Lab3

Elham Pilvar

December 2020

1 Question 1:

In this lab assignment we are going to study the backward and forward Euler methods for vector valued functions, where the backward Euler method uses Newton's iteration as a nonlinear solver. As it says, in each step of the backward Euler method, we must start the Newton iteration with the initial guess which is given by a step of the forward Euler method, and the stop condition is when the distance between two successive iterates falls within for example 10^{-4}.

2 Question 2:

In this part we are going to use the backward and forward Euler methods on a problem: $y' = \lambda y$, $y(0) = 1$ for three cases. For this problem we use time steps h=0.1, 0.05, 0.02, 0.01, 0.005.

a) in this part we will plot $log \|y(t_n) - y_n\|$ against $logh$ at the fixed time $t = 2$ where n satisfies $nh = 2$ and also $t \in [0, 2]$. it means that in each step we are computing the error between the exact and numerical solution at the final time for different fixed step-size h, as a function of h. as we can see in this graph...

b) In this part we are going to compare the two methods, the backward and forward Euler methods. for this reason we will compute the $log \|y(t_n) - y_n\|$ and we will plot this errors against N where N is the total number of function evaluation in the integration. we should not that for the forward Euler method N is equal to n, but for the backward Euler method it will be the total number of Newton iteration in the first n steps.

c) In this step we will consider a fixed h and we will plot $log \|y(t_n) - y_n\|$ against t_n.

2.0.1 Forward Euler Method : ($\lambda = -23$, $y_0 = 1$):

log (norm (error)) against log h:
The plotted graph shows that in this case the error is decreasing as h becomes

smaller in the sense of logarithmic scale. starting from 5 to -2 (approximately) in logarithmic scale

log (norm (error)) against N:
for the second graph, it can be seen that the graph is dramatically decreasing as we N becomes bigger, and then it starts to decrease but with a smooth slope.starting from 5 to 2 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
The error in this case is linearly decreasing. starting from -5 to -45 (approximately) in terms of logarithmic scale.

2.0.2 Forward Euler Method : ($\lambda = 1$, $y_0 = 1$):

log (norm (error)) against log h:
The error graph is linearly decreasing as h becomes smaller but with different slopes at each part of it. So, the error is decreasing. starting from 0 to -1.2 (approximately)in terms of logarithmic scale.

log (norm (error)) against N:
In this case, the error is decreasing, linearly not the whole graph, but if we consider it as a collection of some lines. starting from 0 to 1.2 in terms of logarithmic scale.

log (norm (error)) against x_n:
In this case, the error graph is increasing. starting from -11 to -3 in terms of logarithmic scale.

2.0.3 Forward Euler Method : ($y_0 = (1, 1)$) and λ is a matrix:

log (norm (error)) against log h:
The graph shows that as h gets smaller, the error at first is increasing then decreasing dramatically and at the end at very small h, it gets increased but with a very smooth slope. starting from 15 to 16 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
The graph in this case is increasing, then decreasing and after n=100 it starts to increase but with a very smooth slope.starting from 15 to 16 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
Obviously this graph is increasing. starting from -2.5 to 2 (approximately) in

terms of logarithmic scale.

2.0.4 Backward Euler Method : ($\lambda = -23$, $y_0 = 1$):

log (norm (error)) against log h:
The graph is fluctuating, meaning that it decreases for a while and then it increases but with a very smooth slope and then decreases again with a smooth slope and finally it increases dramatically as h becomes smaller.starting from 55 to 0, from 0 to 2 and then from 2 to 0 and finally from 0 to 25 in terms of logarithmic scale.

log (norm (error)) against N:
This graph is decreasing at first, and then it increases linearly and with a middle slope (not too sharp not too smooth). Starting from 0 to 20 in terms of logarithmic scale.

log (norm (error)) against x_n:
The graph is fluctuating, but it is linearly if we consider each part separately. The graph will stay constant after the fluctuation.it stays at 5 in terms of logarithmic scale.

2.0.5 Backward Euler Method : ($\lambda = 1$, $y_0 = 1$):

log (norm (error)) against log h:
The graph is increasing as h becomes smaller, so the error in this case is increasing. Starting from 18.5 to 25(approximately) in terms of logarithmic scale.

log (norm (error)) against N:
This graph(error) is increasing, meaning that the error is increasing.starting from 19 to 25(approximately) in terms of logarithmic scale.

log (norm (error)) x_n:
This graph(error) is also increasing. starting from 5 to 22.5 in terms of logarithmic scale.

2.0.6 Backward Euler Method : ($y_0 = (1, 1)$, and λ is a matrix):

log (norm (error)) against log h:
The error in this case is increasing as h gets smaller but with different slopes. starting from 8.7 to 11 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
The error in this case is also increasing as N gets bigger but with different slopes.

starting from 8.7 to 11 (approximately) in terms of logarithmic scale.

log (norm (error)) x_n:
In this case the error at first is increasing then it stays constant at 8.0 (approximately) in terms of logarithmic scale.

