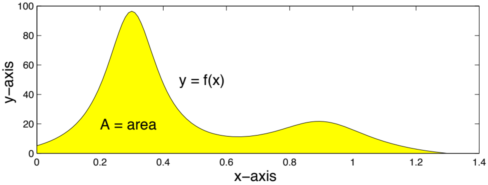
Numerical integration (quadrature)

Chapter 5.1 until page 211, 5.3 until page 235, 5.4 until page 241.

Numerical integration (or numerical quadrature) is to calculate integral numerically:

$$
\begin{aligned}
I &= \int_a^b f(x)dx \\
&= \text{the area under the curve of } f \text{ (if } f \text{ is positive).}
\end{aligned}
$$



Mathematical example

Example of integral, which can be calculated analytically

$$\int_0^1 e^{-x} dx = -e^{-x}\Big|_0^1 = -\left(\frac{1}{2.7183} - 1\right) = 0.6321.$$

Example of integral, which cannot be expressed in terms of elementary functions:

$$\phi = \int_0^1 \exp\left(-x^2\right) dx$$
$$\psi = \int_a^{\pi/2} \sqrt{1 + \cos(x)^2} dx.$$

The integral ϕ commonly appears in probability theory.
The integral ψ commonly appears in analytic geometry.

The numerical problem

In order to determine the numerical value of an integral

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

we can calculate an *approximation* through

$$I \approx \int_a^b f_n(x)dx,$$

where f_n is a polynomial that approximates f. We can integrate f_n analytically, since its integral has a "closed" expression. In this way we obtain an approximation to the integral of f.

The main difference between different numerical integration algorithms is how to choose this "approximate" polynomial.

Approximation through interpolating polynomials

For a table

x	x_0	x_1	$\dots$	x_n
y	$f(x_0)$	$f(x_1)$	$\dots$	$f(x_n)$

the Lagrange form of the interpolating polynomial is $p_n(x) = \sum_{i=0}^{n} l_i(x) f(x_i)$.

Hence, the approximation of the integration f is

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i) dx$$
$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i) A_i,$$

where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \dots, n$, can be predetermined.

The approximation of the integral is just a weighted linear combination of the function values $f(x_0), f(x_1), \dots, f(x_n)$.

Approximation through interpolating polynomials

For a table $\begin{array}{c|cccc} x & x_0 & x_1 & \dots & x_n \\ \hline y & f(x_0) & f(x_1) & \dots & f(x_n) \end{array}$ the Lagrange form of the interpolating polynomial is $p_n(x) = \sum_{i=0}^{n} l_i(x) f(x_i)$.

Hence, the approximation of the integration f is

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i) dx$$
$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i) A_i,$$

where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \dots, n$, can be predetermined.

The approximation of the integral is just a weighted linear combination of the function values $f(x_0), f(x_1), \dots, f(x_n)$. **Not much calculation is needed! (-:**

Newton-Cotes formulas

For $n+1$ equally spaced nodes $x_i = a + ih$, where $h = (b-a)/n$, with predetermined A_i we obtain the Newton-Cotes quadrature formula. We can estimate the error through the first/second interpolation error theorem.

> *Second Interpolation Error Theorem:* *Let f be a function such that $f^{(n+1)}$ is continuous on $[a, b]$ and satisfies $\left|f^{(n+1)}(x)\right| \leq M$. Let p_n be the polynomial of degree at most n that interpolates f at the $n+1$ equally spaced nodes in $[a, b]$ with the spacing between nodes $h = (b-a)/n$, including the endpoints. Then, for all $x \in [a, b]$, we have*
>
> $$\left|f(x) - p_n(x)\right| \leq \frac{1}{4(n+1)}\left|f^{(n+1)}(\xi)\right| h^{n+1} \leq \frac{1}{4(n+1)} M h^{n+1}.$$

Thus,

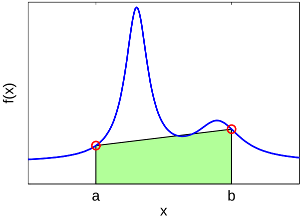
$$\left|\int_a^b f(x)dx - \int_a^b p_n(x)dx\right| \leq \int_a^b \left|f(x) - p_n(x)\right| dx \leq (b-a)\frac{1}{4(n+1)} M h^{n+1},$$

but we can get even better error estimation for specific n.

