersection of the highlighted regions with the negative real-axis. For the explicit scheme this interval is approx $(-18/23, 0) \approx (-0.7826, 0)$ obtained for $r=-1$, whereas for the implicit scheme is given by $\mathbb{R} < 0$.

1.3.3 PECE

In order to find the stability polynomial we apply the method to the IVP $x' = \lambda x$

The predictor becomes

$$\hat{x}_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{\hat{h}}{24}\left(41x_n - 40x_{n-1} + 11x_{n-2}\right), \quad \hat{h} = \lambda h \tag{17}$$

whereas the corrector

$$x_{n+1} = \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{\hat{h}}{72}\left(\hat{x}_{n+1} + 3x_n - 7x_{n-2}\right), \quad \hat{h} = \lambda h \tag{18}$$

Plugging (17) into (18) we get:

$$\begin{aligned} x_{n+1} &= \frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{\hat{h}}{72}\left(40\left(\frac{3}{2}x_n - \frac{1}{2}x_{n-1} + \frac{\hat{h}}{24}\left(41x_n - 40x_{n-1} + 11x_{n-2}\right)\right) + 3x_n - 7x_{n-2}\right) \\ &= \left(\frac{3}{2} + \frac{7\hat{h}}{8} + \frac{205\hat{h}^2}{216}\right)x_n + \left(-\frac{1}{2} - \frac{5\hat{h}}{18} - \frac{25\hat{h}^2}{27}\right)x_{n-1} + \left(-\frac{7\hat{h}}{72} + \frac{55\hat{h}^2}{216}\right)x_{n-2} \end{aligned} \tag{19}$$

Therefore the stability polynomial for PECE method is

$$p(r) = r^3 - \left(\frac{3}{2} + \frac{7\hat{h}}{8} + \frac{205\hat{h}^2}{216}\right) r^2 - \left(-\frac{1}{2} - \frac{5\hat{h}}{18} - \frac{25\hat{h}^2}{27}\right) r - \left(-\frac{7\hat{h}}{72} + \frac{55\hat{h}^2}{216}\right) \tag{20}$$

Similarly as before, substituting of $r = e^{is}$ into $p(r) = 0$, given by (20), and rearranging, leads to the equation

$$\left(-\frac{205}{216}e^{2is} + \frac{25}{27}e^{is} - \frac{55}{216}\right)\hat{h}^2 + \left(-\frac{7}{8}e^{2is} + \frac{5}{18}e^{is} + \frac{7}{72}\right)\hat{h} + \frac{1}{2}e^{is} - \frac{3}{2}e^{2is} + e^{3is} = 0 \tag{21}$$

Solving this second order equation we get $\hat{h}_1(s)$ and $\hat{h}_2(s)$ that will lead us to the boundary of the absolute stability region as shown in Figure (2). For each of the subregions, we have substituted a value of $\hat{h}$ within that subregion into the stability polynomial and checked the norm of the its roots. We eventually find that the region of absolute stability is the one highlighted in red.

