Lecture 5
Numerical Integration

1 Numerical Integration

In this lecture we will discuss how to evaluate numerically a definite integral:

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

Approximating a definite integral using a numerical method is called **quadrature**.

The definition of the definite integral as a limit of Riemann sums suggests how to perform quadrature:

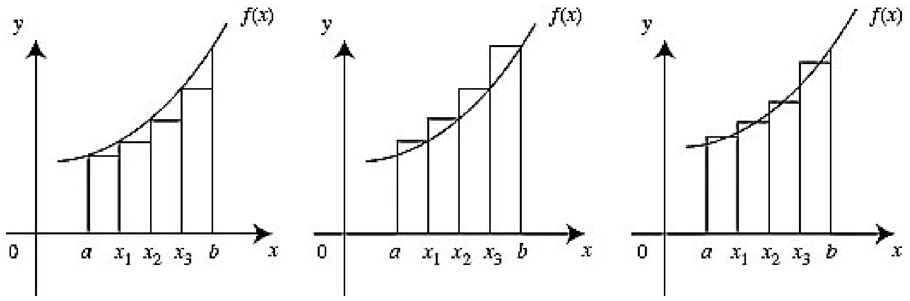

Figure 1: Left-end point, right-end point and midpoint Riemann sums

Fundamental Theorem of Calculus, also known as Newton-Leibniz formula, tells us that if $F(x)$ is an antiderivative of $f(x)$, then

$$I = \int_a^b f(x)\, dx = F(x)\Big|_a^b = F(b) - F(a).$$

In this manner, we can evaluate many integrals analytically, for ex.

$$\int_{-1}^{1} (e^x + x^3)\, dx \qquad \text{or} \qquad \int_0^{\pi} x \sin(x/2)\, dx.$$

However, there are functions whose antiderivatives can not be expressed in terms of elementary functions, for example

$$\int_0^1 e^{-x^2}\, dx.$$

And even if antiderivative can be computed, quadrature may be much simpler and easier to use.

For example, consider evaluating integral

$$\int_1^{2000} \exp(\sin(\cos(\sinh(\cosh(\tan^{-1}(\log(x))))))) \, dx$$

or even a simpler integral

$$\int_0^1 \frac{1}{1+x^{10}}\, dx$$

have an extremely complicated antiderivative and it turns out that it is easier to evaluate these integrals by quadrature. A simple code in Python (using quad function) can do the necessary evaluations much faster.

```
>>> import scipy.integrate as spi
>>> from math import *
>>> def f(x):
...       return exp(sin(cos(sinh(cosh(atan(log(x)))))))
>>> spi.quad(f,1,2000)
(1514.7806778270258, 4.231109728875231e-06)
```

Similar simple code in GNU Octave (open source, similar with MATLAB)

```
octave:1> function y=f(x)
   >> y=exp(sin(cos(sinh(cosh(atan(log(x)))))));
   >> endfunction
octave:2> [q, ier, nfun, err] = quad ("f", 1, 2000)
  q = 1514.780677827026
  ier = 0
  nfun = 357
  err = 4.231109731546272e-06
```

Quadrature also generalizes naturally to higher dimensions, and allows us to compute integrals on irregular domains in 2D or 3D. For example, we can approximate a double integral on a triangle

$$\iint_T f(x,y)\,dA = \sum_i \omega_i f(x_i,y_i)$$

based on a finite weighted sum of samples at quadrature points (Gaussian quadrature) $\{(x_i,y_i)\}$ as depicted in figure 2.

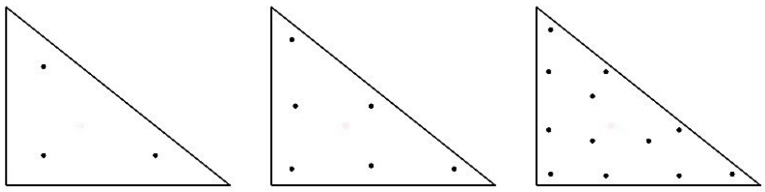

Figure 2: 3, 6 and 12 quadrature points (x_i,y_i)

In Finite Element Method (FEM) applied to engineering problems we might need to compute a large number of double integrals on triangles from a given mesh (triangulation), see figure 3.

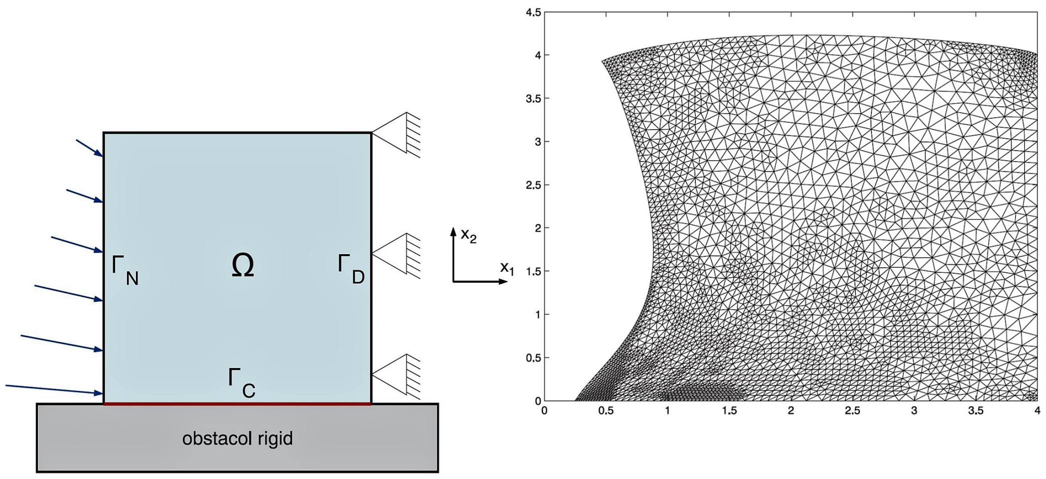

Figure 3: Contact mechanics problem (left) and its finite element solution (right)

Recall the principle from root-finding discussed earlier:

Numerical Analysis Principle

If you cannot solve the given problem, then solve a ”nearby” problem, whose solution will be close enough to the solution of initial problem.

Recall that our problem is to evaluate

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

In many quadrature, $f(x)$ is substituted by some simpler approximation $f_A(x)$, for which integral is evaluated easily. Thus, instead of computing the original integral we evaluate

$$I \approx I_A \equiv \int_a^b f_A(x)\,dx.$$

QUESTION: How "good" is this approximation?

Let's take a look at the error:

$$\text{Error} = I - I_A = \int_a^b \Big(f(x) - f_A(x)\Big)\,dx.$$

and estimate it:

$$\begin{aligned} |\text{Error}| &\leqslant \int_a^b \Big|f(x) - f_A(x)\Big|\,dx \\ &\leqslant (b-a)\max_{x\in[a,b]} \big|f(x) - f_A(x)\big| \\ &= (b-a)\,\|f - f_A\|_\infty. \end{aligned}$$

Therefore, we have the following error estimate:

$$|\text{Error}| \leqslant (b-a)\,\|f - f_A\|_\infty.$$

We can see from the last estimate that the error in numerical integration depends on how "well" function f_A approximates original function f. As mentioned, we want the approximates f_A to be directly and easily integrable. Examples of such functions are polynomials, trigonometric functions, piece-wise polynomials, etc.

