

© Igor V. Shevchenko (2022) This coursework is provided for the personal study of students taking this module. The distribution of copies in part or whole is not permitted.

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

Are you a Year 4 student?...

Coursework 1

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 (The Euler method for scalar ODEs) **% of course mark:** **/3.0**

- a) Apply the Euler method to the initial value problem

$$x' = \frac{1}{8}(5 - x - 5025e^{-8t}), \quad x(0) = 100, \quad t \in [0, 10]. \quad (1)$$

- b) Compute the numerical solution of (1) with the time steps $h = \{0.1, 0.05, 0.025\}$.
c) Compute the global error $|e_N|$ at $t = 10$ for $h = \{0.1, 0.05, 0.025\}$.

Exercise 2 (Taylor series methods for systems of ODEs) **% of course mark:** **/3.0**

- a) Apply the TS(3) method to the initial value problem

$$x'' + 3x' + 2x = t^2, \quad x(0) = 1, \quad x'(0) = 0, \quad t \in [0, 3]. \quad (2)$$

- b) Compute the numerical solution of (2) with the time steps $h = \{0.1, 0.05, 0.025\}$.
c) Find an approximated number of time steps such that the global error $|e_N| < 10^{-3}$ at $t = 3$.

Exercise 3 (Linear multistep methods for systems of ODEs) **% of course mark:** **/4.0**
This exercise is for Year-3 students only!

- a) Apply the 2-step Adams-Bashforth to the initial value problem (Chua's circuit)

$$\begin{cases} x' = \alpha(y - ax^3 - cx), \\ y' = x - y + z, \\ z' = -\beta y - \gamma z, \end{cases} \quad (3)$$

$$x(0) = 1, \quad y(0) = z(0) = 0, \quad t = [0, 100], \\ \alpha = 10, \quad \beta = 15, \quad \gamma = 0.01, \quad a = 0.1, \quad c = -0.2.$$

b) Compute the numerical solution of (3) with the time step $h = 0.01$.

c) Explain how you start the method.

Exercise 4 (Linear multistep methods for systems of ODEs) % of course mark: /6.5★

a) Derive the 3-step Adams-Bashforth method using Lagrange polynomials.

b) Calculate the order of the local truncation error of the method.

c) Apply the method to the Lorenz system

$$\begin{cases} x' = \sigma(y - x), \\ y' = x(\rho - z) - y, \\ z' = xy - \beta z, \\ \sigma = 10, \quad \beta = 8/3, \quad \rho = 28, \quad x(t_0) = y(t_0) = z(t_0) = 1, \quad t = [0, 100]. \end{cases} \quad (4)$$

d) Compute the numerical solution of (4) with the time step $h = 0.01$.

e) Explain how you start the method.

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 drop box 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 as one zip-file containing the pdf-report and MATLAB (m-files only) or Python (py-files only) code, both the report and the code should be in the directory named CID_Coursework#. The zip-file should be named CID_Coursework#.zip . The executable MATLAB (or Python) scripts for the exercises should be named as follows: exercise1.m, exercise2.m, etc.

1)
a) We have $x' = f(t, x)$ where $f(t, x) = \frac{1}{8}(5 - x - 5025e^{-8t})$
 $x(0) = 100 \quad t \in [0, 10]$

The Euler method is

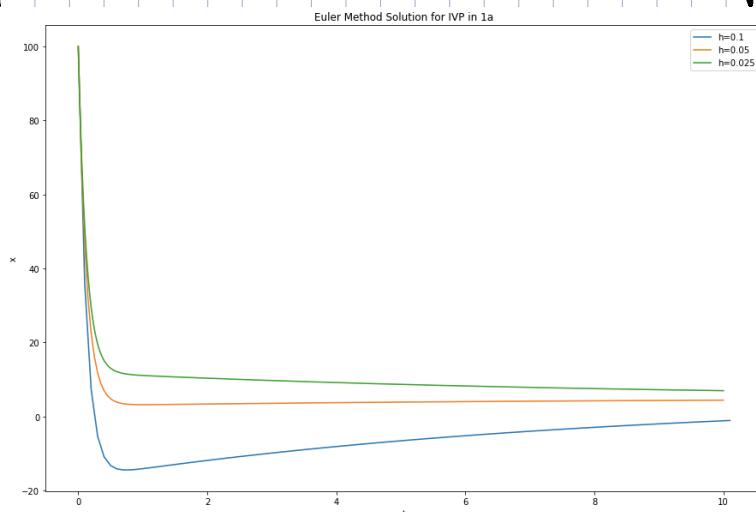
$$x_{n+1} = x_n + h f_n \quad \text{where } f_n = \frac{1}{8}(5 - x_n - 5025e^{-8t_n})$$

can also be written

$$x_{n+1} = x_n + \frac{h}{8}(5 - x_n - 5025e^{-8t_n}) \quad x_0 = 100, t_0 = 0$$

$$t_{n+1} = t_n + h$$

b) We compute the solution and visualize for each h .