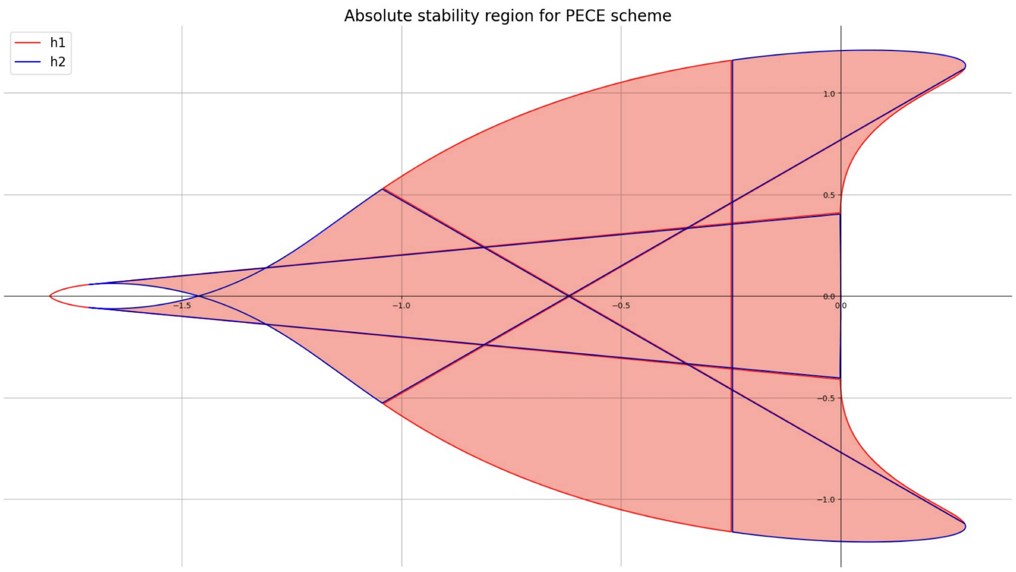

Figure 2: In the figure above the regions of absolute stability for both the explicit and implicit method are shown highlighted in red (coloring done by hand). Red and blue lines both represent the solution to (21). The interval of absolute stability is approximately $(-1.45, 0)$

1.3.4 Comparison

Comparing Figure (1) and Figure (2), we first notice that the shape of the absolute stability region of the PECE scheme is more complex than the one obtained for the implicit and explicit methods and it resembles more the one of the predictor. Also, coherently to the theory, we see that the interval of absolute stability, given by the intersection of the absolute stability region with the negative real-axis, for the PECE ($\approx (-1.42, 0)$) is larger that the one given by the predictor (here the explicit method) but smaller than the one given by the corrector (here the implicit method).

2 Question 2

The Robertson chemical reaction system is given by the following system of equations:

$$x' = -0.04x + 10^4 yz, \quad y' = 0.04x - 10^4 yz - 3 \cdot 10^7 y^2, \quad z' = 3 \cdot 10^7 y^2 \tag{22}$$

and parameters: $x(0) = 1$, $y(0) = z(0) = 0$, $t = [0, 100]$. For notation we write the above IVP as $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ where $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$, $\mathbf{x} = (x, y, z)$ and $\mathbf{x}_n$ represents the numerical solution at the nth step.

The algorithm is implemented according to the following steps:

1. Implementation of the function f that takes as input the vector (x, y, z) and outputs their derivative according to the given IVP (22).
2. Implementation of the predictor function that takes as input the matrix $(\mathbf{x}_{n-2}, \mathbf{x}_{n-1}, \mathbf{x}_n)$ and outputs $\hat{\mathbf{x}}_{n+1}$ according to (3). This function uses the f function described above.
3. Implementation of the corrector function that takes as input the matrix $(\mathbf{x}_{n-2}, \mathbf{x}_{n-1}, \mathbf{x}_n)$ and the vector $\hat{\mathbf{x}}_{n+1}$ and outputs $\mathbf{x}_{n+1}$ according to (4). This function uses the f function described above.
4. Initialization of the solution matrix with initial column given by the initial values of the given problem and implementation of the Euler method (twice) to get $\mathbf{x}_1$ and $\mathbf{x}_2$. (i.e. $\mathbf{x}_{i+1} = \mathbf{x}_i + h\mathbf{f}(t_i, \mathbf{x}_i)$, $i = 0, 1$).
5. Implementation of the for loop that at each time step performs the predictor function to get $\hat{\mathbf{x}}_{n+1}$ and then performs the corrector function to get $\mathbf{x}_{n+1}$. It eventually stores this value in the solution matrix and moves to the subsequent time step.

