Numerical Methods; Sklyar, S. N.
Moritz Konarski
25.05.2019

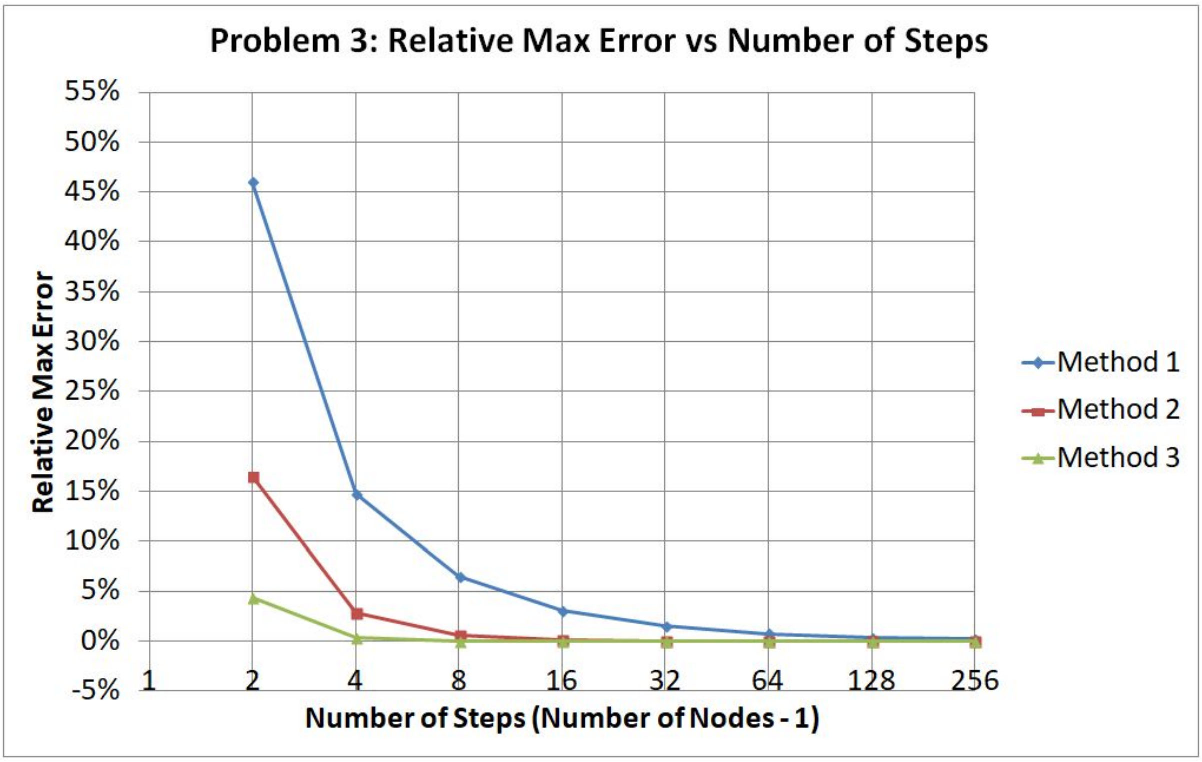
Report: Numerical Solving of Differential Equations

In this report I will present my results for methods 1, 2, and 3 for the problems 3, 5, 12.2, 13.1 and 14 from the third lab. For each problem I will look at the accuracy of the methods by using the error as well as visual proximity of the methods to the known solution. Furthermore, I will investigate if the methods converge for large numbers of nodes or steps.

For each of the problems I created a graph that shows the error of the calculation with respect to the number of nodes that are used for the calculation. For all of the problems, I used the same parameters where applicable. I took ε=0.5, λ=0.5 (for second method), σ=1 (third method) and φ=2 (problems 3 and 14). The resulting graphs are then used to examine the convergence of the methods as well as any abnormal behavior that one would not expect.

One would generally expect that the second method of Runge-Kutta will be more accurate than the first method. Similarly, the third method should be more accurate than the second method. The reason for this is that taking more steps for each calculation in each interval will result in better accuracy and lower error values. This claim will be tested for each problem.

