CS/Math 514, Fall 2019 – Numerical Analysis (Roos)
Homework assignment 7.
Due Monday, November 4.

Problems with an asterisk (*) are optional.

1. Determine an explicit formula for the Gaussian quadrature rule $\mathcal{G}_n(f)$ for approximating the integral $\int_{-1}^{1} f(x)\sqrt{1-x^2}dx$. That is, find explicit formulas for the Gaussian quadrature weights $(w_k)_{k=0,\dots,n}$ and points $(\xi_k)_{k=0,\dots,n}$ such that $\mathcal{G}_n(f) = \sum_{k=0}^{n} w_k f(\xi_k)$ integrates polynomials f of degree $2n+1$ exactly.
Hint: The relevant orthogonal polynomials can be expressed in terms of Chebyshev polynomials.

2. In class we saw how the Euler-Maclaurin expansion is useful for numerical approximation of definite integrals. This reasoning can be reversed: Euler-Maclaurin expansion is also useful for computing the value of some slowly convergent infinite series. Consider for $x > 1$ the series

$$\zeta(x) = \sum_{n=1}^{\infty} \frac{1}{n^x}.$$

(a) Write a program `zeta1(x, N)` that uses naive partial summation $\sum_{n=1}^{N} \frac{1}{n^x}$ to compute $\zeta(x)$.
(b) Use Theorem 7.3 (Euler-Maclaurin expansion) in the textbook for a fixed order k to derive an exact formula comparing $\sum_{n=N+1}^{M} f(n)$ with $\int_{N+1}^{M} f(t)dt$ for $M > N+1$.
(c*) Write a program `zeta2(x, N, k)` that uses the partial sum $\sum_{n=1}^{N} \frac{1}{n^x}$ and Euler-Maclaurin expansion up to order k (using the formula from (b) with $f(t) = t^{-x}$) to estimate the remaining sum $\sum_{n=N+1}^{\infty} \frac{1}{n^x}$. *Comments:* In this setting we may let $M \to \infty$ in the formula from (b) (why?). Note that the error term in that formula tends to 0 quickly as N becomes larger (why? how quickly?), so it may be ignored in the program.
(d*) Assuming arbitrary precision, how large would you need to make N in the program `zeta1` to compute $\zeta(x)$ up to d correct digits after the decimal point? (Give an explicit estimate for that N up to a factor of 10, depending on x.)

3*. Experimenting appropriately with values of N, k, compare the programs `zeta1` and `zeta2` from Problem 2 by attempting to compute $\zeta(x)$ for different values of $x > 1$, say $x = 2, 5/4, 11/10, 101/100$, each up to 10 correct decimal digits after the decimal point. Be wary of the limits of machine precision! If necessary you may use high precision floating point libraries (try to avoid it for the program `zeta2` as an additional challenge). Here are the correct results for comparison:

$$\zeta(2) = 1.644934066..$$
$$\zeta(5/4) = 4.595111825..$$
$$\zeta(11/10) = 10.5844484649..$$
$$\zeta(101/100) = 100.5779433384..$$