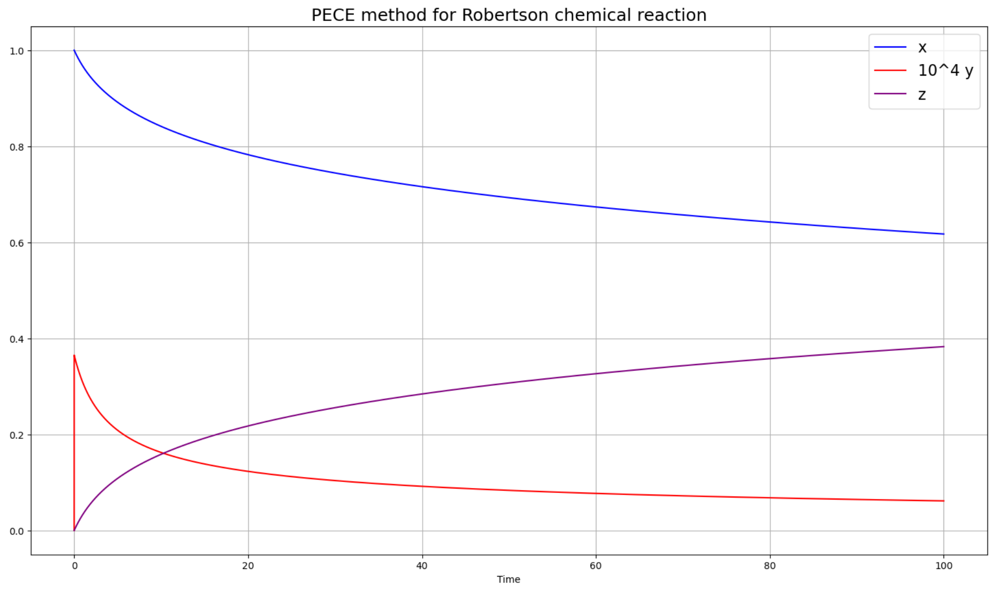

Figure 3: In the figure above the solution to Robertson chemical reaction, obtained by PECE method, is shown. Notice that the value of y has been multiplied by a factor 10^4 for visualisation purposes. Here $h = 10^{-4}$.

3 Question 3

The Rabinovich–Fabrikant system is given by the following system of equations

$$x' = y(z - 1 + x^2) + \gamma x, \quad y' = x(3z + 1 - x^2) + \gamma y, \quad z' = -2z(\alpha + xy) \tag{23}$$

and parameters: $x(0) = -1$, $y(0) = 0$, $z(0) = 0.5$, $\alpha = 1.1$, $\gamma = 0.87$, $t = [0, 50]$. For notation we write the above IVP as $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ where $\mathbf{x} = (x, y, z)$ and $\mathbf{x}_n$ represents the numerical solution at the n-th step.

3.1 Fixed Point iteration

To solve for $\mathbf{x}_{n+1}$ in the implicit numerical scheme on this non-linear set of ODEs we first use the fixed point iteration method. This means that, *at each time step*, we iterate the equation

$$\mathbf{x}_{n+1}^{i+1} = \frac{3}{2}\mathbf{x}_n - \frac{1}{2}\mathbf{x}_{n-1} + \frac{h}{72}\left(40\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}^i) + 3\mathbf{f}(t_n, \mathbf{x}_n) - 7\mathbf{f}(t_{n-2}, \mathbf{x}_{n-2})\right) \tag{24}$$

i-times until a stopping criterion is met, which here has been chosen as $\|\mathbf{x}_{n+3}^{i+1} - \mathbf{x}_{n+3}^i\|_2 = \|\mathbf{x}_{n+3}^{i+1}\|_2 < \epsilon$, where ϵ is a given tolerance. This is implemented using a *while loop* in Python, that checks whether the stopping criterion is met and inside it computes $\mathbf{x}_{n+1}^{i+1}$ given $(\mathbf{x}_{n-2}, \mathbf{x}_{n-1}, \mathbf{x}_n)$ and $\mathbf{x}_{n+1}^i$. The initial guess $\mathbf{x}_{n+1}^0$ is set equal to $\mathbf{x}_n$ at each n. The algorithm of the overall numerical scheme is once again implemented using a for loop that takes into account each time step performing the while loop of the FPIM. This 3-step method is initialised using forward Euler twice to compute $\mathbf{x}_1$ and $\mathbf{x}_2$. (i.e. $\mathbf{x}_{i+1} = \mathbf{x}_i + h\mathbf{f}(t_i, \mathbf{x}_i)$, $i = 0, 1$).

