# Assignment #9

## Due Thursday, 16 November at the start of class

Please read sections 5.6–5.8 of the textbook (J. Epperson, *An Intro. to Numerical Methods and Analysis*, 2nd edition).

**Section 5.6, pages 297–299:** I recommend doing exercise **1 (a),(b)** at the command line, so that you become familiar with the numbers. Then do **P11** below, so that you have a tested code. Then do exercise **2 (a),(e)** using your code.

- **Exercise 1.** Do parts **(a),(b)** only.
- **Exercise 2.** Do parts **(a),(e)** only.
- **Exercise 11.**
- **Exercise 13.**
- **Exercise 15.**

**P11.** Write a code `gauss4.m` with first line

```
function I = gauss4(f,a,b)
```

which applies $n = 4$ point Gauss quadrature to approximate $\int_a^b f(x)\,dx$. There are two things you will need from the book, first being the change of interval formula from the bottom of page 294 (and exercise 15 above), i.e.

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\left(a + \frac{b-a}{2}(z+1)\right)\,dz.$$

The second thing is that you will need the $n = 4$ nodes and weights from Table 5.5 on page 289. Test your code on integrals of the form

$$\int_a^b x^k\,dx$$

for several different combinations of $a, b \in \mathbb{R}$ and integers $k = 0, \dots, 7$. The code should get all of these exact. Then test on $\int_a^b x^8\,dx$; it should *not* be exact. (*Thus confirm that your implementation works* and *has the promised degree of precision* $p = 2n - 1 = 7$.)

**P12.** *This problem asks you to implement the "Romberg integration" described in class. Recall that Romberg's idea was to extrapolate the results of the composite trapezoid rule to otherwise-unattainable* $h = 0$ *spacing. Recall that* `trap.m` *is posted online:*

http://bueler.github.io/M310F17/matlab/trap.m

*(You can also use* `trapol.m`*. ) For the integral* $\int_a^b f(x)\,dx$*, it computes the composite trapezoid rule with* $n$ *subintervals:* $T_n(f) =$ `trap(f,a,b,n)`*.*

Write a new code `romberg.m` which does $K$-level Romberg integration:

```
z = romberg(f,a,b,K)
```

It does $K$ composite trapezoid rule applications by calling `trap.m`:

$$T_2(f), T_4(f), \dots, T_{2^K}(f).$$

It also calculates the corresponding spacings,

$$h_1, h_2, \dots, h_K.$$

Then it uses `polyfit` to compute the polynomial $p$ which goes through the "data"

$$(h_1^2, T_2(f)), \quad (h_2^2, T_4(f)), \quad \dots, \quad (h_K^2, T_{2^K}(f)).$$

Then it evaluates this polynomial at zero to give the result:

$$\mathtt{z} = p(0).$$

Compare accuracy of `gauss4(f,0,2)` to `trap(f,0,2,`$2^K$`)` and `romberg(f,0,2,`$K$`)` for $K = 2, 3, 4, 5$, on the integral

$$\int_0^2 xe^{-x}\, dx.$$

(*Start by computing the exact value of this integral.*) Compare both the accuracy and the number of function evaluations in a table.

(*The table should have 9 rows, one for each calculation, and columns for the absolute errors and the number of $f$ evaluations.*)

**P13.** *EXTRA CREDIT.*

Explain how to optimize `romberg.m` to

- eliminate all redundant function evaluations, so that it does exactly $2^K + 1$ function evaluations *in total*, just like $T_{2^K}(f)$ itself, and
- minimize the amount of arithmetic in the extrapolation stage.

For the second optimization you may want to use divided differences. (*Do not call* `polyfit` *or any other polynomial interpolation code.*)

Then implement your method and check it produces the same results as `romberg.m`.