Numerical integration

Compute $I = \int_a^b f(x)dx$

Once again, we should assume that f(x) is expensive to evaluate, e.g. f(x) may be the result of a computer simulation of some physical system with input x.

Example of use: Robotic spray painting

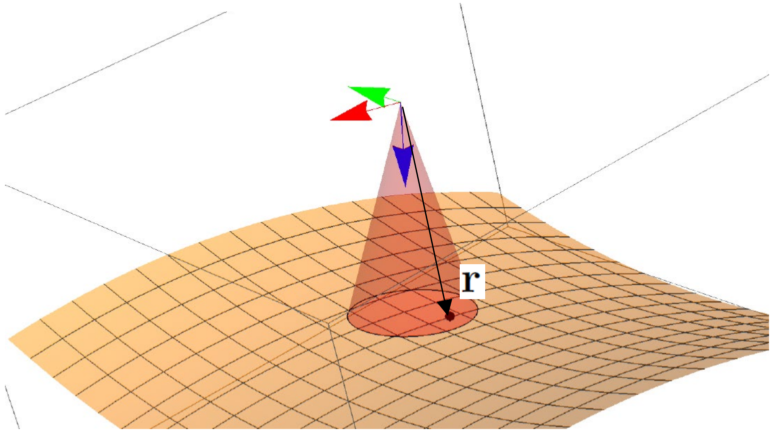
- Optimize robot trajectory so that the paint coverage thickness is close to the specified value everywhere
- Paint is coming from nozzle (wet paint)
- Model the paint flux from the nozzle
- Compute the resulting paint coverage thickness
- Requires CAD model of the object to be painted

Paint flux model

$$\mathbf{j}(\mathbf{r},\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3)=\alpha\frac{Q\left(\frac{\mathbf{r}\cdot\mathbf{u}_3}{\mathbf{r}^T\mathbf{A}\mathbf{r}}\right)}{(\mathbf{r}^T\mathbf{A}\mathbf{r})^{\frac{3}{2}}}\mathbf{r}$$

$$\mathbf{A}=\mathbf{u}_1\mathbf{u}_1^T+\alpha^2\mathbf{u}_2\mathbf{u}_2^T+\mathbf{u}_3\mathbf{u}_3^T$$

The vector u3 (blue is in the direction of the cone axis of the ellipsoidal spray cone, and u1, u2 (red, green) are principal axes of the ellipsoid. The vector r (black dot) is the vector from the nozzle to the point of interest on the surface measured in the coordinate frame (u1,u2,u3) at the nozzle. Q(x) is a nozzle specific function that can be calibrated.



The flux model is established based on the following three conditions:

The streamlines radiate from the tool center, which ensures that the paint flux is contained within a cone whose silhouettes are straight lines,

The divergence of j must be zero, which ensures that the total flux across any surface completely intersecting the cone is a constant, expressing conservation of paint.

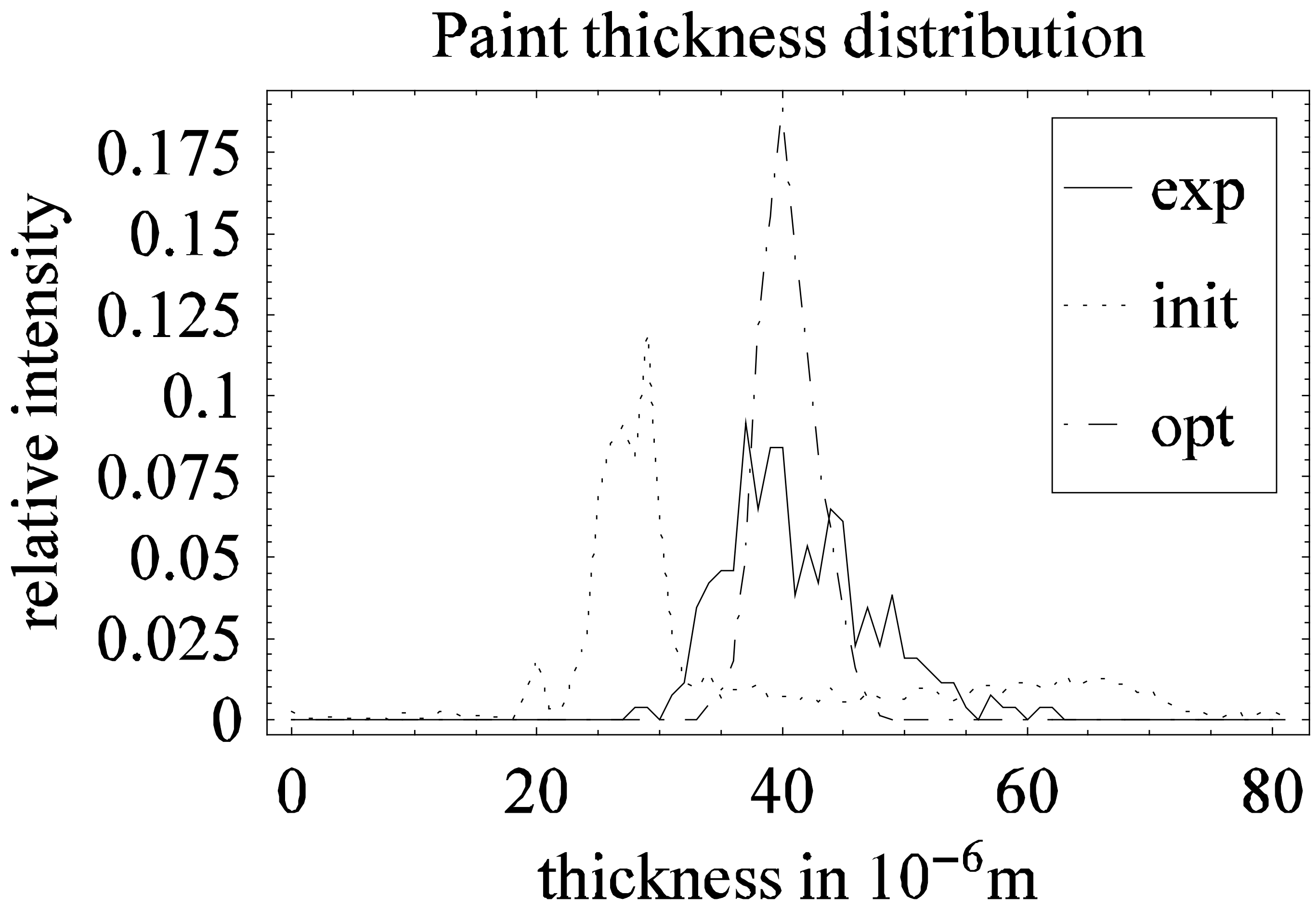
If the cone is intersected by a plane perpendicular to the cone axis, the curves of constant flux in this plane are ellipses,