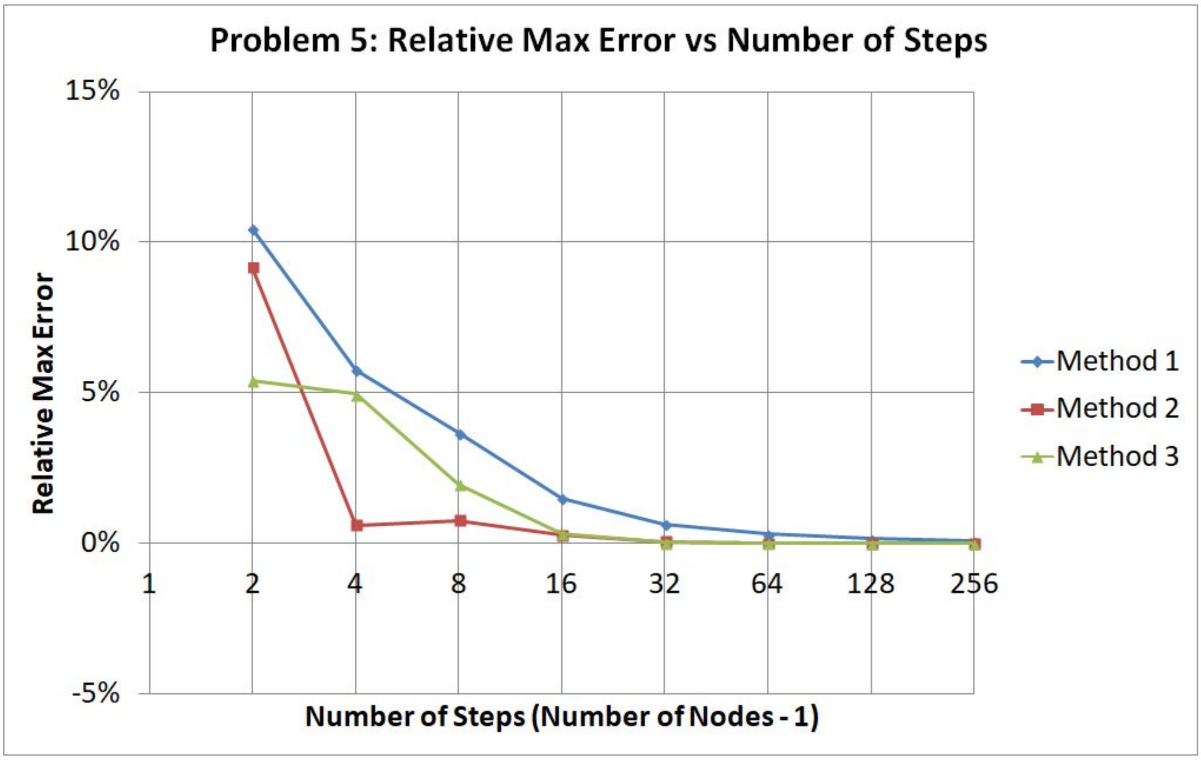
Problem 3



This graph shows that method 1 is the least accurate and that method 3 is the most accurate. The second method is closer to method 3 than to method 1 when it comes to accuracy. This is exactly like predicted in the introduction, the more steps you take for each interval, the higher the accuracy is.

