3. Numerical Quadrature

Where analytical abilities end . . .

3.1. Preliminary Remarks

The Integration Problem

- **Numerical quadrature** denotes numerical computation of a definite integral of the kind

$$I(f) := \int_{\Omega} f(x)\mathrm{d}x$$

  to a given function $f : \mathbb{R}^d \supseteq \Omega \to \mathbb{R}$, the **integrand**, and a given **integration domain** Ω.

- In the following, we will deal solely with **univariate** quadrature, i.e. with the case $d = 1$ of an interval $\Omega = [a, b]$. Of course, the big challenge is the higher dimensional case of **multivariate** quadrature. Especially the high dimensional case of $d = 100$ (occuring in statistics, physics, and in mathematical finance), or even higher dimensions demands sophisticated numerical methods.

- Numerical quadrature should only be used when all other methods such as closed integration using substitution or partial integration or splitting into sums of integrals between discontinuous points of f or a derivation of f etc. fail.

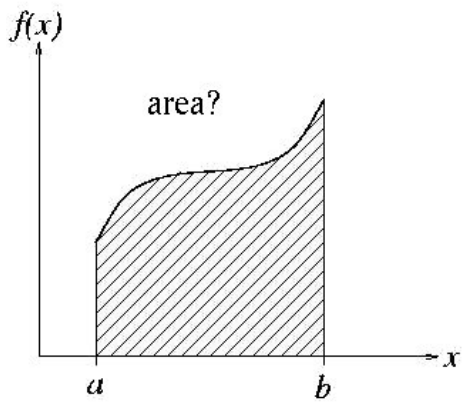
- Most methods of numerical quadrature demand a sufficient smoothness (differentiability) of the integrand.

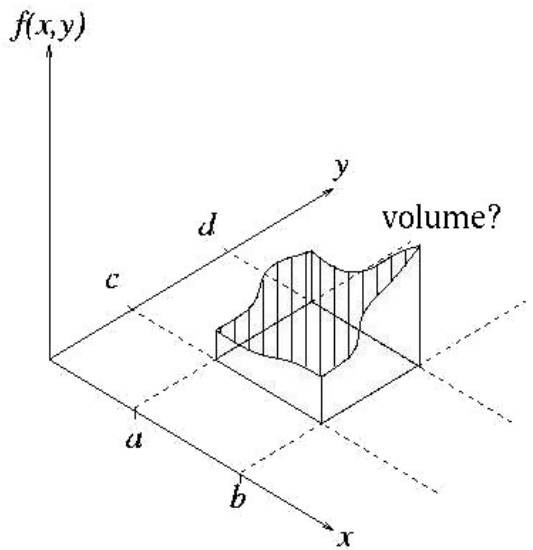
- Almost all **rules of quadrature**, i.e. instructions for numerical quadrature, can be written as **weighted sums of function values (samples)**:

$$I(f) \approx Q(f) := \sum_{i=0}^{n} g_i f(x_i) =: \sum_{i=0}^{n} g_i y_i$$

with **weights** g_i and pairwise different **nodes** x_i, where $a \leq x_0 < x_1 < \ldots < x_{n-1} < x_n \leq b$.

- As the evaluation of the integrand is often an expensive issue (especially when f is only given implicitly: for example, in some applications, a differential equation has to be solved every time to be able to evaluate f at a point x), one is interested in rules that allow a high accuracy (a small **error of quadrature**) with moderate n.





Exact Integration of a Polynomial Interpolant

- How do we get appropriate rules of quadrature? The standard ansatz is to replace the integrand f by an approximation $\tilde{f}$ that is both easy to construct and easy to integrate which will then be integrated *exactly*, thus

$$Q(f) := \int_a^b \tilde{f}(x)\mathrm{d}x\,.$$

- For reasons of simplicity, a **polynomial interpolant** $p(x)$ of $f(x)$ to the nodes x_i is chosen as approximant $\tilde{f}$. In this case, the representation of $p(x)$ via the Lagrange polynomials $L_i(x)$ of degree n delivers the weights g_i virtually free of cost:

$$Q(f) := \int_a^b p(x)\mathrm{d}x = \int_a^b \sum_{i=0}^{n} y_i L_i(x)\mathrm{d}x = \sum_{i=0}^{n} \left(y_i \cdot \int_a^b L_i(x)\mathrm{d}x \right),$$

with which the weights g_i are directly defined by

$$g_i := \int_a^b L_i(x)\mathrm{d}x$$

The integrals of Lagrange polynomials can obviously be calculated in advance – although they depend on the chosen grid (the nodes), they do not depend on the integrand f.

