Imperial College London
Department of Mathematics
Dr Igor Shevchenko

Numerical Solution of Ordinary Differential Equations

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

First Name .... Marwan

Family Name......... Riach

CID 01349928

Coursework 3

Before you start working on this coursework, please read coursework guidelines. Fill in this page and include it as a cover sheet to your report, otherwise the coursework will not be marked. 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 (LMM and Absolute Stability) **% of course mark: /3.0**

Find the largest interval of absolute stability for the LMM

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

and the coefficients α_2, α_0, β_0 which give this interval?

Exercise 2 (Absolute Stability and Global Error) **% of course mark: /3.0**

Develop the method with the set of α_2, α_0, β_0 from Exercise 1, and study how the global error behaves for time steps taken inside and outside of the interval of absolute stability when applied to the system of equations

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 998 & -999 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 2\sin(t) \\ 999(\cos(t) - \sin(t)) \end{pmatrix}, \quad \begin{pmatrix} x_1(t_0) \\ x_2(t_0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad t = (0, 20].$$

Exercise 3 (PECE) **% of course mark: /2.0**

Develop a predictor-corrector method based on an explicit predictor and an implicit corrector of your choice. Ensure that the global error of the method is at least $O(h^3)$.

Exercise 4 (PECE and Absolute Stability) **% of course mark: /2.0**

Find the region of absolute stability and the interval of absolute stability of the predictor-corrector method developed in Exercise 3.

Exercise 5 (PECE and Nonlinear Systems) **% of course mark: /5.0**

Solve the Lorenz system

$$\begin{cases} x' = \sigma(y - x), \\ y' = x(\rho - z) - y, \\ z' = xy - \beta z, \\ \sigma = 10, \ \beta = 8/3, \ \rho = 28, \ x(t_0) = y(t_0) = z(t_0) = 1, \ t = [0, 100] \end{cases}$$

 In hand date: 23:59 GMT, 13 Nov 2020 Due date: 23:59 GMT, 27 Nov 2020

with
(1) the method developed in Exercise 2 and
(2) the method developed in Exercise 3.

Exercise 6 (Implicit LMM and Nonlinear Systems) **% of course mark:** **/5.0***

Solve the Lorenz system

$$\begin{cases} x' = \sigma(y-x), \\ y' = x(\rho - z) - y, \\ z' = xy - \beta z, \\ \sigma = 10,\ \beta = 8/3,\ \rho = 28,\ x(t_0) = y(t_0) = z(t_0) = 1,\ t = [0, 100] \end{cases}$$

with the implicit LMM developed in Coursework 2. Use the Newton method to solve the nonlinear system of equations.

Coursework mark: **% of course mark**

In order to produce the plots and figures in this document, run the *exercise.m* files in the CW3 repository in MATLAB.

1 Exercise One (LMM and Absolute Stability)

Before attempting to find the largest interval of absolute stability across the three unknown variables, we can reduce the problem by making use of the fact that the method should be consistent and convergent; a LMM which is not consistent is of no use, and a convergent method is preferred given that we will be using it in subsequent exercises to obtain numerical solutions.

To account for the consistency of the method, we look at the first and second characteristic polynomials:

$$\rho(r) = r^3 + \alpha_2 r^2 + \alpha_0$$
$$\sigma(r) = \beta_0$$

For the LMM to be consistent we require the following:

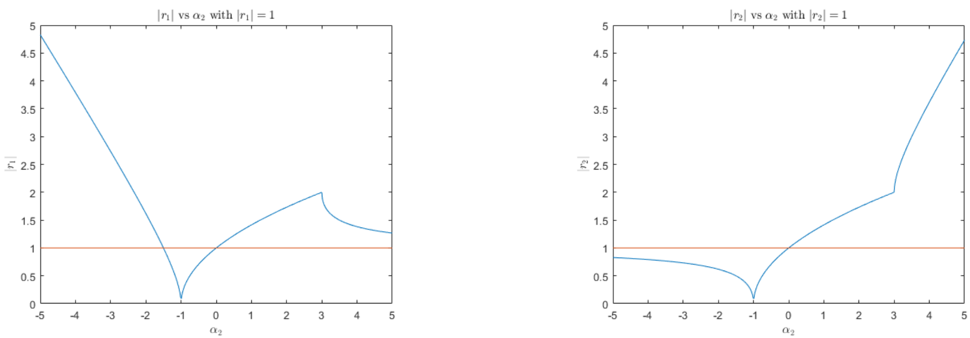
$$\rho(1) = 0$$
$$\rho'(1) = \sigma(1)$$
$$\Rightarrow \alpha_0 = -(1+\alpha_2),\ \beta_0 = 3 + 2\alpha_2$$