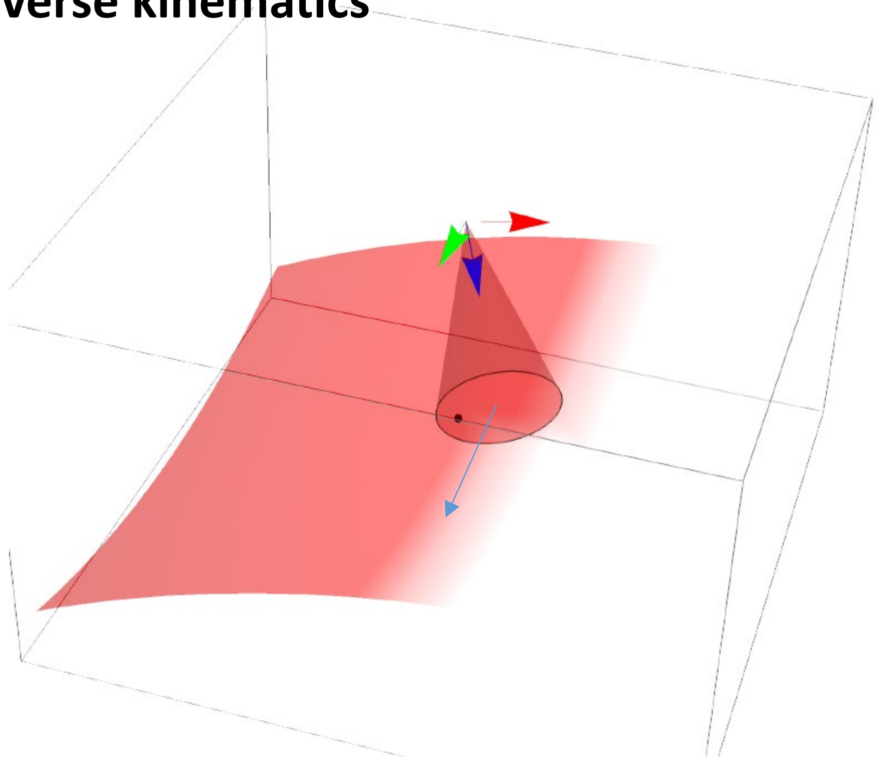
The coverage thickness at a point **R** on the surface with surface normal **N** is then computed by integrating **j**(**R**-**p**(t),$\mathbf{u}_1$(t),$\mathbf{u}_2$(t),$\mathbf{u}_3$(t)).**N** over the time periods where the cone is hitting **R**, where {**p**(t), $\mathbf{u}_1$(t),$\mathbf{u}_2$(t),$\mathbf{u}_3$(t)} is given by the nozzle (robot) trajectory.