- Because of the uniqueness of the interpolation problem (which polynomial interpolant of degree less or equal to n interpolates the $n+1$ points $(x_i, 1)$, $i = 0, ..., n$?), we have

$$\sum_{i=0}^{n} L_i(x) \equiv 1$$

and therefore also

$$\sum_{i=0}^{n} g_i = \int_a^b \sum_{i=0}^{n} L_i(x)\mathrm{d}x = \int_a^b 1\mathrm{d}x = b - a\,.$$

That means that the sum of the weights is always equal to $b - a$ when a polynomial interpolant is used for quadrature.

About the Condition of Numerical Quadrature

- How is the problem of numerical quadrature conditioned? Particularly, are there any requirements for the weights g_i to ensure a good condition?
  - To answer this question, we examine small changes δy_i in the input data – the nodes $y_i = f(x_i)$ – as well as their effect on the approximate value of the integral.
  - If all input fluctuations are less or equal ε, we have for a change $\delta Q(f)$ of the approximation value

$$|\delta Q(f)| = \left| \sum_{i=0}^{n} g_i \delta y_i \right| \leq \varepsilon \cdot \sum_{i=0}^{n} |g_i| .$$

  - If all weights g_i are positive, then the sum on the right-hand side takes its minimal value $b - a$ and the entire right-hand side is of order of magnitude of $O(\varepsilon)$. Thus, in this case, the quadrature problem is well-conditioned.
  - If, in contrast, negative weights occur, then the sum on the right-hand side of the estimation above might become very big – and with it the upper bound of $\delta Q(f)$. In case of negative weights, the problem of quadrature can be ill-conditioned.

– As we will see in the following section, this demand for positivity of the weights is an exclusion criterion for the integration via the polynomial interpolant in case of big n. As we have already seen at polynomial interpolation, polynomials of high degree are problematic.

3.2. Simple and Composite Rules

The Rectangle Rule

- In the following, we will introduce some of the most important rules of quadrature. We distinguish between **simple** and **composite** quadrature rules:
  - A simple rule deals with the whole integration domain $[a, b]$ in one go. For its length, we will use the denotation $H := b - a$.
  - A composite rule splits the integration domain into subdomains, applies simple rules there, and forms the total approximation by summing up. This procedure is very similar to the spline interpolation of chapter 2.
- The simplest simple rule is the **rectangle rule**

$$Q_R(f) := H \cdot f\left(\frac{a+b}{2}\right) = I(p_0)\,,$$

where p_0 denotes the polynomial interpolant of f of degree 0 with the only node $x_0 := (a+b)/2$

  - For the **remainder** $R_R(f) := Q_R(f) - I(f)$, the relation

$$R_R(f) := -H^3 \cdot \frac{f^{(2)}(\xi)}{24}$$

can be shown for an intermediate point $\xi \in]a, b[$, if f is twice continuously differentiable on $]a, b[$.

- From this relation of the remainder, we first learn that polynomials of degree 0 or 1 are integrated *exactly*. That might surprise at first glance as only a constant interpolant is used. However, in the linear case, the quadrature errors left and right to the centre point of the interval cancel each other.
- Second, we see via the H-Terms what has already been clear geometrically: On a small integration interval, the error is asymptotically smaller as well. However, for now, we only examine constant integration domains for the simple rules.

The Trapezoidal Rule

- If we use the linear polynomial interpolant p_1 instead of the constant polynomial interpolant p_0, we get the **trapezoidal rule**:

$$Q_T(f) \;:=\; H \cdot \frac{f(a)+f(b)}{2} \;=\; I(p_1)\,.$$

  - Here, the linear interpolant to the two interval bounds $x_0 := a$ and $x_1 := b$ is integrated exactly.
  - For the remainder, it can be shown (again without proof)

$$R_T(f) \;=\; H^3 \cdot \frac{f^{(2)}(\xi)}{12}\,,$$

    where ξ again identifies an intermediate point in the interior of the integration domain.
  - At first glance, the trapezoidal rule might seem hardly helpful: We invest twice as much as before (two function evaluations), but only get a result of comparable quality because the integration errors left and right to the centre point of the interval do not cancel each other in case of a quadratic interpolant. The justification of the trapezoidal rule lies in its superb eligibility as starting point for composite methods.
- The maximal polynomial degree that can be treated exactly by a quadrature rule is called **degree of accuracy** or shortly **accuracy** of the method. Thus, the rectangle rule as well as the trapezoidal rule have accuracy 1.

