Order of convergence

Suppose $\{p_n\}_{n=0}^{\infty}$ 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 $\{p_n\}_{n=0}^{\infty}$ converges to p of order α, with asymptotic error constant λ.

Linear and Quadratic Order of convergence

(i) If $\alpha = 1$ (and $\lambda < 1$), the sequence is linearly convergent.

(ii) If $\alpha = 2$, the sequence is quadratically convergent.

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 $p_0 > 0$.

Newton's Method for $\sqrt{a}$ converges quadratically if $p_0 > 0$

- $\{p_k\}_{k=1}^{\infty}$ bounded below by $\sqrt{a}$:

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

- $\{p_k\}_{k=1}^{\infty}$ monotonically decreasing:

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

- $\lim_{k\to\infty} p_k \stackrel{def}{=} p$ exists and satisfies Newton Iteration:

$$p = \frac{1}{2}\left(p + \frac{a}{p}\right) \geq \sqrt{a}, \quad \text{therefore} \quad p = \sqrt{a}.$$

- Newton's Method quadratically convergent

$$\lim_{k\to\infty} \frac{|p_k - \sqrt{a}|}{\left|p_{k-1} - \sqrt{a}\right|^2} = \lim_{k\to\infty} \frac{1}{2p_{k-1}} = \frac{1}{2\sqrt{a}}.$$

Linear Convergence Theorem of Fixed Point Iteration

- **Let** $g \in C[a, b]$ satisfy $g(x) \in [a, b]$ for all $x \in [a, b]$.
- **Assume** $g'(x)$ continuous with $|g'(x)| \leq \kappa$ for all $x \in (a, b)$ and constant $\kappa < 1$.
- **Then** Fixed Point Iteration

$$p_k = g(p_{k-1}), k = 1, 2, \cdots ,$$

for any $p_0 \in [a, b]$ converges to fixed point $p \in [a, b]$ with

$$\lim_{k \to \infty} \frac{|p_{k+1} - p|}{|p_k - p|} = |g'(p)| \leq \kappa, \quad \text{if } p_k \neq p \text{ for all } k.$$

(not linear convergence if $g'(p) = 0$)

Proof of Linear Convergence

- By Fixed Point Theorem, Fixed Point $p \in [a, b]$ uniquely exists, and FPI converges to p.
- By Mean Value Theorem, there exists a ξ_k in between p_k and p such that

$$g(p_k) - g(p) = g'(\xi_k)(p_k - p), \quad \text{for all } k.$$

- Therefore

$$\begin{aligned}
\lim_{k\to\infty} g'(\xi_k) &= g'(p) \\
\lim_{k\to\infty} \frac{|p_{k+1} - p|}{|p_k - p|} &= \lim_{k\to\infty} \frac{|g(p_k) - g(p)|}{|p_k - p|} = \lim_{k\to\infty} \frac{|g'(\xi_k)(p_k - p)|}{|p_k - p|} \\
&= |g'(p)|
\end{aligned}$$

Quadratic Convergence of Fixed Point Iteration

- **Assume** FPI $\{p_k\}_{k=1}^{\infty}$ converges to p, with $g'(p) = 0$, then

$$\lim_{k\to\infty} \frac{|p_{k+1} - p|}{|p_k - p|^2} = \frac{1}{2}|g''(p)|, \quad \text{if } p_k \neq p \text{ for all } k.$$

(not quadratic convergence if $g''(p) = 0$)

Proof of Quadratic Convergence

- Taylor expansion at p:

$$
\begin{aligned}
g(p_k) &\xlongequal{\xi_k \text{ between } p,\ p_k} g(p) + g'(p)(p_k - p) + \frac{1}{2} g''(\xi_k)(p_k - p)^2 \\
&\xlongequal{\qquad\qquad} g(p) + \frac{1}{2} g''(\xi_k)(p_k - p)^2
\end{aligned}
$$

- Therefore

$$
\begin{aligned}
\lim_{k\to\infty} \frac{|p_{k+1} - p|}{|p_k - p|^2} &= \lim_{k\to\infty} \frac{|g(p_k) - p|}{|p_k - p|^2} = \frac{1}{2} \lim_{k\to\infty} \frac{|g''(\xi_k)(p_k - p)^2|}{|p_k - p|^2} \\
&= \frac{1}{2} |g''(p)|
\end{aligned}
$$