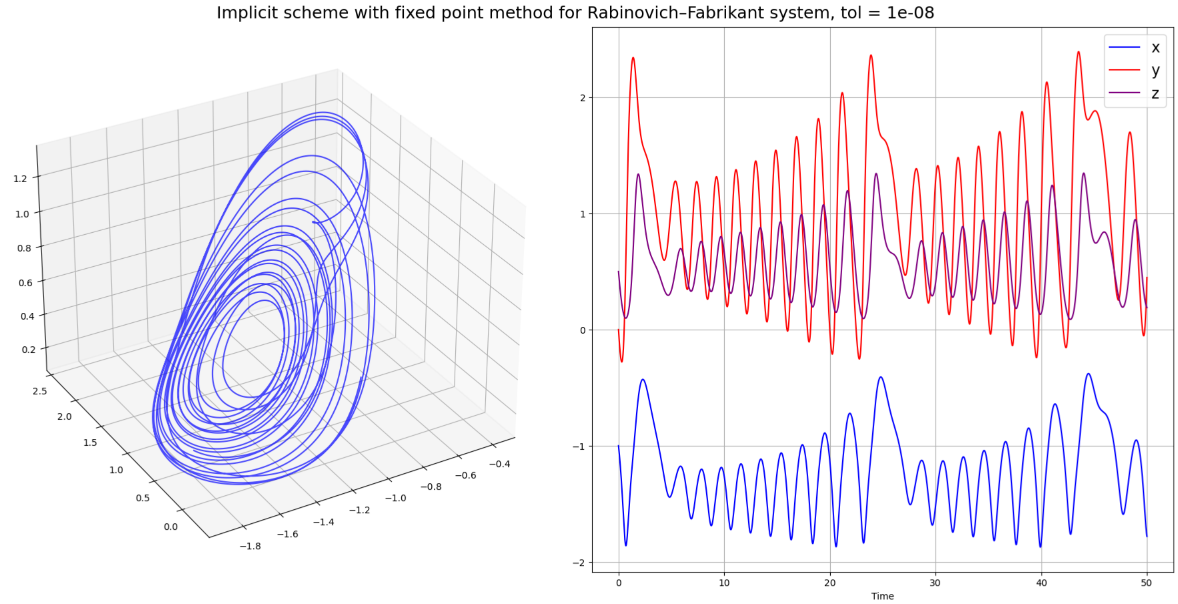

Figure 4: In the figure above the numerical solution to the Rabinovich–Fabrikant system, obtained through the implicit scheme paired with the fixed point iteration method, is shown both in a 3D space and against time. The value for tolerance used is $\epsilon = 10^{-8}$ and $h = 10^{-4}$.

3.2 Newton method

Another way to implement the implicit scheme for a non-linear system of equations is the Newton method. We write the equation we want to solve at each time step as $\mathbf{F}(\mathbf{x}_{n+1}; \mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, h) = 0$ where $\mathbf{x}_{n+1}$ is the implicit variable and $(\mathbf{x}_{n-2}, \mathbf{x}_{n-1}, \mathbf{x}_n, h)$ have fixed values. In this form, $\mathbf{F}$ becomes

$$\mathbf{F}(\mathbf{x}_{n+1}) = \mathbf{x}_{n+1} - \frac{3}{2}\mathbf{x}_n + \frac{1}{2}\mathbf{x}_{n-1} - \frac{h}{72}\left(40\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}) + 3\mathbf{f}(t_n, \mathbf{x}_n) - 7\mathbf{f}(t_{n-2}, \mathbf{x}_{n-2})\right), \quad n = 0, 1, 2, \ldots$$

where $\mathbf{f}$ is given by (23).

The Newton method at each time step n of the numerical scheme reads

$$\mathbf{x}_{n+1}^{i+1} = \mathbf{x}_{n+1}^{i} - (\mathbf{F}'(\mathbf{x}_{n+1}^{i}))^{-1}\mathbf{F}(\mathbf{x}_{n+1}^{i}), \quad i = 1, 2, \ldots \tag{25}$$

where

$$\mathbf{F}(\mathbf{x}_{n+1}^{i}) = \mathbf{x}_{n+1}^{i} - \frac{3}{2}\mathbf{x}_n + \frac{1}{2}\mathbf{x}_{n-1} - \frac{h}{72}\left(40\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}^{i}) + 3\mathbf{f}(t_n, \mathbf{x}_n) - 7\mathbf{f}(t_{n-2}, \mathbf{x}_{n-2})\right), \quad n = 0, 1, 2, \ldots$$