Kepler's Rule and Newton-Cotes Formulas

- You may guess it: Next, we will take the quadratic interpolant p_2 to the three nodes $x_0 := a$, $x_1 := (a+b)/2$ and $x_2 := b$ and, with a short calculation, we get **Keplers's rule**

$$Q_F(f) \;:=\; H \cdot \frac{f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)}{6} \;=\; I(p_2)\,.$$

  (By the way, this and more can be admired in the Kepler Museum in *Weil der Stadt*.)
  - Here, the remainder fulfils

$$R_F(f) \;=\; H^5 \cdot \frac{f^{(4)}(\xi)}{2880}\,.$$

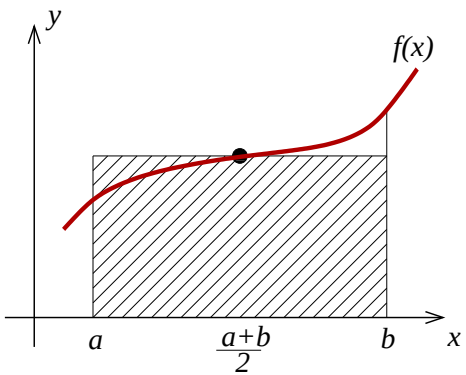
  - If the integrand is four times continuously differentiable, Kepler's rule thus presents a method of order 5 and of accuracy 3.
- The class of quadrature rules that follows from this principle is called **Newton-Cotes formulas**. Here, it is thus defined

$$Q_{\mathrm{NC}(n)}(f) \;:=\; I(p_n)\,,$$

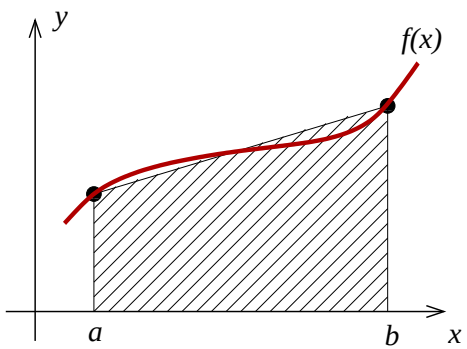
  where p_n is the polynomial interpolant of f of the degree n to the $n+1$ equidistant nodes

$$x_i \;:=\; a + H \cdot \frac{i}{n}\,, \quad i = 0, ..., n\,.$$

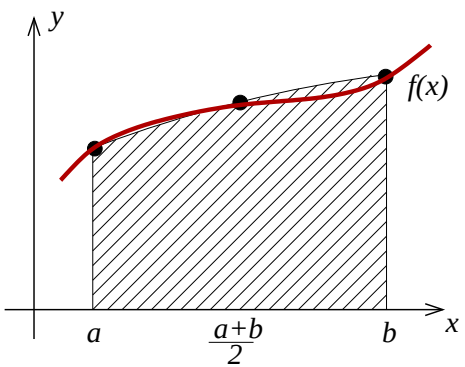
– Under the assumption of adequately high differentiability, we get with the Newton-Cotes rules for growing n, we get methods of higher order and higher accuracy.
– **Attention**: For $n = 8$ and $n \geq 10$ negative weights occur. As hinted before, in those cases, the Newton-Cotes formulas are practically useless.



rectangle rule



trapezoidal rule



Kepler's rule

Clenshaw-Curtis Formulas

- The problem of negative weights g_i at higher polynomial degree n does not mean that polynomial interpolants of higher degree cannot serve for purposes of numerical quadrature in principle. One possible remedy is to give up the equidistance of the nodes.
- Exactly this is the principle of the **Clenshaw-Curtis rules**, in which instead of the integration interval $[a, b]$ the semicircle angle $[0, \pi]$ is uniformly subdivided:

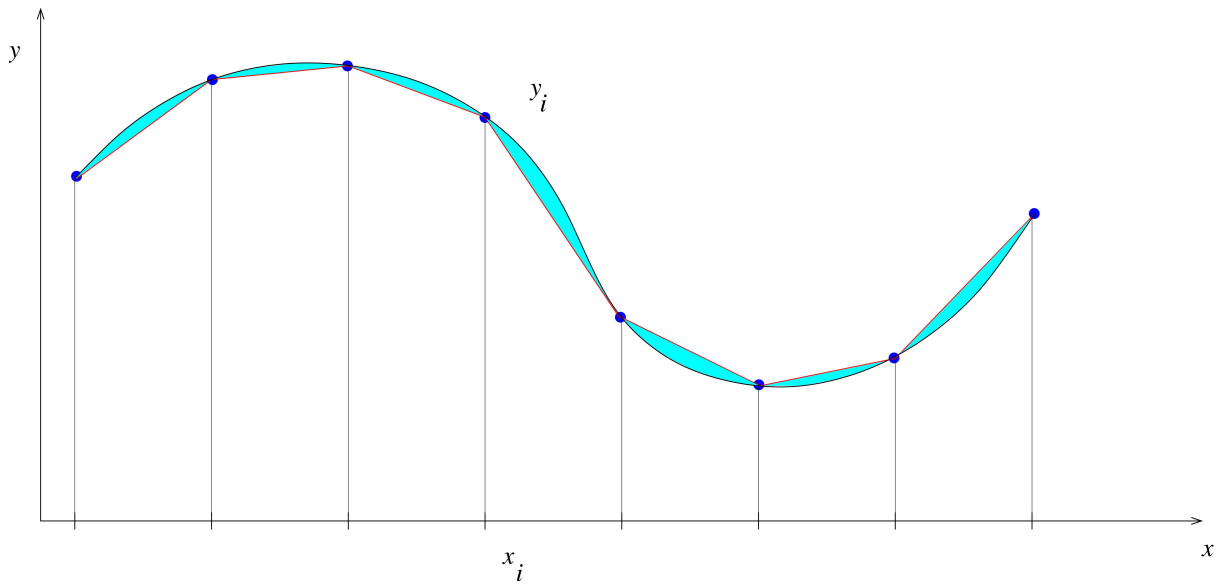
$$x_i \;:=\; a + H \cdot \frac{1 - \cos\left(\frac{i\pi}{n}\right)}{2} \,, i = 0, ..., n \,.$$