Quadratic Convergence of Newton Iteration

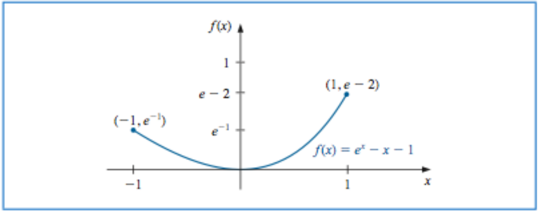
- Newton Iteration is FPI with
$$g(x) = x - \frac{f(x)}{f'(x)},$$
and with $f(p) = 0$.
- derivatives
$$\begin{aligned} g'(x) &= \frac{f(x) f''(x)}{\left(f'(x)\right)^2} \\ g''(x) &= \frac{f''(x)}{f'(x)} + \frac{f'(x)\, f'''(x) - 2\left(f''(x)\right)^2}{\left(f'(x)\right)^3}\, f(x) \end{aligned}$$
- therefore
$$\begin{aligned} g'(p) &= 0, \quad \text{and} \quad g''(p) = \frac{f''(p)}{f'(p)}, \\ \lim_{k\to\infty} \frac{|p_{k+1} - p|}{|p_k - p|^2} &= \frac{1}{2}\left|\frac{f''(p)}{f'(p)}\right|, \quad \text{if } f'(p) \neq 0. \end{aligned}$$

Multiple roots: $f(p) = 0, f'(p) = 0$

- **Definition**: A solution p of $f(x) = 0$ is a root of multiplicity m of f if for $x \neq p$, we can write
$$f(x) = (x-p)^m \, q(x), \quad \text{where} \quad \lim_{x \to p} q(x) \neq 0.$$
- **Theorem**: The function $f \in C^m[a,b]$ has a root of multiplicity m at p in (a,b) if and only if
$$0 = f(p) = f'(p) = f''(p) = \cdots = f^{(m-1)}(p), \text{ but } f^{(m)}(p) \neq 0.$$
- **simple root**: f satisfies $f(p) = 0$, but $f'(p) \neq 0$.

Example: $f(x) = e^x - x - 1, f(0) = f'(0) = 0, f''(0) = 1$

Function

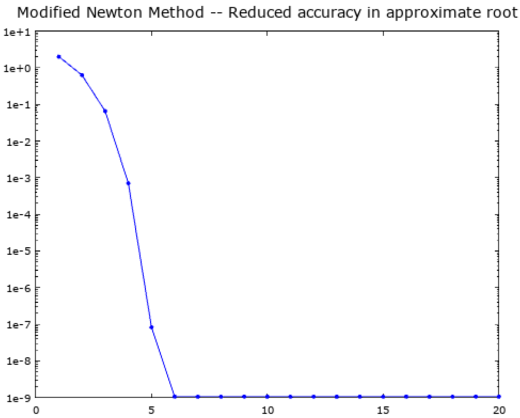


Newton's Method

n	p_n
0	1.0
1	0.58198
2	0.31906
3	0.16800
4	0.08635
5	0.04380
6	0.02206
7	0.01107
8	0.005545
9	2.7750×10^{-3}
10	1.3881×10^{-3}
11	6.9411×10^{-4}
12	3.4703×10^{-4}
13	1.7416×10^{-4}
14	8.8041×10^{-5}
15	4.2610×10^{-5}
16	1.9142×10^{-6}

Modified Newton Method

$$p_{k+1} = p_k - \frac{m\, f(p_k)}{f'(p_k)}, \quad \text{for} \quad k = 1, 2, \cdots, \quad (m = \text{multiplicity of root } p.)$$



- $m = 2$ for function $f(x) = e^x - x - 1$,

Modified Newton Method is Quadratically Convergent

$$p_{k+1} = g(p_k), \quad g(x) = x - \frac{m\, f(x)}{f'(x)}, \text{ for } k = 1, 2, \cdots,$$

- Let $f(x) = (x-p)^m q(x), \quad q(p) \neq 0.$
- $$g(x) = x - \frac{m\, f(x)}{f'(x)} = x - \frac{m\,(x-p)q(x)}{mq(x) + (x-p)q'(x)}.$$
- $g'(p) = 0$, hence quadratic convergence.