Optimize paint nozzle trajectory, and find robot movement by Inverse kinematics

$$cov(\mathbf{R}) = \int_{t_1}^{t_2} \mathbf{j}(\mathbf{R} - \mathbf{p}(t), \mathbf{u}_1(t), \mathbf{u}_2(t), \mathbf{u}_3(t)) \cdot \mathbf{N} dt$$

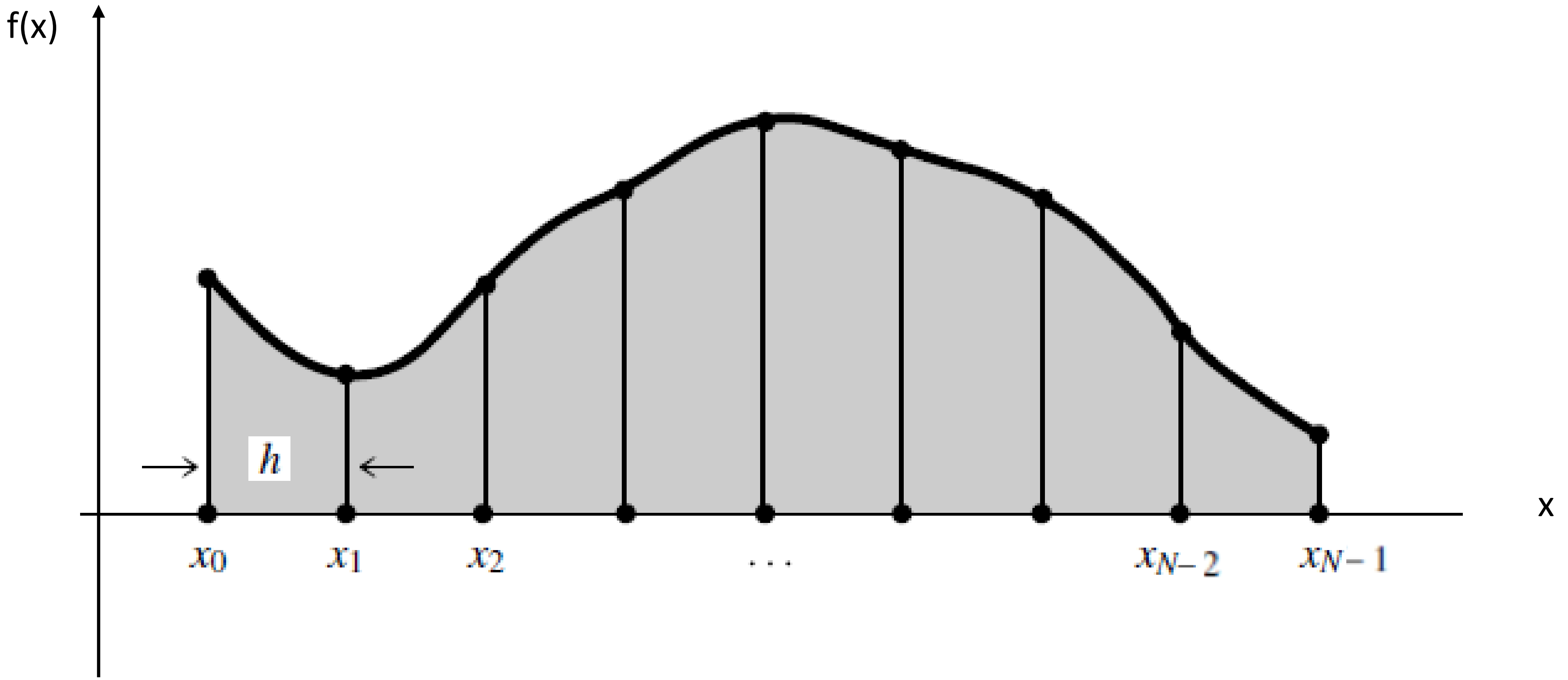
Wheelbarrow example result:





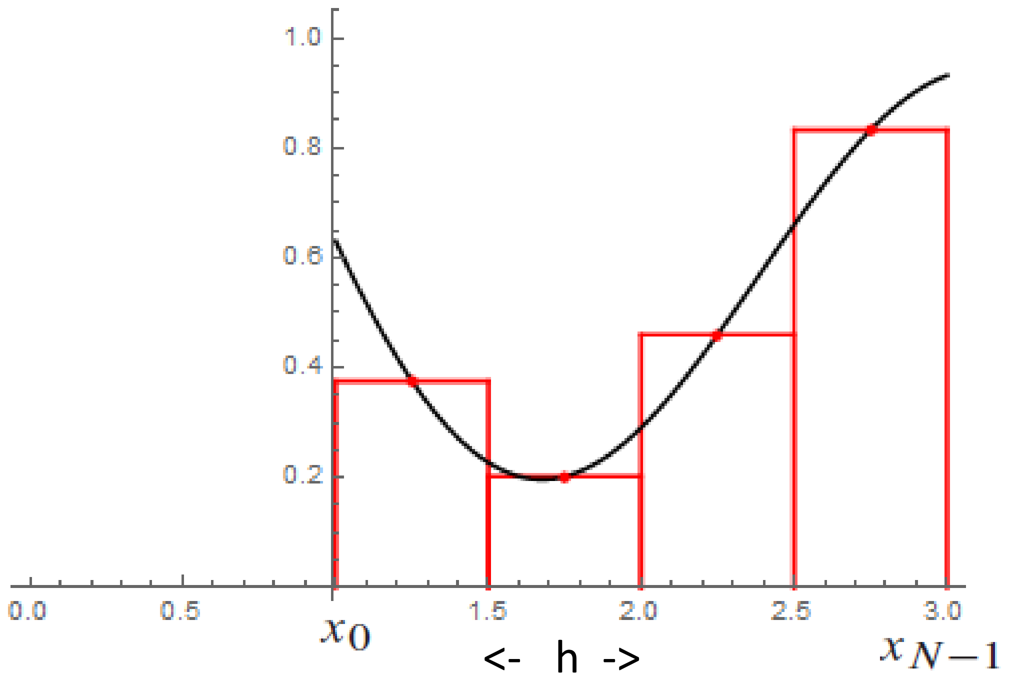
Compute $I = \int_a^b f(x)dx$

Only requirement: For an input x, we need a method to compute f(x)



Discretization : $x_i = x_0 + ih \qquad i = 0, 1, \ldots, N\text{-}1$

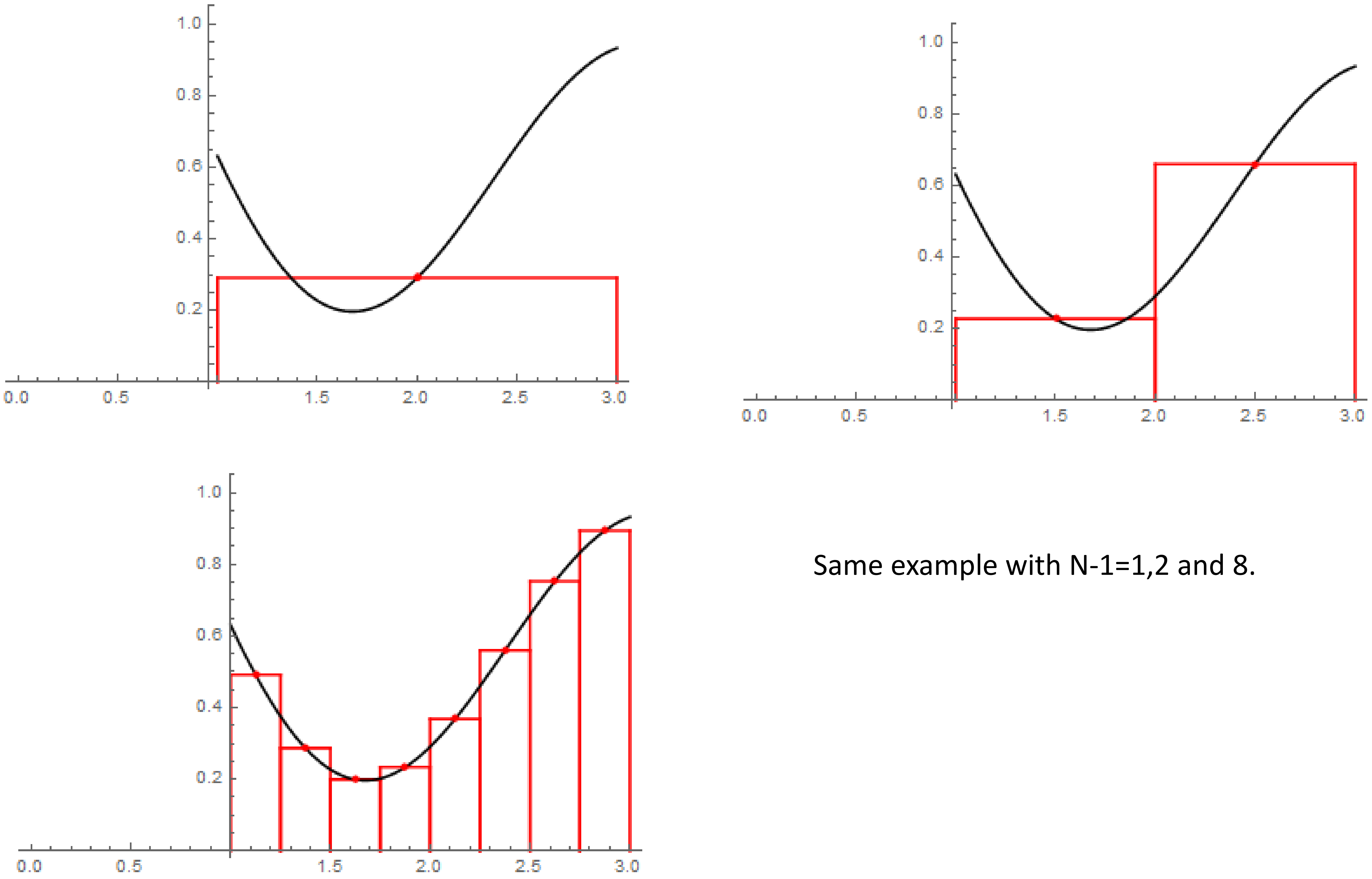
Extended Midpoint (Rectangle) Method



$$I = \int_a^b f(x)dx$$

We work on an example with a=1, b=3. We put x_0 =a and x_{N-1}=b. Here we show N-1=4.

