Numerical Integration

1 Introduction

We want to approximate the integral

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

where we are given a, b and the function f as a subroutine.

We evaluate f at points $x_1, \ldots, x_n$ and construct out of the function values an approximation Q. We want to have a small quadrature error $|Q-I|$ using as few function evaluations as possible.

We can do this using interpolation:

- construct the interpolating polynomial $p(x)$
- let $Q := \int_a^b p(x)dx$
- by writing Q in terms of the function values we obtain a quadrature rule of the form

$$Q = w_1 f(x_1) + \cdots + w_n f(x_n)$$

In the special case that the function $f(x)$ is a polynomial of degree$\leq n-1$ we obtain $p(x) = f(x)$ since the interpolating polynomial is unique, and hence $Q = I$. Therefore the quadrature rule is exact for all polynomials of degree$\leq n-1$.

2 Midpoint Rule, Trapezoid Rule, Simpson Rule

We consider some special cases with $n = 1, 2, 3$ points:

Midpoint Rule: Let $n = 1$ and pick the midpoint $x_1 := (a+b)/2$. Then $p(x) = f(x_1)$ (constant function) and

$$\boxed{Q^{\text{Midpt}} = (b-a)f(x_1)}$$

Trapezoid Rule: Let $n = 2$ and pick the endpoints: $x_1 := a$, $x_2 := b$. Then $p(x)$ is a linear function and Q is the area of the trapezoid:

$$\boxed{Q^{\text{Trap}} = (b-a)\frac{f(a)+f(b)}{2}}$$

Simpson Rule: Let $n = 3$ and pick the endpoints and midpoint: $x_1 := a$, $x_2 := (a+b)/2$, $x_3 := b$. Then $p(x)$ is a quadratic function and we obtain

$$\boxed{Q^{\text{Simpson}} = (b-a)\frac{f(x_1)+4f(x_2)+f(x_3)}{6}.}$$

Proof: Let us consider the interval $[a,b] = [-r,r]$ where $r = (b-a)/2$. We know that

$$Q = \int_a^b p(x)dx = w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3)$$

and we want to find w_1, w_2, w_3. We also know that we must have $Q = I$ for $f(x) = 1$, $f(x) = x$, $f(x) = x^2$ yielding the equations

$$\begin{aligned} w_1 \cdot 1 + w_2 \cdot 1 + w_3 \cdot 1 &= \int_{-r}^{r} 1dx = 2r \\ w_1 \cdot (-r) + w_2 \cdot 0 + w_3 \cdot r &= \int_{-r}^{r} x\,dx = 0 \\ w_1 \cdot r^2 + w_2 \cdot 0 + w_3 \cdot r^2 &= \int_{-r}^{r} x^2 dx = \tfrac{2}{3}r^3 \end{aligned}$$

Solving this system for w_1, w_2, w_3 yields $w_1 = w_3 = \frac{r}{3}$, $w_2 = \frac{4}{3}r$.

The midpoint rule is guaranteed to be exact for polynomials of degree 0. But actually it is also exact for all polynomials of degree 1: On the interval $[-r,r]$ consider $f(x) = c_0 + c_1 x$. Then the term c_0 is exactly integrated by the midpoint rule. For the term $c_1 \cdot x$ the exact integral is zero, and the midpoint rule also gives zero for this term.

The Simpson rule is guaranteed to be exact for polynomials of degree≤ 2. But actually it is also exact for all polynomials of degree≤ 3: On the interval $[-r,r]$ consider $f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$. Then the term $c_0 + c_1 x + c_2 x^2$ is exactly integrated by the Simpson rule. For the term $c_3 \cdot x^3$ the exact integral is zero, and the Simpson rule also gives zero for this term.

2.1 Errors for the Midpoint Rule, Trapezoid Rule, Simpson Rule

Note that we have for the quadrature error

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

and we know for the interpolating polynomial that

$$|f(x) - p(x)| \leq \frac{1}{n!} \left(\max_{t\in[a,b]} \left| f^{(n)}(x) \right| \right) |(x-x_1)\cdots(x-x_n)|$$

yielding

$$|I - Q| \leq \frac{1}{n!} \left(\max_{t\in[a,b]} \left| f^{(n)}(x) \right| \right) \cdot \int_a^b |(x-x_1)\cdots(x-x_n)|\, dx. \tag{1}$$

Error for Trapezoid Rule: Here we need to compute $\int_a^b |(x-a)(x-b)|\, dx$. Let us consider the interval $[a,b] = [-r,r]$:

$$\int_a^b |(x-a)(x-b)|\, dx = \int_{-r}^r |(x+r)(x-r)|\, dx = \int_{-r}^r (r^2 - x^2) dx = \left[r^2 x - \tfrac{1}{3} x^3 \right]_{-r}^r = \frac{4}{3} r^3$$

As $r = (b-a)/2$ and $n = 2$ the formula (1) becomes

$$\boxed{\left| I - Q^{\text{Trap}} \right| \leq \frac{(b-a)^3}{12} \cdot \max_{t\in[a,b]} \left| f''(x) \right|}$$

Error for Midpoint Rule: We want to exploit that the Midpoint Rule is exact for polynomials of degree 1 and consider the interpolating polynomial $\tilde{p}(x)$ which interpolates f at the nodes x_1, x_1 (which is the tangent line):

$$\tilde{p}(x) = f[x_0] + f[x_0, x_0](x - x_0) = p(x) + f[x_0, x_0](x - x_0)$$
$$\int_a^b \tilde{p}(x) dx = \int_a^b p(x) dx + f[x_0, x_0] \cdot \int_a^b (x - x_0) dx = Q + 0$$

Hence we have using the interpolation error for $\tilde{p}(x)$

$$|I - Q| = \left| \int_a^b (f(x) - \tilde{p}(x))\, dx \right| \leq \frac{1}{2!} \left(\max_{t\in[a,b]} \left| f''(x) \right| \right) \cdot \underbrace{\int_a^b |(x-x_1)(x-x_1)|\, dx}_{\left[\frac{1}{3}(x-x_1)^3\right]_a^b = \frac{2}{3}\left(\frac{b-a}{2}\right)^3}$$

yielding

$$\boxed{\left| I - Q^{\text{Midpt}} \right| \leq \frac{(b-a)^3}{24} \cdot \max_{t\in[a,b]} \left| f''(x) \right|}$$