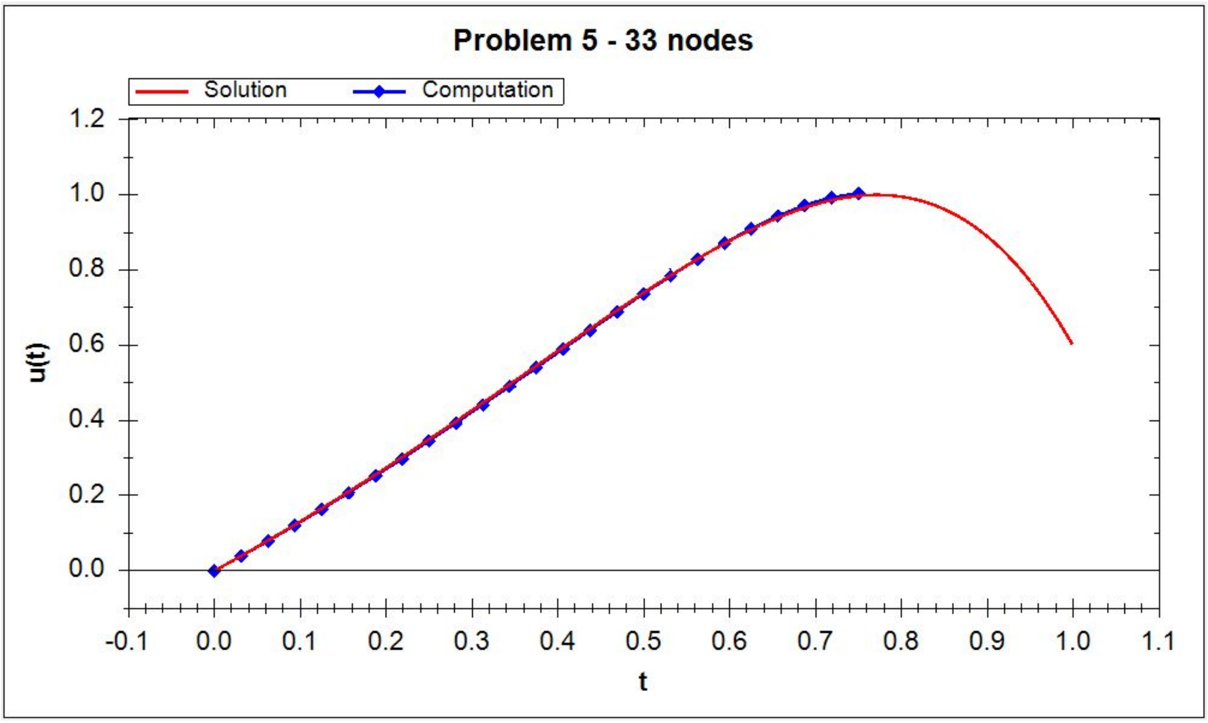
The graph also shows that the three methods seem to converge for large numbers of intervals and that their errors approach zero once there are at least 128 nodes. This graph does not show that the third method's error is only 1/10000th as large as the error of the first method for 256 intervals, so there is still a big difference in accuracy, only the magnitude is very small.

Problem 5

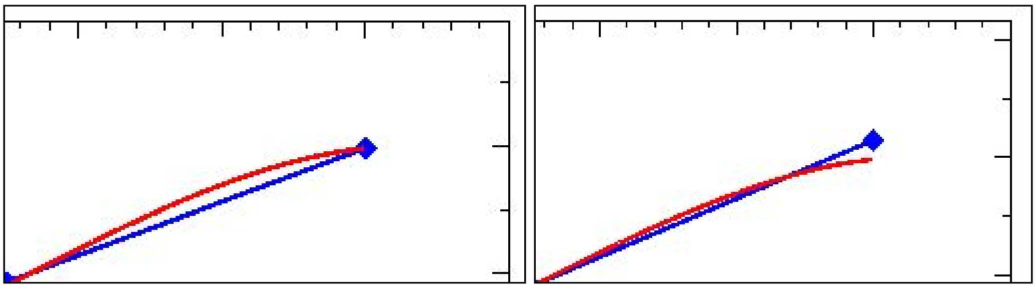


With problem 5 I had the issue that the given formula contains the term $\sqrt{1-[u(t)]^2}$. When ε<0.67, the curve has a peak somewhere around t=0.7 (depending on ε). The maximum value of $u(t)$ at that point is larger than 1, which means that the aforementioned square-root does not return a value, as the value under the root is negative. This then makes the formula unusable. If ε>=0.67, this issue no longer happens. This is why my test has ε=0.67 instead on ε=0.5. At the top of the next page is an image showing this problem for 33 nodes, first method and ε=0.4.