Accelerating Convergence: Aitken's Δ^2 Method

- **Suppose** $\{p_k\}_{k=1}^{\infty}$ linearly converges to limit p,

$$\lim_{k\to\infty} \frac{p_{k+1}-p}{p_k-p} = \lambda, \quad |\lambda| < 1.$$

- **Define**

$$\frac{p_{k+1}-p}{p_k-p} \stackrel{def}{=} \lambda_k,$$

  so that $\{\lambda_k\}_{k=1}^{\infty}$ converges to λ.

- It follows that

$$0 \approx \lambda_{k+1} - \lambda_k = \frac{p_{k+2}-p}{p_{k+1}-p} - \frac{p_{k+1}-p}{p_k-p}.$$

- Solve for p:

$$p = \frac{p_{k+1}^2 - p_k p_{k+2}}{2p_{k+1} - p_k - p_{k+2}} + \frac{(\lambda_{k+1}-\lambda_k)(p_{k+1}-p)(p_k-p)}{2(p_{k+1}-p) - (p_k-p) - (p_{k+2}-p)}.$$

Accelerating Convergence: Aitken's Δ^2 Method

$$\widehat{p}_k \stackrel{def}{=} \frac{p_{k+1}^2 - p_k p_{k+2}}{2p_{k+1} - p_k - p_{k+2}}$$

$$= p_k - \frac{(p_{k+1} - p_k)^2}{p_{k+2} - 2p_{k+1} + p_k} \stackrel{def}{=} \{\Delta^2\}(p_k).$$

Approximation Error

$$|\widehat{p}_k - p| = \left| \frac{(\lambda_{k+1} - \lambda_k)(p_{k+1} - p)(p_k - p)}{2(p_{k+1} - p) - (p_k - p) - (p_{k+2} - p)} \right|$$

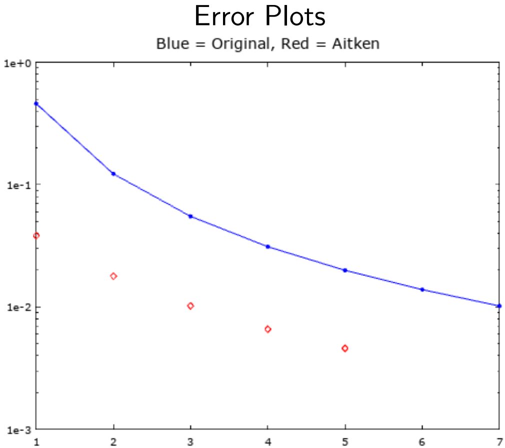
$$= \left| \frac{(\lambda_{k+1} - \lambda_k)(p_k - p)}{2 - \left(\frac{p_{k+1}-p}{p_k-p}\right)^{-1} - \frac{p_{k+2}-p}{p_{k+1}-p}} \right|$$

$$\approx \left| \frac{(\lambda_{k+1} - \lambda_k)(p_k - p)}{2 - \lambda^{-1} - \lambda} \right| \ll O\left(|p_k - p|\right).$$

Accelerating Convergence: Aitken's Δ^2 Method

$$p_n = \cos(1/n)$$

n	p_n	$\hat{p}_n$
1	0.54030	0.96178
2	0.87758	0.98213
3	0.94496	0.98979
4	0.96891	0.99342
5	0.98007	0.99541
6	0.98614	
7	0.98981	



Accelerating Convergence: Steffensen's Method

- Aitken's Acceleration for a given $\{p_k\}_{k=1}^{\infty}$:

$$\{\Delta^2\}(p_k) = p_k - \frac{(p_{k+1} - p_k)^2}{p_{k+2} - 2p_{k+1} + p_k}.$$

- Steffensen's Method: use Aitken's Acceleration as soon as iterations are generated:
  - Given $p_0^{(0)}$
  - for $k = 0, 1, 2, \cdots,$

$$p_1^{(k)} = g(p_0^{(k)}), \quad p_2^{(k)} = g(p_1^{(k)}), \quad p_0^{(k+1)} = \{\Delta^2\}(p_0^{(k)}).$$

Steffensen's

To find a solution to $p = g(p)$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p_1 = g(p_0)$; (*Compute* $p_1^{(i-1)}$.)
$p_2 = g(p_1)$; (*Compute* $p_2^{(i-1)}$.)
$p = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$. (*Compute* $p_0^{(i)}$.)

