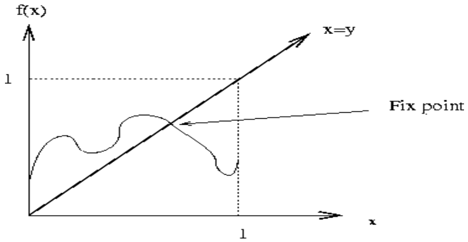
Fixed Point Theorem (I)

Theorem 2.3 **(i)** If $g \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$, then g has at least one fixed point in $[a,b]$.

(ii) If, in addition, $g'(x)$ exists on (a,b) and a positive constant $k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a,b),$$

then there is exactly one fixed point in $[a,b]$. (See Figure 2.4.) ■



Proof of Thm 2.3

- If $g(a) = a$ or $g(b) = b$,then g has a fixed point at an endpoint.
- Otherwise, $g(a) > a$ and $g(b) < b$. The function $h(x) = g(x) - x$ is continuous on $[a, b]$, with

$$h(a) = g(a) - a > 0 \quad \text{and} \quad h(b) = g(b) - b < 0.$$

- This implies that there exists $p \in (a, b)$, $h(p) = 0$.
- $g(p) - p = 0$, or $p = g(p)$.

If $|g'(x)| \leq k < 1$ for all x in (a, b), and p and q are two $\boxed{\text{distinct}}$ fixed points in $[a, b]$. Then a number ξ exists (Mean Value Theorem)

$$\frac{g(p) - g(q)}{p - q} = g'(\xi) < 1.$$

So

$$1 = \frac{p - q}{p - q} = \frac{g(p) - g(q)}{p - q} = g'(\xi) < 1.$$

Proof of Thm 2.3

- If $g(a) = a$ or $g(b) = b$,then g has a fixed point at an endpoint.
- Otherwise, $g(a) > a$ and $g(b) < b$. The function $h(x) = g(x) - x$ is continuous on $[a, b]$, with

$$h(a) = g(a) - a > 0 \quad \text{and} \quad h(b) = g(b) - b < 0.$$

- This implies that there exists $p \in (a, b)$, $h(p) = 0$.
- $g(p) - p = 0$, or $p = g(p)$.

If $|g'(x)| \leq k < 1$ for all x in (a, b), and p and q are two $\boxed{\text{distinct}}$ fixed points in $[a, b]$. Then a number ξ exists (Mean Value Theorem)

$$\frac{g(p) - g(q)}{p - q} = g'(\xi) < 1.$$

So

$$1 = \frac{p - q}{p - q} = \frac{g(p) - g(q)}{p - q} = g'(\xi) < 1. \boxed{\Longrightarrow \boxed{\text{distinct}} \Longleftarrow}$$

This contradiction implies uniqueness of fixed point.

Fixed Point Iteration

Given initial approximation p_0, define *Fixed Point Iteration*

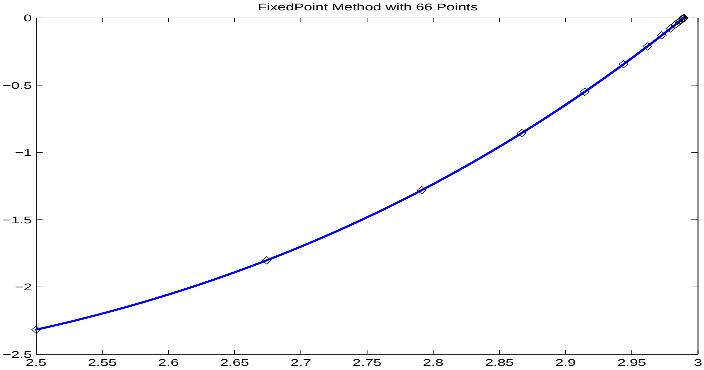
$$p_n = g(p_{n-1}), \quad n = 1, 2, \cdots,$$

If iteration converges to p, then

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

Fixed Point Example $x - \log(2 + 2x^2) = 0$: normal convergence

$$
\begin{aligned}
g(x) &= \log(2 + 2x^2) \in [2, 3] \quad \text{for} \quad x \in [2, 3], \\
|g'(x)| &\le \frac{4}{5} < 1.
\end{aligned}
$$