We notice that as $h \downarrow 0$ the solutions tend to get flatter after $t=1$.

c) To compute global error we first need to solve the IVP analytically.

$$\frac{dx}{dt} + \frac{x}{8} = \frac{1}{8}(-5025e^{-8t} + 5)$$

$$\text{let } \mu(t) = e^{\int \frac{1}{8} dt} = e^{\frac{t}{8}}$$

Multiply both sides by $\mu(t)$

$$e^{\frac{t}{8}} \frac{dx}{dt} + \frac{1}{8} e^{\frac{t}{8}} \cdot x = -\frac{1}{8} e^{\frac{t}{8}} (5025e^{-8t} - 5)$$

$$e^{\frac{t}{8}} \frac{dx}{dt} + \frac{d}{dt}(e^{\frac{t}{8}})x = -\frac{1}{8}e^{\frac{t}{8}}(5025e^{-8t} - 5)$$

Apply the reverse product rule to LHS

$$\frac{d}{dt}(e^{\frac{t}{8}} \cdot x) = -\frac{1}{8}e^{\frac{t}{8}}(5025e^{-8t} - 5)$$

$$\int \frac{d}{dt} e^{\frac{t}{8}} \cdot x \, dt = \int -\frac{1}{8}e^{\frac{t}{8}}(5025e^{-8t} - 5) \, dt$$

$$e^{\frac{t}{8}} \cdot x = \frac{5}{8} \left(3e^{t/8} + \frac{2630}{21} e^{-\frac{63t}{8}} \right) + C$$

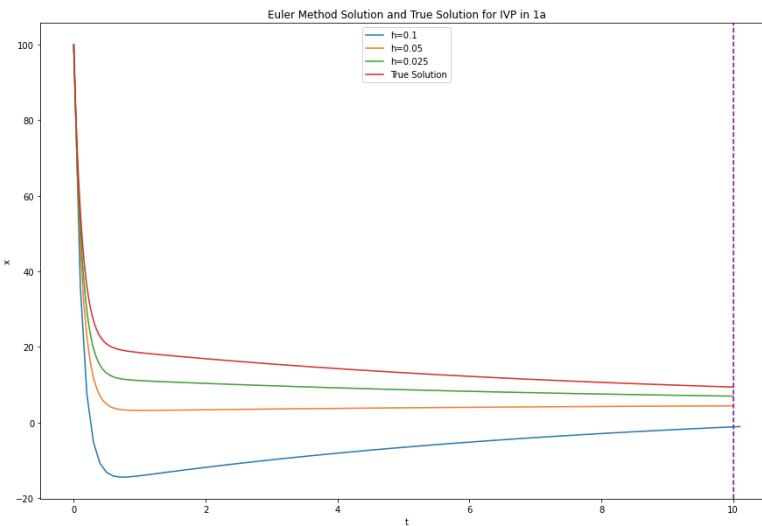
$$\Rightarrow x = \frac{1675e^{-8t}}{21} + C e^{-\frac{t}{8}} + 5$$

Plug in initial conditions:

$$C + \frac{1730}{21} = 100 \Rightarrow C = \frac{320}{21}$$

$$\Rightarrow x = \frac{5}{21} (64e^{-\frac{t}{8}} + 335e^{-8t} + 21)$$

We visualize the true solution to verify the Euler method solution is similar.



We notice that the solutions converge to the true solution as $h \downarrow 0$ and the difference between solutions is quite clear since with Euler we have convergence of $O(h)$.

We can also compute the global error $|e_N| = |x(t_N) - x_n|$ at $t=10$ for the given h .

h	$ e_N $
0.1	10.461190676644826
0.05	4.966152138155016
0.025	2.4066008156698206

We can see that as h gets smaller $|e_N|$ also decreases. This is a good verification of the implementation since we know the Euler method is convergent.

