

Numerical integration

Anne Kværnø, Markus Grasmair

Sep 2, 2022

Corresponding Python code: `quadrature.py`.

1 Introduction

In this note, we will discuss numerical methods for the approximation of (finite) integrals of the form

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

A *numerical quadrature* or a *quadrature rule* is a formula for approximating such integrals. Quadratures are usually of the form

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

where x_i , w_i for $i = 0, 1, \dots, n$ are respectively the *nodes* and the *weights* of the quadrature rule. If the function f is given from the context, we will for simplicity denote the integral and the quadrature simply as $I(a, b)$ and $Q(a, b)$. Examples of numerical quadratures known from previous courses are the [trapezoidal rule](#), the [midpoint rule](#) and [Simpson's rule](#).

In practice, we will not use a single (or *simple*) quadrature for the whole interval $[a, b]$, but rather choose a partitioning

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

into m sub-intervals, apply a quadrature on each of the subintervals $[X_j, X_{j+1}]$, and then add the results together. This leads to *composite quadratures* yielding the approximation

$$I[f](a, b) = \sum_{j=0}^{m-1} I[f](X_j, X_{j+1}) \approx \sum_{j=0}^{m-1} Q[f](X_j, X_{j+1}).$$

In this note we will see how quadrature rules can be constructed from integration of interpolation polynomials. We will demonstrate how to do error analysis and how to find error estimates both for simple and composite quadratures. Moreover, we will demonstrate how the partitioning of the integration interval can be chosen automatically based on these error estimates; this idea is called *adaptive integration*.

In the sequel, we will use material from *Preliminaries*, section 3.2, 4 and 5.

2 Quadrature based on polynomial interpolation.

This section relies on the content of the note on polynomial interpolation, in particular the section on Lagrange polynomials.

Choose $n + 1$ distinct nodes x_i , $i = 0, \dots, n$ in the interval $[a, b]$, and let $p_n(x)$ be the interpolation polynomial satisfying the interpolation condition

$$p_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n$$

We will then use $\int_a^b p_n(x)dx$ to approximate $\int_a^b f(x)dx$. By using the Lagrange form of the polynomial

$$p_n(x) = \sum_{i=0}^n f(x_i) \ell_i(x)$$

with the cardinal functions $\ell_i(x)$ given by

$$\ell_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j},$$

the following quadrature formula is obtained

$$Q[f](a, b) = \int_a^b p_n(x)dx = \sum_{i=0}^n f(x_i) \int_a^b \ell_i(x)dx = \sum_{i=0}^n w_i f(x_i).$$

The weights in the quadrature are simply the integral of the cardinal functions over the interval:

$$w_i = \int_a^b \ell_i(x) dx.$$

Some comments on notation: When the function f and/or the interval is clear from the context, they will often be omitted in the expression for Q and I , thus we may write $I(a, b)$ rather than $I[f](a, b)$, etc. Further, Q is here used as a generic name for the quadrature, but some well known methods have their own commonly used notation. Thus the trapezoidal rule will be denoted $T[f](a, b)$ and Simpson's rule $S[f](a, b)$ rather than $Q[f](a, b)$.

Let us derive two schemes for integration over the interval $[0, 1]$, and apply them to the integral

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

Example 1: Let $x_0 = 0$ and $x_1 = 1$. The cardinal functions and thus the weights are given by

$$\begin{aligned} \ell_0(x) &= 1 - x, & w_0 &= \int_0^1 (1 - x)dx = 1/2 \\ \ell_1(x) &= x, & w_1 &= \int_0^1 xdx = 1/2 \end{aligned}$$

and the corresponding quadrature rule, better known as the trapezoidal rule and usually denoted by T , is given by

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

This formula applied to the function $f(x) = \cos(\pi x/2)$ gives

$$T(0, 1) = \frac{1}{2} \left[\cos(0) + \cos\left(\frac{\pi}{2}\right) \right] = \frac{1}{2},$$

and the error is

$$I(0, 1) - T(0, 1) = \frac{2}{\pi} - \frac{1}{2} = 0.138 \dots$$

Example 2 (Gauss-Legendre quadrature with two nodes): Let $x_0 = 1/2 + \sqrt{3}/6$ and $x_1 = 1/2 - \sqrt{3}/6$. Then

$$\begin{aligned} \ell_0(x) &= -\sqrt{3}x + \frac{1 + \sqrt{3}}{2}, & w_0 &= \int_0^1 \ell_0(x)dx = 1/2, \\ \ell_1(x) &= \sqrt{3}x + \frac{1 - \sqrt{3}}{2}, & w_1 &= \int_0^1 \ell_1(x)dx = 1/2. \end{aligned}$$