- It can be easily demonstrated that the nodes are located more densely at the borders of the integration domain than at its center.
- It can be shown that for this principle of construction of quadrature rules all appearing weights are always positive.
- Therefore, the Clenshaw-Curtis rules also are suited for larger values of n in principle.
- However, those ansatzs only play a little role in practice. Here, composite rules, which we will examine in the following, are more important.

The Trapezoidal Sum

- The probably most important composite rule is the **trapezoidal sum**:
  - Firstly, the integration interval $[a, b]$ is divided into n subintervals of length
$$h := \frac{b-a}{n}.$$
  - The equidistant junctions $x_i := a + ih$, $i = 0, ...n$, serve as nodes.
  - Now the trapezoidal rule is applied to every subinterval $[x_i, x_{i+1}]$.
  - Finally, the integral values computed this way are summed up to the total integral value:
$$Q_{\mathrm{TS}}(f;h) := h \cdot \left(\frac{f_0}{2} + f_1 + f_2 + ... + f_{n-1} + \frac{f_n}{2} \right),$$
    where $f_i := f(x_i)$.
  - For the remainder term of the trapezoidal rule, we have the relation
$$R_{\mathrm{TS}}(f;h) = h^2 \cdot (b-a) \cdot \frac{f^{(2)}(\xi)}{12} = h^2 \cdot H \cdot \frac{f^{(2)}(\xi)}{12},$$
    which follows immediately from the remainder formula of the trapezoidal rule by summation of the single remainder terms and multiple application of the intermediate value theorem.

- What does this formula tell us about the remainder term of the trapezoidal sum?
  - Compared to the trapezoidal rule, the accuracy remains 1, whereas the order declines to 2 (one order of magnitude is lost by summation).
  - However, now we have a rule that is easy to implement and with which n can be arbitrarily increased without getting numerical problems.



The Simpson Sum

- In the same way the trapezoidal sum is a composite quadrature rule on the basis of the trapezoidal rule, the **Simpson sum** is a natural generalization of Kepler's rule.
- Starting with the same partition of the integration interval $[a, b]$ as before, now Kepler's rule is applied to every two neighboring subintervals conjointly:

$$Q_{\mathrm{SS}}(f;h) \;:=\; \frac{h}{3} \cdot (f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n)\,.$$

- As remainder term $R_{\mathrm{SS}}(f)$ we get

$$R_{\mathrm{SS}}(f;h) \;=\; h^4 \cdot (b-a) \cdot \frac{f^{(4)}(\xi)}{180} \;=\; h^4 \cdot H \cdot \frac{f^{(4)}(\xi)}{180}\,.$$

  - Again the order is reduced by 1 to 4 in comparison to the respective simple rule, whereas the accuracy stays unchanged at 3.
  - The Simpson sum also offers a simple way to improve the approximation by increasing the number of nodes without getting problems with condition.

3.3. Extrapolation

Increasing the Order Using Linear Combinations

- Once more, we consider Kepler's rule. It is possible to get this formula from the **linear combination of three trapezoidal rules** (once with step width H and twice with step width $H/2$):

$$\begin{aligned} Q_F(f) &= H \cdot \frac{f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)}{6} \\ &= \frac{4}{3}\left(\frac{H}{2} \cdot \frac{f(a) + f\left(\frac{a+b}{2}\right)}{2}\right) + \frac{4}{3}\left(\frac{H}{2} \cdot \frac{f\left(\frac{a+b}{2}\right) + f(b)}{2}\right) \\ &\quad - \frac{1}{3}\left(H \cdot \frac{f(a) + f(b)}{2}\right) \\ &= \frac{4}{3}\frac{H}{2}\left(\frac{f(a)}{2} + f(\frac{a+b}{2}) + \frac{f(b)}{2}\right) - \frac{1}{3}H\left(\frac{f(a)}{2} + \frac{f(b)}{2}\right). \end{aligned}$$

As the last line shows, this can also be interpreted as a **linear combination of two trapezoidal sums**.