The LMM can now be rewritten as:

$$x_{n+3} + \alpha_2 x_{n+2} - (1+\alpha_2)x_n = h(3+2\alpha_2)f_n$$

To guarantee convergence, we require zero-stability. For this, we turn to the first characteristic polynomial of our simplified LMM and find the values of α_2 for which the root condition is obeyed:

$$\rho(r) = r^3 + \alpha_2 r^2 - 1 - \alpha_2 = (r-1)(r^2 + (1+\alpha_2)r + (1+\alpha_2))$$
$$\Rightarrow r = 1,\ \frac{-(1+\alpha_2) \pm \sqrt{(\alpha_2 - 3)(\alpha_2 + 1)}}{2}$$



(a) r_1 (b) r_2

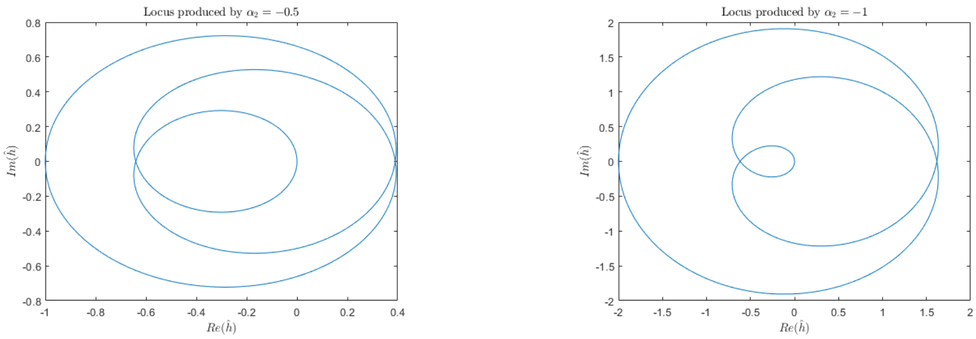
Figure 1: Modulus of Roots of ρ Vs α_2

Let us call the two roots that depend on α_2, r_1 and r_2 with them respectively referring to the root wth a plus and a minus. We require that the modulus of each of r is less than or equal to 1 and this is marked by the red lines in Figure 1. Looking at Figure 1, it can be seen that $\alpha_2 \in [-1.5, 0]$ satisfies the constraints for both r_1 and r_2 and across these α_2 values neither r value equals 1, ensuring all roots are simple.

To investigate which alpha value gives the largest interval, we will approach this numerically by using the stability polynomial and substitution $r = e^{is}$ for $s \in [0, 2\pi]$. To do this, we will produce some locus plots to see which segments of the loci form the region of absolute stability.

The stability polynomial is given as $p(r)$ and setting it to zero produces the following:

$$p(r) = r^3 + \alpha_2 r^2 - (1 + \alpha_2) - \hat{h}(3 + 2\alpha_2) = 0$$
$$\Rightarrow \hat{h} = \frac{r^3 + \alpha_2 r^2 - (1 + \alpha_2)}{3 + 2\alpha_2}$$
$$\Rightarrow \hat{h} = \frac{e^{3is} + \alpha_2 e^{2is} - (1 + \alpha_2)}{3 + 2\alpha_2}$$



(a) $\alpha_2 = -0.5$ (b) $\alpha_2 = -1$

Figure 2: The locus of points for which the stability polynomial has the root $|r| = 1$

The region of absolute stability appears to be the inner most region which intersects with the negative real axis beginning at $x = 0$. The length of the line segment that intersects with the negative real axis and is in the inner most region will be measured numerically by computing the horizontal distance between the staring point $(0, 0)$ and the first point at which the locus returns to an imaginary component of 0. This is done by considering the plot of the imaginary component of $\hat{h}$ and locating the first point at which it returns to zero and then extracting the corresponding real value. The code for this can be found in the *exercise*1.*m* file.

Below is a plot of the α_2 value vs the length of the interval of absolute stability:

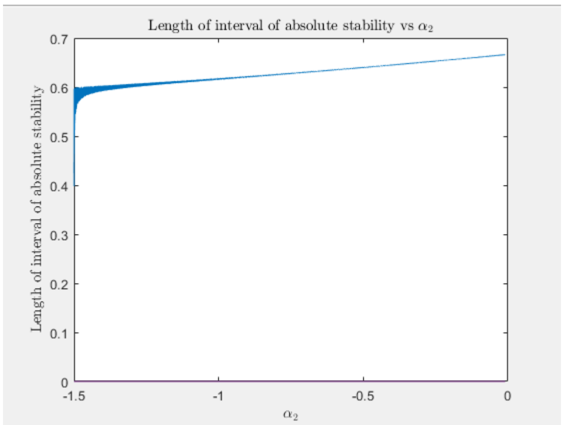


Figure 3: Length of interval vs α_2

As can be seen, $\alpha_2 = 0$ produces the interval of the greatest length.

