Numerical Analysis

Xiaojing Ye, Math & Stat, Georgia State University

Fall 2022

Section 1

Review of Calculus

Limit and continuity

Definition (Euclidean ball)

An **(open) ball in** $\mathbb{R}^n$ is $B_r(x_0) := \{x \in \mathbb{R}^n : |x - x_0| < r\}$.

Definition (Limit of a function)

The **limit** of $f(x)$ as x approaches x_0 is L if $\forall\, \epsilon > 0$, $\exists\, \delta > 0$ such that for all $x \in B_\delta(x_0)$ there is

$$|f(x) - L| < \epsilon.$$

Definition (Continuous functions)

f is **continuous at** x_0 if $\lim_{x \to x_0} f(x) = f(x_0)$.
f is **continuous in** X if f is continuous at every $x \in X$.

Limit and continuity

Definition (Limit of a sequence)

A sequence $\{x_n : n \in \mathbb{N}\}$ has **limit** x if $\forall\, \epsilon > 0$, $\exists\, N \in \mathbb{N}$, such that $|x_n - x| < \epsilon$ for all $n \geq N$.

Theorem

The following two statements are equivalent:

- *f is continuous at x.*
- *If $x_n \to x$, then $f(x_n) \to f(x)$.*

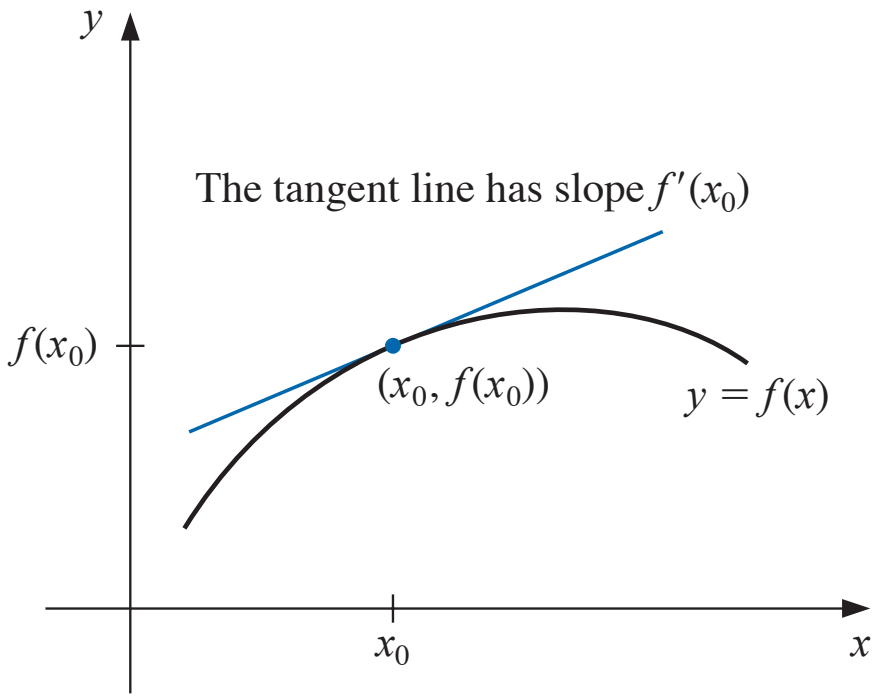
Differentiability

Definition (Derivative of a function)

f is **differentiable** at x_0 if the following limit exists:

$$\lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

The value of this limit is called the **derivative** of f at x_0.



Differentiability

Theorem

f is differentiable at $x \Longrightarrow f$ is continuous at x.

Theorem (Rolle's Theorem)

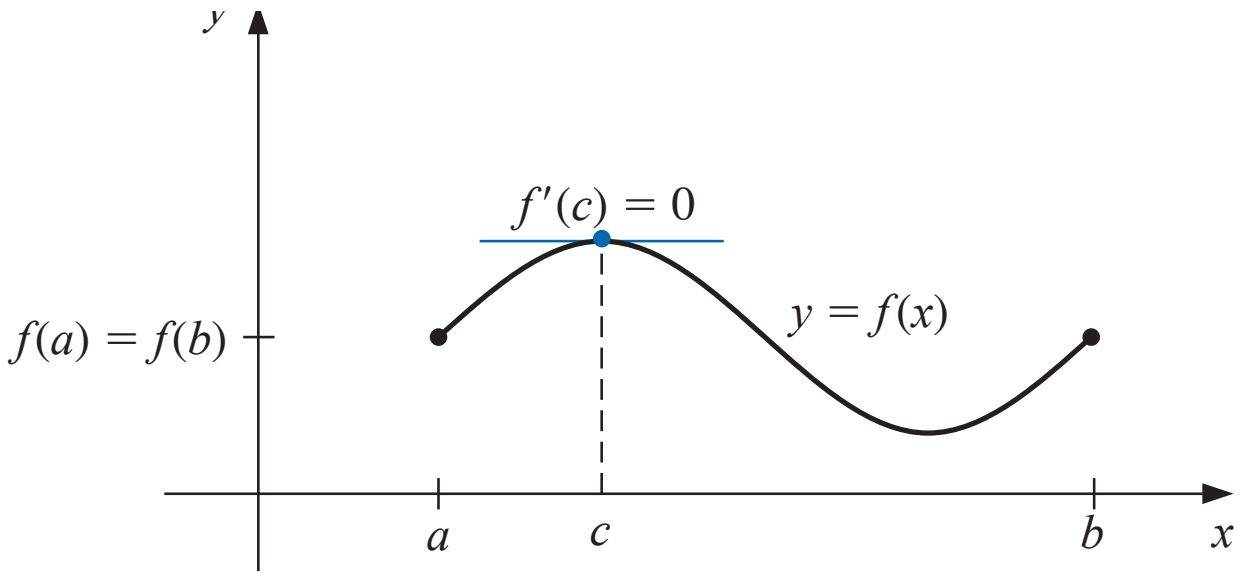
Suppose $f \in C[a, b]$, f is differentiable in (a, b) and $f(a) = f(b)$, then $\exists\, c \in (a, b)$ such that $f'(c) = 0$.

Proof of Rolle's theorem.

Hint: $f \in C[a, b]$ implies that f attains max or min in $[a, b]$ by the extreme value theorem (see soon). □

Rolle's theorem

Illustration of the Rolle's theorem:

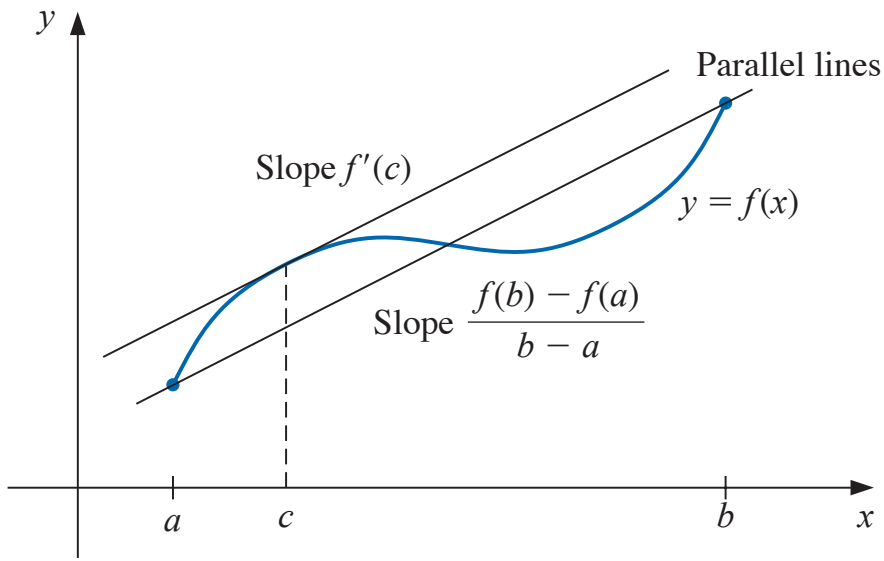


Mean Value Theorem

Theorem (Mean Value Theorem)

If $f \in C[a,b]$ and f is differentiable on (a,b), then $\exists\, c \in (a,b)$ such that

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$



Proof.

Define $g(x) = f(x) - f(a) - \frac{f(b)-f(a)}{b-a}(x - a)$. Then $g(a) = g(b) = 0$. Apply Rolle's theorem to g. □

Extreme Value Theorem

Theorem (Extreme Value Theorem)

If $f \in C[a,b]$, *then* $\exists\, c_1, c_2 \in [a,b]$ *such that*

$$f(c_1) \leq f(x) \leq f(c_2)$$

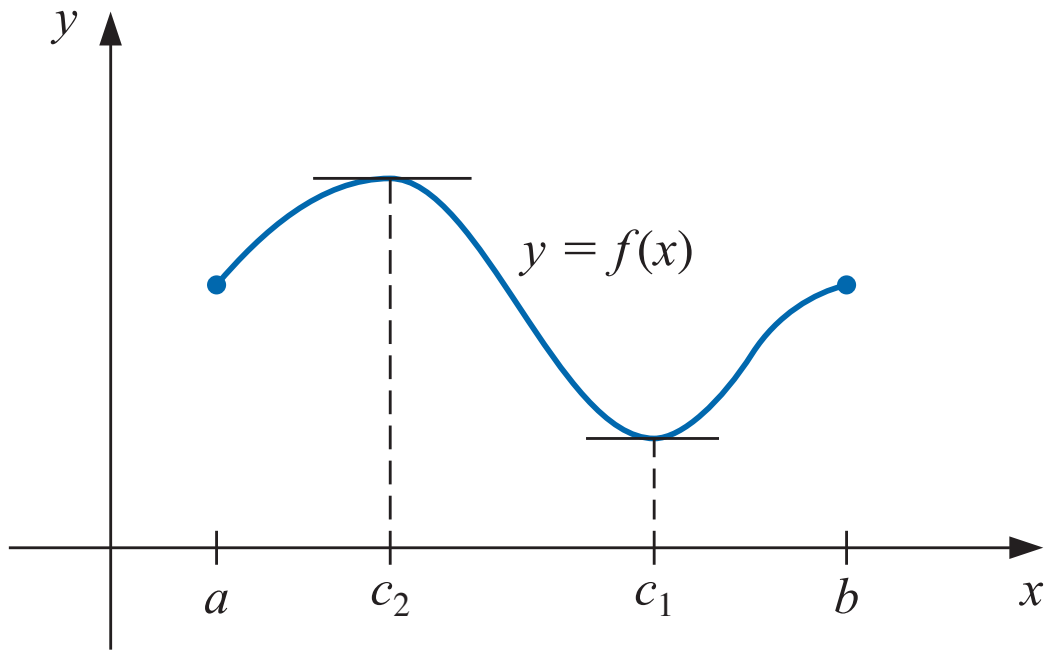
for all $x \in [a,b]$. *In addition, if* f *is differentiable in* (a,b), *then* c_1 *and* c_2 *occur either at* a, b, *or where* $f' = 0$.

Proof.

Suppose $f(x_k) \to \inf_{a \leq x \leq b} f(x)$, then $\exists$ subseq $x_{k_j} \to c_1 \in [a,b]$ such that $f(x_{k_j}) \to f(c_1)$ ($\because$ f continuous). Hence we have $f(c_1) = \min_{a \leq x \leq b} f(x)$. $\square$

Extreme Value Theorem

Illustration of the Extreme Value Theorem



Generalized Rolle's theorem

Theorem (Generalized Rolle's Theorem)

Suppose $f \in [a, b]$ *and is* n *times differentiable. Let* $\{x_0, \ldots, x_n\}$ *be a partition of* $[a, b]$*, i.e.,* $a = x_0 < x_1 < \cdots < x_n = b$*, such that* $f(x_i) = 0$ *for all* $i = 1, \ldots, n$*, then* $\exists\, c \in (a, b)$ *such that* $f^{(n)}(c) = 0$*.*

Proof.

By Rolle's theorem, $\exists\, y_1, \ldots, y_n$ s.t. $x_0 < y_1 < x_1 < \cdots < y_n < x_n$ and $f'(y_i) = 0$ for $i = 1, \ldots, n$. Keep applying Rolle's theorem for another $n - 1$ times to show that $\exists\, c \in (a, b)$ s.t. $f^{(n)}(c) = 0$. □

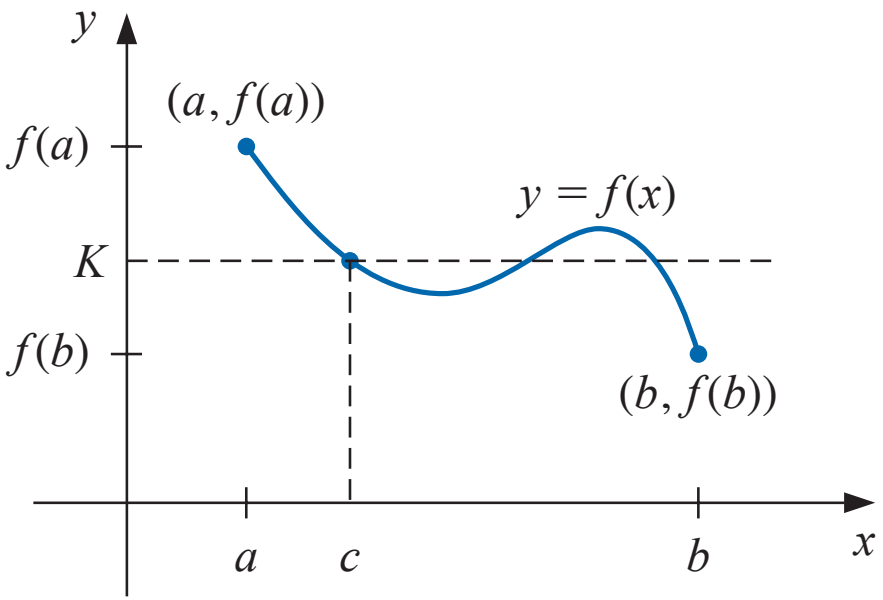
Intermediate value theorem

Theorem (Intermediate Value Theorem (IVT))

If $f \in C[a, b]$ and k is a number between $f(a)$ and $f(b)$, then $\exists\, c \in (a, b)$ such that $f(c) = k$.

Proof.

By continuity of f on $[a, b]$. □



Example

Example (Application of IVT)

Show that $x^5 - 2x^3 + 3x^2 - 1 = 0$ has a solution in $[0, 1]$.

Solution. Set $f(x) = x^5 - 2x^3 + 3x^2 - 1$. Then we need to show that $\exists\, c \in [0, 1]$ such that $f(c) = 0$. Since $f(0) = -1$ and $f(1) = 1$, we know such c exists by IVT.

Integration

Definition (Riemann integral)

The **Riemann integral** of f on $[a, b]$ is the limit

$$\int_a^b f(x)\,\mathrm{d}x := \lim_{\max_i \Delta x_i \to 0} \sum_{i=1}^{n} f(z_i)\Delta x_i$$

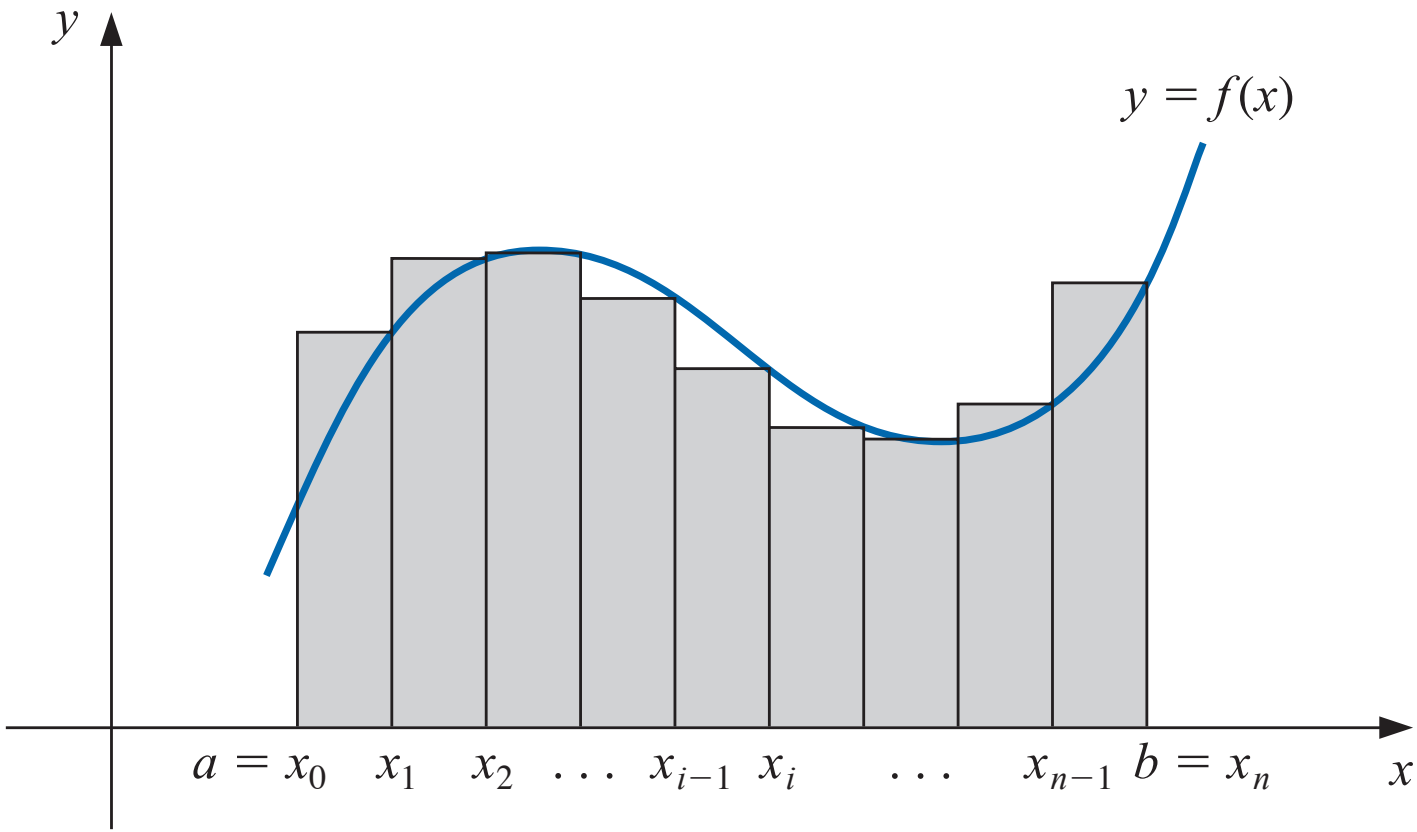
where $\{x_0, \ldots, x_n\}$ is a paritition of $[a, b]$, $\Delta x_i := x_i - x_{i-1}$ and z_i is arbitrary in $[x_{i-1}, x_i]$.

If $f \in C[a, b]$, this simply means

$$\int_a^b f(x)\,\mathrm{d}x = \lim_{n\to\infty} \frac{b-a}{n} \sum_{i=1}^{n} f(x_i)$$

where $\{x_0, \ldots, x_n\}$ is an equal partition of $[a, b]$ into n segments, $\Delta x_i = \frac{b-a}{n}$, $\forall\, i$.

Riemann integral



Mean value theorem for integrals

Theorem (Mean Value Theorem for Integrals)

Suppose $f \in C[a,b]$, and g is Riemann integrable over $[a,b]$ and does not change sign, then $\exists\, c \in (a,b)$ s.t.

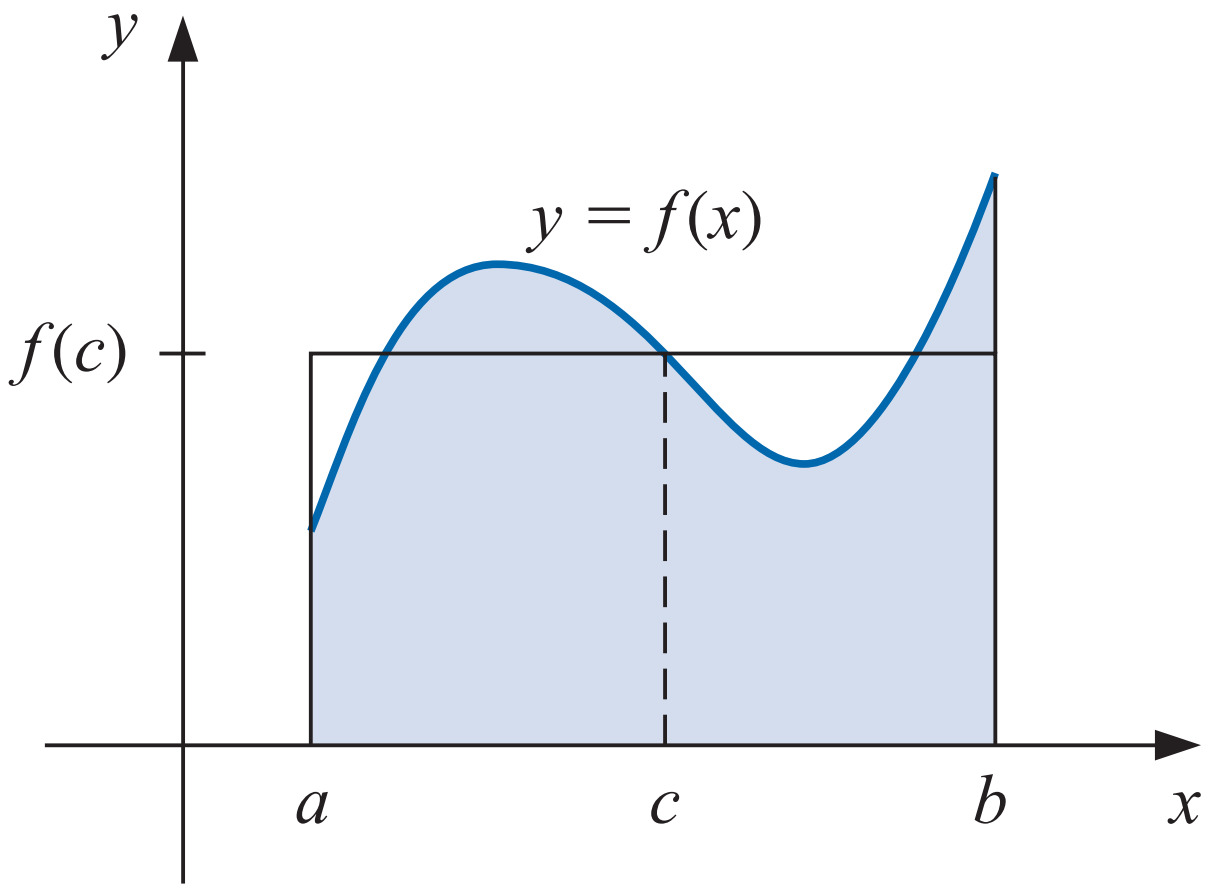
$$\int_a^b f(x)g(x)\,\mathrm{d}x = f(c)\int_a^b g(x)\,\mathrm{d}x$$

If $g(x) \equiv 1$, then $\exists\, c \in [a,b]$, s.t. $f(c) = \frac{1}{b-a}\int_a^b f(x)\,\mathrm{d}x$

Proof.

Hint: WLOG $g \geq 0$, then $m\int_a^b g(x)\,\mathrm{d}x \leq \int_a^b f(x)g(x)\,\mathrm{d}x \leq M\int_a^b g(x)\,\mathrm{d}x$ where m, M are min,max of f. So $m \leq r := \frac{\int_a^b f(x)g(x)\,\mathrm{d}x}{\int_a^b g(x)\,\mathrm{d}x} \leq M$. By IVT $\exists\, c \in [a,b]$ s.t. $f(c) = r$. □

Mean value theorem for integrals



Taylor series and polynomials

Theorem (Taylor's theorem)

Suppose $f \in C^n[a,b]$, $f^{(n+1)}$ exists in (a,b), $x_0 \in [a,b]$. Then for every $x \in (a,b)$, there exists a number $\xi(x)$ such that

$$f(x) = P_n(x) + R_n(x),$$

where $P_n(x)$ is a polynomial of degree n:

$$P_n(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n$$

and $R_n(x)$ is the remainder term:

$$R_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi(x))(x - x_0)^{n+1}.$$

Proof.

For any fixed x_0, x, define $r := \frac{f(x) - P_n(x)}{(x - x_0)^{n+1}}$ and $F(t) := f(t) - P_n(t) - r \cdot (t - x_0)^{n+1}$. Prove that $F(x_0) = F'(x_0) = \cdots = F^{(n)}(x_0) = 0$ and $F(x) = 0$. Then apply Rolle's theorem repeatedly to show $\exists\, \xi(x) \in (x_0, x)$ s.t. $F^{(n+1)}(\xi(x)) = 0$, i.e., $r = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}$. □

Example

Example (Taylor polynomial)

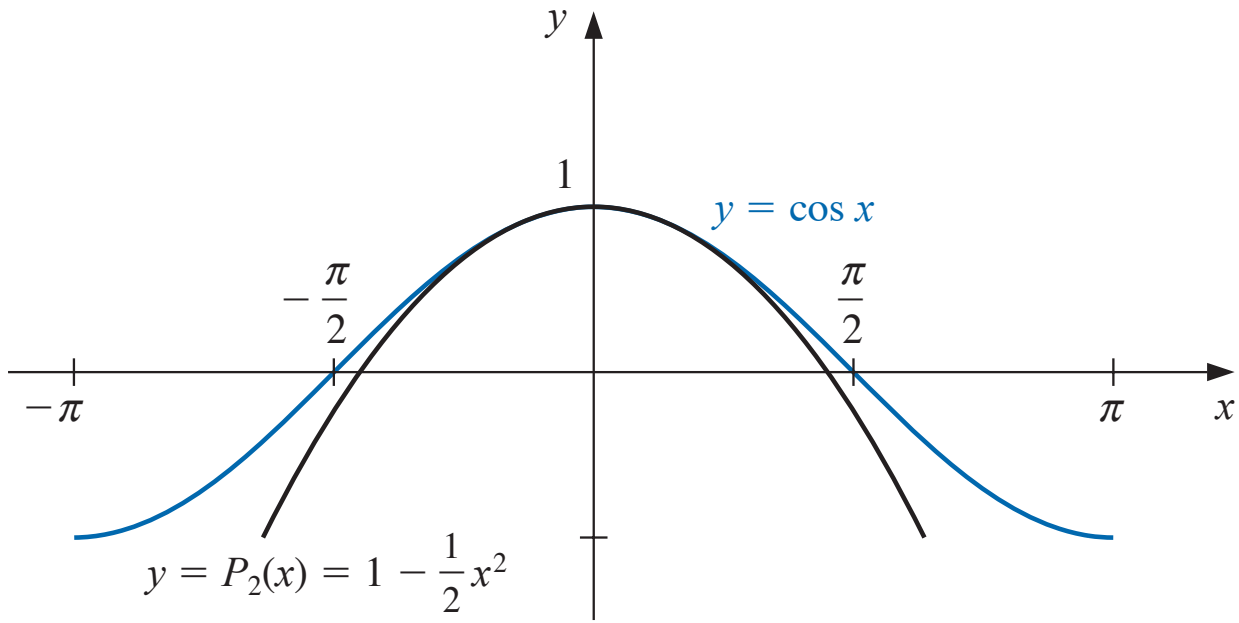
Let $f(x) = \cos x$ and $x_0 = 0$. Find $P_3(x)$, the Taylor polynomial of degree 3 (i.e., the polynomial by expanding f at x_0 to the 3rd order).

Solution. $f(x_0) = \cos(0) = 1$, $f'(x_0) = -\sin(0) = 0$, $f''(x_0) = -\cos(0) = 1$, $f'''(x_0) = \sin(0)$. So

$$P_3(x) = \sum_{k=0}^{3} f^{(k)}(x_0)(x - x_0)^k = 1 - \frac{1}{2}x^2$$

Taylor series and polynomials

Approximating $f(x) = \cos x$ by Taylor's polynomial $P_2(x)$:



Section 2

Solutions of Equations in One Variable (Root-Finding)

Root-finding

Definition (Roots of a function)

Let $f : \mathbb{R} \to \mathbb{R}$ (univariate), then x is called a **root**, or **zero**, of f if $f(x) = 0$.

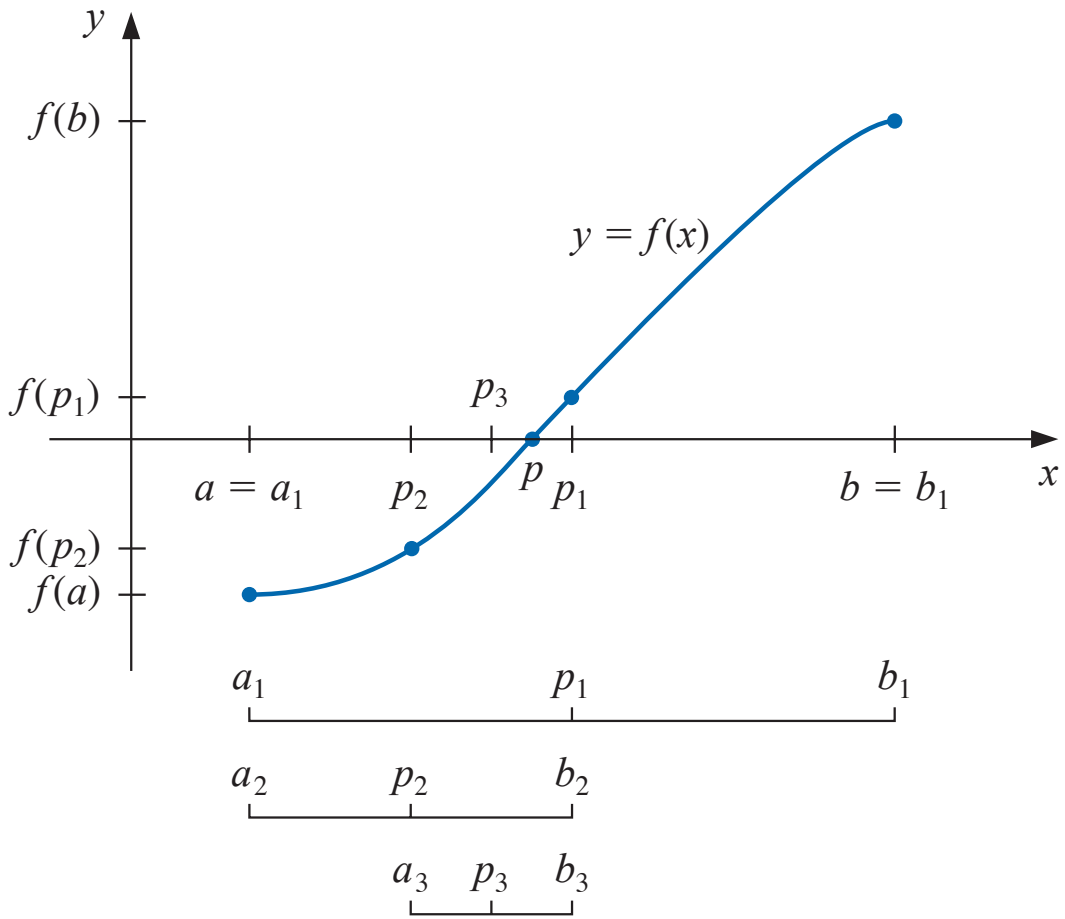
Example (Roots of a function)

Find the root(s) of $f(x)$ defined by
(a) $(x-1)(x+1)$;
(b) $(x-1)^2$;
(c) x^2+1;
(d) ax^2+bx+c;
(e) $\cos(x)$.

Question: Given a general function f, how can we find its root/roots?

Bisection method

Suppose f is *continuous* on $[a, b]$, and $f(a)f(b) < 0$ (WLOG $f(a) < 0, f(b) > 0$). Then f has at least one root in (a, b) by IVT.



Bisection method

Suppose f is *continuous* on $[a, b]$, and $f(a)f(b) < 0$ (WLOG $f(a) < 0, f(b) > 0$). Then f has at least one root in (a, b) by IVT.

Bisection method

- **Input.** Endpoints a, b. Tolerance ϵ_{tol}. Maximum number of iterations $N_{\max}$. Set iteration counter $N = 1$.
- While $N \leq N_{\max}$, do
  1. Set $p = \frac{a+b}{2}$, compute $f(p)$. If $f(p) = 0$ or $b - a < \epsilon_{\text{tol}}$, break.
  2. If $f(p) > 0$, set $b = p$. If $f(p) < 0$, set $a = p$.
  3. $N \leftarrow N + 1$.
- **Output.** If $i = N_{\max}$, print("Maximum iteration reached."). Return p.

Termination condition

Bisection method can run forever if we do not set termination condition (e.g., ϵ_{tol}, $N_{\max}$).