Error for Simpson Rule: We want to exploit that the Simpson Rule is exact for polynomials of degree 3 and consider the interpolating polynomial $\tilde{p}(x)$ which interpolates f at the nodes x_1, x_2, x_3, x_2 (which also has the correct slope in the midpoint):

$$\tilde{p}(x) = p(x) + f[x_1, x_2, x_3, x_2](x-x_1)(x-x_2)(x-x_3)$$
$$\int_a^b \tilde{p}(x)dx = \int_a^b p(x)dx + f[x_1, x_2, x_3, x_2] \cdot \int_a^b (x-x_1)(x-x_2)(x-x_3)dx = Q + 0$$

since the function $(x-x_1)(x-x_2)(x-x_3)$ is antisymmetric with respect to the midpoint x_2. Hence we have using the interpolation error for $\tilde{p}(x)$

$$|I-Q| = \left| \int_a^b (f(x) - \tilde{p}(x))\, dx \right| \le \frac{1}{4!} \left(\max_{t\in[a,b]} \left| f^{(4)}(x) \right| \right) \cdot \int_a^b \left| (x-x_1)(x-x_2)^2(x-x_3) \right| dx.$$

We consider the interval $[a,b] = [-r,r]$ with $r = (b-a)/2$. Then we have for the integral

$$\int_a^b \left| (x-x_1)(x-x_2)^2(x-x_3) \right| dx = \int_{-r}^r \left| (x+r)x^2(x-r) \right| dx = \int_{-r}^r (r^2 - x^2)x^2\, dx = \left[r^2 \frac{x^3}{3} - \frac{r^5}{5} \right]_{-r}^r = \frac{4}{15} r^5$$

yielding

$$\boxed{\left| I - Q^{\text{Simpson}} \right| \le \frac{(b-a)^5}{90 \cdot 32} \cdot \max_{t\in[a,b]} \left| f^{(4)}(x) \right|.}$$

2.2 Higher Order Rules

For given nodes $x_1, \ldots, x_n$ we can construct a quadrature rule $Q = w_1 f(x_1) + \cdots + w_n f(x_n)$ with an interpolating polynomial of degree $n-1$. Using the method from the Simpson rule we can find the weights $w_1, \ldots, w_n$ by solving a linear system.

For equidistant nodes this gives for $n = 9$ nodes weights w_j which are alternatingly positive and negative, and for larger values of n the size of the weights w_j increases exponentially: For $[0,1]$ we get

n	15	25	35	45
$\sum_{j=1}^n \lvert w_j \rvert$	$2.0 \cdot 10^1$	$5.6 \cdot 10^3$	$2.5 \cdot 10^6$	$1.4 \cdot 10^9$

This means that in machine arithmetic there will be substantial subtractive cancellation. The reason for the negative weights is that interpolating polynomials of large degree tend to have very large oscillations, as we saw earlier.

For interpolation we have seen that one can avoid these problems by carefully placing the nodes in a nonuniform way so that they are more closely clustered together at the endpoints. For interpolation a good choice are the so-called Chebyshev nodes (which are the zeros of Chebyshev polynomials).

This choice of nodes is also useful for numerical integration. Instead of the zeros of Chebyshev polynomials one can also choose the extrema of Chebyshev polynomials, and in this case there is an efficient algorithm to compute Q (*Clenshaw-Curtis quadrature*).

Another choice are the so-called Gauss nodes for *Gaussian quadrature*, see section 5 below. These nodes are also more closely clustered near the endpoints, but they are chosen to maximize the polynomial degree for which the rule is exact.

3 Composite Rules

For a practical integration problem it is better to increase the accuracy by first subdividing the interval into smaller subintervals with a partition

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$$

and interval sizes

$$h_j := x_j - x_{j-1}.$$

Then we apply one of the basic rules (midpoint, trapezoid or Simpson rule) on each subinterval and add everything together. This is called a *composite rule*. For example, the **composite trapezoid rule** is defined by

$$\boxed{Q_N^{\text{Trap}} := Q_{[x_0,x_1]}^{\text{Trap}} + \cdots + Q_{[x_{N-1},x_N]}^{\text{Trap}}}$$

where $Q_{[x_{j-1},x_j]}^{\text{Trap}} = h_j \frac{1}{2}\left(f(x_{j-1}) + f(x_j)\right)$. Similarly we can define the composite midpoint rule and the composite Simpson rule.

Work: For the composite trapezoid rule with N subintervals we use $N+1$ function evaluations.
For the composite midpoint rule with N subintervals we use N function evaluations.
For the composite Simpson rule with N subintervals we use $2N+1$ function evaluations.

3.1 Error for Composite Rules

The error of the composite trapezoid rule is the sum of the errors on each subinterval:

$$I - Q_N^{\text{Trap}} = \sum_{j=1}^{N} \left(I_{[x_{j-1},x_j]} - Q_{[x_{j-1},x_j]}^{\text{Trap}} \right) \leq \sum_{j=1}^{N} \left| I_{[x_{j-1},x_j]} - Q_{[x_{j-1},x_j]}^{\text{Trap}} \right|$$

$$\left| I - Q_N^{\text{Trap}} \right| \leq \sum_{j=1}^{N} \left| I_{[x_{j-1},x_j]} - Q_{[x_{j-1},x_j]}^{\text{Trap}} \right| \leq \frac{1}{12} \sum_{j=1}^{N} \left(\max_{[x_{j-1},x_j]} \left| f''(t) \right| \right) h_j^3$$

Similarly we can obtain estimates for the composite midpoint rule and the composite Simpson rule.

3.2 Subintervals of equal size

The simplest choice is to choose all subintervals of the same size $h = (b-a)/N$. In this case we obtain for the **composite trapezoid rule**

$$\left| I - Q_N^{\text{Trap}} \right| \leq \sum_{j=1}^{N} \frac{1}{12} \left(\max_{[x_{j-1},x_j]} \left| f''(t) \right| \right) h^3 \leq \frac{1}{12} \left(\max_{[a,b]} \left| f''(t) \right| \right) h^3 \left(\sum_{j=1}^{N} 1 \right)$$

$$\boxed{\left| I - Q_N^{\text{Trap}} \right| \leq \frac{1}{12} \cdot \frac{(b-a)^3}{N^2} \cdot \max_{[a,b]} \left| f''(t) \right|}$$

If $f''(x)$ is continuous for $x \in [a,b]$ we therefore obtain with $C = \frac{(b-a)^3}{12} \cdot \max_{[a,b]} |f''(t)|$ that

$$\left| I - Q_N^{\text{Trap}} \right| \leq \frac{C}{N^2}.$$

This shows that the error tends to zero as $N \to \infty$.