Trapezoid rule

For $n = 1$ we choose $x_0 = a$ and $x_1 = b$.

The approximation by a polynomial of degree 1, p_1, is in fact the line through the points $\big(a, f(a)\big)$ and $\big(b, f(b)\big)$.



$$
\begin{aligned}
\int_a^b f(x)dx \approx \int_a^b p_1(x)dx &= \int_a^b \Big(f(a)l_0(x) + f(b)l_1(x)\Big)dx \\
&= f(a)\int_a^b l_0(x)dx + f(b)\int_a^b l_1(x)dx \\
&= f(a)\frac{b-a}{2} + f(b)\frac{b-a}{2} \\
&= h\frac{1}{2}\Big(f(a) + f(b)\Big),
\end{aligned}
$$

where $h = (b - a)/1$.

Error analysis for trapezoid rule

In the book, it used the first interpolation error theorem,
$f(x) - p_1(x) = \frac{1}{(1+1)!} f^{(1+1)}\big(\zeta(x)\big)(x-a)(x-b)$, to analyze the error

$$\begin{aligned}
\int_a^b f(x)dx - \int_a^b p_1(x)dx &= \int_a^b \frac{1}{2} f^{''}\big(\zeta(x)\big)(x-a)(x-b)dx \\
&= \frac{1}{2} f''(\xi) \int_a^b (x-a)(x-b)dx \\
&= -\frac{1}{12}(b-a)^3 f''(\xi), \text{ where } a < \xi < b.
\end{aligned}$$

A difficulty in this proof is that we know nothing about $\zeta(x)$, which can be discontinuous. But it can be shown (not in this course) that the function $x \to f^{''}\big(\zeta(x)\big)$ is indeed continuous and thus integrable. Then, based on the mean-value theorem for integrals, which you maybe do not know, we get the second equivalence.

Now we give an alternative explanation.

Error analysis for trapezoid rule by Taylor series

Recall the Taylor series $f(a+h) = f(a) + hf'(a) + \frac{h^2}{2!}f''(a) + \dots$.
Thus, Taylor series of Trapezoid rule is

$$\begin{aligned}\frac{h}{2}\Big(f(a) + f(a+h)\Big) &= \frac{h}{2}\Big(2f(a) + hf'(a) + \frac{h^2}{2!}f''(a) + \dots\Big) \\ &= hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{4}f''(a) + \dots\end{aligned}$$

Taylor series of $F(t) = \int_a^t f(x)dx$ is:

$$F(a+h) = F(a) + hF'(a) + \frac{h^2}{2!}F''(a) + \frac{h^3}{3!}F'''(a) + \dots \ .$$

We have $F(a) = 0$, $F'(a) = f(a)$, $F''(a) = f'(a)$ and $F'''(a) = f''(a)$, so

$$F(a+h) = hf(a) + \frac{h^2}{2!}f'(a) + \frac{h^3}{3!}f''(a) + \dots \ .$$

Error analysis for trapezoid rule by Taylor series

Now compare the two Taylor series:

$$F(a+h) = \int_a^b f(x)dx = hf(a) + \frac{h^2}{2!}f'(a) + \frac{h^3}{3!}f''(a) + \dots \text{ and}$$

$$\frac{h}{2}\Big(f(a) + f(a+h)\Big) = hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{4}f''(a) + \dots$$

This shows that the error is

$$\int_a^b f(x)dx - \frac{h}{2}\Big(f(a) + f(a+h)\Big)$$

$$= \Big(\frac{1}{6} - \frac{1}{4}\Big)h^3 f''(a) + \dots = -\frac{1}{12}h^3 f''(a) + \dots = -\frac{1}{12}(b-a)^3 f''(a) + \dots.$$

We recognize the constant $\frac{1}{12}(b-a)^3$ from before. (-:

Trapezoid rule

We write:

$$I = \int_a^b f(x)dx = h\frac{1}{2}\Big(f(a) + f(b)\Big) + E_t, \qquad h = b - a,$$

where E_t is *truncation error*. We had shown that:

$$E_t = -\frac{1}{12}h^3 f''(\xi), \qquad h = b - a$$

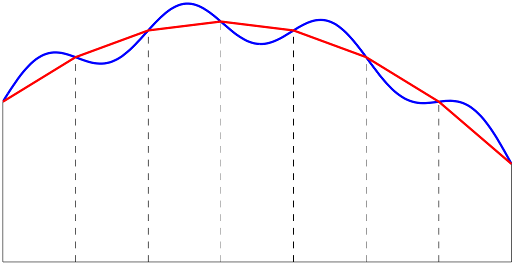
where ξ is an unknown value between a and b. Thus,

$$|E_t| \leq \frac{1}{12}h^3 \max_{a \leq \xi \leq b} |f''(\xi)|.$$

Hence, the error is large if $|f''(\xi)|$ is large, or h is large.

We must balance a **large** $|f''(\xi)|$ by a small spacing h.

Composite trapezoid rule



We use that

$$\int_a^b f(x)dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)dx.$$

Composite trapezoid rule

We set $h = (b-a)/n$ and $x_i = a + ih$ and apply trapezoid rule to every subinterval $[x_{i-1}, x_i]$

$$\int_{x_{i-1}}^{x_i} f(x)dx = \frac{h}{2}\left(f_{i-1} + f_i\right) - \frac{h^3}{12} f''(\xi_i), \qquad x_{i-1} < \xi_i < x_i.$$

Thus, we have

$$\int_a^b f(x)dx = h\left(\tfrac{1}{2} f_0 + f_1 + f_2 + \cdots + f_{n-1} + \tfrac{1}{2} f_n\right) + E_t,$$

where the error is

$$E_t = -\frac{(b-a)\,h^2}{12}\left(\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)\right)$$

with $a \le x_{i-1} < \xi_i < x_i \le b$.

Composite trapezoid rule

Since f'' is continuous, we know that the average

$$\left(\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)\right)$$

lies between the smallest and largest values of f'' on the interval (a, b). By the intermediate-value theorem, there is a point $\zeta \in (a, b)$ such that

$$\left(\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)\right) = f''(\zeta),$$

Theorem on precision of composite trapezoid rule

If f'' exists and is continuous on the interval $[a, b]$ and if the interval is divided into n equal subintervals with points $x_i = a + ih$ and spacing $h = (b - a)/n$, then the composite trapezoid rule is

$$\int_a^b f(x)dx = h\big(\tfrac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{n-1} + \tfrac{1}{2}f_n\big) + E_t,$$

where the error term is

$$E_t = -\frac{(b-a)h^2}{12}\left(\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)\right) = -\frac{(b-a)h^2}{12}\, f''(\zeta)$$

with $a < \zeta < b$.

The error is $\mathcal{O}(h^2)$, i.e., it is proportional to h^2 and $\frac{1}{n^2}$.

Example of trapezoid rule (Pseudocode page 204)

Example: Calculate $\int_0^1 (1+x)^{-1}\,dx = \ln 2 \approx 0.693147$.
Results:

n	h	$T(h)$	$T(h) - \ln 2$
1	1	0.750000	0.056853
2	0.5	0.708333	0.015186
4	0.25	0.697024	0.003877
8	0.125	0.694122	0.000975

Notice that the error is reduced by a factor $4 = 2^2$ when h is halved (reduced by a factor 2).

Simpson's rule

For a table $\begin{array}{c|ccc} x & x_0 & x_1 & x_2 \\ \hline y & f(x_0) & f(x_1) & f(x_2) \end{array}$ where $x_0 = a$, $x_2 = b$ and x_1 is chosen as the midpoint $\frac{a+b}{2}$, the Lagrange form of the interpolating polynomial is $p_2(x) = \sum_{i=0}^{2} l_i(x) f(x_i)$.