- This means: Clever combination of approximation values of *low* approximation order leads to an approximation of *higher* approximation order – without explicitly using the formula of higher order.
- This is a very common principle in numerics that can also be successfully used for other problems than quadrature.
- For the trapezoidal rule this method is less common, but for the trapezoidal sum it is an extremely worthwhile starting point which we will study in the following.

The Principle of Extrapolation

- The starting point is the **Euler-Maclaurin sum formula** which – to put it simple – delivers an h^2-series expansion of the trapezoidal sum if f is at least $2m$-times continuously differentiable:

$$Q_{\mathrm{TS}}(f;h) \;=\; I(f) + \tau_1 h^2 + \ldots + \tau_{m-1} h^{2m-2} + \tau_m(h) \cdot h^{2m} \,,$$

  with a remainder $\tau_m(h)$ bounded from above for all h (that is the crux of the matter – without this constraint it could always be written like that).

- **Idea**: Compute approximation values $T(h_i)$ for the sought integral $I(f)$ using the trapezoidal sum, namely for the different step widths h_i.

- For the choice of the (sensibly converging to zero) sequence of step width h_i, $i = 1, 2, 3, \ldots$, there are two important alternatives:
  - successive bisection: $h_i := \frac{b-a}{n_i}$, $n_i := 2^i$,
  - slow accretion: $h_i := \frac{b-a}{n_i}$, $n_i := 2, 3, 4, 6, 8, 12, \ldots$.

The Principle of Extrapolation (2)

- Now, write the expansions for the chosen h_i one below the other:

$$\begin{aligned} T(h_1) &= I(f) + \tau_1 {h_1}^2 + \ldots + \tau_{m-1} {h_1}^{2m-2} + \tau_m(h_1) \cdot {h_1}^{2m} \\ T(h_2) &= I(f) + \tau_1 {h_2}^2 + \ldots + \tau_{m-1} {h_2}^{2m-2} + \tau_m(h_2) \cdot {h_2}^{2m} \\ T(h_3) &= I(f) + \tau_1 {h_3}^2 + \ldots + \tau_{m-1} {h_3}^{2m-2} + \tau_m(h_3) \cdot {h_3}^{2m} \\ \ldots &= \ldots \end{aligned}$$

  Note that the left hand sides (the computed trapezoidal sum values) are known whereas $I(f)$ as well as all τ_i are not known and are only given formally. Above, everything unknown is marked red.

- Now it is clear how we have to proceed: Combine pairs of rows such that the respective leading term of order in h^2 is eliminated – and, hey presto, we have approximations that are two orders better.

The Principle of Extrapolation (3)

- Romberg had a different perception for this idea of Richardson and attained a more efficient calculation scheme:
  - Due to the Euler-Maclaurin sum formula, it is natural to consider T as a function of h^2, to approximate it by a polynomial p of degree $m-1$ that interpolates T in the nodes $(h_i^2, T(h_i))$, $i = 1, ..., m$, and to use $p(0)$ as approximation for $I(f) = T(h \to 0)$.
  - As the sought "intermediate point" $h = 0$ here is located outside of the interval of nodes h_i, we speak of **extrapolation** instead of interpolation.
  - The technical realization of the computation of $p(0)$ is done best with Neville's scheme (see section 2.2).
  - From this results the following algorithm, the **Romberg quadrature**:

```
for i:=1 to m do
  chose n[i]; h[i]:=(b-a)/n[i];
  T[i]:=trapezoidal sum of step width h[i]
  for k:=i-1 downto 1 do
    T[k]:=T[k+1]+(T[k+1]-T[k])/(h[k]**2/h[i]**2-1)
  od;
od;
p(0):=T[1];
```