For polynomial approximations, two choices can be considered:

– Taylor polynomials approximating function$f$;

– Interpolation polynomials approximating function f.

We will focus on the second approach since it is generally more difficult to do numerical integration by constructing Taylor polynomial approximations than by constructing interpolating polynomials. Quadrature, (or numerical integration schemes) based on interpolation polynomials are called **Newton-Cotes quadrature/schemes/formulas/rules**.

1.1 Trapezoidal Rule

The approximation f_A to be used in quadrature is the interpolation polynomial P_n at (initially) evenly spaced points:

$$\int_a^b f(x)dx \approx \int_a^b P_n(x)dx.$$

Consider linear interpolant to function f, interpolating at a and b:

$$P_1(x) = f(a)\frac{b-x}{b-a} + f(b)\frac{x-a}{b-a}$$

to obtain

$$\begin{aligned} \int_a^b f(x)dx &\approx \int_a^b P_1(x)dx \\ &= \boxed{\frac{b-a}{2}\big(f(a)+f(b)\big) \equiv T_1(f).} \end{aligned}$$

The quadrature $\int_a^b f(x)dx \approx T_1(f)$ is called trapezoidal rule .

The error for trapezoidal rule

$$\big|I - T_1(f)\big|$$

is the area of the region between the graph of $f(x)$ (red) and trapezoid (yellow), see figure 4.

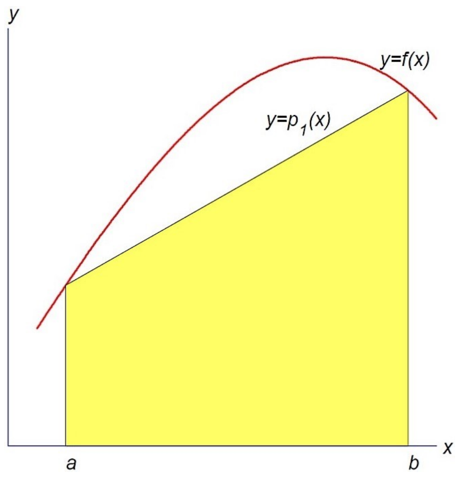

Figure 4: Trapezoidal rule. $I \approx T_1(f)$

Example 1 *Consider applying trapezoidal rule* T_1 *to approximate the integral of function* $\sin x$ *on interval* $[0, \pi/2]$, *whose true value is* 1.

$$\begin{aligned}\int_0^{\pi/2} \sin x\, dx &\approx \frac{\pi}{4}\left(\sin 0 + \sin\frac{\pi}{2}\right)\\ &= \frac{\pi}{4} \approx 0.785398.\\ Error &= 0.215\end{aligned}$$

Observe that in this case, trapezoidal rule exhibits quite a large error of 0.215.

Next question is how to get a greater accuracy? Again, there are two approaches:

1. Increase the degree of the approximation, e.g. consider the quadratic interpolation, or even a higher degree interpolation;
2. Use the piece-wise linear interpolation.

1.2 Simpson's Rule

Consider the first approach. We will get the so-called **Simpson's rule**. To this end, approximate $\int_a^b f(x)\, dx$ using a quadratic interpolant $P_2(x)$ on points $a, c = \frac{a+b}{2}, b$:

$$P_2(x) = \frac{(x-c)(x-b)}{2h^2} f(a) + \frac{(x-a)(x-b)}{-h^2} f(c) + \frac{(x-a)(x-c)}{2h^2} f(b),$$

where $h = \frac{b-a}{2}$. Replace $f(x)$ with $P_2(x)$ to get the approximation

$$\begin{aligned}\int_a^b f(x)\, dx &\approx \int_a^b P_2(x)\, dx\\ &= \boxed{\tfrac{h}{3}\Big(f(a) + 4f(c) + f(b)\Big) \equiv S_2(f).}\end{aligned}$$

This is called Simpson's rule.
Notice that error for trapezoidal rule $S_2(f)$ (area between graph and yellow region in figure 5)

$$|I - S_2(f)|$$

is smaller than for trapezoidal rule.

Example 2 *consider approximating the integral form example 1 using Simpson's rule.*

$$\begin{aligned}\int_0^{\pi} \sin x\, dx &\approx \tfrac{\pi/2}{3}\left(\sin 0 + 4\sin\tfrac{\pi}{4} + \sin\tfrac{\pi}{2}\right)\\ &\approx 1.00227987749221\\ Error &= -0.00228\end{aligned}$$

Indeed the error for Simpson's rule S_2 *is much smaller than the error for trapezoidal rule* T_1.

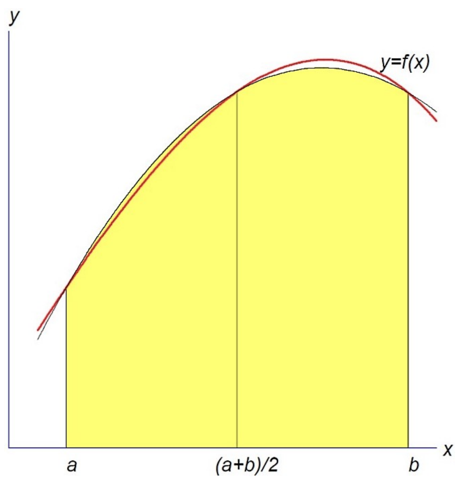

Figure 5: Simpson's rule. $I \approx S_2(f)$

1.3 Other integration rules

As another approximation to

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

replace $f(x)$ by the degree four interpolation polynomial $P_4(x)$ at five evenly spaced points

$$x_j = a + j \cdot h, \quad j = 0, 1, 2, 3, 4, \quad h = \tfrac{b-a}{4}.$$

It can be shown (similar to what was done for trapezoidal and Simpson's rule cases) this leads to the approximation formula, known as Boole's rule :

$$\boxed{B_4(f) = \frac{2h}{45}\Big(7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\Big).}$$

How about developing other integration Newton-Cotes rules by considering higher degree interpolating polynomials at equally spaced points? Newton–Cotes of order n formulas are based on polynomial interpolation at equally spaced $n+1$ points. Therefore, they're susceptible to Runge's phenomenon, and thus we expect them to be inaccurate for large values n.

1.4 Composite Trapezoidal Rule

Considering higher degree interpolation on evenly spaced points for development of new integration schemes for large $n \geqslant 5$ is not a useful idea. Instead, in order to get quadrature rules that converge as $n \to \infty$ we can use piece-wise polynomial interpolation as suggested previously and obtain the so-called *Composite Trapezoidal* and *Composite Simpson's* rules. Let's first consider the Composite Trapezoidal rule for approximating

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

and apply Trapezoidal rule T_1 to each half of interval $[a, b]$:

$$\begin{aligned} I &= \int_a^c f(x)\,dx + \int_c^b f(x)\,dx, \qquad c = \tfrac{a+b}{2} \\ &\approx \frac{c-a}{2}\Big(f(a) + f(c)\Big) + \frac{b-c}{2}\big(f(c) + f(b)\big) \\ &= \boxed{\frac{h}{2}\Big(f(a) + 2f(c) + f(b)\Big) \equiv T_2(f),} \end{aligned}$$

where $h = \frac{b-a}{2}$ is the partition step. This is Composite Trapezoidal rule $T_2(f)$.