The quadrature rule is

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

And this quadrature applied to $f(x) = \cos(\pi x/2)$ is given by

$$Q(0, 1) = \frac{1}{2} \left[\cos\left(\frac{\pi}{2}x_0\right) + \cos\left(\frac{\pi}{2}x_1\right) \right] = 0.635647\dots$$

with an error

$$I(0, 1) - Q(0, 1) = 9.72\dots \cdot 10^{-4}.$$

So the choice of nodes clearly matters!

Before concluding this section, let us present simple indication on the quality of a method:

Definition: Degree of precision.

A numerical quadrature has degree of precision d if:

- $Q[p](a, b) = I[p](a, b)$ for all $p \in \mathbb{P}_d$.
- There exists $p \in \mathbb{P}_{d+1}$ such that $Q[p](a, b) \neq I[p](a, b)$.

Since both integrals and quadratures are linear in the integrand f , the degree of precision is d if and only if the following conditions hold:

$$\begin{aligned} I[x^j](a, b) &= Q[x^j](a, b), & j = 0, 1, \dots, d, \\ I[x^{d+1}](a, b) &\neq Q[x^{d+1}](a, b) \end{aligned}$$

All quadratures constructed from Lagrange interpolation polynomials in $n + 1$ distinct nodes will automatically be of precision at least n . This follows immediately from the way these quadratures are constructed: Indeed, if $p \in \mathbb{P}_n$ is a polynomial of degree at most n , then the interpolation polynomial will simply be equal to p itself, and thus the integration is performed exactly.

Note, however, that the degree of precision can actually be larger than n . It is left to the reader to show that the trapezoidal rule from Example 1 has degree of precision 1, whereas formula from Example 2 has degree of precision 3.

3 Construction and analysis of quadrature rules

In the following, we will discuss the necessary steps for the construction of realistic algorithms for numerical integration, similar to those used in software like Matlab or SciPy. The steps are:

Construction.

1. Choose $n + 1$ distinct nodes on a standard interval $[-1, 1]$.
2. Let $p_n(t)$ be the polynomial interpolating some general function $f(t)$ in the nodes, and let the $Q[f](-1, 1) = I[p_n](-1, 1)$.
3. Transfer the formula Q from $[-1, 1]$ to an arbitrary 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.

We will in this chapter go through the steps 1.-5. for one specific method, namely Simpson's formula, for which we will adopt the standard notation $S[f](a, b)$. After that, we will briefly discuss the same approach for the trapezoidal and midpoint rules. The strategy is quite generic, so it is more important to understand, and remember, how results are derived, not exactly what they are.

Step 6 — error estimation and creation of adaptive algorithms — is more intricate and is therefore relegated to a separate chapter.

3.1 Construction of Simpson's rule

The quadrature formula on the standard interval $[-1, 1]$. The quadrature rule is defined by the choice of nodes on a standard interval $[-1, 1]$. For Simpson's rule, choose the nodes $t_0 = -1$, $t_1 = 0$ and $t_2 = 1$. The corresponding cardinal functions are

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

which gives the weights

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

such that

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

Simpson's rule has degree of precision 3 (check it yourself).

Example 3:

$$\int_{-1}^1 \cos\left(\frac{\pi t}{2}\right) dt = \frac{4}{\pi} \approx \frac{1}{3} [\cos(-\pi/2) + 4\cos(0) + \cos(\pi/2)] = \frac{4}{3}.$$

Transfer the integral and the quadrature to the interval $[a, b]$. The integral and the quadrature is transferred to some arbitrary interval $[a, b]$ by the transformation

$$x = \frac{b-a}{2}t + \frac{b+a}{2}, \quad \text{so} \quad dx = \frac{b-a}{2}dt.$$

By this transformation, the nodes $t_0 = -1$, $t_1 = 0$, and $t_2 = 1$ in the interval $[-1, 1]$ are mapped to

$$x_0 = a, \quad x_1 = \frac{a+b}{2}, \quad x_2 = b.$$

Thus we obtain the quadrature

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) dt \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{b+a}{2}\right) + f(b) \right].$$