Common choices of termination condition:

- Fixed number of iterations $N_{\max}$.
- $|p_N - p_{N-1}| < \epsilon_{\text{tol}}$
- $|f(p_N)| < \epsilon_{\text{tol}}$
- $\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon_{\text{tol}}$

Example

Example (Bisection method)

$f(x) = x^3 + 4x^2 - 10$. Find a root in $[1, 2]$ using the bisection method.

Solution. Hint: First check if $f(1)f(2) < 0$ (if not, bisection method may not apply). Then apply bisection.

Bisection method

Theorem

Suppose $f \in C[a, b]$ and $f(a)f(b) < 0$, then p_n generated by the bisection method converges to p, a root of f, with $|p_n - p| < \frac{b-a}{2^n}$

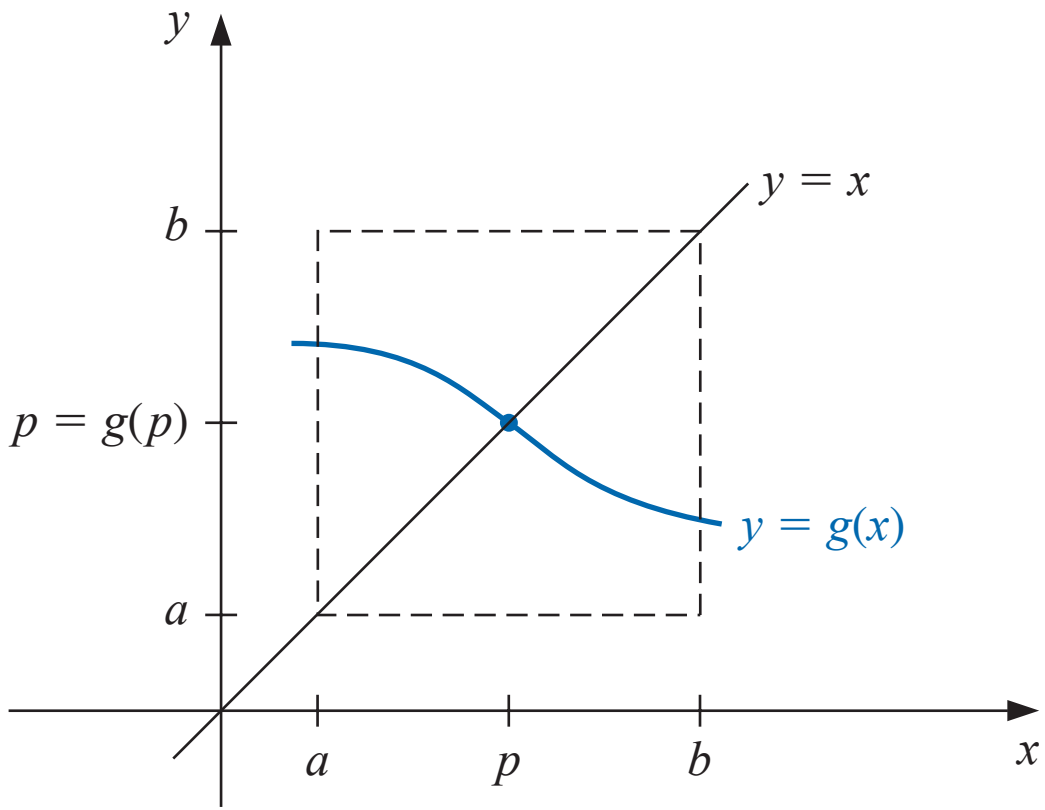
Drawbacks of the bisection method:

- inefficient
- may discard some roots

Fixed point iteration

Definition (Fixed point)

Let $g : \mathbb{R} \to \mathbb{R}$, then p is a **fixed point** of g if $g(p) = p$.



Fixed point

Example (Fixed point and root)

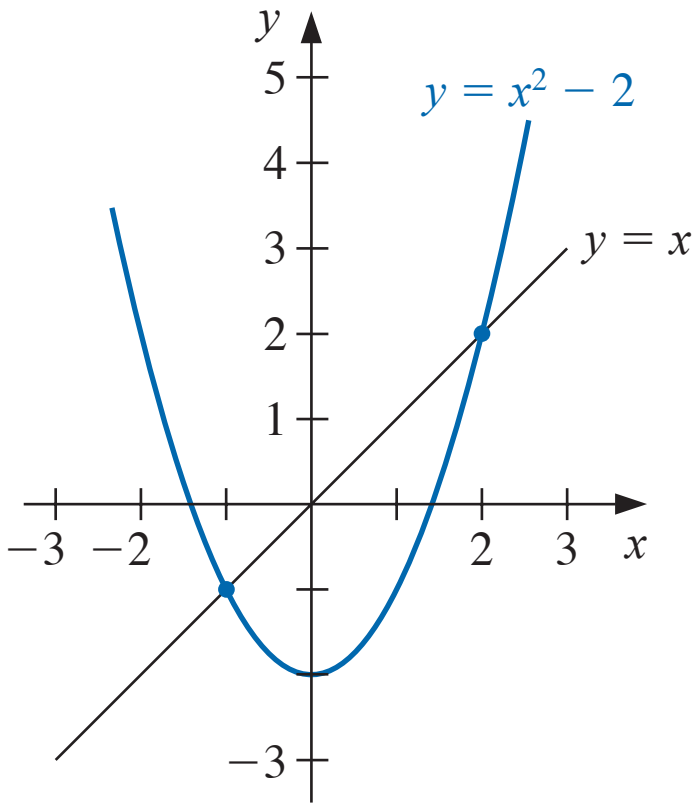
Suppose $\alpha \neq 0$. Show that p is a root of $f(x)$ iff p is a fixed point of $g(x) := x - \alpha f(x)$

Example

Example (Fixed point)

Find the fixed point(s) of $g(x) = x^2 - 2$.

Solution. p is a fixed point of g if $p = g(p) = p^2 - 2$. Solve for p to get $p = 2, -1$.



Fixed point theorem

Theorem (Fixed point theorem)

1. *If* $g \in C[a, b]$ *and* $a \leq g(x) \leq b$ *for all* $x \in [a, b]$, *then* g *has at least one fixed point in* $[a, b]$.
2. *If, in addition,* g' *exists in* $[a, b]$, *and* $\exists\, k < 1$ *such that* $|g'(x)| \leq k < 1$ *for all* x, *then* g *has a unique fixed point in* $[a, b]$.

Fixed point theorem

Proof.

1. If $g(a) = a$ or $g(b) = b$, then done. Otherwise, $g(a) > a$ and $g(b) < b$. Define $f(x) = x - g(x)$, then $f(a) = a - g(a) < 0$, and $f(b) = b - g(b) > 0$. By IVT and f is continuous, $\exists\, p \in (a, b)$ s.t. $f(p) = 0$, i.e., $p - g(p) = 0$.
2. If $\exists\, p, q \in [a, b]$ are two distinct fixed points of g, then $\exists\, \xi \in (p, q)$ s.t.

$$1 = \frac{p - q}{p - q} = \left| \frac{g(p) - g(q)}{p - q} \right| = |g'(\xi)| \leq k < 1$$

by MVT. Contradiction.

□

Example

Example (Application of Fixed Point Theorem)

Show that $g(x) = \frac{x^2-1}{3}$ has a unique fixed point in $[-1, 1]$.

Proof.

First we need show $g(x) \in [-1, 1]$, $\forall\, x \in [-1, 1]$. Find the max and min values of g as $-\frac{1}{3}$ and 0 (Hint: find critical points of g first). So $g(x) \in [-\frac{1}{3}, 0] \subset [-1, 1]$.

Also $|g'(x)| = |\frac{2x}{3}| \leq \frac{2}{3} < 1$, $\forall\, x \in [-1, 1]$, so g has unique fixed point in $[-1, 1]$ by FPT. □

Remark: We can solve for this fixed point: $p = g(p) = \frac{p^2-1}{3} \Longrightarrow p = \frac{3-\sqrt{13}}{2}$.

Example

Example (Fixed Point Theorem – Failed Case 1)

$g(x) = \frac{x^2-1}{3}$ has a unique fixed point in $[3, 4]$. But we can't use FPT to show this.

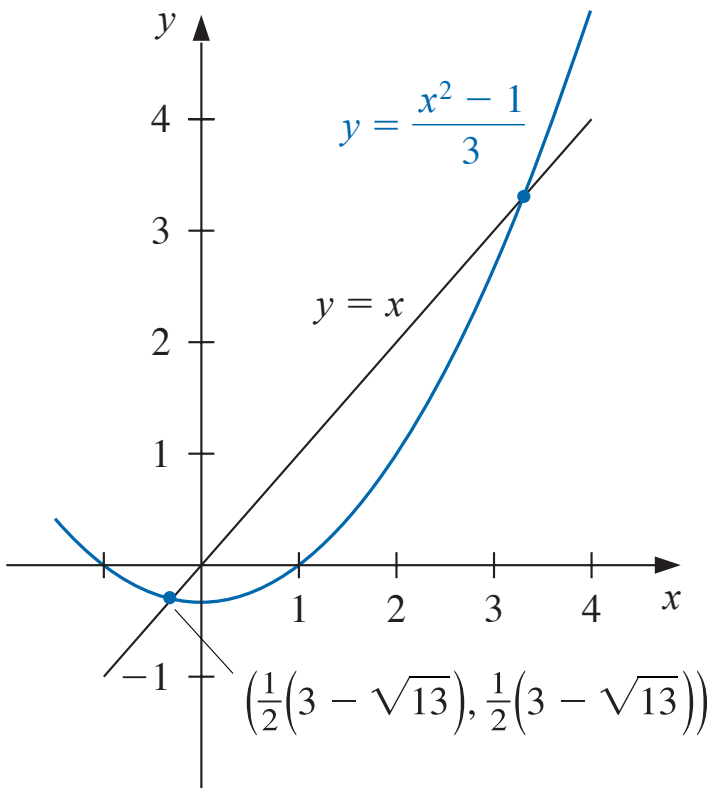
Remark: Note that there is a unique fixed point in $[3, 4]$ ($p = \frac{3+\sqrt{13}}{2}$), but $g(4) = 5 \notin [3, 4]$, and $g'(4) = 8/3 > 1$ so we cannot apply FPT here.

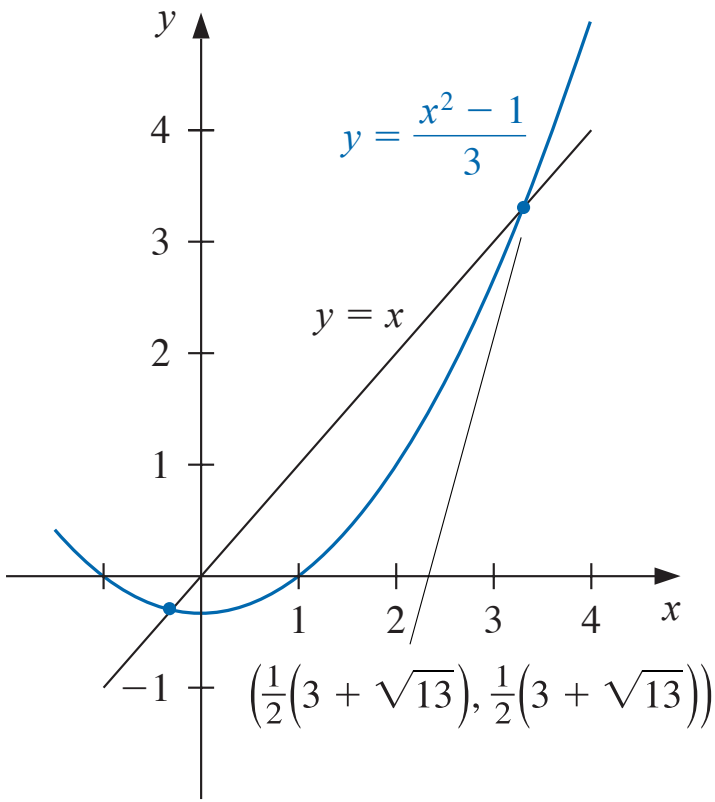
From this example, we know FPT provides a **sufficient but not necessary** condition.

Example

Example (Fixed Point Theorem – Failed Case 1)

$g(x) = \frac{x^2-1}{3}$ has a unique fixed point in $[3, 4]$. But we can't use FPT to show this.





Example

Example (Fixed Point Theorem – Failed Case 2)

We can use FPT to show that $g(x) = 3^{-x}$ must have FP on $[0, 1]$, but we can't use FPT to show if it's unique (even though the FP on $[0, 1]$ is unique in this example).

Solution. $g'(x) = (3^{-x})' = -3^{-x} \ln 3 < 0$, therefore $g(x)$ is strictly decreasing on $[0, 1]$. Also $g(0) = 3^0 = 1$ and $g(1) = 3^{-1}$, so $g(x) \in [0, 1]$, $\forall\, x \in [0, 1]$. So a FP exists by FPT.

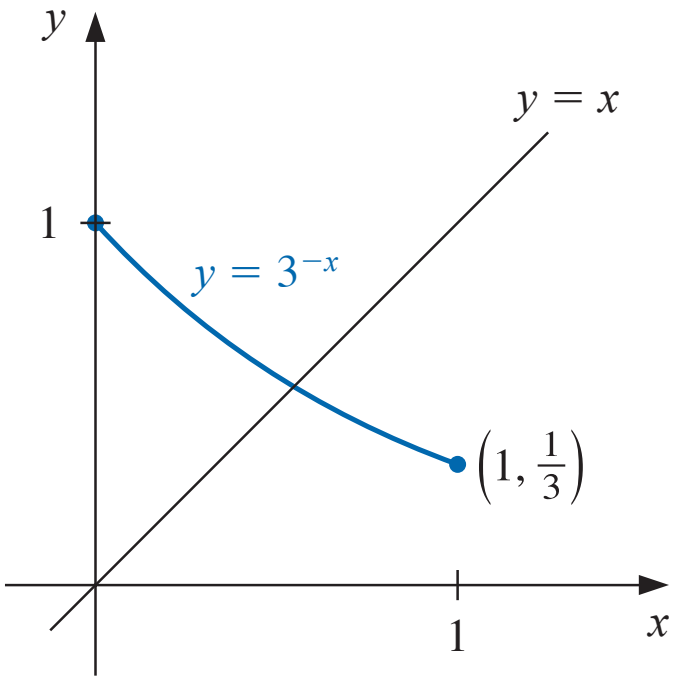
However, $g'(0) = -\ln 3 \approx -1.098$, so we do not have $|g'(x)| < 1$ over $[0, 1]$. Hence FPT does not apply.

Nevertheless, the FP must be unique since g strictly decreases and intercepts with $y = x$ line only once.

Example

Example (Fixed Point Theorem – Failed Case 2)

We can use FPT to show that $g(x) = 3^{-x}$ must have FP on $[0, 1]$, but we can't use FPT to show whether it is unique (even though the FP on $[0, 1]$ is indeed unique in this example).



Fixed point iteration

We now introduce a method to find a fixed point of a *continuous* function g.

Fixed point iteration:
Start with an initial guess p_0, recursively define a sequence p_n by

$$p_{n+1} = g(p_n)$$

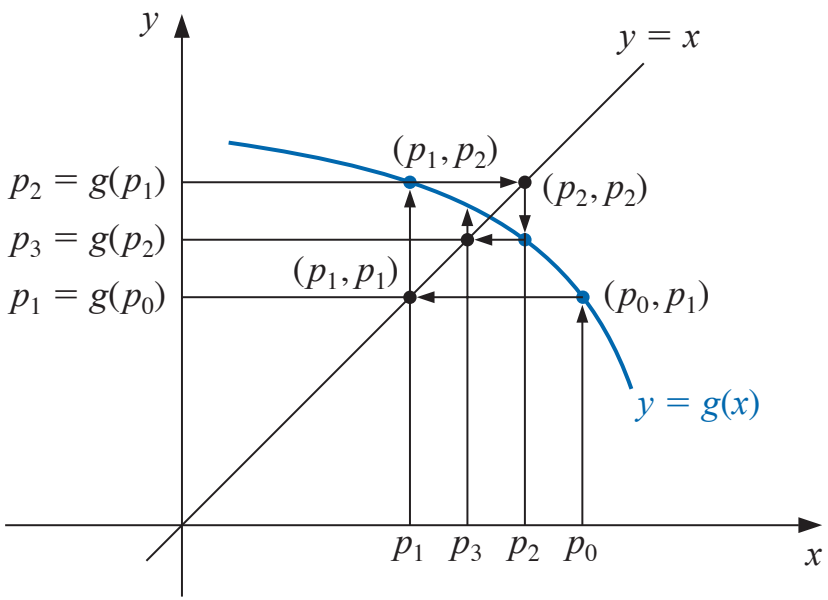
If $p_n \to p$, then

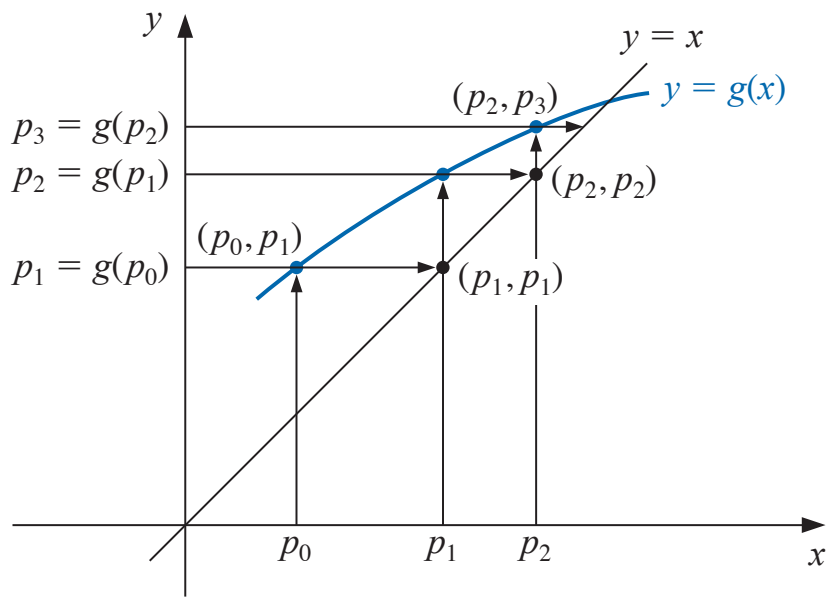
$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g(\lim_{n\to\infty} p_{n-1}) = g(p)$$

i.e., the limit of p_n is a fixed point of g.

Fixed point iteration

Example trajectories of fixed point iteration:





Fixed point iteration

Fixed Point Iteration Algorithm:

- **Input:** initial p_0, tolerence ϵ_{tol}, max iteration $N_{\max}$. Set iteration counter $N = 1$.
- While $N \leq N_{\max}$, do:
  1. Set $p = g(p_0)$ (update p_N to p_{N+1})
  2. If $|p - p_0| < \epsilon_{\text{tol}}$, then STOP
  3. Set $N \leftarrow N + 1$
  4. Set $p_0 = p$ (prepare p_N for the next iteration)
- **Output:** If $N \geq N_{\max}$, print("Max iteration reached."). Return p.

FPI for root-finding

We can also use FPI to find the root of a function f:

1. Determine a function g, such that $p = g(p) \Leftrightarrow f(p) = 0$.[1]
2. Apply FPI to g and find FP p.

[1] We can use $\Longrightarrow$ only, but we may miss some roots of f.

Example

Example (FPI algorithm for root-finding)

Find a root of $f(x) = x^3 + 4x^2 - 10$ using FPI.

Solution. First notice that

$$\begin{aligned} x^3 + 4x^2 - 10 = 0 &\iff 4x^2 = 10 - x^3 \\ &\iff x^2 = \frac{10 - x^3}{4} \\ &\iff x = \pm\sqrt{\frac{10 - x^3}{4}} \\ &\iff x^2 = \frac{10 - 4x^2}{x} \\ &\iff \dots \end{aligned}$$

Example

Example (FPI algorithm for root-finding)

Find a root of $f(x) = x^3 + 4x^2 - 10$ using FPI.

Solution. So we can define several g:

$$g_1(x) = x - (x^3 + 4x^2 - 10)$$

$$g_2(x) = \sqrt{\frac{10}{x} - 4x}$$

$$g_3(x) = \sqrt{\frac{10 - x^3}{4}}$$

$$g_4(x) = \sqrt{\frac{10}{4 + x}}$$

$$g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

Which g to choose? – All these g have the the same FP p. But g_3, g_4, g_5 converge (g_5 fastest) while g_1, g_2 do not.

Convergence of FPI algorithm

Theorem (Convergence of FPI Algorithm)

Suppose $g \in C[a,b]$ *s.t.* $g(x) \in [a,b]$, $\forall\, x \in [a,b]$. *If* $\exists\, k \in (0,1)$ *s.t.* $|g'(x)| \leq k$, $\forall\, x \in (a,b)$, *then* $\{p_n\}$ *generated by FPI algorithm converges to the unique FP of* $g(x)$ *on* $[a,b]$.

Proof.

$g(x) \in [a,b]$ and $|g'(x)| \leq k < 1$, $\forall\, x \in [a,b] \Longrightarrow \exists\,!$ FP p on $[a,b]$ by FPT. Moreover, $\exists\, \xi(p_{n-1})$ between p and p_{n-1} s.t.

$$|p_n - p| = |g(p_{n-1}) - g(p)| = |g'(\xi(p_{n-1}))||p_{n-1} - p| \leq k|p_{n-1} - p|$$

Apply this inductively, we get

$$|p_n - p| \leq k|p_{n-1} - p| \leq k^2|p_{n-2} - p| \leq \cdots \leq k^n|p_0 - p| \to 0$$

since $k^n \to 0$ as $n \to \infty$. □

Convergence rate of FPI algorithm

Corollary (Convergence rate of FPI Algorithm)

With the same conditions as above, we have for all $n \geq 1$

- $|p_n - p| \leq k^n \max\{p_0 - a, b - p_0\}$
- $|p_n - p| \leq \frac{k^n}{1-k}|p_1 - p_0|$

Proof.

1. $|p_0 - p| \leq \max\{p_0 - a, b - p_0\}$. Then apply the proof above.
2. Apply the proof above to get $|p_{n+1} - p_n| \leq k^n |p_1 - p_0|$. Then

$$|p_m - p_n| \leq |p_1 - p_0| \sum_{i=0}^{m-n-1} k^{n+i} = \frac{1 - k^{m-n}}{1-k} k^n |p_1 - p_0|$$

Let $m \to \infty$ to get the estimate.

□

Example

Example (FPI algorithm for root-finding)

Find a root of $f(x) = x^3 + 4x^2 - 10$ using FPI algorithm.

Solution. Recall the functions g we defined:

$$g_1(x) = x - (x^3 + 4x^2 - 10)$$

$$g_2(x) = \sqrt{\frac{10}{x} - 4x}$$

$$g_3(x) = \sqrt{\frac{10 - x^3}{4}}$$

$$g_4(x) = \sqrt{\frac{10}{4 + x}}$$

$$g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

Apply the theorem above, check $|g'(x)|$, and explain why FPI algorithm converges with g_3, g_4, g_5.

Fixed point iteration for root-finding

To find a good FPI algorithm for root-finding $f(p) = 0$, find a function g s.t.

- $g(p) = p \Longrightarrow f(p) = 0$
- g is continuous, differentiable
- $|g'(x)| \leq k \in (0, 1)$, $\forall\, x$ with k as small as possible

Newton's method

Suppose p is a root of f and p_0 is sufficiently close to p, then

$$f(p) = f(p_0) + f'(p_0)(p - p_0) + \frac{1}{2}f''(\xi(p))(p - p_0)^2$$

for some $\xi(p)$ between p_0 and p.

Since $f(p) = 0$, and $(p - p_0)^2$ is close to 0, we have

$$0 \approx f(p_0) + f'(p_0)(p - p_0)$$

Therefore (assume $f'(p_0) \neq 0$),

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} =: p_1$$

So p_1 is our guess for p now!

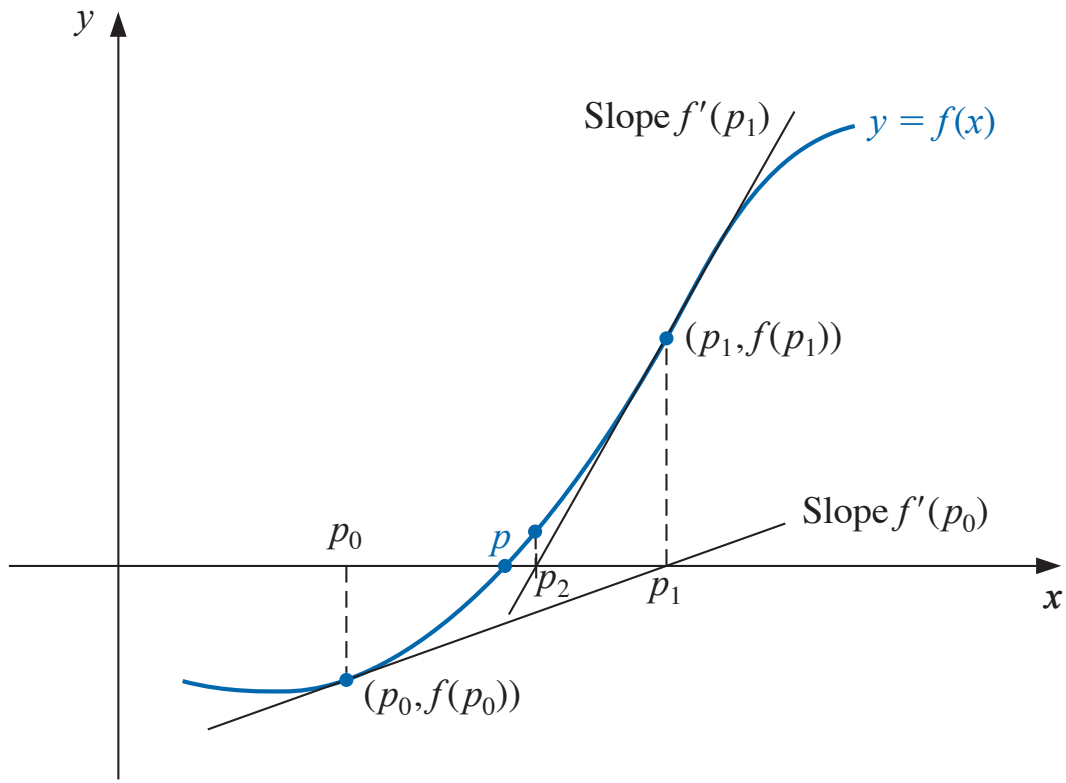
Netwon's method

Newton's method:

Start from initial guess p_0 (close to the FP p), and iterate:

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

Then we hope $p_n \to p$ quickly.



Newton's method

Newton's method

- **Input.** Initial guess p_0, ϵ_{tol}, $N_{\max}$. Set $N = 1$.
- While $N \leq N_{\max}$, do:
  1. Set $p = p_0 - \frac{f(p_0)}{f'(p_0)}$ (compute p_n using p_{n-1})
  2. If $|p - p_0| < \epsilon_{\text{tol}}$, STOP
  3. Set $N = N + 1$
  4. Set $p_0 = p$ (update p_{n-1} using p_n for next iteration)
- **Output.** Approximate solution p. If $N \geq N_{\max}$, print("Max iteration reached").

Newton's method

Newton's method is equivalent to fixed point iteration algorithm with

$$g(x) := x - \frac{f(x)}{f'(x)}$$

So p is a FP of g iff $f(p) = 0$.

Convergence of Newton's method

Theorem (Convergence of Newton's method)

If $f \in C^2[a,b]$ and $\exists\, p \in (a,b)$ such that $f(p) = 0$ and $f'(p) \neq 0$, then $\exists\, \delta > 0$ such that Newton's method convergent starting from any $p_0 \in (p-\delta, p+\delta)$.

Proof.

Hint: Check $g'(x)$:

$$g'(x) = 1 - \frac{(f')^2 - ff''}{(f')^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

$f \in C^2$, $f(p) = 0$, $f'(p) \neq 0$ together imply $\exists\, \delta > 0$ s.t. $|g'(x)| < 1$ for $x \in (p_0 - \delta, p_0 + \delta)$. □

Secant method

A problem with Newton's method is that $f'(x)$ may not be easy to calculate, so we approximate $f'(p_{n-1})$ in the Newton's method by

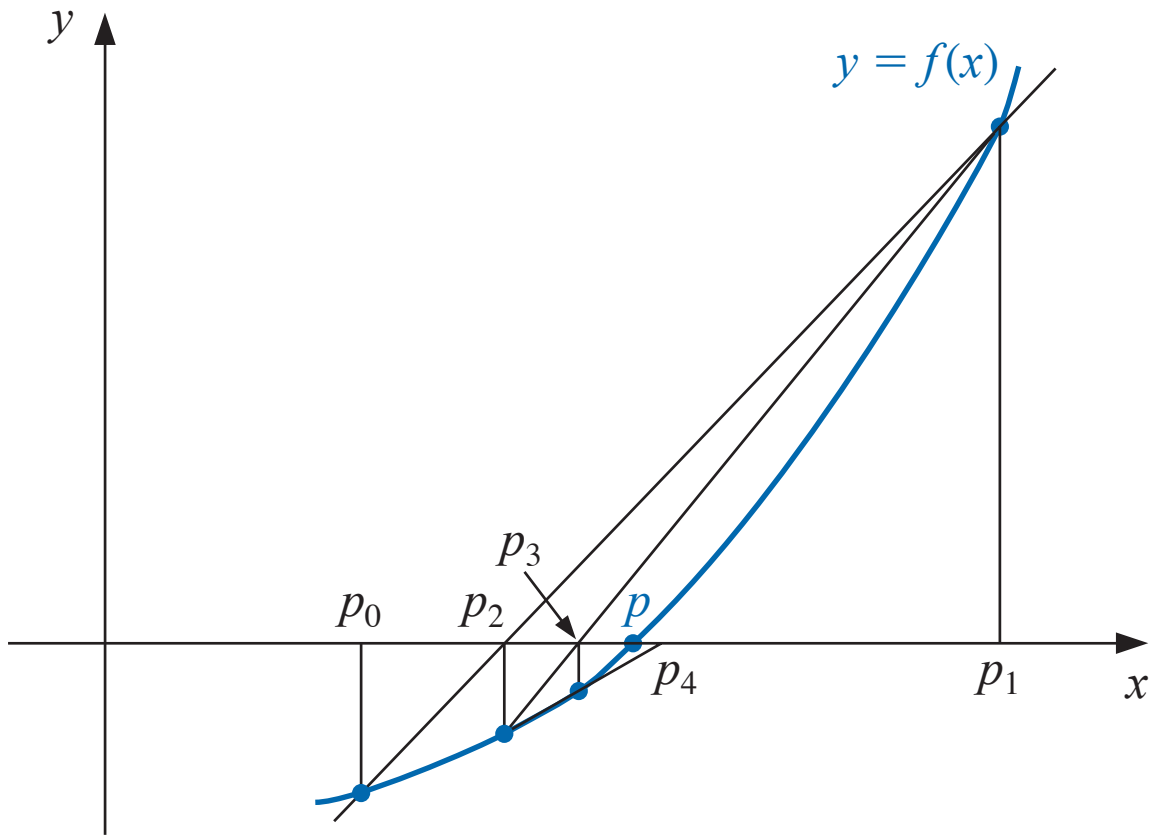
$$f'(p_{n-1}) \approx \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}$$

After simple algebra, we get the **secant method**:

$$\begin{aligned} p_n &= p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})} \\ &= \frac{p_{n-2}f(p_{n-1}) - p_{n-1}f(p_{n-2})}{f(p_{n-1}) - f(p_{n-2})} \end{aligned}$$

Secant method

Illustration of the secant method:



Secant method

- **Input.** Initial guess p_0, p_1, $q_0 = f(p_0), q_1 = f(p_1)$, ϵ_{tol}, $N_{\max}$. Set $N = 1$.
- While $N \leq N_{\max}$, do:
  1. Set $p = p_1 - \frac{q_1(p_1 - p_0)}{q_1 - q_0} = \frac{p_0 q_1 - p_1 q_0}{q_1 - q_0}$ (compute p_n using p_{n-1}, p_{n-2})
  2. If $|p - p_1| < \epsilon_{\text{tol}}$, STOP
  3. Set $N = N + 1$
  4. Set $p_1 = p, q_1 = f(p), p_0 = p_1, q_0 = q_1$ (update p_{n-1}, p_{n-2} for next iteration)
- **Output.** Approximate solution p. If $N \geq N_{\max}$, print("Max iteration reached").

Error analysis

Definition (Order of convergence)

Suppose $p_n \to p$. If $\exists\, \lambda, \alpha > 0$ s.t.

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$$

then $\{p_n\}$ is said to converge to p of **order** α, with asymptotic error constant λ.