Example 3 *Consider the same function as in examples 1 and 2 and apply rule T_2.*

$$\begin{aligned}\int_0^{\pi/2} \sin x\,dx &\approx \tfrac{\pi}{8}\Big(\sin 0 + 2\sin\tfrac{\pi}{4} + \sin\tfrac{\pi}{2}\Big)\\ &\approx 0.948059\\ Error &= 0.0509\end{aligned}$$

Note that the error in this case is smaller than the error in example 1.

This approach can be extended by considering 2 intermediate points x_1, x_2 to get $T_3(f)$ (as in figure 6), and so on.

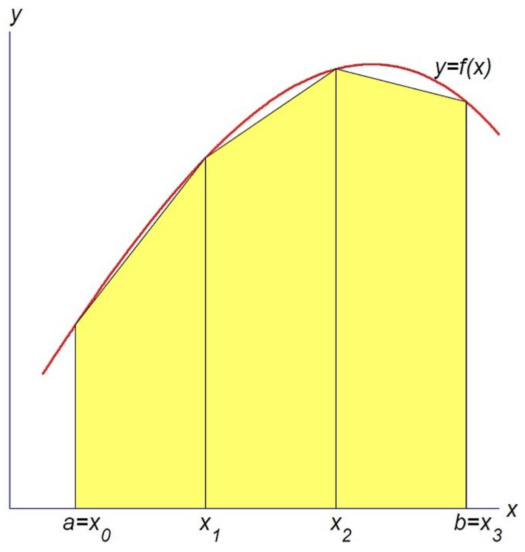

Figure 6: Trapezoidal rule. $I \approx T_3(f)$

Observe that error for trapezoidal rule $T_3(f)$ (area between graph and yellow region in figure 6)

$$\big|I - T_3(f)\big|$$

is obviously smaller than that for $T_1(f)$ or $T_2(f)$.

Let's generalize Composite Trapezoidal rule. Consider an evenly spaced partition of interval $[a, b]$:

$$\begin{aligned}&a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,\\ &x_j = a + j\cdot h,\ j = 0, 1, \ldots, n, \quad h = \tfrac{b-a}{n}\end{aligned}$$

and apply $T_1(f)$ formula $\int_\alpha^\beta f(x)dx \approx \frac{\beta-\alpha}{2}\big(f(\alpha) + f(\beta)\big)$ for each sub-interval $[x_j, x_{j+1}]$ of length h:

$$\begin{aligned}I &= \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)\,dx\\ &= \tfrac{h}{2}\big(f(x_0) + f(x_1)\big) + \tfrac{h}{2}\big(f(x_1) + f(x_2)\big) + \cdots + \tfrac{h}{2}\big(f(x_{n-1}) + f(x_n)\big)\\ &= \boxed{\tfrac{h}{2}\Big(f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)\Big) \equiv T_n(f).}\end{aligned}$$

This is called the Composite Trapezoidal rule.

Example 4 *Consider applying Composite Trapezoidal rule to approximate integral*

$$\int_0^{\pi/2} \sin x\,dx = 1.$$

Numerical results are presented in table 1. Since we know the true value of considered integral we can evaluate the errors E_n in approximation and also we can compute the ratio R_n

$$R_n = \frac{E_{n/2}}{E_n}$$

i.e. the factors by which the error decreases if we double the number of points. Note that the erros will decrease by a factor of 4 then we double the number of points (when we halve h).

n	$T_n(f)$	Error E_n	Ratio R_n
1	0.785398163	2.15E-1	
2	0.948059449	5.19E-2	4.13
4	0.987115801	1.29E-2	4.03
8	0.996785172	3.21E-3	4.01
16	0.999196680	8.03E-4	4.00
32	0.999799194	2.01E-4	4.00
64	0.999949800	5.02E-5	4.00
128	0.999987450	1.26E-5	4.00
256	0.999996863	3.14E-6	4.00

Table 1: Composite Trapezoidal rule for example 4

1.5 Composite Simpson's rule

As with Composite Trapezoidal rule, in a similar manner let's apply the Simpson's rule on smaller subintervals in order to get a better accuracy in approximation. Again consider an evenly spaced partition of $[a, b]$ with $h = \frac{b-a}{n}$ and n even integer.

$$\begin{aligned} I &= \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)\,dx \\ &= \tfrac{h}{3}\Big(f(x_0) + 4f(x_1) + f(x_2)\Big) + \tfrac{h}{3}\Big(f(x_2) + 4f(x_3) + f(x_4)\Big) + \cdots + \tfrac{h}{3}\Big(f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\Big) \\ &= \boxed{\tfrac{h}{3}\Big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\Big) \equiv S_n(f).} \end{aligned}$$

This is called the Composite Simpson's rule.

Example 5 *Apply Composite Simpson's rule to the integral from previous example. Results are presented in table 2. Observe that ratio's R_n decrease in this case by a factor of 16 if we double the number of points.*

n	$S_n(f)$	Error E_n	Ratio R_n
2	1.00227987749221	–2.28E-3	
4	1.00013458497419	–1.35E-4	16.94
8	1.00000829552397	–8.30E-6	16.22
16	1.00000051668471	–5.17E-7	16.06
32	1.00000003226500	–3.23E-8	16.01
64	1.00000000201613	–2.02E-9	16.00
128	1.00000000012600	–1.26E-10	16.00
256	1.00000000000788	–7.88E-12	16.00
512	1.00000000000049	–4.92E-13	15.99

Table 2: Composite Simpson's rule for example 5

Example 6 *Now, let's consider three more integrals*

$$I^{(1)} = \int_0^1 e^{-x^2}\,dx \approx 0.746824132812427,$$
$$I^{(2)} = \int_0^4 \frac{1}{1+x^2}\,dx \approx 1.32581766366803,$$
$$I^{(3)} = \int_0^{2\pi} \frac{1}{2+\cos x} = \frac{2\pi}{3} \approx 3.62759872846844.$$

and evaluate them numerically using Composite Trapezoidal and Composite Simpson's rules.

Numerical results for Composite Trapezoidal rule applied to all three integrals are presented in table 3.

	$I^{(1)}$		$I^{(2)}$		$I^{(3)}$	
n	Error	Ratio	Error	Ratio	Error	Ratio
2	1.6E-2		–1.3E-1		–5.6E-1	
4	3.8E-3	4.02	–3.6E-3	37.0	–3.8E-2	14.09
8	9.6E-4	4.01	5.6E-4	-6.04	–1.9E-4	195
16	2.4E-4	4.00	1.4E-4	3.9	–5.2E-9	37600
32	6.0E-5	4.00	3.6E-5	4.00		
64	1.5E-5	4.00	9.0E-6	4.00		
128	3.7E-6	4.00	2.3E-6	4.00		

Table 3: Composite Trapezoidal rule for example 6

Note that in integrals $I^{(1)}$ and $I^{(2)}$ the ratio converges to 4.00, while in the third integral the ratio is growing up fast, since errors decrease dramatically.

Numerical results for Composite Simpson's rule applied to all three integrals are presented in table 4.