To see what this length is, we can find the region of absolute stability analytically with $\alpha_2 = 0$:

$$p(r) = r^3 - 1 - 3\hat{h} = 0$$
$$\Rightarrow r = (1 + 3\hat{h})^{\frac{1}{3}}$$
$$\Rightarrow |(1 + 3\hat{h})^{\frac{1}{3}}| < 1$$
$$\Rightarrow |(1 + 3\hat{h})| < 1$$
$$\Rightarrow 3\hat{h} \in (-2, 0)$$
$$\Rightarrow \hat{h} \in (-\frac{2}{3}, 0)$$

This shows that the length of the interval is $\frac{2}{3}$ and that all values within this subdomain of the plane are absolutely stable. This theoretical length can be supported by Figure 3 which shows the length approaching 0.6667 as you approach 0. The coefficients which produce this interval are:

$$\alpha_2 = 0,\ \alpha_0 = -1,\ \beta_0 = 3$$

2 Exercise Two (Absolute Stability and Global Error)

The LMM in exercise 1 is carried out by obtaining the solutions at the first three times t_0, t_1, and t_2. t_0 is given by the initial condition and t_1, and t_2 are worked out by the Euler method and Adams Bashforth 2 method respectively.

The Euler method used is:

$$\mathbf{x_1} = \mathbf{x_0} + h\mathbf{f_0}$$

And the AB(2) method used is:

$$\mathbf{x_2} = \mathbf{x_1} + \frac{h}{2}(3\mathbf{f_1} - \mathbf{f_0})$$

where we use the $\mathbf{x_1}$ derived from the Euler method.

Then we use the LMM method in exercise 1 for the remainder of the time march:

$$\mathbf{x_{n+3}} = \mathbf{x_n} + 3h\mathbf{f_n}$$

We apply this to the following IVP and the code implementation can be seen in the $exercise2.m$ file in the MATLAB CW3 directory:

$$\begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix} = \begin{pmatrix} -2u(t) + v(t) + 2sin(t) \\ 998u(t) - 999v(t) + 999(cos(t) - sin(t)) \end{pmatrix},$$

$$\begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix},\ t \in [0, 20]$$

$$\mathbf{x_n} = \begin{pmatrix} u_n \\ v_n \end{pmatrix},\ \mathbf{f_n} = \begin{pmatrix} -2u_n + v_n + 2sin(t_n) \\ 998u_n - 999v_n + 999(cos(t_n) - sin(t_n)) \end{pmatrix}$$

From previous courseworks, we know that the numerical solution to this IVP is:

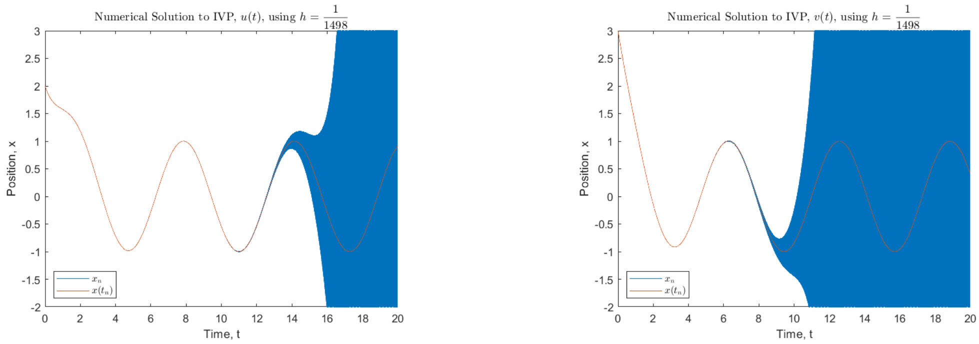
$$\mathbf{x} = 2e^{-t}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} sin(t) \\ cos(t) \end{pmatrix}$$

and that that eigenenvalues of the matrix in our IVP are λ = -1, -1000.

The interval of absolute stability for the LMM in exercise 1 is $\hat{h} \in (-\frac{2}{3}, 0)$. Since the system of ODEs is of order 2, we have to find to the smallest interval of absolute stability; this will be given by the eigenvalue of -1000.

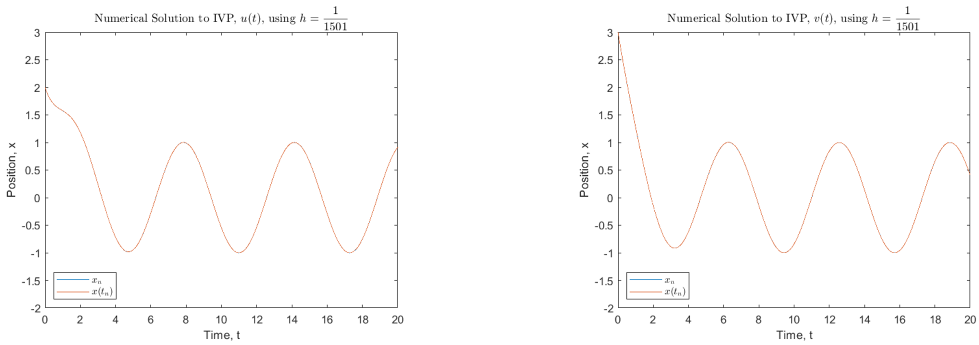
When $\lambda = -1000$, $\hat{h} \in (-\frac{2}{3}, 0) \Rightarrow h \in (0, \frac{2}{3000}) = (0, \frac{1}{1500})$, and so to investigate how the global error behaves, we will investigate with time steps just inside and just outside the given interval for h.