2)

a) In order to apply TS3 we convert the ODE to a first order system.

$$\begin{aligned} x = u & \Rightarrow u' = v & x(0) = 1 & \Rightarrow u(0) = 1 \\ x' = v & \Rightarrow v' = t^2 - 2u - 3v & x'(0) = 0 & \Rightarrow v(0) = 0 \end{aligned}$$

NOW we can write the TS3 method for the IVP

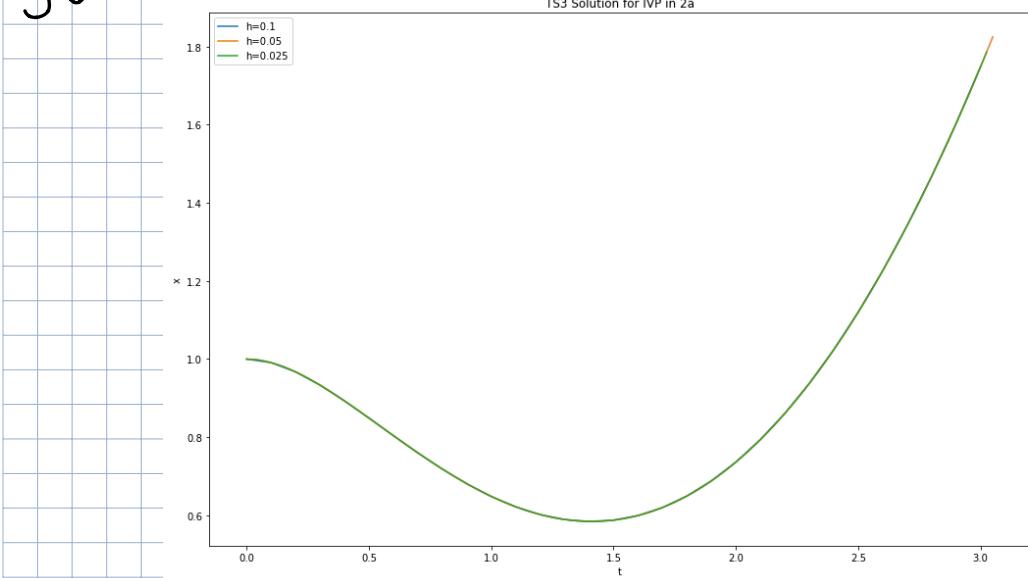
$$y_{n+1} = y_n + h f_n + \frac{h^2}{2!} f'_n + \frac{h^3}{3!} f''_n \quad \text{where } y_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}, y_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$f_n = \begin{pmatrix} v_n \\ t_n^2 - 2u_n - 3v_n \end{pmatrix} \quad f'_n = \begin{pmatrix} v'_n \\ 2t_n - 2u'_n - 3v'_n \end{pmatrix} \quad f''_n = \begin{pmatrix} v''_n \\ 2 - 2u''_n - 3v''_n \end{pmatrix}$$

$$\text{where } v''_n = 2t_n - 2v_n - 3v'_n \quad u''_n = t_n^2 - 2u_n - 3v_n$$

$$v'_n = t_n^2 - 2u_n - 3v_n \quad u'_n = v_n$$

b) We compute the solution and visualize for each h , by just taking the u values from y , since $x = u$.



The solutions overlap heavily. This is since with TS3 we have order 3 convergence hence we can afford to have less time steps (and so h a little larger than Euler) for good convergence.

c) First we must solve the ODE analytically:

Complementary solution:

Assume solution $\propto e^{\lambda t}$

$$\frac{d^2}{dt^2}(e^{\lambda t}) + 3 \frac{d}{dt}(e^{\lambda t}) + 2e^{\lambda t} = 0$$

$$(\lambda^2 + 3\lambda + 2)e^{\lambda t} = 0 \Rightarrow \lambda = -2, -1$$

$$x = C_1 e^{-2t} + C_2 e^{-t}$$

Particular solution:

$$\text{Assume } x_p = a_1 + a_2 t + a_3 t^2$$

$$\text{Substitute in } \Rightarrow 2a_3 + 3(a_2 + 2a_3 t) + 2(a_1 + a_2 t + a_3 t^2) = t^2$$

$$\text{equating coefficients } \Rightarrow a_1 = \frac{7}{4}, a_2 = -\frac{3}{2}, a_3 = \frac{1}{2}$$

