# Numerical Methods I – Lab 1

Fall Semester 2005

September 13, 2005

1. Use Octave to compute

   (a) $\sin(2\pi \cdot 2^k)$

   (b) $1 - \cos(2\pi \cdot 2^k)$

   for $k = 0, \dots, 60$. Plot the log of the error vs. $i$. Explain.

2. The limit

$$C = \lim_{n\to\infty} c_n = 0.577\,215\,664\,901\,532\ldots \qquad \text{with} \qquad c_n = \sum_{k=1}^{n} \frac{1}{k} - \ln n$$

   is called Euler constant.

   Can you use this formulation to compute $C$ to absolute accuracy $10^{-12}$ in less than one minute?

   (a) Write an Octave function for computing $c_n$ which accepts as optional input $c_m$ for $1 \le m < n$. Test its performance (runtime and absolute error with respect to the given value for $C$) for $n \le 10^7$.

   (b) Use a back-of-the-envelope calculation to check if your program can be used to solve the problem.

   (c) Check whether your implementation faces any error propagation issues.

   (d) Verify numerically (again for $n \le 10^7$) that the approximation error satisfies

$$E_n \equiv c_n - C = \frac{1}{2n} + O\Big(\frac{1}{n^2}\Big)$$

   as $n \to \infty$. Assuming that this relation holds, compute a new sequence

$$d_n = 2\,c_{2n} - c_n$$

   and check the new error sequence $F_n \equiv d_n - C$ numerically. Explain the result!

   (e) Find an explicit formula for $n_n$ and show that it can be computed with the same number of terms as $c_n$ (and not $c_{2n}$!). Modify the function from (a) to directly compute $d_n$.