	$I^{(1)}$		$I^{(2)}$		$I^{(3)}$	
n	Error	Ratio	Error	Ratio	Error	Ratio
2	–3.6E-4		8.7E-2		–1.3E-0	
4	–3.1E-5	11.4	3.9E-3	2.2	1.4E-1	–9.02
8	–2.0E-6	15.7	2.0E-3	20	1.2E-2	11.2
16	–1.3E-7	15.9	4.0E-6	485	6.4E-5	191
32	–7.8E-9	16.00	2.3E-8	172	1.7E-9	37600
64	–4.9E-10	16.00	1.5E-9	16.00		
128	–3.0E-11	16.00	9.2E-11	16.00		

Table 4: Composite Simpson's rule for example 6

Note that again in integrals $I^{(1)}$ and $I^{(2)}$ the ratio converges to 16.00, while in the third integral the ratio is growing up fast, since errors decrease dramatically.

1.6 Error formula for Composite Trapezoidal rule

The error in applying the Composite Trapezoidal rule is given by the following theorem:

Theorem 1 *Let f be continuously differentiable function on interval $[a,b]$ ($f \in C^2[a,b]$). Then*

$$E_n^T(f) = \int_a^b f(x)\,dx - T_n(f) = -\frac{h^2(b-a)}{12} f''(\xi)$$

for some (generally unknown) $\xi \in [a,b]$.

This error formula tells us that the error decreases proportionally to h^2:

$$E_n^T(f) \approx C_1 h^2 = \frac{C_2}{n^2},$$

where $h = \frac{b-a}{n}$. It follows from the above approximation that if we double n (and thus if we halve h), then the error will decrease by a factor of 4. Exactly what we've observed in example 4.

Example 7 *Apply Composite Trapezoidal rule $T_n(f)$ to approximate the integral:*

$$\int_0^2 \frac{1}{1+x^2}\,dx.$$

How large should be n in order to ensure that the error $E_n^T(f) \leqslant 5 \cdot 10^{-6}$?

Solution. Compute 2nd order derivative of f and its maximum on $[0,2]$

$$f''(x) = \frac{-2+6x^2}{(1+x^2)^3} \quad \text{with} \max_{x\in[0,2]} |f''(x)| = 2$$

and apply error formula with $a = 0$, $b = 2$ and given function f. Since we don't know ξ, we bound $f''(\xi)$ by its maximum on $[0,2]$.

$$E_n^T(f) = -\frac{h^2(b-a)}{12} f''(\xi),$$
$$\left|E_n^T(f)\right| = \frac{h^2 \cdot 2}{12} \cdot |f''(\xi)| \leqslant \frac{h^2}{6} \cdot 2 = \frac{h^2}{3}.$$

We have the estimate of the error:

$$\left|E_n^T(f)\right| \leqslant \frac{h^2}{3}.$$

We want

$$\left|E_n^T(f)\right| \leqslant 5 \cdot 10^{-6}.$$

In order to ensure this, we need

$$\left|E_n^T(f)\right| \leqslant \frac{h^2}{3} \leqslant 5 \cdot 10^{-6},$$

which is equivalent to

$$h \leqslant \sqrt{15 \cdot 10^{-6}} \approx 0.003873$$

or

$$n = \frac{2}{h} \geqslant \frac{2}{0.003873} = 516.4.$$

So, if $n \geqslant 517$, we will have the error in trapezoidal rule smaller than $5 \cdot 10^{-6}$.

1.7 Periodic integrands

Recall integral $I^{(3)}$ from example 6, for which a faster rate of convergence was observed in both trapezoidal and Simpson's rules. Why?

Definition 1 *A function $f(x)$ is called **periodic** with period T if there exists a smallest real number $T > 0$ such that*

$$f(x+T) = f(x)$$

for all $x \in \mathbb{R}$.

Basically, a periodic function is one which repeats itself over intervals of length T. For example, trigonometric functions $\sin x$ and $\cos x$ are periodic with period 2π, whilst $\tan x$ and $\cot x$ are periodic functions with period π.

It can be shown that, if function f is periodic with $b-a$ an integer multiple of the period T, and if f is also infinitely differentiable, then the error $I - T_n$ decreases to zero more rapidly than n^2. Also, it can be shown that

For periodic integrands, the trapezoidal rule is an optimal numerical integration method.

1.8 Error formula for Composite Simpson's rule

The error in applying the Composite Simpson's rule is given by the following theorem:

Theorem 2 *Let $f \in C^4[a,b]$ (four times continuously differentiable on $[a,b]$). Then the error in Simpson's rule is given by*

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

for some $\xi \in [a,b]$.

This error formula tells us that the error decreases proportionally to h^4:

$$E_n^S(f) \approx C_1 h^4 = \frac{C_2}{n^4}.$$

Thus, if we double n (and thus we halve h), then the error will decrease by a factor of 16. Exactly what we've observed in example 5.

Example 8 *Apply Composite Simpson's rule $S_n(f)$ to approximate integral:*

$$\int_0^2 \frac{1}{1+x^2}\,dx.$$

How large should be n in order to ensure that $E_n^S(f) \leqslant 5 \cdot 10^{-6}$?

Solution. Similar to example 7, need to compute the 4th order derivative of function f and its maximum

$$f^{(4)}(x) = 24\frac{5x^4 - 10x^2 + 1}{(1+x^2)^5} \quad \text{with} \quad \max_{x\in[0,2]} f^{(4)}(x) = 24$$

and apply the error formula from Theorem 2:

$$\begin{aligned} \left|E_n^S(f)\right| &\leqslant \frac{h^4 \cdot 2}{180} \cdot 24 = \frac{4h^4}{15} \\ \frac{4h^4}{15} &\leqslant 5 \cdot 10^{-6} \\ h &\leqslant \sqrt[4]{\frac{75 \cdot 10^{-6}}{4}} \approx 0.0658 \\ n &\geqslant 30.39 \end{aligned}$$

Note that in Simspon's rule in order to reach the same level of accuracy we will need only $n = 32$ intermediate points compared to more than 517 in trapezoidal.

1.9 Integrating non-smooth functions

Consider applying Composite Trapezoidal and Composite Simpson's rules for approximating integral

$$\int_0^1 \sqrt{x}\,dx.$$

Numerical results are presented in table 5.

Observe that the ratio's of errors are smaller than predicted by theory (ratio 4 for Trapezoidal rule and ratio 16 for Simpson's rule, respectively), and therefore the rate of convergence is slower.

It happens because function $f(x) = \sqrt{x}$ is not sufficiently many times differentiable on $[0,1]$. Recall that according to Theorem 1 function f should be in class C^2 for the error formula in trapezoidal rule to be valid, and it should be in class C^4 for the corresponding error formula for Simpson's rule from Theorem 2.

n	E_n^T	Ratio	E_n^S	Ratio
2	6.311E-2		2.860E-2	
4	2.338E-2	2.70	1.012E-2	2.82
8	8.536E-3	2.74	3.587E-3	2.83
16	3.085E-3	2.77	1.268E-3	2.83
32	1.108E-3	2.78	4.485E-4	2.83
64	3.959E-4	2.80	1.586E-4	2.83
128	1.410E-4	2.81	5.606E-5	2.83

Table 5: Integration of a non-smooth function

Instead, since derivative of function f

$$f'(x) = \left(\sqrt{x}\right)' = \frac{1}{2\sqrt{x}}$$

is not continuous on $[0,1]$ (derivative at 0 is discontinuous), both methods converge with a rate only proportional to $h^{1.5}$.

