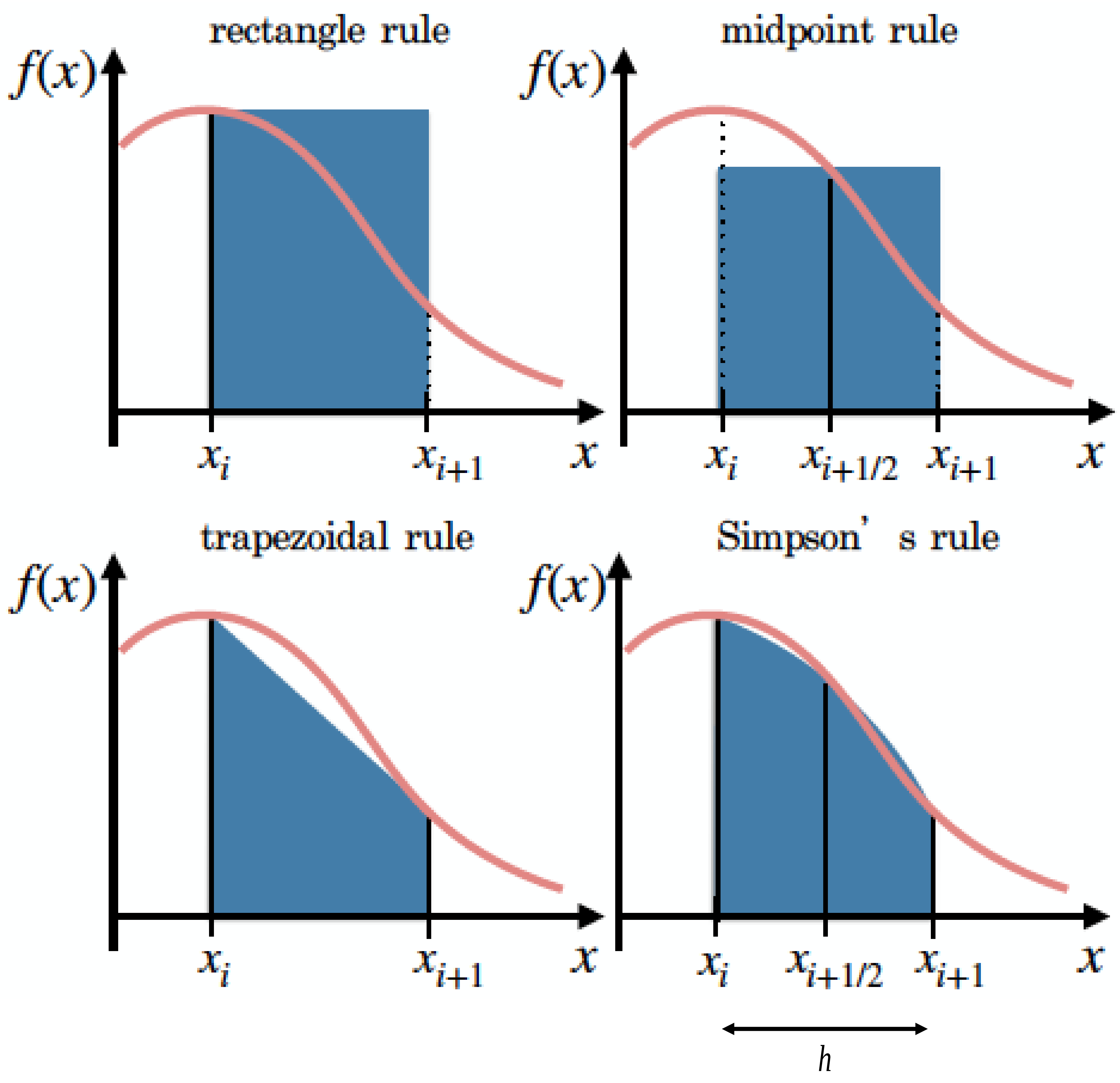
MAD Exercise Session – Tuesday, 21.04.2020