Error analysis

Definition (Convergence order of numerical methods)

An iterative method $p_n = g(p_{n-1})$ is of **order** α if the generated $\{p_n\}$ converges to the solution p of $p = g(p)$ at order α.

In particular:

- $\alpha = 1$: **linearly convergent**
- $\alpha = 2$: **quadratically convergent**

Example

Example (Speed comparison: linear vs quadratic)

Suppose p_n (and q_n respectively) converges to 0 linearly (quadratically) with constant 0.5, enumerate the upper bound of $|p_n|$ and $|q_n|$.

Solution. By definition of convergence order, we know

$$\lim_{n\to\infty} \frac{|p_{n+1}|}{|p_n|} = 0.5 \qquad \text{and} \qquad \lim_{n\to\infty} \frac{|q_{n+1}|}{|q_n|^2} = 0.5$$

Suppose that p_0 and q_0 are close enough to 0 s.t. $|p_{n+1}|/|p_n| \approx 0.5$ and $|q_{n+1}|/|q_n| \approx 0.5$ for all n, then

$$|p_n| \approx 0.5|p_{n-1}| \approx 0.5^2|p_{n-2}| \approx \cdots \approx 0.5^n|p_0|$$

$$|q_n| \approx 0.5|q_{n-1}|^2 \approx 0.5 \cdot 0.5^2|q_{n-2}|^4 \approx \cdots \approx 0.5^{2^n-1}|q_0|^{2^n}$$

Example

Example (Speed comparison: linear vs quadratic)

Suppose $p_0, q_0 \approx 0.5$. Then

n	Linear 0.5^n	Quadratic 0.5^{2^n-1}
1	5.0000×10^{-1}	5.0000×10^{-1}
2	2.5000×10^{-1}	1.2500×10^{-1}
3	1.2500×10^{-1}	7.8125×10^{-3}
4	6.2500×10^{-2}	3.0518×10^{-5}
5	3.1250×10^{-2}	4.6566×10^{-10}
6	1.5625×10^{-2}	1.0842×10^{-19}
7	7.8125×10^{-3}	5.8775×10^{-39}

Convergence rate of fixed point iteration algorithm

Theorem (FPI alg has linear convergence rate)

Suppose $g \in C^1[a,b]$ *s.t.* $g(x) \in [a,b]$, $\forall\, x \in [a,b]$. *If* $\exists\, k \in (0,1)$ *s.t.* $|g'(x)| \leq k$, $\forall\, x \in (a,b)$, *then* $\{p_n\}$ *generated by FPI algorithm converges to the unique FP of* $g(x)$ *on* $[a,b]$ **linearly**.

Proof.

We already know $p_n \to p$ where p is the unique fixed point of g by FPT. Also

$$p_{n+1} - p = g(p_n) - g(p) = g'(\xi(p_n))(p_n - p)$$

where $\xi(p_n)$ is between p_n and p. Hence

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = \lim_{n\to\infty} |g'(\xi(p_n))| = |g'(\lim_{n\to\infty} \xi(p_n))| = |g'(p)| \leq k < 1$$

Therefore $p_n \to p$ linearly with constant k. □

Improve convergence order of FPI to quadratic

Theorem (Additional condition for quadratic rate)

If $g \in C^2[a,b]$ *and* $g'(p) = 0$ *for a FP* $p \in (a,b)$*, then* $\exists\, M > 0$ *s.t.* $|g''(x)| \le M$*,* $\forall\, x \in [a,b]$ *and* $\exists\, \delta > 0$ *s.t. sequence* $\{p_n\}$ *by FPI stared in* $[p-\delta, p+\delta]$ *satisfies*

$$|p_{n+1} - p| \le \frac{M}{2}|p_n - p|^2, \qquad \forall\, n$$

Improve convergence order of FPI

Proof.

Notice that $g \in C^2, g(p) = p, g'(p) = 0$ together imply that $\exists\, \delta > 0$ and $k \in (0, 1)$ s.t.

$$|g'(x)| \leq k < 1, \qquad x \in [p - \delta, p + \delta]$$

and

$$g : [p - \delta, p + \delta] \to [p - \delta, p + \delta]$$

Also

$$g(p_n) = g(p) + g'(p)(p_n - p) + \frac{1}{2} g''(\xi(p_n))(p_n - p)^2$$

where $\xi(p_n)$ is between p_n and p.

Since $p_{n+1} = g(p_n)$, $g(p) = p$, and $g'(p) = 0$, we have

$$p_{n+1} = p + \frac{1}{2} g''(\xi(p_n))(p_n - p)^2$$

So

$$\frac{|p_{n+1} - p|}{|p_n - p|^2} = \frac{1}{2} |g''(\xi(p_n))| \leq \frac{M}{2}$$

Improve convergence order of FPI

Suppose we have a fixed point method with $g(x) = x - \phi(x)f(x)$. How to choose ϕ such that FPI converges quadratically?

We need g s.t. $g'(p) = 0$ at a FP p (root of f):

$$g'(p) = 1 - \phi'(p)f(p) - \phi(p)f'(p) = 0$$

Since $f(p) = 0$ we have $\phi(p) = \frac{1}{f'(p)}$. Choose $\phi(x) = \frac{1}{f'(x)}$ s.t.

$$g(x) = x - \frac{f(x)}{f'(x)}$$

This is exactly Newton's method!

So Newton's method converges quadratically.

Convergence of Newton's method when $f'(p) = 0$

We mentioned condition $f'(p) \neq 0$ at the root p of f in the convergence proof of Newton's method above.

What if $f'(p) = 0$? When will this happen and how to address it?

Multiple roots

$f'(p) = 0$ at root p means p is not a "simple root".

Definition (Root multiplicity)

A root p of $f(x)$ is a **root (zero) of multiplicity** m if $f(x) = (x-p)^m q(x)$ for some q s.t. $\lim_{x \to p} q(x) \neq 0$.

Definition (Simple root)

p is a **simple root (zero)** of f if its multiplicity $m = 1$.

Multiple roots

Theorem (Sufficient and necessary condition for simple root)

$f \in C^1[a,b]$ has a simple root $p \in (a,b)$ iff $f(p) = 0$ and $f'(p) \neq 0$.

Proof.

"$\Longrightarrow$": $f(x) = (x-p)q(x)$ where $\lim_{x\to p} q(x) \neq 0$. Then $f'(x) = q(x) + (x-p)q'(x)$. So $f \in C^1$ implies

$$f'(p) = \lim_{x\to p} f'(x) = \lim_{x\to p} (q(x) + (x-p)q'(x)) \neq 0$$

"$\Longleftarrow$": $f(x) = f(p) + f'(\xi(x))(x-p)$ where $\xi(x)$ between x and p. Define $q(x) = f'(\xi(x))$ then

$$\lim_{x\to p} q(x) = \lim_{x\to p} f'(\xi(x)) = f'(\lim_{x\to p} \xi(x)) = f'(p) \neq 0$$

So f has a simple root at p. □

Multiple roots

Theorem (Sufficient and necessary condition for multiple root)

$f \in C^m[a,b]$ has a zero p of multiplicity m iff

$$f(p) = f'(p) = \cdots = f^{(m-1)}(p) = 0 \quad \textit{and} \quad f^{(m)}(p) \neq 0$$

Proof.

Hint: Follow the proof above and use

$$(uv)^{(n)} = \sum_{k=0}^{n} \binom{n}{k} u^{(k)} v^{(n-k)}$$

□

Example

Example (Multiple root)

Let $f(x) = e^x - x - 1$, show that $f(x)$ has a zero of multiplicity 2 at $x = 0$.

Solution. $f(x) = e^x - x - 1$, $f'(x) = e^x - 1$, and $f''(x) = e^x$. So $f(0) = f'(0) = 0$ and $f''(0) = 1 \neq 0$. By Theorem above f has root (zero) at $x = 0$ of multiplicity 2.

Modified Newton's method

Instead of using $f(x)$ in Newton's method, we can replace f by

$$\mu(x) := \frac{f(x)}{f'(x)}$$

We need to show:

$$\boxed{p \text{ is a root (simple or not) of } f \implies p \text{ is a simple root of } \mu}$$

Modified Newton's method

Recall that f has a root p of multiplicity m if $f(x) = (x-p)^m q(x)$ for some q with $\lim_{x\to p} q(x) \neq 0$.

Now there is

$$\mu(x) = \frac{f(x)}{f'(x)} = \frac{(x-p)^m q(x)}{m(x-p)^{m-1}q(x) + (x-p)^m q'(x)}$$
$$= (x-p) \cdot \frac{q(x)}{mq(x) + (x-p)q'(x)}$$

where $\frac{q(x)}{mq(x)+(x-p)q'(x)} \to \frac{1}{m} \neq 0$ as $x \to p$.

By definition, $\mu(x)$ has simple root at p, i.e., $\mu(p) = 0$ and $\mu'(p) \neq 0$.

Modified Newton's method

Now we use $\mu(x)$ instead of $f(x)$ in Newton's method:

$$g(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{(f(x)/f'(x))}{(f(x)/f'(x))'} = \cdots = x - \frac{f(x)f'(x)}{(f'(x))^2 - f(x)f''(x)}$$

The **modified Newton's method** is

$$p_n = p_{n-1} - \frac{f(p_{n-1})f'(p_{n-1})}{(f'(p_{n-1}))^2 - f(p_{n-1})f''(p_{n-1})}$$

Drawbacks of the modified Newton's method:

- Needs f'' in computation.
- Denominator approximates 0 as $p_n \to p$, so round-off may degrade convergence.

Accelerating convergence

We showed that FPI generally has linear convergence only. How to improve?

Suppose N is large, and p_n, p_{n+1}, p_{n+2} satisfy

$$\frac{p_{n+1}-p}{p_n-p} \approx \frac{p_{n+2}-p}{p_{n+1}-p}$$

$$\iff \quad (p_{n+1}-p)^2 \approx (p_n-p)(p_{n+2}-p) = p_n p_{n+2} - p(p_{n+2}+p_n) + p^2$$

$$\iff \quad p \approx \frac{p_n p_{n+2} - p_{n+1}^2}{p_{n+2}-2p_{n+1}+p_n} = \cdots = p_n - \frac{(p_{n+1}-p_n)^2}{p_{n+2}-2p_{n+1}+p_n}$$

Aitken's Δ^2 method

Denote $\Delta p_n := p_{n+1} - p_n$, called **forward difference**, and

$$\begin{aligned}\Delta^2 p_n :=& \Delta(\Delta p_n) = \Delta(p_{n+1} - p_n) \\ =& (p_{n+2} - p_{n+1}) - (p_{n+1} - p_n) \\ =& p_{n+2} - 2p_{n+1} + p_n\end{aligned}$$

So the result above can be written as $p \approx p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$.

Aitken's Δ^2 method:
Given $\{p_n\}$ generated by FPI, set $\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}$. Then $\hat{p}_n \to p$ faster than p_n.

Aitken's Δ^2 method

What does it mean by "faster"?

Theorem (Faster convergence by Aitken's Δ^2 method)

If $p_n \to p$ linearly with $\lim_{n\to\infty} \frac{p_{n+1}-p}{p_n-p} < 1$, then $\hat{p}_n$ computed by Aitken's Δ^2 method satisfy

$$\lim_{n\to\infty} \frac{\hat{p}_n - p}{p_n - p} = 0$$

Proof.

Hint: Define $e_n := p_n - p$, then $\Delta e_n = \Delta p_n$, $\Delta^2 e_n = \Delta^2 p_n$, and $\frac{e_{n+1}}{e_n} \to \lambda < 1$.
Then

$$\frac{\hat{p}_n - p}{p_n - p} = \frac{p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} - p}{p_n - p} = \frac{e_n - \frac{(\Delta e_n)^2}{\Delta^2 e_n}}{e_n} = \frac{\frac{e_{n+2}}{e_{n+1}} - \frac{e_{n+1}}{e_n}}{\frac{e_{n+2}}{e_{n+1}} - 2 + \frac{e_n}{e_{n+1}}} \to \frac{\lambda - \lambda}{\lambda - 2 + \frac{1}{\lambda}} = 0$$

□

Steffenson's method

Aitken's method computes $\hat{p}_n$ separately from p_n. Steffenson's method makes use of $\hat{p}_n$ to compute future p_n.

Steffenson's method: given g for FPI, compute

$$
\begin{aligned}
& p_0^{(0)}, && p_1^{(0)} = g(p_0^{(0)}), \quad p_2^{(0)} = g(p_0^{(0)}) \\
& p_0^{(1)} = p_0^{(0)} - \frac{(\Delta p_0^{(0)})^2}{\Delta^2 p_0^{(0)}}, && p_1^{(1)} = g(p_0^{(1)}), \quad p_2^{(1)} = g(p_1^{(1)}) \\
& p_0^{(2)} = p_0^{(1)} - \frac{(\Delta p_0^{(1)})^2}{\Delta^2 p_0^{(1)}}, && p_1^{(2)} = g(p_0^{(2)}), \quad p_2^{(2)} = g(p_1^{(2)}) \\
& \vdots
\end{aligned}
$$

Steffenson's method

Steffenson's method

- **Input.** Initial guess p_0, ϵ_{tol}, $N_{\max}$. Set $N = 1$.
- While $N \leq N_{\max}$, do :
  1. Set $p_1 = g(p_0)$, $p_2 = g(p_1)$ and $p = p_0 - \frac{(p_1 - p_0)^2}{p_2 - 2p_1 + p_0}$
  2. If $|p - p_0| < \epsilon_{\text{tol}}$, STOP
  3. $p_0 = p$
  4. Set $N = N + 1$
- **Output.** Return p. If $N \geq N_{\max}$, print("Max iteration reached.").

Steffenson's method

Theorem

Suppose $g(x)$ has a fixed point p and $g'(p) \neq 1$. If $\exists\, \delta > 0$, s.t. $f \in C^3[p-\delta, p+\delta]$, then Steffenson's method generates a sequence $\{p_n\}$ converging to p quadratically for any initial $p_0 \in [p-\delta, p+\delta]$.

Section 3

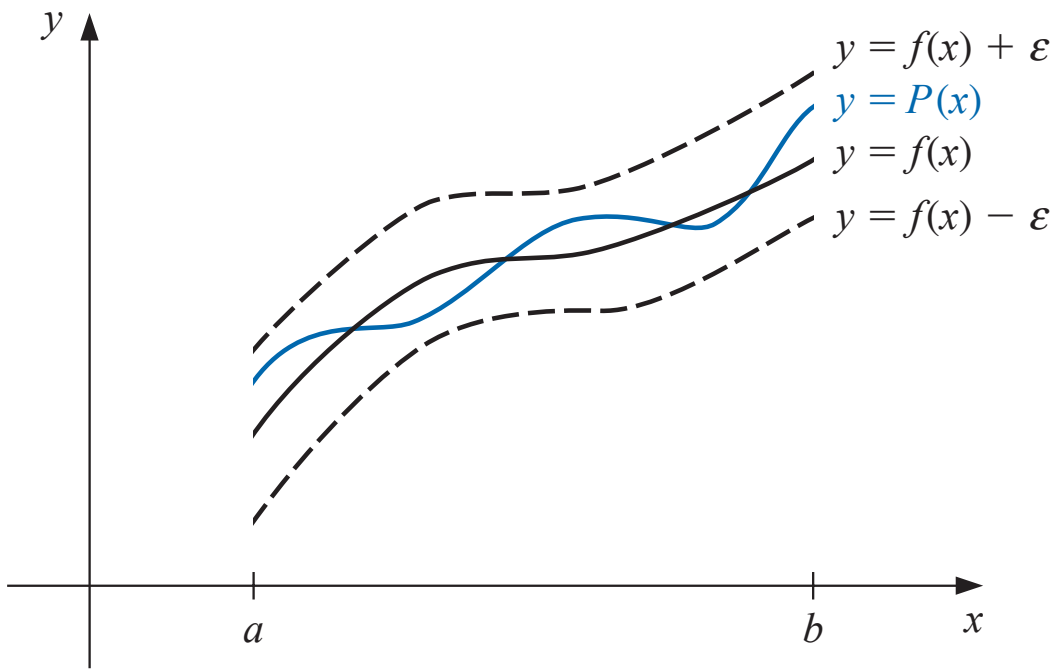
Interpolation and Polynomial Approximation

Interpolation

Given data points $\{(x_i, y_i) : i = 1, \ldots, n\}$, can we find a function to "fit" the data?

Theorem (Weierstrass approximation theorem)

Suppose $f \in C[a, b]$, then $\forall\, \epsilon > 0$, $\exists$ a polynomial $P(x)$ such that $|f(x) - P(x)| < \epsilon$, $\forall\, x \in [a, b]$.



Polynomial interpolation

So polynomials could work. But how to find the polynomial?

First Try: Taylor's polynomial

For any given function $f(x)$ and a point x_0, we approximate $f(x)$ by the Taylor's polynomial $P_n(x)$:

$$f(x) \approx P_n(x) := f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n$$

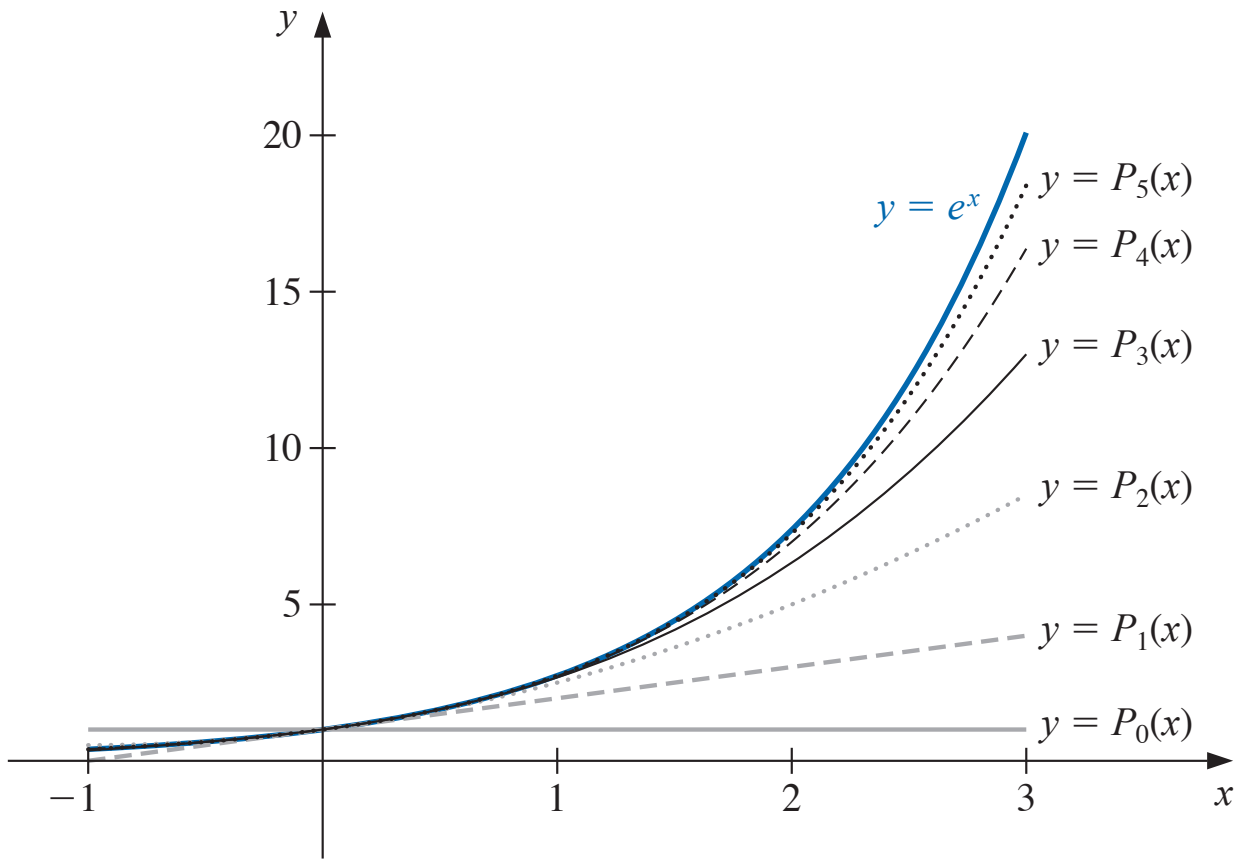
Example (Problem with Taylor's polynomial)

Let $f(x) = e^x$ and $x_0 = 0$. See how Taylor's polynomial behaves.

Solution. Taylor's polynomial $P_n(x) = 1 + x + \cdots + \frac{1}{n!}x^n$.

However, no matter how large we choose n, $P_n(x)$ is far from $f(x)$ where x is slightly large.

Issue with Taylor's polynomial approximation



Example

Example (Problem with Taylor's polynomial)

Let $f(x) = \frac{1}{x}$ and $x_0 = 1$. See how Taylor's polynomial behaves.

Solution. We know $f^{(n)}(x) = \frac{(-1)^n n!}{x^{n+1}}$. Then Taylor's polynomial is

$$P_n(x) = \sum_{i=0}^{n} (-1)^n (x-1)^n = 1 - (x-1) + (x-1)^2 + \cdots + (-1)^n (x-1)^n$$

Suppose we use $P_n(x)$ to approximate f at $x = 3$, we get

$P_0(3)$	$P_1(3)$	$P_2(3)$	$P_3(3)$	$P_4(3)$	$P_5(3)$	$P_6(3)$	$P_7(3)$
1	-1	3	-5	11	-21	43	-85

But the true value is $f(3) = \frac{1}{3}$.

Lagrange interpolating polynomial

We should not use Taylor's polynomial since it only approximates well locally.

Suppose we have two points (x_0, y_0) and (x_1, y_1), then best use a straight line to interpolate. Define two linear polynomials:

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

So L_0 and L_1 are polynomials of degree 1, and

$$L_0(x_1) = 0, \quad L_0(x_0) = 1, \quad L_1(x_0) = 0, \quad L_1(x_1) = 1$$

Now set $P(x) = f(x_0)L_0(x) + f(x_1)L_1(x)$, then $P(x)$ coincides $f(x)$ at x_0 and x_1.

Example

Recall that the polynomial we derived is

$$P(x) = f(x_0)L_0(x) + f(x_1)L_1(x) = \frac{x - x_0}{x_1 - x_0} f(x_0) + \frac{x - x_1}{x_0 - x_1} f(x_1)$$

$P(x)$ is called the **Lagrange interpolating polynomial** of f given values at x_0 and x_1.

Example (Linear Lagrange interpolating polynomial)

Use linear Lagrange interpolating polynomial of f where $f(2) = 4$ and $f(5) = 1$.

Solution. $P(x) = -x + 6$.

Lagrange interpolating polynomial

Given $n+1$ points $\{(x_i, f(x_i)) : 0 \le i \le n\}$. Define:

$$L_{n,k}(x) = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

for $k = 0, 1, \dots, n$. Then it is easy to verify

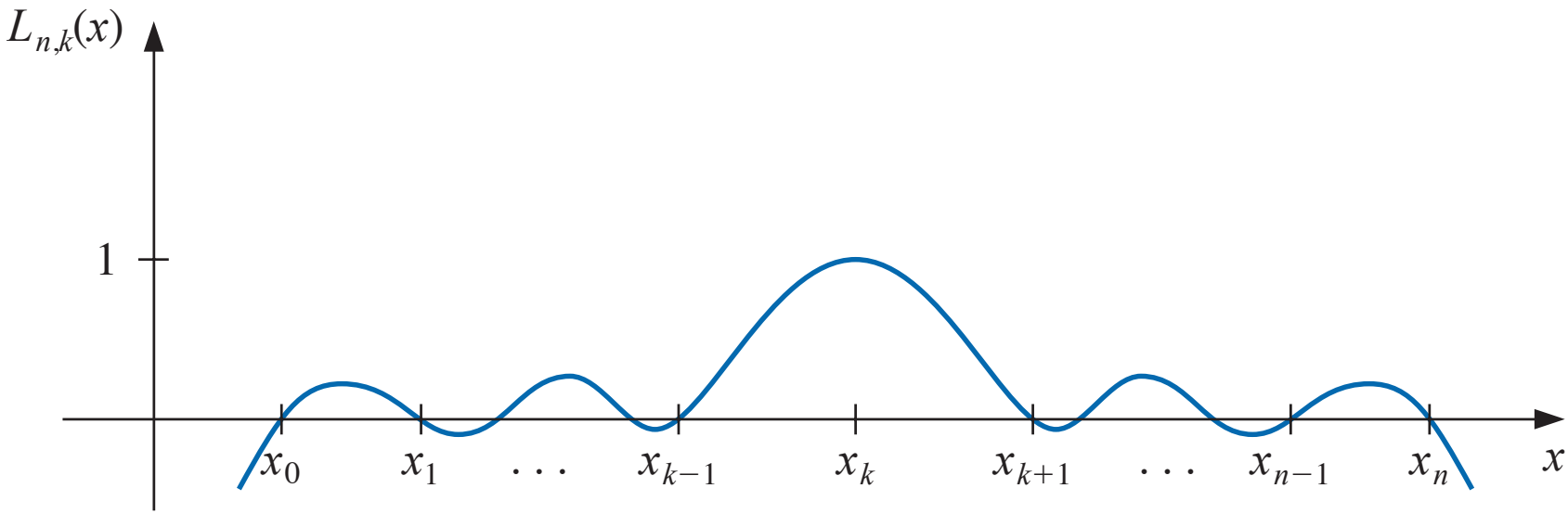
$$L_{n,k}(x) = \begin{cases} 1 & \text{if } x = x_k \\ 0 & \text{if } x = x_j, \text{ where } j \neq k \end{cases}$$

Then the ***n*th Lagrange interpolating polynomial** of f is

$$P(x) = \sum_{k=0}^{n} f(x_k) L_{n,k}(x)$$

Lagrange interpolating polynomial

Illustration of $L_{n,k}(x)$:



Lagrange interpolating polynomial

The ***n*th Lagrange interpolating polynomial** of f at $x_0, \ldots, x_n$ is

$$P(x) = \sum_{k=0}^{n} f(x_k) L_{n,k}(x)$$

Properties:

- $P(x)$ is a polynomial of degree n
- $P(x_k) = f(x_k)$ for all $k = 0, \ldots, n$.

Example

Example (Lagrange interpolating polynomial)

Let $f(x) = \frac{1}{x}$, $x_0 = 2$, $x_1 = 2.75$, $x_2 = 4$. Find the 2nd Lagrange interpolating polynomial $P(x)$ of $f(x)$ and compute $P(3)$.

Solution. First we compute $L_{2,k}$ for $k = 0, 1, 2$:

$$L_{2,0}(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 2.75)(x - 4)}{(2 - 2.75)(2 - 4)}$$

$$L_{2,1}(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - 2)(x - 4)}{(2.75 - 2)(2.75 - 4)}$$

$$L_{2,2}(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - 2)(x - 2.75)}{(4 - 2)(4 - 2.75)}$$

Then the 2nd Lagrange interpolating polynomial is

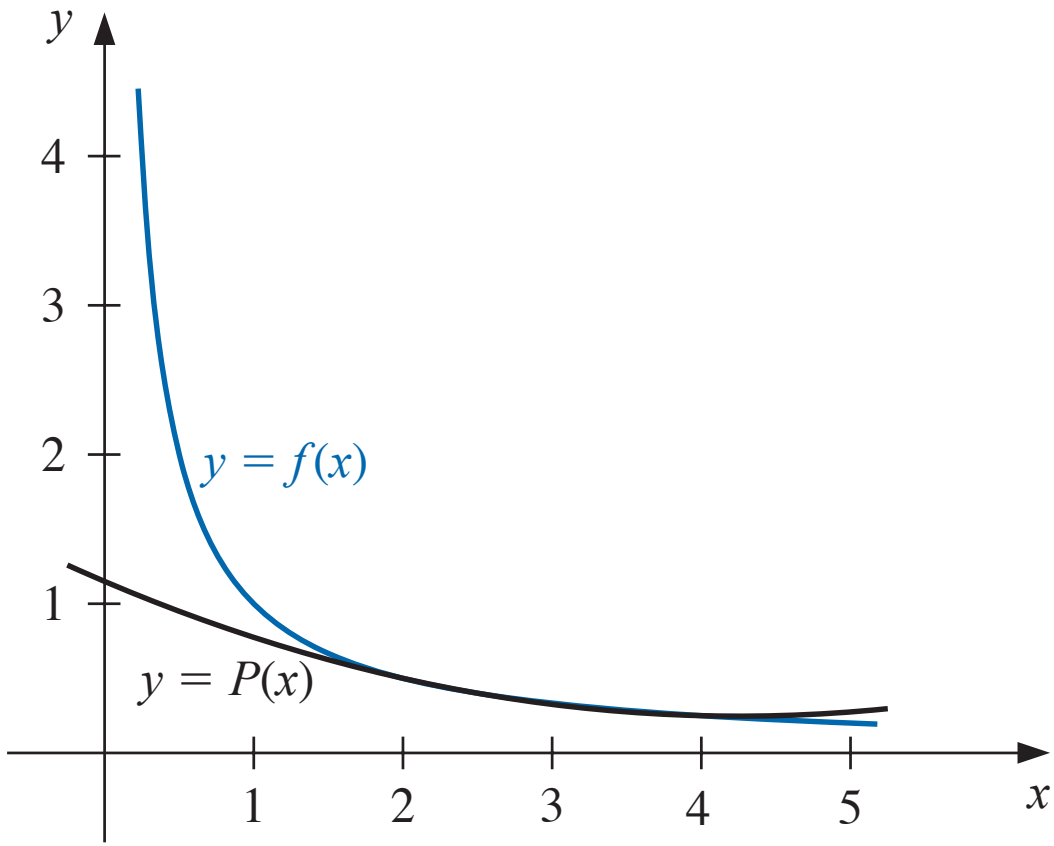
$$P(x) = \sum_{k=0}^{2} f(x_k) L_{2,k}(x) = \cdots = \frac{x^2}{22} - \frac{35x}{88} + \frac{49}{44}$$

Note that $P(3) = \frac{3^2}{22} - \frac{35 \times 3}{88} + \frac{49}{44} \approx 0.32955$, close to $f(3) = \frac{1}{3}$.

Example

Example (Lagrange interpolating polynomial)

Let $f(x) = \frac{1}{x}$, $x_0 = 2$, $x_1 = 2.75$, $x_2 = 4$. Find the 2nd Lagrange interpolating polynomial $P(x)$ of $f(x)$ and compute $P(3)$.



Lagrange interpolating polynomial

Theorem (Error of Lagrange interpolating polynomial)

Suppose $f(x) \in C^{n+1}[a,b]$. *Then for every* $x \in [a,b]$, $\exists\, \xi(x)$ *between* $x_0, \ldots, x_n$, *s.t.*

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)\ldots(x - x_n)$$

Error of Lagrange interpolating polynomial

Proof.

For any given $x \in [a, b]$ different from $x_0, \ldots, x_n$, define $g(t)$ as

$$g(t) = f(t) - P(t) - \underbrace{(f(x) - P(x))\frac{(t - x_0)\ldots(t - x_n)}{(x - x_0)\ldots(x - x_n)}}_{\text{polynomial of } t, \text{ degree } n+1}$$

Note that $f(t) = P(t)$ and $(t - x_0)\ldots(t - x_n) = 0$ for $t = x_k$ and $k = 0, \ldots, n$. So $g(t) = 0$ for $t = x, x_0, \ldots, x_n$ (total $n + 2$ points). By generalized Rolle's Thm, $\exists \xi(x)$ between $x_0, \ldots, x_n$ s.t.

$$0 = g^{(n+1)}(\xi(x)) = f^{(n+1)}(\xi(x)) - \frac{(n+1)! \cdot (f(x) - P(x))}{(x - x_0)\ldots(x - x_n)}$$

since $P(t)$ is a poly of t with degree n and $(t - x_0)\cdots(t - x_n)$ is a monic poly of t with degree $n + 1$. □

Example

Example (Estimate error of Lagrange interpolating polynomial)

Let $f(x) = \frac{1}{x}$, $x_0 = 2$, $x_1 = 2.75$, $x_2 = 4$. Estimate the maximal error of the 2nd Lagrange interpolating polynomial $P(x)$ given above on $[2, 4]$.