→ Instructors: Making Course Materials Accessible (click to expand)

Search for files

0 items selected

- Numerical Analysis (Fall 2023)
  - Exams and Solutions
  - Homework
  - Lecture Notes
  - Matlab
    - Chapter 2

Name ▲

- AitkenMethod.m
- bisection.m
- dmyfunc.m
- fixedpoint.m
- MullerDriver.m
- myfunc.m
- myfunc1.m
- NewtonMethod.m
- RootFinderDriver.m
- SecantMethod.m
- Steffensen.m

Fixed Point Theorem (II)

Theorem 2.4 (Fixed-Point Theorem)

Let $g \in C[a,b]$ be such that $g(x) \in [a,b]$, for all x in $[a,b]$. Suppose, in addition, that g' exists on (a,b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \le k, \quad \text{for all } x \in (a,b).$$

Then for any number p_0 in $[a,b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \ge 1,$$

converges to the unique fixed point p in $[a,b]$. ∎

Fixed Point Theorem (II)

Theorem 2.4 **(Fixed-Point Theorem)**

Let $g \in C[a,b]$ be such that $g(x) \in [a,b]$, for all x in $[a,b]$. Suppose, in addition, that g' exists on (a,b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a,b).$$

Then for any number p_0 in $[a,b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

converges to the unique fixed point p in $[a,b]$. ∎

$\boxed{\text{PRO: simple iteration}}$

$\boxed{\text{CON: conditions hard to verify}}$

No algorithm for finding $[a, b]$

Proof of Thm 2.4

- A unique fixed point $p \in [a, b]$ exists.
- $$|p_n - p| = |g(p_{n-1}) - g(p)| = |g'(\xi_n)(p_{n-1} - p)| \leq k|p_{n-1} - p|$$
- $$|p_n - p| \leq k|p_{n-1} - p| \leq k^2|p_{n-2} - p| \leq \cdots \leq k^n|p_0 - p|.$$
- Since
$$\lim_{n\to\infty} k^n = 0,$$
$\{p_n\}_{n=0}^{\infty}$ converges to p.

$\boxed{\text{No Harm Principle}}$ in numerical algorithm design

What we do not know never harms us

$\boxed{\text{No Harm Principle}}$ in numerical algorithm design

What we do not know never harms us

(NOT REALLY!!!)

$\boxed{\text{No Harm Principle}}$ in numerical algorithm design

What we do not know never harms us

(NOT REALLY!!!)

Trust but Verify

§2.3 Newton's Method for solving $f(p) = 0$

- Suppose that $f \in C^2[a, b]$.
- Let $p_0 \in [a, b]$ be an approximation to p with

$$f'(p_0) \neq 0, \quad \text{and} \quad |p - p_0| \quad \text{"small"}.$$

- Taylor expand $f(x)$ at $x = p$:

$$0 = f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p)).$$

"Solve" for p:

$$\begin{aligned} p &= p_0 - \frac{f(p_0)}{f'(p_0)} - \frac{(p - p_0)^2}{2f'(p_0)} f''(\xi(p)) \\ &\approx p_0 - \frac{f(p_0)}{f'(p_0)} \stackrel{def}{=} p_1. \end{aligned}$$

Newton's Method: $p_{k+1} = p_k - \frac{f(p_k)}{f'(p_k)}, \quad k = 0, 1, \cdots$

Newton's Method for solving $f(p) = 0$

$$\begin{aligned} p &= p_0 - \frac{f(p_0)}{f'(p_0)} - \frac{(p-p_0)^2}{2f'(p_0)} f''(\xi(p)) \\ &\approx p_0 - \frac{f(p_0)}{f'(p_0)} \stackrel{def}{=} p_1. \end{aligned}$$

- If p_0 "close to" p, we can expect fast convergence.
- Best hope in practice: p_0 "not too far from" p. Newton's method may or may not converge.
- If Newton's method converges, it converges quickly.