Simpson's rule over the interval $[a, b]$ becomes therefore

$$S(a, b) = \frac{b-a}{6} [f(a) + 4f(c) + f(b)], \quad \text{with} \quad c = \frac{b+a}{2}.$$

Composite Simpson's rule. Next we will have to discuss the corresponding composite rule. Here we have to choose a partition of the interval $[a, b]$ into sub-intervals, evaluate the quadrature on each of the sub-intervals, and finally add all results together. The final result will heavily rely on the choice of the sub-intervals, and we will discuss later an automated strategy for their construction. For now, we content ourselves with the simplest construction, where we take sub-intervals of equal lengths.

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

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{j=0}^{m-1} \int_{x_{2j}}^{x_{2j+2}} f(x)dx \\ &\approx \sum_{j=0}^{m-1} S(x_{2j}, x_{2j+2}) = \sum_{j=0}^{m-1} \frac{h}{3} [f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})] \\ &= \frac{h}{3} \left[f(x_0) + 4 \sum_{j=0}^{m-1} f(x_{2j+1}) + 2 \sum_{j=1}^{m-1} f(x_{2j}) + f(x_{2m}) \right] \end{aligned}$$

We will use the notation $S_m(a, b)$ for the composite Simpson's rule on m subintervals.

Implementation and testing. It is now time to implement the composite Simpson's method, and see how well it works.

See the function `simpson` in the file `quadrature.py`.

The first thing to do is to test if the code is correct. We know that Simpson's rule has precision 3, thus all third degree polynomials can be integrated exactly. Choose one such polynomial, find the exact integral, and compare.

Numerical experiment 1: Apply the code on the integral, and compare with the exact result.

$$\int_{-1}^2 (4x^3 + x^2 + 2x - 1)dx = 18.$$

Comment: If you have no clue of the precision of your scheme, repeat the experiment on $\int_a^b x^j$ for $j = 0, 1, 2, 3, \dots$

Numerical experiment 2: We will assume that the error decreases when the number of subintervals m increases. But how much?

Apply the composite method on the integral (again with a known solution):

$$\int_0^1 \cos\left(\frac{\pi x}{2}\right) dx = \frac{2}{\pi}.$$

Use the function 'simpson' with $m = 1, 2, 4, 8, 16$ and see how the error changes with m . Comment on the result.

From the experiment we observe that the error is reduced by a factor approximately $0.0625 = 1/16$ whenever the number of subintervals increases with a factor 2. In the following, we will prove that this is in fact what can be expected.

3.2 Error analysis of Simpson's rule

First we will find an expression for the error $E(a, b) = I(a, b) - S(a, b)$ over one interval (a, b) . This will then be used to find an expression for the composite formula.

Let $c = (a + b)/2$ be the midpoint of the interval, and $h = (b - a)/2$ be the distance between c and the endpoints a and b . Do a Taylor series expansion of the integrand f around the midpoint, and integrate each term in the series.

$$\begin{aligned}\int_a^b f(x)dx &= \int_{-h}^h f(c+s)ds = \int_{-h}^h \left(f(c) + sf'(c) + \frac{1}{2}s^2 f''(c) + \frac{1}{6}s^3 f'''(c) + \frac{1}{24}s^4 f^{(4)}(c) + \dots \right) ds \\ &= 2hf(c) + \frac{h^3}{3}f''(c) + \frac{h^5}{60}f^{(4)}(c) + \dots\end{aligned}$$

Similarly, do a Taylor series expansion of the quadrature $S(a, b)$ around c :

$$\begin{aligned}S(a, b) &= \frac{h}{3} (f(c-h) + 4f(c) + f(c+h)) \\ &= \frac{h}{3} \left(f(c) - hf'(c) + \frac{1}{2}h^2 f''(c) - \frac{1}{6}h^3 f'''(c) + \frac{1}{24}h^4 f^{(4)}(c) + \dots \right. \\ &\quad \left. + 4f(c) \right. \\ &\quad \left. + f(c) + hf'(c) + \frac{1}{2}h^2 f''(c) + \frac{1}{6}h^3 f'''(c) + \frac{1}{24}h^4 f^{(4)}(c) + \dots \right) \\ &= 2hf(c) + \frac{h^3}{3}f''(c) + \frac{h^5}{36}f^{(4)}(c) + \dots\end{aligned}$$

The series expansion of the error becomes:

$$E(a, b) = \int_a^b f(x)dx - S(a, b) = -\frac{h^5}{90}f^{(4)}(c) + \dots = -\frac{(b-a)^5}{2^5 \cdot 90}f^{(4)}(c) + \dots,$$

using $h = (b - a)/2$.