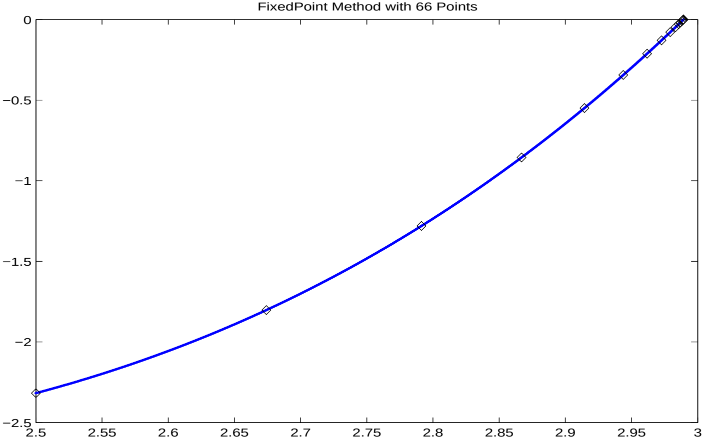
Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*Procedure completed successfully.*)
STOP.

Step 5 Set $i = i + 1$.

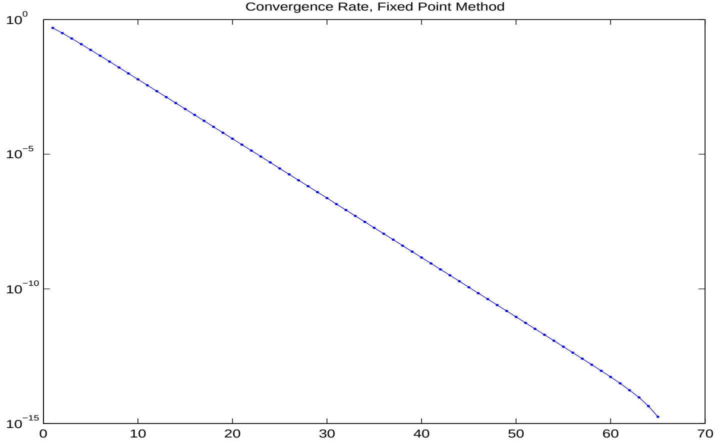
Step 6 Set $p_0 = p$. (*Update* p_0.)

Step 7 OUTPUT ('Method failed after N_0 iterations, N_0 =', N_0);
(*Procedure completed unsuccessfully.*)
STOP.

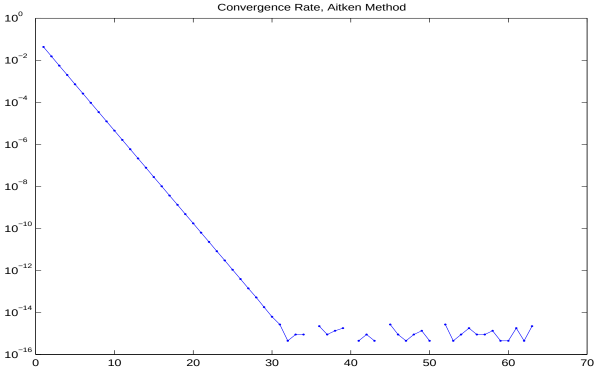
Fixed point for $g(x) = log(2 + 2x^2)$



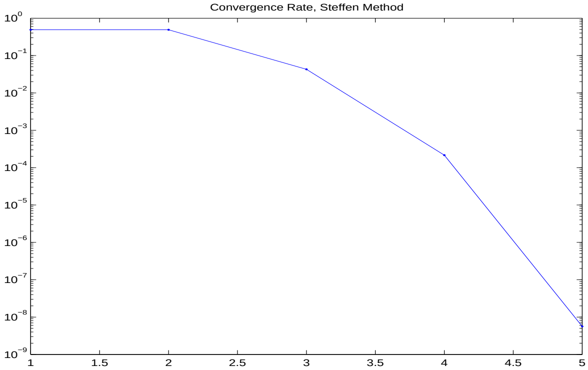
Fixed Point Iteration: Linear Convergence



