NTNU

Norwegian University of Science and Technology

TMA4125 Matematikk 4N

Numerical Integration

Ronny Bergmann

Department of Mathematical Sciences, NTNU.

January 25, 2022

Introduction.

Imagine you want to compute the (finite) integral

$$I[f](a,b) := \int_a^b f(x)\,\mathrm{d}x$$

The "usual" way is to find a primitive function F (also known as indefinite integral f) satisfying $F'(x) = f(x)$. Then we can compute

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

Challenge. Computing F analytically might be hard or F might not have a closed analytical form. For example

$$f(x) = \mathrm{e}^{-x^2} \qquad \text{(no elementary function } F\text{)}$$

$$f(x) = \frac{\log(2 + \sin(\frac{1}{2} - \sqrt{x})^6)}{\log(\pi + \arctan(\sqrt{1 - \exp(-2x - \sin(x))}))} \qquad \text{(complicated)}$$

Numerical quadrature.

A numerical quadrature or a quadrature rule is a formula for approximating $I[f](a, b)$. Quadratures are usually of the form

$$Q[f](a, b) = \sum_{i=0}^{n} w_i f(x_i),$$

where x_i, w_i, $i = 0, 1, \ldots, n$, are the nodes (points) and the weights of the quadrature rule, respectively.

A quadrature rule $Q[f](a, b)$ is defined by its quadrature nodes $\{x_i\}_{i=0}^n$ and weights $\{w_i\}_{i=0}^n$

- If f is given from the context, we write just short $I(a, b)$ and $Q(a, b)$.
- quadrature rules are linear, i. e. for functions f, g and $\alpha, \beta \in \mathbb{R}$ it holds
$$Q[\alpha f + \beta g](a, b) = \alpha Q[f](a, b) + \beta Q[g](a, b)$$

Known examples.

You already know from Calculus 1:

Mid point rule. The mid point rule is the simplest possible rule

$$M[f](a,b) := w_0 f(x_0) = (b-a) f\left(\frac{a+b}{2}\right)$$

The only node is the mid point $x_0 = \frac{a+b}{2}$ with weight $w_0 = b - a$.
Note. Instead of Q we use specific letters for these quadrature rules.

Known examples.

You already know from Calculus 1:

Trapezoidal rule. We use both boundaries to form a trapezoid.

$$T[f](a,b) := w_0 f(x_0) + w_1 f(x_1) = (b-a) f\left(\frac{f(a)+f(b)}{2}\right)$$

So here we have $x_0 = a$, $x_1 = b$ and $w_0 = w_1 = \frac{b-a}{2}$.
Note. Instead of Q we use specific letters for these quadrature rules.

Known examples.

You already know from Calculus 1:

Simpson rule. We use all 3 nodes from before

$$S[f](a,b) := w_0 f(x_0) + w_1 f(x_1) + w_2 f(x_2) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

with $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$ and weights $w_0 = w_2 = \frac{b-a}{6}$ and $w_1 = \frac{2(b-a)}{3}$.
Note. Instead of Q we use specific letters for these quadrature rules.

Roadmap

1. construct the (known) quadratures from integration of interpolation polynomials
2. error analysis
3. composite quadrature rules – how to “divide and conquer”
4. adaptive quadrature rules – how to “divide cleverly”
5. Newton-Côtes & Gauß quadrature

Quadrature from integrating interpolation polynomials

Recap. Choose $n+1$ distinct nodes $x_0, \dots, x_n$ in the interval $[a,b]$.
Denote by p_n the interpolation polynomial satisfying the interpolation conditions

$$p_n(x_i) = f(x_i), \qquad i = 0, \dots, n.$$

Idea. Integrating polynomials is easy!

$\Rightarrow$ Use $\displaystyle\int_a^b p_n(x)\,\mathrm{d}x$ as an approximation to $\displaystyle\int_a^b f(x)\,\mathrm{d}x$.

We consider the quadrature

$$I[f](a,b) \approx Q[f](a,b) := \int_a^b p_n(x)\,\mathrm{d}x.$$

But what about the weights?