$$\Rightarrow x = \frac{t^2}{2} - \frac{3t}{2} + C_1 e^{-2t} + C_2 e^{-t} + \frac{7}{4}$$

$$x' = t - 2c_1 e^{-2t} - c_2 e^{-t} - \frac{3}{2}$$

Plug in initial conditions

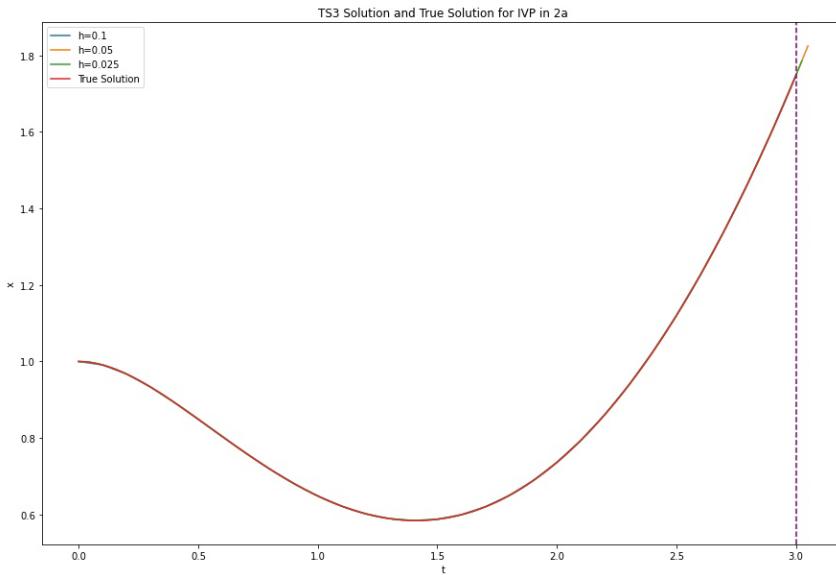
$$\Rightarrow c_1 + c_2 + \frac{7}{4} = 1, -2c_1 - c_2 - \frac{3}{2} = 0$$

$$\Rightarrow c_1 = -\frac{3}{4}, c_2 = 0$$

Hence the general solution is:

$$x = \frac{1}{4}(-3e^{-2t} + 2t^2 - 6t + 7).$$

We plot the true solution with the TS3 solution for the given h to verify TS3 solutions.



We can see that $h=0.1$ was already enough to have a near identical solution to the true one, again due to higher order (3) convergence of this method.

To find the number of time steps needed for $|E_n| < 10^{-3}$ at $t=3$ we try for $n=1, \dots, 30$ and use the relation $h = 3/n$ at $t=3$.

We obtain that for the following values of n that $|e_N| < 10^{-3}$.

N	h	$ e_N $
10	0.3333333333333333	0.009459484208271585
13	0.25	0.004937837225500896
14	0.23076923076923078	0.004124079097300948
15	0.21428571428571427	0.003492448864515163
16	0.2	0.0029931336954396315
17	0.1875	0.002592072402723966
20	0.15789473684210525	0.001771310901792944
24	0.13043478260869565	0.0011648955745104939
25	0.125	0.0010617144985360927
26	0.12	0.0009715450402125736
27	0.11538461538461539	0.0008923029696439322
28	0.1111111111111111	0.0008223028525846655
29	0.10714285714285714	0.0007601703205200039
30	0.10344827586206896	0.0007047759619065452

We see that 10 time steps is sufficient enough for $|e_N| < 10^{-3}$ at $t=3$. We would take the smallest n that satisfies the bound since this minimizes computational work. Of course, as n gets larger than all n will provide the bound, we see that after 24 steps all number of step methods give the bound. Overall 10 steps is relatively low which is due to higher order (3) convergence resulting in less computational work which is desirable.

