Solution of Nonlinear Equations

We would like to find the roots of equations of the form

$$f(x) = 0.$$

We will focus on the one-dimensional case (in which $f : [a,b] \to \mathbb{R}$).

In several dimensions one is often interested in solving several simultaneous equations. This can be hard.

We will look at several iterative methods:

Bisection

$$\text{related} \begin{cases} \text{Regula Falsi} \\ \text{Modified Regula Falsi} \\ \text{Secant Method} \end{cases}$$

Newton's Method

Other methods include Fixed Point Iteration and various hybrid methods (see NR:C).

Later we will discuss the special cases in which f is a polynomial or a linear function in higher dimensions.

Applications

- □ Intersection/collision detection
- □ Optimization (we'll see the connection later)

Bisection

Idea: Find two points a_0, b_0 such that $f(a_0)$ & $f(b_0)$ have opposite signs. If f is well-behaved, then it will have a root inbetween a_0 & b_0. Now, halve the interval $[a_0, b_0]$ while still bracketing the root, and repeat.

Formally: Start with $f(a_0) f(b_0) \le 0$

For $n = 0, 1, 2, \ldots$ until satisfied do:

$c \leftarrow \frac{1}{2}(a_n + b_n)$

If $f(a_n) f(c_n) \le 0$ then set $a_{n+1} \leftarrow a_n$; $b_{n+1} \leftarrow c$

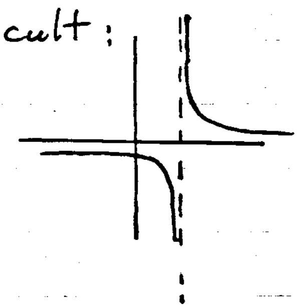
else set $a_{n+1} \leftarrow c$; $b_{n+1} \leftarrow b_n$

Comments:

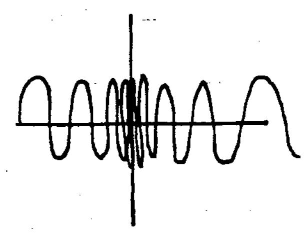
- The first part of the idea is critical to many root-finding techniques, namely to find an interval that brackets a root of f.

  This can be difficult:

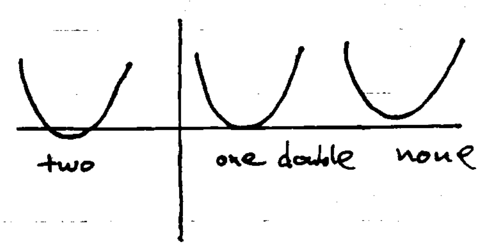
  Singularities

  Many roots

  Double roots

- Bisection can be slow, but it is simple and robust. It is therefore sometimes used as a "failsafe" backup for more complicated algorithms.

Note:

The pseudo-code

"If $f(a_n)f(c_n) \le 0$..."