Composite midpoint rule: If $f''(x)$ is continuous for $x \in [a,b]$ we obtain in the same way

$$\boxed{\left| I - Q_N^{\text{Midpt}} \right| \leq \frac{1}{24} \cdot \frac{(b-a)^3}{N^2} \cdot \max_{[a,b]} \left| f''(t) \right|}$$

where we also have $\left| I - Q_N^{\text{Midpt}} \right| \leq \frac{C}{N^2}$.

Composite Simpson rule: If $f^{(4)}(x)$ is continuous for $x \in [a,b]$ we obtain in the same way

$$\boxed{\left| I - Q_N^{\text{Simpson}} \right| \leq \frac{1}{90 \cdot 32} \cdot \frac{(b-a)^5}{N^4} \cdot \max_{[a,b]} \left| f^{(4)}(t) \right|}$$

In this case we have $\left| I - Q_N^{\text{Simpson}} \right| \leq \frac{C}{N^4}$, so the composite Simpson rule will converge faster than the composite trapezoid or midpoint rule.

3.3 If we only know that $f(x)$ is continuous

What happens if $f(x)$ is not smooth enough, i.e., there does not exist a continuous second derivative $f''(x)$ on $[a,b]$?

Assume that $f(x)$ is continuous on $[a,b]$. Then we know from calculus that we obtain the integral as the limit of Riemann sums: Define a subdivision $a = x_0 < x_1 < \cdots < x_N = b$ with maximal interval size $d_N := \max_{j=1,\ldots,N} |x_j - x_{j-1}|$. Then pick points $t_j \in [x_{j-1}, x_j]$ in each subinterval and define the **Riemann sum**

$$R_N := \sum_{j=1}^{N} f(t_j)(x_j - x_{j-1})$$

If we use a sequence of subdivisions with $d_N \to 0$ we have $R_N \to I$ as $N \to \infty$.

For a given subdivision define R_N^{left} as the Riemann sum where we use the left endpint $t_j := x_{j-1}$ of each subinterval. Let R_N^{right} denote the Riemann sum where we use the right endpoint $t_j := x_j$, and let R_N^{mid} denote the Riemann sum where we use the midpoint $t_j := \frac{1}{2}(x_{j-1} + x_j)$. Each of these Riemann sums converges to I for a sequence of subdivisions with $d_N \to 0$. Note that we have

$$Q_N^{\text{Midpt}} = R_N^{\text{mid}}, \qquad Q_N^{\text{Trap}} = \tfrac{1}{2}\left(R_N^{\text{left}} + R_N^{\text{right}}\right), \qquad Q_N^{\text{Simpson}} = \tfrac{1}{6}\left(R_N^{\text{left}} + 4R_N^{\text{mid}} + R_N^{\text{right}}\right)$$

and hence

$$Q_N^{\text{Midpt}} \to I, \qquad Q_N^{\text{Trap}} \to I, \qquad Q_N^{\text{Simpson}} \to I \qquad \text{for a sequence of subdivisions with } d_N \to 0.$$

3.4 Subintervals of different size, adaptive subdivision

For the compositive trapezoid rule Q_N^{Trap} the quadrature error

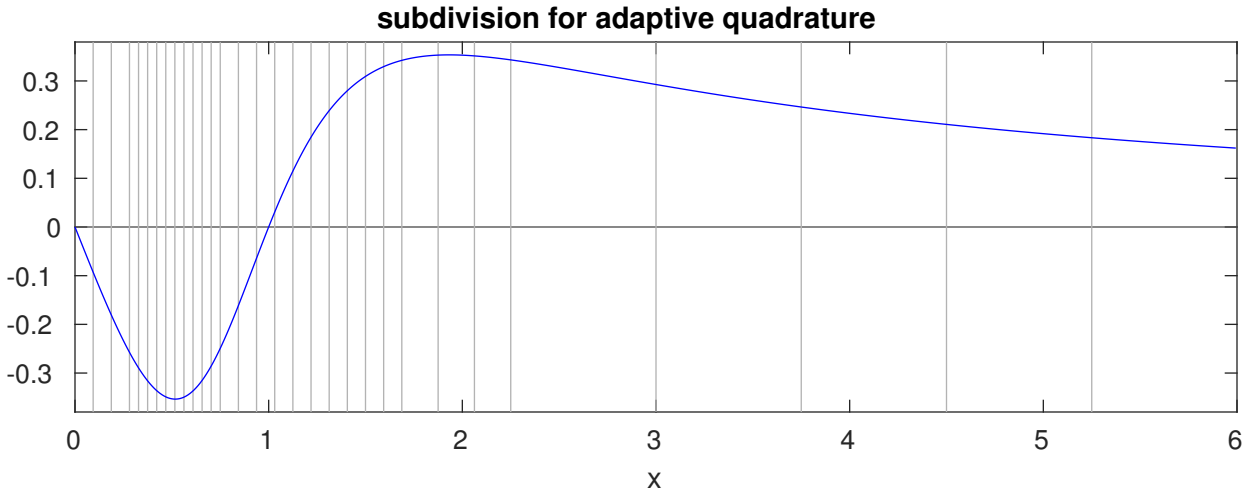
$$\left|I - Q_N^{\text{Trap}}\right| \le \sum_{j=1}^{N} \left|I_{[x_{j-1},x_j]} - Q_{[x_{j-1},x_j]}^{\text{Trap}}\right| \le \frac{1}{12} \sum_{j=1}^{N} \left(\max_{[x_{j-1},x_j]} |f''(t)|\right) h_j^3$$

depends on the size of the 2nd derivative $\max_{[x_{j-1},x_j]} |f''(t)|$ on each subinterval, multiplied by h_j^3 where h_j is the length of the subinterval. If $|f''(x)|$ is small in one part of $[a,b]$ we can use large interval sizes h_j there. If $|f''(x)|$ is large in another part of $[a,b]$ we should compensate for that with small interval sizes h_j there. This is called an **adaptive subdivision**.