To summarize, we have the following Composite Trapezoidal and Composite Simpson's error formulas

$$\begin{aligned}
E_n^T(f) &= -\frac{h^2(b-a)}{12} f''(\theta) \approx C_1 h^2 = \frac{C_2}{n^2}, && \left[f \in C^2[a,b]\right]\\
E_n^S(f) &= -\frac{h^4(b-a)}{180} f^{(4)}(\theta) \approx C_3 h^4 = \frac{C_4}{n^4}, && \left[f \in C^4[a,b]\right]
\end{aligned}$$

where $h = \frac{b-a}{n}$ and C_i are some constants.

Note that both these formulas have the form

$$E_n(f) \approx \frac{C}{n^p}$$

for appropriate constant C and exponent p. These formulas occur for many quadrature rules, and p can be less than predicted, if function f is not sufficiently many times differentiable.

1.10 Richardson's extrapolation formula

Suppose that for some general integration scheme (rule) I_n we have

$$I - I_n \approx \frac{c}{n^p}.$$

Assume initially that we know exponent p. Then, by substituting n with $2n$ we get

$$I - I_{2n} \approx \frac{c}{(2n)^p} = \frac{c}{n^p} \cdot \frac{1}{2^p} = \left(I - I_n\right)\frac{1}{2^p}.$$

Solve the last relation for I to obtain

$$\begin{aligned}
I - I_n &\approx 2^p\left(I - I_{2n}\right),\\
2^p \cdot I - I &\approx 2^p I_{2n} - I_n,\\
\left(2^p - 1\right) I &\approx 2^p I_{2n} - I_n,\\
I &\approx \frac{1}{2^p - 1}\left(2^p I_{2n} - I_n\right),\\
I &\approx I_{2n} + \frac{I_{2n} - I_n}{2^p - 1}.
\end{aligned}$$

The last formula is called **Richardson's extrapolation formula**. It allows us to get an even better estimate for the integral I once the approximations I_n and I_{2n} are computed.

Example 9 *With the Composite Trapezoidal rule and with the integrand f having two continuous derivatives, Richardson's extrapolation formula becomes ($p = 2$):*

$$I \approx T_{2n} + \frac{1}{3}\left(T_{2n} - T_n\right).$$

With Composite Simpson's rule and with the integrand f having four continuous derivatives, Richardson's extrapolation formula is ($p = 4$):

$$I \approx S_{2n} + \frac{1}{15}\left(S_{2n} - S_n\right).$$

Also, we can use Richardson's extrapolation formula to estimate the error in numerical integration:

$$\boxed{I - I_{2n} \approx \frac{I_{2n} - I_n}{2^p - 1},}$$

which is called **Richardson's error estimate**. Note that in order to use the above formulas, p should be known.

1.11 Aitken's extrapolation

Suppose again that

$$I - I_n \approx \frac{c}{n^p},$$

but this time we don't know neither exponent p nor constant$C$.

Assume that we have computed I_n, I_{2n} and I_{4n}. Then,

$$\begin{aligned} I - I_n &\approx \frac{c}{n^p}, \\ I - I_{2n} &\approx \frac{c}{2^p n^p}, \\ I - I_{4n} &\approx \frac{c}{4^p n^p}. \end{aligned}$$

Let's estimate I directly. By division we get:

$$\frac{I - I_n}{I - I_{2n}} \approx 2^p \approx \frac{I - I_{2n}}{I - I_{4n}}.$$

Solve for I to obtain

$$\left(I - I_{2n}\right)^2 \approx \left(I - I_n\right), \left(I - I_{4n}\right)$$

$$I \approx \frac{I_n I_{4n} - I_{2n}^2}{I_n + I_{4n} - 2I_{2n}}.$$

Thus, from last formula, after rewriting it in order to avoid the loss of significance error, we will obtain the so called **Aitken extrapolation formula**:

$$\boxed{I \approx I_{4n} - \frac{\left(I_{4n} - I_{2n}\right)^2}{\left(I_{4n} - I_{2n}\right) - \left(I_{2n} - I_n\right)}.}$$

To estimate exponent p, we may use

$$2^p \approx \frac{I_{2n} - I_n}{I_{4n} - I_{2n}} = R_n,$$

in other words

$$p = \log_2 R_n = \frac{\ln R_n}{\ln 2}.$$

Example 10 *Consider the following table of numerical results, obtained after applying some integration rule.*
What is the order of convergence? How we can estimate the error $I - I_{64}$?

n	I_n	$I_n - I_{\frac{1}{2}n}$	R_n
2	0.28451779686		
4	0.28559254576	1.075e-3	
8	0.28570248748	1.099e-4	9.78
16	0.28571317731	1.069e-5	10.28
32	0.28571418363	1.006e-6	10.62
64	0..28571427643	9.289e-8	10.84

Table 6: Numerical results for example 10

Solution. From the table we get

$$\begin{aligned} 2^p &\approx 10.84, \\ p &\approx \log_2 10.84 \approx 3.44. \end{aligned}$$

Combine this with Richardson's error estimate to get

$$\begin{aligned} I - I_{64} &\approx \frac{1}{10.84 - 1}\left(I_{64} - I_{32}\right) \\ &\approx \frac{1}{9.84} \cdot 9.289 \cdot 10^{-8} = 9.43 \cdot 10^{-9}. \end{aligned}$$

1.12 Adaptive quadrature

Another approach that will enhance the convergence of quadrature consists in not using evenly spaced partitions. Composite quadrature rules are very flexible in the sense that we need not choose evenly spaced partition. Intuitively, we should use smaller intervals where function f varies rapidly, and larger intervals where f varies slowly. As an example see figure 7, where obviously the neighborhood of the interval $[0.1, 0.3]$ needs a denser partition.

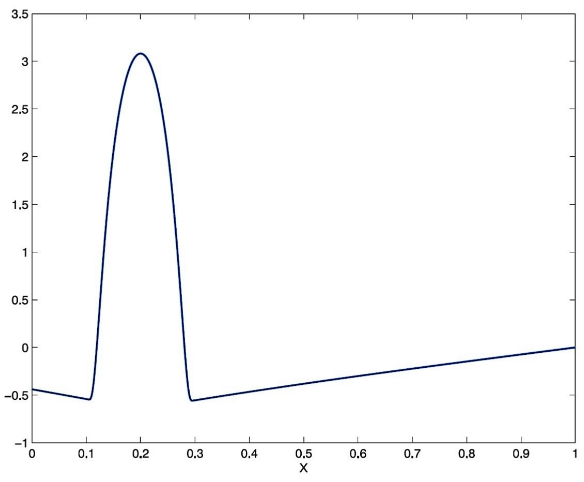

Figure 7: Function f

This can be achieved by adaptive quadrature:

1. Initialize to $m = 1$ (one interval);
2. On each interval, evaluate quadrature rule and estimate quadrature error;
3. If error estimate $> TOL$ on interval i, subdivide to get two smaller intervals and return to step 2 (where TOL is a specified error tolerance).

As an error estimator, we can use Richardson's error estimate.

Adaptive quadrature algorithm leads to a bigger density of integration points in the area of faster variations of function f and to a sparser distribution in the region where f varies slowly.