The diagram above shows that the errors of all three methods converge for large numbers of nodes. Here it is interesting to see that the second method is more accurate than the third one for 4 and 8 intervals, in all other cases the third method is at least as accurate as the second. This can be explained by the curvature of the function close to t=1. As can be seen in the image on the next page, $u(t)$ curves downwards as t approaches 1. For method 2, the last node's y value corresponds to the last y-value of the solution, while for method 3, it is larger. This causes the higher accuracy of method 2 for 4 and 8 intervals. This is illustrated by the two small pictures on the next page.

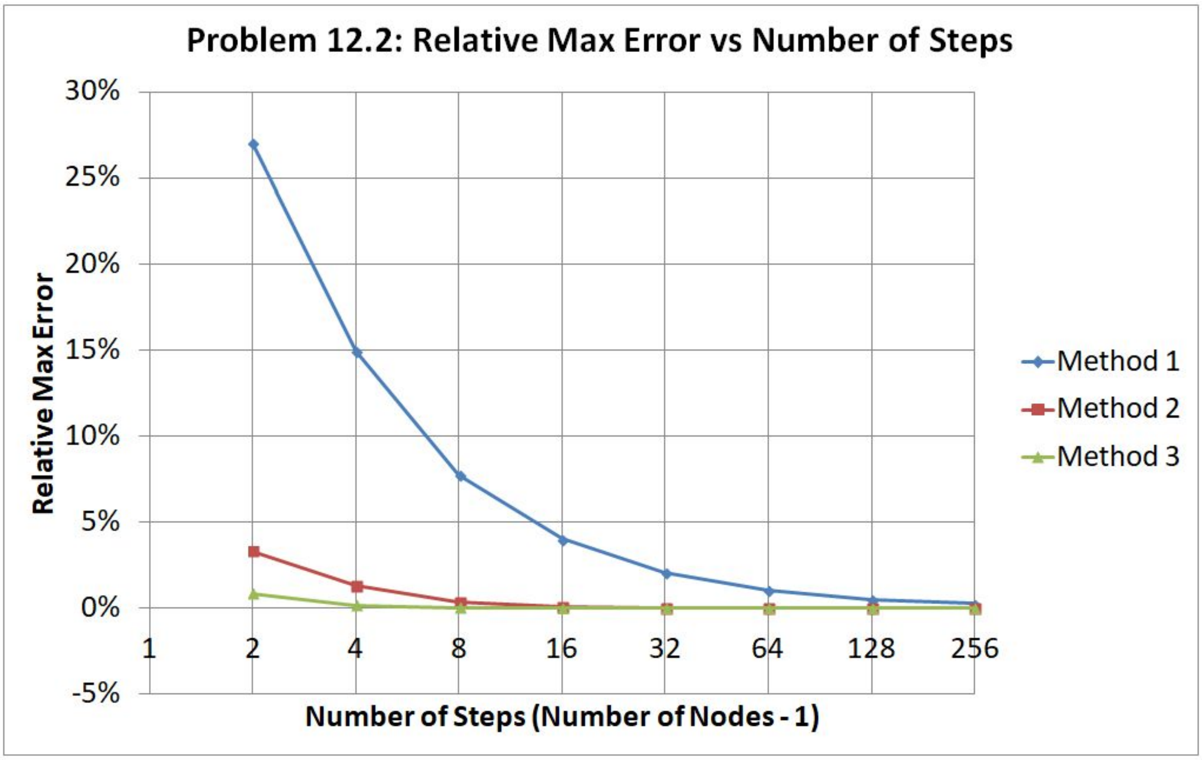


The two pictures below show method 2 (left) and method 3 (right) for 4 intervals. The



different end-points explain the lower error of method 2 vs method 3.