4)

a) Consider $x' = f(t, x)$ $x(t_0) = \alpha$, $t \in [t_0, t_n]$

We can integrate over $[t_n, t_{n+1}]$ giving

$$\int_{t_n}^{t_{n+1}} x' dt = \int_{t_n}^{t_{n+1}} f(t, x) dt$$

$$\Rightarrow x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x) dt$$

To proceed with the numerical solution we approximate $f(t, x)$ with the Lagrange polynomial through t_n, t_{n-1}, t_{n-2} (in order to have 3 steps)

$$L(t) = \frac{(t-t_n)(t-t_{n-1})}{(t_{n-2}-t_n)(t_{n-2}-t_{n-1})} f_{n-2} + \frac{(t-t_{n-2})(t-t_n)}{(t_{n-1}-t_{n-2})(t_{n-1}-t_n)} f_{n-1} + \frac{(t-t_{n-2})(t-t_{n-1})}{(t_n-t_{n-2})(t_n-t_{n-1})} f_n$$

$$\int_{t_n}^{t_{n+1}} f(t, x) dt \approx \int_{t_n}^{t_{n+1}} d(t) dt$$

$$= \frac{(t - t_n)(t - t_{n-1})}{(t_{n-2} - t_n)(t_{n-2} - t_{n-1})} f_{n-2} + \frac{(t - t_{n-2})(t - t_n)}{(t_{n-1} - t_{n-2})(t_{n-1} - t_n)} f_{n-1} + \frac{(t - t_{n-2})(t - t_{n-1})}{(t_n - t_{n-2})(t_n - t_{n-1})} f_n dt$$

let $u = \frac{t - t_n}{h}$ $0 \leq u \leq 1$ s.t. $t = t_n + hu$ $dt = h du$

$$t - t_{n+1} = -h + hu, t - t_n = hu, t - t_{n-1} = h + hu, t - t_{n-2} = 2h + hu$$

Substituting new variable \Rightarrow

$$\begin{aligned} & \int_{t_n}^{t_{n+1}} \frac{(t - t_n)(t - t_{n-1})}{(t_{n-2} - t_n)(t_{n-2} - t_{n-1})} f_{n-2} + \frac{(t - t_{n-2})(t - t_n)}{(t_{n-1} - t_{n-2})(t_{n-1} - t_n)} f_{n-1} + \frac{(t - t_{n-2})(t - t_{n-1})}{(t_n - t_{n-2})(t_n - t_{n-1})} f_n dt \\ &= f_{n-2} \cdot h \int_0^1 \frac{1}{2} u(u+1) du + f_{n-1} \cdot h \int_0^1 -u(u+2) du + f_n \cdot h \int_0^1 \frac{1}{2}(u+1)(u+2) du \\ &= \frac{5}{12} h f_{n-2} - \frac{4}{3} h f_{n-1} + \frac{23}{12} h f_n \end{aligned}$$

So we obtain the AB(3) method

$$x_{n+1} = x_n + \frac{h}{12} (5f_{n-2} - 16f_{n-1} + 23f_n)$$

b) To compute the LTE we first rewrite AB(3) as

$$x(t_n + h) = x(t_n) + \frac{h}{12} (23x'(t_n) - 16x'(t_n - h) + 5x'(t_n - 2h))$$

now we find the two Taylor expansions in the above

$$x'(t_n - h) = x'(t_n) - h x''(t_n) + \frac{h^2}{2} x'''(t_n) - \frac{h^3}{6} x''''(t_n) + O(h^4)$$

$$x'(t_n - 2h) = x'(t_n) - 2h x''(t_n) + 2h^2 x'''(t_n) - \frac{4}{3} h^3 x''''(t_n) + O(h^4)$$

Now we plug this into the AB(3) to give

$$\begin{aligned} x(t_n + h) &= x(t_n) + \frac{h}{12} (23x'(t_n) - 16(x'(t_n) - h x''(t_n) + \frac{h^2}{2} x'''(t_n) - \frac{h^3}{6} x''''(t_n)) \\ &+ 5(x'(t_n) - 2h x''(t_n) + 2h^2 x'''(t_n) - \frac{4}{3} h^3 x''''(t_n)) + O(h^5) \end{aligned}$$

$$= x(t_n) + \frac{h}{12} (12x'(t_n) + 6hx''(t_n) + 2h^2 x'''(t_n) - 4h^3 x^{(IV)}(t_n)) + O(h^5)$$

$$= x(t_n) + hx'(t_n) + \frac{h^2}{2} x''(t_n) + \frac{h^3}{3!} x'''(t_n) - \frac{h^4}{3} x^{(IV)}(t_n) + O(h^5)$$

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2} x''(t) + \frac{h^3}{3!} x'''(t) + \frac{h^4}{4!} x^{(IV)}(t) + O(h^5)$$

hence $x(t+h) - x(t-h) = \frac{3h^4}{8} x^{(IV)}(t) + O(h^5) = O(h^4)$

where we use the localizing assumption

\Rightarrow LTE is of order 4.