Aitken's Method: Faster, but Linear Convergence



Steffenson's Method: Quadratic Convergence



Newton's Method on Polynomial roots = Horner's Method

Let

$$P(x) = a_n x_n + a_{n-1} x_{n-1} + \cdots + a_1 x + a_0.$$

Horner's Method computes $P(x_0)$ and $P'(x_0)$

- **define** $b_n = a_n$
- **for** $k = n-1, n-2, \cdots, 1, 0$

$$b_k = a_k + b_{k+1} x_0,$$

- then

$$\begin{aligned} b_0 &= P(x_0) \\ P(x) &= (x - x_0)\, Q(x) + b_0 \\ Q(x) &= b_n x_{n-1} + b_{n-1} x_{n-2} + \cdots + b_2 x + b_1. \end{aligned}$$

- it follows

$$\begin{aligned} P'(x) &= Q(x) + (x - x_0)\, Q'(x) \\ P'(x_0) &= Q(x_0) \end{aligned}$$

Horner's Method

INPUT degree n; coefficients $a_0, a_1, \ldots, a_n; x_0$.

OUTPUT $y = P(x_0); z = P'(x_0)$.

Step 1 Set $y = a_n$; *(Compute b_n for P.)*

$z = a_n$. *(Compute b_{n-1} for Q.)*

Step 2 For $j = n-1, n-2, \ldots, 1$

set $y = x_0 y + a_j$; *(Compute b_j for P.)*

$z = x_0 z + y$. *(Compute b_{j-1} for Q.)*

Step 3 Set $y = x_0 y + a_0$. *(Compute b_0 for P.)*

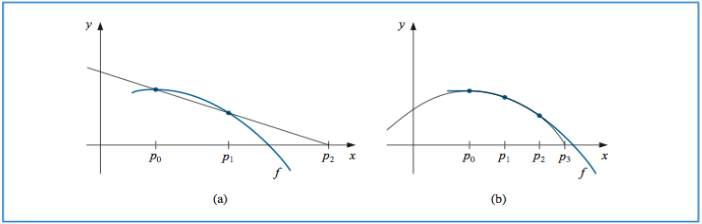
Step 4 OUTPUT (y, z);

STOP.

Muller's Method: finding complex roots

- **Given** three points $(p_0, f(p_0)), (p_1, f(p_1))$, and $(p_2, f(p_2))$.
- **Construct** a parabola through them,
- p_3 is the intersection of x-axis with parabola closer to p_2.

Secant Method vs. Muller's Method



Muller's Method: derivation

Choose parabola

$$P(x) = a(x - p_2)^2 + b(x - p_2) + c.$$

a, b, c satisfy

$$f(p_0) = a(p_0 - p_2)^2 + b(p_0 - p_2) + c,$$

$$f(p_1) = a(p_1 - p_2)^2 + b(p_1 - p_2) + c,$$

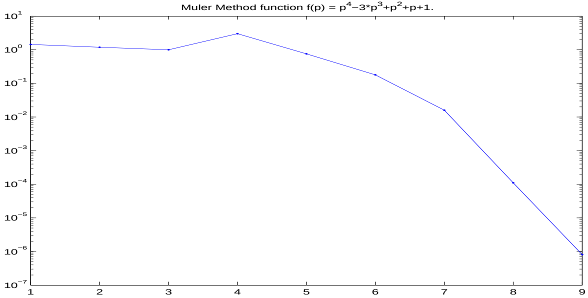
$$f(p_2) = a \cdot 0^2 + b \cdot 0 + c = c$$

Muller's Method: finding complex roots

- p_3 is the intersection of x-axis with parabola closer to p_2.

$$p_3 = p_2 - \frac{2c}{b + \operatorname{sgn}(b)\sqrt{b^2 - 4ac}},$$

Muller's Method



Choices, and more Choices

- Bisection Method: slow (linearly convergent); reliable (always finds a root given interval $[a, b]$ with $f(a) \cdot f(b) < 0$.)
- Fixed Point Iteration: slow (linearly convergent); no need for interval $[a, b]$ with $f(a) \cdot f(b) < 0$.
- Newton's Method: fast (quadratically convergent); could get burnt (need not converge.)
- Secant Method: between Bisection and Newton in speed; need not converge; no derivative needed.
- Steffensen's Method: fast (quadratically convergent); no derivative needed. not a method of choice in higher dimensions.
- Muller's Method: can handle complex roots; need not converge.