Example: We want to find

$$I = \int_0^6 f(x)dx, \qquad \text{with } f(x) = \frac{x^3 - x}{1 + x^4} \tag{2}$$

Here $|f''|$ is small for $x > 3$, so we can use large subintervals. For smaller x and in particular close to $x = .5$ we have large values for $|f''|$, hence we should use smaller subintervals. An adaptive subdivision should look like this:



4 Adaptive quadrature

In practice we do not know $f''(x)$. We are only given a subroutine $f(x)$, the interval $[a,b]$ and a desired tolerance Tol (e.g., Tol $= 10^{-5}$) . We then want to find a subdivision $x_0 < x_1 < \cdots < x_N$ such that the composite trapezoid rule gives an approximation Q_N^{Trap} with $\left|Q_N^{\text{Trap}} - I\right| \leq$ Tol. How can we do this?

For an **adaptive algorithm** we need the following ingredients: **On each subinterval $[\alpha,\beta]$ of length $h := \beta - \alpha$ we need**

- **an approximation for the integral $I_{[\alpha,\beta]}$:** we use the trapezoid rule $Q_{[\alpha,\beta]}^{\text{Trap}} = \dfrac{h}{2}(f(\alpha)+f(\beta))$
- **an error estimate for $\left|Q_{[\alpha,\beta]}^{\text{Trap}} - I_{[\alpha,\beta]}\right|$:** Obviously we don't know the exact integral $I_{[\alpha,\beta]}$. But we can evaluate the function in the midpoint $\gamma = (\alpha+\beta)/2$ and find the Simpson rule approximation $Q_{[\alpha,\beta]}^{\text{Simp}} = \frac{h}{6}(f(\alpha)+4f(\gamma)+f(\beta))$. We know that $Q_{[\alpha,\beta]}^{\text{Trap}}$, $Q_{[\alpha,\beta]}^{\text{Simp}}$ satisfy

$$\left|Q_{[\alpha,\beta]}^{\text{Trap}} - I_{[\alpha,\beta]}\right| \leq \frac{h^3}{12}\max_{t\in[\alpha,\beta]}\left|f''(t)\right| = C_2 h^3, \qquad \left|Q_{[\alpha,\beta]}^{\text{Simp}} - I_{[\alpha,\beta]}\right| \leq \frac{h^5}{90\times 32}\max_{t\in[\alpha,\beta]}\left|f^{(4)}(t)\right| = C_4 h^5$$

  Clearly for small interval length h the Simpson approximation is much closer to $I_{[\alpha,\beta]}$ than the trapezoid approximation. Therefore we can approximate the error using

$$\left|Q_{[\alpha,\beta]}^{\text{Trap}} - I_{[\alpha,\beta]}\right| \approx \left|Q_{[\alpha,\beta]}^{\text{Trap}} - Q_{[\alpha,\beta]}^{\text{Simp}}\right|$$

  where the right hand side can be easily computed. This will be a good approximation for the error if the subinterval is small.

- **an accuracy goal $\left|Q_{[\alpha,\beta]}^{\text{Trap}} - I_{[\alpha,\beta]}\right| \leq \text{Tol}_{[\alpha,\beta]}$ for the subinterval:** On the whole interval we want an error $\left|Q_N^{\text{Trap}} - I\right| \leq$ Tol. Therefore it seems reasonable to require for the subinterval $[\alpha,\beta]$ a tolerance proportional to its length h: We want

$$\left|Q_{[\alpha,\beta]}^{\text{Trap}} - I_{[\alpha,\beta]}\right| \leq \frac{h}{b-a}\text{Tol} \tag{3}$$

  (e.g., for an subinterval of half the length we want half of the quadrature error).

We can implement these ideas using a **recursive Matlab function `Q=adaptint(f,a,b,Tol)`** as follows:

```
function Q = adaptint(f,a,b,Tol)
fa = f(a); fb = f(b);
QT = (b-a)/2*(fa+fb);          % Trapezoid rule
c = (a+b)/2; fc = f(c);        % evaluate f in midpoint
QS = (b-a)/6*(fa+4*fc+fb);     % Simpson rule

% for small intervals we can approximate error QT-I by QT-QS
if abs(QT-QS)<=Tol             % if estimated error is <= Tol
  Q = QT;                      %   accept trapezoid rule value
else
  Q = adaptint(f,a,c,Tol/2) + adaptint(f,c,b,Tol/2);
                               %   use algorithm for [a,c] and [c,b] with Tol/2
end
```

We save this as an m-file `adaptint.m`. Then we can approximate the integral in our example (2) using

```
>> f = @(x) (x^3-x)/(1+x^4)
>> Q = adaptint(f,0,6,1e-2)
Q =
          1.0214243535841
```

The actual error is $|Q-I| \approx 9.85 \cdot 10^{-4}$. Here `adaptint` uses the subdivision shown in the figure on page 5 with $N = 31$ subintervals. Note that we evaluate the function also in the midpoints of these intervals, so the total number of function evaluations needed is 63.

Remarks:

1. **The recursion will terminate: Assume that $f(x)$ is continuous on $[a,b]$.** Then for any given $\varepsilon > 0$ there exists $\delta > 0$ such that

$$s,t \in [a,b] \text{ with } |s-t| < \delta \implies |f(s)-f(t)| < \varepsilon \tag{4}$$

We are given a tolerance Tol. After k levels of recursion we have subintervals $[\alpha,\beta]$ of length $\beta-\alpha = 2^{-k}(b-a)$. We need to show that the condition

$$\left| Q^{\text{Trap}}_{[\alpha,\beta]} - Q^{\text{Simp}}_{[\alpha,\beta]} \right| \le \text{Tol} \cdot \frac{\beta-\alpha}{b-a} \tag{5}$$

is satisfied if k is sufficiently large. We have with $\gamma := \frac{1}{2}(\alpha+\beta)$

$$Q^{\text{Trap}}_{[\alpha,\beta]} - Q^{\text{Simp}}_{[\alpha,\beta]} = \tfrac{1}{3}(\beta-\alpha)\left([f(\alpha)-f(\gamma)]+[f(\beta)-f(\gamma)]\right)$$

Therefore (5) holds if we have

$$\left| Q^{\text{Trap}}_{[\alpha,\beta]} - Q^{\text{Simp}}_{[\alpha,\beta]} \right| \le \tfrac{1}{3}(\beta-\alpha)\left(|f(\alpha)-f(\gamma)|+|f(\beta)-f(\gamma)|\right) \overset{!}{\le} \text{Tol} \cdot \frac{\beta-\alpha}{b-a}$$

We now use $\varepsilon := \dfrac{3}{2} \cdot \dfrac{\text{Tol}}{b-a}$ and obtain a $\delta > 0$ such that (4) holds. Therefore (5) will hold if k satisfies $2^{-k}(b-a) < \delta$.