NB! By choosing to do the Taylor-expansions around the midpoint, every second term disappear thanks to symmetry. Choosing another point \hat{c} in the interval will give the same dominant error term (with c replaced by \hat{c}), but the calculations will be much more cumbersome.

Usually, we will assume h to be small, such that the first nonzero term in the series dominates the error, and the rest of the series can be ignored. The precise statement about the error is given in the following theorem, the full proof is unfortunately far from trivial and considered outside the present scope.

Theorem: Error in Simpson's method.

Let $f(x) \in C^4[a, b]$. There exists a $\xi \in (a, b)$ such that

$$E(a, b) = \int_a^b f(x)dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{b+a}{2}\right) + f(b) \right] = -\frac{(b-a)^5}{2880}f^{(4)}(\xi).$$

NB!: Since $p^{(4)}(x) = 0$ for all $p \in \mathbb{P}_3$ the degree of precision is 3.

Use the theorem to find an expression for the error in the composite Simpson's formula $S_m(a, b)$:

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

where $\xi_j \in (x_{2j}, x_{2j+2})$. We can then use the generalized mean value theorem, see *Preliminaries*, section 5. According to this, there is a $\xi \in (a, b)$ such that

$$\sum_{j=0}^{m-1} f^{(4)}(\xi_j) = m f^{(4)}(\xi).$$

Use $2mh = b - a$, and the following theorem has been proved:

Theorem: Error in composite Simpson's method.

Let $f(x) \in C^4[a, b]$. There exists a $\xi \in (a, b)$ such that

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

Example 4: Find the upper bound for the error when the composite Simpson's rule is applied to the integral $\int_0^1 \cos(\pi x/2) dx$.

In this case $f^{(4)}(x) = (\pi^4/16) \cos(\pi x/2)$, so that $|f^{(4)}(x)| \leq (\pi/2)^4$. The error bound becomes

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

If m is increased by a factor 2, the error will be reduced by a factor of $1/16$, as indicated by Numerical experiment 2.

Numerical exercise: Include the error bound in the output of Numerical experiment 2, and confirm that it really holds.

Remark: The result above shows that the composite Simpson rule with equidistant nodes converges with convergence order 4 (in terms of the node distance h) to the actual integral. That is, the convergence order is equal to the degree of precision + 1. This relation between degree of precision and convergence order can be shown to hold in general: If a composite quadrature rule with equidistant nodes is based on a simple quadrature rule, as constructed above, with degree of precision p , then the composite rule will have convergence order $p + 1$ for all functions $f \in C^{p+1}[a, b]$.

3.3 Trapezoidal and midpoint rule

Without going into the details of the respective derivations, we will now summarise the results one obtains for the trapezoidal and the midpoint rule.

Trapezoidal rule. On the standard interval $[-1, 1]$, the trapezoidal rule T is given as

$$T(-1, 1) = f(-1) + f(1).$$

That is, we have the nodes and weights

$$\begin{aligned} t_0 &= -1, & t_1 &= 1, \\ w_0 &= 1, & w_1 &= 1. \end{aligned}$$

In order to transfer the rule to an arbitrary interval $[a, b]$, we again apply the transformation

$$x = \frac{a+b}{2} + \frac{b-a}{2}t \quad \text{with } dx = \frac{b-a}{2}dt.$$

This yields the new nodes $x_0 = a$ and $x_1 = b$, and the quadrature

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right) dt \approx \frac{b-a}{2} (f(a) + f(b)).$$

This gives us the (well known) formula for the trapezoidal rule on an arbitrary interval,

$$T(a, b) = \frac{b-a}{2}(f(a) + f(b)).$$

Next, we find the composite formula for the trapezoidal rule, given a partition of the interval $[a, b]$. As for Simpson's rule, the result depends on the type of partition we are choosing, the simplest case, again, being that of a uniform partition of $[a, b]$. For that, we divide the interval into m sub-intervals $[X_j, X_{j+1}]$ of equal length $h = (b-a)/m$. That is, we have $X_j = a + jh$. If we then apply the (simple) trapezoidal rule on each subinterval, we obtain the approximation

$$\int_a^b f(x) dx \approx \sum_{j=0}^{m-1} T(X_j, X_{j+1}) = \sum_{j=0}^{m-1} \frac{h}{2} [f(X_j) + f(X_{j+1})] = h \left[\frac{1}{2} f(a) + \sum_{j=1}^{m-1} f(X_j) + \frac{1}{2} f(b) \right].$$