Weights for the quadrature based on p_n

To compute the weights we use the Lagrange form:

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x), \qquad \text{where } \ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, \ldots, n$$

Due to linearity of the integral we get for the weights w_i

$$\begin{aligned} Q[f](a,b) &= \int_a^b p_n(x)\,\mathrm{d}x = \int_a^b \sum_{i=0}^{n} f(x_i)\ell_i(x)\,\mathrm{d}x \\ &= \sum_{i=0}^{n} f(x_i) \int_a^b \ell_i(x)\,\mathrm{d}x = \sum_{i=0}^{n} f(x_i) w_i \end{aligned}$$

So the weights are simply computed as

$$w_i = \int_a^b \ell_i(x)\,\mathrm{d}x, \quad i = 0, \ldots, n,$$

and are independent of f.

Revisiting some old and new quadratures

Let's consider $a = 0$ and $b = 1$ for $f(x) = \cos(\frac{\pi x}{2})$

Then we can compute (analytically)

$$I(0,1) = \int_0^1 \cos\left(\frac{\pi x}{2}\right) = \frac{2}{\pi} = 0.636619\ldots$$

Goal. Since we know the exact solution here can check how good the following rules are.

The trapezoidal rule revisited

Let $n = 1$, and take $x_0 = 0$, $x_1 = 1$.
We obtain for the cardinal functions and weights:

$$\ell_0(x) = 1 - x, \qquad w_0(x) = \int_0^1 (1-x)\,\mathrm{d}x = \frac{1}{2}$$

$$\ell_1(x) = x, \qquad w_1(x) = \int_0^1 x\,\mathrm{d}x = \frac{1}{2}$$

And the corresponding quadrature rule is actually the trapezoidal rule $T(a, b)$ with $[a, b] = [0, 1]$

$$T(0,1) = \frac{1}{2}\big(f(0) + f(1)\big).$$

Exercise. show that on $[a, b]$ with $n = 1$, $x_0 = a$, $x_1 = b$ this approach yields the general trapezoidal rule.

The Gauß-Legendre quadrature

This also works for more complicated choice of the nodes x_0, x_1.

Let $n = 1$, and take $x_0 = \frac{1}{2} - \frac{\sqrt{3}}{6}$, $x_1 = \frac{1}{2} + \frac{\sqrt{3}}{6}$.
We obtain for the cardinal functions and weights:

$$\ell_0(x) = -\sqrt{3}x + \frac{1+\sqrt{3}}{2}, \qquad w_0(x) = \int_0^1 \ell_0(x)\,\mathrm{d}x = \frac{1}{2}$$

$$\ell_1(x) = \sqrt{3}x + \frac{1-\sqrt{3}}{2}, \qquad w_1(x) = \int_0^1 \ell_1(x)\,\mathrm{d}x = \frac{1}{2}$$

And the corresponding quadrature rule is

$$Q(0,1) = \frac{1}{2}\left(f\left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right) + f\left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)\right).$$

The Simpson's rule revisited

We construct the Simpson's rule on the interval $[0, 1]$ by choosing the nodes $x_0 = 0, x_1 = \frac{1}{2}, x_2 = 1$.
The corresponding cardinal functions are

$$\ell_0(x) = 2(x - \frac{1}{2})(x-1), \quad \ell_1(x) = 4x(1-x), \quad \ell_2(x) = 2x(x - \frac{1}{2})$$

and we obtain the weights

$$w_0 = \int_0^1 \ell_0(x)\,\mathrm{d}x = \frac{1}{6}, \quad w_1 = \int_0^1 \ell_1(x)\,\mathrm{d}x = \frac{4}{6} \quad w_2 = \int_0^1 \ell_2(x)\,\mathrm{d}x = \frac{1}{6},$$

such that $I[f](0,1) = \int_0^1 f(x)\,\mathrm{d}x$ can be approximated by

$$S[f](0,1) = \int_0^1 p_2(x)\,\mathrm{d}x = \sum_{i=0}^{2} w_i f(x_i)$$
$$= \frac{1}{6}\Big(f(0) + 4f\big(\frac{1}{2}\big) + f(1)\Big).$$

Exercise. Accuracy of quadrature rules

Back to our example.

For $I(0,1) = \int_0^1 \cos\left(\frac{\pi x}{2}\right)\,\mathrm{d}x = \frac{2}{\pi} = 0.636619$ we can now check, how accurate / good the quadratures are since we have the actual value for comparison.

Let's compare a few.

Remark. Observe that the Gauß-Legendre quadrature gives a much more accurate answer than the trapezoidal rule. The choice of nodes clearly matters. Simpson's rule gives very similar results to Gauß-Legendre quadrature but uses 3 instead of 2 quadrature nodes.

