Lecture 7

Finding the Roots of $f(x) = 0$: Fixed Point, Bisection, Newton

L. Olson

Department of Computer Science
University of Illinois at Urbana-Champaign

Slides based on NMM slides from Recktenwald

February 6, 2006

Root-Finding Algorithms

Recall, once we find an initial guess (`brackPlot.m`) we can use a root-finding algorithm:

- Fixed point iteration
- Bisection
- Newton's method
- Secant method

Fixed Point Iteration

To solve

$$f(x) = 0$$

we rewrite as

$$x_{\text{new}} = g(x_{\text{old}})$$

Listing 1: Fixed Point Iteration

```
initialize:  x_0 = ...
for k = 1,2,...
  x_k = g(x_{k-1})
  if converged, stop
end
```

Fixed Point Iteration Example (1)

To solve

$$x - x^{1/3} - 2 = 0$$

rewrite as

$$x_{\text{new}} = g_1(x_{\text{old}}) = x_{\text{old}}^{1/3} + 2$$

or

$$x_{\text{new}} = g_2(x_{\text{old}}) = \left(x_{\text{old}} - 2\right)^3$$

or

$$x_{\text{new}} = g_3(x_{\text{old}}) = \frac{6 + 2x_{\text{old}}^{1/3}}{3 - x_{\text{old}}^{2/3}}$$

Are these $g(x)$ functions equally effective?

Fixed Point Iteration Example (2)

$$g_1(x) = x^{1/3} + 2$$
$$g_2(x) = (x-2)^3$$
$$g_3(x) = \frac{6 + 2x^{1/3}}{3 - x^{2/3}}$$

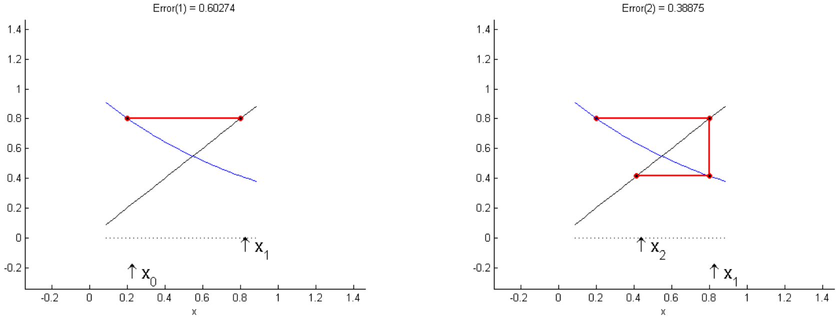
k	$g_1(x_{k-1})$	$g_2(x_{k-1})$	$g_3(x_{k-1})$
0	3	3	3
1	3.4422495703	1	3.5266442931
2	3.5098974493	-1	3.5213801474
3	3.5197243050	-27	3.5213797068
4	3.5211412691	-24389	3.5213797068
5	3.5213453678	-1.451×10^{13}	3.5213797068
6	3.5213747615	-3.055×10^{39}	3.5213797068
7	3.5213789946	-2.852×10^{118}	3.5213797068
8	3.5213796042	∞	3.5213797068
9	3.5213796920	∞	3.5213797068