2. During the recursion the above code actually re-evaluates the already computed values `fa` and `fb` again. We can fix this using

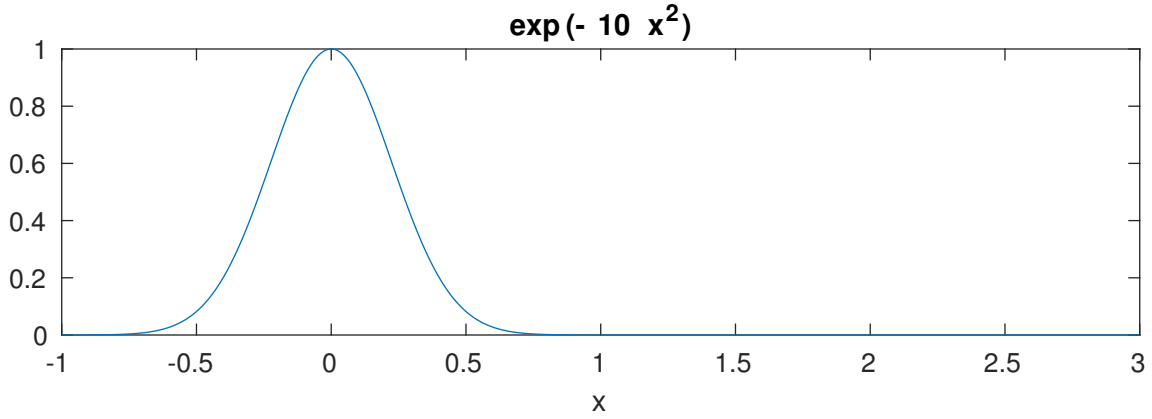
```
function Q = adaptint(f,a,b,Tol,fa,fb)
if nargin==4                % if function is called as adaptint(f,a,b,Tol)
  fa = f(a); fb = f(b);     %   compute fa,fb
end                         % otherwise we get fa,fb passed as arguments
...
  Q = adaptint(f,a,c,Tol/2,fa,fc) + adaptint(f,c,b,Tol/2,fc,fb);
end
```

3. The algorithm is based on the assumption that `QS` is a better approximation than `QT`. Hence we should get a more accurate result by **changing the line `Q = QT` to `Q = QS`**. For the above example we then get $Q = 1.02040470316526$ with the smaller error $|Q-I| \approx 3.47 \cdot 10^{-5}$. Note that we are now using a composite Simpson rule approximation for Q, but using a subdivision based on error bounds for the trapezoid rule. In order to obtain a good error estimate for the Simpson rule we would need additional function evaluations to compute a value $\tilde{Q}$ which is more precise than QS so that we can approximate $QS-I$ by $QS-\tilde{Q}$.

4. Note that **adaptive quadrature can give completely wrong results**:

```
f = @(x)exp(-10*x^2)
Q = adaptint(f,-1,3,1e-4)
Q =
      9.07998595249697e-05
```

The correct value is $I = \displaystyle\int_{-1}^{3} e^{-10x^2} dx \approx 0.560497$. What happened? The function $f(x)$ has a sharp peak near $x = 0$ and is almost zero outside of $[-.8,.8]$:



The adaptive quadrature evaluates for $a=-1$ and $b=3$ the values $f(-1)\approx 0$, $f(3)\approx 0$ yielding $QT\approx 0$. At the midpoint $c=1$ we get $f(1)\approx 0$, so also $QS\approx 0$ and hence $|QT-QS|\approx 6\cdot 10^{-5}$ which is less than our tolerance 10^{-4}. Hence our program accepts the trapezoid rule value QT, based on only three function values at $x=-1,1,3$, completely missing the peak near $x=0$.

A quadrature method can never guarantee that the error is less than the tolerance: The only information we have are finitely many function values, and the function could have some crazy behavior between those values.

If a function has features like sharp peaks or singularities which the quadrature may miss we can "help the quadrature method" by subdividing the integral: In our example we can split the integral at $x=0$:

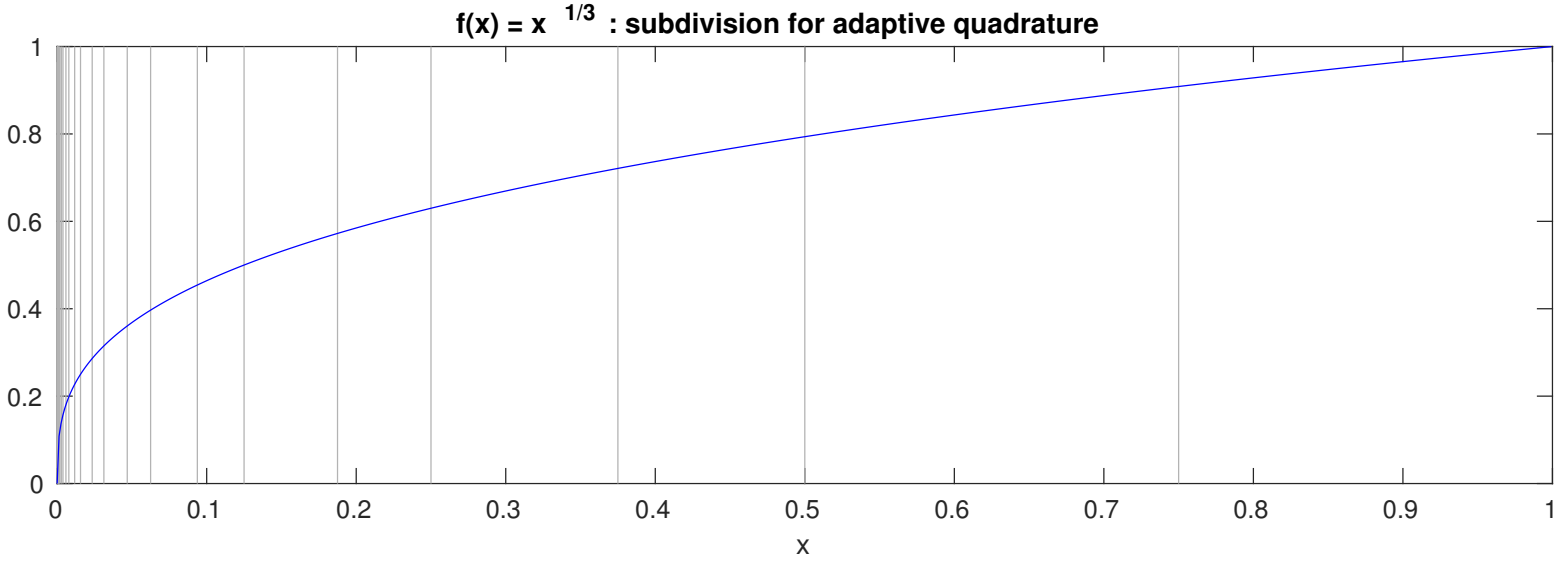
```
Q = adaptint(f,-1,0,1e-4) + adaptint(f,0,3,1e-4)
Q =
        0.560539838164273
```

which gives an error $|Q-I|\approx 4.3\cdot 10^{-5}$.

