

## Assignment #8

**Due Monday, 18 November 2024, at the start of class**

With the exception of **P10**, this Assignment is based on Chapter 11 of our textbook,<sup>1</sup> which is a good introduction to solving ordinary differential equations on computers. Please read all of sections 11.1 and 11.2; the latter is very substantial! You can skip section 11.3, but please read section 11.4 through page 289; you can skip the rest (pages 290–294). In addition to doing the exercises, I recommend playing with additional `ode45()` and Euler’s method examples—for the latter writing the codes yourself.

These expectations always apply to homework:

1. Please put the problems in the order they appear below.
2. When you use MATLAB/etc, show the commands and the results.
3. Keep a clear distinction between codes, input commands, computed results, and figures.
4. Other than the text you write, please minimize use of paper.

**Do these exercises from the textbook:**

### CHAPTER 11

- Exercise 1 (a), (b) on page 295.
- Exercise 3 on page 295. *Hint. There are 2 solutions to this “bad” ODEIVP.*
- Exercise 5 on page 295. *Skip Heun’s method.*
- Exercise 6 on page 295. *Skip the “local truncation error” part.*

**Do these additional problems:**

**P10.** (a) Apply `mytrap.m` (see the **Codes tab**) to accurately approximate

$$\int_0^5 \cos(x^2) dx$$

by the trapezoid rule. (I do not know how to find the exact value.) Use increasing large values for  $n$ , and report what you did. Your goal should be at least 6 correct digits. How many correct digits do you think you have?

(b) Implement Romberg integration as described in lecture,<sup>2</sup> in the form of

```
[z, count] = myromberg(f, a, b, M)
```

<sup>1</sup>Greenbaum & Chartier, *Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms*, Princeton University Press 2012).

<sup>2</sup>On Wednesday 6 November, in particular.

Here `z` is the integral and `count` is the total number of function evaluations. The input `M` is the number of times you double the number of subintervals in the trapezoid rule evaluation, so I would not expect you to use  $M > 10$  for example. You might start with two subintervals when you first evaluate the trapezoid rule. In any case, you may call `mytrap.m`. You may use<sup>3</sup> `polyfit` to do the extrapolation, in  $h^2$ , to  $h = 0$ . Test your implementation on an integral or two where you *do* know the exact answer.

(c) Now use Romberg integration on the integral in part (a). You should be able to get the same number of correct digits with far fewer function evaluations.

**P11. (a)** Verify that  $y(t) = \ln(t^2/2 + t + 1)$  exactly solves

$$y' = (t + 1)e^{-y}, \quad y(0) = 0.$$

(b) Using  $h = 1.0$ , apply 10 steps of Euler's method, formula (11.7), to the ODE IVP in part (a). Put the  $t_k$  and  $y_k$  values in a table, along with the exact  $y$  values at the  $t_k$  points.

(c) Add another column to the table by applying the midpoint rule, formulas (11.14) and (11.15), to the same problem, again using  $h = 1.0$ . Comment on accuracy.

**P12.** Consider the ODE IVP

$$y' = 1 + y^2, \quad y(0) = 0.$$

Note that  $f(t, y) = 1 + y^2$  when we write this in the form (11.1).

(a) Apply the trapezoid method (11.18) to the above ODE. That is, write down the nonlinear equation for  $y_{k+1}$  at each time step, based on knowing  $y_k$  from the previous step, for this  $f$ .

(b) For  $y_0 = 0$  and  $h = 0.1$ , part (a) gives an equation which must be solved to find  $y_1$ . Write this equation. Solve it exactly, but observe that you get two solutions. Which is the one which is the correct step of the trapezoid rule?

(c) In part (b) you have an equation to solve for  $y_1$ . Use Euler's method to give a first guess  $y_1^{(0)}$  for the solution to that equation. Then do one step of Newton's method to get  $y_1^{(1)}$ ; this will be very close to one of the two values from part (b).

*It is common to accept a fixed and small number of steps of Newton's method when doing trapezoid rule steps. That is, to not solve the implicit step equation accurately.*

---

<sup>3</sup>Or you may implement following the book's version on page 243. However, I think you will have more fun with setting up polynomial extrapolation to zero!