Example

Solution. Let $P(x)$ be the Lagrange interpolating polynomial, then

$$f(x) - P(x) = \frac{f^{(3)}(\xi(x))}{3!}(x-2)(x-2.75)(x-4)$$

We know $f'(x) = -\frac{1}{x^2}$, $f''(x) = \frac{2}{x^3}$, $f'''(x) = -\frac{3!}{x^4}$, so

$$\left|\frac{f^{(3)}(\xi(x))}{3!}\right| = \left|-\frac{1}{(\xi(x))^4}\right| \leq \frac{1}{2^4} \qquad (\because \xi(x) \in [2,4])$$

Further, denote $h(x) := (x-2)(x-2.75)(x-4)$, find critical points and then the max/min values of $h(x)$ on $[2,4]$ to claim $|h(x)| \leq \frac{9}{16}$ for all $x \in [2,4]$. Hence

$$|f(x) - P(x)| = \left|\frac{f^{(3)}(\xi(x))}{3!}h(x)\right| \leq \frac{1}{2^4}\frac{9}{16} \approx 0.00586.$$

Example

Example (Estimate error of Lagrange interpolating polynomial)

Suppose we use uniform partition of $[0, 1]$ and linear Lagrange interpolating polynomial on each segment to approximate $f(x) = e^x$. How small the step size h should be to guarantee the error $< 10^{-6}$ everywhere?

Example

Solution. With step size h, we have $x_j = jh$ for $j = 0, 1, \ldots$.

Then we use linear Lagrange polynomial to approximate e^x on each $[x_j, x_{j+1}]$. The error is

$$\frac{1}{2} f^{(2)}(\xi(x))(x - x_j)(x - x_{j+1})$$

So $|\frac{f^{(2)}(\xi(x))}{2}| = |\frac{e^{\xi(x)}}{2}| \leq \frac{e}{2}$ ($\because \xi(x) \in [0, 1]$).

Again take $h(x) = (x - x_j)(x - x_{j+1})$ which has max $\frac{h^2}{4}$. Then

$$\left| \frac{f^{(2)}(\xi(x))}{2}(x - x_j)(x - x_{j+1}) \right| \leq \frac{e}{2}\frac{h^2}{4} \leq 10^{-6}$$

So we need $h \leq (8 \times 10^{-6} \times e^{-1})^{1/2} \approx 1.72 \times 10^{-3}$.

Recursive constructions of interpolating polynomials

Given points $x_0, \ldots, x_n$ and function values $f(x_k)$ for $k = 0, \ldots, n$.

There are several questions regarding the use Lagrange interpolating polynomial:

- Can we use a subset of points to construct Lagrange interpolating polynomials with lower degree?
- If yes, which interpolating points among $x_0, \ldots, x_n$ to choose?
- If the result is not satisfactory, can we improve the constructed polynomial to get a polynomial of higher degree?

Example

Example (Which points to choose?)

Consider the interpolation of the function f with 5 points:

k	x_k	$f(x_k)$
0	1.0	0.7651977
1	1.3	0.6200860
2	1.6	0.4554022
3	1.9	0.2818186
4	2.2	0.1103623

If we use an interpolating polynomial of degree $n < 4$, then we need to decide which points to use.

For example, if $n = 2$, then we need to chose 3 points. Should we choose x_0, x_1, x_2 or x_1, x_2, x_3, or x_0, x_2, x_4?

Neville's method

We do not know which choice is better, since true $f(x)$ is unknown. But we can compute all and see the trend.

Question: can we use polynomials obtained earlier (with lower degree) to get the later ones (with higher degree)?

Definition (Partial interpolating polynomial)

Let f be a function with known values at $x_0, \ldots, x_n$ and suppose $m_1, \ldots, m_k$ are k integers among $0, 1, \ldots, n$. Then the partial Lagrange interpolating polynomial that agrees with f at $x_{m_1}, \ldots, x_{m_k}$ is denoted by $P_{m_1,\ldots,m_k}(x)$.

Example

Example (Partial interpolating polynomial)

Let $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, $x_3 = 4$, $x_4 = 6$ for $f(x) = e^x$. Find $P_{1,2,4}(x)$ and approximate the value $f(5)$.

Solution. We only use x_1, x_2, x_4 to get $P_{1,2,4}(x)$:

$$\begin{aligned} P_{1,2,4}(x) &= \frac{(x-x_2)(x-x_4)}{(x_1-x_2)(x_1-x_4)} f(x_1) + \frac{(x-x_1)(x-x_4)}{(x_2-x_1)(x_2-x_4)} f(x_2) + \frac{(x-x_1)(x-x_2)}{(x_4-x_1)(x_4-x_2)} f(x_4) \\ &= \frac{(x-3)(x-6)}{(2-3)(2-6)} e^2 + \frac{(x-2)(x-6)}{(3-2)(3-6)} e^3 + \frac{(x-2)(x-3)}{(6-2)(6-3)} e^6 \\ P_{1,2,4}(5) &= -\frac{1}{2} e^2 + e^3 + \frac{1}{2} e^6 \approx 218.105 \end{aligned}$$

Recursive construction of interpolating polynomials

Now we show how to recursively construct Lagrange interpolating polynomials:

Theorem (Recursive construction of interpolating polynomials)

Let f be defined at $x_0, \ldots, x_k$, and x_i and x_j are two distinct points among them. Then

$$P_{0,1,\ldots,k}(x) = \frac{(x - x_j)P_{0,\ldots,\hat{j},\ldots,k}(x) - (x - x_i)P_{0,\ldots,\hat{i},\ldots,k}(x)}{x_i - x_j}$$

Recursive construction of interpolating polynomials

Proof.

Denote the RHS by $P(x)$.

Both $P_{0,\ldots,\hat{j},\ldots,k}(x)$ and $P_{0,\ldots,\hat{i},\ldots,k}(x)$ are polynomials of degree $k-1$, we know $P(x)$ is a polynomial of degree $\leq k$.

Verify that $P(x_s) = f(x_s)$ for $s = 0, 1, \ldots, k$. So $P(x) = P_{0,\ldots,k}(x)$. □

Neville's method

Suppose there are 5 points $x_0, \ldots, x_4$, and $P_i := f(x_i)$ for all i, then we can construct the following table:

$$
\begin{array}{llllll}
x_0 & P_0 & & & & \\
x_1 & P_1 & P_{0,1}(x) = \frac{(x-x_0)P_1-(x-x_1)P_0}{x_1-x_0} & & & \\
x_2 & P_2 & P_{1,2}(x) = \frac{(x-x_1)P_2-(x-x_2)P_1}{x_2-x_1} & P_{0,1,2}(x) = \frac{(x-x_0)P_{1,2}(x)-(x-x_2)P_{0,1}(x)}{x_2-x_0} & & \\
x_3 & P_3 & P_{2,3}(x) = \frac{(x-x_2)P_3-(x-x_3)P_2}{x_3-x_2} & P_{1,2,3}(x) = \frac{(x-x_1)P_{2,3}(x)-(x-x_3)P_{1,2}(x)}{x_3-x_1} & \ddots & \\
x_4 & P_4 & P_{3,4}(x) = \frac{(x-x_3)P_4-(x-x_4)P_3}{x_4-x_3} & P_{2,3,4}(x) = \frac{(x-x_2)P_{3,4}(x)-(x-x_4)P_{2,3}(x)}{x_4-x_2} & \ldots &
\end{array}
$$

Neville's method

We introduce a new notation $Q_{ij} = P_{i-j,i-j+1,\ldots,i}$ (i is the ending index and $j+1$ is the length), then the previous table is just

$$
\begin{array}{lllll}
x_0 & Q_{0,0} & & & \\
x_1 & Q_{1,0} & Q_{1,1} & & \\
x_2 & Q_{2,0} & Q_{2,1} & Q_{2,2} & \\
x_3 & Q_{3,0} & Q_{3,1} & Q_{3,2} & Q_{3,3} \\
x_4 & Q_{4,0} & Q_{4,1} & Q_{4,2} & Q_{4,3} \quad Q_{4,4}
\end{array}
$$

For example $Q_{3,3} = P_{0,1,2,3}$, $Q_{4,3} = P_{1,2,3,4}$, etc.

Example (Neville's method)

Consider the interpolation of the function f with 5 points:

k	x_k	$f(x_k)$
0	1.0	0.7651977
1	1.3	0.6200860
2	1.6	0.4554022
3	1.9	0.2818186
4	2.2	0.1103623

In addition, interpolate $f(1.5)$ and compare to the true value[2].

[2] The data in this table were retrieved from a Bessel function with true value $f(1.5) = 0.5118277$.

Neville's iterated interpolation

Neville's iterated interpolation method:

- **Input.** $x_0, \ldots, x_n$ and values $Q_{i,0} = f(x_i)$ for all i.
- For each $i = 1, \ldots, n$: compute $Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$ for $j = 1, \ldots, i$.
- **Output.** Table Q with $P(x) = Q_{n,n}$.

Properties of Neville's method:

1. Add new interpolating nodes easily.
2. Can stop if $|Q_{i,i} - Q_{i-1,i-1}| < \epsilon_{\text{tol}}$.

Divided difference

We can also get the polynomials, not just the interpolating values.

Consider the polynomial $P_n(x)$ of degree n defined by

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1})$$

To make it the Lagrangian interpolating polynomial of f at $x_0, \ldots, x_n$, we need to find a_i s.t. $P_n(x_i) = f(x_i)$ for all x_i.

It is easy to check that:

$$\begin{aligned} P_n(x_0) = a_0 = f(x_0) \quad &\Longrightarrow \quad a_0 = f(x_0) \\ P_n(x_1) = a_0 + a_1(x_1 - x_0) = f(x_1) \quad &\Longrightarrow \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ \vdots \end{aligned}$$

Divided difference

We define the following notations of **divided difference**:

$$f[x_i] = f(x_i)$$

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Once the $(k-1)$th divided differences are determined, we can get the kth divided difference as

$$f[x_0, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0}$$

until we get $f[x_0, \ldots, x_n]$. Then set $a_k = f[x_0, \ldots, x_k]$ for all k:

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, \ldots, x_k](x - x_0) \ldots (x - x_k)$$

Divided difference

We can construct a table of divided difference as follows:

x_0	$f[x_0]$				
x_1	$f[x_1]$	$f[x_0, x_1]$			
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$	
x_4	$f[x_4]$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_3, x_4]$

Divided difference

We can introduce a new notation $F_{i,j} = f[x_{i-j}, \ldots, x_i]$, then the table can be written as

$$
\begin{array}{lllll}
x_0 & F_{0,0} & & & \\
x_1 & F_{1,0} & F_{1,1} & & \\
x_2 & F_{2,0} & F_{2,1} & F_{2,2} & \\
x_3 & F_{3,0} & F_{3,1} & F_{3,2} & F_{3,3} \\
x_4 & F_{4,0} & F_{4,1} & F_{4,2} & F_{4,3} & F_{4,4}
\end{array}
$$

Newton's divided difference formula

Newton's divided difference

- **Input.** $x_0, \ldots, x_n$ and values $F_{i,0} = f(x_i)$ for all i.
- For each $i = 1, \ldots, n$: set $F_{i,j} = \frac{F_{i,j-1} - F_{i-1,j-1}}{x_i - x_{i-j}}$ for $j = 1, \ldots, i$.
- **Output.** $F_{i,i}$ for $i = 0, \ldots, n$, and set

$$P_n(x) = F_{0,0} + \sum_{i=1}^{n} F_{i,i}(x - x_0) \ldots (x - x_{i-1})$$

Special case

In the special case where $x_{i+1} - x_i = h$ for all i, then $x_i = x_0 + ih$.
Now if we want to know the value of f at $x_s = x_0 + sh$ (s can be non-integer), then

$$
\begin{aligned}
P_n(x_s) &= f[x_0] + \sum_{k=1}^{n} f[x_0, \ldots, x_k](x_s - x_0)\ldots(x_s - x_{k-1}) \\
&= f[x_0] + \sum_{k=1}^{n} f[x_0, \ldots, x_k](sh)((s-1)h)\ldots((s-k+1)h) \\
&= f[x_0] + \sum_{k=1}^{n} f[x_0, \ldots, x_k]h^k \frac{s(s-1)\ldots(s-k+1)}{k!} k! \\
&= f[x_0] + \sum_{k=1}^{n} f[x_0, \ldots, x_k]h^k k! \binom{s}{k}
\end{aligned}
$$

Special case

If we adopt the Aitkin's Δ^2 to simplify notations:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1}{h}(f(x_1) - f(x_0)) = \frac{1}{h}\Delta f(x_0)$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{1}{2h}(\frac{1}{h}\Delta f(x_1) - \frac{1}{h}\Delta f(x_0)) = \frac{1}{2h^2}\Delta^2 f(x_0)$$

$$\vdots$$

$$f[x_0, \ldots, x_k] = \cdots = \frac{1}{k!h^k}\Delta^k f(x_0)$$

Newton's divided difference becomes:

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

Backward difference

We can also use the backward differences:

$$\nabla p_n := p_n - p_{n-1} \quad \text{and} \quad \nabla^k p_n = \nabla(\nabla^{k-1} p_n) \qquad [3]$$

Suppose the points are in reverse order: $x_n, x_{n-1}, \ldots, x_0$, then

$$P_n(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + \cdots + f[x_n, \ldots, x_0](x - x_n)\ldots(x - x_1).$$

If $x_s = x_n + sh$ (s is negative non-integer), then we can derive:

$$P_n(x) = f[x_n] + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_n)$$

[3] For example, $\nabla^2 p_n = (p_n - p_{n-1}) - (p_{n-1} - p_{n-2}) = p_n - 2p_{n-1} + p_{n-2}$.

Hermite interpolation

Suppose we also have derivatives $f^{(k)}(x_i)$ at points x_i for $k = 0, \ldots, m_i$, we can find the polynomial $P(x)$ s.t.

$$P^{(k)}(x_i) = f^{(k)}(x_i), \qquad \forall\, i, k$$

The total number of conditions (values) we have is

$$\hat{n} := \sum_{i=0}^{n}(m_i + 1) = (n+1) + \sum_{i=0}^{n} m_i$$

So we can find a polynomial P of degree $\hat{n} - 1$.

Such a polynomial is called an *osculating polynomial*.

Hermite polynomial

We're mostly interested in the case with $m_i = 1, \forall\, i$. That is, we have $f(x_i)$ and $f'(x_i)$ at each x_i.

We want to construct a polynomial $P(x)$ of degree $2n+1$, s.t. $P(x_i) = f(x_i)$ and $P'(x_i) = f'(x_i), \forall\, i$.

Let $L_{n,j}(x)$ be the Lagrange polynomial of degree n such that

$$L_{n,j}(x_i) = \begin{cases} 0, & \text{if} \quad i \neq j \\ 1, & \text{if} \quad i = j \end{cases}$$

We define two polynomials (both of degree $2n+1$):

$$H_{n,j}(x) = (1 - 2(x - x_j)L'_{n,j}(x_j))L^2_{n,j}(x)$$
$$\hat{H}_{n,j}(x) = (x - x_j)L^2_{n,j}(x)$$

Hermite polynomial

Theorem (Construction of Hermite polynomial)

If $f \in C^1[a,b]$ and $x_0, \ldots, x_n \in [a,b]$ are distinct, then the polynomial of least degree that satisfies $P(x_i) = f(x_i)$ and $P'(x_i) = f'(x_i)$ is

$$H_{2n+1}(x) := \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \hat{H}_{n,j}(x)$$

which has degree $\leq 2n+1$.

Hermite polynomial

Proof.

It's clear the degree $\leq 2n+1$. Also,

$$H_{n,j}(x_i) = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \quad \text{and} \quad \hat{H}_{n,j}(x_i) = 0, \forall\, i$$

So $H_{2n+1}(x_i) = f(x_i)\ \forall\, i$. Also

$$H'_{n,j}(x) = -2L'_{n,j}(x_j)L^2_{n,j}(x) + (2 - 4(x - x_j)L'_{n,j}(x_j))L_{n,j}(x)L'_{n,j}(x)$$
$$\hat{H}'_{n,j}(x) = L^2_{n,j}(x) + 2(x - x_j)L_{n,j}(x)L'_{n,j}(x)$$

Therefore

$$H'_{n,j}(x_i) = 0 \quad \forall\, i, \quad \text{and} \quad \hat{H}'_{n,j}(x_i) = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$$

Hence $H'_{2n+1}(x) = f'(x_i)$, $\forall\, i$. □

Hermite polynomials

We can also construct Hermite polynomials using divided difference.

Suppose we have $x_0, x_1, \ldots, x_n$ and $f(x_i), f'(x_i)$ are given. Define $z_{2i} = z_{2i+1} = x_i$ for $i = 0, \ldots, n$

For example, $z_0 = z_1 = x_0$, $z_2 = z_3 = x_1$, etc.

Now we have $z_0, z_1, \ldots, z_{2n+1}$, total of $2(n+1)$ points. So

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n} f[z_0, \ldots, z_k](x - z_0) \cdots (x - z_k)$$

and use $f'(x_i)$ as $f[z_{2i}, z_{2i+1}]$ for all $i = 0, \ldots, n$.

Hermite polynomial

Then we construct the table as follows,

$$
\begin{array}{llllll}
z_0 = x_0 & f[z_0] = f(x_0) & & & & \\
z_1 = x_0 & f[z_1] = f(x_0) & f[z_0, z_1] = f'(x_0) & & & \\
z_2 = x_1 & f[z_2] = f(x_1) & f[z_1, z_2] = \frac{f[z_2]-f[z_1]}{z_2-z_1} & f[z_0, z_1, z_2] & & \\
z_3 = x_1 & f[z_3] = f(x_1) & f[z_2, z_3] = f'(x_1) & f[z_1, z_2, z_3] & f[z_0, z_1, z_2, z_3] & \\
z_4 = x_2 & f[z_4] = f(x_2) & f[z_3, z_4] = \frac{f[z_4]-f[z_3]}{z_4-z_3} & f[z_2, z_3, z_4] & f[z_1, z_2, z_3, z_4] & f[z_0, z_1, z_2, z_3, z_4] \\
z_5 = x_3 & f[z_5] = f(x_3) & f[z_4, z_5] = f'(x_2) & f[z_3, z_4, z_5] & f[z_2, z_3, z_4, z_5] & f[z_1, z_2, z_3, z_4, z_5] \\
\vdots & & & & &
\end{array}
$$

Hermite interpolation

Hermite interpolation polynomial

- **Input.** Distinct $x_0, \ldots, x_n$, $f(x_i), f'(x_i)\ \forall\, i$.
- For $i = 0, \ldots, n$, do (# Assign values $Q_{\cdot,0}$, $Q_{\cdot,1}$)
  1. Set $z_{2i} = z_{2i+1} = x_i$, $Q_{2i,0} = Q_{2i+1,0} = f(x_i)$, $Q_{2i+1,1} = f'(x_i)$.
  2. If $i \neq 0$, then set $Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}$.
- For $i = 2, \ldots, 2n+1$ and $j = 2, \ldots, i$, set

$$Q_{i,j} = \frac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$$

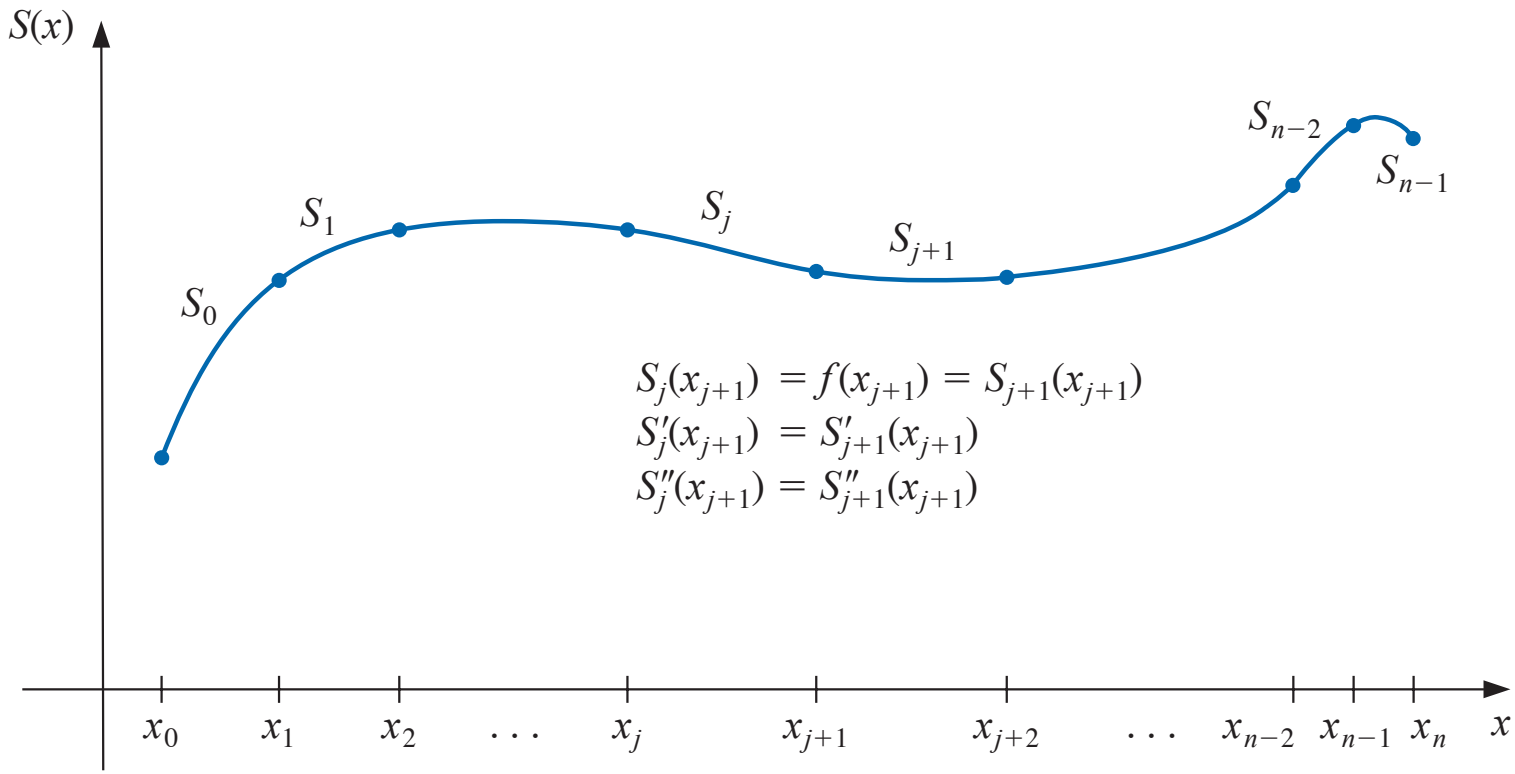
- **Output.** Hermite polynomial coeff. $Q_{0,0}, \ldots, Q_{2n+1,2n+1}$, s.t.

$$\begin{aligned} H(x) =& Q_{0,0} + Q_{1,1}(x - x_0) + Q_{2,2}(x - x_0)^2 + \cdots \\ &+ Q_{2n+1,2n+1}(x - x_0)^2 \ldots (x - x_n)^2 \end{aligned}$$

Cubic spline interpolation

High-degree polynomial fitting has strong oscillations.

Can we get a piecewise "low degree" polynomial interpolation instead?



Cubic spline interpolation

Suppose we are given $x_0, \ldots, x_n$ and $f(x_i)$, $\forall\, i$, we want to find a cubic spline interpolation $S(x)$, s.t.

1. $S(x)$ is a cubic polynomial, denoted by $S_j(x)$, on $[x_j, x_{j+1}]$;
2. $S_j(x_j) = f(x_j)$, $S_j(x_{j+1}) = f(x_{j+1})$
3. $S_j(x_j) = S_{j+1}(x_j)$ for all j (consequence of Item 2.)
4. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for all j
5. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for all j
6. One of the following boundary condition is satisfied:
   - $S''(x_0) = S''(x_n) = 0$ (natural/free boundary condition)
   - $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary condition)

Cubic spline interpolation

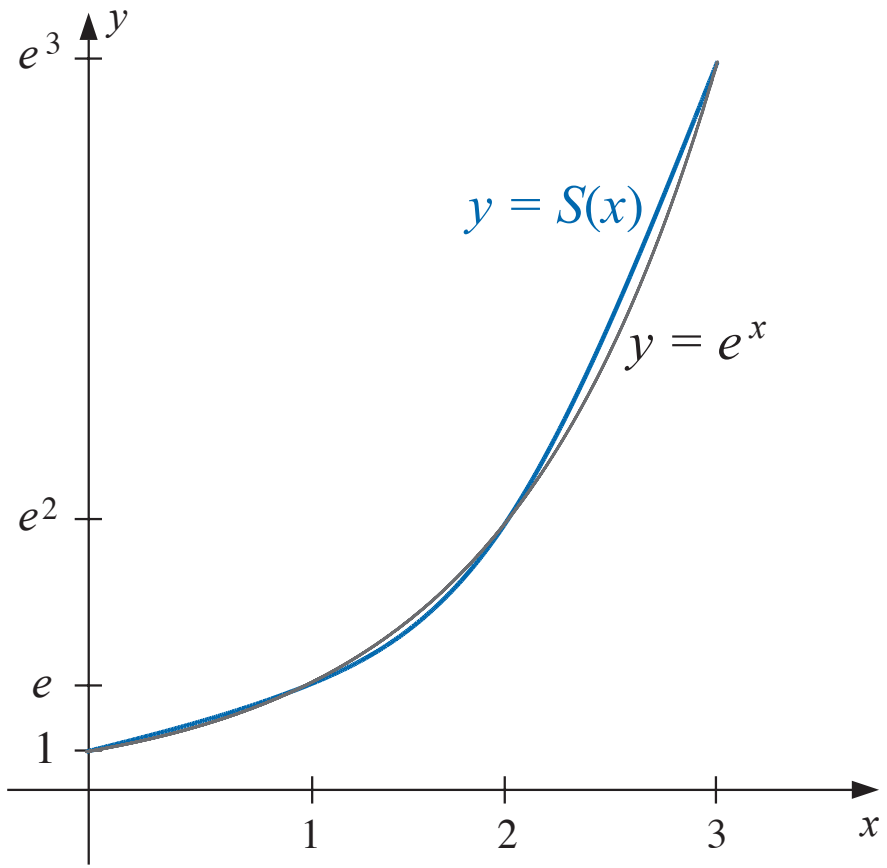
Remarks

1. $S(x)$ only agrees with $f(x)$ at x_i, not necessarily $f'(x)$.
2. Clamped boundary condition is more accurate than natural boundary condition, but needs $f'(a), f'(b)$.

Example

Example (Construct a natural cubic spline)

Construct natural cubic spline for $f(x) = e^x$ using $x_i = i$ for $i = 0, 1, 2, 3$.



Construction of cubic splines

Given $\{(x_i, f(x_i)) : i = 0, \ldots, n\}$, we need to construct n cubic polynomials, each with 4 coefficients

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \text{ on } [x_j, x_{j+1}], \ \forall j$$

So we have $4n$ unknowns to determine:

$$a_j, b_j, c_j, d_j, \quad \text{for } j = 0, \ldots, n-1$$

Construction of cubic splines

The cubic spline conditions will determine these $4n$ coefficients uniquely ($h_j := x_{j+1} - x_j$) according to the 6 rules:

1. By definition of S_j.
2. Since $S_j(x_j) = a_j = f(x_j)$, we get a_j for $j = 0, \ldots, n-1$.
3. $a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$.
4. $S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$, therefore $S_j'(x_j) = b_j$ and $b_{j+1} = S_{j+1}'(x_{j+1}) = S_j'(x_{j+1}) = b_j + 2c_j h_j + 3d_j h_j^2$.
5. $S_j''(x) = 2c_j + 6d_j(x - x_j)$. Then we have $S_j''(x_j) = 2c_j$. So $2c_{j+1} = S_{j+1}''(x_{j+1}) = S_j''(x_j) = 2c_j + 6d_j h_j$.
6. Use the required boundary condition.
   - Natural boundary condition: $c_0 = c_n = 0$
   - Clamped boundary condition: $b_0 = f'(a)$, $b_n = f'(b)$.

Construction of cubic splines

As we have known the values of a_j, we can combine equations from the last 3 items to solve for c_j and obtain

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

for each $j = 1, \ldots, n-1$. If we assume natural splines with $S_0''(x_0) = S_{n-1}''(x_n) = 0$, then $c_0 = c_n = 0$.

Section 4

Numerical Differentiation and Integration

Numerical differentiation

Recall the definition of derivative is

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

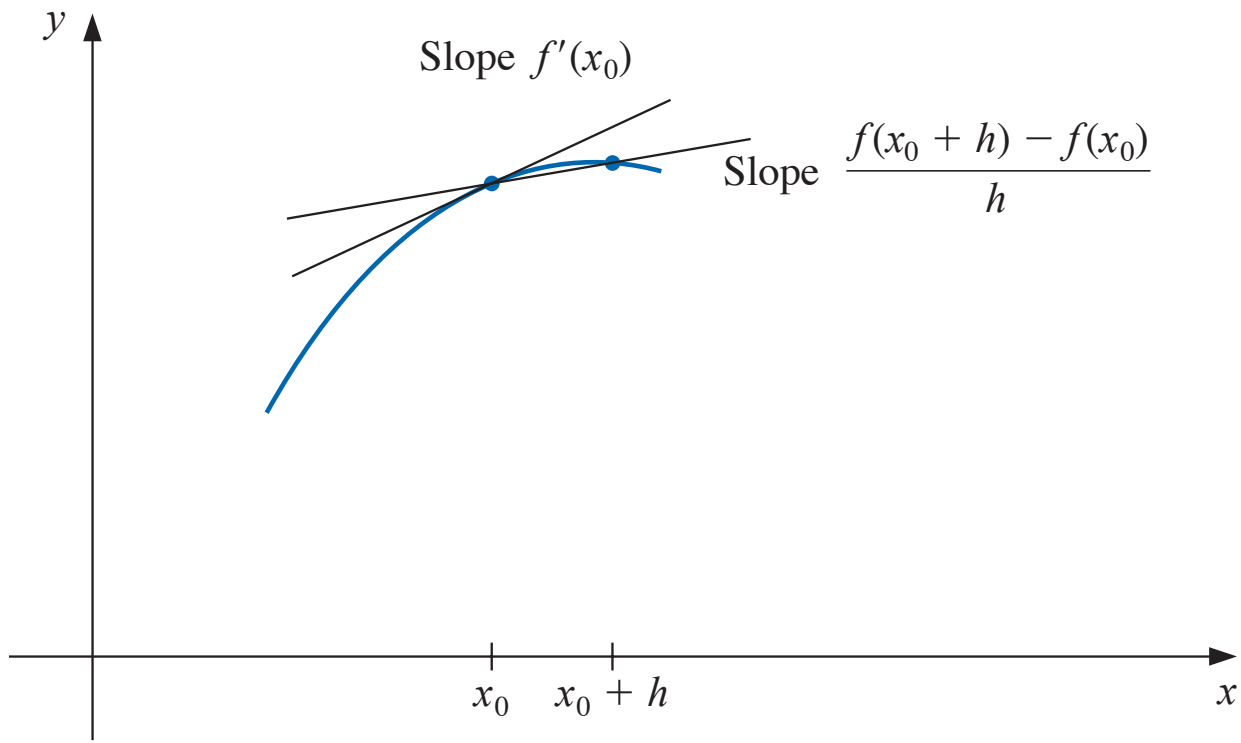
We can approximate $f'(x_0)$ by

$$\frac{f(x_0 + h) - f(x_0)}{h}, \quad \text{for some small } h$$

Numerical differentiation

Approximate $f'(x_0)$ by

$$\frac{f(x_0+h)-f(x_0)}{h}, \quad \text{for some small } h$$



How to quantify the error of this approximation?

Numerical differentiation

If $f \in C^2$, then Taylor's theorem says $\exists\, \xi \in (x_0, x_0 + h)$ s.t.

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2}f''(\xi)h^2$$

$$\iff \quad f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{1}{2}f''(\xi)h$$

If $\exists\, M > 0$ s.t. $|f''(x)| \leq M$ for all x near x_0, then

$$\text{Error} = \left| f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h} \right| = \left| \frac{1}{2}f''(\xi)h \right| \leq \frac{Mh}{2}$$

So the error is of order "$O(h)$".

Example

Example (Error of numerical differentiations)

Let $f(x) = \ln(x)$ at $x_0 = 1.8$. Use $h = 0.1, 0.05, 0.01$ to approximate $f'(x_0)$. Determine the approxiamtion errors.

Solution. We compute for $h = 0.1, 0.05, 0.01$ that

$$\frac{f(1.8+h) - f(1.8)}{h} = \frac{\ln(1.8+h) - \ln(1.8)}{h}$$

Then $|f''(x)| = |-\frac{1}{x^2}| \leq \frac{1}{1.8^2} =: M$ for all $x > 1.8$. Error is bounded by $\frac{Mh}{2}$.

Example (Error of numerical differentiations)

Let $f(x) = \ln(x)$ at $x_0 = 1.8$. Use $h = 0.1, 0.05, 0.01$ to approximate $f'(x_0)$. Determine the approxiamtion errors.

