**CS/Math 514, Fall 2019 – Numerical Analysis (Roos)**
**Homework assignment 5.**
Due Monday, October 14.

**1.** In this exercise we want to compute $I = \int_a^b f(t)dt$ by forming weighted sums $I_N$ of the values of $f$ at points $t_j = a + jh$ for $j = 0, 1, \ldots, N$ and $h = \frac{b-a}{N}$.
(i) Implement the composite trapezoid and Simpson rules as functions `trapezoid(f, a, b, N)`, `simpson(f, a, b, N)`.
(ii) For each of the following, plot the negative logarithmic error $-\log_{10}(|I - I_N|)$ of composite trapezoid and composite Simpson rules against $N$ with $N = 2^k$, $k = 2, \ldots, 20$.

$$(a) \int_0^1 t^3 dt \quad (b) \int_{-2}^2 \frac{dt}{1+t^2} \quad (c) \int_0^1 \cos(10^5 t)dt$$

(The true values are (a) $\frac{1}{4}$, (b) $2\arctan(2)$, (c) $10^{-5}\sin(10^5)$.)
(iii) Compare and explain the results (how do the Simpson and trapezoid error plots look different and why? How do the results for (a), (b), (c) differ from each other and why?).

**2.** Do Problems 7.1, 7.2, 7.3 in the textbook.

**Honors Problem 2.** The Newton-Cotes formula of order $N$ for computing $I = \int_0^1 f$ is given by

$$I_N = \sum_{j=0}^{N} f(t_j) w_j$$

with $t_j = j/N$ for $j = 0, \ldots, N$ and $w_j = \int_0^1 \prod_{j \neq i} \frac{t-t_i}{t_j - t_i} dt$.
(i) Write from scratch (using no advanced library functions other than standard `numpy` array operations and Python's builtin arbitrary precision integers) a program that can compute the weights $(w_j)_{j=0,\ldots,N}$ for given $N$ and test it by computing the weights exactly (they are rational numbers) for $N = 15$.
(ii) Compute the output $I_N$ of the Newton-Cotes formula of degree $N$ applied to $I = \int_{-2}^2 \frac{dt}{1+t^2}$ for $N \in \{1, 2, \ldots, 15\}$. Compare the output to the true value of $I$ and explain what happened.