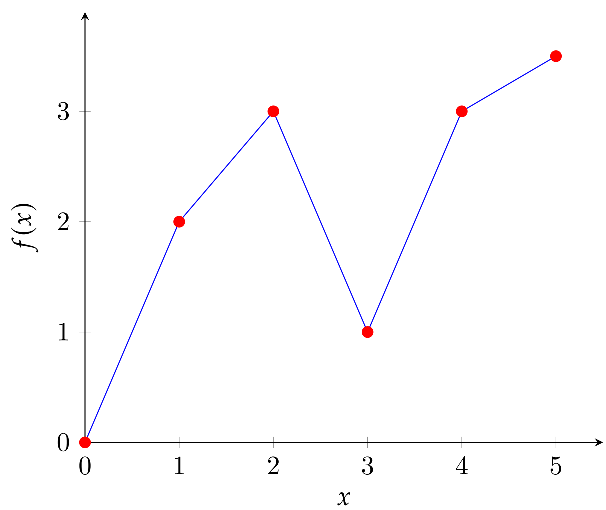
# Numerical Analysis
Semester 4

# Contents

# Chapter 1

# Interpolation

## 1.1 Linear Interpolation




| $x$ | $f(x)$ |
|---|---|
| 0 | 0 |
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |
| 4 | 3 |
| 5 | 3.5 |

Linear interpolation is just drawing lines between the data points.

**Definition 1.1.1: Linear Interpolation(lerp) equation**

The equation of the lines between data points is

$$y = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) + y_i.$$

**Theorem 1.1.1** Error due to linear interpolation

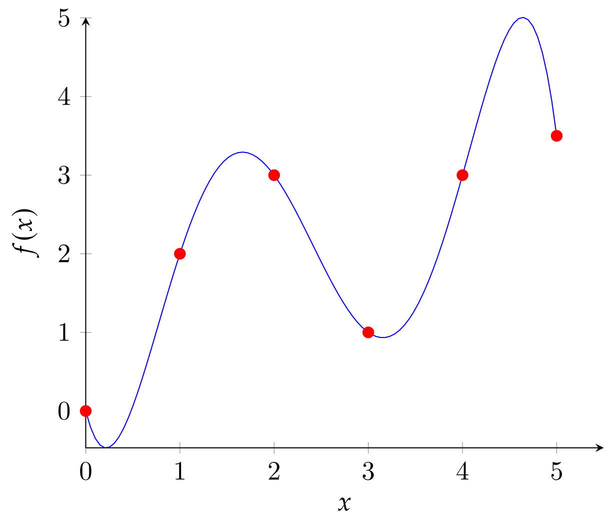
Let $f$ be a continuous and differentiable on $[a, b]$. We define the error $z(x)$ to be

$$|z(x)| \leqslant \frac{(b-a)^2}{8} \sup_{a \leqslant x \leqslant b} |f''(x)|.$$

## 1.2 Polynomial Interpolation

### 1.2.1 Lagrange Polynomials

Really nice video here explaining Lagrange polynomials.




**Theorem 1.2.1** Lagrange polynomial equation

Consider a set of $n$ points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. The Lagrange polynomial for this set of data is

$$L(x) = \sum_{k=0}^{n} y_k \ell_k(x).$$

where

$$\ell_k(x) = \prod_{\substack{i=1 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}.$$

### Case of equidistant points

If the set of $x_i$ are equidistant from each other with a distance of $h = x_{i+1} - x_i$, then we can represent any point as $x_k = x_0 + kh$ where $k \in \mathbb{N}$ and any number $x = x_0 + sh$ where $s \in \mathbb{R}$. We can rewrite the formula as

$$Q(s) = \sum_{k=0}^{n} \ell_k(s) f(x_k).$$

where

$$\ell_k(s) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{s - j}{k - j}.$$

by substitution

$$s = \frac{x - x_0}{h}.$$

### Existence

***Proof:*** $P(x)$ belongs to the vectorial space of polynomial of degree of, at most, $n$. Now, we must fins a basis for this vectorial space. Find the polynomial $\ell_k$ of degree $\leqslant n$ such that

$$\ell_k(x_i) = \delta_{ki} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}.$$