Python, GNU OCTAVE and MATLAB have quad functions, although with different implementations. MATLAB's quad function implements an adaptive Simpson rule:

```
>> help quad
QUAD Numerically evaluate integral, adaptive Simpson
quadrature. Q = QUAD(FUN,A,B) tries to approximate the
integral of scalar-valued function FUN from A to B to
within an error of 1.e-6 using recursive adaptive Simpson
quadrature.
```

Python and GNU OCTAVE quad functions both use Clenshaw-Curtis method based on Chebyshev nodes.

1.13 Degree of precision for quadrature rules

Another convenient way to compare accuracy of quadrature rules is to compare the polynomial degree they integrate exactly. Exact integration means that numerical evaluation by quadrature rule is the same as the exact value of definite integral, i.e. the error is zero.

Definition 2 *If the quadrature rule has zero error when integrating any polynomial of degree less or equal to r and if the error is nonzero for some polynomial of degree $r+1$, then we say that the rule has a* ***degree of precision*** *equal to r.*

Newton–Cotes quadrature of order n is based on polynomial interpolation, hence in general integrates polynomials of degree n exactly. Therefore, trapezoidal rule has degree of precision at least 1, while the Simpson's rule has degree of precision at least 2. In other words,

$$I = \int_a^b f(x)dx = T_1(f)$$

for any function f that is a polynomial of degree 1, i.e. it is linear. Similarly,

$$I = \int_a^b g(x)dx = S_2(g)$$

for any function g that is a quadratic polynomial.

It can be shown that the following quadrature rules have the corresponding degree of precision:

$$\begin{array}{rcl} M_1(f) & : & r=1 \\ T_1(f) & : & r=1 \\ S_2(f) & : & r=3 \\ B_4(f) & : & r=5 \end{array}$$

Here, $M_1(f)$ denotes the midpoint rule (see homework 4) and $B_4(f)$ is the Boole's rule introduced earlier. Let's show that Simpson's rule $S_2(f)$ has degree of precision 3, i.e. it is exact for polynomials of degree 3, but fails to be exact for polynomials of degree > 3.

Since quadrature rules are linear, including Simpson's rule as well, it is enough to check the exactness for monomials 1, x, x^2 and x^3.

Furthermore, as it was mentioned above, since Simpson's rule is based on quadratic interpolation, and using the fact that interpolation polynomial is unique, it is sufficient to show the exactness of Simpson's rule only for monomial x^3. Need to show that

$$\int_a^b x^3\,dx = \tfrac{b^4-a^4}{4} \equiv S_2(x^3) = \tfrac{h}{3}\left(a^3+4c^3+b^3\right),$$

where $c = \frac{a+b}{2}$ and $h = \frac{b-a}{2}$. Indeed, substituting expressions for c and h in the right side expression, after some algebraic transformations we get the left side.

How about Composite Trapezoidal and Composite Simpson's rule? The following theorem can be proved:

Theorem 3 *If a Newton-Cotes quadrature rule has degree of precision r, then the corresponding (equidistant) composite rule has order of accuracy $r+1$ for integrands $f \in C^{r+1}[a,b]$.*

Example 11 *Trapezoidal rule has degree of precision 1. Therefore, by theorem 3, Composite Trapezoidal rule will have order of accuracy 2 for integrands $f \in C^2[a,b]$. Indeed, $E_n^T(f) \approx Ch^2$.*

Example 12 *Simpson's rule has degree of precision 3. Therefore, by theorem 3, Composite Simpson's rule will have order of accuracy 4 for integrands $f \in C^4[a,b]$. Indeed, $E_n^S(f) \approx Ch^4$.*

Remark 1
The integrand f must be smooth enough to get the stated order of accuracy: $f \in C^{r+1}$ is sufficient for a simple degree r rule. If f is less regular, and we only have $f \in C^s$ for some $s < r+1$, but $f \notin C^{r+1}$, then the order of accuracy will typically be less than $r+1$. There is thus usually no point in using high order methods for functions with low regularity.

Example 13 *Recall example of evaluating*

$$\int_0^1 \sqrt{x}\, dx$$

from section 1.9. Function $\sqrt{x} \in C^0[0,1]$, but it is not continuously differentiable on $[0,1]$, since $\left(\sqrt{x}\right)' = \frac{1}{2\sqrt{x}}$ and obviously this is not continuous at 0. Thus, $\sqrt{x} \notin C^1[0,1]$. Numerical example from section 1.9 confirmed that both Composite Trapezoidal and Composite Simpson's rules have order of accuracy ≈ 1.5 : $E_n(\sqrt{x}) \approx Ch^{1.5}$.

Remark 2
When integrand f is periodic and smooth the Newton–Cotes formulas are considerably more accurate than indicated above. In fact the quadrature error $E(f)$ typically decays exponentially with $n \sim \frac{1}{h}$ rather than just algebraically, as in integral $I^{(3)}$ for Composite Trapezoidal and Composite Simpson's rules. This is because when f is periodic, exact integration of interpolating piece-wise constant/linear/quadratic polynomials agrees with exact integration of interpolating trigonometric functions (Fourier interpolation), which is very accurate when f is periodic and smooth.

2 Gaussian quadrature

All quadrature rules we considered so far have the form

$$\int_a^b f(x)\, dx \approx \omega_1 f(x_1) + \omega_2 f(x_2) + \cdots + \omega_n f(x_n) = \sum_{i=1}^{n} \omega_i f(x_i),$$

for some weights ω_i and nodes x_i from interval $[a,b]$.

Let's consider a different approach in development of quadrature rules. For simplicity, let's consider initially interval $[-1,1]$ and look for a quadrature of the form

$$\int\limits_{-1}^{1} f(x)\, dx \approx \sum_{i=1}^{n} \omega_i f(x_i),$$

which is exact for polynomials of as large degree as possible. There are no restrictions placed on the nodes $\{x_j\}_{j=1}^n$ nor the weights $\{\omega_j\}_{j=1}^n$ in working towards that goal.

As mentioned in previous section, it turns out to be a bad idea to have the node points $\{x_i\}_{i=1}^n$ to be evenly spaced over the interval of integration (Runge's example). But without this restriction on $\{x_i\}$ we will be able to develop a very accurate set of quadrature rule.

Case $n=1$. We look for a formula

$$\omega_1 f(x_1) \approx \int\limits_{-1}^{1} f(x)\, dx.$$

Weight ω_1 and node x_1 are to be chosen such that this formula is exact for polynomials of as large degree as possible. To do this we substitute $f(x) = 1$ and $f(x) = x$ into the above formula.

Choice $f(x) \equiv 1$ leads to

$$\begin{aligned} \omega_1 \cdot 1 &= \int_{-1}^{1} 1\,dx \\ \omega_1 &= 2 \end{aligned}$$

Choice $f(x) \equiv x$ leads to

$$\begin{aligned} \omega_1 x_1 &= \int_{-1}^{1} x\,dx \\ 2x_1 &= 0 \\ x_1 &= 0 \end{aligned}$$

The looked for formula becomes

$$\int_{-1}^{1} f(x)\,dx \approx 2f(0).$$

This integration rule is exact for linear polynomials and note that it is exactly the midpoint rule $M_1(f)$.