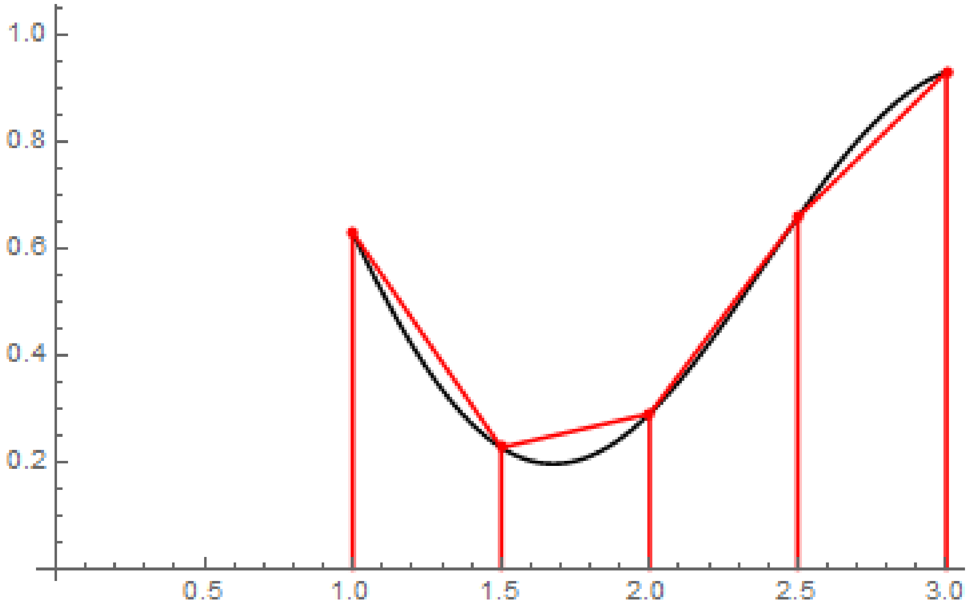
$$\int_{x_0}^{x_{N-1}} f(x)dx = h[f_{1/2} + f_{3/2} + f_{5/2} + \cdots + f_{N-5/2} + f_{N-3/2}] + O\left(\frac{1}{N^2}\right) \quad (4.1.19)$$



Same example with N-1=1,2 and 8.