Then, $\ell_k(x) = \lambda(x - x_0) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$ where

$$\lambda = \frac{1}{(x_k - x_0) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)}.$$

The $(n+1)$ polynomials $\ell_k(x)$ for a system of generators in the vectorial space of polynomials of degree at most $n$.

$$\lambda_0\ell_0(x) + \lambda_1\ell_1(x) + \cdots + \lambda_k\ell_k(x) + \cdots + \lambda_n\ell_n(x) = 0.$$

for $x = x_k$

$$\begin{aligned} \lambda_0\ell_0(x_k) + \lambda_1\ell_1(x_k) + \cdots + \lambda_k\ell_k(x_k) + \cdots + \lambda_n\ell_n(x_k) &= 0 \\ 0 + 0 + \cdots + \lambda_k 1 + \cdots + 0 &= 0 \\ \lambda_k &= 0. \end{aligned}$$

$\therefore$ the set of $\ell_k$ for a basis in the vector space $\Rightarrow$ there has to exist a polynomial passing through the given set of points.

$\square$

**Uniqueness**

***Proof:*** Let $P$ and $Q$ be 2 Lagrange polynomials of degrees $\leqslant n \Big/ P(x_i) = Q(x_i) = f(x_i) \quad \forall i = 0, 1, \ldots, n$.
Let

$$\left.\begin{array}{l} R = P - Q \text{ of degree } \leqslant n \\ R = 0\ (n+1) \text{ times} \end{array}\right\} R \equiv 0 \Rightarrow P = Q\ \forall x.$$

$\square$

### 1.2.2 Newton Polynomial

**Definition 1.2.1: Newton Polynomial equation**

Consider a set of $n$ points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The Newton polynomial for this set of data is

$$p_n(x) = \underbrace{a_0}_{A_0} + \underbrace{a_1(x - x_0)}_{A_1} + \underbrace{a_2(x - x_0)(x - x_1)}_{A_2} + \cdots + \underbrace{a_n \prod_{i=0}^{n-1}(x - x_i)}_{A_n}.$$

where

$$a_i = f[x_0, x_1, \ldots, x_i].$$

Here $f[\ldots]$ is the divided difference of the inputted data.

**Definition 1.2.2: Backwards formula**

$$P_n(x) = f_n + A_1 + A_2 + \cdots + A_n.$$

where

$$A_i = f[x_n, x_{n-1}, \ldots, x_{n-i}] \prod_{j=n-i+1}^{n} (x - x_j).$$

The divided difference has 2 formulas, the recurrence formula

$$f[x_0, x_1, \ldots, x_{n+1}] = \frac{f[x_1, x_2, \ldots, x_{n+1}] - f[x_0, x_1, \ldots, x_n]}{x_{n+1} - x_0}.$$

and a general formula

$$f[x_0, x_1, \ldots, x_n] = \sum_{i=1}^{n} \frac{y_i}{\prod_{\substack{k=0 \\ k \neq i}}^{n} (x_i - x_k)}.$$

Now forget you ever saw those cause there is an easier method to finding the divided difference.

**Divided Difference Table**

$$
\begin{array}{llllll}
x_0 & y_0 & & & & \\
& & \frac{y_1-y_0}{x_1-x_0} = f[x_0,x_1] & & & \\
x_1 & y_1 & & \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} & & \\
& & \frac{y_2-y_1}{x_2-x_1} = f[x_1,x_2] & & \dots & \\
x_2 & y_2 & & \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} & & \dots \\
& & \frac{y_3-y_2}{x_3-x_2} = f[x_2,x_3] & & \dots & \\
x_3 & y_3 & & \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2} & & \\
& & \frac{y_4-y_3}{x_4-x_3} = f[x_3,x_4] & & & \\
x_4 & y_4 & & & &
\end{array}
$$

After we have constructed the table we can find the divided difference we want by looking at the top diagonal

$$
\begin{array}{llllll}
x_0 & y_0 & & & & \\
& & f[x_0,x_1] & & & \\
x_1 & y_1 & & f[x_0,x_1,x_2] & & \\
& & \dots & & f[x_0,x_1,x_2,x_3] & \\
x_2 & y_2 & & \dots & & f[x_0,x_1,x_2,x_3,x_4] \\
& & \dots & & \dots & \\
x_3 & y_3 & & \dots & & \\
& & \dots & & & \\
x_4 & y_4 & & & &
\end{array}
$$