Hence, the corresponding approximation of the integral of f is

$$I = \int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \int_a^b \sum_{i=0}^{2} l_i(x) f(x_i) dx$$

$$= \sum_{i=0}^{2} f(x_i) \int_a^b l_i(x)dx = \frac{h}{3}\Big(f(x_0) + 4f(x_1) + f(x_2)\Big), \qquad h = \frac{b-a}{2},$$

The weights of the function values $f(x_0), f(x_1), f(x_2)$ can be calculated directly or be derived as follows $\rightarrow$ next page.

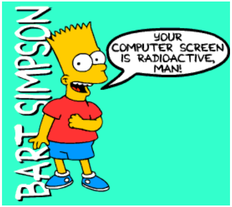
Simpson's rule

Let $a = -1$ and $b = 1$, and consider the formula

$$\int_{-1}^{1} f(x)dx \approx Af(-1) + Bf(0) + Cf(1).$$

This formula must give correct values for the integral whenever f is a polynomial of degree at most 2, i.e., it integrates 1, x and x^2 correctly. We use this to determine the coefficients A, B and C:

$$\int_{-1}^{1} 1dx = 2 = A + B + C$$
$$\int_{-1}^{1} xdx = 0 = -A + 0B + C$$
$$\int_{-1}^{1} x^2dx = \frac{2}{3} = A + 0B + C$$



The solution is $(A, B, C) = (1/3, 4/3, 1/3)$.
Using the linear mapping $y = \frac{1}{2}(b-a)x + \frac{1}{2}(a+b)$ from $[-1, 1]$ to $[a, b]$, we obtain the general formula on the previous page.

Error of Simpson's rule

Simpson's rule is designed to give correct values for the integral of a polynomial of degree at most 2.

We can apply Taylor series to obtain the error term of Simpson's rule

$$E_t = -\frac{h^5}{90} f^{(4)}(\xi) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi), \qquad a < \xi < b,$$

where $f^{(4)}$ is the fourth order derivative of f.
(You can use Maple to check it! `Day5SimpsonErrorEstimation.mw`)

Error of Simpson's rule

Simpson's rule is designed to give correct values for the integral of a polynomial of degree at most 2.

We can apply Taylor series to obtain the error term of Simpson's rule

$$E_t = -\frac{h^5}{90} f^{(4)}(\xi) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi), \qquad a < \xi < b,$$

where $f^{(4)}$ is the fourth order derivative of f.
(You can use Maple to check it! `Day5SimpsonErrorEstimation.mw`)

Nice surprise: Simpson's rule also gives correct values for the integral of the polynomial of degree 3!

Composite Simpson's rule

Similar to the composite trapezoid rule, we can divide $[a, b]$ into n subintervals, each of width $h = (b-a)/n$ and write

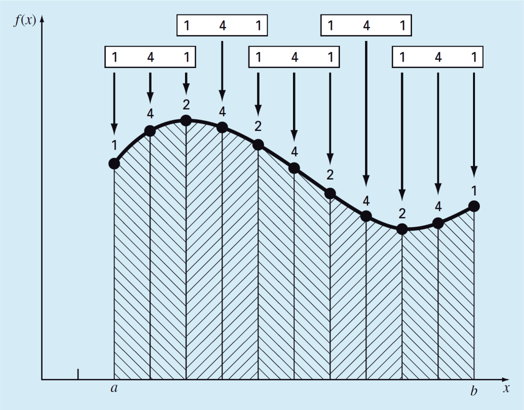
$$\int_a^b f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx$$
$$= \sum_{i=1}^{n/2} \left(\frac{h}{3}(f_{2i-2} + 4f_{2i-1} + f_{2i}) - \frac{h^5}{90} f^{(4)}(\eta_i) \right)$$
$$= \frac{h}{3}\big(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n\big) + E_t,$$

where we again apply the intermediate-value theorem on the average of $f^{(4)}$ and obtain the error

$$E_t = -\frac{b-a}{180} h^4 f^{(4)}(\xi) = -\frac{(b-a)^5}{180n^4} f^{(4)}(\xi), \qquad a < \xi < b.$$

The error is proportional with h^4 and $\frac{1}{n^4}$. Nice!