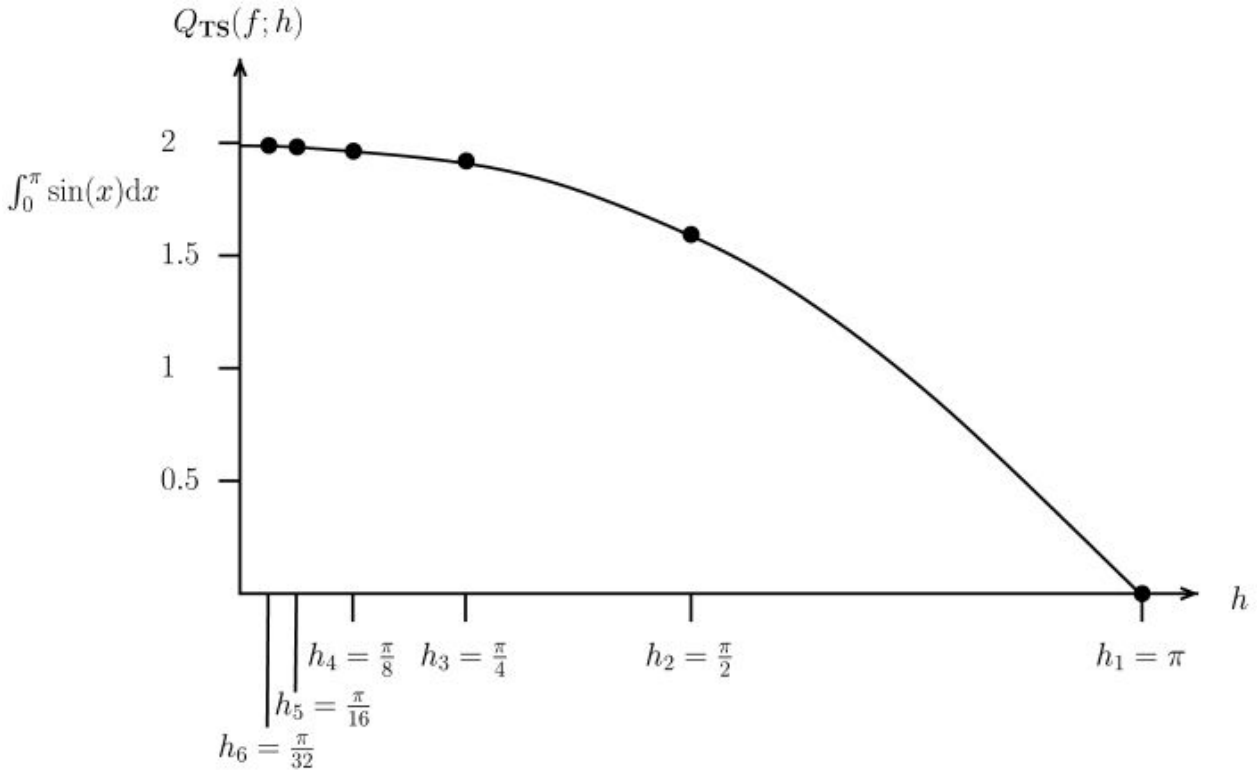
(Verify this algorithm with the help of Neville's algorithm!)

Short Analysis of Extrapolation

- Accuracy of extrapolation:
  - The attractiveness of the quadrature via extrapolation lies in its significantly higher approximation quality compared to the trapezoidal rule. For the error of quadrature, it can be shown that

$$|p(0) - I(f)| \;=\; O\left(h_1^2 \cdot h_2^2 \cdot \ldots \cdot h_m^2\right),$$

    if f is $2m$-times continuously differentiable and p denotes the interpolating polynomial with $p(h_i^2) = T(h_i)$, $i = 1, \ldots, m$.
  - A calculation example for several values of m may illustrate the meaning of "significant" in this context!
- The price for better approximation is higher differentiability of the integrand f that has to be presumed. Therefore, extrapolation is to be used whenever very smooth functions have to be integrated. However, for functions with kinks etc., it does not help.
- This also is a very typical situation for numerics. Not only performance and speed are of importance but also universality (the so-called **robustness**): How big is the class of functions to which a method can be applied?



h	h_1	h_2	h_3	h_4	h_5	h_6
$Q_{\mathrm{TS}}(f;h)$	0	1.571	1.896	1.974	1.993	1.998
error	2	0.429	0.104	$2.6 \cdot 10^{-2}$	$7 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

For comparison:
Romberg quadrature for h_6 delivers the error $= 2 \cdot 10^{-9}$ (smaller by a factor of 10^{-6} to the error of the trapezoidal sum for h_6!).

3.4. Other Approaches

Monte-Carlo Quadrature

- We have only discussed the one-dimensional case here. Of course, more interesting is the multivariate case of integrands with several variables. Here, the **curse of the dimension** leads to a fast increase of the number of nodes and thereby also of the computational costs and memory requirements: A product approach of a rule with n points in one dimension already requires n^2 in two and n^d in d space dimensions.
- Nowadays, totally different methods than the examined ones are used for large values of d, for example **Monte-Carlo methods** as well as **quasi Monte-Carlo methods** (also called **minimal discrepancy methods**).
- **Monte-Carlo methods** are of stochastic nature.
- We examine a simple two-dimensional example:
  - Let $\chi_A(x, y)$ and $\chi_B(x, y)$ be the characteristic functions to the circle $A := \{(x, y) : x^2 + y^2 \leq 1\}$ and to the rectangle $B := [-1, 1]^2$, respectively (with value 1, if inside, and with value 0, if outside, respectively). The integral of $\chi_A(x, y)$ is sought, hence, de facto the circle constant π.
  - Our random experiment: Choose a point $(x_i, y_i) \in B$ randomly (uniform distribution!).