**Case of equidistant points**

Bla bla bla the formula becomes

$$P(t) = a_0 + a_1(t-0) + a_2(t-0)(t-1) + \cdots + a_n \prod_{i=0}^{n-1}(t-i).$$

where in this case

$$a_k = \frac{\nabla^k[y](x_k)}{k!}.$$

and

$$x = x_0 + th.$$

where $\nabla^k[y]$ is the discrete difference.

$$\nabla[y](x_i) = y(x_i + h) - y(x_i).$$

and the backwards formula is

$$P(t) = f_n + A_1 + A_2 + \cdots + A_n.$$

where

$$A_i = \frac{\bar{\nabla}^i f_n}{i!} \prod_{j=n-i+1}^{n} (t-j).$$

**Definition 1.2.3: Discrete Difference**

Forward discrete difference:

$$
\begin{aligned}
\nabla[y](x_i) &= y(x_i + h) - y(x_i) \\
\nabla^2[y](x_i) &= \nabla[y](x_i + h) - \nabla[y](x_i) \\
&= y(x_i + 2h) - 2y(x_i + h) + y(x_i) \\
\nabla^k[y](x_i) &= \nabla\left(\nabla^{k-1}[y](x_i)\right)
\end{aligned}
$$

Backwards discrete difference:

$$
\begin{aligned}
\bar{\nabla}[y](x_i) &= y(x_i) - y(x_i - h) \\
\bar{\nabla}^k[y](x_i) &= \bar{\nabla}\left(\bar{\nabla}^{k-1}[y](x_i)\right)
\end{aligned}
$$

$$
\begin{array}{cccccc}
x_0 & y_0 & & & & \\
 & & \nabla[y](x_i) & & & \\
x_1 & y_1 & & \nabla^2[y](x_i) & & \\
 & & \dots & & \nabla^3[y](x_i) & \\
x_2 & y_2 & & \dots & & \nabla^4[y](x_i) \\
 & & \dots & & \dots & \\
x_3 & y_3 & & \dots & & \\
 & & \dots & & & \\
x_4 & y_4 & & & &
\end{array}
$$

### 1.2.3 Error due to polynomial interpolation

Let $f(x)$ be of class $C^{n+1} \quad \forall x \in [a, b]$ and let the polynomial $P(x)$ interpolate it.

The error function is bounded by

$$
|\text{Error}| = |f(x) - P(x)| \leqslant \frac{\left|\prod_{i=0}^{n}(x - x_i)\right|}{(n+1)!} \sup_{x \in [a,b]} \left| f^{(n+1)}(x) \right|.
$$

### 1.2.4 Hermite Interpolation

**Definition 1.2.4: Hermite interpolation formula**

Consider $(n+1)$ sets of point $(x_i, y_i, y_i')$ representing $f(x)$ ($y_i = f(x_i)$ and $y_i' = f'(x_i)$), the hermite polynomial $P(x)$ interpolates $f(x)$ such that $P'(x) = f'(x)$.

$$
P(x) = \sum_{i=0}^{n} h_i(x) y_i + \sum_{i=0}^{n} k_i(x) y_i'.
$$

where

$$
\begin{aligned}
h_i(x) &= \left(1 - 2(x - x_i)\ell_i'(x_i)\right) \ell_i^2(x) \\
k_i(x) &= (x - x_i)\ell_i^2(x) \\
\ell_i(x) &= \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}
\end{aligned}
$$

**Theorem 1.2.2** Error due to Hermite interpolation

$$|\text{Error}| = |f(x) - P(x)| \leqslant \frac{\left|\prod_{i=0}^{n}(x - x_i)^2\right|}{(2n+2)!} \sup_{x\in[a,b]} |f^{(2n+2)}(x)|.$$

**Existence**

***Proof:***

$$P(x) = \sum_{i=0}^{n} h_i(x)y_i + \sum_{i=0}^{n} k_i(x)y_i'.$$

where

$$\begin{aligned}
h_i(x) &= \left(1 - 2(x - x_i)\ell_i'(x_i)\right)\ell_i^2(x) \\
k_i(x) &= (x - x_i)\ell_i^2(x) \\
\ell_i(x) &= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}
\end{aligned}$$