Degree of exactness

Definition. A numerical quadrature has degree of exactness d if

$$Q[p](a,b) = I[p](a,b) \quad \text{for all} \quad p \in \mathbb{P}_d$$

and there is at least one $p \in \mathbb{P}_{d+1}$ such that $Q[p](a,b) \neq I[p](a,b)$.

Since both integrals and quadratures are linear, the degree of exactness is d if

$$I[x^j](a,b) = Q[x^j](a,b), \qquad j = 0,\dots,d$$
$$I[x^{d+1}](a,b) \neq Q[x^{d+1}](a,b).$$

Degree of exactness for some quadrature rules

Observation. All quadratures constructed from Lagrange interpolation polynomials in $n+1$ distinct nodes will automatically have a degree of exactness of least n. This follows immediately from the fact the interpolation polynomial $p_n \in \mathbb{P}_n$ of any polynomial $q \in \mathbb{P}_n$ is just the original polynomial q itself.

We could do this on paper (it's not so hard) or convince ourselves numerically. How?

We get

- the trapezoidal rule: $n+1=2$ points and degree of exactness 1
- the Simpson rule: $n+1=3$ points and degree of exactness 3
- the Gauß-Legendre rule: $n+1=2$ points and degree of exactness 3.

An error estimate for a quadrature rules

Theorem. (Error Estimate for quadratures with degree of exactness n)

Assume that $f \in C^{n+1}[a,b]$ and let $Q[\cdot](a,b)$ be a quadrature rule with nodes $\{x_i\}_{i=0}^n$ and weights $\{w_i\}_{i=0}^n$ which has degree of exactness n.

Then the quadrature error $|I[f] - Q[f]|$ can be estimated by

$$|I[f] - Q[f]| \leq \frac{M}{(n+1)!} \int_a^b \prod_{i=0}^n |x - x_i| \, \mathrm{d}x$$

where $M = \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|$.

Proof of the error estimate

Let $p_n \in \mathbb{P}_n$ denote the interpolation polynomial satisfying $p_n(x_i) = f(x_i), i = 0, \dots, n$

We know from the error of interpolation

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

for some $\xi(x) \in (a, b)$. Since $Q(a, b)$ has degree of exactness n we have $I[p_n] = Q[p_n] = Q[f]$ and thus

$$\begin{aligned}
|I[f] - Q[f]| = |I[f] - I[p_n]| &\leq \int_a^b |f(x) - p_n(x)| \,\mathrm{d}x \\
&= \int_a^b \left| \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right| \mathrm{d}x \\
&\leq \frac{M}{(n+1)!} \int_a^b \prod_{i=0}^{n} |x - x_i| \,\mathrm{d}x
\end{aligned}$$

This concludes the proof.

Improved error bounds

While this theorem was easy to prove, one can often find sharper bounds (better estimates of the error) for specific cases. For example (without proof)

Theorem. For the trapezoidal rule and $f \in C^2[a, b]$, there is a $\xi \in (a, b)$ such that

$$I[f] - T[f] = \frac{(b-a)^3}{12} f''(\xi).$$

For Simpson we use the following idea: Let's Taylor expand $I[f](a, b)$ and $S[f](a, b)$ around the center point $c = \frac{a+b}{2}$.

Indeed one can show.

Theorem. For Simpson's rule $S(a, b)$ and $f \in C^4[a, b]$, there is a $\xi \in (a, b)$ such that

$$I[f] - S[f] = -\frac{(b-a)^5}{2880} f^{(4)}(\xi).$$

Quadrature in Practice: Divide and Conquer

In the following, you will learn the steps on how to construct realistic algorithms for numerical integration, similar to those used in software like Matlab or SciPy/NumPy. The steps are:

1. Choose $n+1$ distinct nodes on a standard interval $[-1, 1]$.
2. Let $p_n(x)$ be the polynomial interpolating some general function f in the nodes, and let the $Q[f](-1, 1) = I[p_n](-1, 1)$.
3. Transfer the formula Q from $[-1, 1]$ to some interval $[a, b]$.
4. Find the composite formula, by dividing the interval $[a, b]$ into subintervals and applying the quadrature formula on each subinterval.
5. Find an expression for the error $E[f](a, b) = I[f](a, b) - Q[f](a, b)$.
6. Find an expression for an estimate of the error, and use this to create an adaptive algorithm.