Case $n = 2$. We look for a formula

$$\omega_1 f(x_1) + \omega_2 f(x_2) \approx \int_{-1}^{1} f(x)\,dx.$$

Weights ω_1, ω_2 and nodes x_1, x_2 are to be chosen such that this formula is exact for polynomials of as large degree as possible. Observe that we have 4 unknowns $\omega_1, \omega_2, x_1, x_2$ and consequently we will need at least 4 conditions. Substitute $f(x) = 1$, $f(x) = x$, $f(x) = x^2$ and $f(x) = x^3$ in the quadrature formula to obtain the sys

$$\begin{aligned} \omega_1 + \omega_2 &= \int_{-1}^{1} 1\,dx = 2, \\ \omega_1 x_1 + \omega_2 x_2 &= \int_{-1}^{1} x\,dx = 0, \\ \omega_1 x_1^2 + \omega_2 x_2^2 &= \int_{-1}^{1} x^2\,dx = \tfrac{2}{3}, \\ \omega_1 x_1^3 + \omega_2 x_2^3 &= \int_{-1}^{1} x^3\,dx = 0. \end{aligned}$$

This system of 4 **nonlinear** equations has the solution:

$$\omega_1 = 1, \quad \omega_2 = 1, \quad x_1 = -\tfrac{1}{\sqrt{3}}, \quad x_2 = \tfrac{1}{\sqrt{3}}.$$

Thus, our quadrature formula is

$$\int_{-1}^{1} f(x)\,dx \approx f\left(-\tfrac{1}{\sqrt{3}}\right) + f\left(\tfrac{1}{\sqrt{3}}\right).$$

This last quadrature rule has degree of precision 3, since it integrates exactly all polynomials of degree $\leqslant 3$. It can be verified directly that it does not integrate exactly monomial x^4:

$$\int_{-1}^{1} x^4 dx = \tfrac{2}{5}$$

$$\left(-\tfrac{1}{\sqrt{3}}\right)^4 + \left(\tfrac{1}{\sqrt{3}}\right)^4 = \tfrac{2}{9}$$

Example 14 *Integrate*

$$\int_{-1}^{1} \frac{1}{3+x}\, dx = \ln 2 \approx 0.69314718$$

Applying the last quadrature rule we get

$$\frac{1}{3-\frac{1}{\sqrt{3}}} + \frac{1}{3+\frac{1}{\sqrt{3}}} = 0.69230769.$$

Clearly

$$Error = 0.69314718 - 0.69230769 = 0.000839.$$

2.1 Gaussian quadrature. General case

Want to find n weights $\{\omega_i\}_{i=1}^n$ and n nodes $\{x_i\}_{i=1}^n$ such that the quadrature rule

$$\int_{-1}^{1} f(x)\, dx \approx \sum_{i=1}^{n} \omega_i f(x_i)$$

is exact for polynomials $f(x)$ of as large degree as possible. There are $2n$ unknowns:

$$\omega_1\, \omega_2\, \ldots, \omega_n \quad \text{and} \quad x_1,\, x_2,\, \ldots, x_n$$

and therefore it makes sense to impose $2n$ conditions as to obtain $2n$ equations. So, we require that the above quadrature rule to be exact for the cases:

$$f(x) = x^i, \quad i = 0, 1, 2, 3, \ldots, 2n-1$$

and obtain consequently $2n$ equations:

$$\omega_1 x_1^i + \omega_2 x_2^i + \ldots + \omega_n x_n^i = \int_{-1}^{1} x^i\, dx,$$

for $i = 0, 1, 2, 3, \ldots, 2n-1$. In other words we have a system of $2n$ equations (generally nonlinear)

$$\begin{aligned}
\omega_1 + \omega_2 + \ldots + \omega_n &= \int_{-1}^{1} dx,\\
\omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n &= \int_{-1}^{1} x\, dx,\\
\omega_1 x_1^2 + \omega_2 x_2^2 + \ldots + \omega_n x_n^2 &= \int_{-1}^{1} x^2\, dx,\\
\omega_1 x_1^3 + \omega_2 x_2^3 + \ldots + \omega_n x_n^3 &= \int_{-1}^{1} x^3\, dx,\\
\vdots \qquad \vdots \qquad & \vdots \qquad \vdots\\
\omega_1 x_1^{2n-1} + \omega_2 x_2^{2n-1} + \ldots + \omega_n x_n^{2n-1} &= \int_{-1}^{1} x^{2n-1}\, dx,
\end{aligned}$$

for $2n$ unknowns ω_i and x_i. The right sides can be computed:

$$\int_{-1}^{1} x^i\,dx = \begin{cases} \frac{2}{i+1}, & i = 0,2,4,\dots,2n-2 \\ 0, & i = 1,3,5,\dots,2n-1 \end{cases}$$

The system of $2n$ equations with $2n$ variables $\omega_i,\ x_i,\ i = 0,1,2,\dots,2n-1$:

$$\begin{cases} \omega_1 + \omega_2 + \ldots + \omega_n = 2 \\ \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_n x_n = 0 \\ \omega_1 x_1^2 + \omega_2 x_2^2 + \ldots + \omega_n x_n^2 = \frac{2}{3} \\ \omega_1 x_1^3 + \omega_2 x_2^3 + \ldots + \omega_n x_n^3 = 0 \\ \ldots\ldots \quad \ldots\ldots\ldots\ldots \\ \omega_1 x_1^{2n-2} + \omega_2 x_2^{2n-2} + \ldots + \omega_n x_n^{2n-2} = \frac{2}{2n-1} \\ \omega_1 x_1^{2n-1} + \omega_2 x_2^{2n-1} + \ldots + \omega_n x_n^{2n-1} = 0 \end{cases}$$

has a unique solution. The resulting numerical integration rule

$$\boxed{\int_{-1}^{1} f(x)\,dx \approx \omega_1 f(x_1) + \omega_2 f(x_2) + \cdots + \omega_n f(x_n)}$$

is called Gaussian quadrature in n points.
Note that this is a system of **nonlinear** equations. Solving such a system is not straightforward. In fact, the nodes and weights are not found by solving this system directly. Rather, the nodes and weights have other properties which enable them to be found more easily by other methods (see next 2 slides). Most subroutine libraries for Fortran, C/C++, Python, MATLAB, GNU Octave, Julia, etc have either a program to produce them or tables of them for commonly used cases. Gaussian nodes and weights are tabulated for $[-1,1]$ and are presented in the table below. Also, from figure 8, it can be observed that Gaussian nodes cluster toward ± 1, preventing thus Runge's phenomenon!

Nr. of points n	Quad points x_i	Quad weights ω_i
1	0	2
2	$-\frac{1}{\sqrt{3}},\quad \frac{1}{\sqrt{3}}$	$1,\quad 1$
3	$-\sqrt{\frac{3}{5}},\quad 0,\quad \sqrt{\frac{3}{5}}$	$\frac{5}{9},\quad \frac{8}{9},\quad \frac{5}{9}$
$\vdots$	$\vdots$	$\vdots$

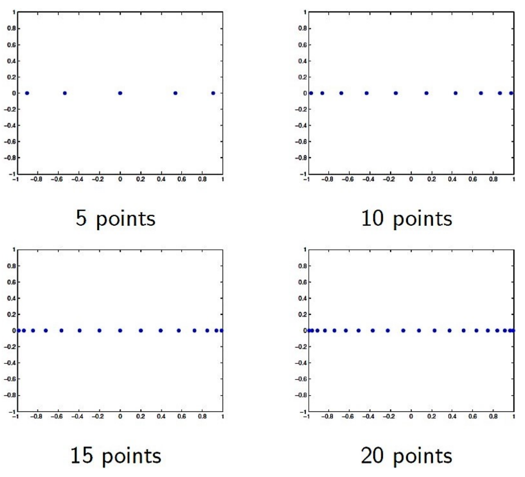

Figure 8: Gaussian nodes cluster toward ± 1, preventing Runge's phenomenon!