Pascal Auf der Maur

Computational **S**cience and **E**ngineering **Lab**
ETH Zürich

Lecture Recap

Numerical Quadrature



$$\textbf{Rectangle Rule:} \quad I \approx \Delta_x \sum_{i=0}^{N-1} f(x_i),$$

$$\textbf{Midpoint Rule:} \quad I \approx \Delta_x \sum_{i=0}^{N-1} f\left(\frac{x_i + x_{i+1}}{2}\right), \tag{6.9}$$

$$\textbf{Trapezoidal Rule:} \quad I \approx \frac{\Delta_x}{2}\left(f(x_0) + 2\sum_{i=1}^{N-1} f(x_i) + f(x_N)\right),$$

$$\textbf{Simpson's Rule:} \quad I \approx \frac{\Delta_x}{3}\left(f(x_0) + 4\sum_{\substack{i=1 \\ i=\text{odd}}}^{N-1} f(x_i) + 2\sum_{\substack{i=2 \\ i=\text{even}}}^{N-2} f(x_i) + f(x_N)\right). \tag{6.10}$$

$$h = \frac{a-b}{N} \qquad x_0 = a, x_{N-1} = b$$

Richardson Extrapolation

- Richardson's idea: combine and in a smart way

$$G(h) = G(0) + c_1 h + c_2 h^2 + \cdots$$

$$G(h/2) = G + \frac{1}{2} c_1 h + \frac{1}{4} c_2 h^2 + \cdots$$

$$G_1(h) = 2G(h/2) - G(h)$$

$$2\left(G(0) + c_1 h + c_2 h^2 + \cdots\right) - \left(G + \frac{1}{2} c_1 h + \frac{1}{4} c_2 h^2 + \cdots\right)$$

$$G + \frac{1}{2} c_2 h^2 + \ldots$$

Leading order term is now second order!

- Can be repeated:

$$G_2(h) = \frac{1}{3}\left(4G_1(h/2) - G_1(h)\right) = G + O\left(h^3\right)$$

$$G_n(h) = \frac{1}{2^n - 1}\left(2^n G_{n-1}(h/2) - G_{n-1}(h)\right) = G + O\left(h^{n+1}\right)$$

Error: $\epsilon(h/2) \approx G(h/2) - G(h)$

If is small () good!

If is too large keep subdividing

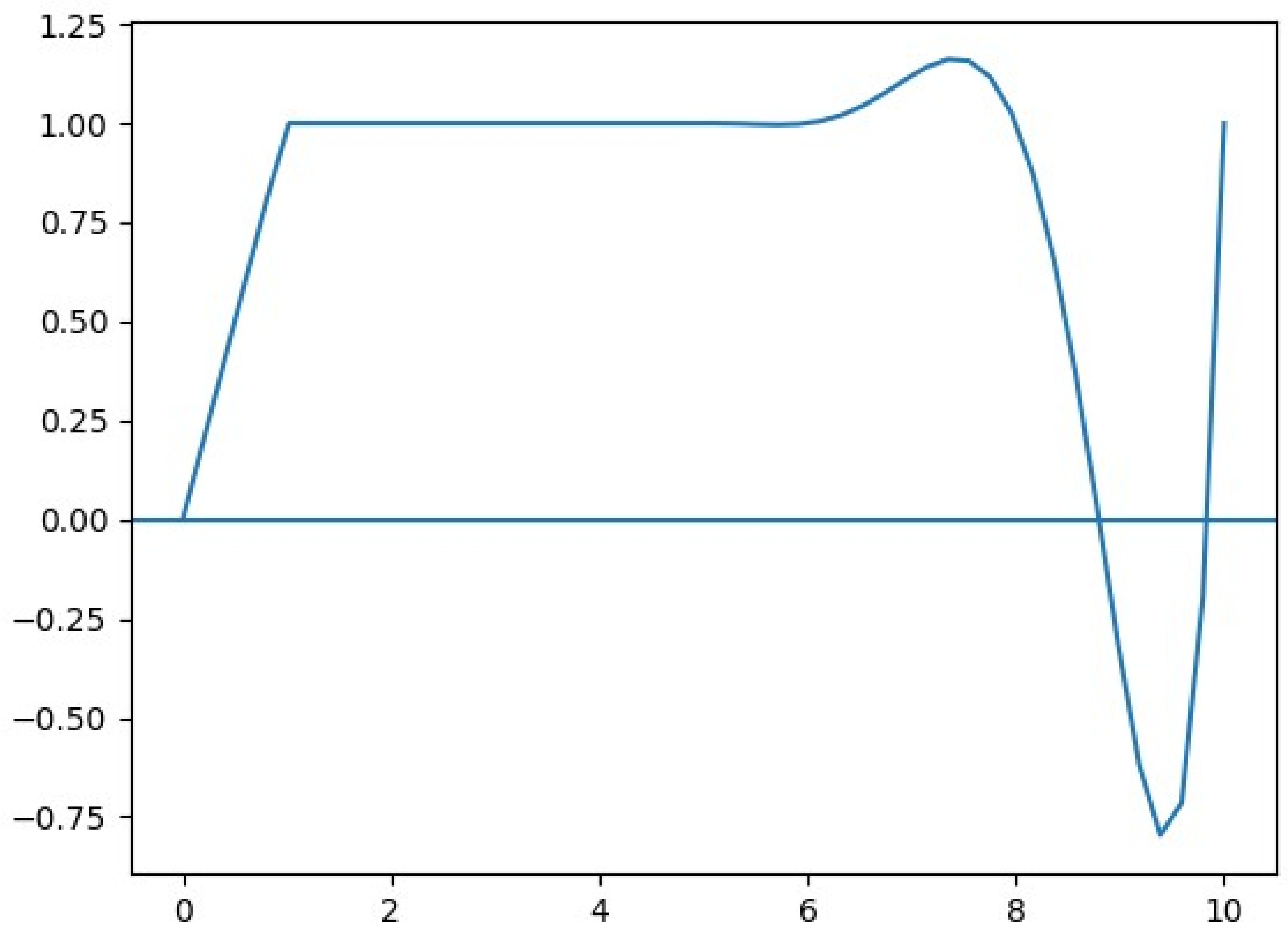
Good way to estimate the error of a discretization