Constructing quadrature rules on a single interval

We already have seen how to construct quadrature rules based on polynomial interpolation:

For $n+1$ quadrature points $\{x_i\}_{i=0}^{n} \subset [a,b]$ compute the weights by

$$w_i = \int_a^b \ell_i(x)\,\mathrm{d}x, \qquad \text{for } i = 0,\dots,n$$

where ℓ_i are (again) the cardinal functions.

$\Rightarrow$ resulting quadrature rule has (at least) exactness equal to n.

Transfer the formula from $[-1, 1]$ to $[a, b]$

NTNU

What if we have different intervals to tackle, say $[a, b]$ and $[c, d]$?

Construct your method on a reference interval $\hat{I} = [-1, 1]$, determine your quadrature points $\{\hat{x}_i\}_{i=0}^n$ and weights $\{\hat{w}_i\}_{i=0}^n$ and use the transformation

$$x = \frac{b-a}{2}\hat{x} + \frac{b+a}{2} \quad \text{so } \mathrm{d}x = \frac{b-a}{2}\,\mathrm{d}\hat{x}$$

and thus we define the points $\{x_i\}_{i=0}^n$ and weights $\{w_i\}_{i=0}^n$ for $[a, b]$ as

$$x_i = \frac{b-a}{2}\hat{x}_i + \frac{b+a}{2}, \quad w_i = \frac{b-a}{2}\hat{w}_i \qquad \text{for } i = 0, \ldots, n.$$

Short example: Simpson's rule

Simpson's rule on $[-1, 1]$ uses the nodes $t_0 = -1$, $t_1 = 0$ and $t_2 = 1$. With the cardinal functions

$$\ell_0 = \frac{1}{2}(t^2 - t), \qquad \ell_1(t) = 1 - t^2, \qquad \ell_2(t) = \frac{1}{2}(t^2 + t).$$

We get the weights

$$w_0 = \int_{-1}^{1} \ell_0(t)dt = \frac{1}{3}, \qquad w_1 = \int_{-1}^{1} \ell_1(t)dt = \frac{4}{3}, \qquad w_2 = \int_{-1}^{1} \ell_2(t)dt = \frac{1}{3}$$

such that

$$\int_{-1}^{1} f(t)dt \approx \int_{-1}^{1} p_2(t)dt = \sum_{i=0}^{2} w_i f(t_i) = \frac{1}{3}\Big(f(-1) + 4f(0) + f(1)\Big).$$

The transformation yields the points $x_0 = a$, $x_1 = \frac{b+a}{2}$, $x_2 = b$ and we get

$$S(a,b) = \frac{b-a}{6}\Big(f(a) + 4f\big(\frac{b+a}{2}\big) + f(b)\Big).$$

Improving a quadrature rule

To generate more accurate quadrature rules $Q[f](a, b)$ we have in principle two possibilities

- Increase the order of the interpolation polynomial used to construct $Q(a, b)$.
- Subdivide the interval $[a, b]$ into smaller subintervals and apply a quadrature rule on each of the subintervals, leading to Composite Quadrature Rules.

Composite quadrature rules

For a composite quadrature rule select $m \geq 2$ and divide $[a, b]$ into m equispaced subintervals

$$[x_{i-1}, x_i] \qquad \text{where } x_i = a + ih, \quad i = 1, \ldots, m, \quad h = \frac{b-a}{m}$$

Then for a given quadrature rule $Q[\cdot](x_{i-1}, x_i)$ and define the composite quadrature rule by

$$\int_a^b f(x)\,\mathrm{d}x \approx CQ(f)([x_{i-1}, x_i]_{i=1}^m) := \sum_{i=1}^m Q[f](x_{i-1}, x_i)$$

Composite Simpson's rule

Idea. Split $[a, b]$ into m subintervals, do Simpson's rule on each.
$\Rightarrow$ we also need the mid points. So:

Divide $[a, b]$ into $2m$ equal intervals of length $h = (b - a)/(2m)$. Let $x_j = a + jh$, $i = 0, \cdots, 2m$, and apply Simpson's rule on each subinterval $[x_{2j}, x_{2j+2}]$ (with nodes $x_{2j}, x_{2j+1}, x_{2j+2}$). The result is:

$$\int_a^b f(x)dx = \sum_{j=0}^{m-1} \int_{x_{2j}}^{x_{2j+2}} f(x)\,\mathrm{d}x \approx S_m(a, b) := \sum_{j=0}^{m-1} S(x_{2j}, x_{2j+2})$$