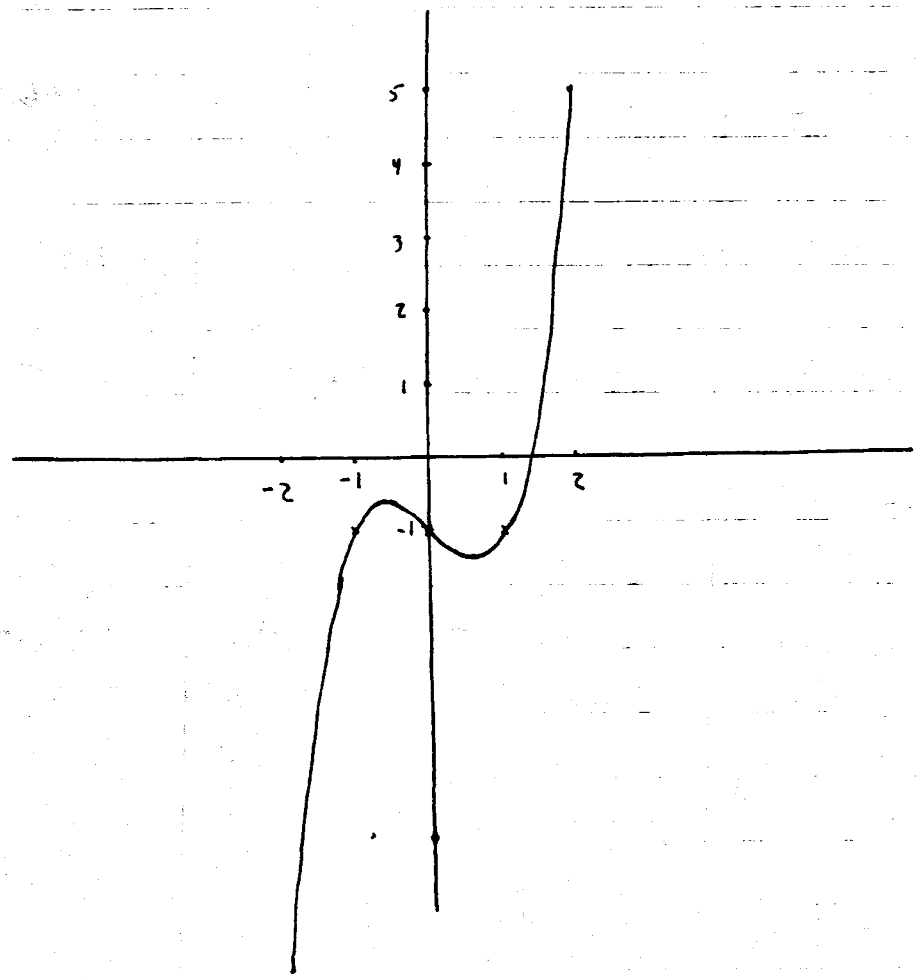
Is just shorthand for

"If $f(a_n)$ and $f(c_n)$ have opposite sign (or one is zero) ..."

Multiplication is probably not the best way to implement this test.

Example

$$f(x) = x^3 - x - 1$$



Note: Since f is a cubic it has either one zero or three zeros. (real zeros)

A quick local extrema computation shows that it can't have three zeros. So, we see that there must be a single zero, initially bracketed by $x=1$ & $x=2$.

Let's see if we can isolate this zero a bit more.

Using bisection

(to x-accuracy of $5\cdot10^{-5}$)

i	$[a_i, b_i]$	$f(a_i)$	$f(b_i)$	$\frac{a_i+b_i}{2}$	$f\left(\frac{a_i+b_i}{2}\right)$
0	[1, 2]	-1	5	1.5	.875
1	[1, 1.5]	-1	.875	1.25	$-.296875$
2	[1.25, 1.5]	$-.296875$	.875	1.375	.22460937
3	[1.25, 1.375]	$-.296875$	.22460937	1.3125	$-.05151367$
4	[1.3125, 1.375]	$-.051513672$	.22460937		
⋮					
14	[1.324707, 1.3247681]	$-4.659\cdot10^{-5}$	$2.137\cdot10^{-4}$	1.3247375	$8.355\cdot10^{-5}$
15	[1.324707, 1.3247375]	$-4.659\cdot10^{-5}$	$8.355\cdot10^{-5}$	1.3247223	$1.848\cdot10^{-5}$

$x \approx \underbrace{1.3247}_{\text{significant digits}}223$

interval width $\approx 3\cdot10^{-5}$

In each step of the bisection method the length of the bracketing interval is halved.
Hence each step produces one more correct binary digit (i.e., bit) in the approximation to the root.

In other words, the max error ε_n satisfies

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} = \frac{1}{2}.$$

This is known as linear convergence (since ε_{n+1} depends linearly on ε_n).

Locally, it often makes sense to assume that a function is linear (say for analytic or smooth functions).

This idea can be used to improve convergence.

Regula Falsi

Instead of using the midpoint of the bracketing interval to select a new root estimate, use a weighted average.

(*) $$w = \text{new estimate of root} = \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$$

In all other respects the bracketing & subdivision proceeds as with bisection.

