ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Programming Techniques for Scientific Simulations Exercise 1

HS 12
Prof. M. Troyer

Problem 1.1 Endianness (Block A)

Write a program to determine the endianness of your machine.

Problem 1.2 Machine epsilon (Block A)

Write a program to determine the numeric machine precision ϵ for several datatypes, such as `float`, `double` and `long double`.

Problem 1.3 Simpson numerical integration (Block B)

Write a program to implement the following numerical integration using the Simpsons rule[1]

$$\int_0^{\pi} dx \sin(x) \ . \tag{3}$$

Hint for testing/debugging: The Simpson integration should integrate polynomials up to the 2$^{\text{nd}}$ order precisely with any number of bins. So you may for instance integrate $\int_0^1 \mathrm{d}x\, x(1-x) = 1/6$ with $N = 1, 2, 3, 10$ bins for the testing purpose.

---

[1] Recall that for the 1-dimensional Simpson integration you approximate the function by a parabola in each bin stretching from x to $x + \Delta x$. For that you need 3 function values at x, $x + \Delta x/2$ and $x + \Delta x$. The integral over the interpolating parabola $\tilde{f}(x)$ gives

$$\int_x^{x+\Delta x} \mathrm{d}x\ \tilde{f}(x) = \frac{\Delta x}{6}\Big[f(x) + 4f(x + \Delta x/2) + f(x + \Delta x)\Big] \ . \tag{1}$$

In order to numerically integrate a function from a to b you discretize it to N bins and use the interpolation formula within each bin. If you use regular mesh (= equally sized bins) with bin size $\Delta x = (b-a)/N$ then the complete formula for Simpson integration looks

$$\begin{aligned}\int_a^b \mathrm{d}x\ f(x) \quad &= \quad \frac{\Delta x}{6}\Big[f(a) + 4f(a + \Delta x/2) + 2f(a + \Delta x) + 4f(a + 3\Delta x/2) + \ldots \\ &\quad + \ldots + 2f(b - \Delta x) + 4f(b - \Delta x/2) + f(b)\Big] + O\left(N^{-4}\right) \ . \end{aligned} \tag{2}$$