Plugging in all small Simpson's rules we get

$$\begin{aligned} S_m(a, b) &:= \sum_{j=0}^{m-1} \frac{h}{3}\Big(f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\Big) \\ &= \frac{h}{3}\left(f(x_0) + 4\sum_{j=0}^{m-1} f(x_{2j+1}) + 2\sum_{j=1}^{m-1} f(x_{2j}) + f(x_{2m})\right) \end{aligned}$$

Numerical Example for Composite Simpson's rule

We again consider $f(x) = \cos\left(\frac{\pi x}{2}\right)$.

Intuitively.
If we spend more points, the error should decrease.
But How much does it decrease – or in other words – how fast?

From the experiment we observe that the error is reduced by a factor of approx. $0.0625 = \frac{1}{16}$ when doubling the number of subintervals m.

Two interpretations:
In number of points m**.** If we write $E_m(a,b) = |I(a,b) - S_m(a,b)|$, then

$$\frac{1}{16}E_m(a,b) \approx E_{2m}(a,b)$$

In step size $h = \frac{b-a}{m}$**.** We have $16 = 2^4$ so the error has to behave like a constant C times h^4, since

$$C\left(\frac{h}{2}\right)^4 = \frac{C}{2^4}h^4 = \frac{C}{16}h^4$$

Towards an error estimate for composite Simpson's

From the error of Simpson's rule: For $f \in C^4[a,b]$ there is a $\xi \in (a,b)$ such that

$$I[f] - S[f] = -\frac{(b-a)^5}{2880} f^{(4)}(\xi).$$

Remember. If $f(x) = p(x) \in \mathbb{P}_3$ then $f^{(4)} \equiv 0 \Rightarrow$ Degree of exactness 3.

Approach. Apply this to every "small" Simpson's rule in the composite Simpson's rule

$$\int_a^b f(x)dx - S_m(a,b) = \sum_{j=0}^{m-1} \left(\int_{x_{2j}}^{x_{2j+2}} f(x)dx - \frac{h}{3}\Big(f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\Big) \right)$$
$$= \sum_{j=0}^{m-1} -\frac{(2h)^5}{2880} f^{(4)}(\xi_j)$$

where $\xi_j \in (x_{2j}, x_{2j+2})$. Using the generalized mean value theorem there is a $\xi \in (a,b)$ such that $\sum^{m-1} f^{(4)}(\xi_j) = m f^{(4)}(\xi)$.

An error estimate for composite Simpson's rule

By using $2mh = (b-a)$ in the previous idea we obtain the following theorem.

Theorem. Let $f \in C^4[a,b]$. Then there esists a $\xi \in (a,b)$ such that

$$\int_a^b f(x)dx - S_m(a,b) = -\frac{(b-a)h^4}{180} f^{(4)}(\xi).$$

For our ongoing numerical example $f(x) = \cos\left(\frac{\pi x}{2}\right)$ we have $f^{(4)}(x) = \frac{\pi^4}{16}\cos\left(\frac{\pi x}{2}\right)$ which is less than $\frac{\pi^4}{16}$ and we get

$$|I(a,b) - S_m(a,b)| \leq \frac{1}{180}\left(\frac{1}{2m}\right)^4\left(\frac{\pi}{2}\right)^4 = \frac{\pi^4}{46080}\frac{1}{m^4}$$

Interlude: Convergence of h-dependent approximations

Let X be an exact solution and $X(h)$ some numerical solution depending on h. Consider the error $e(h) = \|X - X(h)\|$.

The approximation $X(h)$ converges to X if $\lim_{h \to 0} e(h) = 0$.

The order of approximation is p if there exists a constant M such that

$$e(h) \leq Mh^p$$

In Big-$\mathcal{O}$ notation we simply write

$$e(h) = \mathcal{O}(h^p) \quad \text{as } h \to 0.$$

Usually we are interested in p not that much in M.
We get a measure of the quality of convergence.

Interlude: Convergence of h-dependent approximations

To numerically find p (approximately) in $e(h) = Ch^p$: Take some $H \in \mathbb{R}$.