Below are plots of the global errors and the numerical solutions alongside the analytical solutions for $h = \frac{1}{1498}, \frac{1}{1501}$.



(a) $u(t)$ (b) $v(t)$

Figure 4: Behaviour of numerical solution against time for $h = \frac{1}{1498}$

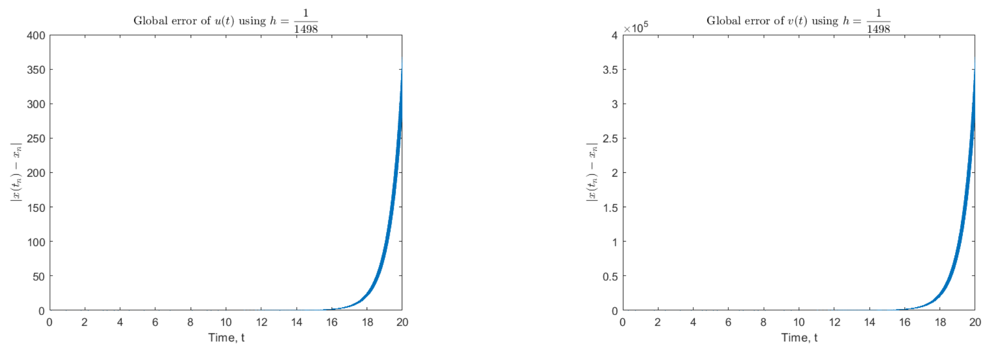


(a) $u(t)$ (b) $v(t)$

Figure 5: Behaviour of numerical solution against time for $h = \frac{1}{1501}$

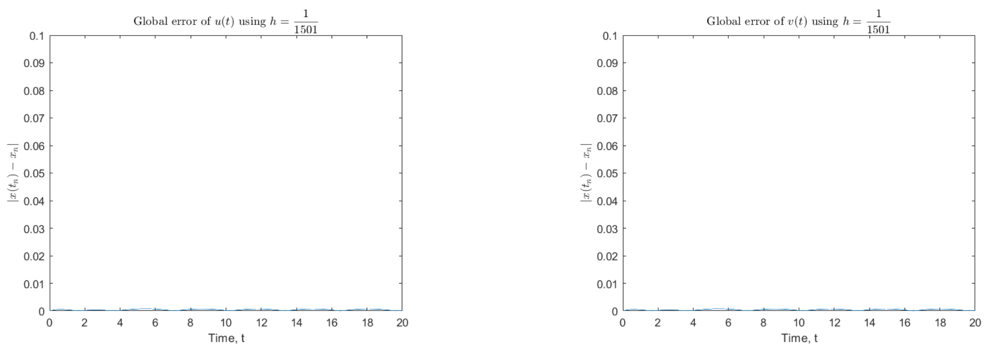
What can be seen in Figure 4 is that as time progresses, the numerical solution begins to diverge from the analytical solution in the method that uses $h = \frac{1}{1498}$. There is no divergence in the method which uses $h = \frac{1}{1501}$ according to Figure 5. This makes sense as $\hat{h}$ in this case falls within the theoretically derived interval of absolute stability when $h = \frac{1}{1501}$ and lies outside the interval of absolute stability when $h = \frac{1}{1498}$.

The figures overleaf show that the global error for $h = \frac{1}{1501}$ is negligible and does not increase with time and that for $h = \frac{1}{1498}$, the global error increases as you progress with time. This again supports our theoretical intervals as the LMM method exhibits instability when the time step is not small enough.



(a) $u(t)$ (b) $v(t)$

Figure 6: Global error of numerical solution against time for $h = \frac{1}{1498}$



(a) $u(t)$ (b) $v(t)$

Figure 7: Global error of numerical solution against time for $h = \frac{1}{1501}$

3 Exercise Three (PECE)

The predictor-corrector method we will develop will use the Adams-Bashforth 2 method as the predictor and the Adams-Moulton 3 method as the corrector:

$$\hat{x}_{n+1} = x_n + h\left(\frac{3}{2}f(t_n, x_n) - \frac{1}{2}f(t_{n-1}, x_{n-1})\right)$$
$$x_{n+1} = x_n + h\left(\frac{5}{12}f(t_{n+1}, \hat{x}_{n+1}) + \frac{2}{3}f(t_n, x_n) - \frac{1}{12}f(t_{n-1}, x_{n-1})\right)$$