Solution (cont.)

h	$\frac{f(1.8+h)-f(1.8)}{h}$	$\frac{Mh}{2}$
0.10	0.5406722	0.0154321
0.05	0.5479795	0.0077160
0.01	0.5540180	0.0015432

The exact value is $f'(1.8) = \frac{1}{1.8} = 0.55\bar{5}$.

Three-point endpoint formula

Recall the Lagrange interpolating polynomial for $x_0, \ldots, x_n$ is

$$f(x) = \sum_{k=0}^{n} f(x_k) L_k(x) + \frac{(x - x_0) \cdots (x - x_n)}{(n+1)!} f^{(n+1)}(\xi(x))$$

Suppose we have $x_0, x_1 \triangleq x_0 + h, x_2 \triangleq x_0 + 2h$, then

$$f(x) = \sum_{k=0}^{2} f(x_k) L_k(x) + \frac{(x - x_0)(x - x_1)(x - x_2)}{6} f^{(3)}(\xi(x))$$

where $\xi(x) \in (x_0, x_0 + 2h)$.

Three-point endpoint formula

Take derivative w.r.t. x of

$$f(x) = \sum_{k=0}^{2} f(x_k) L_k(x) + \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f^{(3)}(\xi(x))$$

and set $x = x_0$ yields[4] the **Three-point endpoint formula**:

$$f'(x_0) = \frac{1}{2h}\left[-3f(x_0) + 4f(x_0+h) - f(x_0+2h)\right] + \frac{h^2}{3} f^{(3)}\left(\xi(x_0)\right)$$

where $\xi(x_0) \in (x_0, x_0+2h)$.

[4] Note that $\frac{(x-x_0)(x-x_1)(x-x_2)}{6} \frac{\mathrm{d}f^{(3)}(\xi(x))}{\mathrm{d}x}|_{x=x_0} = 0$.

Three-point midpoint formula

Suppose we have $x_{-1} = x_0 - h, x_0, x_1 \triangleq x_0 + h$, then

$$f(x) = \sum_{k=-1}^{1} f(x_k) L_k(x) + \frac{(x - x_{-1})(x - x_0)(x - x_1)}{6} f^{(3)}(\xi_1)$$

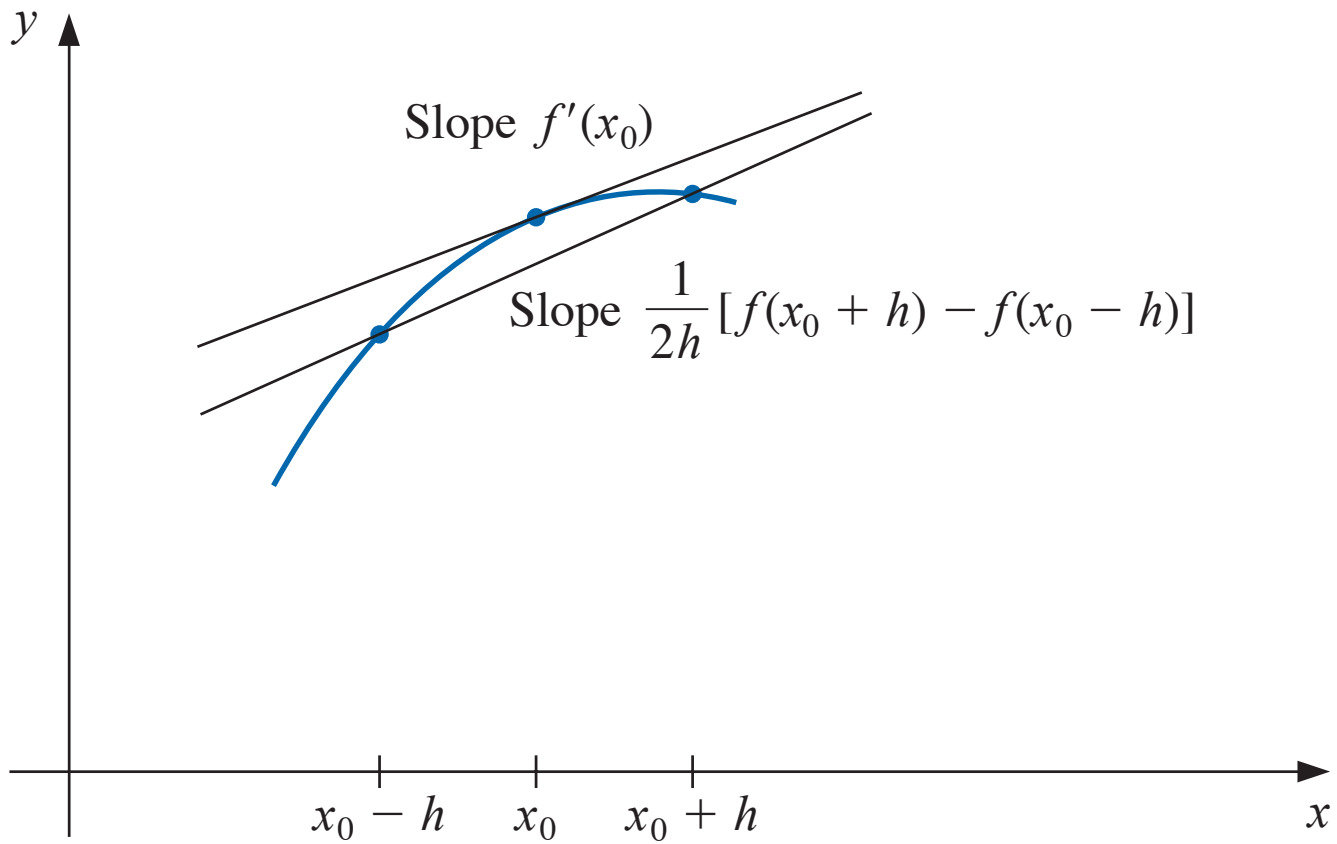
where $\xi_1 \in (x_0 - h, x_0 + h)$.

Take derivative w.r.t. x, and set $x = x_0$ yields **Three-point midpoint formula**:

$$f'(x_0) = \frac{1}{2h}\left[f(x_0 + h) - f(x_0 - h)\right] - \frac{h^2}{6} f^{(3)}(\xi_1)$$

Three-point midpoint formula

Illustration of **Three-point midpoint formula**:



Five-point midpoint formula

We can also consider $x_k = x_0 + kh$ for $k = -2, -1, 0, 1, 2$, then

$$f(x) = \sum_{k=-2}^{2} f(x_k) L_k(x) + \frac{\prod_{k=-2}^{2}(x - x_k)}{5!} f^{(5)}(\xi_0)$$

where $\xi_0 \in (x_0 - 2h, x_0 + 2h)$.

Show that you can get the **Five-point midpoint formula**:

$$\begin{aligned} f'(x_0) = & \frac{1}{12h}\left[f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)\right] \\ & + \frac{h^4}{30} f^{(5)}(\xi_0) \end{aligned}$$

Five-point endpoint formula

We can also consider $x_k = x_0 + kh$ for $k = 0, 1, \ldots, 4$, then

$$f(x) = \sum_{k=0}^{4} f(x_k) L_k(x) + \frac{\prod_{k=0}^{4}(x - x_k)}{5!} f^{(5)}(\xi_0)$$

where $\xi_0 \in (x_0, x_0 + 4h)$.

Show that you can get the **Five-point endpoint formula**:

$$\begin{aligned} f'(x_0) = & \frac{1}{12h}\Big[-25f(x_0) + 48f(x_0 + h) - 36f(x_0 + 2h) \\ & + 16f(x_0 + 3h) - 3f(x_0 + 4h)\Big] + \frac{h^4}{5} f^{(5)}(\xi_0) \end{aligned}$$

Example

Example (3-point and 5-point formulas)

Use the values in the table to find $f'(2.0)$:

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Compare your result with the true value $f'(2) = 22.167168$.

Hint: Use three-point midpoint formula with $h = 0.1, 0.2$, endpoint with $h = \pm 0.1$, and five-pint midpoint formula with $h = 0.1$.

Second derivative midpoint formula

Expand f in a third Taylor polynomial about a point x_0 and evaluate at $x_0 + h$ and $x_0 - h$:

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2 + \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_1)h^4$$

$$f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{1}{2}f''(x_0)h^2 - \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_{-1})h^4$$

where $\xi_{\pm 1}$ is between x_0 and $x_0 \pm h$.

Adding the two and using IVT $f^{(4)}(\xi) = \frac{1}{2}\left[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})\right]$ (assuming $f \in C^4$) yield:

$$f''(x_0) = \frac{1}{h^2}\left[f(x_0 - h) - 2f(x_0) + f(x_0 + h)\right] - \frac{h^2}{12}f^{(4)}(\xi)$$

where $x_0 - h < \xi < x_0 + h$.

Roundoff error instability

Recall we have three-point midpoint approximation

$$f'(x_0) = \frac{1}{2h}\left[f(x_0+h) - f(x_0-h)\right] - \frac{h^2}{6}f^{(3)}(\xi_1)$$

for $\xi_1 \in (x_0-h, x_0+h)$.

Will we get better accuracy as $h \to 0$? Not necessarily.

Round-off error instability

In numerical computations, round-off error is inevitable:

$$f(x_0+h) = \tilde{f}(x_0+h) + e(x_0+h)$$
$$f(x_0-h) = \tilde{f}(x_0-h) + e(x_0-h)$$

Hence we're approximating $f'(x_0)$ by $\frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h}$ with error:

$$f'(x_0) - \frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h} = \frac{e(x_0+h)-e(x_0-h)}{2h} - \frac{h^2}{6}f^{(3)}(\xi_1)$$

Suppose $|e(x)| \leq \varepsilon$, $\forall\, x$, then the error bound is:

$$\left| f'(x_0) - \frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h} \right| \leq \frac{\varepsilon}{h} + \frac{h^2}{6}M$$

So the error does not go to 0 as $h \to 0$, due to the round-off error.

Richardson's extrapolation

Goal: generate high-accuracy results by *low-order* formula.

Suppose we have formula $N_1(h)$ to approximate M with [5]

$$M = N_1(h) + K_1 h + K_2 h^2 + K_3 h^3 + \cdots$$

with some unknown $K_1, K_2, K_3, \ldots$.

For h small enough, the error is dominated by $K_1 h$, then

$$M = N_1\left(\frac{h}{2}\right) + K_1 \frac{h}{2} + K_2 \frac{h^2}{4} + K_3 \frac{h^3}{8} + \cdots$$

[5] E.g., $M = f'(x_0)$ and $N_1(h) = \frac{f(x_0+h)-f(x_0)}{h}$.

Richardson's extrapolation

Therefore

$$M = N_1\left(\frac{h}{2}\right) + \left[N_1\left(\frac{h}{2}\right) - N_1(h)\right] + K_2\left(\frac{h^2}{2} - h^2\right) + K_3\left(\frac{h^3}{4} - h^3\right) + \cdots$$

Define

$$N_2(h) = N_1\left(\frac{h}{2}\right) + \left[N_1\left(\frac{h}{2}\right) - N_1(h)\right]$$

then M can be approximated by $N_2(h)$ with order $O(h^2)$:

$$M = N_2(h) - \frac{K_2}{2}h^2 - \frac{3K_3}{4}h^3 - \cdots$$

Example

Example (Richardson's extrapolation)

Let $f(x) = \ln(x)$. Approximate f' at $x_0 = 1.8$ with forward difference using $h = 0.1$ and $h = 0.05$. Then approximate using $N_2(0.1)$.

Solution. We know the forward difference is $O(h)$, and

$$N_1(h) = \frac{f(x_0 + h) - f(x_0)}{h} = \begin{cases} 0.5406722, & \text{for } h = 0.1 \\ 0.5479795, & \text{for } h = 0.05 \end{cases}$$

$$N_2(0.1) = N_1(0.05) + (N_1(0,05) - N_1(0.1)) = 0.555287.$$

Formula	$N_1(0.1)$	$N_1(0.05)$	$N_2(0.1)$
Error	1.5×10^{-2}	7.7×10^{-3}	2.7×10^{-4}

Richardson's extrapolation

Suppose $M = N_1(h) + K_1 h^2 + K_2 h^4 + K_3 h^6 + \ldots$, then for $j = 2, 3, \ldots$, we have $O(h^{2j})$ approximation:

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{4^{j-1} - 1}$$

We can show the order of generating these $N_j(h)$ [6]:

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
1: $N_1(h)$			
2: $N_1(\frac{h}{2})$	3: $N_2(h)$		
4: $N_1(\frac{h}{4})$	5: $N_2(\frac{h}{2})$	6: $N_3(h)$	
7: $N_1(\frac{h}{8})$	8: $N_2(\frac{h}{4})$	9: $N_3(\frac{h}{2})$	10: $N_4(h)$

[6] Exercise: write a computer program for Richardson's extrapolation.

Example

Example (Richardson's extrapolation)

Consider approximation of $f'(x_0)$:

$$f'(x_0) = \frac{1}{2h}\left[f(x_0+h) - f(x_0-h)\right] - \frac{h^2}{6}f'''(x_0) - \frac{h^4}{120}f^{(5)}(x_0) - \cdots$$

Find the approximation errors of order $O(h^2), O(h^4), O(h^6)$ for $f'(2.0)$ when $f(x) = xe^x$ and $h = 0.2$.

Solution. We have $O(h^2)$ approximation

$$f'(x_0) = N_1(h) - \frac{h^2}{6}f'''(x_0) - \frac{h^4}{120}f^{(5)}(x_0) - \cdots$$

where $N_1(h) = \frac{1}{2h}\left[f(x_0+h) - f(x_0-h)\right]$. Then compute $N_1(h), N_1(\frac{h}{2}), N_2(h), N_1(\frac{h}{4}), N_2(\frac{h}{2}), \ldots$ in order.

Numerical integration

Recall that Lagrange interpolation of f by

$$f(x) = \underbrace{\sum_{i=0}^{n} f(x_i) L_{n,i}(x)}_{\text{Lagrange polynomial } P_n(x)} + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

So we can take integral on both sides:

$$\begin{aligned} \int_a^b f(x)\,\mathrm{d}x &= \int_a^b \sum_{i=0}^{n} f(x_i)\, L_{n,i}(x)\,\mathrm{d}x + \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)\,\mathrm{d}x \\ &= \sum_{i=0}^{n} a_i f(x_i) + E(f) \end{aligned}$$

where for $i = 0, \ldots, n$,

$$a_i = \int_a^b L_{n,i}(x)\,\mathrm{d}x \text{ and } E(f) = \frac{1}{(n+1)!} \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)\,\mathrm{d}x$$

Trapezoidal rule

Suppose we know f at $x_0 = a$ and $x_1 = b$, then

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)$$

Then taking integral of f yields

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \int_{x_0}^{x_1} \left[\frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)\right] \mathrm{d}x \\ + \frac{1}{2}\int_{x_0}^{x_1} f''(\xi(x))\,(x - x_0)\,(x - x_1)\,\mathrm{d}x\end{aligned}$$

Trapezoidal rule

Integral of the first term on the right is straightforward.

Note that the second term on the right is

$$\begin{aligned}
&\int_{x_0}^{x_1} f''(\xi(x))\,(x-x_0)\,(x-x_1)\,\mathrm{d}x \\
&= f''(\xi)\int_{x_0}^{x_1} (x-x_0)\,(x-x_1)\,\mathrm{d}x \\
&= f''(\xi)\left[\frac{x^3}{3} - \frac{(x_1+x_0)}{2}x^2 + x_0x_1x\right]_{x_0}^{x_1} \\
&= -\frac{h^3}{6}f''(\xi)
\end{aligned}$$

where $\xi \in (x_0, x_1)$ by MVT for integrals and

Trapezoidal rule

Therefore, we obtain

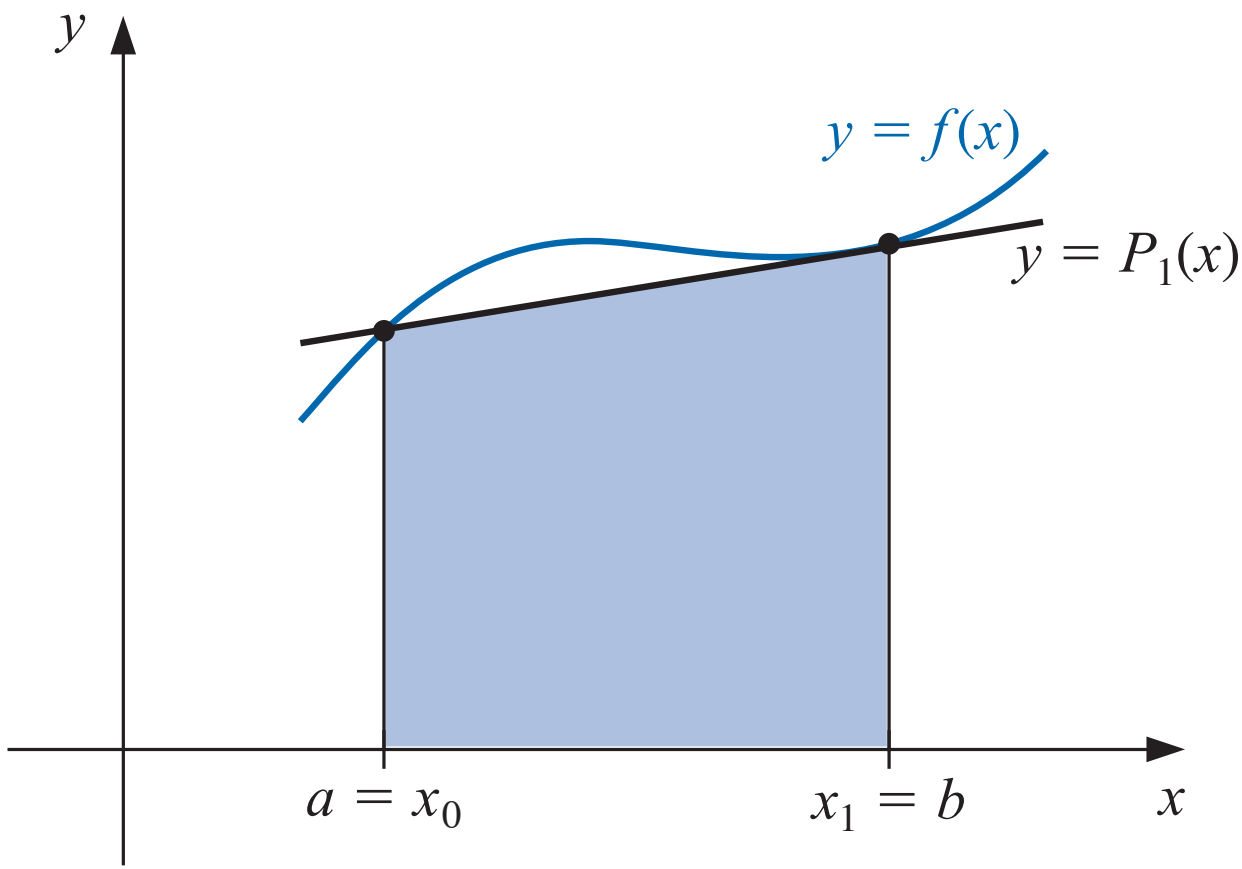
$$\int_a^b f(x)dx = \left[\frac{(x-x_1)^2}{2(x_0-x_1)}f(x_0) + \frac{(x-x_0)^2}{2(x_1-x_0)}f(x_1)\right]_{x_0}^{x_1} - \frac{h^3}{12}f''(\xi)$$
$$= \frac{(x_1-x_0)}{2}\left[f(x_0)+f(x_1)\right] - \frac{h^3}{12}f''(\xi)$$

Trapezoidal rule:

$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left[f(x_0)+f(x_1)\right] - \frac{h^3}{12}f''(\xi)$$

Trapezoidal rule

Illustration of Trapezoidal rule:



Simpson's rule

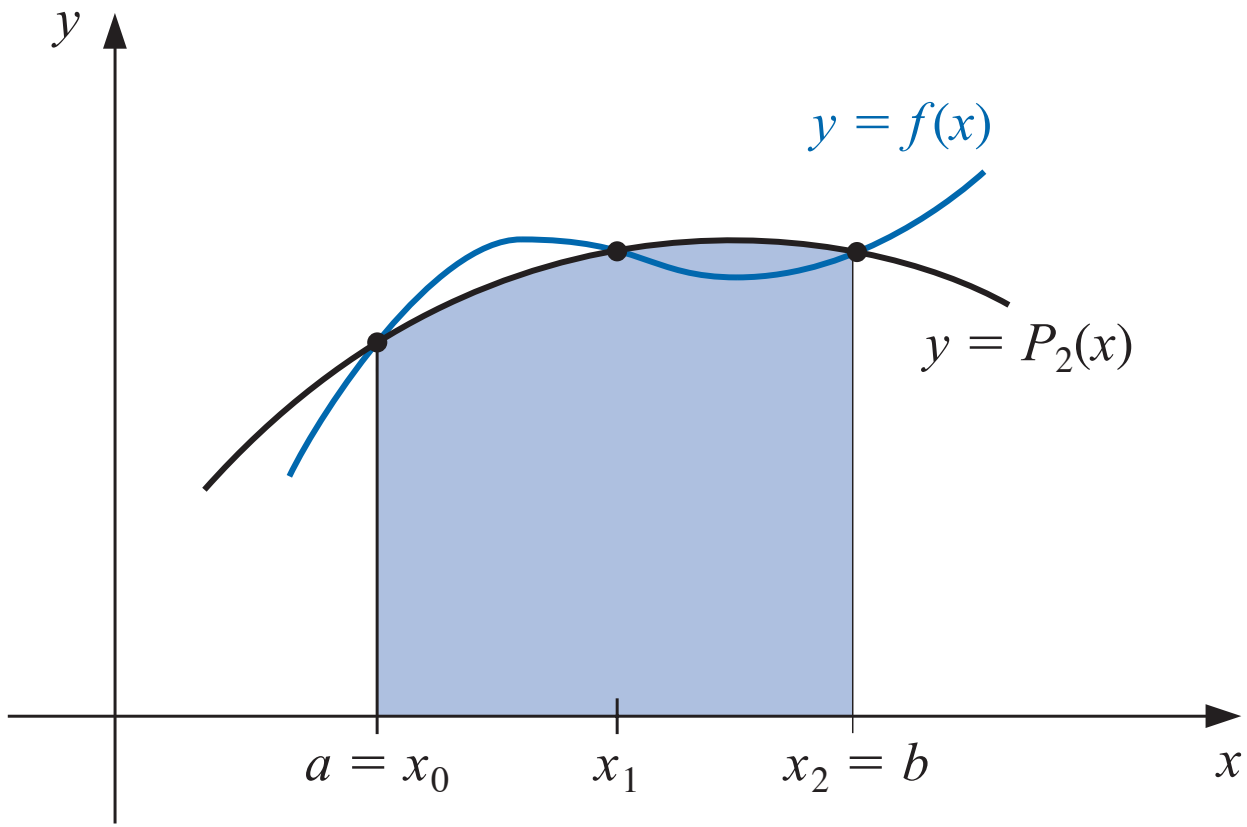
If we have values of f at $x_0 = a$, $x_1 = \frac{a+b}{2}$, and $x_2 = b$. Then

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \int_{x_0}^{x_2} \Bigg[& \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) \\ & + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)\Bigg]\,\mathrm{d}x \\ & + \int_{x_0}^{x_2} \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f^{(3)}(\xi(x))\,\mathrm{d}x\end{aligned}$$

Simpson's rule

With similar idea, we can derive the **Simpson's rule**:

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] - \frac{h^5}{90}f^{(4)}(\xi)$$



Example

Example (Trapezoidal and Simpson's rules for integration)

Compare Trapezoidal and Simpson's rules on $\int_0^2 f(x)\,\mathrm{d}x$ where f is

(a) x^2 (b) x^4 (c) $(x+1)^{-1}$
(d) $\sqrt{1+x^2}$ (e) $\sin x$ (f) e^x

Solution. Apply the the formulas respectively to get:

Problem	(a)	(b)	(c)	(d)	(e)	(f)
$f(x)$	x^2	x^4	$(x+1)^{-1}$	$\sqrt{1+x^2}$	$\sin x$	e^x
Exact value	2.667	6.400	1.099	2.958	1.416	6.389
Trapezoidal	4.000	16.000	1.333	3.326	0.909	8.389
Simpson's	2.667	6.667	1.111	2.964	1.425	6.421

Newton-Cotes formula

We can follow the same idea to get higher-order approximations, called the **Netwon-Cotes** formulas.

For $n = 3$ where $\xi \in (x_0, x_3)$:

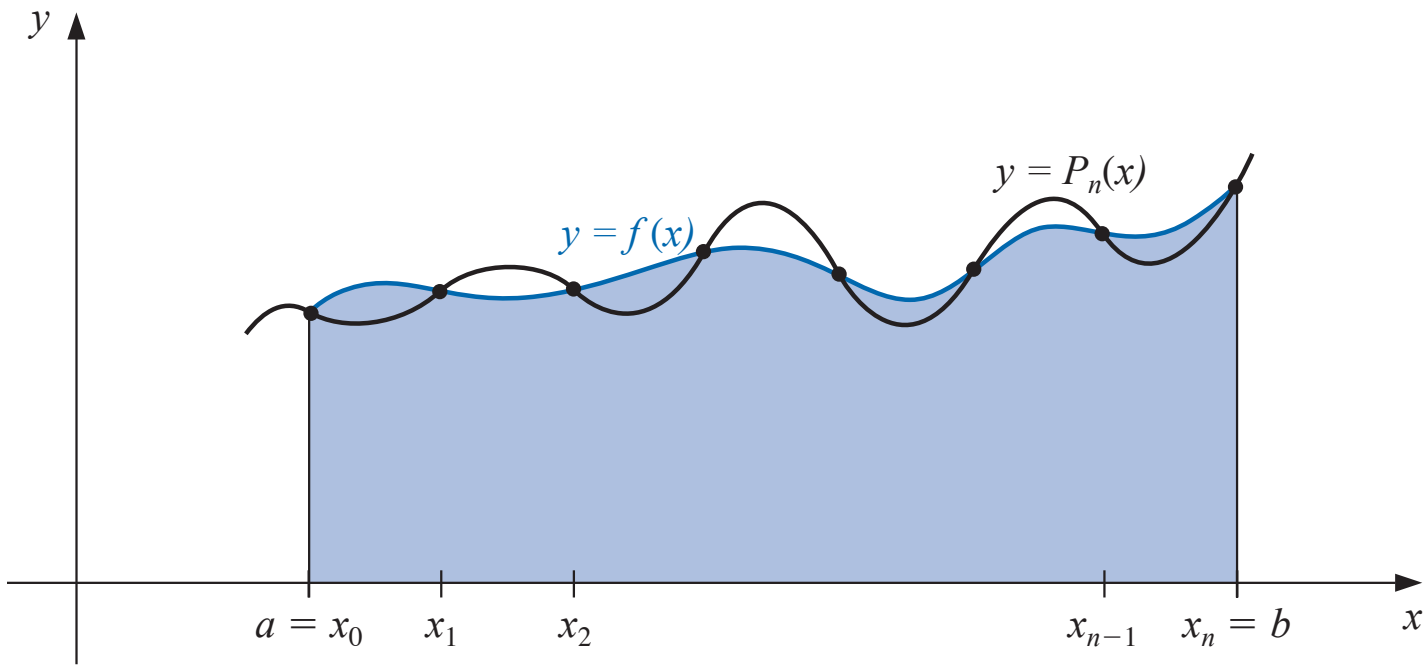
$$\int_{x_0}^{x_3} f(x)\,\mathrm{d}x = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right] - \frac{3h^5}{80} f^{(4)}(\xi)$$

For $n = 4$ where $\xi \in (x_0, x_4)$:

$$\begin{aligned}\int_{x_0}^{x_4} f(x)\,\mathrm{d}x =& \frac{2h}{45}\left[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\right] \\ & - \frac{8h^7}{945} f^{(6)}(\xi)\end{aligned}$$

Composite numerical integration

Problem with Newton-Cotes rule for high degree is oscillations.

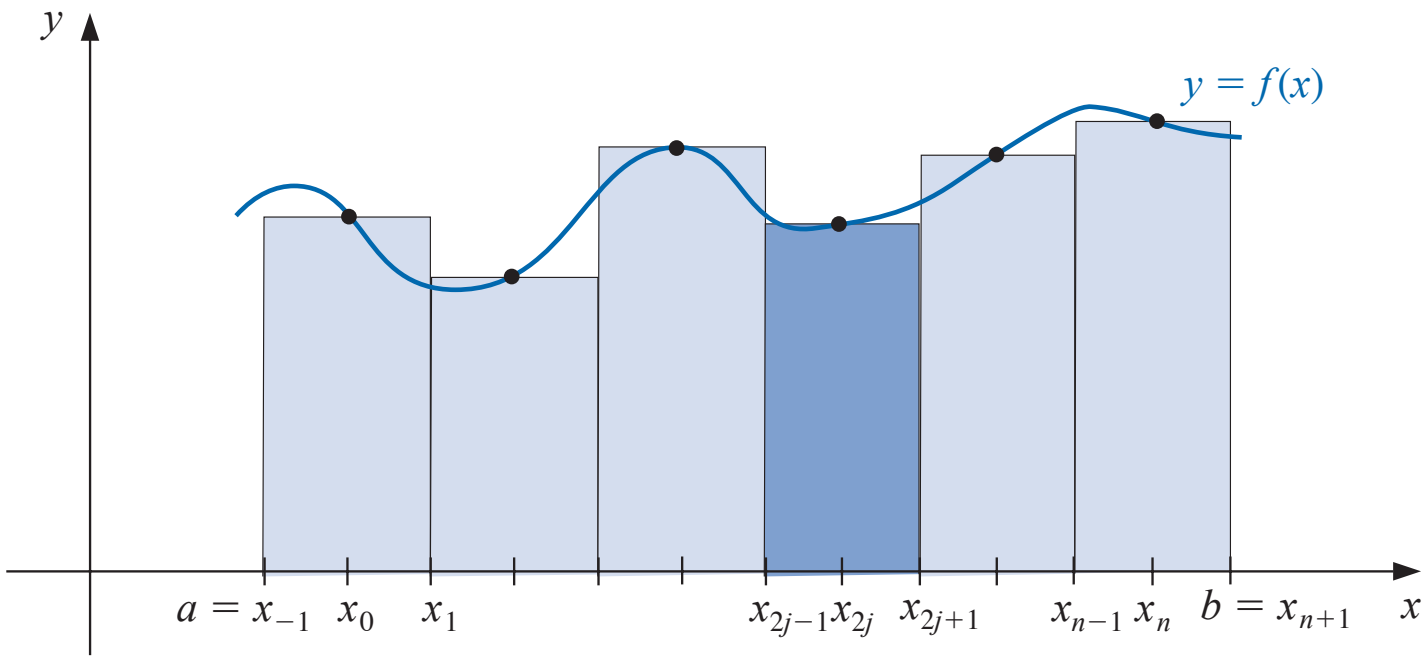


Instead, we can approximate the integral "piecewisely".

