

Math 2280-2
Maple Project 1, Part 2
Numerical Methods, sections 2.2-2.4
Solution Template

(Two problems were added on Tuesday January 30)

(1) Enter your Name and Student number in this space: (You should know by now that you can mix computations with word processing in a Maple document.)

(2) Consider the same initial value problem as in the class handout, namely $dy/dx = y$, $y(0)=1$, which has solution $y=\exp(x)$. Use the improved Euler algorithm, with $n=100$, on the x -interval $[0,1]$, to approximate the solution. Print out your approximations at x multiples of 0.1 . Compare the accuracy of the $n=100$ improved Euler approximation to e , versus the results in the handout for unimproved Euler.

```
[ > restart:with(linalg):
```

(3) Do Runge-Kutta for the same initial value problem and x -interval, with $n=20$. Compare the accuracy of your results to the work you did with improved Euler $n=100$, in problem (2) above. There should be a huge improvement.

(4) Do problem #25 in Section 2.4, page 114. This problem illustrates how a numerical algorithm may accidentally pass right through a place where the real solution blows up. You may use the method of linear differential equations to solve the initial value problem in 25(a) by hand. [If you try to use `dsolve` I recollect that Maple runs into a little trouble.] Use Maple to help answer parts (b) and (c), however. Insert commands and text comments as necessary.

(5) Do problem #5 in sections 2.4, 2.5, 2.6 homework. This is the same differential equation in each case, studied successively with Euler, improved Euler, and Runge-Kutta.

(6) Do problem #25 in section 2.5

(7) Go to example 4 on page 134-135. Work through and understand what the text has to say on these two pages. Recreate the data in the last two columns of Figure 2.6.9 on page 134 using the Runge-Kutta routine from the class handout. That is, recompute the the Runge-Kutta approximations with $h=0.05$ and compare to the actual solution values, at multiples of 0.4 .

Understand (for yourself) how mathematical instability caused by the solution to the homogeneous problem (and exhibited in the slope field in Figure 2.6.10) can lead to numerical instability like that illustrated in this important example of problem, even when you're using a "good" numerical technique like Runge Kutta and a "small" time step. This example illustrates why people who create numerical solutions to problems must also be well-grounded in the mathematical theory behind the problems.