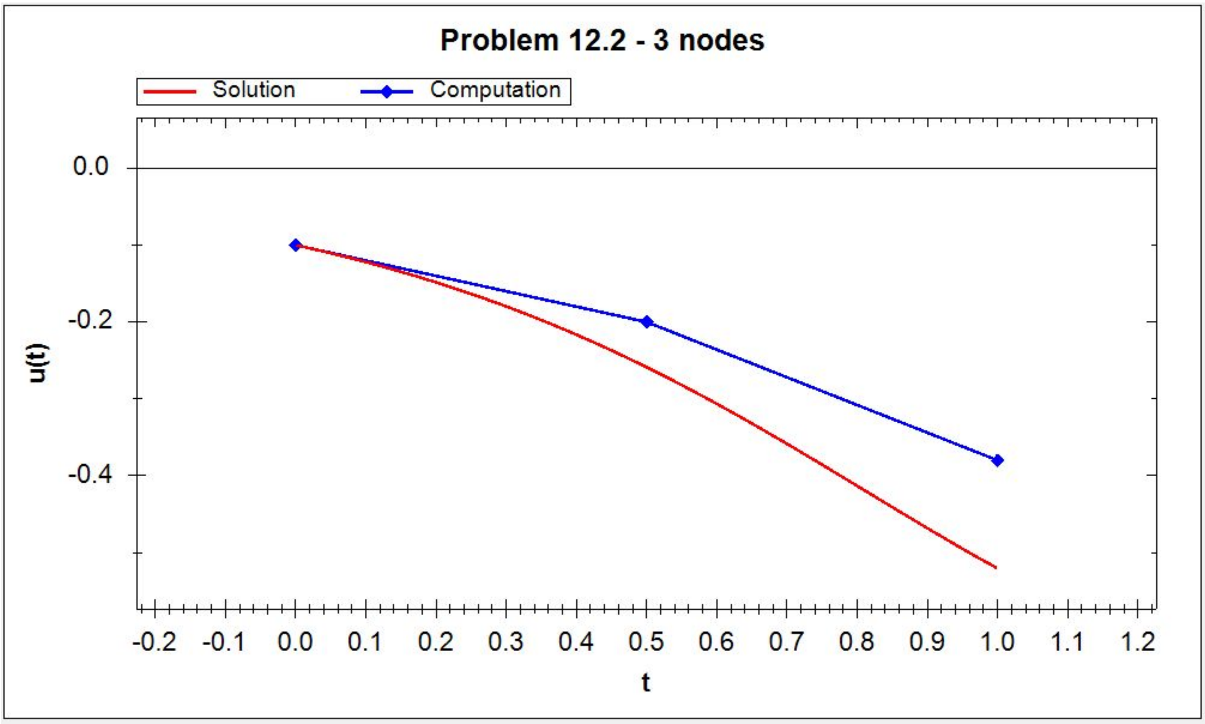
Problem 12.2

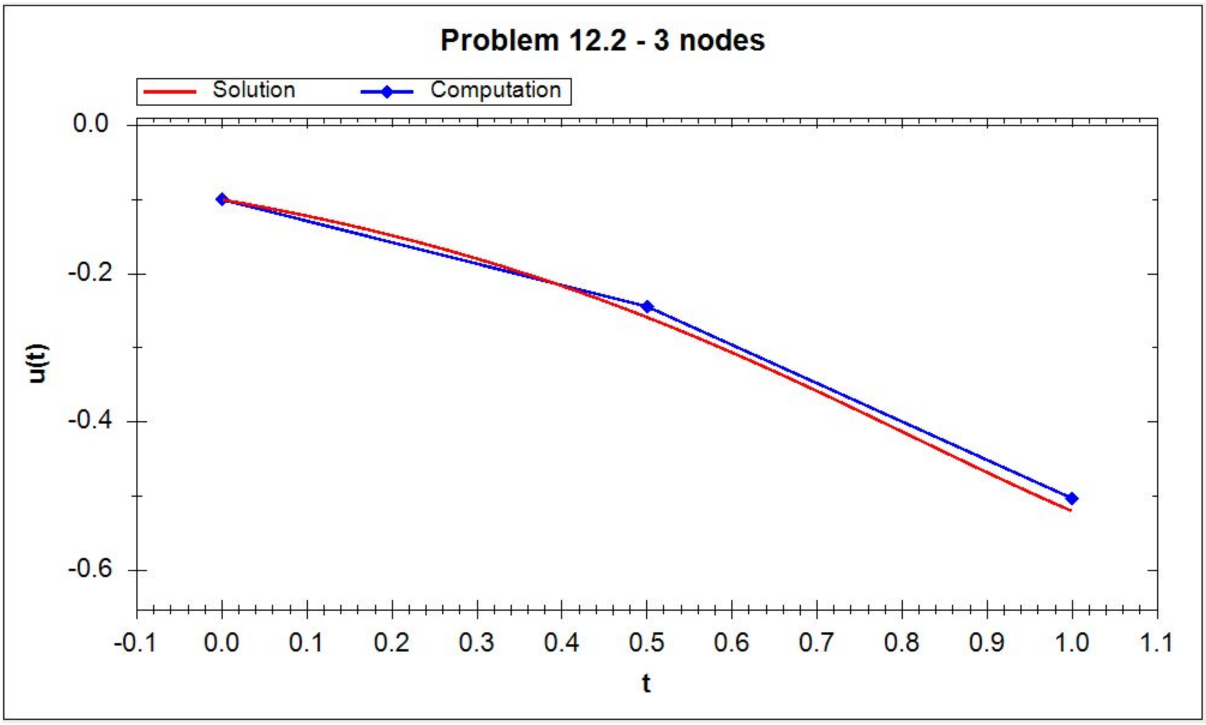


This graph again shows that the three methods seem to converge for high numbers of intervals. Notable here is that the second and third methods are very accurate and close together, with method one being the most inaccurate, which only starts to get close to 0 for 128 intervals.

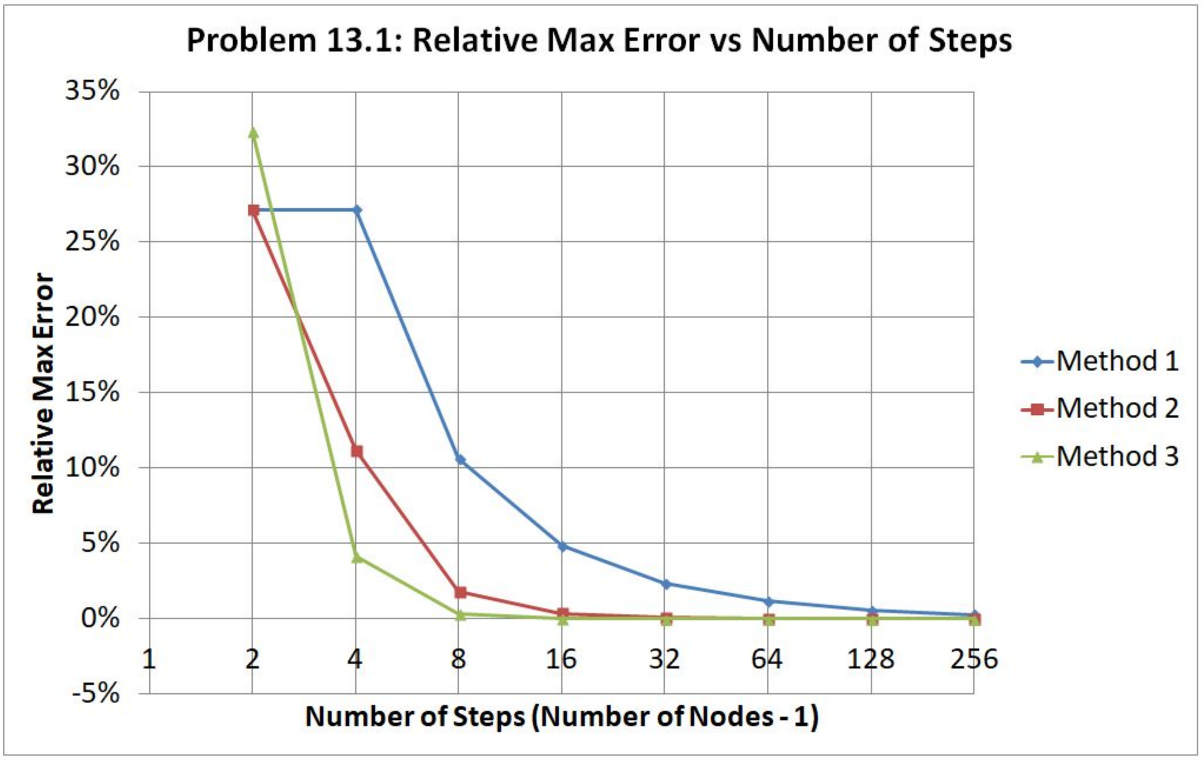
The big difference in the errors can be explained by the fact that problem twelve, with the given parameters, has a negative slope that get larger (in the negative direction) as t increases. Because method 1 only takes one point per interval and then bases the next interval's values on that, it can not keep up with a continuously decreasing slope, so the error is really large. Method 2 and 3 use more steps/points per interval, allowing them to follow the slope more closely.

