Prof. Dr. Jens Honore Walther
Dr. Georgios Arampatzis
ETH Zentrum, CLT
CH-8092 Zürich

Set 7

Issued: 28.04.2021

In this exercise, you will practice Richardson extrapolation for improving the accuracy of function evaluation and implement Romberg integration in an engineering problem.

Question 1: Finite differences with Richardson extrapolation

a) A finite difference approximation (i.e., a numerical approximation) of the first derivative of a function $f(x)$ at $x = 0$ is

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = G_0(h),$$

and the n-th application of Richardson extrapolation is given by the formula

$$G_n(h) = \frac{1}{2^n - 1}(2^n G_{n-1}(h/2) - G_{n-1}(h)).$$

Let $f(x) = x + e^x$. Set $h = 0.4$ and compute the Richardson extrapolation up to $G_2(h)$. Keep 5 decimal points throughout the calculations.

b) Since the exact value is known ($f'(0) = 2$), you can compute the error $E_n(h) = |G_n(h) - 2|$ for each term in a). Is the accuracy improved over the iterations?

Question 2: Pseudocode for Romberg integration

Write a pseudocode for Romberg integration. Write your own code from scratch or use the skeleton pseudocode below.

Question 3: Romberg Integration

The sine integral $\mathrm{Si}(x) = \int_0^x \frac{\sin(t)}{t}\,\mathrm{d}t$ can not be easily integrated.

Find an approximation of $\mathrm{Si}(\pi)$ with the use of Romberg integration. Start with an interval size of π and approximate the integral using the trapezoidal rule up to 2^{nd} order (I_2^1). (Hint: $\frac{\sin(0)}{0} = 1$)

Algorithm 1 Romberg integration

Input:

function $f(x)$
interval boundaries a, b
number of iterations K

Output:

$I_K^1 =$ `integral`$[K, 0]$ approximation to the integral $\int_a^b f(x)\,\mathrm{d}x$

Steps:

```
maxNumIntervals ← 2^K

// Precompute and store function evaluations
hmin ← (b − a)/maxNumIntervals
for i ← 0, ..., maxNumIntervals do
    .
end for

// Compute level 0 integrals
for r ← 0, ..., K do // refinement
    numIntervals ← 2^r
    step ← 2^(K−r) // step between two function evaluations for this refinement
    .
    .
    .
    .
    // composite trapezoidal rule:
    .
end for

//Advance to higher precision according to Romberg
for l ← 1, ..., K do //level
    .
    .
    .
end for
```