Let $i \neq j$.

$$\begin{aligned}
k_i(x_j) &= (x_j - x_i)\ell_i^2(x_j) = 0 \\
k_i(x_i) &= (x_i - x_i)\ell_i^2(x_i) = 0
\end{aligned}$$

and

$$\begin{aligned}
h_i(x_j) &= (1 - 2(x_j - x_i)\ell_i'(x_i))\ell_i^2(x_j) = 0 \\
h_i(x_j) &= (1 - 2(x_i - x_i)\ell_i'(x_i))\ell_i^2(x_i) = 1
\end{aligned}$$

We conclude that $P(x_i) = f(x_i)$

Now we have to prove that $P'(x_i) = f'(x_i)$

$$\begin{aligned}
h_i'(x) &= -2\ell_i'(x_i)\ell_i^2(x) + 2(1 - 2(x - x_i)\ell_i'(x_i))\ell_i(x)\ell_i'(x) \\
k_i'(x) &= \ell_i^2(x) + 2(x - x_i)\ell_i(x)\ell_i'(x)
\end{aligned}$$

$$\begin{aligned}
h_i'(x_j) &= -2\ell_i'(x_i)\ell_i^2(x_j) + 2(1 - 2(x_j - x_i)\ell_i'(x_i))\ell_i(x_j)\ell_i'(x_j) = 0 \\
h_i'(x_i) &= -2\ell_i'(x_i)\ell_i^2(x_i) + 2(1 - 2(x_i - x_i)\ell_i'(x_i))\ell_i(x_i)\ell_i'(x_i) = 0
\end{aligned}$$

$$\begin{aligned}
k_i'(x_j) &= \ell_i^2(x - j) + 2(x_j - x_i)\ell_i(x_j)\ell_i'(x_j) = 0 \\
k_i'(x_j) &= \ell_i^2(x - i) + 2(x_i - x_i)\ell_i(x_i)\ell_i'(x_i) = 1
\end{aligned}$$

$\therefore P'(x_i) = f'(x_i)$ ◻

**Uniqueness**

***Proof:*** Suppose that there exists 2 polynomials $P$ and $Q$ of degree $n \leqslant 2n+1$ such that $P(x_i) = Q(x_i) = f(x_i)$ and $P'(x_i) = Q'(x_i) = f'(x_i)\ \forall i = 0, 1, \ldots, n$.
Let $R(x) = P(x) - Q(x)$.
$R = 0\ (n+1)$ times $\Rightarrow$ according to Rolle's theorem $\exists\, n$ points $\neq x_i \Big/ R' = 0$
$R' = 0\ n$ times as a consequence of Rolle's theorem then

$$\left.\begin{array}{l} R'(x) = 0\ (2n+1) \text{ times} \\ R'(x) \text{ is of degree } 2n \end{array}\right\} R'(x) = 0\ \forall x.$$

$$R'(x) = 0 \Rightarrow R(x) = \text{cnst} \quad \text{and} \quad R(x_i) = P(x_i) - Q(x_i) = 0 \Rightarrow \text{cnst} \ = 0.$$

$$R(x) = P(x) - Q(x) = 0\ \forall x.$$

$\therefore P(x) = Q(x)$ □

# Chapter 2

# Finding $f(x) = 0$

We will assume that every function is defined in the interval $I = [a, b]$ and that every $x_0 \in I$

## 2.1 Bisection Method

Suppose that $f$ is a continuous and monotone function on the domain $I = [a, b]$ such that $f(a)f(b) < 0 \Rightarrow \exists r \in ]a, b[: \ f(r) = 0$.

At each step in the algorithm, in an iteration we let $c = (a + b)/2$, then we check the value of $f(c)$, if it is 0 then we are done.
However when it is not, then we define a new interval such that

$$I = \begin{cases} [a, c] & \text{if } f(c)f(a) < 0 \\ [c, b] & \text{if } f(c)f(b) < 0 \end{cases}.$$

We repeat this step until the length of the interval reaches a certain number (for example $|b - a| < 10^{-5}$), at this point we stop and the best guess for the root would be $(a + b)/2$

### Error of the Bisection Method

After $n$ iterations, the error of the approximated root would be

$$\text{Error} \leqslant \frac{|b - a|}{2^{n+1}}.$$

## 2.2 Lagrange Method

Suppose that $f$ is a continuous and monotone function on the domain $I = [a, b]$ such that $f(a)f(b) < 0 \Rightarrow \exists r \in ]a, b[: \ f(r) = 0$.