Composite midpoint rule

Let $x_{-1} = a, x_0, x_1, \ldots, x_n, x_{n+1} = b$ be a uniform partition of $[a, b]$ with $h = \frac{b-a}{n+2}$. Then we obtain the **composite midpoint rule**:

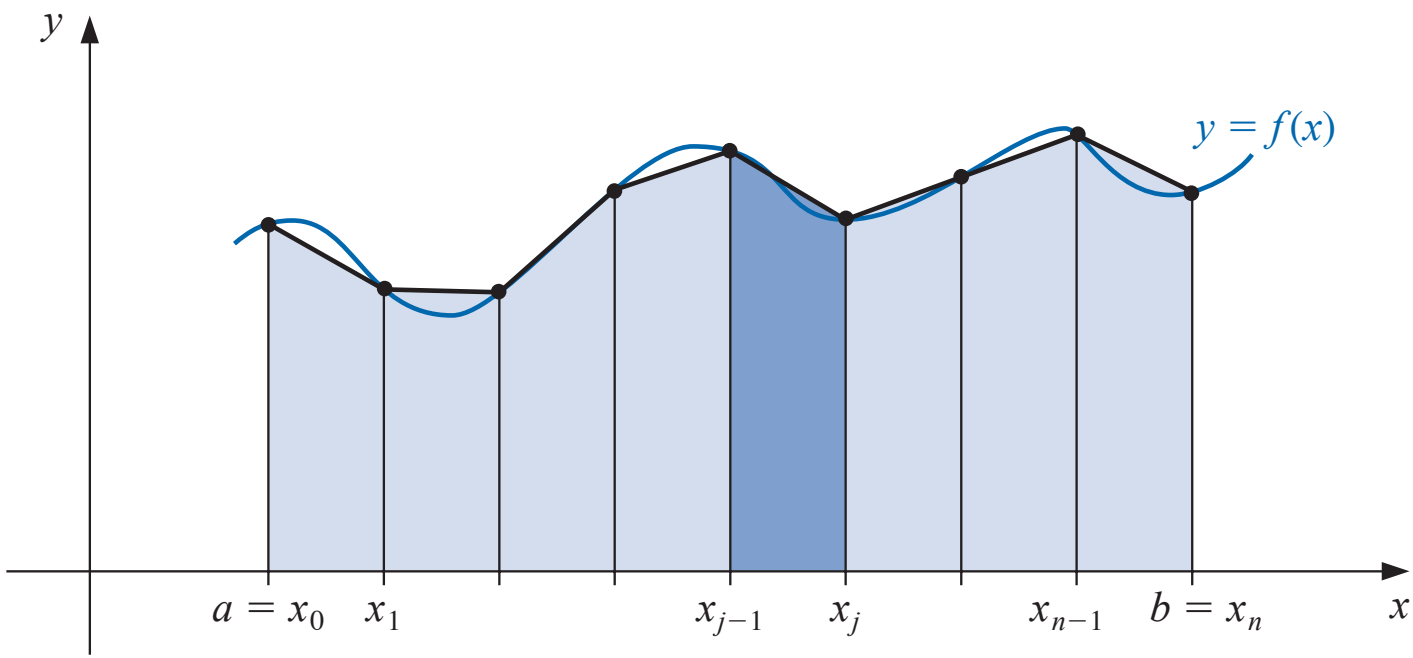
$$\int_a^b f(x)\,\mathrm{d}x = 2h \sum_{j=0}^{n/2} f\left(x_{2j}\right) + \frac{b-a}{6} h^2 f''(\mu)$$



Composite trapezoidal rule

Let $x_0 = a, x_1, \ldots, x_n = b$ be a uniform partition of $[a, b]$ with $h = \frac{b-a}{n}$. Then we obtain the **composite Trapezoidal rule**:

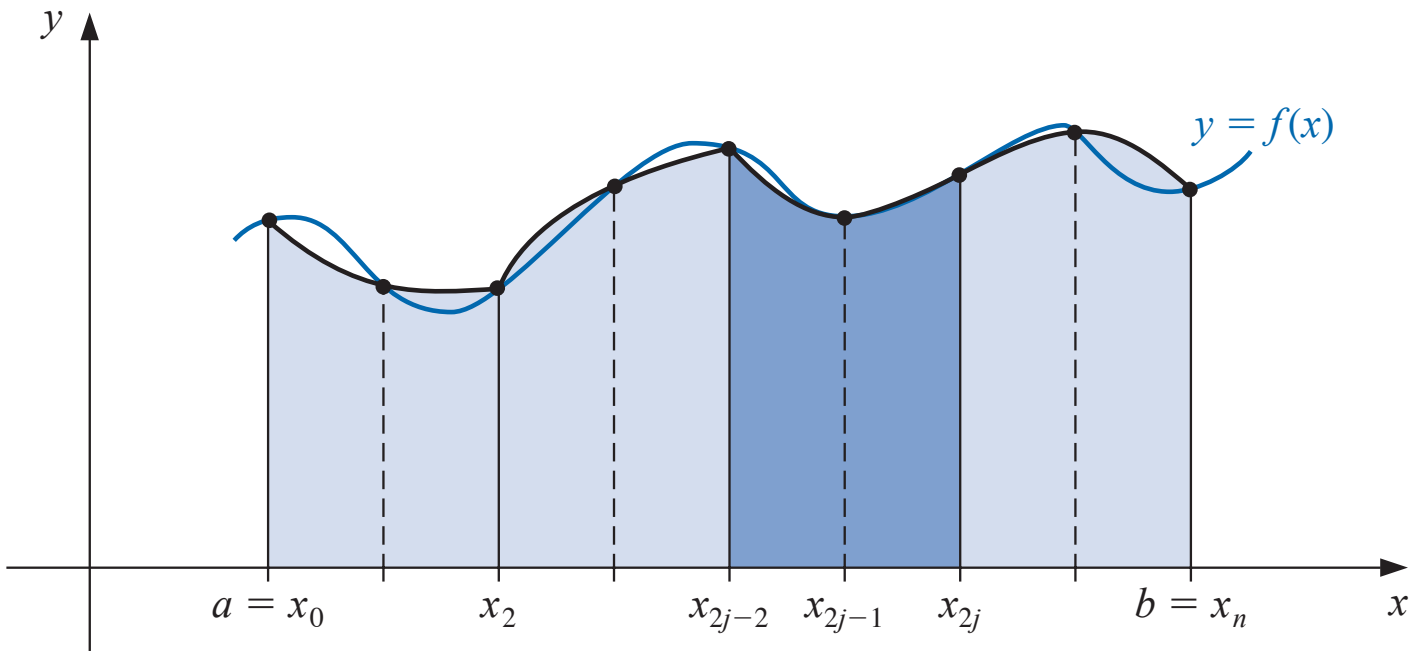
$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu)$$



Composite Simpson's rule

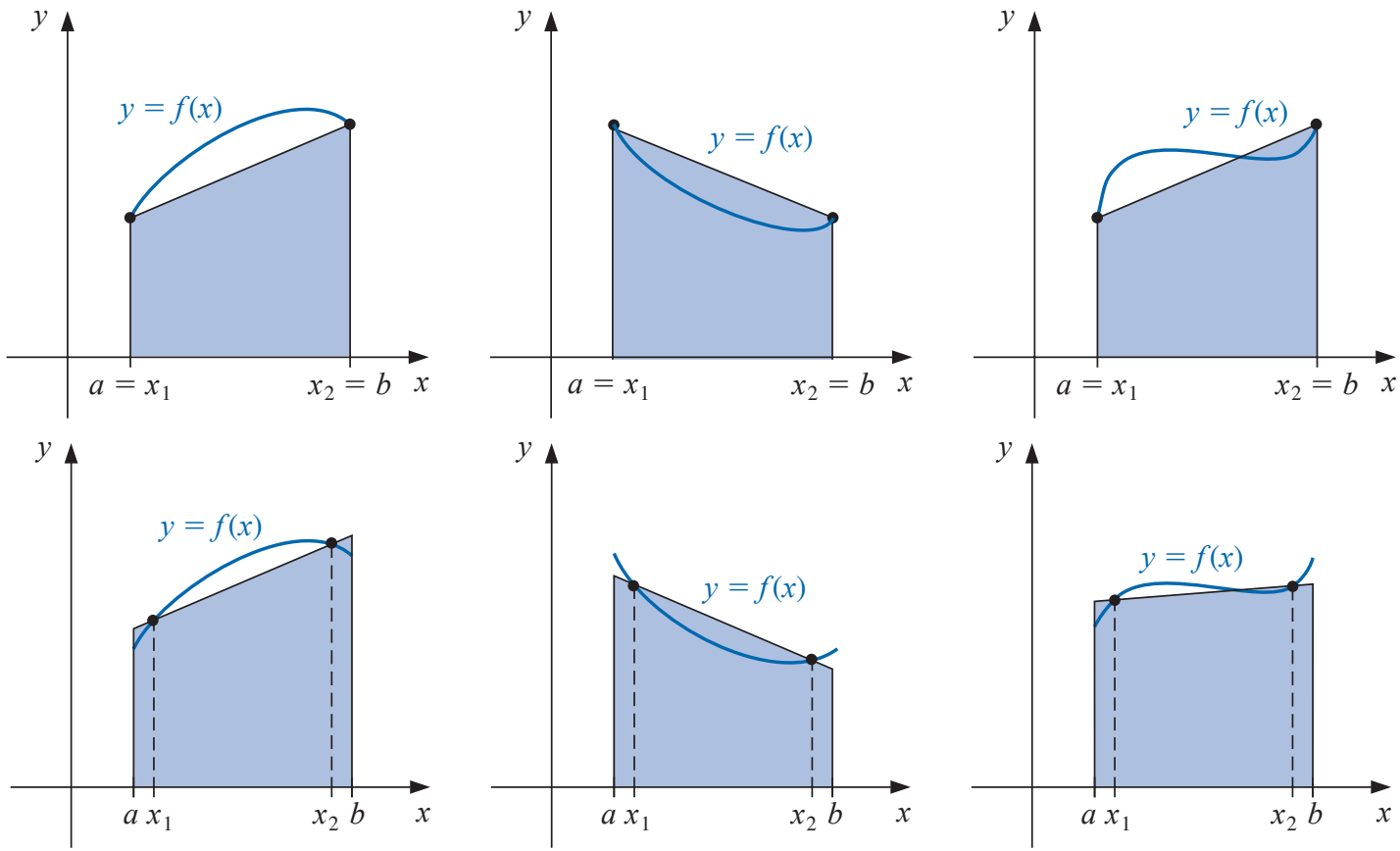
Let $x_0, x_1, \ldots, x_n$ (n even) be a uniform partition of $[a, b]$. Then apply Simpson's rule on $[x_0, x_2], [x_2, x_4], \ldots,$ a total of n such intervals. Then we obtain the **composite Simpson's rule**:

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a) + 2\sum_{j=1}^{(n/2)-1} f\left(x_{2j}\right) + 4\sum_{j=1}^{n/2} f\left(x_{2j-1}\right) + f(b)\right] - \frac{b-a}{180}h^4 f^{(4)}(\mu)$$



Gauss quadrature

Previously we chose points (nodes) with fixed gaps. What if we are allowed to to choose points $x_0, \ldots, x_n$ and evaluate f there?



Gauss quadrature

Gauss quadrature tries to determine $x_1, \ldots, x_n$ and $c_1, \ldots, c_n$ s.t.

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} c_i f\left(x_i\right)$$

Conceptually, since we have $2n$ parameters, i.e., c_i, x_i for $i = 1, \ldots, n$, we expect to get "=" if $f(x)$ is a polynomial of degree $\leq 2n - 1$.

Gauss quadrature

Let's first try the case with interval $[-1, 1]$ and two points $x_1, x_2 \in [-1, 1]$. Then we need to find x_1, x_2, c_1, c_2 such that

$$\int_{-1}^{1} f(x)dx \approx c_1 f(x_1) + c_2 f(x_2)$$

and "=" holds for all polynomials of degree ≤ 3.

Gauss quadrature

We first note

$$\int \left(a_0 + a_1 x + a_2 x^2 + a_3 x^3\right) \mathrm{d}x = a_0 \int 1 \, \mathrm{d}x + a_1 \int x \, \mathrm{d}x + a_2 \int x^2 \, \mathrm{d}x + a_3 \int x^3 \, \mathrm{d}x$$

Then we need x_1, x_2, c_1, c_2 s.t. $\int_{-1}^{1} f(x) \, \mathrm{d}x = c_1 f(x_1) + c_2 f(x_2)$ for $f(x) = 1, x, x^2$, and x^3:

$$\begin{aligned}
c_1 \cdot 1 + c_2 \cdot 1 &= \int_{-1}^{1} 1 \, \mathrm{d}x = 2, \\
c_1 \cdot x_1 + c_2 \cdot x_2 &= \int_{-1}^{1} x \, \mathrm{d}x = 0 \\
c_1 \cdot x_1^2 + c_2 \cdot x_2^2 &= \int_{-1}^{1} x^2 \, \mathrm{d}x = \frac{2}{3}, \\
c_1 \cdot x_1^3 + c_2 \cdot x_2^3 &= \int_{-1}^{1} x^3 \, \mathrm{d}x = 0
\end{aligned}$$

Gauss quadrature

Solve the system of four equations to obtain x_1, x_2, c_1, c_2:

$$c_1 = 1, \quad c_2 = 1, \quad x_1 = -\frac{\sqrt{3}}{3}, \quad \text{and} \quad x_2 = \frac{\sqrt{3}}{3}$$

So the approximation is

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx f\left(\frac{-\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

which is exact for all polynomials of degree ≤ 3.

This point and weight selection is called **Gauss quadrature**.

Legendre polynomials

To obtain Gauss quadrature for larger n, we need the **Legendre polynomials** $\{P_n : n = 0, 1, \dots\}$: which are determined to satisfy:

1. All P_n are monic (leading coefficient $=1$)
2. For each $n \geq 1$, there is

$$\int_{-1}^{1} P(x)P_n(x)\,\mathrm{d}x = 0$$

   for all polynomial P of degree less than n.

Thus the Legendre polynomials P_n are like an orthogonal basis of polynomials (orthogonal in the sense that $\int_{-1}^{1} P_n(x)P_m(x)\,\mathrm{d}x = 0$ for all $n \neq m$.

Legendre polynomials

The first five Legendre polynomials:

$$
\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= x^2 - \frac{1}{3} \\
P_3(x) &= x^3 - \frac{3}{5}x \\
P_4(x) &= x^4 - \frac{6}{7}x^2 + \frac{3}{35}
\end{aligned}
$$

In practice, there is a simple recursive formula to obtain P_n using P_{n-1} and P_{n-2} (with some scalings of the monic ones):

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

Gauss quadrature and Legendre polynomial

Theorem (Obtain Gauss quadrature by Legendre polynomials)

Suppose $x_1, \ldots, x_n$ are the roots of the nth Legendre polynomial $P_n(x)$, and define

$$c_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \, \mathrm{d}x$$

If $P(x)$ is any polynomial of degree less than $2n$, then

$$\int_{-1}^{1} P(x) dx = \sum_{i=1}^{n} c_i P(x_i)$$

Gauss quadrature and Legendre polynomial

Proof. First consider the the case $\deg(P) \leq n-1$. Given the roots $x_1, \ldots, x_n$ of $P_n(x)$, let $L_{n-1,i}(x)$ be the Lagrange polynomial for x_i. Then we know

$$P(x) = \sum_{i=1}^{n} P(x_i) L_i(x) = \sum_{i=1}^{n} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} P(x_i)$$

and therefore

$$\int_{-1}^{1} P(x)dx = \int_{-1}^{1} \left[\sum_{i=1}^{n} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} P(x_i) \right] dx$$

$$= \sum_{i=1}^{n} \left[\int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx \right] P(x_i) = \sum_{i=1}^{n} c_i P(x_i)$$

Proof (cont.) If $n \leq \deg(P) \leq 2n-1$, then we know

$$P(x) = Q(x)P_n(x) + R(x)$$

for some polynomials $Q(x), R(x)$ that are of degree at most $n-1$. Since $x_1, \ldots, x_n$ are roots of $P_n(x)$, we have

$$P(x_i) = Q(x_i)P_n(x_i) + R(x_i) = R(x_i)$$

for all $i = 1, \ldots, n$. Furthermore, we know

$$\int_{-1}^{1} P(x)dx = \int_{-1}^{1} \left[Q(x)P_n(x) + R(x)\right] dx = \int_{-1}^{1} R(x)dx = \sum_{i=1}^{n} c_i R(x_i) = \sum_{i=1}^{n} c_i P(x_i)$$

because $\int_{-1}^{1} Q(x)P_n(x)\,\mathrm{d}x = 0$ given that the degree of Q is at most $n-1$.

Gauss quadrature

n	**Roots** $r_{n,i}$	**Coefficients** $c_{n,i}$
2	0.5773502692	1.0000000000
	-0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	-0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	-0.3399810436	0.6521451549
	-0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	-0.5384693101	0.4786286705
	-0.9061798459	0.2369268850

Example

Example (Gauss quadrature)

Approximate $\int_{-1}^{1} e^x \cos x \, dx$ using Gauss quadrature with $n = 3$.

Solution. We need to use the roots of Legendre polynomial and coefficient values for $n = 3$:

n	Roots $r_{n,i}$	Coefficients $c_{n,i}$
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	-0.7745966692	0.5555555556

$$\begin{aligned}\int_{-1}^{1} e^x \cos x \, dx \approx & 0.\overline{5} e^{0.77459692} \cos(0.774596692) + 0.\overline{8} \cos(0) \\ & + 0.\overline{5} e^{-0.77459692} \cos(-0.774596692) \\ = & 1.9333904\end{aligned}$$

True value is $\int_{-1}^{1} e^x \cos x \, dx = 1.9334214$. Our error is 3.2×10^{-5}.

Gauss quadrature on arbitrary interval

So far the Gauss quadrature is only considered on $[-1, 1]$.

To find Gauss quadrature on arbitrary $x \in [a, b]$, just do a change of variable:

$$t = \frac{2x - a - b}{b - a} \quad \Longleftrightarrow \quad x = \frac{1}{2}[(b - a)t + a + b]$$

Then $t \in [-1, 1]$ and the integral is

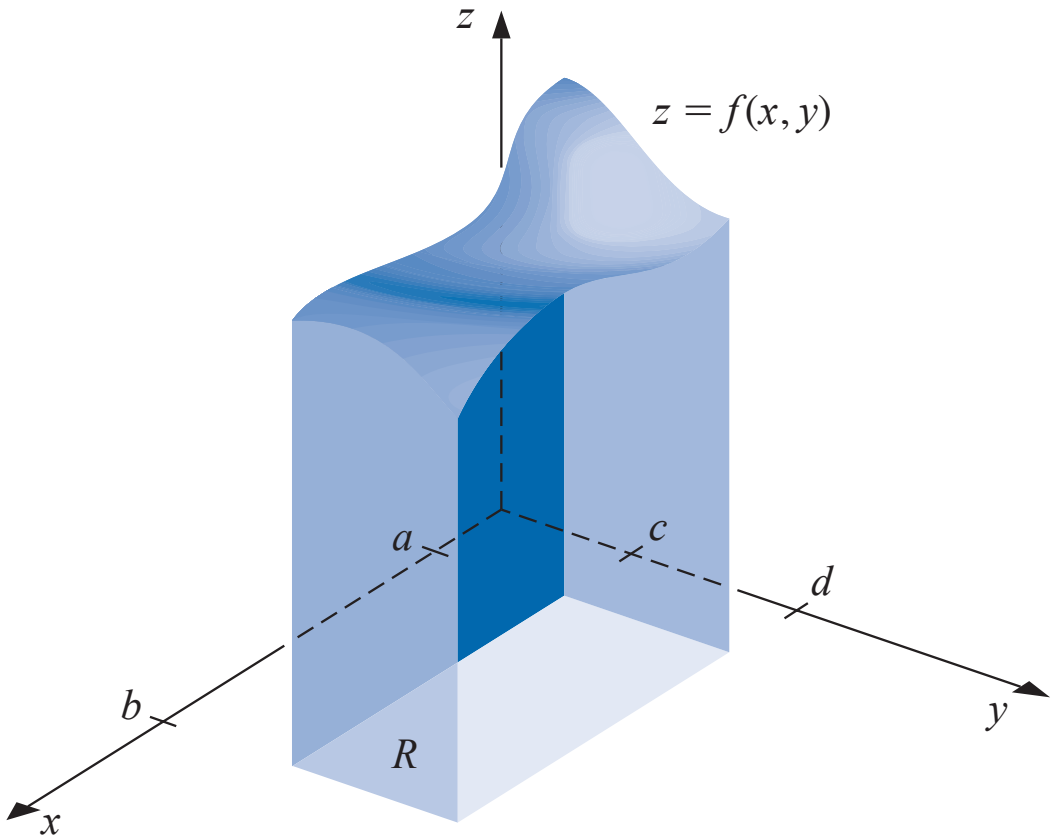
$$\int_a^b f(x)\,\mathrm{d}x = \int_{-1}^1 f\left(\frac{(b - a)t + (b + a)}{2}\right)\frac{(b - a)}{2}\,\mathrm{d}t$$

Then apply Gauss quadrature to the right side.

Multiple integrals

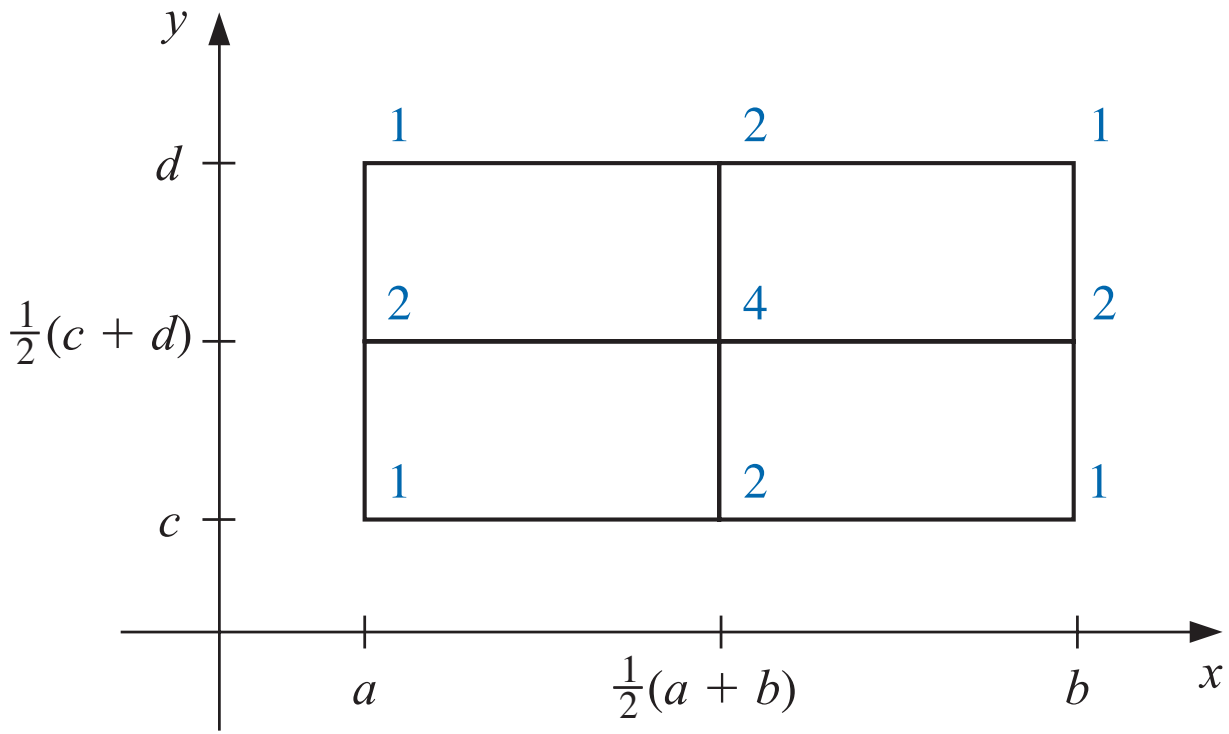
Now we consider multiple integral

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$



Multiple integrals

First consider a 2×2 grid on the domain $[a, b] \times [c, d]$:



Here $k = \frac{d-c}{2}$ and $h = \frac{b-a}{2}$.

Multiple integrals

We first approximate the inner integral using composite Trapezoidal rule:

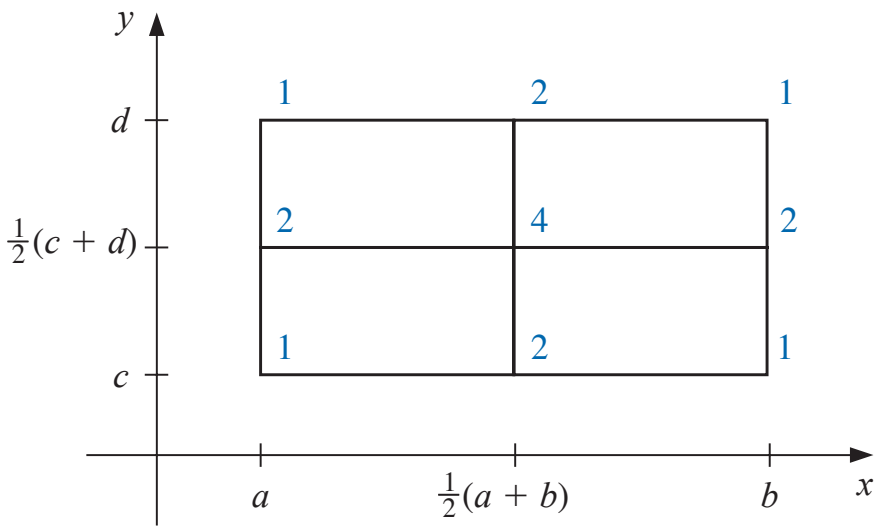
$$\begin{aligned}\int_c^d f(x,y)\,\mathrm{d}y &= \int_c^{c+k} f(x,y)\,\mathrm{d}y + \int_{c+k}^d f(x,y)\,\mathrm{d}y \\ &\approx \frac{k}{2}(f(x,c)+f(x,c+k)) + \frac{k}{2}(f(x,c+k)+f(x,d)) \\ &= \frac{k}{2}(f(x,c)+2f(x,c+k)+f(x,d)) =: g(x)\end{aligned}$$

Then approximate the outer integral:

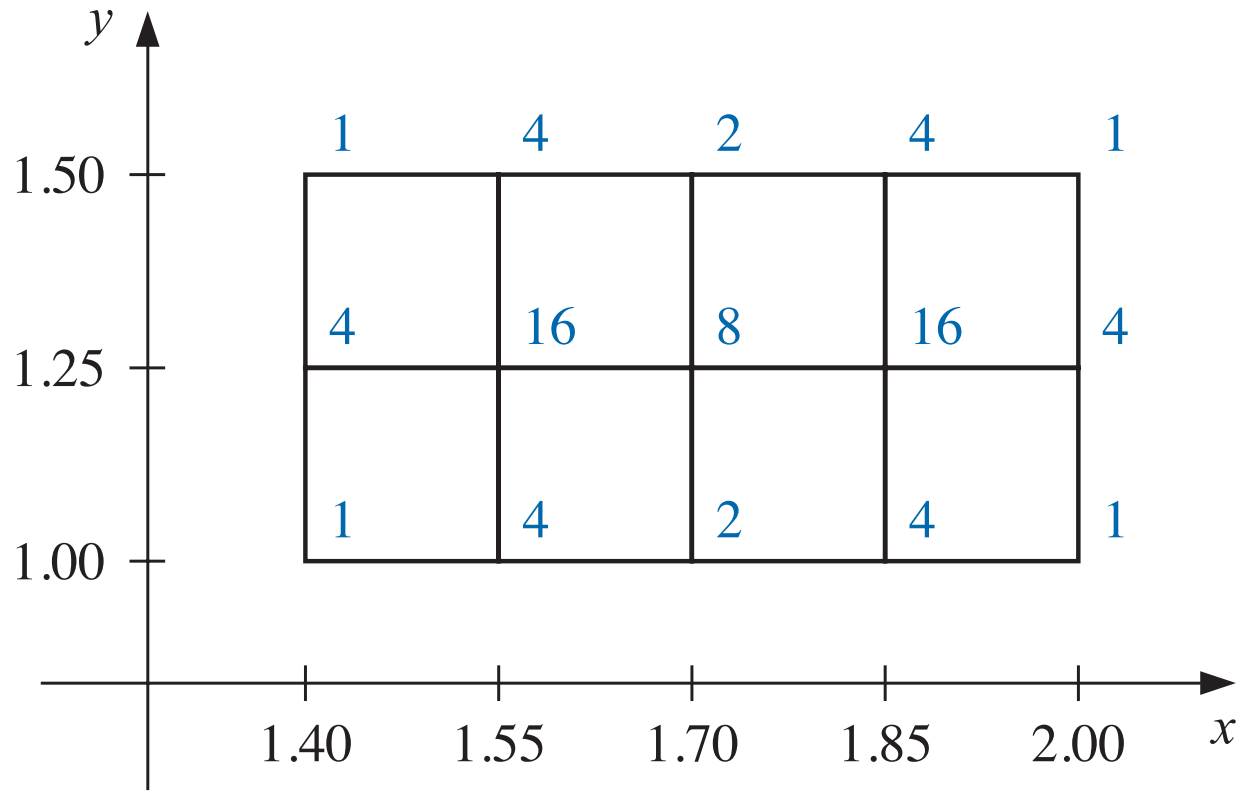
$$\int_a^b g(x)\,\mathrm{d}x \approx \frac{h}{2}(g(a)+2g(a+h)+g(b))$$

Combine the two to obtain:

$$\int_a^b \left(\int_c^d f(x,y)\,dy \right) dx \approx \frac{(b-a)(d-c)}{16} \left\{ f(a,c) + f(a,d) + f(b,c) + f(b,d) \right.$$
$$+ 2\left[f\left(\frac{a+b}{2}, c\right) + f\left(\frac{a+b}{2}, d\right) + f\left(a, \frac{c+d}{2}\right) \right.$$
$$\left. \left. + f\left(b, \frac{c+d}{2}\right) \right] + 4f\left(\frac{a+b}{2}, \frac{c+d}{2}\right) \right\}$$



We can also consider a 2×4 grid on the domain $[a, b] \times [c, d]$:



Here $k = \frac{d-c}{4}$ and $h = \frac{b-a}{2}$.

Composite Simpson's rule on non-rectangular region

Now we consider multiple integrals on non-rectangular regions:

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$

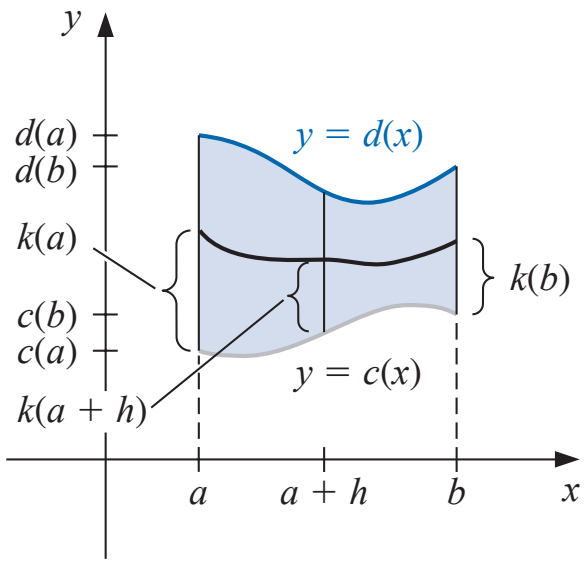
For each integral set $k(x) = \frac{d(x)-c(x)}{2}$, then

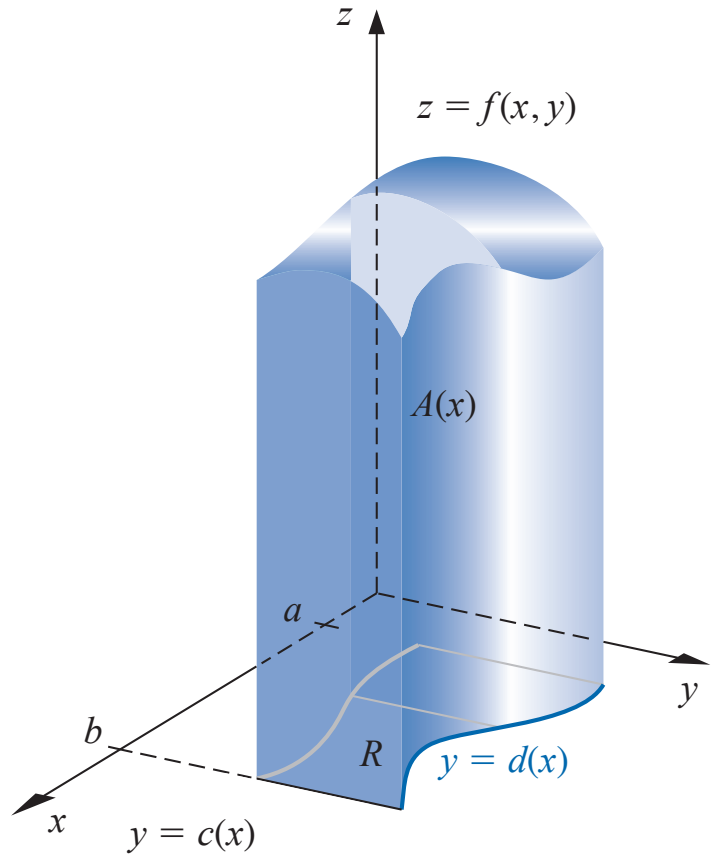
$$\begin{aligned}
\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x &\approx \int_a^b \frac{k(x)}{3}[f(x,c(x)) + 4f(x,c(x)+k(x)) + f(x,d(x))]\,\mathrm{d}x \\
&\approx \frac{h}{3}\left\{ \frac{k(a)}{3}[f(a,c(a)) + 4f(a,c(a)+k(a)) + f(a,d(a))] \right. \\
&\quad + \frac{4k(a+h)}{3}[f(a+h,c(a+h)) + 4f(a+h,c(a+h) \\
&\quad + k(a+h)) + f(a+h,d(a+h))] \\
&\quad \left. + \frac{k(b)}{3}\Big[f(b,c(b)) + 4f(b,c(b)+k(b)) + f(b,d(b))\Big] \right\}
\end{aligned}$$

Gauss quadrature for non-rectangular region

We can also use Gauss quadrature for non-rectangular region:

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$





Gauss quadrature for non-rectangular region

We can also use Gauss quadrature for non-rectangular region:

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x$$

For each $x \in [a,b]$, transform $[c(x), d(x)]$ into variable t in $[-1,1]$:

$$f(x,y) = f\left(x, \frac{(d(x)-c(x))t + d(x) + c(x)}{2}\right)$$

$$\mathrm{d}y = \frac{d(x)-c(x)}{2}\,\mathrm{d}t$$

Gauss quadrature for non-rectangular region

So the inner integral can be approximated by Gauss quadrature:

$$\begin{aligned}\int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y &= \frac{d(x)-c(x)}{2}\int_{-1}^{1} f\left(x, \frac{(d(x)-c(x))t + d(x) + c(x)}{2}\right)\mathrm{d}t \\ &\approx \frac{d(x)-c(x)}{2}\sum_{j=1}^{n} c_{n,j} f\left(x, \frac{(d(x)-c(x))r_{n,j} + d(x) + c(x)}{2}\right) \\ &=: g(x)\end{aligned}$$

Then we apply Gauss quadrature to the outer integral:

$$\begin{aligned}\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}y\,\mathrm{d}x &\approx \int_a^b g(x)\,\mathrm{d}x \\ &= \int_{-1}^{1} g\left(\frac{(b-a)t + (b+a)}{2}\right)\frac{(b-a)}{2}\,\mathrm{d}t \\ &\approx \sum_{i=1}^{m} c_{m,i}\, g\left(\frac{(b-a)r_{m,i} + (b+a)}{2}\right)\frac{(b-a)}{2}\end{aligned}$$

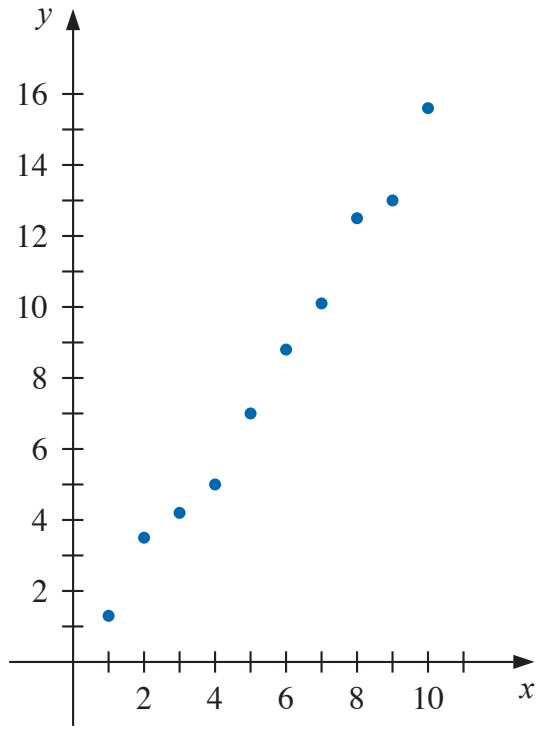
Section 5

Approximation Theory

Least squares approximation

Given N data points $\{(x_i, y_i)\}$ for $i = 1, \ldots, N$, can we determine a linear model $y = a_1 x + a_0$ (i.e., find a_0, a_1) that fits the data?

x_i	y_i	x_i	y_i
1	1.3	6	8.8
2	3.5	7	10.1
3	4.2	8	12.5
4	5.0	9	13.0
5	7.0	10	15.6



Matrix formulation

We can simplify notations by using matrices and vectors:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \quad X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \in \mathbb{R}^{N \times 2}$$

So we want to find $a = (a_0, a_1)^\top \in \mathbb{R}^2$ such that $y \approx Xa$.