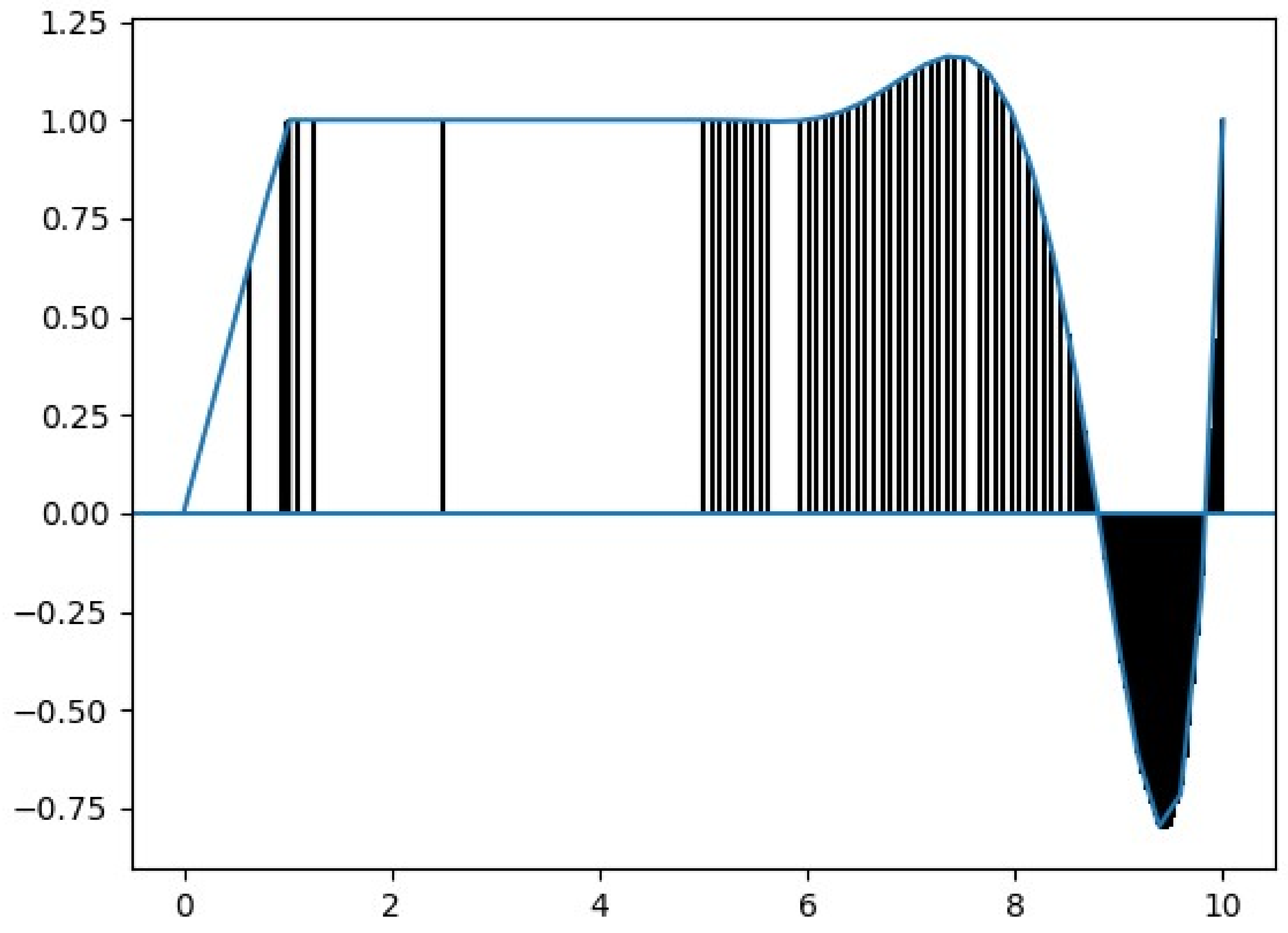
Problem

- Up until now the main way to increase accuracy is to have more function evaluations
- Today we will have a look at how to reduce the number of function evaluations with
  - Adaptive quadrature to enhance the algorithm
  - Gauss quadrature to improve the quadratures

Adaptive Quadrature

- On some functions it would be beneficial if we sample the function non uniformly
- Linear or constant intervals can be exactly approximated with a single interval of the Trapezoidal rule





Adaptive Quadrature

- Pseudocode for a recursive Implementation

Algorithm 1 Adaptive integration using recursion

```
function ADAPTIVESIMPSON(a, b)
    apply Simpson's rule in interval [a, b]
    subdivide the interval into [a, m] and [m, b] with m = (a + b)/2
    apply Simpson's rule in intervals [a, m] and [m, b]
    estimate error in [a, b] using Richardson's extrapolation
    if accuracy is worse than desired then
        return ADAPTIVESIMPSON(a, m) + ADAPTIVESIMPSON(m, b)
    else
        return value of Simpson's rule (the accurate one)
    end if
end function
```

Gauss Quadrature

- Gauss Quadrature aims to improve where we are evaluating the function in the intervals:

$$I = \int a_0 + a_1 x + a_2 x^2 + a_3 x^3 \, dx \approx c_1 f(x_1) + c_2 f(x_2)$$