Taking the continuous form of the predictor and corrector, we can work out the local truncation error of the predictor-corrector method. We will also use the fact that $f(t_n, x(t_n)) = x'(t_n)$:

$$\hat{x}(t_{n+1}) = x(t_n) + h\left(\frac{3}{2}x'(t_n) - \frac{1}{2}x'(t_{n-1})\right)$$
$$x(t_{n+1}) = x(t_n) + h\left(\frac{5}{12}\hat{x}'(t_{n+1}) + \frac{2}{3}x'(t_n) - \frac{1}{12}x'(t_{n-1})\right)$$

Plugging in the first equation into the second equation gives:

$$x(t_{n+1}) = x(t_n) + h\left(\frac{2}{3}x'(t_n) + \frac{5}{12}x'(t_n) - \frac{1}{12}x'(t_{n-1})\right) + h^2\left(\frac{5}{8}x''(t_n) - \frac{5}{24}x''(t_{n-1})\right)$$

Considering the Taylor expansion of the RHS of the previous equation gives:

$$x(t_{n+1}) = x(t_n) + hx'(t_n) + \frac{h^2}{2}x''(t_n) + \frac{h^2}{6}x^{(3)}(t_n) - \frac{13h^4}{144}x^{(4)}(t_n) + O(h^5)$$

The difference between this and the exact solution, $x(t+h)$, is $O(h^4)$. Thus the local truncation error is of order 4. Using this information, we can now deduce the order of the global error.

Prior to deducing the global error we shall assume that f and its derivative, f', are both Lipschitz continuous with Lipschitz constants L_0 and L_1 respectively. We will also rewrite the predictor-corrector method as follows:

$$x_{n+1} = x_n + h(\frac{13}{12}f_n - \frac{1}{12}f_{n-1}) + h^2(\frac{5}{8}f'_n - \frac{5}{24}f'_{n-1})$$

This makes the global error at time, t_{n+1}:

$$\begin{aligned} e_{n+1} = x(t_{n+1}) - x_{n+1} = x(t_n) - x_n + \frac{13h}{12}(f(t_n, x(t_n)) - f_n) - \frac{h}{12}(f(t_{n-1}, x(t_{n-1})) - f_{n-1}) \\ + \frac{5h^2}{8}(f'(t_n, x(t_n)) - f'_n) - \frac{5h^2}{24}(f'(t_{n-1}, x(t_{n-1})) - f'_{n-1}) + R_{n+1} \end{aligned}$$

where $R_{n+1} = O(h^4)$ since this is the order of convergence of the local truncation error.

We will now take the modulus of both sides and use the aforementioned Lipschitz constants which are such that:

$$\forall t \in [t_0, t_N],\ |f(t_n, x(t_n)) - f_n| \leq L_0|x(t_n) - x_n|,\ |f'(t_n, x(t_n)) - f'_n| \leq L_1|x(t_n) - x_n|$$

Using this information, we can now establish bounds on $|e_{n+1}|$ by taking the modulus and using the triangle inequality:

$$|e_{n+1}| \leq |e_n| + \frac{13h}{12}L_0|e_n| + \frac{h}{12}L_0|e_{n-1}| + \frac{5h^2}{8}L_1|e_n| + \frac{5h^2}{24}L_1|e_{n-1}| + |R_{n+1}|$$

Let $L = max(L_0, \sqrt{L_1})$ and let $\hat{h} = hL$. This now gives us the following bound on $|e_{n+1}|$:

$$|e_{n+1}| \leq |e_n| + \frac{13\hat{h}}{12}|e_n| + \frac{\hat{h}}{12}|e_{n-1}| + \frac{5\hat{h}^2}{8}|e_n| + \frac{5\hat{h}^2}{24}|e_{n-1}| + |R_{n+1}|$$

Introducing the error bounding function $\delta_n = \max|e_i|$, $0 \leq i \leq n$, $n \in [0, N]$ and rewriting the above in equality in terms of δ_n gives:

$$\delta_{n+1} \leq (1 + \frac{7\hat{h}}{6} + \frac{5\hat{h}^2}{6})\delta_n + |R_{n+1}|,$$

with $\delta_0 = 0$.
For $n = 1, 2, ..., N$, we find that:

$$\delta_N \leq \sum_{i=1}^{N}(1 + \frac{7\hat{h}}{6} + \frac{5\hat{h}^2}{6})^{N-i}|R_i|$$

In order to estimate $|\delta_N|$, we use the following inequality:

$$(1 + \frac{7\hat{h}}{6} + \frac{5\hat{h}^2}{6}) \leq (1 + 2\hat{h} + 2\hat{h}^2) \leq e^{2\hat{h}}$$
$$\Rightarrow (1 + \frac{7\hat{h}}{6} + \frac{5\hat{h}^2}{6})^{N-i} \leq e^{2\hat{h}(N-i)}$$

We will also make use of the fact that $|R_i| \leq Ch^4$ for some positive constant C since it represents the local truncation error and its order.