Trapezoidal Method

$$I = \int_a^b f(x)dx$$

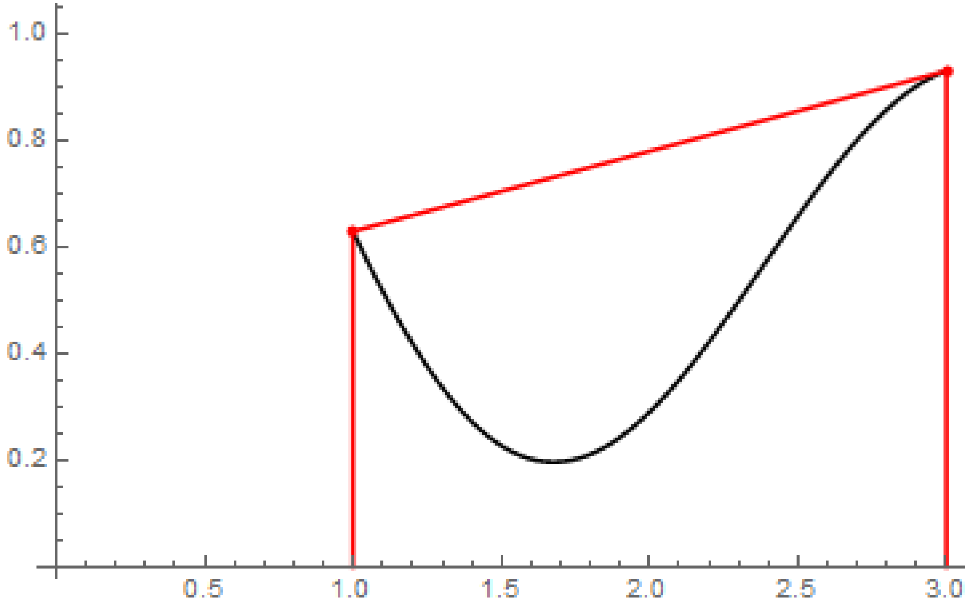


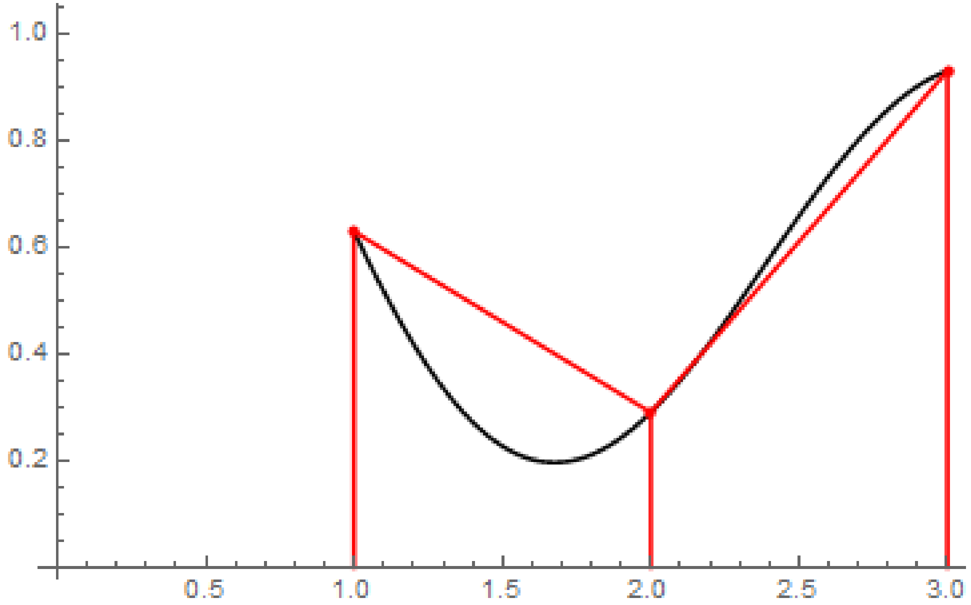
x_0 <- h -> x_{N-1}

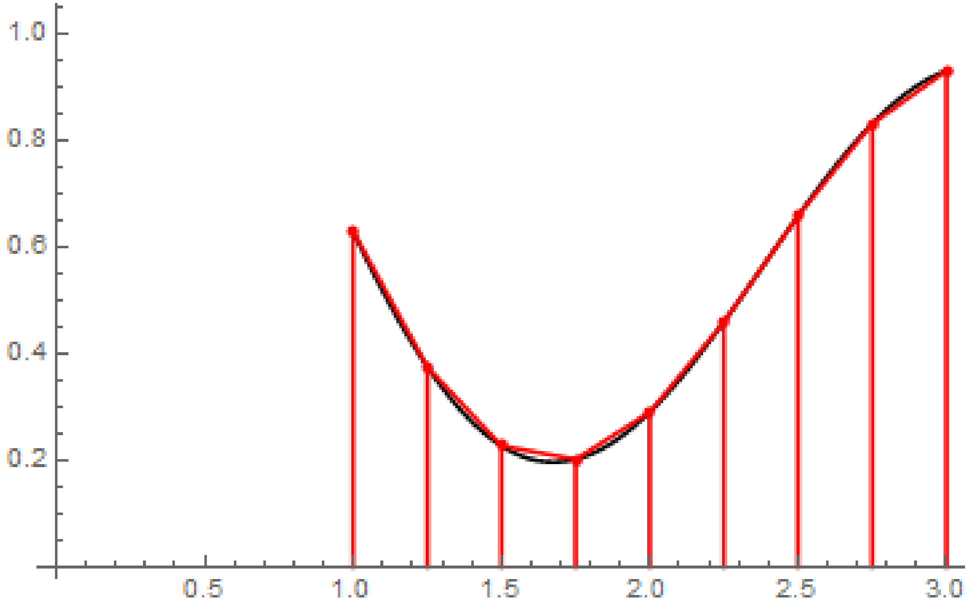
We work on an example with a=1, b=3. We put x_0 =a and x_{N-1} =b. Here we show N-1=4.

$$\int_{x_0}^{x_1} f(x)dx = h\left[\frac{1}{2}f_0 + \frac{1}{2}f_1\right]$$

$$\int_{x_0}^{x_{N-1}} f(x)dx = h\left[\frac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{N-2} + \frac{1}{2}f_{N-1}\right] + O\left(\frac{(b-a)^3 f''}{N^2}\right) \tag{4.1.11}$$