The starting value of $x_0$ depends on the value of $f$

$$x_0 = \begin{cases} a & \text{if } f(a)f''(a) < 0 \\ b & \text{if } f(b)f''(b) < 0 \end{cases}.$$

then we can find a new guess $x$ depending on the value of $x_0$

- if $x_0 = a$

$$x_1 = x_0 - \frac{b - x_0}{f(b) - f(x_0)} f(x_0).$$

- if $x_0 = b$

$$x_1 = x_0 - \frac{a - x_0}{f(a) - f(x_0)} f(x_0).$$

**Error from Lagrange Method**

For the first iteration

$$\text{Error} \leqslant \sup_{x\in[a,b]} |f''(x)| \frac{(b-a)^2}{8}.$$

For the second iteration

$$M_2 = \sup_{x\in[a,b]} |f''(x)|.$$

- if $x_0 = a$

$$\text{Error} \leqslant \frac{M_2}{8}\left|\frac{(b-x_0)^3}{f(b)-f(x_0)}\right|.$$

- if $x_0 = b$

$$\text{Error} \leqslant \frac{M_2}{8}\left|\frac{(a-x_0)^3}{f(a)-f(x_0)}\right|.$$

## 2.3 Newton Method

Suppose that $f$ is a continuous and monotone function on the domain $I = [a, b]$ such that $f(a)f(b) < 0 \Rightarrow \exists r \in ]a, b[: \ f(r) = 0$.

The starting value of $x_0$ depends on the value of $f$

$$x_0 = \begin{cases} a & \text{if } f(a)f''(a) > 0 \\ b & \text{if } f(b)f''(b) > 0 \end{cases}.$$

Then the new guess for the root would be

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

### 2.3.1 Improved Newton Method

To improve the method we first let $\eta = b - a$, and we define condition

$$\frac{\eta M_2}{2|f'(x_0)|} < 1.$$

if the condition is not satisfied we need to choose another interval $[a_1, b_1] \subset I$ where $f(a_1)f(b_1) < 0$

**Error due to Newton Method**

For one iteration

$$\text{Error} =\leqslant \frac{\eta^2 M_2}{2|f'(x_0)|} \quad \text{where} \quad M_2 = \sup_{x\in[x_0-\eta, x_0+\eta]} |f''(x)|.$$

## 2.4 Fixed Point Iteration Method

If a function can be converted to the form $x = g(x)$ along with the sequence $x_{n+1} = g(x_n)$ with initial guess $x_0$, then it is called a fixed point scheme.
The scheme converges if

- $\forall x \in [a, b] : \ g(x) \in [a, b]$
- $g$ is strictly contracting meaning that $\exists \varepsilon \in \mathbb{R}\ 0 \leqslant \varepsilon < 1$

$$\forall x, y \in [a, b], \ |g(x) - g(y)| \leqslant \varepsilon |x - y|.$$

then $\forall x_0$ the sequence converges to $l \in [a, b]$

**Note:-**

$$\sup_{x\in[a,b]} |g'(x)| = L < 1 \Rightarrow g(x) \text{ is strictly contracting.}$$

**Note:-**

Let $l$ be the solution to $g(l) = l$

- If $|g'(l)| < 1$ then there exists an interval $I$ containing $l$ for which the sequence converges to $l$
- If $|g'(l)| > 1$ then the sequence diverges

## 2.5 Order of Convergence

Order of convergence (Rate of convergence) tells us how the error decreases between 2 iterations. The order of convergence $p$ of a sequence is defined to be

$$\lim_{n\to+\infty} \left|\frac{x_{n+1} - l}{(x_n - l)}\right| \in \mathbb{R}_+^*.$$

**Note:-**

The order of convergence of

- Lagrange Method
$$g'(l) = \frac{(b-l)^2}{2f(b)} f''(c).$$
If $f''(c) \neq 0$ then $g'(l) \neq 0$ then the order is 1.
- Newton method, if $g'(l) = 0$ then the order is at least 2.