- By inspecting the coefficients we find 4 equations for the 4 unknown c_1, c_2, x_1 and x_2
- The found quadrature is known as the 2-point Gauss quadrature and can approximate cubic functions exactly

$$\int f(x)\,dx \approx \frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(\frac{-1}{\sqrt{3}}\right) + \frac{a+b}{2}\right] + \frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(\frac{1}{\sqrt{3}}\right) + \frac{a+b}{2}\right]$$

Gauss Quadrature

- Gauss Quadrature aims to improve where we are evaluating the function in the intervals:

$$I = \int a_0 + a_1 x + a_2 x^2 + a_3 x^3 \, dx \approx c_1 f(x_1) + c_2 f(x_2)$$

- By inspecting the coefficients we find 4 equations for the 4 unknown c_1, c_2, x_1 and x_2
- The found quadrature is known as the 2-point Gauss quadrature and can approximate cubic functions exactly

$$\int f(x)\,dx \approx \frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(\frac{-1}{\sqrt{3}}\right) + \frac{a+b}{2}\right] + \frac{b-a}{2} f\left[\left(\frac{b-a}{2}\right)\left(\frac{1}{\sqrt{3}}\right) + \frac{a+b}{2}\right]$$

Gauss Quadrature

- Here a demonstration of this new Quadrature

$$I = \int x^6 + 0.5x^3 + 2x^2 + x + 4\, dx$$

	Approximation	Error
Trapezoidal Rule	1157.25	800.196
Simpson's Rule	431.906	74.852
2 point Gauss Quadrature	307.875	−49.179
3 point Gauss Quadrature	356.272	−0.782
True Value	357.054	*