Brent's Method: Motivation

- $\boxed{\text{(Mostly) speed of Steffensen's Method}}$
- $\boxed{\text{Reliability of Bisection Method}}$

Brent's Method = Zeroin

Zero In

Zero In Japanese cover

Brent's Method: Zeroin (Wikipedia)

Brent's method

From Wikipedia, the free encyclopedia

In numerical analysis, **Brent's method** is a root-finding algorithm combining the bisection method, the secant method and inverse quadratic interpolation. It has the reliability of bisection but it can be as quick as some of the less-reliable methods. The algorithm

Brent's Method: Zeroin

Cleve's Corner: Cleve Moler on Mathematics and Computing

Scientific computing, math & more

< Zeroin, Part 1: Dekker's Algorithm | Zeroin, Part 3: MATLAB Zero.

Recent Posts Archive

23 JAN Ulps Plots Reveal Math Function Accurary

5 JAN Fitting and Extrapolating US Census Data

25 DEC Season's Greetings Fractal

12 DEC My First Matrix, RGB, YIQ, and Color Cubes

29 NOV Four Fundamental Subspaces of Linear Algebra, Corrected

Categories

History 71

Zeroin, Part 2: Brent's Version 2

Posted by **Cleve Moler**, October 26, 2015

Richard Brent's improvements to Dekker's *zeroin* algorithm, published in 1971, made it faster, safer in floating point arithmetic, and guaranteed not to fail.

Contents

Brent's Method

- Worst case number of iterations $= (n^2)$, where n is number of iterations of Bisection Method for a given tolerance.

Modified Brent's Method

Finding Zeros of Single-Variable, Real Functions

Gautam Wilkins

University of California, San Diego

Modified Zero-In

- Brent's Method may be modified to ensure $O(n)$ time instead of $O(n^2)$.

- Force a bisection if:

  1) If the size of the original interval is not reduced by a factor of 1/2 after five interpolation steps.

  2) If an interpolation step produces a point, x, such that $|f(x)|$ is not a factor of 1/2 smaller than the previous best point.

Modified Zero-In

- The first condition ensures that the complexity of the search is $O(n)$.

- The second condition addresses the issue of very flat functions, for which Brent's Method converges rather slowly.

Modified Brent's Method

Comparison

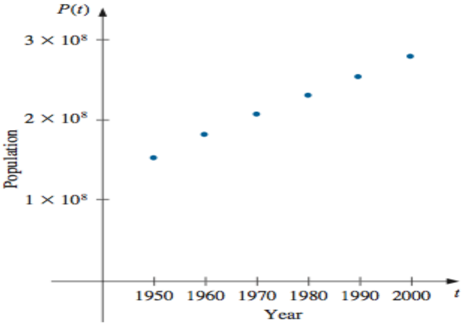
- For the worst-case function shown earlier, when Brent's Method took 2914 iterations, Modified Zero-In took 85 iterations.

- This reduction in complexity, as far as we can tell, comes at virtually no cost to performance in general.

Interpolation and Approximation (connecting the dots)

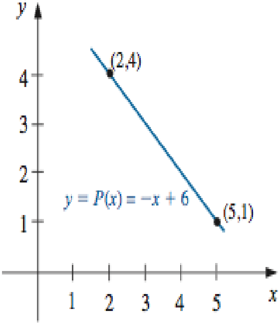
US population census data available every ten years.

Year	1950	1960	1970	1980	1990	2000
Population (in thousands)	151,326	179,323	203,302	226,542	249,633	281,422



- What was US population in year 1996?
- What will US population be in year 2017?

Connecting two dots



- Given two distinct dots, there is a unique line connecting them.

Connecting two dots

- Given two distinct points $(p_0, f(p_0))$ and $(p_1, f(p_1))$, there is a unique line connecting them

$$\begin{aligned} P(x) &= f(p_0) \cdot \frac{x - p_1}{p_0 - p_1} + f(p_1) \cdot \frac{x - p_0}{p_1 - p_0} \\ &\stackrel{def}{=} f(p_0) \cdot L_0(x) + f(p_1) \cdot L_1(x), \end{aligned}$$

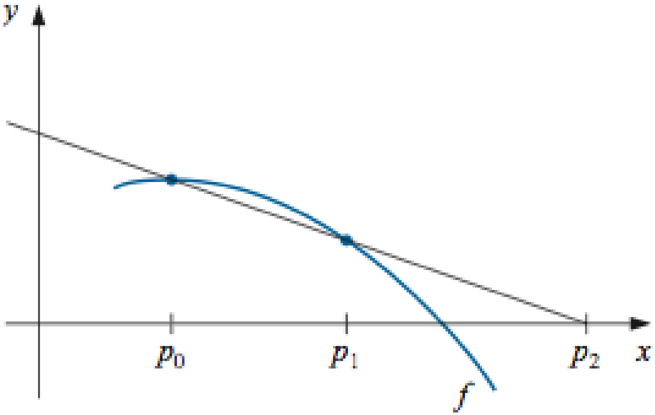
with $L_0(x), L_1(x)$ unrelated to $f(p_0), f(p_1)$.