Below you can see the results for method 1 and 2 for 2 intervals to illustrate this fact. The picture on the top show method 1 and the one below it shows method 2.


Problem 12.2 - 3 nodes
Solution
Computation
u(t)
t
0.0
-0.2
-0.4
-0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2

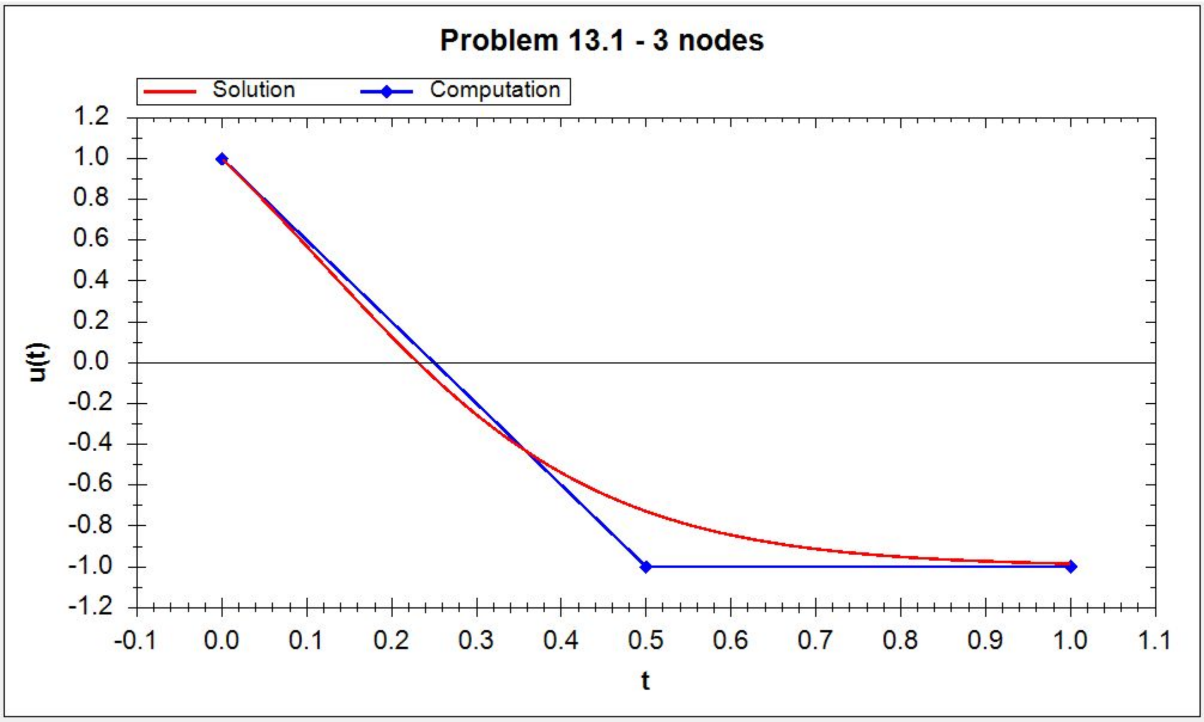

Problem 12.2 - 3 nodes
Solution
Computation
u(t)
t
0.0
-0.2
-0.4
-0.6
-0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1