and the Jacobian (3×3 matrix obtained by differentiating $\mathbf{F}(\mathbf{x}_{n+1}^{i})$ w.r.t. $\mathbf{x}_{n+1}^{i}$):

$$\mathbf{F}'(\mathbf{x}_{n+1}^{i}) = \mathrm{Id}_3 - \frac{40h}{72}\mathbf{f}'(t_{n+1}, \mathbf{x}_{n+1}^{i}), \quad \mathbf{f}'(t, \mathbf{x}) = \begin{bmatrix} 2xy + \gamma & z - 1 + x^2 & y \\ 3z + 1 - 3x^2 & \gamma & 3x \\ -2zy & -2zx & -2(\alpha + xy) \end{bmatrix}$$

For this method we choose the same initial guess and stopping criterion as in fixed point iteration and same initialization procedure to find $\mathbf{x}_1$ and $\mathbf{x}_2$ (twice forward Euler). The results are shown in Figure (5).

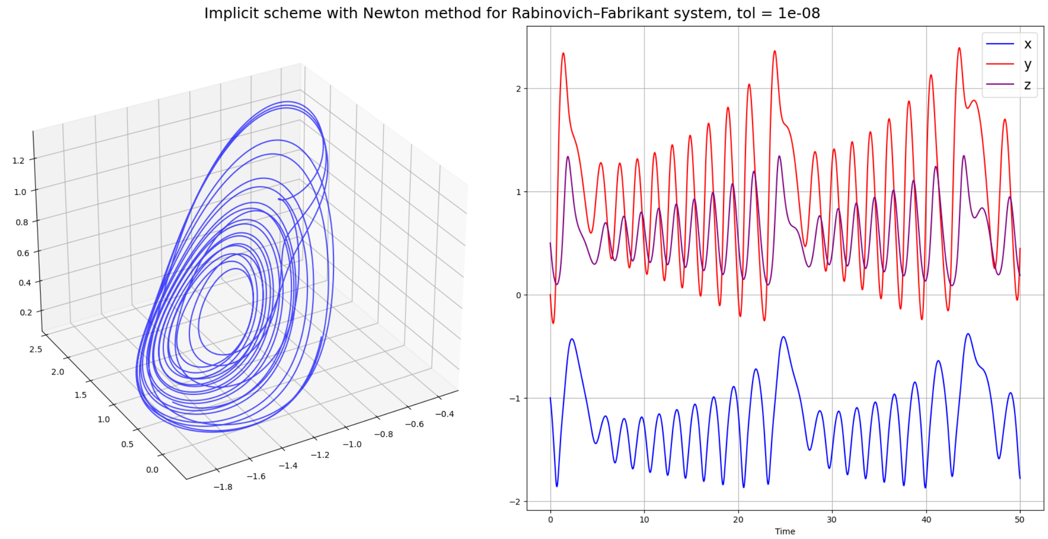

Figure 5: In the figure above the numerical solution to the Rabinovich–Fabrikant system, obtained through the implicit scheme paired with the Newton method. The value for tolerance used is $\epsilon = 10^{-8}$ and $h = 10^{-4}$.

3.3 Comparison and analysis

To begin with, both of the methods converges and no significant difference can be observed in the plots (Figure (4) and (5). At last, from Table (1) we draw the conclusion that the implicit scheme paired with Newton method is computationally slower (as might be expected as matrix inverse is computed) with respect to the fixed point iteration method. Yet, the latter requires a larger number of iteration to meet the required tolerance.

	Time (sec)		**Total iters**		**Avg. iters per step**	
Method	**FPI**	**Newton**	**FPI**	**Newton**	**FPI**	**Newton**
Tol = 10ˆ-7	35.13	44.69	1388353	999996	2.78	2.0
Tol = 10ˆ-13	49.94	66.49	1997259	1480612	3.99	2.96

Table 1: The table above compares different features (computational time, total number of iteration, average number of iterations) for Newton and fixed point iteration methods against a given tolerance.