Several types of fitting criteria

There are several types of criteria for "best fitting":

- Define the error function as

$$E_\infty(a) = \|y - Xa\|_\infty$$

and find $a^* \in \arg\min_a E_\infty(a)$. This is also called the **minimax** problem since the problem $\min_a E_\infty(a)$ can be written as

$$\min_a \max_{1 \le i \le n} |y_i - (a_0 + a_1 x_i)|$$

- Define the error function as

$$E_1(a) = \|y - Xa\|_1$$

and find $a^* \in \arg\min_a E_1(a)$. E_1 is also called the **absolute deviation**.

Least squares fitting

In this course, we focus on the widely used **least squares**.

Define the least squares error function as

$$E_2(a) = \|y - Xa\|_2^2 = \sum_{i=1}^{n} |y_i - (a_0 + a_1 x_i)|^2$$

and the least squares solution a^* is

$$a^* = \arg\min_a E_2(a)$$

Least squares fitting

To find the optimal parameter a, we need to solve

$$\nabla E_2(a) = 2X^\top(Xa - y) = 0$$

This is equivalent to the so-called **normal equation**:

$$X^\top Xa = X^\top y$$

Note that $X^\top X \in \mathbb{R}^{2\times 2}$ and $X^\top y \in \mathbb{R}^2$, so the normal equation is easy to solve!

Least squares fitting

It is easy to show that

$$X^\top X = \begin{bmatrix} N & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{bmatrix}, \quad X^\top y = \begin{bmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N x_i y_i \end{bmatrix}$$

Using the close-form of inverse of 2-by-2 matrix, we have

$$(X^\top X)^{-1} = \frac{1}{N\sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \begin{bmatrix} \sum_{i=1}^N x_i^2 & -\sum_{i=1}^N x_i \\ -\sum_{i=1}^N x_i & N \end{bmatrix}$$

Least squares fitting

Therefore we have the solution

$$a^* = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = (X^\top X)^{-1}(X^\top y)$$

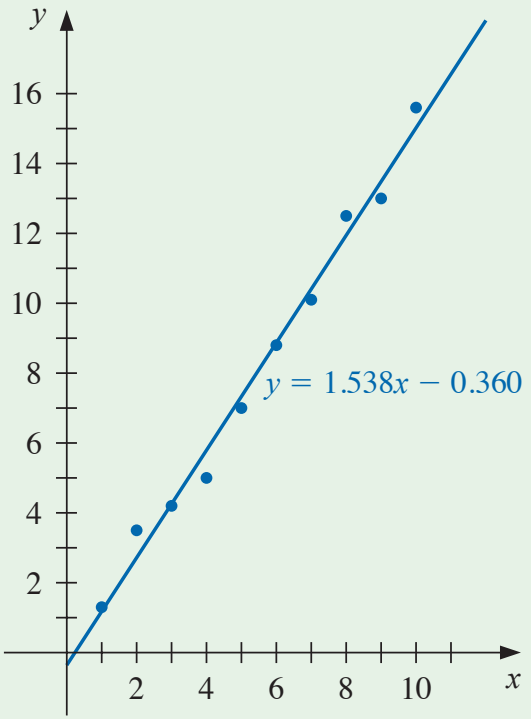
$$= \begin{bmatrix} \frac{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i - \sum_{i=1}^N x_i y_i \sum_{i=1}^N x_i}{N\sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \\ \frac{N\sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N\sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \end{bmatrix}$$

Least squares fitting

Example

Least squares fitting of the data gives $a_0 = -0.36$ and $a_1 = 1.538$.

x_i	y_i	x_i	y_i
1	1.3	6	8.8
2	3.5	7	10.1
3	4.2	8	12.5
4	5.0	9	13.0
5	7.0	10	15.6



Polynomial least squares

The least squares fitting presented above is also called **linear least squares** due to the linear model $y = a_0 + a_1 x$.

For general least squares fitting problems with data $\{(x_i, y_i) : i = 1, \ldots, N\}$, we may use polynomial

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

as the fitting model. Note that $n = 1$ reduces to linear model.

Now the **polynomial least squares** error is defined by

$$E(a) = \sum_{i=1}^{N} |y_i - P_n(x_i)|^2$$

where $a = (a_0, a_1, \ldots, a_n)^\top \in \mathbb{R}^{n+1}$.

Matrices in polynomial least squares fitting

Like before, we use matrices and vectors:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^n \end{bmatrix} \in \mathbb{R}^{N\times(n+1)}$$

So we want to find $a = (a_0, a_1, \ldots, a_n)^\top \in \mathbb{R}^{n+1}$ such that $y \approx Xa$.

Polynomial least squares fitting

Same as above, we need to find a such that

$$\nabla E_2(a) = 2X^\top(Xa - y) = 0$$

which has **normal equation**:

$$X^\top Xa = X^\top y$$

Note that now $X^\top X \in \mathbb{R}^{(n+1)\times(n+1)}$ and $X^\top y \in \mathbb{R}^{n+1}$. From normal equation we can solve for the fitting parameter

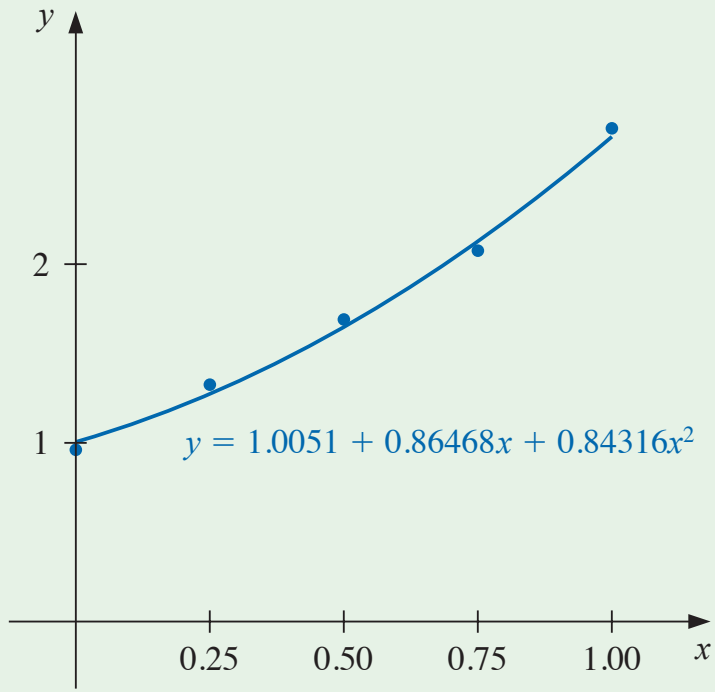
$$a^* = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = (X^\top X)^{-1}(X^\top y)$$

Polynomial least squares

Example

Least squares fitting of the data using $n = 2$ gives
$a_0 = 1.0051, a_1 = 0.86468, a_2 = 0.84316$.

i	x_i	y_i
1	0	1.0000
2	0.25	1.2840
3	0.50	1.6487
4	0.75	2.1170
5	1.00	2.7183



Other least squares fitting models

In some situations, one may design model as

$$y = be^{ax}$$
$$y = bx^{a}$$

as well as many others.

To use least squares fitting, we note that they are equivalent to, respectively,

$$\log y = \log b + ax$$
$$\log y = \log b + a \log x$$

Therefore, we can first convert (x_i, y_i) to $(x_i, \log y_i)$ and $(\log x_i, \log y_i)$, and then apply standard linear least squares fitting.

Approximating functions

We now consider fitting (approximation) of a given function

$$f(x) \in C[a, b]$$

Suppose we use a polynomial $P_n(x)$ of degree n to fit $f(x)$, where

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

with fitting parameters $a = (a_0, a_1, \ldots, a_n)^\top \in \mathbb{R}^{n+1}$. Then the least squares error is

$$E(a) = \int_a^b |f(x) - P_n(x)|^2 \, \mathrm{d}x = \int_a^b \left| f(x) - \sum_{k=0}^{n} a_k x^k \right|^2 \mathrm{d}x$$

Approximating functions

The fitting parameter a needs to be solved from $\nabla E(a) = 0$.

To this end, we first rewrite $E(a)$ as

$$E(a) = \int_a^b (f(x))^2 \, \mathrm{d}x - 2 \sum_{k=0}^{n} a_k \int_a^b x^k f(x) \, \mathrm{d}x + \int_a^b \Big(\sum_{k=0}^{n} a_k x^k \Big)^2 \, \mathrm{d}x$$

Therefore $\nabla E(a) = (\frac{\partial E}{\partial a_0}, \frac{\partial E}{\partial a_1}, \ldots, \frac{\partial E}{\partial a_n})^\top \in \mathbb{R}^{n+1}$ where

$$\frac{\partial E}{\partial a_j} = -2 \int_a^b x^j f(x) \, \mathrm{d}x + 2 \sum_{k=0}^{n} a_k \int_a^b x^{j+k} \, \mathrm{d}x$$

for $j = 0, 1, \ldots, n$.

Approximating functions

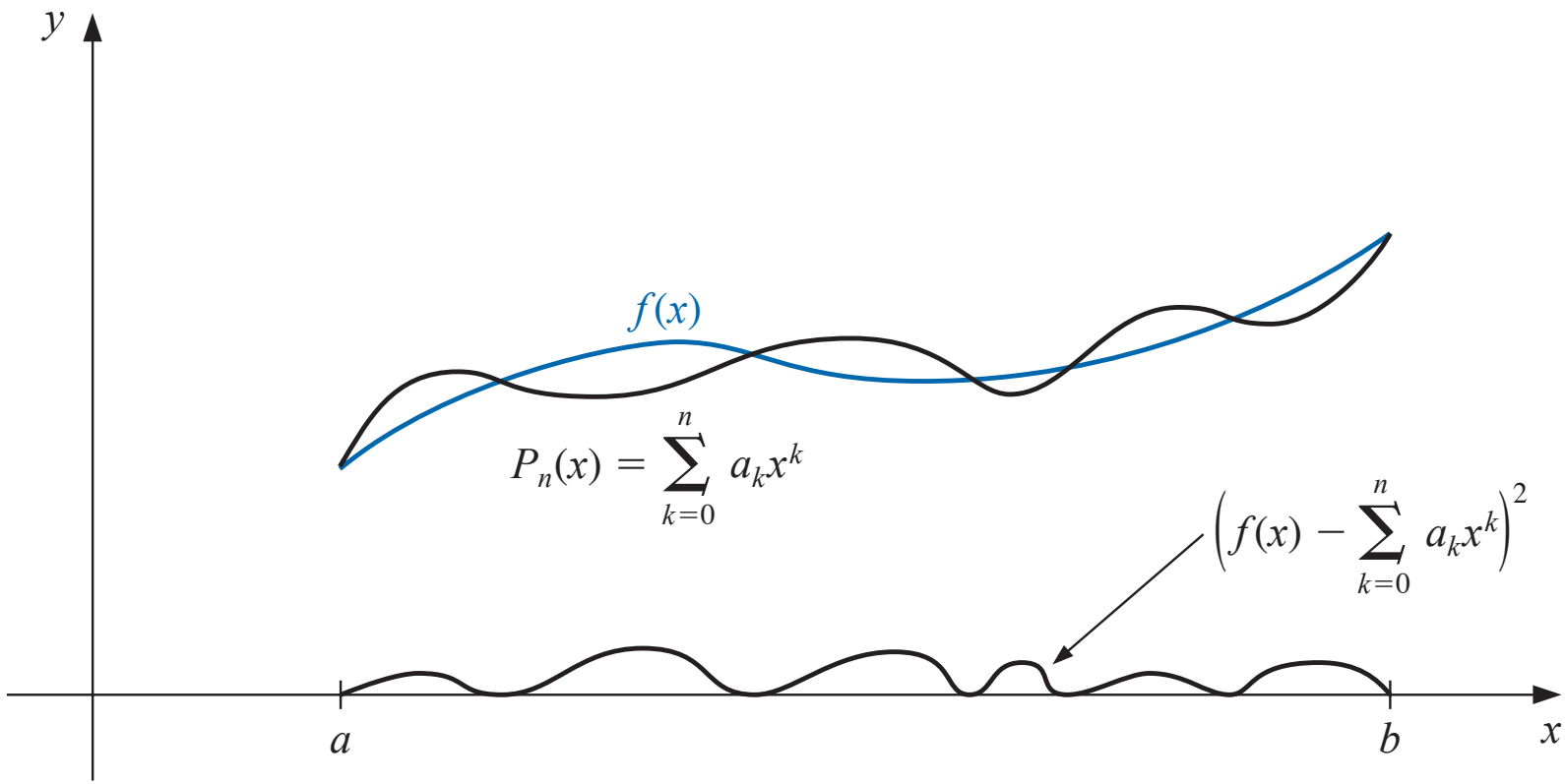
By setting $\frac{\partial E}{\partial a_j} = 0$ for all j, we obtain the **normal equation**

$$\sum_{k=0}^{n} \left(\int_a^b x^{j+k} \, \mathrm{d}x \right) a_k = \int_a^b x^j f(x) \, \mathrm{d}x$$

for $j = 0, \ldots, n$. This is a linear system of $n+1$ equations, from which we can solve for $a^* = (a_0, \ldots, a_n)^\top$.

Approximating functions

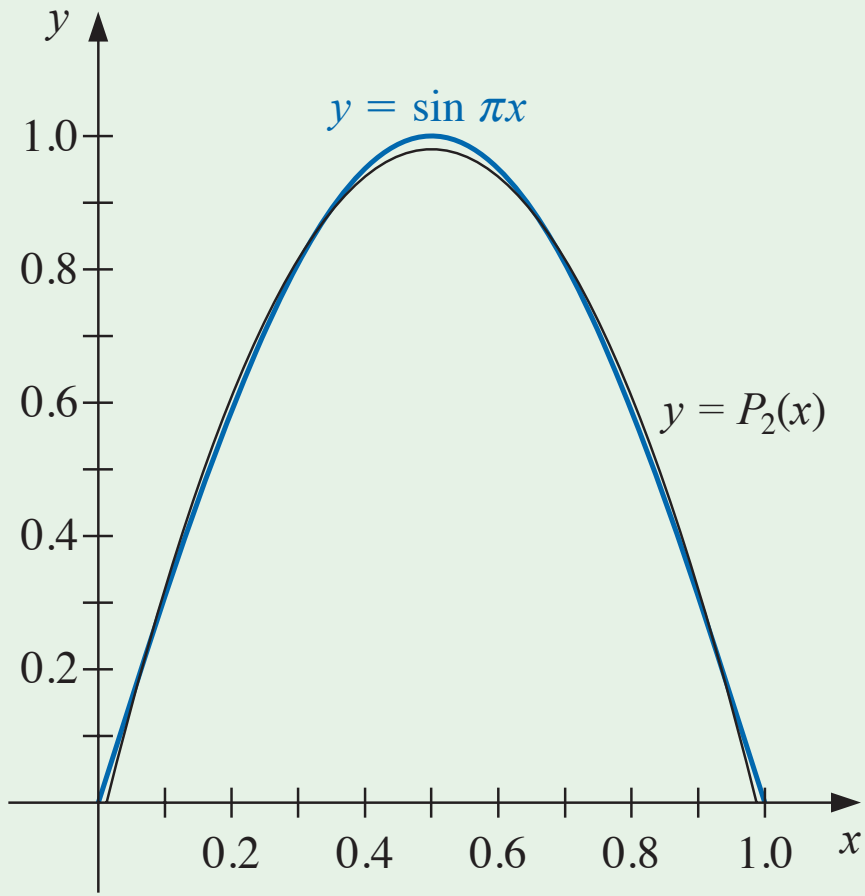
For the given function $f(x) \in C[a, b]$, we obtain least squares approximating polynomial $P_n(x)$:



Approximating functions

Example

Use least squares approximating polynomial of degree 2 for the function $f(x) = \sin(\pi x)$ on the interval $[0, 1]$.



Least squares approximations with polynomials

Remark

- *The matrix in the normal equation is called* **Hilbert matrix**, *with entries of form*

$$\int_a^b x^{j+k}\,\mathrm{d}x = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1}$$

  which is prune to round-off errors.
- *The parameters* $a = (a_0, \ldots, a_n)^\top$ *we obtained for polynomial* $P_n(x)$ *cannot be used for* $P_{n+1}(x)$ *– we need to start the computations from beginning.*

Linearly independent functions

Definition

The set of functions $\{\phi_1, \ldots, \phi_n\}$ is called **linearly independent** on $[a, b]$ if

$$c_1\phi_1(x) + c_2\phi_2(x) + \cdots + c_n\phi_n(x) = 0, \quad \text{for all } x \in [a, b]$$

implies that $c_1 = c_2 = \cdots = c_n = 0$.

Otherwise the set of functions is called **linearly dependent**.

Linearly independent functions

Example

Suppose $\phi_j(x)$ is a polynomial of degree j for $j = 0, 1, \ldots, n$, then $\{\phi_0, \ldots, \phi_n\}$ is linearly independent on any interval $[a, b]$.

Proof.

Suppose there exist $c_0, \ldots, c_n$ such that

$$c_0\phi_0(x) + \cdots + c_n\phi_n(x) = 0$$

for all $x \in [a, b]$. If $c_n \neq 0$, then this is a polynomial of degree n and can have at most n roots, contradiction. Hence $c_n = 0$. Repeat this to show that $c_0 = \cdots = c_n = 0$. □

Linearly independent functions

Example

Suppose $\phi_0(x) = 2, \phi_1(x) = x - 3, \phi_2(x) = x^2 + 2x + 7$, and $Q(x) = a_0 + a_1 x + a_2 x^2$. Show that there exist constants c_0, c_1, c_2 such that $Q(x) = c_0\phi_0(x) + c_1\phi_1(x) + c_2\phi_2(x)$.

Solution. Substitute ϕ_j into $Q(x)$, and solve for c_0, c_1, c_2.

Linearly independent functions

We denote $\Pi_n = \{a_0 + a_1 x + \cdots + a_n x^n \mid a_0, a_1, \ldots, a_n \in \mathbb{R}\}$, i.e., Π_n is the set of polynomials of degree $\leq n$.

Theorem

Suppose $\{\phi_0, \ldots, \phi_n\}$ is a collection of linearly independent polynomials in Π_n, then any polynomial in Π_n can be written uniquely as a linear combination of $\phi_0(x), \ldots, \phi_n(x)$.

$\{\phi_0, \ldots, \phi_n\}$ is called a **basis** of Π_n.

Orthogonal functions

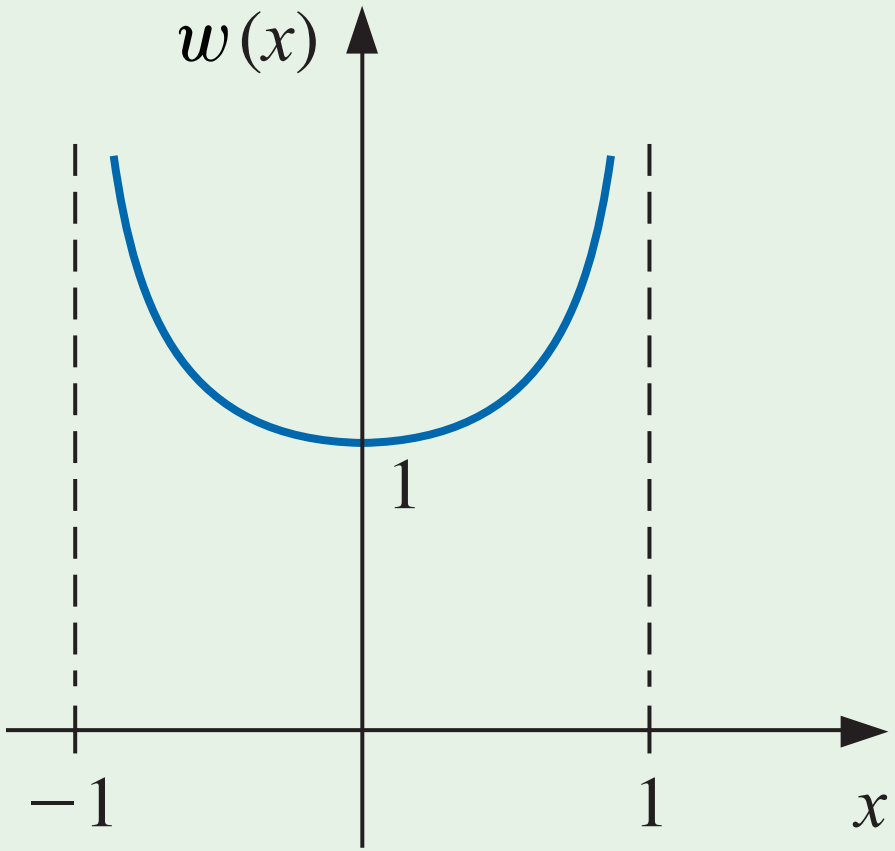
Definition

An integrable function w is called a **weight function** on the interval I if $w(x) \geq 0$, for all $x \in I$, but $w(x) \not\equiv 0$ on any subinterval of I.

Orthogonal functions

Example

Define a weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$ on interval $(-1, 1)$.



Orthogonal functions

Suppose $\{\phi_0, \ldots, \phi_n\}$ is a set of linearly independent functions in $C[a, b]$ and w is a weight function on $[a, b]$. Given $f(x) \in C[a, b]$, we seek a linear combination

$$\sum_{k=0}^{n} a_k \phi_k(x)$$

to minimize the least squares error:

$$E(a) = \int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k \phi_k(x) \right]^2 \mathrm{d}x$$

where $a = (a_0, \ldots, a_n)$.

Orthogonal functions

As before, we need to solve a^* from $\nabla E(a) = 0$:

$$\frac{\partial E}{\partial a_j} = \int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k \phi_k(x) \right] \phi_j(x) \, \mathrm{d}x = 0$$

for all $j = 0, \ldots, n$. Then we obtain the normal equation

$$\sum_{k=0}^{n} \left(\int_a^b w(x) \phi_k(x) \phi_j(x) \, \mathrm{d}x \right) a_k = \int_a^b w(x) f(x) \phi_j(x) \, \mathrm{d}x$$

which is a linear system of $n+1$ equations about $a = (a_0, \ldots, a_n)^\top$.

Orthogonal functions

If we chose the basis $\{\phi_0, \ldots, \phi_n\}$ such that

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,\mathrm{d}x = \begin{cases} 0, & \text{when } j \neq k \\ \alpha_j, & \text{when } j = k \end{cases}$$

for some $\alpha_j > 0$, then the LHS of the normal equation simplifies to $\alpha_j a_j$. Hence we obtain closed form solution a_j:

$$a_j = \frac{1}{\alpha_j} \int_a^b w(x) f(x) \phi_j(x)\,\mathrm{d}x$$

for $j = 0, \ldots, n$.

Orthogonal functions

Definition

A set $\{\phi_0, \ldots, \phi_n\}$ is called **orthogonal** on the interval $[a, b]$ with respect to weight function w if

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,\mathrm{d}x = \begin{cases} 0, & \text{when } j \neq k \\ \alpha_j, & \text{when } j = k \end{cases}$$

for some $\alpha_j > 0$ for all $j = 0, \ldots, n$.

If in addition $\alpha_j = 1$ for all $j = 0, \ldots, n$, then the set is called **orthonormal** with respect to w.

The definition above applies to general functions, but for now we focus on orthogonal/orthonormal polynomials only.

Gram-Schmidt process

Theorem

A set of orthogonal polynomials $\{\phi_0, \ldots, \phi_n\}$ on $[a, b]$ with respect to weight function w can be constructed in the recursive way

- *First define*

$$\phi_0(x) = 1, \quad \phi_1(x) = x - \frac{\int_a^b xw(x)\,\mathrm{d}x}{\int_a^b w(x)\,\mathrm{d}x}$$

- *Then for every $k \geq 2$, define*

$$\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x)$$

where

$$B_k = \frac{\int_a^b xw(x)[\phi_{k-1}(x)]^2\,\mathrm{d}x}{\int_a^b w(x)[\phi_{k-1}(x)]^2\,\mathrm{d}x}, \quad C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x)\,\mathrm{d}x}{\int_a^b w(x)[\phi_{k-2}(x)]^2\,\mathrm{d}x}$$

Orthogonal polynomials

Corollary

Let $\{\phi_0, \ldots, \phi_n\}$ be constructed by the Gram-Schmidt process in the theorem above, then for any polynomial $Q_k(x)$ of degree $k < n$, there is

$$\int_a^b w(x)\phi_n(x)Q_k(x)\,\mathrm{d}x = 0$$

Proof.

$Q_k(x)$ can be written as a linear combination of $\phi_0(x), \ldots, \phi_k(x)$, which are all orthogonal to ϕ_n with respect to w. □

Legendre polynomials

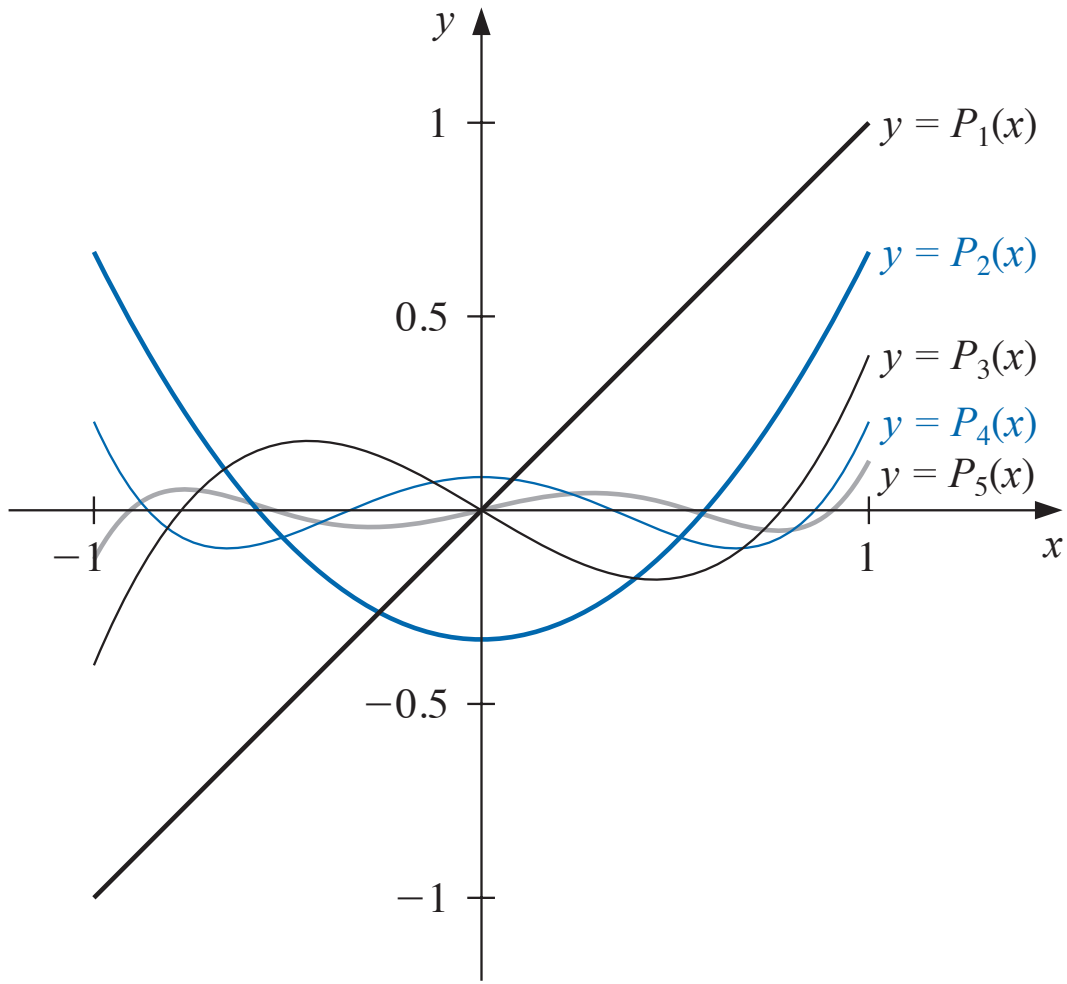
Using weight function $w(x) \equiv 1$ on $[-1, 1]$, we can construct **Legendre polynomials** using the recursive process above to get

$$
\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= x^2 - \frac{1}{3} \\
P_3(x) &= x^3 - \frac{3}{5}x \\
P_4(x) &= x^4 - \frac{6}{7}x^2 + \frac{3}{35} \\
P_5(x) &= x^5 - \frac{10}{9}x^3 + \frac{5}{21}x \\
&\vdots
\end{aligned}
$$

Use the Gram-Schmidt process to construct them by yourself.