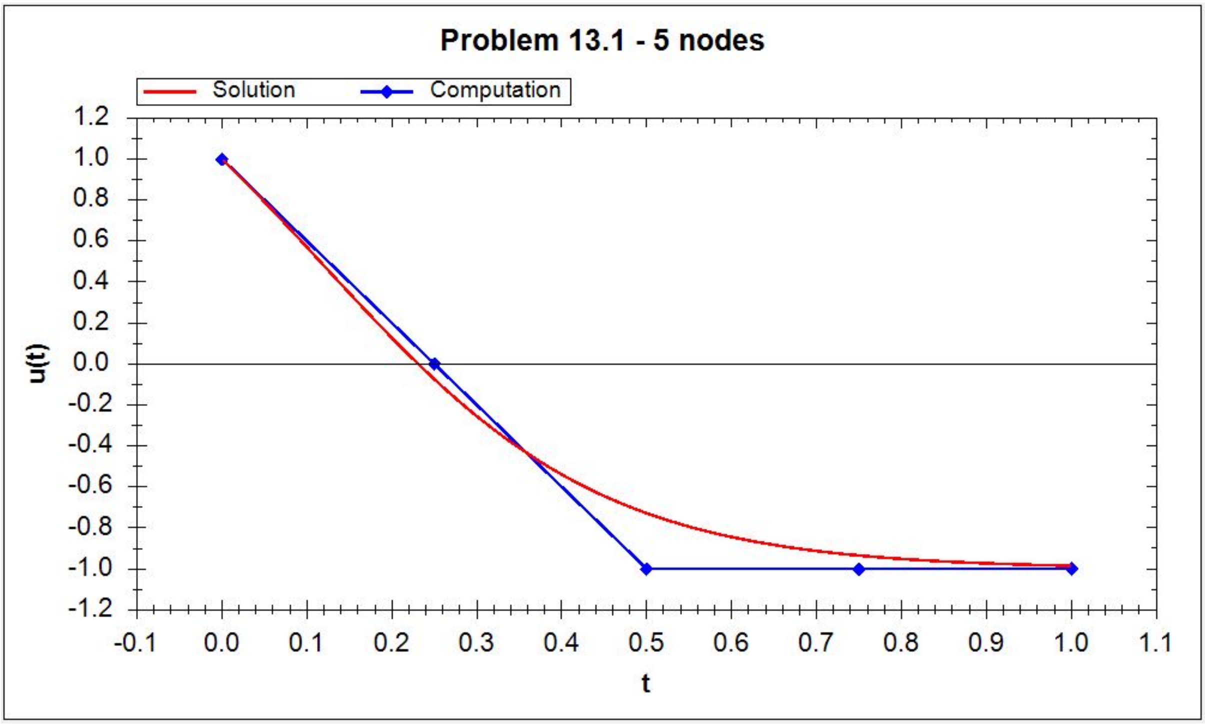
Problem 13.1



This diagram again shows that all three methods converge for high numbers of nodes, but there are several interesting things to see here. First, for two intervals, the third method is actually the least accurate out of all of them. This is caused by the third method following the slope of the curve more closely, but by doing that it ends up too far away.

Secondly, methods 1 and 2 start of on the same values same value, and then the first two values for method 1 are the same as well. I guess that this is caused by the similarities between the two methods, as they only differ in the additional y-term that method 2 has. For λ=0.5, the y-term is in the middle of the interval. This then means that, if the function has a slope that is relatively constant, the additional y-term will not make a difference, as it does not change the value. This is supported by the fact that method 1 for 4 intervals has the same value as method 2 for 2 intervals because method 1 for 4 intervals is, under the previous assumptions, the same as method 2 with 2 intervals. But because the methods are not identical, the second method is quickly seen to be the more accurate one. The following pictures show method 1 for 2 and for 4 intervals to illustrate this behavior. The upper picture shows method 1 for 3 nodes (2 intervals) and the lower picture shows method 1 for 5 nodes (4 intervals).


Problem 13.1 - 3 nodes
Solution
Computation
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
u(t)
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
t


Problem 13.1 - 5 nodes
Solution
Computation
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
u(t)
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
t