**MA 3457 / CS 4033**

**Conference on 12/4**

1. In class, we have been able to derive forward Euler from slope definitions, from a Taylor series expansion, as well as by integrating both sides of the IVP.



Text, letter

Description automatically generated

Text, letter

Description automatically generated

2.

Text, letter

Description automatically generated

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Total Error (sum(exact-approx)) | | | Elapsed Time (seconds) | | |
| step size h | Taylor Method Order 2 | Runge Kutta Order 2 | Implicit Euler Method | Taylor Method Order 2 | Runge Kutta Order 2 | Implicit Euler Method |
| 0.05 | 0.415094 | 0.325391 | 13.73329 | 0.000074 | 0.000059 | 0.000053 |
| 0.1 | 0.841109 | 0.655126 | 15.77001 | 0.000056 | 0.000044 | 0.000039 |
| 0.2 | 1.720148 | 1.323737 | 20.84934 | 0.000043 | 0.000037 | 0.000032 |
| 0.5 | 4.492491 | 3.365712 | 52.28256 | 0.000044 | 0.000031 | 0.000027 |

Table 1: Error and Computation Time using different step size and methods.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Approx. Value at t = 1.4 | | |  | Error at t = 1.4 | | |
| step size h | Taylor Method Order 2 | Runge Kutta Order 2 | Implicit Euler Method | real value | Taylor Method Order 2 | Runge Kutta Order 2 | Implicit Euler Method |
| 0.05 | 2.6099 | 2.6119 | 2.9404 | 2.62036 | 0.01046 | 0.00846 | 0.32004 |
| 0.1 | 2.581 | 2.5888 | 3.3084 | 2.62036 | 0.03936 | 0.03156 | 0.68804 |
| 0.2 | 2.4805 | 2.5096 | 4.2335 | 2.62036 | 0.13986 | 0.11076 | 1.61314 |

Table 2: Approximated Values at t =1.4 and corresponding errors.

Error Analysis: Using any step size, it would appear that Runge Kutta of Order 2 yields the smallest error difference. As expected, using smaller step sizes, decreases the total error for each method. It also looks like doubling the step size would give us double the total error for both the Taylor Method and Runge Kutta of Order 2. For the implicit Euler method, there are a lot of local truncation error at each approximation and it adds up, which is why it gives such a high amount of total error. From Table 2, we can see that doubling the step size increased the error by approx. 4 times more for both the Taylor Method and Runge Kutta Order 2. This is because the local truncation error is O(h2) and doubling the step size will give us 4 times as much LTE. Euler’s Method’s LTE is O(h) so doubling it will double the error and it can be seen from table 2 that the error did double when the step size is doubled.

Computation Time Analysis: Increasing the step size also increases the computation time because there will be more computations. Out of the three methods used, the Implicit Euler Method is the fastest, mainly because it only contains 1 line of function evaulation for each approximation, since I have directly solved for the update rule. The other two requires a bit more computations, thus the longer computational time..

Stability Analysis: The sensitivity of the step size is shown in Table 1. It seems that the total error is increasing for all three methods and it makes sense for the Taylor method and RK2 because they are explicit methods and therefore are conditionally stable. When the step size becomes too large, the error grows, and only small step size would yield a stable solution. The implicit Euler method should be unconditionally stable, but the error seems to grow for increasing step sizes like the other two methods.