Weights in composite Simpson's rule



NB: n must be even!

Example of Simpson's rule

Example: calculate $\int_0^1 (1+x)^{-1} dx = \ln 2$.

n	h	$S(h)$	$S(h) - \ln 2$
2	0.5	0.694444	0.001297
4	0.25	0.693254	0.000107
8	0.125	0.693155	0.000007

Notice that the error is reduced with a factor $16 = 2^4$ when h is halved!

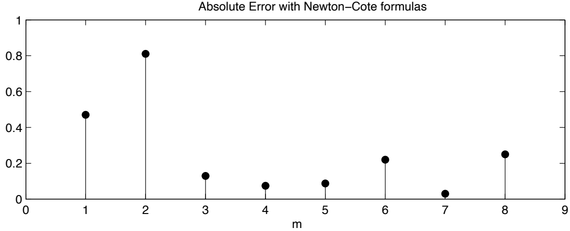
Compare the trapezoid rule and Simpson's rule – same amount of computations, better accuracy:

n	2	4	8
$T_n - \log 2$	0.056853	0.003877	0.000975
$S_n - \log 2$	0.001297	0.000107	0.000007

Should we increase the degree m of the polynomial?

Example: $J = \int_{-1}^{1}(1+25x^2)^{-1}dx \approx 0.549360.$

m	1	2	3	4	5	6	7	8
$I_m - J$	-0.47	0.81	-0.13	-0.075	-0.088	0.22	0.030	-0.25

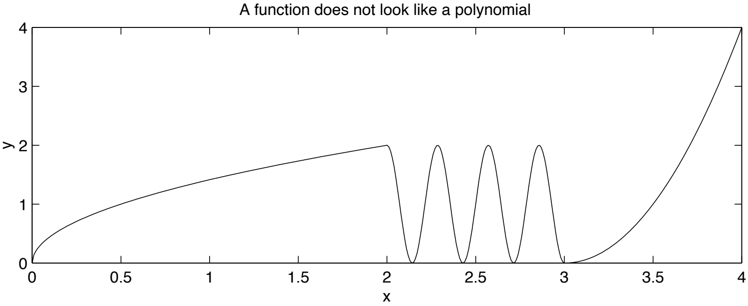


The error does not decrease monotonically with m.

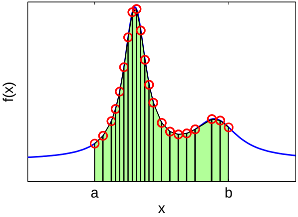
When is the polynomial-based formula effective and accurate?

Until now all methods are based on polynomial interpolation on equally spaced nodes. Therefore, we can conclude:

$\boxed{\text{The method is effective, when the integrand } f \text{ \textit{resembles} a polynomial.}}$



Adaptive algorithm



Basic idea:

- use subintervals of different lengths,
- that are suitable for the function

Specifically, use:

- fewer and larger subintervals where f is "simple" (i.e. varying less),
- more and smaller subintervals where f varies a lot.

The partitioning of the interval is automatically determined.
The partition is generated **adaptively** based on the function.

Recursive adaptive integration

Assume that we have an approximation I_1 with an error E:

$$\int_a^b f(x)dx = I_1 + E.$$

If the error E is too large, we divide $[a, b]$ into two subintervals $[a, c]$ and $[c, b]$ with the midpoint $c = (a + b)/2$:

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx = I_L + E_L + I_R + E_R,$$

where E_L and E_R are the errors from I_L and I_R, respectively.

If the total error $E_L + E_R$ is sufficiently small, then $I_2 = I_L + I_R$ is an acceptable approximation to the integral.
Recursive strategy: use the same idea to approximate I_L and I_R.

And now: with error estimates

Assume that we have an approximation I_1 with an error estimate E:

$$\int_a^b f(x)dx = I_1 + E.$$

If the error estimate E is too large, we divide $[a, b]$ into two subintervals $[a, c]$ and $[c, b]$ with the midpoint $c = (a + b)/2$:

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx = I_L + E_L + I_R + E_R,$$

where E_L and E_R are the error estimates from I_L and I_R, respectively.