5. **Adaptive quadrature works efficiently even for functions with singularities**

We saw in section 3.2: For functions $f(x)$ where $f''(x)$ is continuous we can use subintervals of equal size, and the quadrature error decays like $\left|Q_N^{\text{Trap}}-I\right|\le \frac{c}{N^2}$.

For integrals like $I=\int_0^1 x^{1/3}dx$ this does not work since for $f(x)=x^{1/3}$ the 2nd derivative $f''(x)=-\frac{2}{9}x^{-5/3}$ becomes infinite near 0. We now use our adaptive algorithm:




```
f = @(x) x^(1/3);
Tol = 1e-2;                % given tolerance
Q = adaptint(f,0,1,Tol);
err = abs(Q - 3/4)         % quadrature error, exact integral is I=3/4
```

For `Tol=1e-2,1e-4,1e-6`, etc. we get for the number N of function evaluations and the error $|Q-I|$

Tol	N	$\lvert Q-I\rvert$
10^{-2}	29	$6.5\cdot 10^{-3}$
10^{-4}	243	$5.8\cdot 10^{-5}$
10^{-6}	$2.37\cdot 10^{3}$	$5.5\cdot 10^{-7}$
10^{-8}	$2.34\cdot 10^{4}$	$5.6\cdot 10^{-9}$
10^{-10}	$2.35\cdot 10^{5}$	$5.6\cdot 10^{-11}$
10^{-12}	$2.37\cdot 10^{6}$	$5.0\cdot 10^{-13}$
10^{-14}	$2.35\cdot 10^{7}$	$5.5\cdot 10^{-15}$

We see that each time N gets multiplied by about 10, and the error gets multiplied by about 10^{-2}, hence we have $|Q_N-I|\le CN^{-2}$. Therefore our algorithm gives adaptive subdivisions where the composite trapezoid rule $|Q_N-I|$ decays with the same rate $O(N^{-2})$ we can achieve for smooth functions.

5 Gaussian quadrature

5.1 Introduction

Sometimes we want to find integrals of functions with singularities, e.g.

$$I=\int_0^1 x^{1/3}e^x dx$$

Note that the function $x^{1/3}$ has infinite deriviatives at $x=0$. Therefore composite rules with subintervals of equal size will perform poorly (however, adaptive quadrature works well).

Assume that we want to compute many integrals of the form

$$I=\int_0^1 x^{1/3}f(x)dx$$

where $f(x)$ is a smooth function. Then it makes sense to design a custom quadrature rule

$$\int_0^1 x^{1/3}f(x)dx\approx w_1f(x_1)+\cdots+w_nf(x_n)$$

where we try to find the optimal placement of the nodes $x_1,\dots,x_n$.

5.2 Gauss rule $I=\int_a^b f(x)\rho(x)dx\approx w_1f(x_1)+\cdots+w_nf(x_n)$

We want to approximate integrals of the form

$$I[f]:=\int_a^b f(x)\rho(x)dx$$

where the function $\rho(x)>0$ except for finitely many points where $\rho=0$ or $\rho\to\infty$. We assume that $\int_a^b p(x)\rho(x)dx<\infty$ for all polynomials $p(x)$. In the previous section we had the interval $[a,b]=[0,1]$ and the function $\rho(x)=x^{1/3}$. Note that we can also use $\rho(x)=1$ or $\rho(x)=x^{-1/3}$.

We want to approximate the integral with a quadrature rule

$$I[f]\approx Q[f]:=w_1f(x_1)+\cdots+w_nf(x_n)$$

For any choice of nodes $x_1,\dots,x_n$ we can construct the interpolating polynomial $p(x)$ and approximate $I[f]=\int_a^b f(x)\rho(x)dx$ by

$$Q[f]:=\int_a^b p(x)\rho(x)dx=w_1f(x_1)+\cdots+w_nf(x_n)$$

This rule is exact for all polynomials of degree$\leq n-1$.

Our goal is to **choose the nodes** $x_1,\dots,x_n$ **such that** $Q[f]=I[f]$ **for all polynomials of degree**$\leq 2n-1$**.**

The **node polynomial** is given by $\omega_n(x)=(x-x_1)\cdots(x-x_n)$.

Assume we are given an polynomial $p_{2n-1}(x)$ of degree$\leq 2n-1$. We can divide $p_{2n-1}(x)$ by $\omega_n(x)$ and obtain a quotient polynomial $q_{n-1}(x)$ and a remainder polynomial $r_{n-1}(x)$ where $q_{n-1}(x)$, $r_{n-1}(x)$ are of degree$\leq n-1$:

$$p_{2n-1}(x)=q_{n-1}(x)\cdot\omega_n(x)+r_{n-1}(x)$$

Hence

$$\begin{aligned} I[p_{2n-1}] &= I[q_{n-1}\cdot\omega_n]+I[r_{n-1}] \\ Q[p_{2n-1}] &= Q[q_{n-1}\cdot\omega_n]+Q[r_{n-1}] \end{aligned}$$

Since the rule Q is exact for polynomials of degree$\leq n-1$ we have $Q[r_{n-1}]=I[r_{n-1}]$.

Since $q_{n-1}(x)\cdot\omega_n(x)$ is zero in all nodes $x_1,\dots,x_n$ we have $Q[q_{n-1}\cdot\omega_n]=0$.

Hence we have $Q[p_{2n-1}]=I[p_{2n-1}]$ if and only if $I[q_{n-1}\cdot\omega_n]=0$.