The next step would be the error analysis. Because this is somehow lengthy and does not offer any new insight, it being along the same lines as the analysis of Simpson's rule, we will only state the final results:

Theorem: Error in the trapezoidal rule.

Let $f(x) \in C^2[a, b]$. There exists a $\xi \in (a, b)$ such that

$$E(a, b) = \int_a^b f(x) dx - \frac{b-a}{2}(f(a) + f(b)) = -\frac{(b-a)^3}{12} f''(\xi).$$

Theorem: Error in the composite trapezoidal rule.

Let $f(x) \in C^2[a, b]$. There exists a $\xi \in (a, b)$ such that

$$\int_a^b f(x) dx - T_m(a, b) = -\frac{(b-a)h^2}{12} f''(\xi).$$

Midpoint rule. On the standard interval $[-1, 1]$, the midpoint rule M is given as

$$M(-1, 1) = 2f(0).$$

That is, we have only a single node and single weight

$$t_0 = 0 \quad \text{and} \quad w_0 = 2.$$

The transfer to an arbitrary interval $[a, b]$ works the same as for the other rules, and results in the new node $x_0 = (a+b)/2$ and the quadrature

$$M(a, b) = (b-a)f\left(\frac{a+b}{2}\right).$$

For the composite formula, we, again, only derive an explicit formula for the simplest case of a uniform partition of $[a, b]$. Setting $X_j = a + jh$ with $h = (b-a)/m$, we obtain the formula

$$\int_a^b f(x) dx \approx \sum_{j=0}^{m-1} M(X_j, X_{j+1}) = \frac{b-a}{m} \sum_{j=0}^{m-1} f\left(\frac{X_j + X_{j+1}}{2}\right) =: M_m(a, b).$$

With the abbreviation

$$X_{j+1/2} := \frac{X_j + X_{j+1}}{2},$$

we obtain the simpler form

$$M_m(a, b) = \frac{b-a}{m} \sum_{j=0}^{m-1} f(X_{j+1/2}).$$

Even though the midpoint rule only uses a single node, it turns out to have the same degree of precision as the trapezoidal rule that uses two nodes. In fact, the error for the midpoint rule is actually smaller than that for the trapezoidal rule, as the following theorems show:

Theorem: Error in the midpoint rule.

Let $f(x) \in C^2[a, b]$. There exists a $\xi \in (a, b)$ such that

$$E(a, b) = \int_a^b f(x)dx - (b-a)f\left(\frac{b-a}{2}\right) = \frac{(b-a)^3}{24} f''(\xi).$$

Theorem: Error in the composite midpoint rule.

Let $f(x) \in C^2[a, b]$. There exists a $\xi \in (a, b)$ such that

$$\int_a^b f(x)dx - M_m(a, b) = \frac{(b-a)h^2}{24} f''(\xi).$$

From these results, it appears as if the midpoint rule were preferable over the trapezoidal rule, since it yields a more accurate result while requiring fewer computations. Actual implementations, however, rarely if ever use a uniform partition of the interval $[a, b]$ for the computation of the composite rule, but rather use some type of adaptive strategy, as described in the next chapter. For such adaptive strategies, the trapezoidal rule can be easier to use and also faster, as many calculations, and in particular evaluations of the function f , can be reused. For the midpoint rule, this requires a different adaptation strategy, which may be less efficient in practice.

4 Practical quadrature rules

4.1 Error estimation

From a practical point of view, the error expressions derived in the previous chapter have some limitations, the main difficulty being that they depend on the value of some derivative of f at an unknown point $\xi \in (a, b)$; for Simpson's rule, for instance, we require knowledge of $f^{(4)}(\xi)$. In practice, we can thus at best use an error estimate of the form

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

where $\|f^{(4)}\|_\infty = \max_{x \in [a, b]} |f^{(4)}(x)|$. This bound, however, often vastly overestimates the actual error. In addition, we do not always know (or want to find) $\|f^{(4)}\|_\infty$. So the question arises: How can we find an estimate of the error, without any extra analytical calculations?