If the total error estimate $E_L + E_R$ is sufficiently small, then $I_2 = I_L + I_R$ is an acceptable approximation to the integral.
Recursive strategy: use the same idea to approximate I_L and I_R.

How to calculate the error estimate?

We assume that $f^{(4)}$ is a constant throughout the interval $[a, b]$.
In Simpson's rule, with the length of the subinterval $h_1 = h$:

$$I = I(h_1) + Ch^5, \qquad C = \text{constant.} \tag{1}$$

Similarly, with the length of the interval $h_2 = h_1/2 = h/2$:

$$I = I_L(h_2) + C\left(\frac{h}{2}\right)^5 + I_R(h_2) + C\left(\frac{h}{2}\right)^5 = I(h_2) + \frac{1}{16}Ch^5. \tag{2}$$

Note: It is the **same** constant C in both expressions.

According to the error terms in (1) and (2), we have

$$I - I(h_2) = \tfrac{1}{16}\big(I - I(h_1)\big) \qquad \Leftrightarrow$$

$$\begin{aligned} I &= \tfrac{16}{15}\big(I(h_2) - \tfrac{1}{16}I(h_1)\big) \\ &= I(h_2) + \frac{I(h_2) - I(h_1)}{15} \quad \Rightarrow \quad \text{error estimate} = \tfrac{I(h_2)-I(h_1)}{15}. \end{aligned}$$

This technique is called as *Richardson-extrapolation*.

And how do we start the process?

Calculate the approximations $I(h_1)$ and $I(h_2)$ with Simpson's rule and the length of the interval $h_1 = (b-a)/2$ and $h_2 = (b-a)/4$.
Then we use Richardson-extrapolation to calculate a better approximation to the intergral,

$$I_3 = I(h_2) + \tfrac{1}{15}\big(I(h_2) - I(h_1)\big)$$

and $|I_3 - I(h_2)| = \frac{1}{15}|I(h_2) - I(h_1)|$ is the error estimate for $I(h_2)$.

If the error estimate is less than τ (given in advance), then stop.

Otherwise, repeat the process on each of the subintervals, $[a, c]$ and $[c, b]$, with $c = (a+b)/2$. On each subinterval, the tolerance τ is replaced by $\tau/2$.

Recursive procedure in Python

```
def integrer(f,a,b,tol,I1):

    c = (a+b)/2
    IL = simpson(f,a,c)
    IR = simpson(f,c,b)
    I2 = IL + IR
    I = I2 + (I2 - I1)/15

    if abs(I - I2) > tol:
        IL = integrer(f,a,c,tol/2,IL)
        IR = integrer(f,c,b,tol/2,IR)
    I = IL + IR
    return I
```

Numerical integration with Scipy

Scipy offers a number of functions for numerical integration:

- quad()/quadv(): adaptive Simpson quadrature for a scalar-valued function/vector-valued function.

```
>>> from scipy.integrate import quad
>>> I = quad(lambda x: np.sin(x), 0, np.pi, epsabs = 1e-1)
# Returns an estimate of the integral
# and an estimate of the absolute error
I = 2, 2.220446049250313e-14
```

- dblquad(), tplquad() and nquad() are for multiple integrals.
- quadrature(): adaptive Gauss quadrature.

Gaussian Quadrature Theorem

By a special placement of the nodes, the accuracy of the numerical integration process could be greatly increased:

> *Gaussian Quadrature Theorem: Let q be a nontrivial polynomial of degree $n+1$ such that*
>
> $$\int_a^b x^k q(x)dx = 0, \quad , k = 0, 1, \ldots, n.$$
>
> *Let $x_0, x_1, \ldots, x_n$ be the zeros of q. Then, the formula*
>
> $$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i), \text{where } A_i = \int_a^b l_i(x)dx$$
>
> *with these x_i's as nodes will be exact for all polynomials of* **degree at most** $2\text{n}+1$.