  - Our random variables Z_i: $Z_i = \chi_A(x_i, y_i)$.
  - Then, the law of large numbers states that, with increasing n, $\frac{1}{n} \cdot \sum_{i=1}^{n} Z_i$ converges to $\pi/4$ in the sense of probability (i. e. not for “sure”!).

- The whole thing defined as a problem of quadrature:

$$\pi = \int_B \chi_A(x,y)\mathrm{d}(x,y) = 4 \cdot \int_{\mathbb{R}^2} \frac{1}{4} \cdot \chi_B(x,y) \cdot \chi_A(x,y)\mathrm{d}(x,y)$$

$$= 4 \cdot E(\chi_A((X,Y))) \approx 4 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} \chi_A(x_i, y_i) = \frac{4}{n} \cdot \sum_{i=1}^{n} Z_i \,.$$

Note that the function $\frac{1}{4} \cdot \chi_B(x,y)$ here plays the role of the probability density.

Gaussian Quadrature

- By means of the Clenshaw-Curtis rules we already saw in section 3.2 that it might be of advantage to leave the equidistant distribution. That's exactly what we want to do now in a systematic way.
- For reasons of simplicity we examine the integration interval $[-1, 1]$. Then the standard ansatz is

$$I(f) = \int_{-1}^{+1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} g_i f(x_i)$$

  with nodes x_i and weights g_i, where now the x_i are also free parameters of the method which can be used to fulfill additional requirements.
- The idea of the **Gaussian quadrature** is to place the nodes in a way that polynomials of a degree as big as possible are integrated exactly – the highest possible **accuracy degree** is aimed at:

$$I(p) = \int_{-1}^{1} p(x)\,\mathrm{d}x \stackrel{!}{=} \sum_{i=1}^{n} g_i p(x_i)$$

  for all $p \in \mathbb{P}_k$ with k as big as possible. At this, we keep in the back of our minds the series expansion for f, of which as many leading terms as possible are supposed to be integrated exactly (high orders of error for decreasing interval width).
- With the Gaussian quadrature the (maximum possible) degree of accuracy $2n - 1$ is obtained.

The Nodes of the Gaussian Quadrature

- We skip the derivation and come right to business:
  - There is exactly one set of nodes x_i, $i = 1, ..., n$, with which the degree of accuracy $2n - 1$ is achievable, and those are the pairwise different roots of the **Legendre polynomials**

$$\omega_n(x) := \frac{n!}{(2n)!} D^n \left((x^2 - 1)^n \right) .$$

    At the boundaries, they are located more densely than in the interior of the integration interval.
  - There is exactly one set of weights g_i with which the degree of accuracy $2n - 1$ is achievable, namely

$$g_i := \int_{-1}^{+1} L_i(x) \, \mathrm{d}x$$

    with Lagrange polynomials $L_i \in \mathbb{P}_{n-1}$ that were already introduced in the last chapter.
  - The necessity of this requirement is clear: The L_i have to be integrated exactly as polynomials of degree $n - 1$ by the Gaussian quadrature, in the weighted sum all summands except $g_i L_i(x_i) = g_i$ disappear due to its definition.

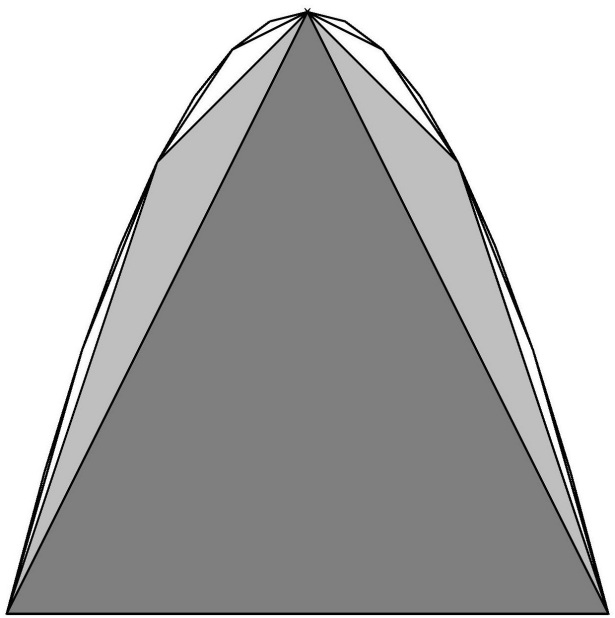
  - As the $L_i^2(x) \in \mathbb{P}_{2n-2}$ are also still integrated exactly, furthermore, it follows that all weights are positive. This underlines the advantages of the Gaussian quadrature.