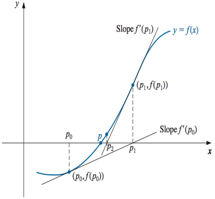
Geometry of Newton's Method

- Taylor expand $f(x)$ at $x = p$:

$$0 = f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p)).$$

- Replace $f(x)$ by a straight line:

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



- $$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} \quad \text{is}$$

  horizontal intercept of straight line

$$y = f(p_0) + (x - p_0)f'(p_0)$$

```
function [fun, dfun, x, out] = NewtonMethod(Fcn, dFcn,x0, params)
% On input:
%   Fun and dFun are the names of function and its derivative.
%   x0 is initial guess, and tol tolerance.
%
% On output
%   fun and dfun contain all function values and derivatives
%   computed by Newton
%   out.flg = 0 means success; otherwise method failed.
%   x(end) is the root if out.flg = 0.
%   out.it = # of iterations.
%
% Written by Ming Gu for Math 128A, Spring 2021
%
out.flg = 1;
x(1)    = x0;

N = params.MaxIt;
tol = params.tol;
out.x = [];
out.f = [];

for k = 1:N
    fun(k)  = Fcn(x(k));
    dfun(k) = dFcn(x(k));
    out.x = [out.x;x(k)];
    out.f =[out.f;fun(k)];
    if (abs(fun(k)) < tol)
       out.flg = 0;
       out.it = k;
       return;
    end
    if (dfun(k) == 0)
       out.it = k;
       return;
    end
    x(k+1) = x(k) - fun(k)/dfun(k);
end
```

Theorem 2.6 Let $f \in C^2[a, b]$. If $p \in (a, b)$ is such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p - \delta, p + \delta]$. ■

Newton's Method as Fixed Point Iteration

Assume $f(p) = 0$ is a root, and x close to p. Would like a fixed point iteration

$$p_{k+1} = g(p_k), \quad k = 0, 1, \cdots,$$

where $g(x) = x - f(x)\,h(x)$. $\boxed{\text{How to choose } h(x)\text{?}}$

Newton's Method as Fixed Point Iteration

Assume $f(p) = 0$ is a root, and x close to p. Would like a fixed point iteration

$$p_{k+1} = g(p_k), \quad k = 0, 1, \cdots,$$

where $g(x) = x - f(x)\,h(x)$. $\boxed{\text{How to choose } h(x)\text{?}}$ Since

$$g'(x) = 1 - f'(x)\,h(x) - f(x)\,h(x)' \approx 1 - f'(x)\,h(x).$$

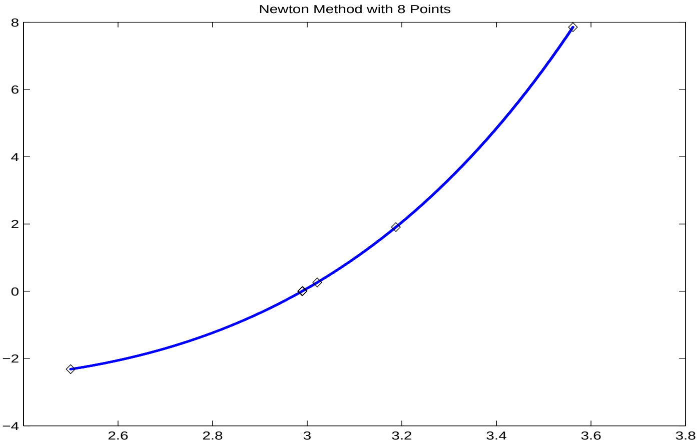
"optimal choice": $g'(x) \approx 0$, so that

$$h(x) = \frac{1}{f'(x)}$$

and so

$$p_{k+1} = p_k - \frac{f(p_k)}{f'(p_k)}, \quad k = 0, 1, \cdots.$$

Newton Method for $f(x) = e^x - (2 + 2x^2)$



Computing square root with Newton's Method

- Given $a > 0$, $p \stackrel{def}{=} \sqrt{a}$ is positive root of equation

$$f(x) \stackrel{def}{=} x^2 - a = 0.$$

- Newton's Method

$$p_{k+1} = p_k - \frac{p_k^2 - a}{2p_k} = \frac{1}{2}\left(p_k + \frac{a}{p_k}\right), k = 0, 1, 2, \cdots,$$

- Newton's Method is well defined for any $p_0 > 0$.