Again, we discuss specifically the case of Simpson's rule. For the other methods, the argumentation is similar, but the actual formulas will be slightly different, depending on the convergence order of the quadrature rule one uses.

Let the interval (a, b) chosen small, such that $f^{(4)}(x)$ can be assumed to be almost constant over the interval. Let $H = b - a$ be the length of the interval. Let $S_1(a, b)$ and $S_2(a, b)$ be the results from Simpson's formula over one and two subintervals respectively. Further, let $C = -f^{(4)}(x)/2880$ for some $x \in [a, b]$, which x does not matter since $f^{(4)}$ is assumed almost constant anyway. The errors of the two approximations are then given by

$$I(a, b) - S_1(a, b) \approx CH^5,$$

$$I(a, b) - S_2(a, b) \approx 2C \left(\frac{H}{2} \right)^5.$$

Subtract the two expressions to eliminate $I(a, b)$:

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

Insert this in the expression for the error:

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

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

This gives us a computable estimate for the error, both in S_1 and S_2 . As the error in $S_2(a, b)$ is about 1/16 of the error in $S_1(a, b)$, and we anyway need to compute both, we will use $S_2(a, b)$ as our approximation. An even better approximation to the integral is given for free by just adding the error estimate,

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

Example 5: Find an approximation to the integral $\int_0^1 \cos(x)dx = \sin(1)$ by Simpson's rule over one and two subintervals. Find the error estimates \mathcal{E}_m , $m = 1, 2$ and compare with the exact error.

Solution:

$$S_1(0, 1) = \frac{1}{6} [\cos(0.0) + 4 \cos(0.5) + \cos(1.0)] = 0.8417720923$$

$$S_2(0, 1) = \frac{1}{12} [\cos(0.0) + 4 \cos(0.25) + 2 \cos(0.5) + 4 \cos(0.75) + \cos(1.0)] = 0.8414893826$$

The exact error and the error estimate become:

$$E_1(0, 1) = \sin(1) - S_1(0, 1) = -3.011 \cdot 10^{-4},$$

$$\mathcal{E}_1(0, 1) = \frac{16}{15}(S_2 - S_1) = -3.016 \cdot 10^{-4},$$

$$E_2(0, 1) = \sin(1) - S_2(0, 1) = -1.840 \cdot 10^{-5},$$

$$\mathcal{E}_2(0, 1) = \frac{1}{16}(S_2 - S_1) = -1.885 \cdot 10^{-5}.$$

In this case, it is a very good correspondence between the error estimate and the exact error. An even better approximation is obtained by adding the error estimate to S_2 :

$$Q = S_2(0, 1) + \mathcal{E}_2(0, 1) = 0.8414705353607151$$

with an error $\sin(1) - Q = 4.4945 \cdot 10^{-7}$. This gives a lot of additional accuracy without any extra work.

Implementation of Simpson's method with an error estimate. The function `simpson_basic` returns

$$S_2(a, b) \approx \int_a^b f(x) dx$$

including an error estimate.

Test: As a first check of the implementation, use the example above, and make sure that the results are the same:

Next, let us see how reliable the quadrature and the error estimates are for another example, which you have to do yourself:

Numerical experiment 3: Given the integral (with solution)

$$I(a, b) = \int_a^b \frac{1}{1 + 16x^2} dx = \left. \frac{\arctan(4x)}{4} \right|_a^b$$

1. Use `simsn_basic` to find an approximation to the integral over the interval $[0, 8]$. Print out $S_2(0, 8)$, the error estimate $\mathcal{E}_2(0, 8)$ and the real error $E_2(0, 8)$. How reliable are the error estimates?
2. Repeat the experiment over the intervals $[0, 1]$ and $[4, 8]$. Notice the difference between exact error of the two intervals.
3. Repeat the experiment over the interval $[0, 0.1]$.

This is what you should observe from the experiment:

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

Why is it so, and how can we deal with it? Obviously, we need small subintervals near $x = 0$, while large subintervals are acceptable in the last half of the interval.