c) let $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

Then we have $X' = f(X)$ where

$$f(X) = \begin{pmatrix} \sigma(y-x) \\ x(\rho-z) - y \\ xy - \beta z \end{pmatrix} \quad X(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

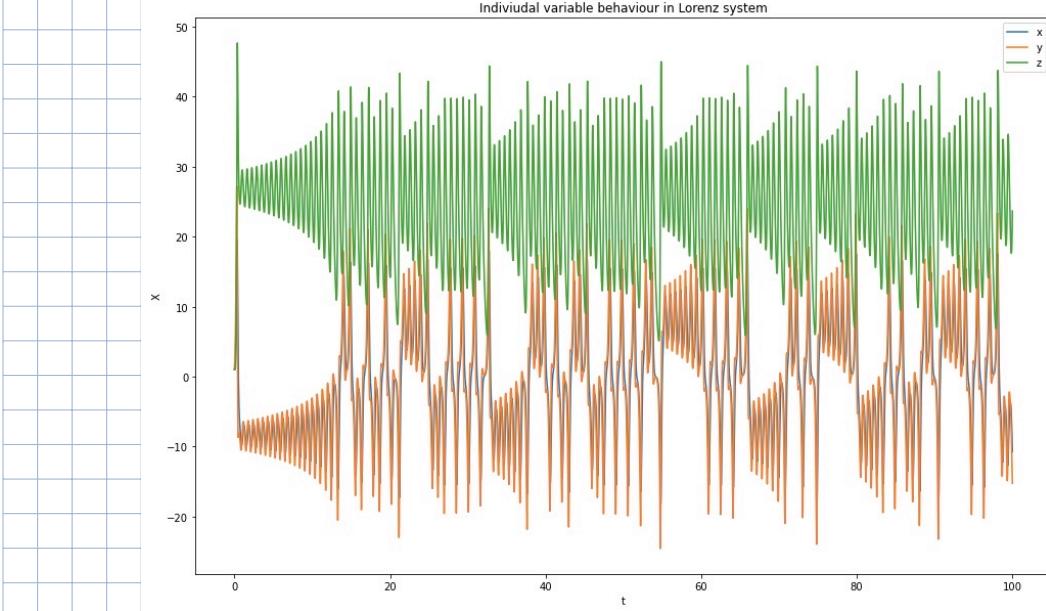
we apply AB(3) method to get

$$X_{n+1} = X_n + \frac{h}{12} (23f_n - 16f_{n-1} + 5f_{n-2}), \quad X_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad t_0 = 0$$

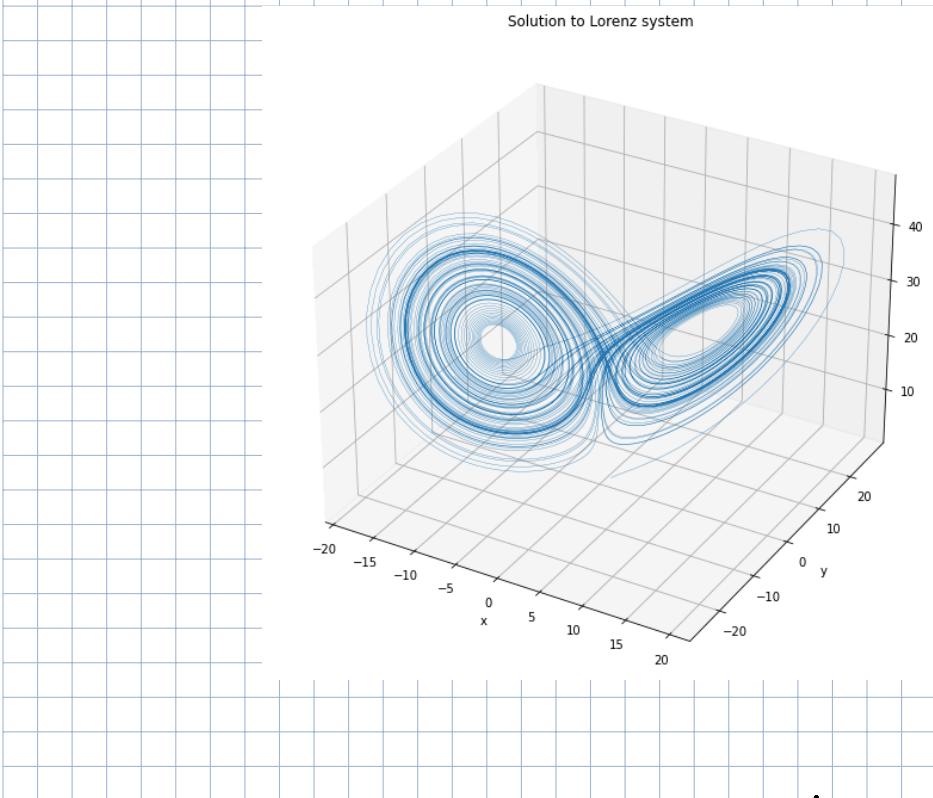
where $f_n = \begin{pmatrix} \sigma(y_n - x_n) \\ z_n(\rho - z_n) - y_n \\ x_n y_n - \beta z_n \end{pmatrix}, \quad f_{n-1} = \begin{pmatrix} \sigma(y_{n-1} - x_{n-1}) \\ z_{n-1}(\rho - z_{n-1}) - y_{n-1} \\ x_{n-1} y_{n-1} - \beta z_{n-1} \end{pmatrix},$