**Note:-**

We stop the iteration method when

- First case $g'(x) < 0$, then we stop iteration when
$$|x_{n+1} - r| < \varepsilon.$$
- Second case $g'(x) > 0$, then we stop iteration when
$$|f(x_n)| < \eta.$$
where
$$\eta = \varepsilon \inf |f'(x)|.$$

## 2.6 Polynomial Shenanigans

### 2.6.1 Roots of $x^3 + px + q = 0$

Let $y_1(x) = x^3 + px$ and $y_2(x) = -q$

- $p \geqslant 0 \Rightarrow \exists 1$ root
- $p < 0$ then we have 3 separate cases

$$27q^2 + 4p^3 \begin{cases} = 0 & \text{we have 2 separate real roots (one double and one single)} \\ > 0 & \text{we have one real root} \\ < 0 & \text{we have 3 separate real roots} \end{cases}.$$

### 2.6.2 Roots of $x^3 + ax^2 + bx + c = 0$

If we replace $x$ with $X + h$ where $h = -\frac{a}{3}$, we can get the cubic in the form

$$X^3 + PX + Q = 0.$$

where

$$P = -\frac{a^2}{3} + b$$
$$Q = \frac{2a^3}{27} - \frac{ab}{3} + c$$

### 2.6.3 Roots of $x^4 + ax^3 + bx^2 + cx + d = 0$

If we replace $x$ with $X + h$ where $h = -\frac{a}{4}$, we can get the quartic in the form

$$X^4 + PX^2 + QX + R = 0.$$

where

$$P = -\frac{3a^2}{8} + b$$
$$Q = \frac{a^3}{8} - \frac{ab}{2} + c$$
$$R = -\frac{3a^4}{256} - \frac{ac}{4} + d$$

Let the circle $C$ be the circle of radius $\left(-\frac{Q}{2}, \frac{1-P}{2}\right)$ and of radius $\sqrt{\left(\frac{P-1}{2}\right)^2 + \frac{Q^2}{4} - R}$.

The roots of the polynomial $X^4 + PX^2 + QX + R = 0$ are the intersection of the circle $C$ and the parabola $Y = X^2$

# Chapter 3

# Numerical Intergration

Let $f$ be a continuous function on $[a, b]$ and $I = \int_a^b f(x)\,\mathrm{d}x$

## 3.1 Rectangle method

We sample the domain of $f$ in to $n$ equal subintervals $(x_i - x_{i+1} = \frac{b-a}{n} = h)$. The approximated value of $I$ becomes

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{n}\left[\sum_{i=0}^{n-1} f(x_i)\right].$$

such that $x_0 = a$ and $x_1 = b$

The error associated with this approximation is

$$|\varepsilon| \leqslant \frac{M_1}{2n}(b-a)^2.$$

where

$$M_1 = \sup_{[a,b]} |f'(x)|.$$

## 3.2 Trapezoid method

Same sampling as before.

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{2n}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right).$$

The error associated with this approximation is

$$|\varepsilon| \leqslant \frac{M_2}{12n^2}\left|(b-a)^3\right|.$$

where

$$M_2 = \sup_{[a,b]} |f''(x)|.$$

## 3.3 Simpson's rule

You get the point by now

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6n}\left(f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i) + 4\left(f\left(\frac{a+x_1}{2}\right) + f\left(\frac{x_1+x_2}{2}\right) + \cdots f\left(\frac{x_{n-1}+b}{2}\right)\right)\right).$$

The error associated with this approximation is

$$|\varepsilon| \leqslant \frac{\left|(b-a)^5\right|}{n^4}\frac{M_4}{2880}.$$

where

$$M_4 = \sup_{[a,b]}\left|f^{(4)}(x)\right|.$$

## 3.4 Newton Cote's method

Let $P$ be the Lagrange polynomial that interpolates the function $f$ at $(n+1)$ points $(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n)$.

$$P(s) = \sum_{i=0}^{n} \ell_i(s) f(x_i).$$

The approximated value of $I$ is

$$\int_a^b f(x)\,\mathrm{d}x \approx (b-a)\sum_{i=0}^{n} H_i f_i.$$

where

$$H_i = \frac{1}{n}\int_0^n \ell_i(s)\,\mathrm{d}s.$$