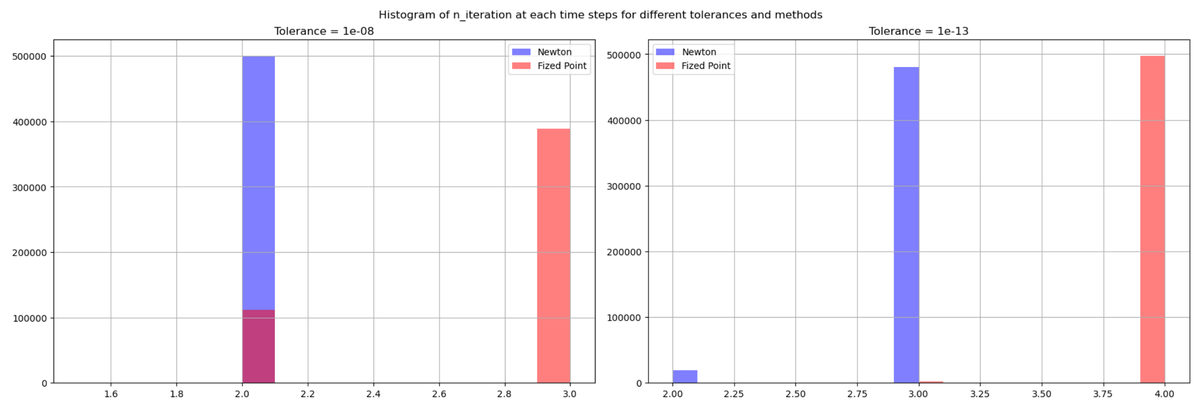

Figure 6: Histograms for different tolerances ϵ (in the left plot $\epsilon = 10^{-8}$, in the right one $\epsilon = 10^{-13}$) of the number of iteration occurring at each time step for Newton and Fixed Point Iteration methods.

4 Question 4

Given the explicit 3-step LMM

$$x_{n+3} + \alpha_2 x_{n+2} + \alpha_0 x_n = h\beta_0 f_n \tag{26}$$

Theorem 1. *[**Dahlquist equivalence theorem**] An LMM is convergent $\iff$ it is both consistent and zero-stable.*

Its first and second order characteristic polynomials are respectively:

$$\rho(r) = r^3 + \alpha_2 r^2 + \alpha_0 \tag{27}$$

$$\sigma(r) = \beta_0 \tag{28}$$

To enforce consistency we require $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$. This implies:

$$\alpha_2 + \alpha_0 = -1, \qquad 3 + 2\alpha_2 = \beta_0 \tag{29}$$

To enforce zero-stability we require the first-characteristic polynomial $\rho(r)$ to have roots such that $|r_i|_{i\in\{1,2\}} < 1$ or if any root $|r_k| = 1$ it has to be simple. We first rewrite $\rho(r)$ using (29), i.e. $\rho(r) = r^3 + \alpha_2 r^3 - 1 - \alpha_2$. The roots are:

$$r_1 = 1, \qquad r_{2,3} = \frac{1}{2}\left(-1 - \alpha_2 \pm \sqrt{\alpha_2^2 - 2\alpha_2 - 3}\right)$$

hence the root condition is satisfied when $|r_i|_{i\in\{2,3\}} \leq 1$ and $(r_i)_{i\in\{2,3\}} \neq 1$. We have to consider separately the cases where the square root is non-negative and negative. If non-negative the root condition is satisfied when $\alpha_2 \in (-3/2, -1]$, if negative $\alpha_2 \in (-1, 0]$. This leads us to choose α_2 such that

$$\alpha_2 \in \left(-\frac{3}{2}, 0\right] \tag{30}$$

After applying the method to $x' = \lambda x$, we write the stability polynomial $p(r) = \rho(r) - h\sigma(r)$ in terms of $\hat{h}$ and we use (29) to get:

$$\hat{h} = \frac{\rho(r)}{\sigma(r)} = \frac{r^3 + \alpha_2 r^2 + \alpha_0}{\beta_0} = \frac{r^3 + \alpha_2 r^2 - 1 - \alpha_2}{3 + 2\alpha_2}, \quad \hat{h} := \lambda h \tag{31}$$