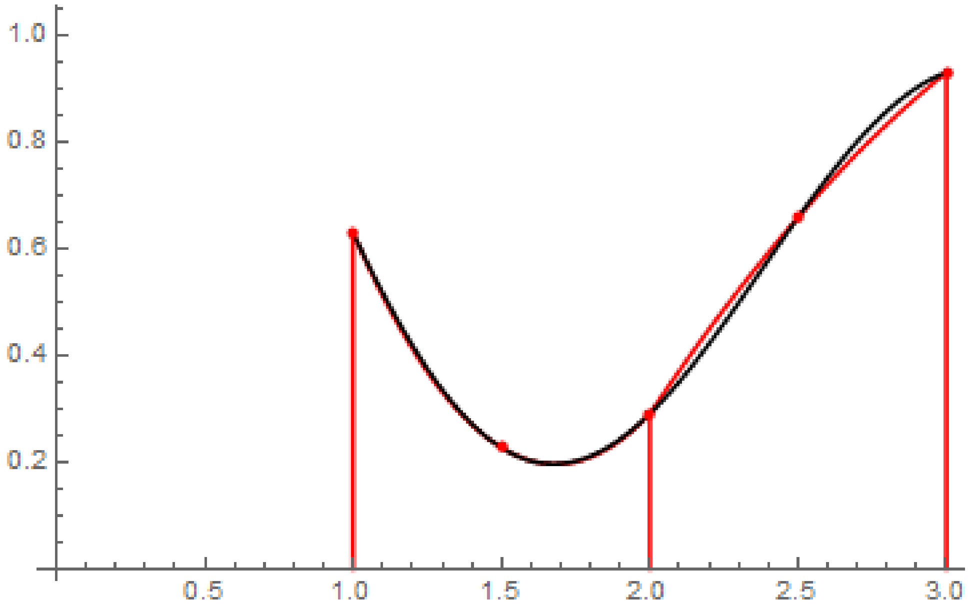




Same example with N-1=1,2 and 8.

Simpsons Method

$$I = \int_a^b f(x)dx$$



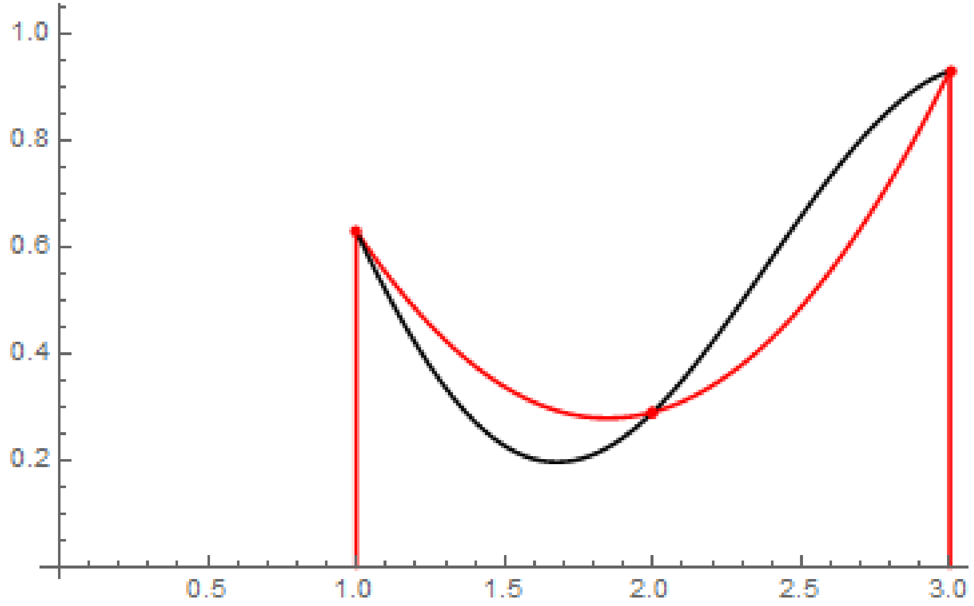
x_0 <- h -> x_{N-1}

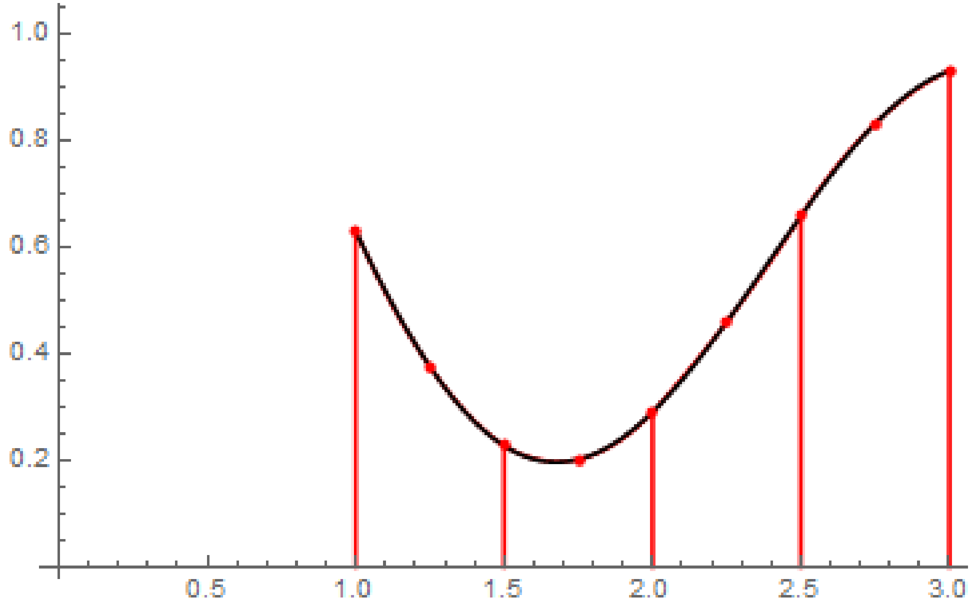
We work on an example with a=1, b=3. We put x_0 =a and x_{N-1}=b. Here we show N-1=4.