Summary

$g_1(x)$ converges, $g_2(x)$ diverges, $g_3(x)$ converges very quickly

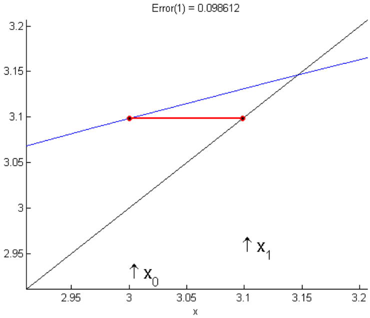
Fixed Point

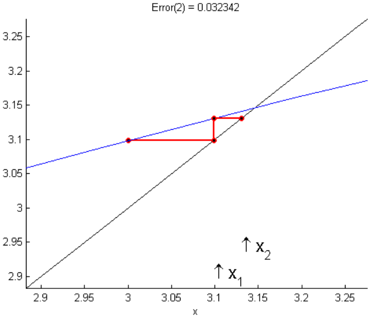
$g(x) = 3^{-x}$



Fixed Point

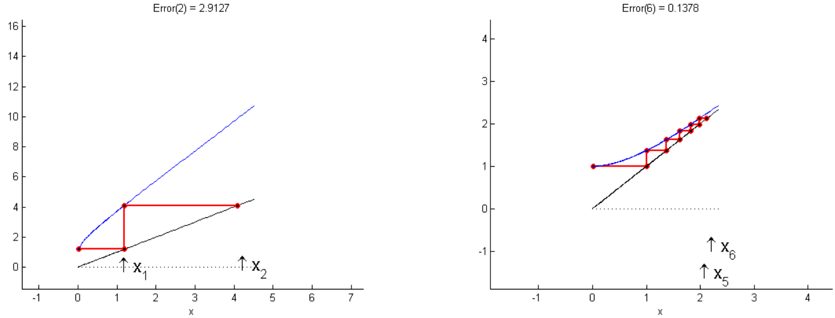
$g(x) = \log(x) + 2$





Fixed Point

$g(x) = e^{x^{0.4}} + x \qquad g(x) = e^{-x} + x$



Fixed Point

```
fixedpointplot.m
```

Fixed Point Theorem

Theorem

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

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

Then, for any guess $x_0 \in [a,b]$, the sequence defined by

$$x_n = g(x_{n-1})$$

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

Fixed Point Theorem: in words

Assuming $g(x)$ is smooth and bounded (in $[a, b]$) and that it's not too steep ($|g'(x)| \leqslant 1$), then we converge to a fixed point.

- in fact, k is associated with the rate of convergence:

$$|x_n - x| \leqslant k^n \max\{x_0 - a, b - x_0\}$$

and

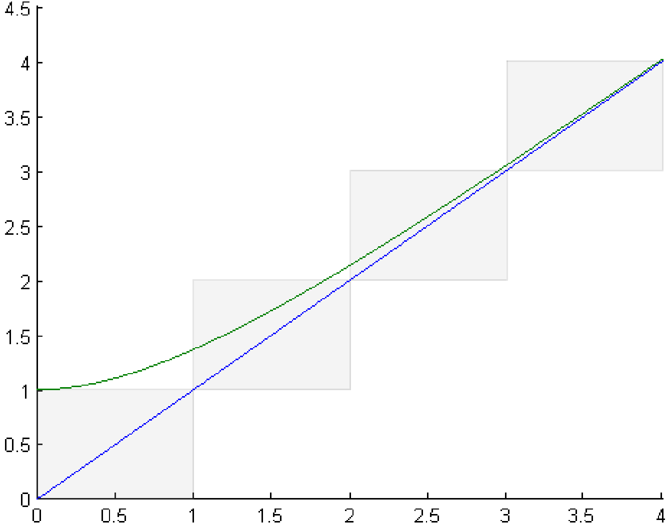
$$|x_n - x| \leqslant \frac{k^n}{1-k}|x_1 - x_0|$$

- so convergence is fast if $k \approx 0$
- and convergence is slow if $k \approx 1$

Fixed Point Theorem: cases

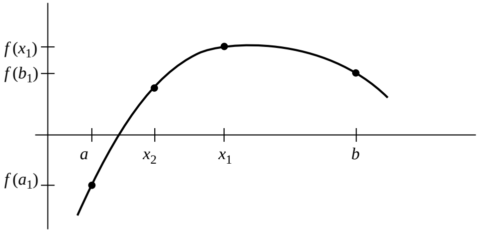
- $g(x) = 3^{-x}$ gives $g'(x) = -3^{-x}\log(3)$ which can be bounded arbitrarily small
- $g(x) = log(x) + 2$ gives $g'(x) = \frac{1}{x}$ which can be bounded arbitrarily small.
- $g(x) = e^{x^{0.4}} + x$ gives $g'(x) = e^{x^{0.4}}\frac{0.4}{x^{0.6}} + 1$ which is always greater than one
- $g(x) = e^{-x} + x$ gives $g'(x) = -e^{-x} + 1$ which is always less than one, but $g(x)$ is not in $[a, b]$

Fixed Point Theorem: $g(x) = exp(-x) + x$



Bisection

Given a bracketed root, halve the interval while continuing to bracket the root



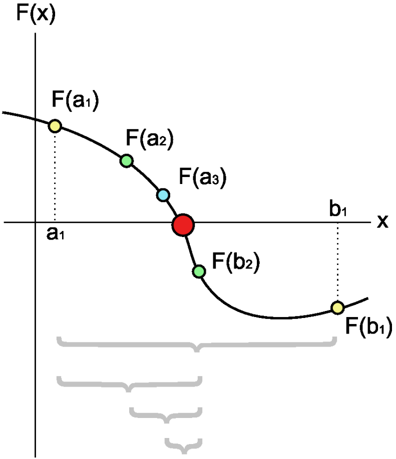
Bisection (2)