Putting all of the above information together yields:

$$\delta_N \leq \sum_{i=1}^{N} e^{2Lt_N}|R_i| \leq Ne^{2Lt_N}Ch^4 \leq e^{2Lt_N}Ch^3t_N,$$

since $t_N = Nh$ and hence $t_N L = NhL = N\hat{h}$.

This therefore means that:

$$\delta_N \leq Bh^3$$

where $B := e^{2Lt_N}Ct_N$.

Thus we have shown that the predictor-corrector method converges to the exact solution with order $p = 3$.

4 Exercise Four (PECE and Absolute Stability)

The LMM we are considering is given by:

$$x_{n+1} = x_n + h(\frac{13}{12}f_n - \frac{1}{12}f_{n-1}) + h^2(\frac{5}{8}f'_n - \frac{5}{24}f'_{n-1})$$

Shifting the index to the right by 1 gives:

$$x_{n+2} = x_{n+1} + h(\frac{13}{12}f_{n+1} - \frac{1}{12}f_n) + h^2(\frac{5}{8}f'_{n+1} - \frac{5}{24}f'_n)$$

Applying the LMM to the ODE $x' = \lambda x$ gives:

$$x_{n+2} = x_{n+1} + \hat{h}(\frac{13}{12}x_{n+1} - \frac{1}{12}x_n) + \hat{h}^2(\frac{5}{8}x_{n+1} - \frac{5}{24}x_n)$$

where $\hat{h} = \lambda h$.
The stability polynomial is then:

$$p(r) = r^2 - (1 + \frac{13}{12}\hat{h} + \frac{5}{8}\hat{h}^2)r + (\frac{\hat{h}}{12} + \frac{5}{24}\hat{h}^2)$$

Solving for $\hat{h}$ when the stability polynomial equals zero gives two roots for $\hat{h}$. They are as follows:

$$\hat{h}_{1,2} = \frac{\frac{13}{12}r - \frac{1}{12} \pm \sqrt{(\frac{13}{12}r - \frac{1}{12})^2 - 4(r^2 - r)(\frac{5}{24} - \frac{5}{8}r)}}{2(\frac{5}{24} - \frac{5}{8}r)}$$

Substituting $r = e^{is}$ for $s \in [0, 2\pi]$ produces the following locus:

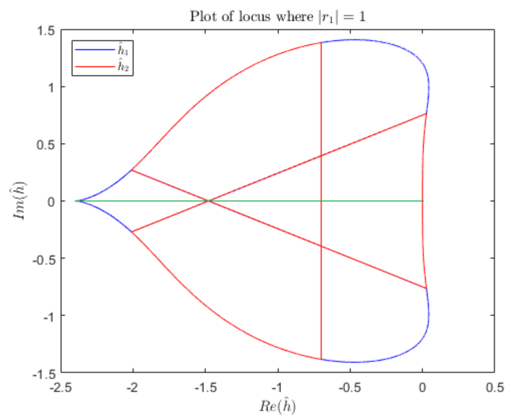


Figure 8: Locus of points for which the stability polynomial has the root $|r| = 1$. The solid green line represents the interval of absolute stability.

To see where the interval ends we will plot the maximum of the modulus of the roots produced as you vary $\hat{h}$ along the negative real axis. The roots are obtained using the *roots* function in MATLAB and this simply makes use of the quadratic formula, since our stability polynomial is a quadratic.

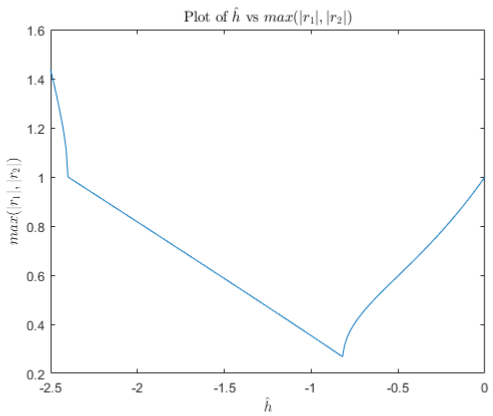


Figure 9: Plot showing that all roots satisfy the root condition when lying in the interval of absolute stability.

What can be seen here is that the point at which the interval of absolute stability comes to an end is when the real component of $\hat{h}$ is equal to -2.4. Thus, the interval of absolute stability is $\hat{h} \in (-2.4, 0)$.