Exercise Sheet 8

Question 1

- Similar to the shown demonstration of the Gauss Quadrature
- Approximate the following integrals with
  - Trapezoidal rule (two intervals)
  - Newton Cotes (Simpson's Rule)
  - Gauss Quadrature

$$I = \int x^6 - x^2 \sin(2x)\, dx$$

$$I = \int 1 - |x-1|\, dx = 1$$

- Use a calculator for the evaluations
- Observe the behaviour when the function is not smooth

Question 2

- Perform an Adaptive Quadrature on the provided functions
- Use the given criterion if the current approximation is good enough

Question 2: Adaptive quadrature

Apply adaptive quadrature by hand, using the Trapezoid Rule with relative tolerance $tol = 0.05$ to approximate the integrals. Relative tolerance is related to the Richardson extrapolation error as :

$$\epsilon(h/2) < 3 \cdot tol \cdot \frac{h}{h_0}$$

where h_0 is the size if the initial interval.

Find the approximation of the integrals and error compared to the exact solution for both functions below.

a) $f(x) = x^2$, $a_0 = 0$, $b_0 = 1$

b) $f(x) = \cos(x)$, $a_0 = 0$, $b_0 = \pi/2$

- Tips:
- Use a calculator
- The first example is short the second is more involved

Notebook 8.1

Here we switch to Jupyter Notebook to view the questions.

Exercise Sheet 7
Review

Exercise set 7

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

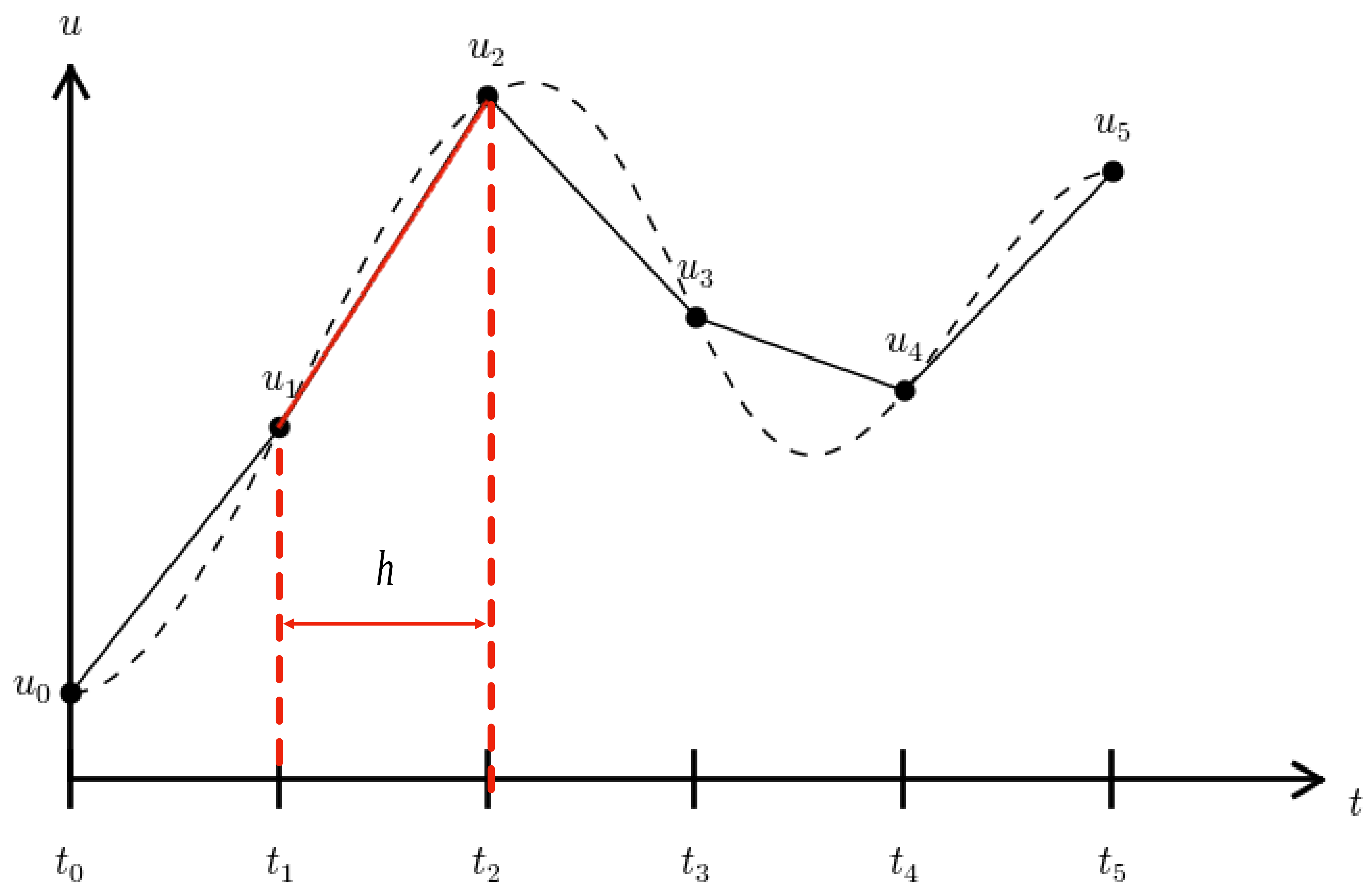
$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = G_0(h),$$

