Numerical Algorithms: Assignment 2

Marco Busato 852074

June 2020

Contents

1 Part 1 - Spring equation

In this part of the assignment is asked to solve a Cauchy problem related to a body of mass m connected to an horizontal spring by using the numerical integration methods seen in the course.
The methods involved are: the forward Euler method (described in **fe_scheme.m**), the backward Euler method (described in **be_scheme.m**), the Crank-Nicolson method (described in **cn_scheme.m**) and the built-in function of Matlab **ode45**.
Moreover for the implicit methods, is asked to implement a custom solver for system of nonlinear equation. My solution implements the Picard method because of its simplicity and its linear convergence (if converge).
The Cauchy problem can be written as a system of first order ODEs by considering the velocity as second variable:

$$\begin{cases} y'(t) = v(t) \\ v'(t) = -\frac{k}{m}y(t) \end{cases} \tag{1}$$

The analytical solution is:

$$y(t) = y_0 cos\left(\sqrt{\frac{k}{m}}t\right) \tag{2}$$

1.1 Results

Variables for setup the CP: t_0=0, T=15s, k=1$\frac{kg}{s^2}$, m=0.25kg, y_0=0.1m.
The time-step h=$2^{(-N)}$ with N=1,..,8
The result presented taking into account only the y variable.

- Comparison between analytical solution and the methods implemented when N=1 and N=8 shown in Figure 1:
- The convergence profiles of error e^N and error ratio $r^(N)$ using tolerance for the nonlinear solver τ=1.0e-9 is shown in Figure 2:
- The convergence profiles of error e^N and error ratio $r^(N)$ using tolerance for the nonlinear solver τ=1.0e-3 is shown in Figure 3:
- Table showing the number of timesteps, the CPU time and the average number of nonlinear iterations:

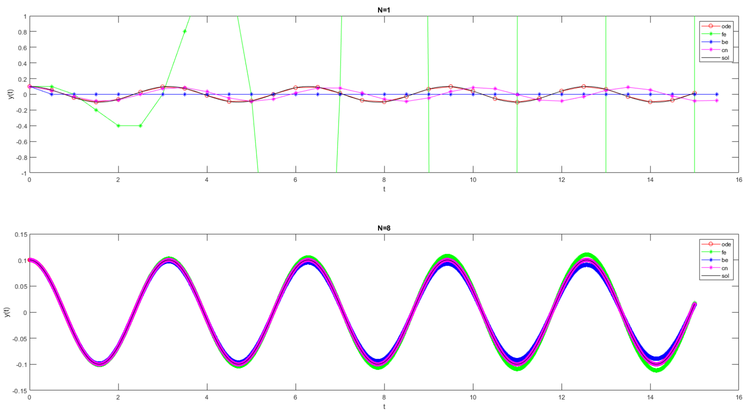

Figure 1: Numerical integration

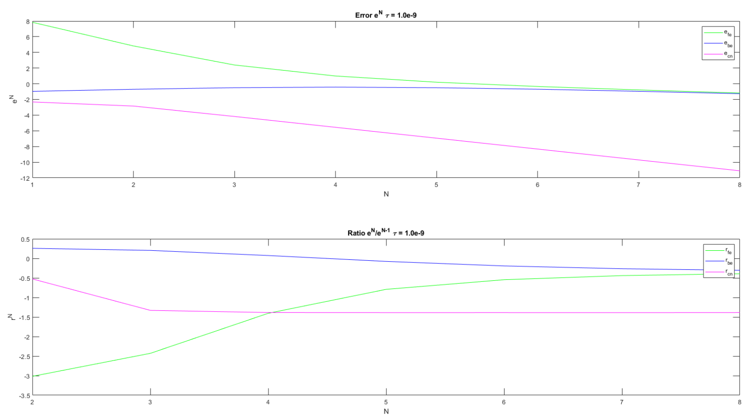

Figure 2: Error e^N and error ratio $\frac{e^N}{e^{N-1}}$ with τ=1.0e-9

h	Timesteps	FE_{CPU}	BE_{CPU}	BE_{IT}	CN_{CPU}	CN_{IT}
1	31	5.897000e-03	1.000090e-02	2.032258e+00	1.046000e-02	2.570968e+01
2	61	2.519000e-04	5.352700e-03	2.040984e+01	4.106600e-03	1.352459e+01
3	121	4.068000e-04	5.871500e-03	1.176033e+01	6.071300e-03	9.347107e+00
4	241	5.794000e-04	9.653300e-03	8.522822e+00	9.683400e-03	6.921162e+00
5	481	1.183800e-03	1.915900e-02	6.787942e+00	2.092880e-02	5.883576e+00
6	961	2.348400e-03	3.554910e-02	5.700312e+00	3.656370e-02	4.941727e+00
7	1921	3.716800e-03	6.178030e-02	4.871421e+00	6.590260e-02	3.991150e+00
8	3841	6.781600e-03	1.567969e-01	3.984900e+00	1.629268e-01	3.946889e+00

Table 1: Performance of the methods

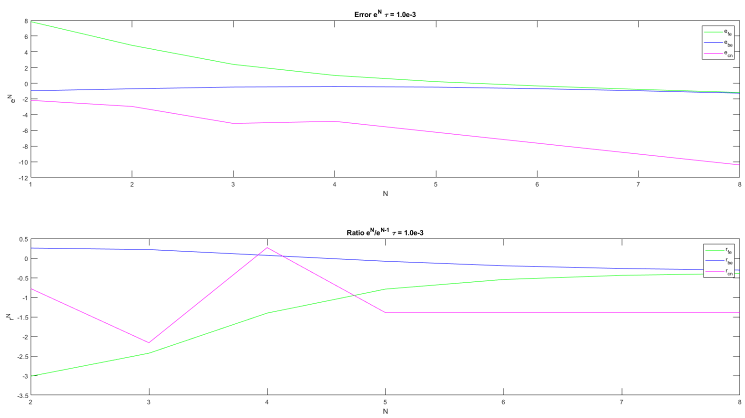

Figure 3: Error e^N and error ratio $\frac{e^N}{e^{N-1}}$ with τ=1.0e-3

1.2 Conclusions

As we can see from Figure 1 with N=1 the Forward Euler method produces some oscillations caused by the great timestep h considered.Therefore this method needs some assumption on h even if the Cauchy problem is Liapunov stable. For a smaller timestep the method nearly reach the same error of the BE scheme, in fact both methods nearly well approximate the considered function. The C-N scheme for N=8 seems the best method that approximate the function.
Finally the built-in function **ode45** in both cases well approximate the function, this because uses some internal method to evaluate the solution [1]

[1]See documentation at `https://it.mathworks.com/help/matlab/ref/ode45.html`

2 Part 2 - Epidemiological model (SEAIRD)

In this second part it has been required to study the SEAIRD model over the epidemiological data of the COVID-19 outbreak in Italy, in particular in the Marche region.
The model is available in **part_2.m** Matlab file.
The initial population considered N_p (set equal to 1,525,271) is taken from the ISTAT site[2]