Computing square root with Newton's Method: Analysis

Assume that $p_0 > 0$, then for $k = 0, 1, 2, \cdots$,

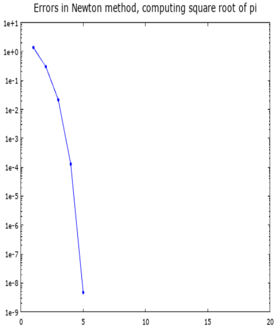
- Newton's Method satisfies

$$p_{k+1} - \sqrt{a} = \frac{1}{2}\left(p_k + \frac{a}{p_k}\right) - \sqrt{a} = \frac{1}{2\,p_k}\left(p_k - \sqrt{a}\right)^2 \geq 0,$$

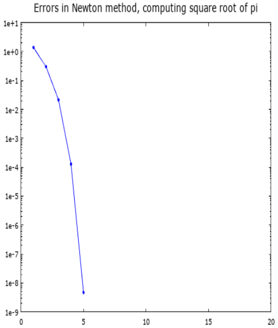
- It follows that $p_k - \sqrt{a} \geq 0$ for all $k \geq 1$.
- Newton's Method converges for any $p_0 > 0$:

$$0 \leq p_{k+1} - \sqrt{a} \leq \frac{1}{2}\left(p_k - \sqrt{a}\right) \leq \cdots \leq \frac{1}{2^k}\left(p_1 - \sqrt{a}\right) \longrightarrow 0.$$

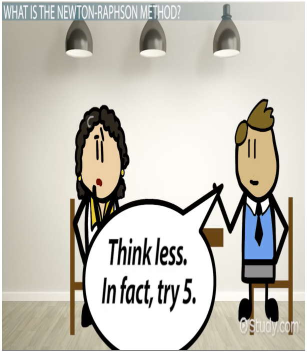
Newton Method for square root



Newton Method for square root



Newton Method: Think 5 (if it works)

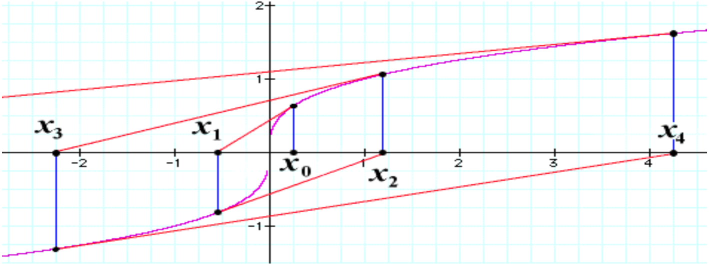


Theorem 2.6 Let $f \in C^2[a, b]$. If $p \in (a, b)$ is such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p - \delta, p + \delta]$. ∎

Proof of Theorem 2.6

- Newton's method is fixed point iteration $p_n = g(p_{n-1}), g(x) = x - \frac{f(x)}{f'(x)}$.
- Since $f'(p) \neq 0$, there exists an interval $[p-\delta_1, p+\delta_1] \subset [a, b]$ on which $f'(x) \neq 0$. Thus, $g(x)$ is defined on $[p-\delta_1, p+\delta_1]$.
- $$g'(x) = 1 - \frac{f'(x)\,f'(x) - f(x)\,f''(x)}{\left(f'(x)\right)^2} = \frac{f(x)\,f''(x)}{\left(f'(x)\right)^2} \in C[p-\delta_1, p+\delta_1].$$
- Since $g'(p) = 0$, there exists $0 < \delta < \delta_1$ so that
$$|g'(x)| \leq \kappa \quad (= \frac{1}{2}), \quad \text{for all} \quad x \in [p-\delta, p+\delta].$$
- If $x \in [p-\delta, p+\delta]$, then
$$|g(x)-p| = |g(x)-g(p)| = |g'(\xi)(x-p)| \leq \kappa|x-p| \leq |x-p|.$$
Therefore $g(x) \in [p-\delta, p+\delta]$.
- $\{p_n\}$ converges to p by Fixed Point Theorem.

Newton Method Divergence Example: $f(x) = x^{1/3}$



$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^{1/3}}{1/3\, x_k^{-2/3}} = x_k - 3\, x_k = -2\, x_k$$

Secant Method: Poor man's Newton Method

Motivation

- Newton method style of fast convergence
- Avoid need for derivative calculations.