Result: Our quadrature rule Q is exact for all polynomials $p_{2n-1}(x)$ of degree$\leq 2n-1$ iff $\omega_n(x)$ satisfies $I[q_{n-1}\cdot\omega_n]=0$ for all polynomials $q_{n-1}(x)$ of degree$\leq n-1$, i.e.,

$$\int_a^b x^k\omega_n(x)\rho(x)dx=0 \qquad \text{for } k=0,\dots,n-1 \tag{6}$$

Step 1: Find node polynomial $\omega_n(x)=(x-x_1)\cdots(x-x_n)=x^n+c_{n-1}x^{n-1}+\cdots+c_1x+c_0$ satisfying (6). I.e., find n unknowns $c_0,\dots,c_{n-1}$ satisfying n linear equations given by (6). After we have found the integrals

$$\alpha_k:=I\left[x^k\right]=\int_a^b x^k\rho(x)dx \qquad \text{for } k=0,\dots,2n-1$$

we can write the linear system (6) as

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_1 & \alpha_2 & & \vdots \\ \vdots & & \ddots & \alpha_{2n-3} \\ \alpha_{n-1} & \cdots & \alpha_{2n-3} & \alpha_{2n-2} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = - \begin{bmatrix} \alpha_n \\ \alpha_{n+1} \\ \vdots \\ \alpha_{2n-1} \end{bmatrix} \tag{7}$$

Solving this linear system gives us the node polynomial $\omega(x)=x^n+c_{n-1}x^{n-1}+\cdots+c_0=(x-x_1)\cdots(x-x_n)$. This linear system has always a unique solution.

Step 2: find the nodes $x_1,\dots,x_n$ **by solving** $\omega(x)=0$**.** This always gives n distinct roots $x_1,\dots,x_n$ in the open interval (a,b).

Step 3: find the weights $w_1,\dots,w_n$ **such that** $Q\left[x^k\right]=I\left[x^k\right]$ **for** $k=0,\dots,n-1$**:** This gives the linear system

$$\begin{bmatrix} x_1^0 & \cdots & x_n^0 \\ \vdots & & \vdots \\ x_1^{n-1} & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} \tag{8}$$

This linear system has always a unique solution. The weights w_j are all positive. Therefore evaluating $Q[f]=w_1f(x_1)+\cdots+w_nf(x_n)$ in machine arithmetic is numerically stable if $f(x)$ does not change sign.

5.3 Example 1: $\int_0^1 x^{4/7} f(x)dx \approx w_1 f(x_1) + w_2 f(x_2)$

Here $\rho(x) = x^{4/7}$, $[a,b] = [0,1]$ and $n = 2$. We first compute the integrals

$$\alpha_0 = \int_0^1 x^{4/7} 1\, dx = \tfrac{7}{11}, \qquad \alpha_1 = \int_0^1 x^{4/7} x\, dx = \tfrac{7}{18}, \qquad \alpha_2 = \int_0^1 x^{4/7} x^2 dx = \tfrac{7}{25}, \qquad \alpha_3 = \int_0^1 x^{4/7} x^3 dx = \tfrac{7}{32}$$

Step 1: find a node polynomial $\omega(x) = x^2 + c_1 x + c_0$ such that

$$\int_0^1 x^{4/7} x^k (x^2 + c_1 x + c_0) dx = 0 \qquad \text{for } k = 0, 1$$

which gives the linear system

$$\begin{bmatrix} \alpha_0 & \alpha_1 \\ \alpha_1 & \alpha_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = - \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} \frac{7}{11} & \frac{7}{18} \\ \frac{7}{18} & \frac{7}{25} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} -\frac{7}{25} \\ -\frac{7}{32} \end{bmatrix}$$

which has the solution $\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} .2475 \\ -1.125 \end{bmatrix}$, hence $\omega(x) = x^2 - 1.125x + .2475 = (x - x_1)(x - x_2)$.

Step 2: find the nodes x_1, x_2 by solving $\omega(x) = 0$: The quadratic formula gives $x_1 = .3$, $x_2 = .825$.

Step 3: **find the weights w_1, w_2 such that $Q\left[x^k\right] = I\left[x^k\right]$ for $k = 0, 1$:**

$$\begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} 1 & 1 \\ .3 & .825 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{7}{11} \\ \frac{7}{18} \end{bmatrix}$$

which has the solution $w_1 = \frac{7}{27}$, $w_2 = \frac{112}{297}$.

Our quadrature rule is exact for all polynomials of degree$\leq 2n - 1 = 3$.

Let us try out this rule for $I = \int_0^1 x^{4/7} e^x dx$: We obtain $Q = w_1 e^{x_1} + w_2 e^{x_2} = 1.21047$, the exact value is $I = 1.21066$, the error is $|Q - I| \approx 2 \cdot 10^{-4}$.

5.4 Example 2: $\int_{-1}^1 f(x)dx \approx w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3)$

Here $\rho(x) = 1$, $[a,b] = [-1,1]$ and $n = 3$. We first compute the integrals

$$\alpha_0 = \int_{-1}^1 1\, dx = 2, \quad \alpha_1 = \int_{-1}^1 x\, dx = 0, \quad \alpha_2 = \int_{-1}^1 x^2 dx = \tfrac{2}{3}, \quad \alpha_3 = \int_{-1}^1 x^3 dx = 0, \quad \alpha_4 = \int_{-1}^1 x^4 dx = \tfrac{2}{5}, \quad \alpha_5 = \int_{-1}^1 x^5 dx = 0$$

Step 1: find a node polynomial $\omega(x) = x^3 + c_2 x^2 + c_1 x + c_0$ such that

$$\int_0^1 x^k (x^3 + c_2 x^2 + c_1 x + c_0) dx = 0 \qquad \text{for } k = 0, 1, 2$$

which gives the linear system

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 \\ \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = - \begin{bmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} 2 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{2}{5} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{2}{5} \\ 0 \end{bmatrix}$$

which has the solution $\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{3}{5} \\ 0 \end{bmatrix}$, hence $\omega(x) = x^3 - \frac{3}{5}x = (x - x_1)(x - x_2)(x - x_3)$.

Step 2: find the nodes x_1, x_2, x_3 by solving $\omega(x) = x\left(x^2 - \frac{3}{5}\right) = 0$. This gives

$$x_1 = -\sqrt{\tfrac{3}{5}}, \qquad x_2 = 0, \qquad x_3 = \sqrt{\tfrac{3}{5}}$$

Step 3: find the weights w_1, w_2, w_3 such that $Q\left[x^k\right] = I\left[x^k\right]$ for $k = 0, 1$:

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ x_1^2 & x_2^2 & x_3^2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} 1 & 1 & 1 \\ -\sqrt{\frac{3}{5}} & 0 & \sqrt{\frac{3}{5}} \\ \frac{3}{5} & 0 & \frac{3}{5} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \end{bmatrix}$$

which has the solution

$$w_1 = \tfrac{5}{9}, \qquad w_2 = \tfrac{8}{9}, \qquad w_3 = \tfrac{5}{9}$$

Our quadrature rule is exact for all polynomials of degree$\leq 2n-1 = 5$.