The subintervals between vertical bars are fitted with a parabola through the endpoints and the midpoint.

$$\int_{x_0}^{x_2} f(x)dx = h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{1}{3}f_2\right]$$

$$\int_{x_0}^{x_{N-1}} f(x)dx = h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{2}{3}f_2 + \frac{4}{3}f_3 + \cdots + \frac{2}{3}f_{N-3} + \frac{4}{3}f_{N-2} + \frac{1}{3}f_{N-1}\right] + O\left(\frac{1}{N^4}\right) \quad (4.1.13)$$





Same example with N-1=2 and 8.

Summary on Newton-Cotes quadratures

Extended midpoint (rectangle) method (interpolation with constant functions

$$\int_{x_0}^{x_{N-1}} f(x)dx = h[f_{1/2} + f_{3/2} + f_{5/2} + \cdots + f_{N-5/2} + f_{N-3/2}] \quad + O\left(\frac{1}{N^2}\right) \qquad (4.1.19)$$

Order 2

Trapezoidal method (interpolation with linear functions)

$$\int_{x_0}^{x_{N-1}} f(x)dx = h\left[\frac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{N-2} + \frac{1}{2}f_{N-1}\right] \quad + O\left(\frac{1}{N^2}\right) \qquad (4.1.11)$$

Order 2

Simpsons method (interpolation with parabolas)

$$\int_{x_0}^{x_{N-1}} f(x)dx = h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{2}{3}f_2 + \frac{4}{3}f_3 + \cdots + \frac{2}{3}f_{N-3} + \frac{4}{3}f_{N-2} + \frac{1}{3}f_{N-1}\right] \quad + O\left(\frac{1}{N^4}\right) \qquad (4.1.13)$$

Order 4

But how can we estimate errors ?

- Input using the Trapezoidal method

$$\begin{pmatrix} \text{N-1} & \text{Numerical Approx.} \\ 1. & 0.593529 \\ 2. & 0.590981 \\ 4. & 0.590287 \\ 8. & 0.59011 \end{pmatrix}$$

What is the accuracy on this result?

Richardson extrapolation

(derivation of the results below is published in the weekly plan – mainly for completeness.)

A(h) is a numerical approximation to an exact value A where h is the stepsize. For example for numerical integration.

Estimation of the order k:

$$\frac{A(h_1) - A(h_2)}{A(h_2) - A(h_3)} \approx \alpha^k \quad \text{for} \quad h_1/h_2 = h_2/h_3 = \alpha. \qquad h_1 > h_2 > h_3$$

Typically alpha=2.

Error estimation and extrapolation:

Extrapolation

$$\begin{aligned} A_R(h_2, h_1) &\equiv \frac{\alpha^k A(h_2) - A(h_1)}{\alpha^k - 1} \\ &= A(h_2) + \frac{A(h_2) - A(h_1)}{\alpha^k - 1}, \quad \alpha = \frac{h_1}{h_2} \qquad h_1 > h_2 \end{aligned}$$

Error estimate on $A(h_2)$

Example from before (Trapezoidal method)

i	A(hi)	A(hi-1)-A(hi)	Rich-alp^k	A(hi)-A	Rich. error estimate	Total f-comp.
1.	0.593529	*	*	0.00347808	*	2.
2.	0.590981	0.00254788	*	0.000930202	*	3.
3.	0.590287	0.000694194	3.67027	0.000236008	0.000231398	5.
4.	0.59011	0.000176796	3.92653	0.0000592124	0.0000589319	9.
5.	0.590066	0.0000443962	3.98222	0.0000148161	0.0000147987	17.
6.	0.590055	0.0000111113	3.99559	3.70485×10^{-6}	3.70376×10^{-6}	33.
7.	0.590052	2.77859×10^{-6}	3.9989	9.26263×10^{-7}	9.26196×10^{-7}	65.
8.	0.590051	6.94694×10^{-7}	3.99973	2.31569×10^{-7}	2.31565×10^{-7}	129.
9.	0.590051	1.73677×10^{-7}	3.99993	5.78925×10^{-8}	5.78922×10^{-8}	257.
10.	0.590051	4.34193×10^{-8}	3.99998	1.44731×10^{-8}	1.44731×10^{-8}	513.

If the order seems to be as expected, extrapolation/error computation must be to the expected order (here 2 as alp^k is around 4) and not the computed order.

Exercises

$$\int_0^1 \cos(x^2)\exp(-x)dx$$

$$\int_0^1 \sqrt{x}\cos(x^2)\exp(-x)dx$$

$$\int_0^1 \frac{1}{\sqrt{x}}\cos(x^2)\exp(-x)dx$$

$$\int_0^1 1000\exp(-1/x)\exp(-1/(1-x))dx$$

For the first exercise try it with all three Newton-Cotes methods.
For the second exercise use Simpson and for the third exercise use the Rectangle method. For the last exercise use Trapezoidal.

Make tables as on slide 15 (previous slide) except the deviation to the unknown exact value A.

Newton-Cotes methods are easy to program, so I recommend that you implement your own program from scratch.