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?



1. Three numerical derivatives have to be calculated and then combined to improve accuracy

$E_0(h) = 0.22956$

$E_1(h) = 0.01554$

$E_0(\frac{h}{2}) = 0.10701$

$E_2(h) = 0.00039$

$E_1(\frac{h}{2}) = 0.00359$

$E_0(\frac{h}{4}) = 0.05171$

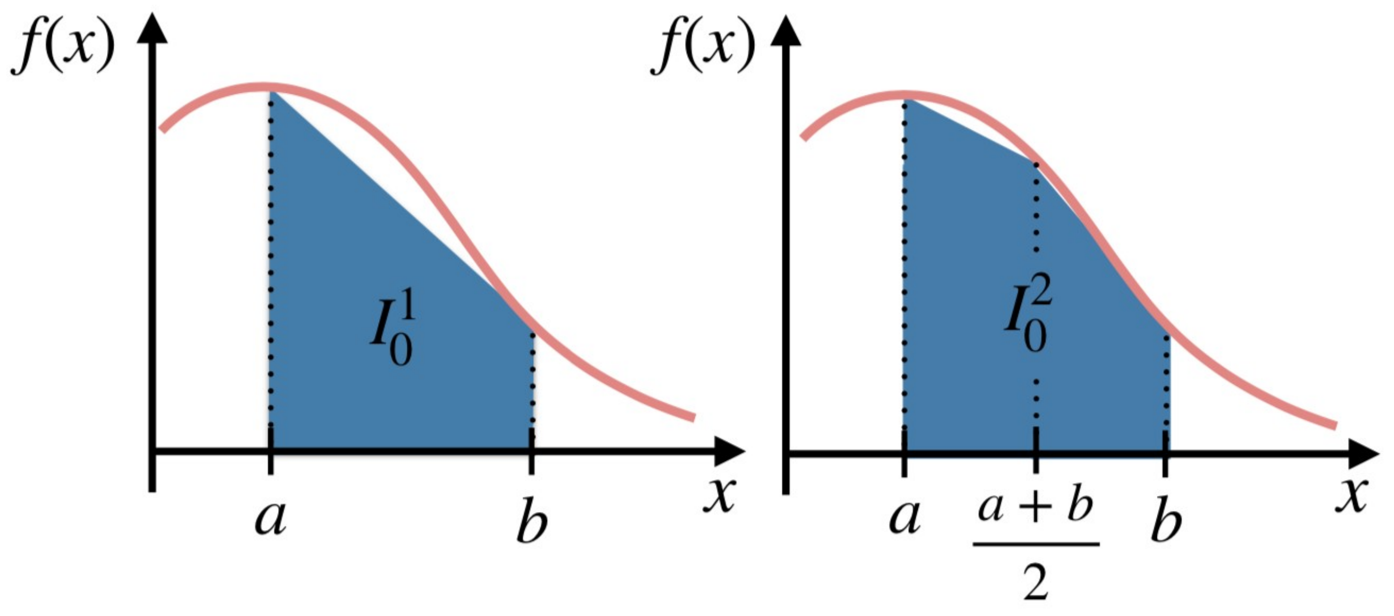
1. 2. The error does indeed decrease over the iterations