Approach

- Newton method: $p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$.
- Replace $f'(p_n)$ by its cheap approximation

$$f'(p_n) = \lim_{x \to} \frac{f(p_n) - f(x)}{p_n - x} \approx \frac{f(p_n) - f(p_{n-1})}{p_n - p_{n-1}}.$$

- Secant method

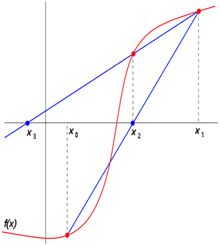
$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}, n = 1, 2, \cdots .$$

Secant Method: Geometry

- "Approximate" $f(x)$ by a straight line

  $$f(x) \approx \frac{(x-p_0)f(p_1)-(x-p_1)f(p_0)}{p_1-p_0}.$$

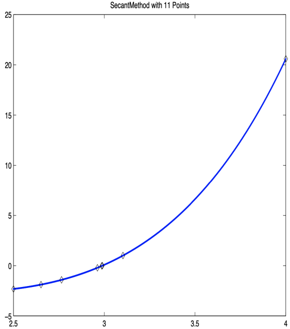
  Both $f(x)$ and straight line go through points $(p_0, f(p_0)), (p_1, f(p_1))$.



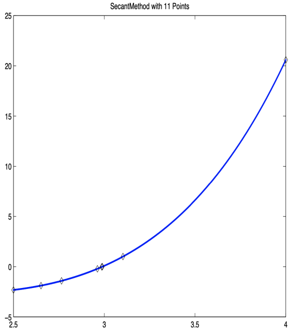
- Let approximate root p_2 be the x-intercept of the straight line

  $$p_2 = \frac{p_0 f(p_1) - p_1 f(p_0)}{f(p_1)-f(p_0)} = p_1 - \frac{f(p_1)(p_1-p_0)}{f(p_1)-f(p_0)}.$$

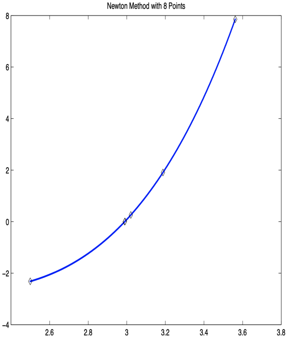
Secant method for $f(x) = e^x - (2 + 2x^2)$



Secant method for
$f(x) = e^x - (2 + 2x^2)$

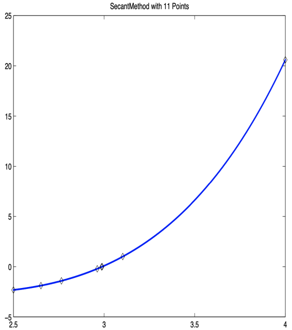


Recall Newton Method

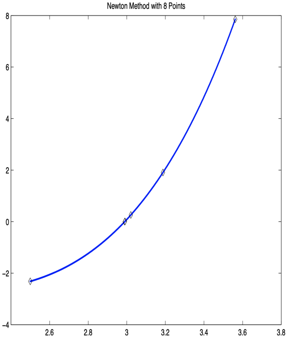


Fewer iterations, and less work?

Secant method for
$f(x) = e^x - (2 + 2x^2)$



Recall Newton Method



$\boxed{\text{Fewer iterations, and less work?}}$

$\boxed{\text{Secant faster than Newton}}$

§2.4 Number of iterations vs. error in the solution

- Function to be considered

$$g(x) = log(2 + 2x^2), \quad f(x) = x - g(x) = x - log(2 + 2x^2).$$

- Root p of f (i.e., $f(p) = 0$)

$$p = 2.98930778246493e + 00.$$

Method comparison:

- Bisection Method: 52 iterations
- Fixed Point Iteration: 66 iterations
- Newton's Method: 8 iterations
- Secant Method: 11 iterations

§2.4 Number of iterations vs. error in the solution

- Function to be considered

$$g(x) = log(2+2x^2), \quad f(x) = x - g(x) = x - log(2+2x^2).$$

- Root p of f (i.e., $f(p) = 0$)

$$p = 2.98930778246493e+00.$$

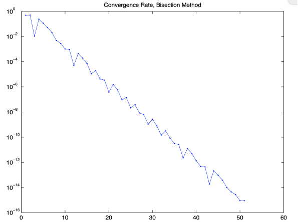
Method comparison:

- Bisection Method: 52 iterations
- Fixed Point Iteration: 66 iterations
- Newton's Method: 8 iterations
- Secant Method: 11 iterations