$$f_{n-2} = \begin{pmatrix} \sigma(y_{n-2} - x_{n-2}) \\ z_{n-2}(\rho - z_{n-2}) - y_{n-2} \\ x_{n-2} y_{n-2} - \beta z_{n-2} \end{pmatrix}, \quad \sigma = 10, \quad \beta = \frac{9}{3}, \quad \rho = 28$$

d) We compute the AB(3) method solution with $h=0.01$ and plot it for each individual variable.



We then plot in 3 dimensions and obtain the well known Lorenz attractor.



c) We start the method using the Euler method (used in the implementation in part d) as

$AB(3)$ is a 3 step method but we only have x_0 given ($y_0 = (1)$). So we use Euler to obtain x_1 and x_2 .

$$x_1 = x_0 + h f_0 = x_0 + h \cdot \begin{pmatrix} \sigma(y_0 - x_0) \\ z_0(\rho - z_0) - y_0 \\ x_0 y_0 - \beta z_0 \end{pmatrix} \text{ where } \sigma = 10, \beta = \frac{8}{3}$$

$$\rho = 28 \quad \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = X_0$$

$$\text{and then } x_2 = x_1 + h f_1 = x_1 + h \cdot \begin{pmatrix} \sigma(y_1 - x_1) \\ z_1(\rho - z_1) - y_1 \\ x_1 y_1 - \beta z_1 \end{pmatrix} \text{ where } \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = X_1$$

Then we are able to find

$$x_3 = x_2 + \frac{h}{12} (23f_2 - 16f_1 + 5f_0)$$

$$\text{where } f_2 = \begin{pmatrix} \sigma(y_2 - x_2) \\ z_2(\rho - z_2) - y_2 \\ x_2 y_2 - \beta z_2 \end{pmatrix}, \quad f_1 = \begin{pmatrix} \sigma(y_1 - x_1) \\ z_1(\rho - z_1) - y_1 \\ x_1 y_1 - \beta z_1 \end{pmatrix},$$

$$f_0 = \begin{pmatrix} \sigma(y_0 - x_0) \\ z_0(\rho - z_0) - y_0 \\ x_0 y_0 - \beta z_0 \end{pmatrix} \quad \text{using the } AB(3) \text{ method}$$

and all $x_n \ n \geq 4$ we also find using $AB(3)$

$$x_{n+1} = x_n + \frac{h}{12} (23f_n - 16f_{n-1} + 5f_{n-2})$$

$$\text{where } f_n = \begin{pmatrix} \sigma(y_n - x_n) \\ z_n(\rho - z_n) - y_n \\ x_n y_n - \beta z_n \end{pmatrix}, \quad f_{n-1} = \begin{pmatrix} \sigma(y_{n-1} - x_{n-1}) \\ z_{n-1}(\rho - z_{n-1}) - y_{n-1} \\ x_{n-1} y_{n-1} - \beta z_{n-1} \end{pmatrix},$$

$$f_{n-2} = \begin{pmatrix} \sigma(y_{n-2} - x_{n-2}) \\ z_{n-2}(\rho - z_{n-2}) - y_{n-2} \\ x_{n-2} y_{n-2} - \beta z_{n-2} \end{pmatrix}, \quad \sigma = 10, \beta = \frac{8}{3}, \rho = 28$$