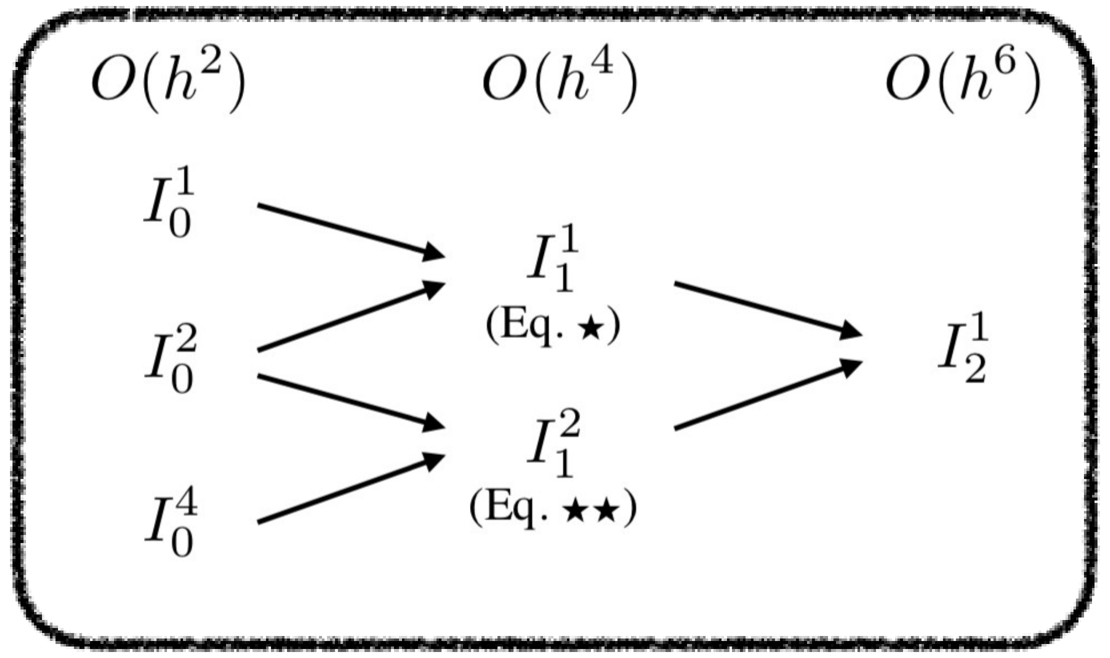
Exercise set 7

Question 2: Pseudocode for Romberg integration

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

$$I_{T_i} = \frac{f(x_{i+1}) - f(x_i)}{2h}$$





1. Precompute all function values
2. Calculate the initial integrals (composite trapezoidal rule)
3. Perform Romberg integration

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 $\int_a^b f(x)\, dx$

Steps:

`maxNumIntervals` $\leftarrow 2^K$

1 // Precompute and store function evaluations
`hmin` $\leftarrow (b - a)/$`maxNumIntervals`
for $i \leftarrow 0, \ldots,$ `maxNumIntervals` **do**
  `fvalues`$[i] \leftarrow f(a + i *$ `hmin`$)$
end for

2 // Compute level 0 integrals
for $r \leftarrow 0, \ldots, K$ **do** // refinement
  `numIntervals` $\leftarrow 2^r$
  `step` $\leftarrow 2^{K-r}$ // step between two function evaluations for this refinement
  `result` $\leftarrow 0$
  for $i \leftarrow$ `step`, $2 *$ `step`, $3 *$ `step`, $\ldots,$ `maxNumIntervals` $-$ `step` **do**
    `result` $\leftarrow$ `result` $+$ `fvalues`$[i]$
  end for
  // composite trapezoidal rule:
  `integral`$[0, r] \leftarrow 0.5 \frac{b-a}{\texttt{numIntervals}}(\texttt{fvalues}[0] + \texttt{fvalues}[\texttt{maxNumIntervals}]$
    $+2 * \texttt{result})$
end for