For the bracket interval $[a, b]$ the midpoint is

$$x_m = \frac{1}{2}(a + b)$$

idea:

1. split bracket in half
2. select the bracket that has the root
3. goto step 1



Bisection Algorithm

Listing 2: Bisection

```
initialize:  a = ..., b = ...
for k = 1,2,...
   x_m = a + (b - a)/2
   if sign(f(x_m)) = sign(f(x_a))
      a = x_m
   else
      b = x_m
   end
   if converged, stop
end
```

Bisection Example

Solve with bisection:

$$x - x^{1/3} - 2 = 0$$

k	a	b	x_{mid}	$f(x_{mid})$
0	3	4		
1	3	4	3.5	-0.01829449
2	3.5	4	3.75	0.19638375
3	3.5	3.75	3.625	0.08884159
4	3.5	3.625	3.5625	0.03522131
5	3.5	3.5625	3.53125	0.00845016
6	3.5	3.53125	3.515625	-0.00492550
7	3.51625	3.53125	3.5234375	0.00176150
8	3.51625	3.5234375	3.51953125	-0.00158221
9	3.51953125	3.5234375	3.52148438	0.00008959
10	3.51953125	3.52148438	3.52050781	-0.00074632

Analysis of Bisection (1)

Let δ_n be the size of the bracketing interval at the n^{th} stage of bisection. Then

$$\begin{aligned}
\delta_0 &= b - a = \text{initial bracketing interval} \\
\delta_1 &= \frac{1}{2}\delta_0 \\
\delta_2 &= \frac{1}{2}\delta_1 = \frac{1}{4}\delta_0 \\
&\vdots \\
\delta_n &= \left(\frac{1}{2}\right)^n \delta_0
\end{aligned}$$

$$\Longrightarrow \qquad \frac{\delta_n}{\delta_0} = \left(\frac{1}{2}\right)^n = 2^{-n}$$

or

$$n = \log_2\left(\frac{\delta_n}{\delta_0}\right)$$

Analysis of Bisection (2)

$$\frac{\delta_n}{\delta_0} = \left(\frac{1}{2}\right)^n = 2^{-n} \qquad \text{or} \qquad n = \log_2\left(\frac{\delta_n}{\delta_0}\right)$$

n	$\frac{\delta_n}{\delta_0}$	function evaluations
5	3.1×10^{-2}	7
10	9.8×10^{-4}	12
20	9.5×10^{-7}	22
30	9.3×10^{-10}	32
40	9.1×10^{-13}	42
50	8.9×10^{-16}	52

Convergence Criteria

An automatic root-finding procedure needs to monitor progress toward the root and stop when current guess is close enough to the desired root.

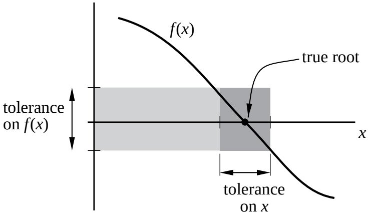
- Convergence checking will avoid searching to unnecessary accuracy.
- Check how closeness of successive approximations

$$|x_k - x_{k-1}| < \delta_x$$

- Check how close $f(x)$ is to zero at the current guess.

$$|f(x_k)| < \delta_f$$

Convergence Criteria on x

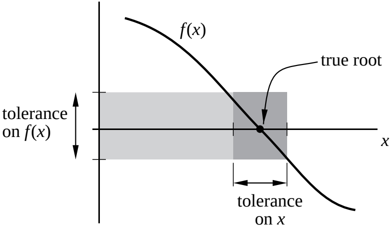


$$x_k = \text{current guess at the root}$$
$$x_{k-1} = \text{previous guess at the root}$$