Explanation: The error in Simpson's method is given by

$$E(a, b) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi).$$

So let us take a look at $f^{(4)}(x)$:

$$f(x) = \frac{1}{1 + 16x^2} \quad \Rightarrow \quad f^{(4)}(x) = 6144 \frac{1280x^4 - 160x^2 + 1}{(1 + 16x^2)^5}$$

It is no surprise that the error is large and the error estimates fail (we have assumed $f^{(4)}$ almost constant for the estimates) over the interval $[0, 1]$. The part of the interval where $f^{(4)}(x)$ is large has to be partitioned in significantly smaller subintervals to get an acceptable result. But how, as $f^{(4)}$ is in general not known? This is the topic of the next section.

4.2 Adaptive integration

Given a basic function, for example `simpson_basic`, returning an approximation $Q(a, b)$ to the integral, as well as an error estimate $\mathcal{E}(a, b)$. Based on this, we want to find a partitioning of the interval:

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

such that

$$|\mathcal{E}(X_j, X_{j+1})| \approx \frac{X_{k+1} - X_k}{b - a} \cdot \text{Tol}$$

where Tol is a tolerance given by the user. In this case

$$\text{Accumulated error over } (a, b) \approx \sum_{j=0}^{m-1} \mathcal{E}(X_j, X_{j+1}) \leq \text{Tol}.$$

Such a partitioning can be done by a recursive algorithm:

Algorithm: Adaptive quadrature.

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

- Calculate $Q(a, b)$ and $\mathcal{E}(a, b)$.
- **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**:
 Let $c = (a + b)/2$, and repeat the process on each of the subintervals $[a, c]$ and $[c, b]$, with tolerance Tol/2.
- Sum up the accepted results from each subinterval.

Implementation. The adaptive algorithm is implemented below with `simpson_basic` as the basic quadrature routine. The function `simpson_adaptive` is a recursive function, that is a function that calls itself. To avoid it to do so infinitely many times, an extra variable `level` is introduced, this will increase by one for each time the function calls itself. If `level` is over some maximum value, the result is returned, and a warning printed.

Numerical experiment 4: Use adaptive Simpson to find an approximation to the integral $\int_0^5 1/(1 + 16x^2)dx$ using the tolerances Tol= $10^{-3}, 10^{-5}, 10^{-7}$. Compare the numerical result with the exact one.

5 Other quadrature formulas

Simpson's rule, the trapezoidal rule, and the midpoint rule are only three examples of quadrature rules derived from polynomial interpolation. There are many others, and the whole process of deriving the methods, doing the error analysis, developing error estimates and finally implementing adaptive algorithms can be repeated.

We will conclude this note with a few other popular classes of methods:

Newton-Cotes formulas. These are based on equidistributed nodes. The simplest choices here — the *closed* Newton-Cotes methods — use the nodes $x_i = a + ih$ with $h = (b - a)/n$. Examples of these are the Trapezoidal rule and Simpson's rule. The main appeal of these rules is the simple definition of the nodes.

If n is odd, the Newton-Cotes method with $n + 1$ nodes has degree of precision n ; if n is even, it has degree of precision $n + 1$. The corresponding convergence order for the composite rule is, as for all such rules, one larger than the degree of precision, provided that the function f is sufficiently smooth.

However, for $n \geq 8$ negative weights begin to appear in the definitions. This has the undesired effect that the numerical integral of a positive function can be negative. In addition, this can lead to cancellation errors in the numerical evaluation, which may result in a lower practical accuracy. Since the rules with $n = 6$ and $n = 7$ yield the same convergence order, this means that it is mostly the rules with $n \leq 6$ that are used in practice.

The *open* Newton-Cotes methods, in contrast, use the nodes $x_i = a + (i + 1/2)h$ with $h = (b - a)/(n + 1)$. The simplest example here is the midpoint rule. Here negative weights appear already for $n \geq 2$. Thus the midpoint rule is the only such rule that is commonly used in applications.

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

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

The resulting quadrature rules have a degree of precision $d = 2n - 1$, and the corresponding composite rules have a convergence order of $2n$. It is possible to show that this is the highest achievable degree of precision with n nodes.

For $n = 1$, one obtains (again) the midpoint rule. For $n = 2$ one obtains the method discussed near the beginning of these notes in example 2, which, on the interval $[-1, 1]$, has the nodes $x_0 = -\sqrt{3}/3$ and $x_1 = +\sqrt{3}/3$ and the corresponding weights $w_0 = w_1 = 1$.