Similarly to Question 1 we make use of the boundary locus method. Plugging $r = e^{is}$, $s \in [0, 2\pi)$ into (31) we obtain:

$$\hat{h}(s, \alpha_s) = \frac{e^{3is} + \alpha_2 e^{2is} - 1 - \alpha_2}{3 + 2\alpha_2} \tag{32}$$

We now seek the largest stability interval (given by the intersection of the region of absolute stability with the negative real axis) of the form $(\mathrm{Re}(\hat{h}), 0)$,

$$\begin{aligned}
\mathrm{Re}(\hat{h}(s, \alpha_s)) &= \frac{cos(3s) + \alpha_2 \cos(2s) - 1 - \alpha_2}{3 + 2\alpha_2}, \\
\mathrm{Im}(\hat{h}(s, \alpha_s)) &= 0
\end{aligned} \tag{33}$$

Since $e^{iks} = \cos(ks) + i \sin(ks)$,

$$\mathrm{Im}(\hat{h}(s, \alpha_s)) = 0 \implies \sin(3s) + \alpha_2 \sin(2s) = 0 \tag{34}$$

Solving for s and using the facts that $\mathrm{Re}(\hat{h}(s, \alpha_s))$ is even and periodic w.r.t. s with period 2π we get four possible $s(\alpha_2)$:

$$s_1(\alpha_2) = 2 \arctan\left(\sqrt{\frac{-5 - 2\sqrt{a^2+4}}{2\alpha_2 - 3}}\right), \quad s_2(\alpha_2) = 2 \arctan\left(\sqrt{\frac{-5 + 2\sqrt{a^2+4}}{2\alpha_2 - 3}}\right) \tag{35}$$

and $s_3(\alpha_2) = 0, \quad s_4(\alpha_2) = \pi$. We know that $\hat{h}(s_3(\alpha_2)) = 0$ which is the already known intersection between the boundary locus and the real axis. Furthermore, $\hat{h}(s_4(\alpha_2)) = \rho(-1)/\sigma(-1) = -2/(3 + 2\alpha_2)$ which is always an intersection point. It is now interesting to understand the behaviour of $\hat{h}(s_1(\alpha_2))$ and $\hat{h}(s_2(\alpha_2))$ with respect to $\alpha_2 \in (-3/2, 0]$. In the plots below, we have plotted these two function against α_2.

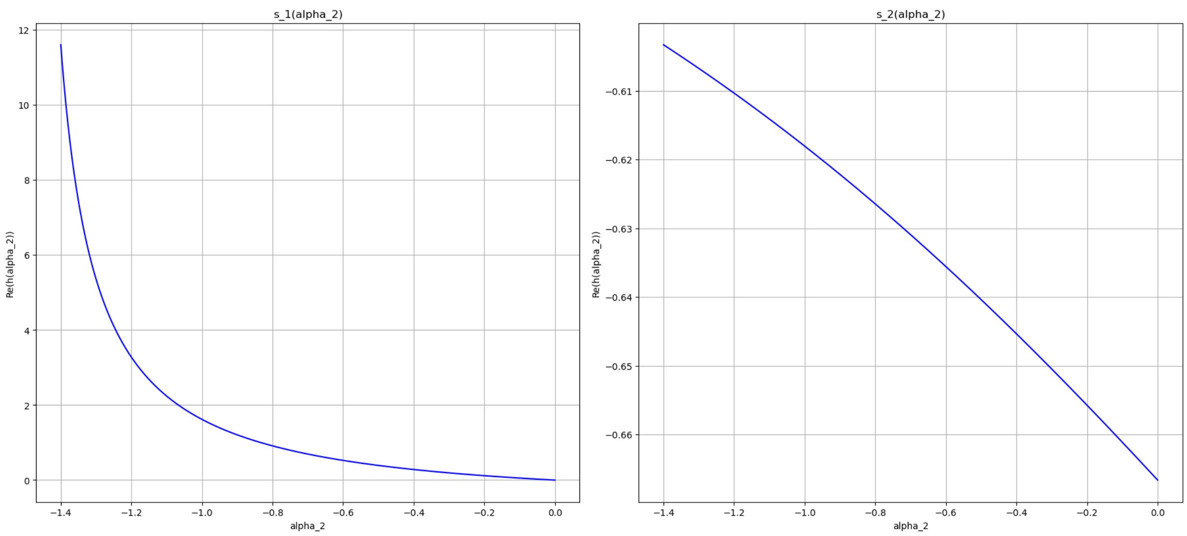

Figure 7: $\hat{h}(s_1(\alpha_2))$ and $\hat{h}(s_2(\alpha_2))$ for $\alpha_2 \in (-3/2, 0]$

Notice that:

1. $\lim_{\alpha_2 \to -3/2} \hat{h}(s_1(\alpha_2)) = +\infty$ and $\hat{h}(s_1(0)) = 0$.