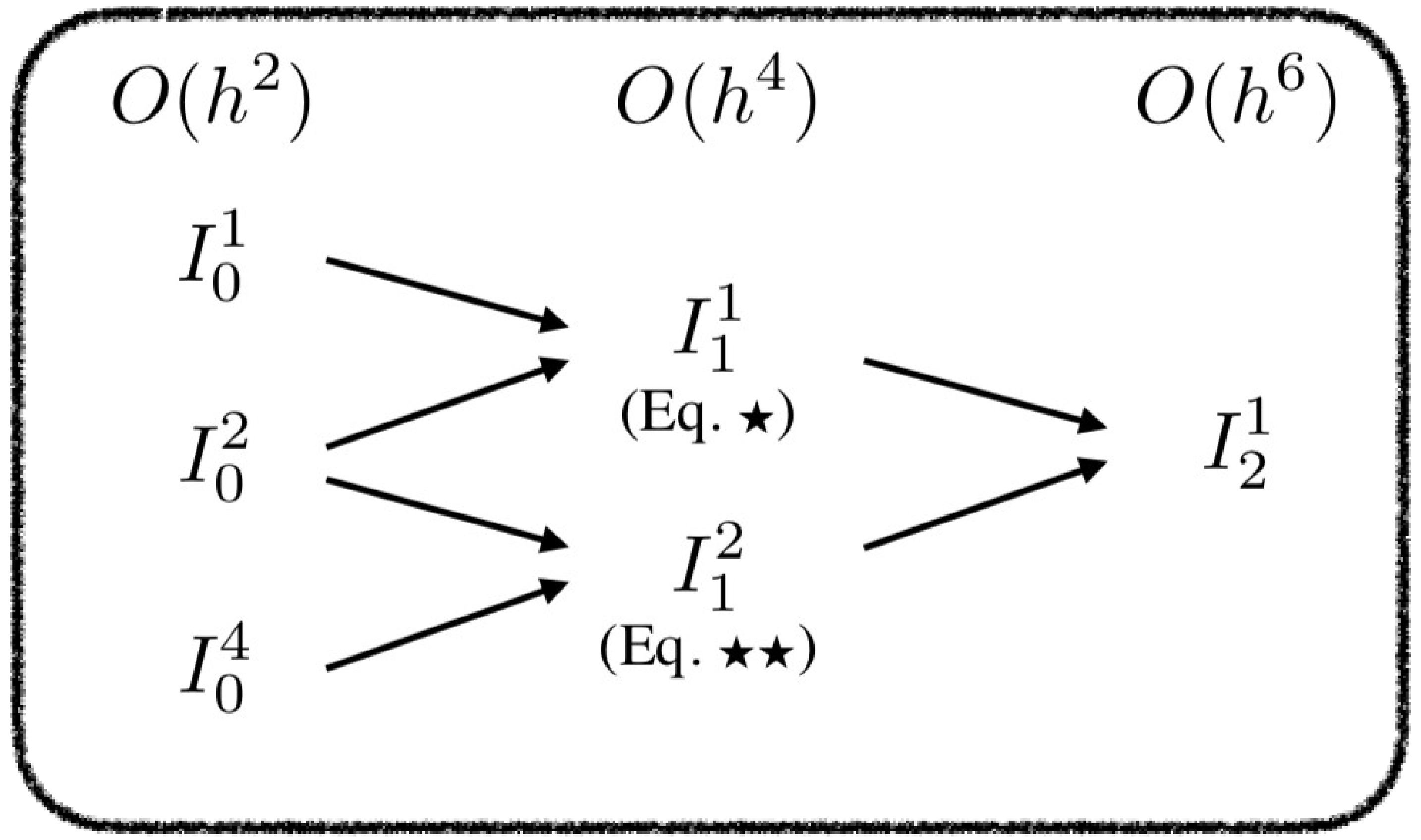
3 //Advance to higher precision according to Romberg
for $l \leftarrow 1, \ldots, K$ **do** //level
  for $r \leftarrow 0, \ldots, K - l$ **do** // refinement
    `integral`$[l, r] \leftarrow \frac{4^l * \texttt{integral}[l-1, r+1] - \texttt{integral}[l-1, r]}{4^l - 1}$
  end for
end for

Exercise set 7

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: $\frac{\sin(0)}{0} = 1$)



1. Analogue to the first Question

$$I_0^1 = \frac{\pi}{2}$$

$$I_0^2 = \frac{4+\pi}{4}$$

$$I_0^4 = \frac{12+16\sqrt{2}+3\pi}{24}$$

$$I_1^1 = \frac{8+\pi}{6}$$

$$I_1^2 = \frac{12+32\sqrt{2}+3\pi}{36}$$

$$I_2^1 = \frac{72+256\sqrt{2}+21\pi}{270}$$

Notebook 7.1

Here we switch to Jupyter Notebook to view the solutions.