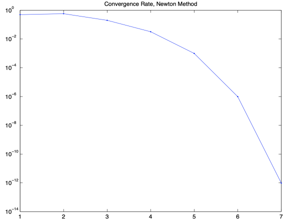
Even in convergence, rates differ

convergence comparison, semi-log scale

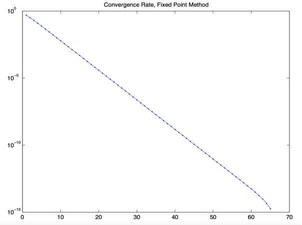
Bisection



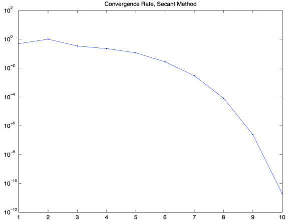
Newton Method



Fixed Point Iteration



Secant Method



Linear and Quadratic Order of convergence

DEFINITION Suppose $\{p_n\}_{n=1}^{\infty}$ is a sequence that converges to p with $p_n \neq p$ for all n. If positive constants λ and α exist with

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

then $\{p_n\}_{n=1}^{\infty}$ converges to p of order α, with asymptotic error constant λ.

Linear and Quadratic Order of convergence

DEFINITION Suppose $\{p_n\}_{n=1}^{\infty}$ is a sequence that converges to p with $p_n \neq p$ for all n. If positive constants λ and α exist with

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

then $\{p_n\}_{n=1}^{\infty}$ converges to p of order α, with asymptotic error constant λ.

- If $\alpha = 2$, then $\{p_n\}_{n=1}^{\infty}$ converges quadratically.
- If $\alpha = 1$ and $\lambda < 1$, $\{p_n\}_{n=1}^{\infty}$ converges linearly.
- If $\alpha = 1$ and $\lambda = 0$, $\{p_n\}_{n=1}^{\infty}$ converges super-linearly.

Recall and contrast: rate of convergence, the Big O

Suppose $\{\beta_n\}_{n=1}^{\infty}$ is a sequence known to converge to zero, and $\{\alpha_n\}_{n=1}^{\infty}$ converges to a number α. If a positive constant K exists with

$$|\alpha_n - \alpha| \leq K|\beta_n|, \quad \text{for large } n,$$

then we say that $\{\alpha_n\}_{n=1}^{\infty}$ converges to α with **rate, or order, of convergence** $O(\beta_n)$. (This expression is read "big oh of β_n".) It is indicated by writing $\alpha_n = \alpha + O(\beta_n)$. ■

the Big $O()$ = rate of convergence

Linear and Quadratic Order of convergence (I)

- Suppose that $\{p_n\}_{n=1}^{\infty}$ is linearly convergent to 0,

$$\lim_{n\to\infty} \frac{|p_{n+1}|}{|p_n|} = 0.5, \quad \text{or roughly} \quad \frac{|p_{n+1}|}{|p_n|} \approx 0.5,$$

hence $\boxed{p_n \approx (0.5)^n |p_0|.}$

- Suppose that $\{\tilde{p}_n\}_{n=1}^{\infty}$ is quadratically convergent to 0,

$$\lim_{n\to\infty} \frac{|\tilde{p}_{n+1}|}{|\tilde{p}_n|^2} = 0.5, \quad \text{or roughly} \quad \frac{|\tilde{p}_{n+1}|}{|\tilde{p}_n|^2} \approx 0.5.$$

But now

$$\begin{aligned}|\tilde{p}_n| &\approx 0.5|\tilde{p}_{n-1}|^2 \approx (0.5)[0.5|\tilde{p}_{n-2}|^2]^2 = (0.5)^3|\tilde{p}_{n-2}|^4\\ &\approx (0.5)^3[(0.5)|\tilde{p}_{n-3}|^2]^4 = (0.5)^7|\tilde{p}_{n-3}|^8\\ &\approx \cdots \approx (0.5)^{2^n-1}|\tilde{p}_0|^{2^n}.\end{aligned}$$

Linear and Quadratic Order of convergence (II)

Consider $\quad p_n \approx (0.5)^n \quad$ and $\quad \tilde{p}_n \approx (0.5)^{2^n}$.