1. Run Experiments Compute $e(h_k)$ with $h_k = \frac{H}{2^k}$, $k = 0, 1, 2, \dots$

2. Compare two succesive runs We can compute

$$\begin{aligned} e(h_{k+1}) &\approx Ch_{k+1}^p \\ e(h_k) &\approx Ch_k^p \end{aligned} \quad \Rightarrow \quad \frac{e(h_{k+1})}{e(h_k)} \approx \left(\frac{h_{k+1}}{h_k}\right)^p \quad \Rightarrow \quad p \approx \frac{\log\left(e(h_{k+1})/e(h_k)\right)}{\log\left(h_{k+1}/h_k\right)}$$

and maybe test this over several k.

We call this Experimental order of convergence (EOC) at refinement level k

$$\mathrm{EOC}(k) \approx \frac{\log\left(e(h_{k+1})/e(h_k)\right)}{\log\left(h_{k+1}/h_k\right)}$$

Error or convergence plot

Alternatively: Since $e(h) \approx Ch^p$ we can obtain p
using the data $(h_k, e(h_k))$ in a plot,
where both axes are logarithmic (log-log-plot) since then

$$y = \log e(h) \approx \log C + p \log h = a + px$$

And we see the p as the slope of this line.

Error estimate in practice I

Goal. Estimate when my error is small enough, so I do not have to increase m anymore.

In practice, an error estimate like we had for S_m, i. e.

$$\left| I(a,b) - S_m(a,b) \right| \leq \frac{(b-a)h^4}{180} f^{(4)}(\xi).$$

is complicated, since we do not know ξ.

We could take the maximum of $f^{(4)}(x)$ on (a,b), i. e. use a bound like

$$\left| I(a,b) - S_m(a,b) \right| \leq \frac{(b-a)h^4}{180} \|f^{(4)}\|_\infty.$$

as an upper bound, but this is usually a large over-estimation.

Question. How can we find an estimate of the error without any extra analytical calculations?

Error estimate in practice II

Idea. Let $[a, b]$ is chosen so small, that $f^{(4)}(x)$ can be assumed to be constant and set $C = -f^{(4)}(x)/2880$ (constant $\Rightarrow$ any $x \in (a, b)$).
We further set

- $H = b - a$
- $S_1(a, b)$ for ($m = 1$, classical) Simpson's rule
- $S_2(a, b)$ for the composite (2 intervals) Simpson's rule

Then errors of the two approximations are then given by

$$I(a, b) - S_1(a, b) \approx CH^5 \qquad \text{and} \qquad I(a, b) - S_2(a, b) \approx 2C\left(\frac{H}{2}\right)^5 = \frac{2CH^5}{32}.$$

Their difference is

$$S_2(a, b) - S_1(a, b) \approx \frac{15}{16}CH^5 \qquad \Rightarrow \qquad CH^5 \approx \frac{16}{15}(S_2(a, b) - S_1(a, b)).$$

We obtain an expression for CH^5.

Error estimate in practice III

Plugging the term for CH^5 from

$$S_2(a,b) - S_1(a,b) \approx \frac{15}{16} CH^5 \qquad \Rightarrow \qquad CH^5 \approx \frac{16}{15}(S_2(a,b) - S_1(a,b)).$$

into the errors, we obtain

$$E_1(a,b) = I(a,b) - S_1(a,b) \approx \frac{16}{15}\big(S_2(a,b) - S_1(a,b) \big) = \mathcal{E}_1(a,b),$$
$$E_2(a,b) = I(a,b) - S_2(a,b) \approx \frac{1}{15}\big(S_2(a,b) - S_1(a,b) \big) = \mathcal{E}_2(a,b).$$

We obtain an error estimate for both $S_1(a,b)$ and $S_2(a,b)$, since we know that they are related by a factor $\frac{1}{16}$ already.

Even better. We get a third, even better approximation for free:

$$I(a,b) \approx S_2(a,b) + \mathcal{E}_2(a,b) = \frac{16}{15} S_2(a,b) - \frac{1}{15} S_1(a,b)$$

Example.

Find an approximation to the integral $\int_0^1 \cos(x)dx = \sin(1)$ by composite Simpson's rules S_m, $m = 1, 2$ over one and two subintervals.
Find the error estimates $\mathcal{E}_m$, $m = 1, 2$ and compare with the exact errors $E_m, m = 1, 2$.