3 Question3:

In this part we will use the forward Euler code as a template, and then we implement the Runge's second order method and the four stage method. we must compare the performance of these methods with the forward and backward Euler methods on the three differential equations of the question 2.

3.0.1 Runge's 2nd Method : ($\lambda = -23$, $y_0 = 1$):

log (norm (error)) against log h:
The graph is decreasing at first, dramatically, then increasing smoothly, and then continue increasing with higher(larger) slope (or dramatically) as h becomes smaller. fluctuating between 1 and -3 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
at first, the error is decreasing, and then it starts to increase, with different slopes, and the slope is getting larger as N increases.starting from -0.8 to 0 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
The error is decreasing linearly. starting from 0 to -23 (approximately) in terms of logarithmic scale.

3.0.2 Runge's 4th Method : ($\lambda = -23$, $y_0 = 1$):

log (norm (error)) against log h:
In this case the error is increasing at first as h gets smaller, and then it decreases as well. starting from -0.65 to -0.35 (approximately) and -0.35 to -1.45 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
As illustrated in the graph, the error in this case is a little increasing and then decreases for most of the part. starting from -0.4 to -1.4 (approximately)in terms of logarithmic scale.

log (norm (error)) against x_n:
The error graph in this case is decreasing linearly, with a constant slope. starting from -2 to -50 (approximately) in terms of logarithmic scale.

3.0.3 Runge's 2nd Method : ($\lambda = 1$, $y_0 = 1$):

log (norm (error)) against log h:
The error in this case is also increasing, as h becomes smaller. starting from -3.6 to 2 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
This is an increasing graph, but with different slopes. starting from 2 to 3.6 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
This graph is at first decreasing and then suddenly it starts to increase with a sharp slope and then it continues to increase, starting from -12 to 2 in terms of logarithmic scale.

3.0.4 Runge's 4th Method : ($\lambda = 1$, $y_0 = 1$):

log (norm (error)) against log h:
In this case the error is decreasing with a constant slope as h gets smaller starting from -0.5 to -1.0 in terms of logarithmic scale.

log (norm (error)) against N:
In this case, the graph is decreasing,but with different slopes, starting from 0.5 to -1.0 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
In this case the graph is increasing starting from -5.25 to -3.25(approximately) in terms of logarithmic scale.

3.0.5 Runge's 2nd Method : (λ is a matrix, $y_0 = (1, 1)$

log (norm (error)) against log h:

as h gets smaller the error is increaing and after $logh = -3.9$ it increases with a very smooth slope. starting from 4 to 5 (approximately) in terms of logarithmic scale.

log (norm (error)) against N:
This error graph is increasing as N gets bigger, starting from 2.6 to 4 (approxi-

mately) in terms of logarithmic scale.

log (norm (error)) against x_n:
This graph is also increasing, starting from -6 to 2 (approximately) in terms of logarithmic scale.

3.0.6 Runge's 4th Method : (λ is a matrix, $y_0 = (1, 1)$

log (norm (error)) against log h:
As h gets smaller, the error at first decreases with 2 different slopes, then after $logh = -3.8$ it stays constant at 5 (approximately).

log (norm (error)) against N:
This error graph also stays constant, at 5 (approximately) in terms of logarithmic scale.

log (norm (error)) against x_n:
This graph illustrates that the error is increasing from -5 to 2 (approximately) in terms of logarithmic scale.

4 Comparison:

4.0.1 Q2$\lambda = -23$

log (norm (error)) against log h:
In my opinion, as shown on the graphs, The forward Euler method works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The forward Euler method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The forward Euler method works better.

4.0.2 Q2$\lambda = 1$

log (norm (error)) against log h:
In my opinion, as shown on the graphs, The forward Euler method works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The forward Euler method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The forward Euler method works better.

4.0.3 Q2$\lambda = (1, 1)$

log (norm (error)) against log h:
In my opinion, as shown on the graphs, The Backward Euler method works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The Backward Euler method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The forward Euler method works better.

4.0.4 Q3$\lambda = -23$

log (norm (error)) against log h:
In my opinion, as shown on the graphs, The 4th Runge's method works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The 4th Runge's method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The 4th Runge's method works better.

4.0.5 Q3$\lambda = 1$

log (norm (error)) against log h:
In my opinion, as shown on the graphs, The 4th Runge'smethod works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The 4th Runge's method method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The 4th Runge's method method works better.

4.0.6 **Q3**$\lambda = (1, 1)$

log (norm (error)) against log h:

In my opinion, as shown on the graphs, The 4th Runge's method works better.

log (norm (error)) against N:
In my opinion, as shown on the graphs, The 2nd Runge's method works better.

log (norm (error)) against x_n:
In my opinion, as shown on the graphs, The 2nd Runge's method works better. although both methods' final error is the same but I think the 2nd Runge's starts at very smaller error with respect to 4th Runge's.