2. $\hat{h}(s_2(0)) = -2/3$ which is the same as $\hat{h}(s_4(0))$.

3. Both $\hat{h}(s_1(\alpha_2))$ and $\hat{h}(s_2(\alpha_2))$ are monotonically decreasing.

4. if $\alpha_2 \neq 0$ we have a positive intersection (left plot of Figure (7) and two negative ones. These negative ones are one greater than $-2/3$ (as shown in the right plot of Figure (7) and one smaller given by $\hat{h}(s_4)$ (as this function, within the α_2 range, has maximum $-2/3$).

Now let's sum up and check the root condition for the stability polynomial $p(r)$:

1. **If** $\boldsymbol{\alpha_2 \in (-3/2, 0)}$ there are 4 intersections with the real axis; one of them is $\hat{h} = 0$ and the others, here denoted by $c_i,\ i \in \{1, 2, 3\}$ are such that $c_1 < -2/3 < c_2 < 0 < c_3$. Since it is trivial to show that the region outside the outer boundary cannot be absolutely stable, it is enough to show that $\hat{h} = -2/3$ does not satisfy the root condition to conclude that in this case the interval of absolute stability is $(c_2, 0) \subset (-2/3, 0)$.

2. **Else, if** $\boldsymbol{\alpha_2 = 0}$ there are only two intersections given by $\hat{h} = 0$ and $\hat{h} = -2/3$. Then we check that $\hat{h} = -0.5$ satisfies the root condition i.e. the roots of $p(r) = r^3 - 1 - 3 \cdot (-0.5) \cdot 3$ are smaller than 1 in absolute value.

Therefore, the largest interval of absolute stability is

$$\hat{h} \in \left(-\frac{2}{3}, 0\right) \tag{36}$$

obtained for $\alpha_2 = 0$, $\alpha_0 = -1$ and $\beta_0 = 3$ and the respective region of absolute stability is plotted below (on the left of Figure (8)).

In full, the method we have found is

$$x_{n+3} - x_n = 3hf_n \tag{37}$$

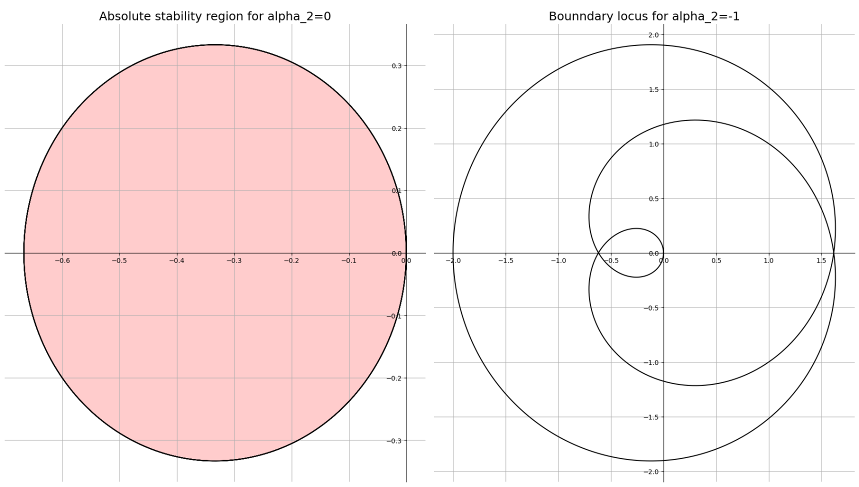

Figure 8: On the left: region of absolute stability for the LMM with $\alpha_2 = 0$. On the right: boundary locus for this LMM with $\alpha_2 = -1$.