Let us try out this rule for $I = \int_{-1}^{1} e^x dx$: We obtain $Q = w_1 e^{x_1} + w_2 e^{x_2} + w_3 e^{x_3} = 2.350337$, the exact value is $I = 2.350402$, the error is $|Q - I| = 6.5 \cdot 10^{-5}$.

5.5 Example 3: $\int_0^\infty f(x) e^{-x} dx \approx w_1 f(x_1) + w_2 f(x_2)$

Here $\rho(x) = e^{-x}$, $(a,b) = (0,\infty)$ and $n = 2$. The infinite interval $(0,\infty)$ is allowed since $\int_0^\infty p(x) e^{-x} dx < \infty$ for all polynomials $p(x)$. We first compute the integrals

$$\alpha_0 = \int_0^\infty e^{-x} 1\, dx = 1, \qquad \alpha_1 = \int_0^\infty e^{-x} x\, dx = 1, \qquad \alpha_2 = \int_0^\infty e^{-x} x^2 dx = 2, \qquad \alpha_3 = \int_0^\infty e^{-x} x^3 dx = 6$$

Step 1: find a node polynomial $\omega(x) = x^2 + c_1 x + c_0$ such that

$$\int_0^\infty e^{-x} x^k (x^2 + c_1 x + c_0) dx = 0 \qquad \text{for } k = 0, 1$$

which gives the linear system

$$\begin{bmatrix} \alpha_0 & \alpha_1 \\ \alpha_1 & \alpha_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = -\begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} -2 \\ -6 \end{bmatrix}$$

which has the solution $\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \end{bmatrix}$, hence $\omega(x) = x^2 - 4x + 2 = (x - x_1)(x - x_2)$.

Step 2: find the nodes x_1, x_2 by solving $\omega(x) = 0$: The quadratic formula gives

$$x_1 = 2 - \sqrt{2}, \qquad x_2 = 2 + \sqrt{2}$$

Step 3: find the weights w_1, w_2 such that $Q\left[x^k\right] = I\left[x^k\right]$ for $k = 0, 1$:

$$\begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}, \qquad \text{i.e.,} \quad \begin{bmatrix} 1 & 1 \\ 2-\sqrt{2} & 2+\sqrt{2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

which has the solution

$$w_1 = \tfrac{1}{2} + \tfrac{1}{4}\sqrt{2}, \qquad w_2 = \tfrac{1}{2} - \tfrac{1}{4}\sqrt{2}$$

Our quadrature rule is exact for all polynomials of degree$\leq 2n-1 = 3$.

Let us try out this rule for $I = \int_0^\infty e^{-x} \sin x dx$: We obtain $Q = w_1 \sin x_1 + w_2 \sin x_2 = 0.43246$, the exact value is $I = 0.5$, the error is $|Q - I| = 6.8 \cdot 10^{-2}$. As we increase n the quadrature error goes to zero:

n	1	2	3	6	10
$\lvert Q_n - I\rvert$	$3 \cdot 10^{-1}$	$7 \cdot 10^{-2}$	$4 \cdot 10^{-3}$	$5 \cdot 10^{-5}$	$2 \cdot 10^{-7}$

5.6 Proofs for Gauss quadrature

We have to prove that all the steps work as claimed.

Step 1: find a node polynomial $\omega(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_0$ such that

$$\int_a^b x^k \omega(x)\rho(x)dx = 0 \qquad \text{for } k = 0,\ldots,n-1 \tag{9}$$

This gives the linear system (7). We claim that this linear system has a unique solution.

Proof: The linear system with zero right hand side vector corresponds to finding $p(x) = c_{n-1}x^{n-1} + \cdots + c_0$ such that

$$\int_a^b x^k p(x)\rho(x)dx = 0 \qquad \text{for } k = 0,\ldots,n-1$$

But then

$$0 = \int_a^b \left(c_{n-1}x^{n-1} + \cdots + c_0x^0\right) p(x)\rho(x)dx = \int_a^b p(x)^2\rho(x)dx$$

Since $\rho(x) > 0$ except for finitely many points we must have $p(x) = 0$. Hence the linear system with zero right hand side vector has only the solution $c_0 = \cdots = c_{n-1} = 0$, and the matrix is nonsingular.□

We can define for two functions u, v on (a,b) the **inner product**

$$(u,v) := \int_a^b u(x)v(x)\rho(x)dx$$

The condition (9) says that $\omega(x)$ is a **polynomial of degree n which is orthogonal on all lower degree polynomial**. Such a polynomial is called an **orthogonal polynomial**.

Step 2: find the nodes $x_1,\ldots,x_n$ by solving $\omega(x) = 0$. We claim that the function $\omega(x)$ has n distinct simple roots in the open interval (a,b).

Proof: Let $t_1,\ldots,t_r$ denote all points in (a,b) where $\omega(x)$ changes sign. We want to show that $r = n$. Assume that $r < n$. As $(x-t_1)\cdots(x-t_r)\omega(x)$ always has for $x \neq t_j$ always the same sign, say positive, we must have

$$\int_a^b (x-t_1)\cdots(x-t_r)\omega(x)\rho(x)dx > 0$$

On the other hand this integral must be zero by the orthogonality (9).□

Step 3: find the weights $w_1,\ldots,w_n$ by solving the linear system (8). We claim that this linear system has a unique solution.

Proof: Finding an interpolating polynomial $c_0 + \cdots + c_{n-1}x^{n-1}$ leads to a linear system with the transpose matrix. The interpolation problem has a unique solution, hence the matrix is nonsingular.□

The weights $w_1,\ldots,w_n$ are all positive: For $j \in \{1,\ldots,n\}$ define the polynomial of degree $n-1$

$$p_j(x) := \prod_{\substack{k=1,\ldots,n \\ k\neq j}} (x - x_k)$$

then $p_j(x)^2$ is of degree $2n-2$ and therefore exactly integrated by the Gauss rule:

$$0 < I\left[p_j^2\right] = Q\left[p_j^2\right] = w_j \underbrace{p_j(x_j)^2}_{>0}$$