To work out the region of absolute stability, we will pick points in the locus given by Figure 8 and see if the corresponding roots satisfy the root condition. Given that we know that the interval of absolute stability, we only need to check regions which do not contain the interval of absolute stability. We will consider the following $\hat{h}$ values: $-0.5+i, -0.5-i, -1.5+0.5i$, and $-1.5-0.5i$, as they cover all regions that do not contain the interval of absolute stability. We will find the modulus of the roots produced and return the biggest out of the two, again by making use of MATLAB's inbuilt functions. The outputs are as follows:

$$\hat{h} = -0.5+i : max(|r_1|,|r_2|) = 0.554$$
$$\hat{h} = -0.5-i : max(|r_1|,|r_2|) = 0.554$$
$$\hat{h} = -1.5+0.5i : max(|r_1|,|r_2|) = 0.722$$
$$\hat{h} = -1.5-0.5i : max(|r_1|,|r_2|) = 0.722$$

From what can be seen, all points within the regions that do not enclose the interval of absolute stability satisfy the root condition. Ergo, the whole of the locus shown in Figure 8 is the region of absolute stability (the love-heart shape).

5 Exercise Five (PECE and Nonlinear Systems)

The IVP is as follows:

$$\begin{cases} x' = \sigma(y-x), \\ y' = x(\rho - z) - y \\ z' = xy - \beta z \end{cases}$$

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

$$\sigma = 10, \beta = \frac{8}{3}, \rho = 28$$

The 3-step LMM used in exercise 2 is as follows:

$$\mathbf{x_{n+3}} = \mathbf{x_n} + 3h\mathbf{f_n}$$

$$\mathbf{x_n} = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}, \quad \mathbf{f_n} = \begin{pmatrix} 10(y_n - x_n) \\ x_n(28 - z_n) - y_n \\ x_n y_n - \frac{8}{3} z_n \end{pmatrix}, \quad \mathbf{x_0} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

This method is contingent on knowing $\mathbf{x_0}$, $\mathbf{x_1}$, and $\mathbf{x_2}$. The IVP only gives $\mathbf{x_0}$ and so in order to start the 3-step LMM, the Euler method is used to find $\mathbf{x_1}$ and AB(2) is used to find $\mathbf{x_2}$. The Euler method could have been used in order to find both $\mathbf{x_1}$, and $\mathbf{x_2}$, however AB(2) is more accurate and so this is used in the second step.

The Euler method used is:

$$\mathbf{x_1} = \mathbf{x_0} + h\mathbf{f_0}$$

And the AB(2) method used is:

$$\mathbf{x_2} = \mathbf{x_1} + \frac{h}{2}(3\mathbf{f_1} - \mathbf{f_0})$$

where we use the $\mathbf{x_1}$ derived from the Euler method.

The solution using the 3 step LMM is given below:

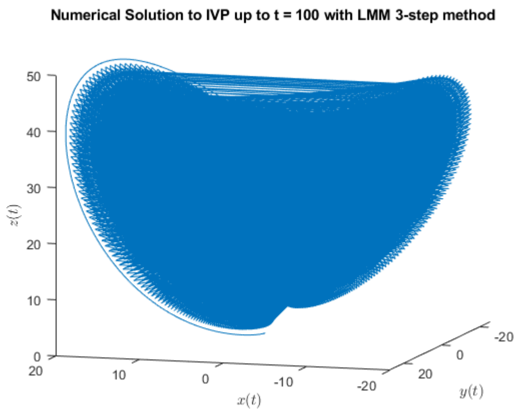


Figure 10: Solution to IVP using 3 step LMM and $h = 0.001$

For the method used in exercise 3, we only require knowledge of $\mathbf{x_0}$ and $\mathbf{x_1}$. We therefore use the Euler method to compute $\mathbf{x_1}$ and then proceed with the following method:

$$\hat{\mathbf{x}}_{\mathbf{n+2}} = \mathbf{x_{n+1}} + h\left(\frac{3}{2}f(t_{n+1}, \mathbf{x_{n+1}}) - \frac{1}{2}f(t_n, \mathbf{x_n})\right)$$
$$\mathbf{x_{n+2}} = \mathbf{x_{n+1}} + h\left(\frac{5}{12}f(t_{n+2}, \hat{\mathbf{x}}_{\mathbf{n+2}}) + \frac{2}{3}f(t_{n+1}, \mathbf{x_{n+1}}) - \frac{1}{12}f(t_n, \mathbf{x_n})\right)$$

$$\mathbf{x_n} = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}, \quad f(t_n, \mathbf{x_n}) = \begin{pmatrix} 10(y_n - x_n) \\ x_n(28 - z_n) - y_n \\ x_n y_n - \frac{8}{3}z_n \end{pmatrix}, \quad \mathbf{x_0} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

The solution using our predictor-corrector method is given below:

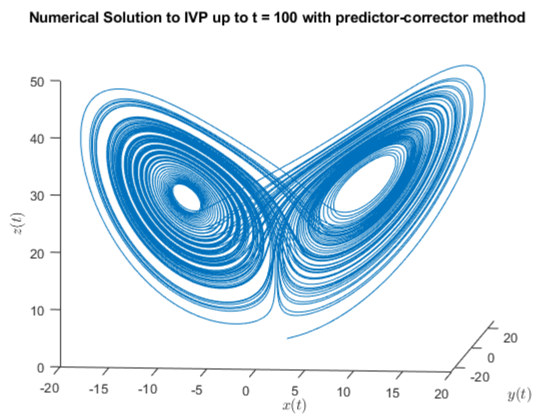


Figure 11: Solution to IVP using predictor-corrector method and $h = 0.001$

As can be seen, the solution that is obtained using the predictor-corrector method is far more accurate than the solution which uses the 3 step LMM obtained in exercise 2. This is because the global error of the predictor-corrector method is $O(h^3)$ and the global error of the 3 step LMM method is $O(h)$. In addition to this, the Lorenz system exhibits chaotic behaviour for our choice of parameters and hence a small error can lead to a huge change in the long term solution, as is shown by the difference in the interior of both plots; both outlines are the same, but the interior is different owing to the sensitivity of the system.

6 Exercise Six (Implicit LMM and Nonlinear Systems)

We will consider the same IVP as in exercise 5. The implicit LMM developed in coursework 2 is as follows:

$$\mathbf{x_{n+3}} = \frac{9}{10}\mathbf{x_{n+2}} + \frac{1}{10}\mathbf{x_{n+1}} + h(-\frac{151}{240}\mathbf{f_{n+3}} + \frac{923}{240}\mathbf{f_{n+2}} - \frac{757}{240}\mathbf{f_{n+1}} + \frac{83}{80}\mathbf{f_n})$$

$$\mathbf{x_n} = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}, \quad \mathbf{f_n} = \begin{pmatrix} 10(y_n - x_n) \\ x_n(28 - z_n) - y_n \\ x_n y_n - \frac{8}{3} z_n \end{pmatrix}, \quad \mathbf{x_0} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

In order to obtain a solution for $\mathbf{x_{n+3}}$, the Newton method will be used since we are presented with a system of nonlinear equations to solve. For the purposes of brevity, we will replace the coefficients with letters as follows:

$$\mathbf{x_{n+3}} = \alpha_2\mathbf{x_{n+2}} + \alpha_1\mathbf{x_{n+1}} + h(\beta_3\mathbf{f_{n+3}} + \beta_2\mathbf{f_{n+2}} + \beta_1\mathbf{f_{n+1}} + \beta_0\mathbf{f_n})$$

The $\mathbf{F}$ in our Newton method is given as:

$$\mathbf{F}(\mathbf{x_{n+3}}) = \mathbf{x_{n+3}} - (\alpha_2\mathbf{x_{n+2}} + \alpha_1\mathbf{x_{n+1}} + h(\beta_3\mathbf{f_{n+3}} + \beta_2\mathbf{f_{n+2}} + \beta_1\mathbf{f_{n+1}} + \beta_0\mathbf{f_n}))$$

The Jacobian matrix, $\mathbf{F}'(\mathbf{x_{n+3}})$, is given by:

$$\mathbf{F}'(\mathbf{x_{n+3}}) = \begin{pmatrix} 1 + 10h\beta_3 & -10h\beta_3 & 0 \\ h\beta_3(z_{n+3} - 28) & 1 + h\beta_3 & h\beta_3 x_{n+3} \\ -h\beta_3 y_{n+3} & -h\beta_3 x_{n+3} & 1 + \frac{8}{3}h\beta_3 \end{pmatrix}$$

This thus makes the Newton method:

$$\mathbf{x^{i+1}_{n+3}} = \mathbf{x^{i}_{n+3}} - \mathbf{F}'(\mathbf{x^{i}_{n+3}})^{-1}\mathbf{F}(\mathbf{x^{i}_{n+3}}), \; i = 0, 1, 2, ..., M.$$

The initial guess, $\mathbf{x^0}$, is the solution at the prior time step as we expect the solution (the Lorenz system) to be continuous, and so this initial guess will be sufficiently close to the root.

To speed up the implementation of the algorithm we will add a stopping criterion with a tolerance of $\epsilon = 10^{-5}$ to stop us always having to iterate through M times for each time step.

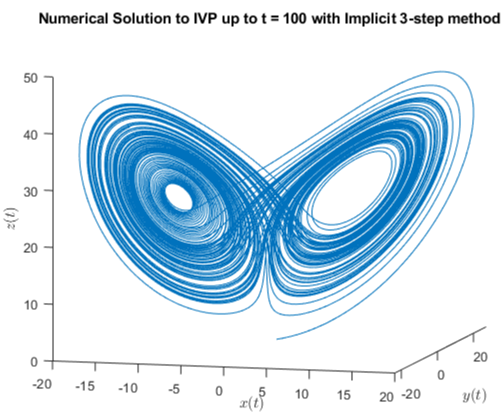


Figure 12: Solution to IVP using implicit method and $h = 0.001$ and $M = 5$