Note that we need to pre-compute all the x_i's.

Summary I

- Approximation of the integrand gives an approximation to the integral

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x)f(x_i)dx$$

$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i)A_i,$$

where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \ldots, n$, can be predetermined.

Summary I

- Approximation of the integrand gives an approximation to the integral

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i) dx$$

$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i) A_i,$$

where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \ldots, n$, can be predetermined.

- **Mathematical overview:** The approximation of the integral is just a linear combination of a set of function values.

Summary I

- Approximation of the integrand gives an approximation to the integral

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i) dx$$
$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i) A_i,$$

  where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \dots, n$, can be predetermined.
- **Mathematical overview:** The approximation of the integral is just a linear combination of a set of function values.
- The approximation by a polynomial of degree 1 gives the trapezoid rule $I = \int_a^b f(x)dx = \frac{h}{2}\big(f(a) + f(b)\big) + E_t$, where $h = b - a$ and $E_t = -\frac{1}{12}h^3 f''(\xi)$, for a $\xi \in (a, b)$.

Summary I

- Approximation of the integrand gives an approximation to the integral

$$I = \int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i) dx$$

$$= \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^{n} f(x_i) A_i,$$

where the constants, $A_i = \int_a^b l_i(x)dx$, $i = 0, \dots, n$, can be predetermined.
- **Mathematical overview:** The approximation of the integral is just a linear combination of a set of function values.
- The approximation by a polynomial of degree 1 gives the trapezoid rule $I = \int_a^b f(x)dx = \frac{h}{2}\big(f(a) + f(b)\big) + E_t$, where $h = b - a$ and $E_t = -\frac{1}{12}h^3 f''(\xi)$, for a $\xi \in (a, b)$.
- Composite trapezoid rule $I = h\big(\frac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{n-1} + \frac{1}{2}f_n\big) + E_t$ for the spacing $h = (b - a)/n$ has the error $E_t = -\frac{(b-a)\,h^2}{12} f''(\zeta)$ with $\zeta \in (a, b)$.

Summary II

- The approximation by a polynomial of degree 2 gives Simpson's rule $\frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big)$ where $h = \frac{b-a}{2}$, and the error term is $E_t = -\frac{h^5}{90} f^{(4)}(\xi)$ for a $\xi \in (a, b)$.

Summary II

- The approximation by a polynomial of degree 2 gives Simpson's rule $\frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big)$ where $h = \frac{b-a}{2}$, and the error term is $E_t = -\frac{h^5}{90} f^{(4)}(\xi)$ for a $\xi \in (a, b)$.
- Composite Simpson's rule has the error $E_t = -\frac{b-a}{180} h^4 f^{(4)}(\xi)$ for a $\xi \in (a, b)$. Note: n must be even.

Summary II

- The approximation by a polynomial of degree 2 gives Simpson's rule $\frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big)$ where $h = \frac{b-a}{2}$, and the error term is $E_t = -\frac{h^5}{90} f^{(4)}(\xi)$ for a $\xi \in (a, b)$.
- Composite Simpson's rule has the error $E_t = -\frac{b-a}{180} h^4 f^{(4)}(\xi)$ for a $\xi \in (a, b)$. Note: n must be even.
- Generally, it does not help to increase the degree of the approximating polynomial in the composite formulas. But adaptive algorithms are preferable.

Summary II

- The approximation by a polynomial of degree 2 gives Simpson's rule $\frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big)$ where $h = \frac{b-a}{2}$, and the error term is $E_t = -\frac{h^5}{90} f^{(4)}(\xi)$ for a $\xi \in (a, b)$.
- Composite Simpson's rule has the error $E_t = -\frac{b-a}{180} h^4 f^{(4)}(\xi)$ for a $\xi \in (a, b)$. Note: n must be even.
- Generally, it does not help to increase the degree of the approximating polynomial in the composite formulas. But adaptive algorithms are preferable.
- **Remember:** Next week there will be a 4-hour exercise. The homework assignment need be handed in before **24 Oct. 10pm**.