$$\begin{aligned} L_0(p_0) = 1, &\quad L_0(p_1) = 0, \\ L_1(p_0) = 0, &\quad L_1(p_1) = 1. \end{aligned}$$

Connecting two dots

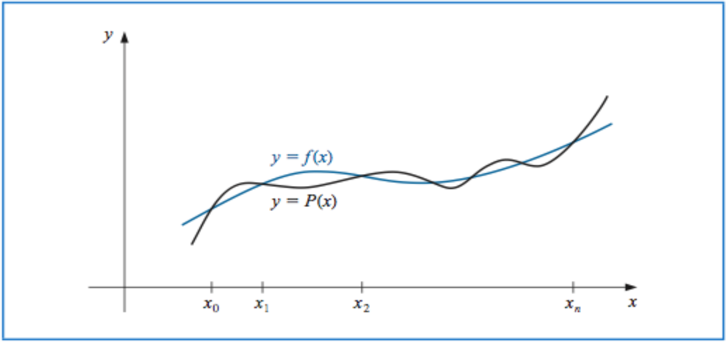
- Given two distinct points $(p_0, f(p_0))$ and $(p_1, f(p_1))$, there is a unique line connecting them

$$P(x) \quad = \quad f(p_0) \cdot \frac{x - p_1}{p_0 - p_1} + f(p_1) \cdot \frac{x - p_0}{p_1 - p_0}$$



Connecting $n+1$ dots

- Given $n+1$ distinct points
$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$
- Find a degree $\leq n$ polynomial
$$\begin{aligned} P(x) &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \\ \text{so that } P(x_0) &= f(x_0), P(x_1) = f(x_1), \cdots, P(x_n) = f(x_n). \end{aligned}$$



Connecting $n+1$ dots

- Would like to write $P(x)$ as

$$P(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x),$$

where $L_0(x), L_1(x), \cdots, L_n(x)$ are unrelated to $f(x_0), f(x_1), \cdots, f(x_n)$.

- at each node x_j, $j = 0, 1, \cdots, n$,

$$f(x_j) = P(x_j) = f(x_0)L_0(x_j) + f(x_1)L_1(x_j) + \cdots + f(x_n)L_n(x_j)$$

- This suggests for all i, j,

$$L_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

Largrangian Polynomials

$$\begin{aligned} P(x) &= f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x), \\ L_i(x_j) &= \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \end{aligned}$$

x_j is a root of $L_i(x)$ for every $j \neq i$, so

$$L_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Quadratic Interpolation

(a) Use the numbers (called *nodes*) $x_0 = 2$, $x_1 = 2.75$, and $x_2 = 4$ to find the second Lagrange interpolating polynomial for $f(x) = 1/x$.

(b) Use this polynomial to approximate $f(3) = 1/3$.

Quadratic Interpolation: Solution

$$L_0(x) = \frac{(x-2.75)(x-4)}{(2-2.5)(2-4)} = \frac{2}{3}(x-2.75)(x-4),$$

$$L_1(x) = \frac{(x-2)(x-4)}{(2.75-2)(2.75-4)} = -\frac{16}{15}(x-2)(x-4),$$

$$L_2(x) = \frac{(x-2)(x-2.75)}{(4-2)(4-2.5)} = \frac{2}{5}(x-2)(x-2.75).$$

Quadratic Interpolation: Solution

$$
\begin{aligned}
P(x) &= \sum_{k=0}^{2} f(x_k) L_k(x) \\
&= \frac{1}{3}(x-2.75)(x-4) - \frac{64}{165}(x-2)(x-4) + \frac{1}{10}(x-2)(x-2.75) \\
&= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}.
\end{aligned}
$$

Quadratic Interpolation: Solution

$$f(3) \approx P(3) = \frac{9}{22} - \frac{105}{88} + \frac{49}{44} = \frac{29}{88} \approx 0.32955.$$

Quadratic Interpolation: Solution