Absolute tolerance: $\left|x_k - x_{k-1}\right| < \delta_x$

Relative tolerance: $\left|\dfrac{x_k - x_{k-1}}{b - a}\right| < \hat{\delta}_x$

Convergence Criteria on $f(x)$



Absolute tolerance: $\left|f(x_k)\right| < \delta_f$

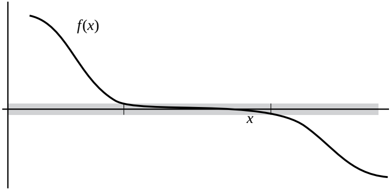
Relative tolerance:

$$|f(x_k)| < \hat{\delta}_f \max\left\{|f(a_0)|, |f(b_0)|\right\}$$

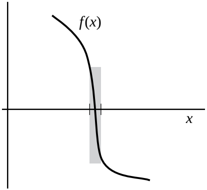
where a_0 and b_0 are the original brackets

Convergence Criteria on $f(x)$

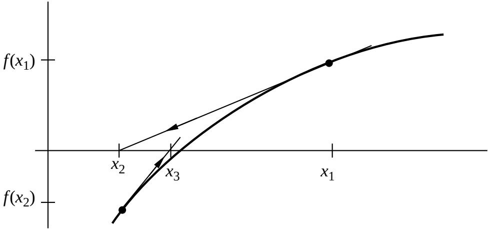
If $f'(x)$ is small near the root, it is easy to satisfy tolerance on $f(x)$ for a large range of Δx. The tolerance on Δx is more conservative



If $f'(x)$ is large near the root, it is possible to satisfy the tolerance on Δx when $|f(x)|$ is still large. The tolerance on $f(x)$ is more conservative



Newton's Method (1)



For a current guess x_k, use $f(x_k)$ and the slope $f'(x_k)$ to predict where $f(x)$ crosses the x axis.

Newton's Method (2)

Expand $f(x)$ in Taylor Series around x_k

$$f(x_k + \Delta x) = f(x_k) + \Delta x \left.\frac{df}{dx}\right|_{x_k} + \frac{(\Delta x)^2}{2} \left.\frac{d^2 f}{dx^2}\right|_{x_k} + \dots$$

Substitute $\Delta x = x_{k+1} - x_k$
and neglect 2^{nd} order terms to get

$$f(x_{k+1}) \approx f(x_k) + (x_{k+1} - x_k) f'(x_k)$$

where

$$f'(x_k) = \left.\frac{df}{dx}\right|_{x_k}$$

Newton's Method (3)

Goal is to find x such that $f(x) = 0$.
Set $f(x_{k+1}) = 0$ and solve for x_{k+1}

$$0 = f(x_k) + (x_{k+1} - x_k) f'(x_k)$$

or, solving for x_{k+1}

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}}$$

Newton's Method Algorithm

```
initialize:  x_1 = ...
for k = 2,3,...
   x_k = x_{k-1} - f(x_{k-1})/f'(x_{k-1})
   if converged, stop
end
```

Newton's Method Example (1)

Solve:

$$x - x^{1/3} - 2 = 0$$

First derivative is

$$f'(x) = 1 - \frac{1}{3}x^{-2/3}$$

The iteration formula is

$$x_{k+1} = x_k - \frac{x_k - x_k^{1/3} - 2}{1 - \frac{1}{3}x_k^{-2/3}}$$

Newton's Method Example (2)

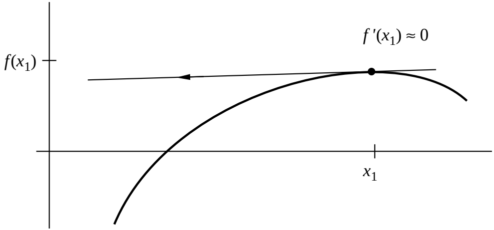
$$x_{k+1} = x_k - \frac{x_k - x_k^{1/3} - 2}{1 - \frac{1}{3}x_k^{-2/3}}$$

k	x_k	$f'(x_k)$	$f(x)$
0	3	0.83975005	-0.44224957
1	3.52664429	0.85612976	0.00450679
2	3.52138015	0.85598641	3.771×10^{-7}
3	3.52137971	0.85598640	2.664×10^{-15}
4	3.52137971	0.85598640	0.0

Conclusion

- Newton's method converges *much* more quickly than bisection
- Newton's method requires an analytical formula for $f'(x)$
- The algorithm is simple as long as $f'(x)$ is available.
- Iterations are not guaranteed to stay inside an ordinal bracket.

Divergence of Newton's Method



Since

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

the new guess, x_{k+1}, will be far from the old guess whenever $f'(x_k) \approx 0$