The following result can be proved:

Theorem 4 *Gaussian quadrature points $\{x_i\}_{i=1}^{n}$ are the roots of Legendre polynomial of degree n.*

Legendre polynomials are similar to Chebyshev polynomials and can be obtained through a triple recurrence relation. Let $P_0(x) = 1$ and $P_1(x) = x$. Then Legendre polynomials of degree $n \geqslant 2$ are recurrently defined by

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x).$$

The graphs of Legendre polynomials of degree up to 5 are shown in figure 9. Also, it can be shown that Gaussian quadrature rules converge as $n \to \infty$.

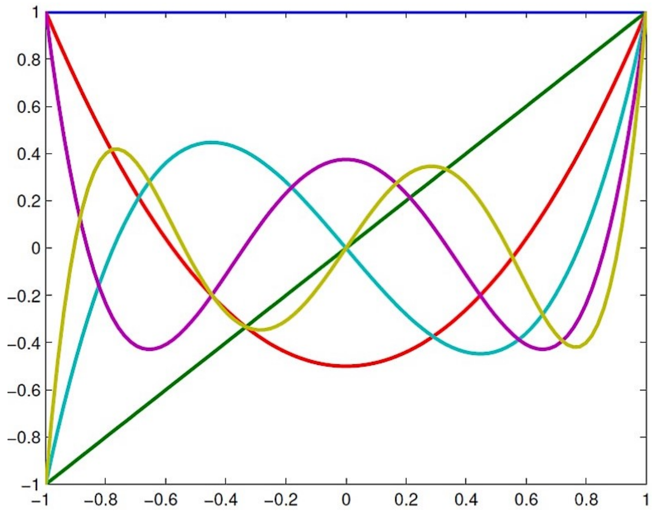

Figure 9: The graphs of the first 6 Legendre polynomials

2.2 Change of interval

Gaussian quadrature formula can be applied to integrals from -1 to 1. What to do in case of integration over generic interval $[a, b]$?

Integrals on other finite intervals $[a, b]$ can be converted to integrals over $[-1, 1]$, as follows:

$$\int_a^b F(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} F\Big(\frac{b+a+t(b-a)}{2}\,dt\Big)$$

based on a change of integration variable

$$x = \frac{b+a+t(b-a)}{2}, \quad t \in [-1, 1].$$

And then apply Gaussian quadrature to get the numerical approximation.

Example 15 *Suppose we need to compute*

$$\int_0^\pi x^2 \cos x\,dx.$$

Over interval $[0, \pi]$, use substitution

$$x = \tfrac{(1+t)\pi}{2}$$

to get

$$\int_0^\pi F(x)\,dx = \tfrac{\pi}{2}\textstyle\int_{-1}^{1} F\left(\tfrac{(1+t)\pi}{2}\right)dt$$
$$\int_0^\pi x^2\cos x\,dx = \tfrac{\pi}{2}\textstyle\int_{-1}^{1} \tfrac{(1+t)^2\pi^2}{4}\cos\left(\tfrac{(1+t)\pi}{2}\right)dt$$

Now, integrate using Gaussian quadrature the integral from the right side.

2.3 Error formula for Gaussian quadrature

Let error in Gaussian quadrature be defined as

$$E_n^G(f) = \int_{-1}^{1} f(x)\,dx - \sum_{i=1}^{n} \omega_i f(x_i).$$

Then the following identity holds

$$\begin{aligned} E_n^G(f) &= \frac{2^{2n+1}(n!)^4}{(2n+1)\left((2n)!\right)^2} \cdot \frac{f^{(2n)}(\xi)}{(2n)!} \\ &= G_n \cdot \frac{f^{(2n)}(\xi)}{(2n)!}, \end{aligned}$$

with some $\xi \in [a,b]$ apriori unknown. Using Stirling's formula we might deduce that

$$G_n \approx \frac{\pi}{4^n}.$$

From the error formula and last remark we get that the the error decreases by a factor of at least 4 with each increase of n to $n+1$. Compare this to the convergence of the Composite Trapezoidal and Composite Simpson's rules for such functions, to help explain the very rapid convergence of Gaussian quadrature rules.

Also, it can be derived a second error formula for Gaussian quadrature:

$$\left|E_n^G(f)\right| \leqslant 2(b-a)\,\rho_{2n-1}(f),$$

where $\rho_{2n-1}(f)$ is the minimax error of degree $2n-1$ for function f on interval $[a,b]$ (see Lecture Notes 4).

Example 16 *Let function $f(x) = e^{-x^2}$ defined on interval $[0,1]$. The minimax errors $\rho_m(f)$ are given below:*

m	$\rho_m(f)$	m	$\rho_m(f)$
1	5.30E-2	6	7.82E-6
2	1.79E-2	7	4.62E-7
3	6.63E-3	8	9.64E-8
4	4.63E-4	9	8.05E-9
5	1.62E-5	10	9.16E-10

Apply second error formula to $\int_0^1 e^{-x^2}\,dx$ and Gaussian quadrature with 3 points to get an error estimate:

$$\left|I - I_3\right| = \left|E_3^G\right| \leqslant 2\,\rho_5\left(e^{-x^2}\right) \approx 3.24 \cdot 10^{-5},$$

which is a close estimate for the true error of $9.55 \cdot 10^{-6}$.

Example 17 *Recall the example of approximating the integral of a non-smooth function by composite Trapezoidal and Composite Simpson's rules. As we have seen the convergence rate of these rules were not as predicted by theory. Now, apply Gaussian quadrature to evaluate*

$$I = \int_0^1 \sqrt{x}\,dx = \tfrac{2}{3}.$$

Numerical results are presented below

n	$I - I_n$	Ratio
2	−7.22E-3	
4	−1.16E-3	6.2
8	−1.69E-4	6.9
16	−2.30E-5	7.4
32	−3.00E-6	7.6
64	−3.84E-7	7.8

The ratio column is converging to 8, which is consistent with

$$I - I_n \approx \frac{C}{n^3},$$

and it is much better than Composite Trapezoidal and Composite Simpson's rules that were consistent with

$$\frac{C}{n^{1,5}}.$$

Also, it should be noted that Gaussian quadrature can be easily extended to 2D and 3D integration, where integration points are chosen appropriately in the 2d or 3d domains.

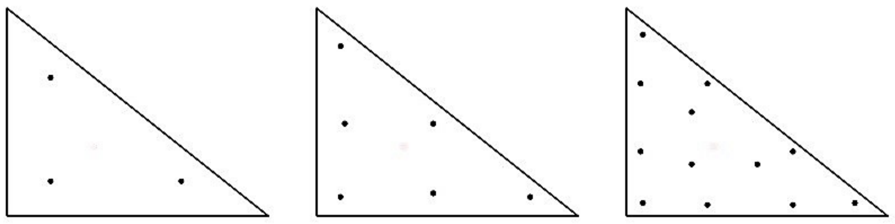

Figure 10: Gaussian nodes for reference triangle ($n = 3, 6, 12$)