- The roots of the Legendre polynomials are tabled for common values of n in respective books.

Archimedes' Quadrature

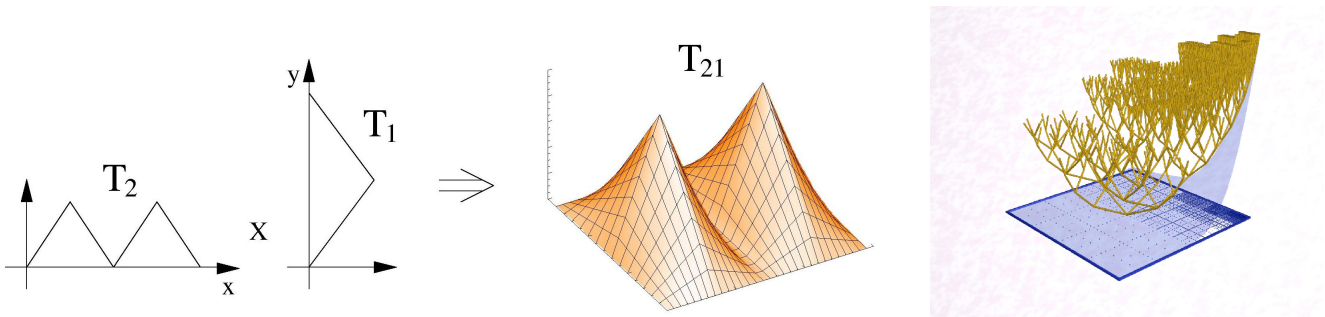
- A very old but nevertheless convincing approach for the numerical quadrature – primarily when considered from the algorithmic point of view – comes from Archimedes.
- This is an ancestor of the algorithmic paradigm **divide et impera** omnipresent in computer science.
- **Problem**: Compute the area (the integral) between the function $f(x) := 1 - x^2$ and the segment of the x-axis between -1 and 1 (numerically – the lucky Archimedes did not yet know about calculus and integration).
- **Idea**:
  - Choose a triangle with vertices $(-1, 0)$, $(0, 1)$, and $(1, 0)$ as a first approximation.
  - The difference area consists of two parabola segments of half width, on which this method is now applied to.
  - Continue recursively, until the partial areas stay under a threshold ε or until a maximum depth is reached.
  - The sum of all partial areas is an approximation for the value of the integral – in the concrete example with the geometric series even the formula for the exact value $4/3$.

- **Advantages**:
  - The method is simple.
  - The method is hierarchically incremental: The first big triangles are also used within a finer resolution. Applying other methods, everything calculated beforehand becomes obsolete when refining (which was a disaster for people without calculator such as Archimedes).
  - The method offers direct access to the **adaptive refinement**: By the areas of the individual triangles, it can be controlled where it should be further refined and where it should not.



Archimedes in Multiple Dimensions

- Archimedes' quadrature can be easily generalized to the case of multiple space dimensions:
  - The higher dimensional space with is dealt with dimensionwise (tensor product ansatz, for the first time demonstrated by Cavalieri).
  - In 2D, the volume is filled with pagoda instead of triangles, in the general d-dimensional case with according equivalents.
  - Here as well, a hierarchical basis underlies – the base of the pagoda gets smaller and smaller.
  - By omitting "dispensable" nodes (pagoda with smaller contribution to the volume), very efficient quadrature techniques (so-called **sparse grids**) result here – especially for the nasty high dimensional case.



3.5. Applications of Quadrature

Numerical Quadrature in Computer Science

- In computer graphics, **radiosity methods** are used to render global lighting as realistic as possible. For this purpose, at every point the incoming and the outgoing light is summed up over all possible directions, i.e. it is integrated.
- In visualization, at **volume rendering**, integration occurs, i.e. when rendering 3D data (e.g. of imaging in medical science), to find suitable color values for the individual pixels (cumulated and weighted values from the interior of the object along a ray, e.g. to show something specific and to hide other things).
- In numerical simulation, **finite elements methods** are a standard method to solve partial differential equations numerically. At this class of methods, so-called ansatz functions have to be integrated over simple structured elements.
- Everywhere in image processing when **smoothing** or **blurring**, integrals are also involved. Especially, every averaging process of continuous data stands for an integration process.
- ...