Legendre polynomials

The first few Legendre polynomials:



Chebyshev polynomials

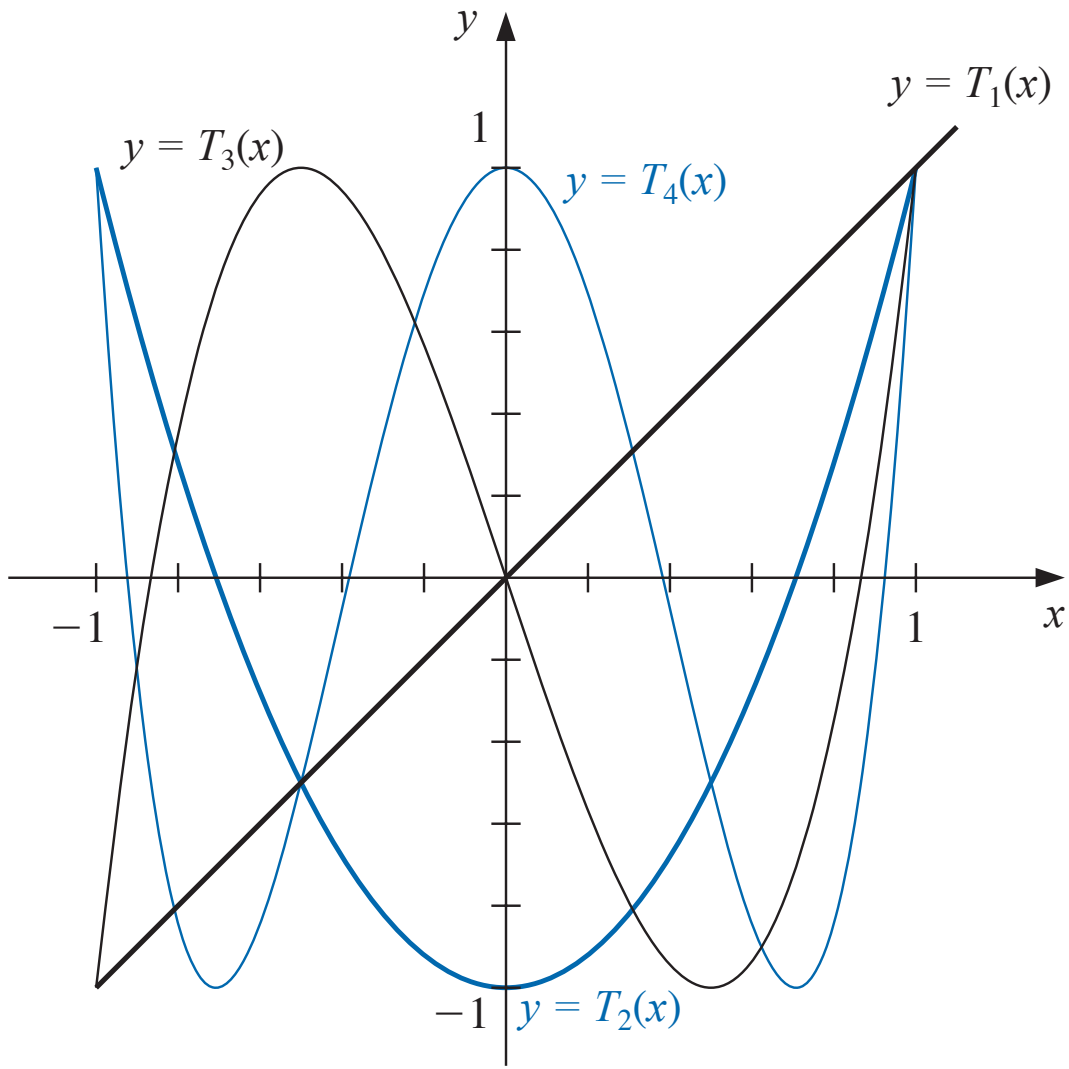
Using weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$ on $(-1, 1)$, we can construct **Chebyshev polynomials** using the recursive process above to get

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
&\vdots
\end{aligned}
$$

It can be shown that $T_n(x) = \cos(n \arccos x)$ for $n = 0, 1, \ldots$

Chebyshev polynomials

The first few Chebyshev polynomials:



Chebyshev polynomials

The Chebyshev polynomials $T_n(x)$ of degree $n \geq 1$ has n simple zeros in $[-1, 1]$ (from right to left) at

$$\bar{x}_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \text{for each } k = 1, 2, \ldots, n$$

Moreover, T_n has maximum/minimum (from right to left) at

$$\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right) \text{ where } T_n(\bar{x}'_k) = (-1)^k \text{ for each } k = 0, 1, 2, \ldots, n$$

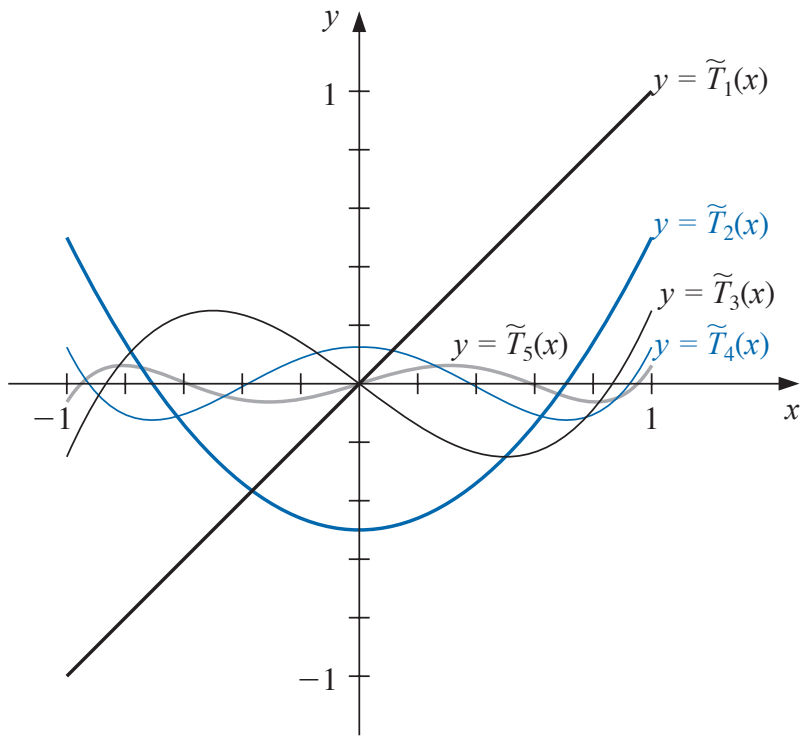
Therefore $T_n(x)$ has n distinct roots and $n+1$ extreme points on $[-1, 1]$. These $2n+1$ points, from right to left, are max, zero, min, zero, max ...

Monic Chebyshev polynomials

The monic Chebyshev polynomials $\tilde{T}_n(x)$ are given by $\tilde{T}_0 = 1$ and

$$\tilde{T}_n = \frac{1}{2^{n-1}} T_n(x)$$

for $n \geq 1$.



Monic Chebyshev polynomials

The monic Chebyshev polynomials are

$$\begin{aligned}
\tilde{T}_0(x) &= 1 \\
\tilde{T}_1(x) &= x \\
\tilde{T}_2(x) &= x^2 - \frac{1}{2} \\
\tilde{T}_3(x) &= x^3 - \frac{3}{4}x \\
\tilde{T}_4(x) &= x^4 - x^2 + \frac{1}{8} \\
&\vdots
\end{aligned}$$

Monic Chebyshev polynomials

The monic Chebyshev polynomials $\tilde{T}_n(x)$ of degree $n \geq 1$ has n simple zeros in $[-1, 1]$ at

$$\bar{x}_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \text{for each } k = 1, 2, \ldots, n$$

Moreover, T_n has maximum/minimum at

$$\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right) \text{ where } T_n(\bar{x}'_k) = \frac{(-1)^k}{2^{n-1}}, \text{ for each } k = 0, 1, \ldots, n$$

Therefore $\tilde{T}_n(x)$ also has n distinct roots and $n+1$ extreme points on $[-1, 1]$.

Monic Chebyshev polynomials

Denote $\tilde{\Pi}_n$ be the set of monic polynomials of degree n.

Theorem

For any $P_n \in \tilde{\Pi}_n$, there is

$$\frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\tilde{T}_n(x)| \leq \max_{x\in[-1,1]} |P_n(x)|$$

The "$=$" holds only if $P_n \equiv \tilde{T}_n$.

Monic Chebyshev polynomials

Proof.

Assume not, then $\exists\, P_n(x) \in \tilde{\Pi}_n$, s.t. $\max_{x\in[-1,1]} |P_n(x)| < \frac{1}{2^{n-1}}$.

Let $Q(x) := \tilde{T}_n(x) - P_n(x)$. Since $\tilde{T}_n, P_n \in \tilde{\Pi}_n$, we know $Q(x)$ is a ploynomial of degree at most $n-1$. At the $n+1$ extreme points $\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right)$ for $k = 0, 1, \ldots, n$, there are

$$Q(\bar{x}'_k) = \tilde{T}_n(\bar{x}'_k) - P_n(\bar{x}'_k) = \frac{(-1)^k}{2^{n-1}} - P_n(\bar{x}'_k)$$

Hence $Q(\bar{x}'_k) > 0$ when k is even and < 0 when k odd. By intermediate value theorem, Q has at least n distinct roots, contradiction to $\deg(Q) \leq n-1$. □

Minimizing Lagrange interpolation error

Let $x_0, \ldots, x_n$ be $n+1$ distinct points on $[-1,1]$ and $f(x) \in C^{n+1}[-1,1]$, recall that the Lagrange interpolating polynomial $P(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$ satisfies

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

for some $\xi(x) \in (-1,1)$ at every $x \in [-1,1]$.

We can control the size of $(x-x_0)(x-x_1)\cdots(x-x_n)$ since it belongs to $\tilde{\Pi}_{n+1}$: set $(x-x_0)(x-x_1)\cdots(x-x_n) = \tilde{T}_{n+1}(x)$. That is, set $x_k = \cos\left(\frac{2k-1}{2n}\pi\right)$, the kth root of $\tilde{T}_{n+1}(x)$ for $k = 1, \ldots, n+1$. This results in the minimal $\max_{x\in[-1,1]} |(x-x_0)(x-x_1)\cdots(x-x_n)| = \frac{1}{2^n}$.

Minimizing Lagrange interpolation error

Corollary

Let $P(x)$ be the Lagrange interpolating polynomial with $n+1$ points chosen as the roots of $\tilde{T}_{n+1}(x)$, there is

$$\max_{x\in[-1,1]} |f(x) - P(x)| \leq \frac{1}{2^n(n+1)!} \max_{x\in[-1,1]} |f^{(n+1)}(x)|$$

Minimizing Lagrange interpolation error

If the interval of apporximation is on $[a, b]$ instead of $[-1, 1]$, we can apply change of variable

$$\tilde{x} = \frac{1}{2}[(b-a)x + (a+b)]$$

Hence, we can convert the roots $\bar{x}_k$ on $[-1, 1]$ to $\tilde{x}_k$ on $[a, b]$,

Minimizing Lagrange interpolation error

Example

Let $f(x) = xe^x$ on $[0, 1.5]$. Find the Lagrange interpolating polynomial using

1. the 4 equally spaced points $0, 0.5, 1, 1.5$.
2. the 4 points transformed from roots of $\tilde{T}_4$.

Minimizing Lagrange interpolation error

Solution. For each of the four points $x_0 = 0, x_1 = 0.5, x_2 = 1, x_3 = 1.5$, we obtain $L_i(x) = \frac{\prod_{j\neq i}(x-x_j)}{\prod_{j\neq i}(x_i-x_j)}$ for $i = 0, 1, 2, 3$:

$$
\begin{aligned}
L_0(x) &= -1.3333x^3 + 4.0000x^2 - 3.6667x + 1,\\
L_1(x) &= 4.0000x^3 - 10.000x^2 + 6.0000x,\\
L_2(x) &= -4.0000x^3 + 8.0000x^2 - 3.0000x,\\
L_3(x) &= 1.3333x^3 - 2.000x^2 + 0.66667x
\end{aligned}
$$

so the Lagrange interpolating polynomial is

$$
P_3(x) = \sum_{i=0}^{3} f(x_i)L_i(x) = 1.3875x^3 + 0.057570x^2 + 1.2730x.
$$

Minimizing Lagrange interpolation error

Solution. (cont.) The four roots of $\tilde{T}_4(x)$ on $[-1,1]$ are $\bar{x}_k = \cos(\frac{2k-1}{8}\pi)$ for $k = 1,2,3,4$. Shifting the points using $\tilde{x} = \frac{1}{2}(1.5x + 1.5)$, we obtain four points

$$\tilde{x}_0 = 1.44291, \tilde{x}_1 = 1.03701, \tilde{x}_2 = 0.46299, \tilde{x}_3 = 0.05709$$

with the same procedure as above to get $\tilde{L}_0, \ldots, \tilde{L}_3$ using these 4 points, and then the Lagrange interpolating polynomial:

$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352.$$

Minimizing Lagrange interpolation error

Now compare the approximation accuracy of the two polynomials

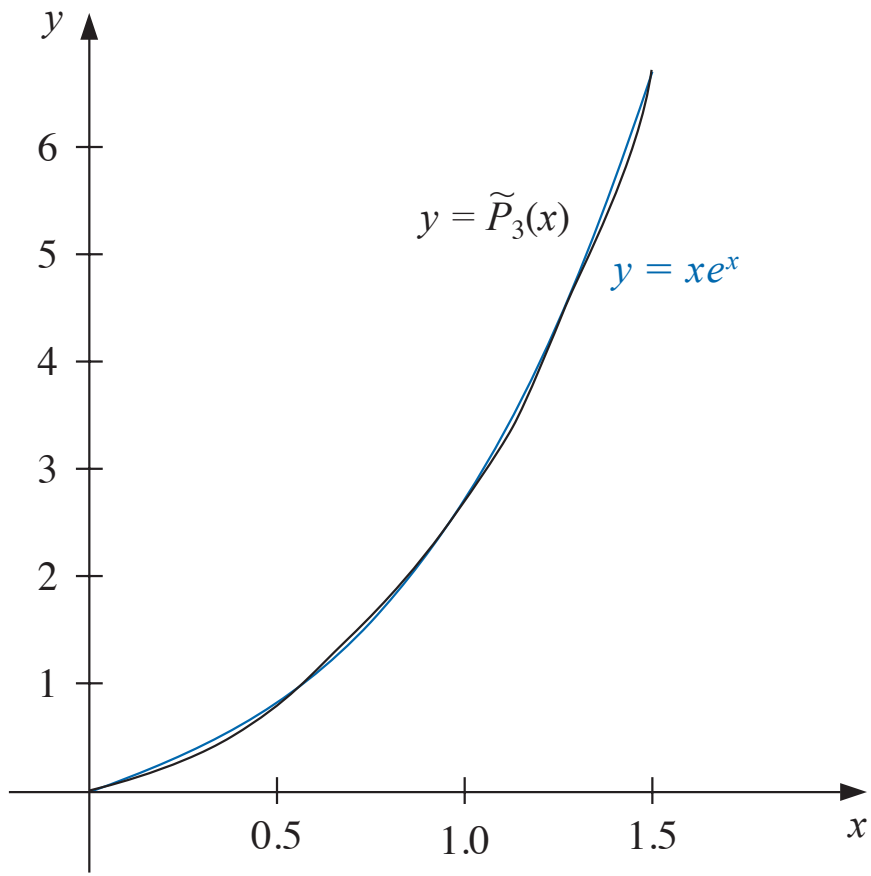
$$P_3(x) = 1.3875x^3 + 0.057570x^2 + 1.2730x$$
$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352$$

x	$f(x) = xe^x$	$P_3(x)$	$\lvert xe^x - P_3(x)\rvert$	$\tilde{P}_3(x)$	$\lvert xe^x - \tilde{P}_3(x)\rvert$
0.15	0.1743	0.1969	0.0226	0.1868	0.0125
0.25	0.3210	0.3435	0.0225	0.3358	0.0148
0.35	0.4967	0.5121	0.0154	0.5064	0.0097
0.65	1.245	1.233	0.012	1.231	0.014
0.75	1.588	1.572	0.016	1.571	0.017
0.85	1.989	1.976	0.013	1.974	0.015
1.15	3.632	3.650	0.018	3.644	0.012
1.25	4.363	4.391	0.028	4.382	0.019
1.35	5.208	5.237	0.029	5.224	0.016

Minimizing Lagrange interpolation error

The approximation using $\tilde{P}_3(x)$

$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352$$



Reducing the degree of approximating polynomials

As Chebyshev polynomials are efficient in approximating functions, we may use approximating polynomials of smaller degree for a given error tolerance.

For example, let $Q_n(x) = a_0 + \cdots + a_n x^n$ be a polynomial of degree n on $[-1, 1]$. Can we find a polynomial of degree $n-1$ to approximate Q_n?

Reducing the degree of approximating polynomials

So our goal is to find $P_{n-1}(x) \in \Pi_{n-1}$ such that

$$\max_{x\in[-1,1]} |Q_n(x) - P_{n-1}(x)|$$

is minimized. Note that $\frac{1}{a_n}(Q_n(x) - P_{n-1}(x)) \in \tilde{\Pi}_n$, we know the best choice is $\frac{1}{a_n}(Q_n(x) - P_{n-1}(x)) = \tilde{T}_n(x)$, i.e., $P_{n-1} = Q_n - a_n\tilde{T}_n$. In this case, we have approximation error

$$\max_{x\in[-1,1]} |Q_n(x) - P_{n-1}(x)| = \max_{x\in[-1,1]} |a_n\tilde{T}_n| = \frac{|a_n|}{2^{n-1}}$$

Reducing the degree of approximating polynomials

Example

Recall that $Q_4(x)$ be the 4th Maclaurin polynomial of $f(x) = e^x$ about 0 on $[-1, 1]$. That is

$$Q_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}$$

which has $a_4 = \frac{1}{24}$ and truncation error

$$|R_4(x)| = |\frac{f^{(5)}(\xi(x))x^5}{5!}| = |\frac{e^{\xi(x)}x^5}{5!}| \leq \frac{e}{5!} \approx 0.023$$

for $x \in (-1, 1)$. Given error tolerance 0.05, find the polynomial of small degree to approximate $f(x)$.

Reducing the degree of approximating polynomials

Solution. Let's first try Π_3. Note that $\tilde{T}_4(x) = x^4 - x^2 + \frac{1}{8}$, so we can set

$$\begin{aligned} P_3(x) &= Q_4(x) - a_4 \tilde{T}_4(x) \\ &= \left(1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}\right) - \frac{1}{24}\left(x^4 - x^2 + \frac{1}{8}\right) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 \in \Pi_3 \end{aligned}$$

Therefore, the approximating error is bounded by

$$\begin{aligned} |f(x) - P_3(x)| &\le |f(x) - Q_4(x)| + |Q_4(x) - P_3(x)| \\ &\le 0.023 + \frac{|a_4|}{2^3} = 0.023 + \frac{1}{192} \le 0.0283. \end{aligned}$$

Reducing the degree of approximating polynomials

Solution. (cont.) We can further try Π_2. Then we need to approximate P_3 (note $a_3 = \frac{1}{6}$) above by the following $P_2 \in \Pi_2$:

$$\begin{aligned} P_2(x) &= P_3(x) - a_3 \tilde{T}_3(x) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 - \frac{1}{6}\left(x^3 - \frac{3}{4}x\right) \\ &= \frac{191}{192} + \frac{9}{8}x + \frac{13}{24}x^2 \in \Pi_2 \end{aligned}$$

Therefore, the approximating error is bounded by

$$\begin{aligned} |f(x) - P_2(x)| &\le |f(x) - Q_4(x)| + |Q_4(x) - P_3(x)| + |P_3(x) - P_2(x)| \\ &\le 0.0283 + \frac{|a_3|}{2^2} = 0.0283 + \frac{1}{24} = 0.0703. \end{aligned}$$

Unfortunately this is larger than 0.05.

Reducing the degree of approximating polynomials

Although the error bound is larger than 0.05, the actual error is much smaller:

x	e^x	$P_4(x)$	$P_3(x)$	$P_2(x)$	$\lvert e^x - P_2(x)\rvert$
−0.75	0.47237	0.47412	0.47917	0.45573	0.01664
−0.25	0.77880	0.77881	0.77604	0.74740	0.03140
0.00	1.00000	1.00000	0.99479	0.99479	0.00521
0.25	1.28403	1.28402	1.28125	1.30990	0.02587
0.75	2.11700	2.11475	2.11979	2.14323	0.02623

Pros and cons of polynomial approxiamtion

Advantages:

- Polynomials can approximate continuous function to arbitrary accuracy;
- Polynomials are easy to evaluate;
- Derivatives and integrals are easy to compute.

Disadvantages:

- Significant oscillations;
- Large max absolute error in approximating;
- Not accurate when approximating discontinuous functions.

Rational function approximation

Rational function of degree $N = n + m$ is written as

$$r(x) = \frac{p(x)}{q(x)} = \frac{p_0 + p_1 x + \cdots + p_n x^n}{q_0 + q_1 x + \cdots + q_m x^m}$$

Now we try to approximate a function f on an interval containing 0 using $r(x)$.

WLOG, we set $q_0 = 1$, and will need to determine the $N + 1$ unknowns $p_0, \ldots, p_n, q_1, \ldots, q_m$.

Padé approximation

The idea of **Padé approximation** is to find $r(x)$ such that

$$f^{(k)}(0) = r^{(k)}(0), \quad k = 0, 1, \ldots, N$$

This is an extension of Taylor series but in the rational form.

Denote the Maclaurin series expansion $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Then

$$f(x) - r(x) = \frac{(\sum_{i=0}^{\infty} a_i x^i) \cdot (\sum_{i=0}^{m} q_i x^i) - \sum_{i=0}^{n} p_i x^i}{q(x)}$$

If we want $f^{(k)}(0) - r^{(k)}(0) = 0$ for $k = 0, \ldots, N$, we need the numerator to have 0 as a root of multiplicity $N + 1$.

Padé approximation

This turns out to be equivalent to

$$\sum_{i=0}^{k} a_i q_{k-i} = p_k, \quad k = 0, 1, \ldots, N$$

for convenience we used convention $p_{n+1} = \cdots = p_N = 0$ and $q_{m+1} = \cdots = q_N = 0$.

From these $N + 1$ equations, we can determine the $N + 1$ unknowns:

$$p_0, p_1, \ldots, p_n, q_1, \ldots, q_m$$

Padé approximation

Example

Find the Padé approximation to e^{-x} of degree 5 with $n = 3$ and $m = 2$.

Solution. We first write the Maclaurin series of e^{-x} as

$$e^{-x} = 1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \frac{1}{24}x^4 + \cdots = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!}x^i$$

Then for $r(x) = \frac{p_0+p_1x+p_2x^2+p_3x^3}{1+q_1x+q_2x^2}$, we need

$$\left(1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \cdots\right)(1 + q_1x + q_2x^2) - (p_0 + p_1x + p_2x^2 + p_3x^3)$$

to have 0 coefficients for terms $1, x, \ldots, x^5$.

Solution. (cont.) By solving this, we get p_0, p_1, p_2, q_1, q_2 and hence

$$r(x) = \frac{1 - \frac{3}{5}x + \frac{3}{20}x^2 - \frac{1}{60}x^3}{1 + \frac{2}{5}x + \frac{1}{20}x^2}$$

x	e^{-x}	$P_5(x)$	$\lvert e^{-x} - P_5(x)\rvert$	$r(x)$	$\lvert e^{-x} - r(x)\rvert$
0.2	0.81873075	0.81873067	8.64×10^{-8}	0.81873075	7.55×10^{-9}
0.4	0.67032005	0.67031467	5.38×10^{-6}	0.67031963	4.11×10^{-7}
0.6	0.54881164	0.54875200	5.96×10^{-5}	0.54880763	4.00×10^{-6}
0.8	0.44932896	0.44900267	3.26×10^{-4}	0.44930966	1.93×10^{-5}
1.0	0.36787944	0.36666667	1.21×10^{-3}	0.36781609	6.33×10^{-5}

where $P_5(x)$ is Maclaurin polynomial of degree 5 for comparison.

Chebyshev rational function approximation

To obtain more uniformly accurate approximation, we can use Chebyshev polynomials $T_k(x)$ in Padé approximation framework.

For $N = n + m$, we use

$$r(x) = \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

where $q_0 = 1$. Also write $f(x)$ using Chebyshev polynomials as

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$$

Chebyshev rational function approximation

Now we have

$$f(x) - r(x) = \frac{\sum_{k=0}^{\infty} a_k T_k(x) \sum_{k=0}^{m} q_k T_k(x) - \sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

We again seek for $p_0, \ldots, p_n, q_1, \ldots, q_m$ such that coefficients of $1, x, \ldots, x^N$ are 0.

To that end, the computations can be simplified due to

$$T_i(x) T_j(x) = \frac{1}{2} \Big(T_{i+j}(x) + T_{|i-j|}(x) \Big)$$

Also note that we also need to compute Chebyshev series coefficients a_k first.

Chebyshev rational function approximation

Example

Approximate e^{-x} using the Chebyshev rational approximation of degree $n = 3$ and $m = 2$. The result is $r_T(x)$.

x	e^{-x}	$r(x)$	$\lvert e^{-x} - r(x)\rvert$	$r_T(x)$	$\lvert e^{-x} - r_T(x)\rvert$
0.2	0.81873075	0.81873075	7.55×10^{-9}	0.81872510	5.66×10^{-6}
0.4	0.67032005	0.67031963	4.11×10^{-7}	0.67031310	6.95×10^{-6}
0.6	0.54881164	0.54880763	4.00×10^{-6}	0.54881292	1.28×10^{-6}
0.8	0.44932896	0.44930966	1.93×10^{-5}	0.44933809	9.13×10^{-6}
1.0	0.36787944	0.36781609	6.33×10^{-5}	0.36787155	7.89×10^{-6}

where $r(x)$ is the standard Padé approximation shown earlier.

Trigonometric polynomial approximation

Recall the Fourier series uses a set of $2n$ orthogonal functions with respect to weight $w \equiv 1$ on $[-\pi, \pi]$:

$$\begin{aligned} \phi_0(x) &= \frac{1}{2} \\ \phi_k(x) &= \cos kx, \quad k = 1, 2, \ldots, n \\ \phi_{n+k}(x) &= \sin kx, \quad k = 1, 2, \ldots, n-1 \end{aligned}$$

We denote the set of linear combinations of $\phi_0, \phi_1, \ldots, \phi_{2n-1}$ by $\mathcal{T}_n$, called the set of trigonometric polynomials of degree $\leq n$.

Trigonometric polynomial approximation

For a function $f \in C[-\pi, \pi]$, we want to find $S_n \in \mathcal{T}_n$ of form

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + b_k \sin kx)$$

to minimize the least squares error

$$E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \int_{-\pi}^{\pi} |f(x) - S_n(x)|^2 \, \mathrm{d}x$$

Due to orthogonality of Fourier series $\phi_0, \ldots, \phi_{2n-1}$, we get

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx \, \mathrm{d}x, \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx \, \mathrm{d}x$$

Trigonometric polynomial approximation

Example

Approximate $f(x) = |x|$ for $x \in [-\pi, \pi]$ using trigonometric polynomial from $\mathcal{T}_n$.

Solution. It is easy to check that $a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} |x|\,\mathrm{d}x = \pi$ and

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} |x|\cos kx\,\mathrm{d}x = \frac{2}{\pi k^2}((-1)^k - 1), \quad k = 1, 2, \ldots, n$$

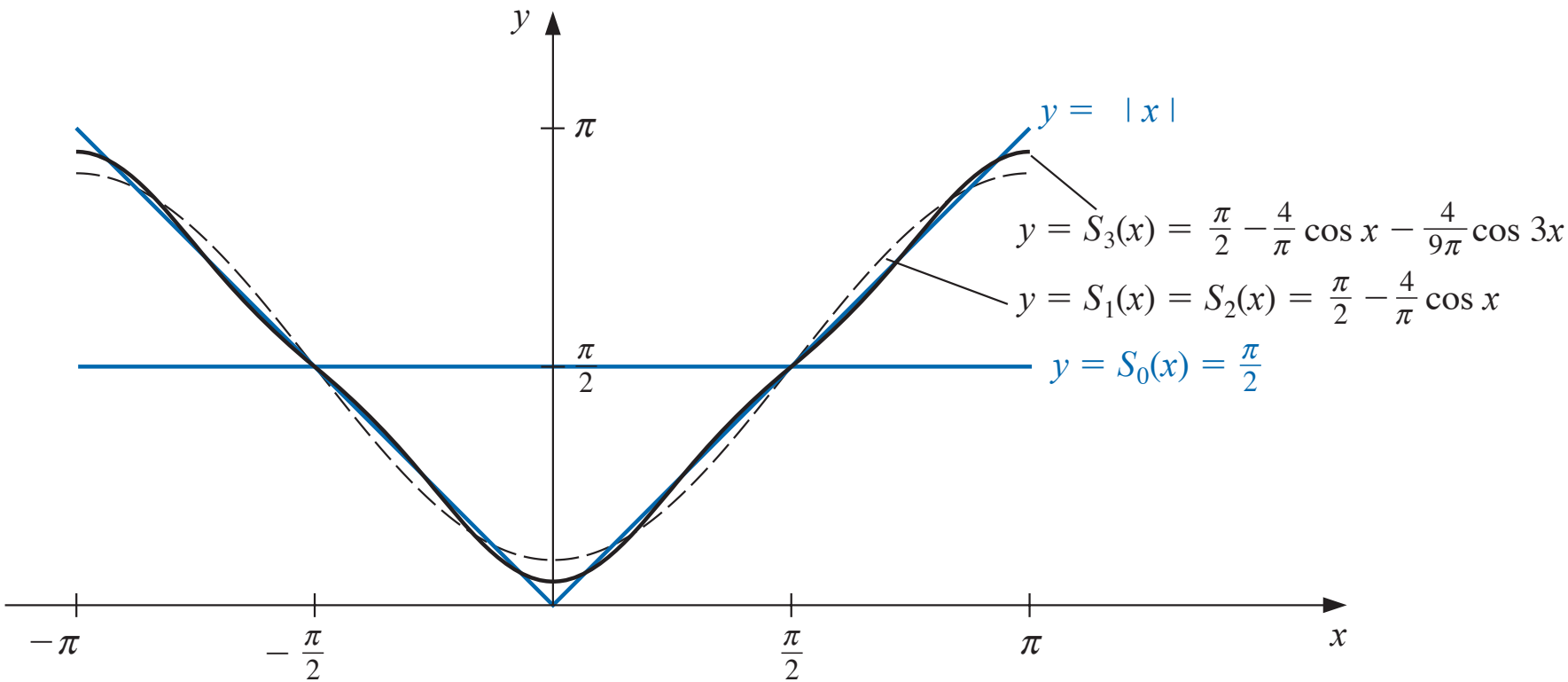
$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} |x|\sin kx\,\mathrm{d}x = 0, \quad k = 1, 2, \ldots, n-1$$

Therefore

$$S_n(x) = \frac{\pi}{2} + \frac{2}{\pi}\sum_{k=1}^{n} \frac{(-1)^k - 1}{k^2}\cos kx$$

Trigonometric polynomial approximation

$S_n(x)$ for the first few n are shown below:



Discrete trigonometric approximation

If we have $2m$ paired data points $\{(x_j, y_j)\}_{j=0}^{2m-1}$ where x_j are equally spaced on $[-\pi, \pi]$, i.e.,

$$x_j = -\pi + \left(\frac{j}{m}\right)\pi, \quad j = 0, 1, \ldots, 2m-1$$

Then we can also seek for $S_n \in \mathcal{T}_n$ such that the discrete least square error below is minimized:

$$E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \sum_{j=0}^{2m-1} \left(y_j - S_n(x_j)\right)^2$$

Discrete trigonometric approximation

Theorem

Define

$$a_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \cos kx_j, \quad b_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \sin kx_j$$

Then the trigonometric $S_n \in \mathcal{T}_n$ defined by

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1}(a_k \cos kx + b_k \sin kx)$$

minimizes the discrete least squares error

$$E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \sum_{j=0}^{2m-1} (y_j - S_n(x_j))^2$$

Fast Fourier transforms

The **fast Fourier transform (FFT)** employs the Euler formula $e^{z\mathrm{i}} = \cos z + \mathrm{i}\sin z$ for all $z \in \mathbb{R}$ and $\mathrm{i} = \sqrt{-1}$, and compute the discrete Fourier transform of data to get

$$\frac{1}{m}\sum_{k=0}^{2m-1} c_k e^{kx\mathrm{i}}, \text{ where } c_k = \sum_{j=0}^{2m-1} y_j e^{k\pi\mathrm{i}/m} \quad k = 0, \ldots, 2m-1$$

Then one can recover $a_k, b_k \in \mathbb{R}$ from

$$a_k + \mathrm{i}b_k = \frac{(-1)^k}{m} c_k \in \mathbb{C}$$

Fast Fourier transforms

The discrete trigonometric approximation for $2m$ data points requires a total of $(2m)^2$ multiplications, not scalable for large m.

The cost of FFT is only

$$3m + m\log_2 m = O(m\log_2 m)$$

For example, if $m = 1024$, then $(2m)^2 \approx 4.2 \times 10^6$ and $3m + m\log_2 m \approx 1.3 \times 10^4$. The larger m is, the more benefit FFT gains.