Exercise. As a homework do the same for Runge's function (prepared in the notebook already) for the intervals $[0, 8]$, $[0, 1]$, $[4, 8]$, $[0, 0.1]$. What you should observe

1. on $[0, 8]$: The error is large, and the error estimate is significantly smaller than the real error (the error is under-estimated).
2. on $[0, 1]$: As for the interval $[0, 8]$.
3. on $[4, 8]$: Small error, and a reasonable error estimate.
4. on $[0, 0.1]$: Similar to $[0, 8]$ but the approximate error is worse

Why is this so and how can we deal with this?
It seems we need smaller intervals near $x = 0$.

Adaptive Integration

Idea. Instead of equispaced points, use a basic function, for example `SimpsonBasic`, that returns a quadrature $Q(a,b)$ and an error estimate $\mathcal{E}(a,b)$ to partition the interval

$$a = X_0 < X_1 < \cdots < X_m = b$$

such that (automatically) for any $k = 0, \ldots, m-1$ we have

$$|\mathcal{E}(X_k, X_{k+1})| \leq \frac{X_{k+1} - X_k}{b-a}\mathrm{Tol}$$

where Tol is a given tolerance (by the user).

This way the accumulated error is

$$\mathcal{E}(a,b) \approx \sum_{j=0}^{m-1} \mathcal{E}(X_k, X_{k+1}) \leq \mathrm{Tol}.$$

Algorithm. Adaptive quadrature

Given f, a, b and a user defined tolerance Tol.

1. Calculate $Q(a,b)$ and $\mathcal{E}(a,b)$.

2. If $|\mathcal{E}(a,b)| \leq \text{Tol}$

- Accept the result, return $Q(a,b) + \mathcal{E}(a,b)$ as an approximation to $I(a,b)$.

else

- set $c = (a+b)/2$, and repeat the process on each of the subintervals $[a,c]$ and $[c,b]$, with tolerance $\text{Tol}/2$.

3. Sum up the accepted results from each subinterval.

Newton-Côtes Formulae

Newton-Côtes Formulae are quadratures based on polynomial interpolation with the equispaced nodes in $[a, b]$, i.e.

$$x_k = x_0 + kh, \qquad k = 0, \ldots, n.$$

A Newton-Côtes formula is called

- closed if $x_0 = a$ and $x_n = b$ and we obtain $h = \frac{b-a}{n}$, $n \geq 1$
- open if $x_0 = a + h$ and $x_n = b - h$ and we have $h = \frac{b-a}{n+2}$ $n \geq 0$

Known. Midpoint (open, $n = 0$), Trapezoidal (closed, $n = 1$), and Simpson (closed, $n = 2$).

Properties of Newton-Côtes Formulae

For a closed Newton-Côtes formula with $x_0, \dots, x_n$ as nodes

- degree of exactness is n
- for even n (e. g. Simpson): degree of exactness $n+1$

Obs for $n \geq 8$: negative weights w_i occur.

$\Rightarrow$ There exist $f \geq 0$ everywhere such that these rules ($n \geq 8$) yield $Q[f] < 0$. They are also not numerically stable

- since $n = 6$ and $n = 7$ have same degree of exactness

$\Rightarrow$ the rules with $n \leq 6$ are used in practice.

For open Newton-Côtes formulae negative weights appear for $n \geq 2$, so only the mid point rule is commonly used.

Gauß-Quadrature

Remember. The Gauß-Legendre quadrature with $n = 1$ (2 points) had degree of exactness $3 = 2n + 1$!

A **Gauß quadrature** uses orthogonal polynomials $p_0, \ldots, p_n$, $p_j \in \mathbb{P}_j$, (constructed with Gram-Schmidt) such that all roots are in $[a, b]$ and uses these roots as nodes.
A Gauß quadrature with $n + 1$ nodes $x_0, \ldots, x_n$ has degree of exactness $2n + 1$. This is the best you can get.

Example. Gauss-Legendre quadrature. For the standard interval $[-1, 1]$ choose the nodes as the zeros of the polynomial of degree n:

$$L_n(t) = \frac{\mathrm{d}^n}{\mathrm{d}t^n}(t^2 - 1)^n.$$

and orthogonality means

$$\int_{-1}^{1} L_i(t) L_j(t) \,\mathrm{d}t = 0 \quad \text{if } i \neq j$$

Special case: $n = 1$: mid point rule.