2.1 Results

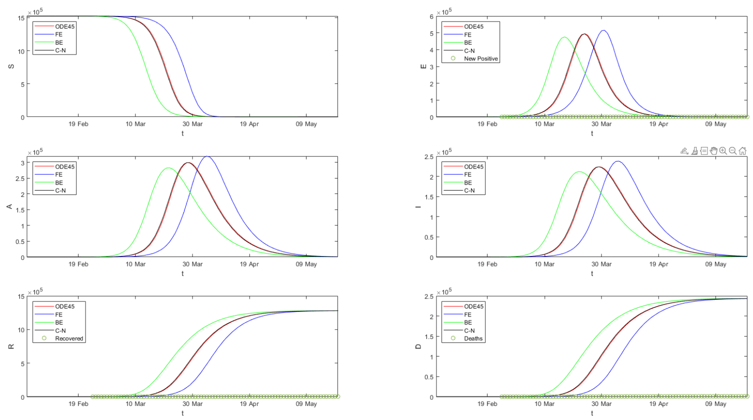

Figure 4: SEIARD model with N=0

As for the spring equation, the same analysis is performed and reported in Figure 4. For N=0, the results are far away from the predicted model, this because for the strong assumption given to the SEAIRD model such as the starting time of the epidemic on February 1^{st}.
Thus it can be considered a good model for possible outcomes if the Italian government had not implemented some restrictions, for example the entire lock down of the nation and social distancing.
The Figure 5 show the error of the method considered respect to the "real" solution provided by **ode45** Matlab function. Changing the tolerance does not change the results, so the plot for τ=1.0e-3 is omitted.
The Table 2 shows the performance of the three methods.

[2]Data here

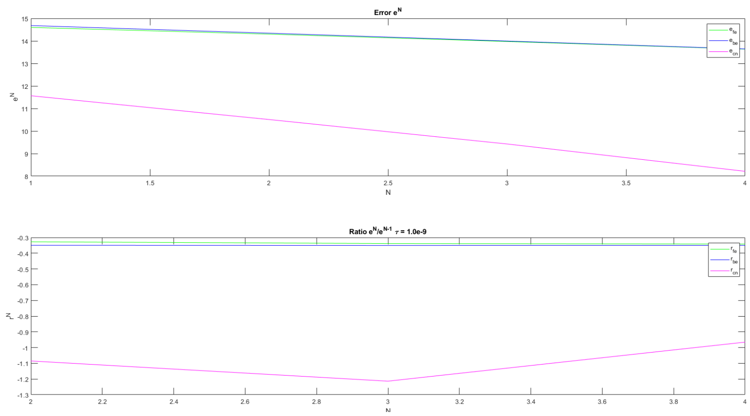

Figure 5: Errors and ratio of the errors respect to ode45 (τ=1.0e-9)

h	Timesteps	FE_CPU	BE_CPU	BE_IT	CN_CPU	CN_IT
1	219	1.171400e-03	1.774440e-02	1.467123e+01	1.529790e-02	1.073516e+01
2	437	1.867000e-03	2.609150e-02	1.050343e+01	2.349970e-02	8.437071e+00
3	873	2.923300e-03	4.172070e-02	8.286369e+00	4.660600e-02	7.119129e+00
4	1745	4.942700e-03	8.788430e-02	6.907163e+00	1.007325e-01	6.205731e+00

Table 2: Performance of the methods

2.2 Conclusions

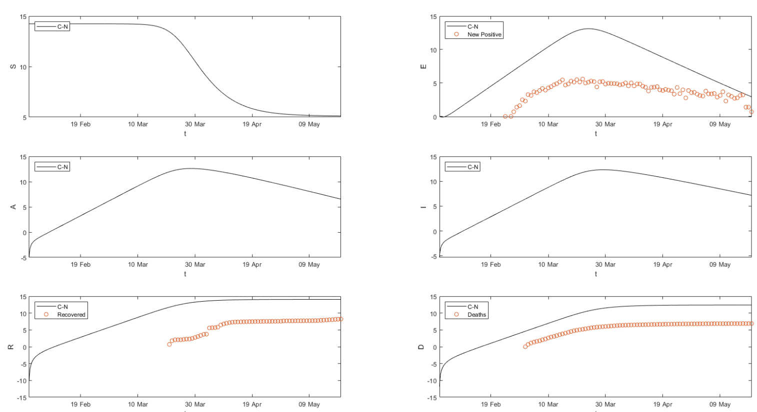

Figure 6: SEAIRD model with Crank-Nicolson method and N=4

Taking Cranck-Nicolson method as a reference to the outbreak without restrictions and N=4 there would have been at May 20^{th} 243,493 deaths against the 987 verified deaths.

The Table 2 shows the data of new positive, recovered and deaths up to the 20 of May.

May 20^{th}	Real data	C-N	Cases averted
New Positive	2	18	16
Recovered	3,716	1,279,601	1.275.885
Death	987	243,493	242.506

Table 3: Comparison between real data and the hypothetical data

Considering the cases without any restriction, on to the May 20^{th} there would have been about the 83% of the Marche population hospitalized.