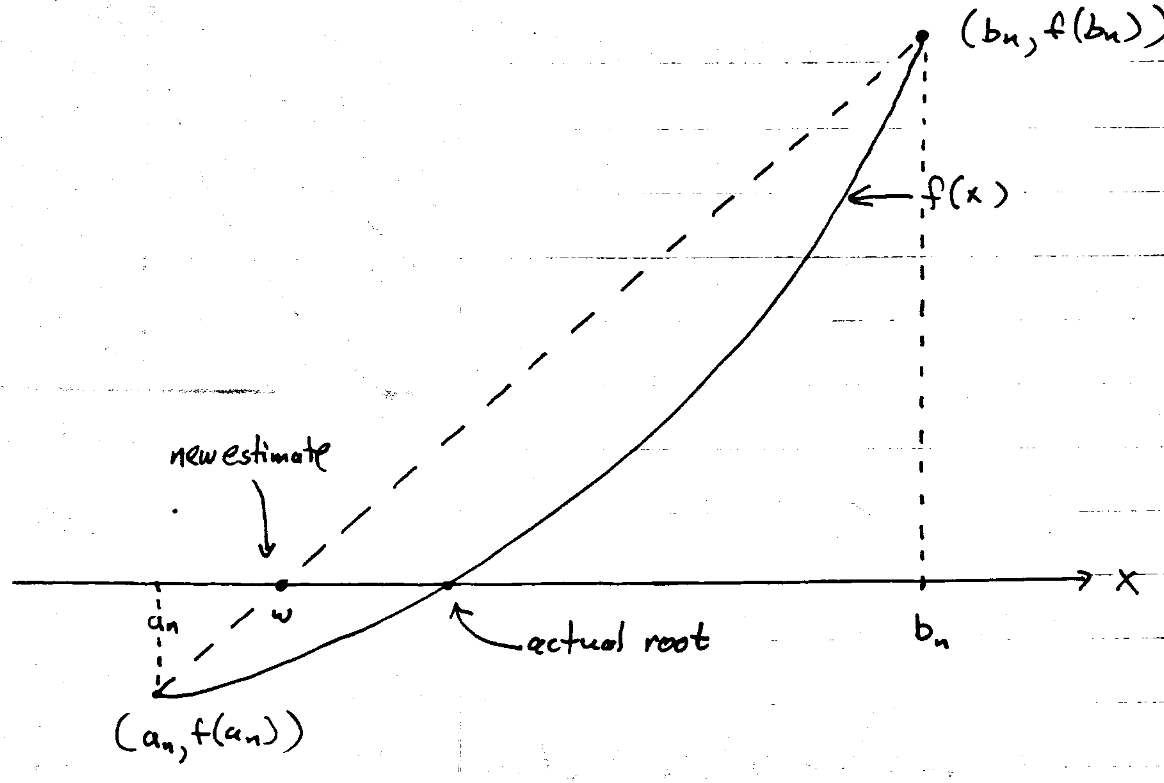
Note: w is just the weighted average of a_n & b_n, with weights $|f(b_n)|$ and $|f(a_n)|$, that is,

(**) $$w = \frac{|f(b_n)|}{|f(b_n)| + |f(a_n)|}\, a_n + \frac{|f(a_n)|}{|f(b_n)| + |f(a_n)|}\, b_n$$

(To verify that (*) & (**) are the same, recall that $f(b_n)$ and $f(a_n)$ have opposite sign.)

In particular, if $|f(b_n)|$ is larger than $|f(a_n)|$ then the new root estimate w is closer to a_n than to b_n.

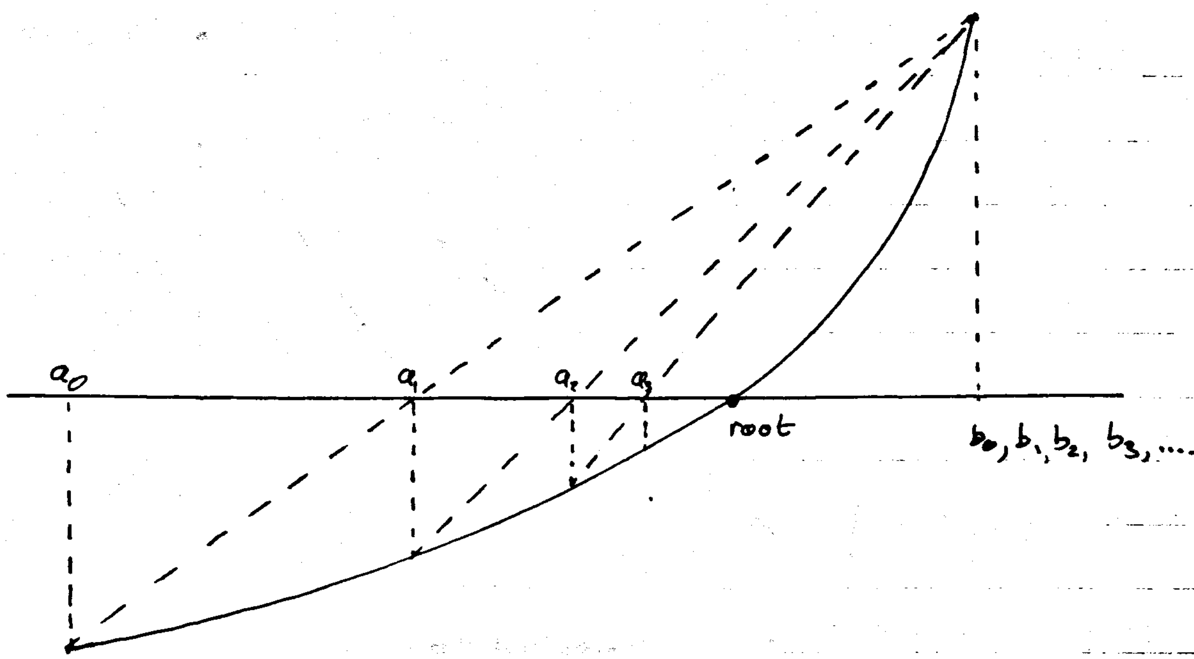
Picture



Observe that the weighted average w is the intersection with the x-axis of the line through the points $(a_n, f(a_n))$ and $(b_n, f(b_n))$.

(Why? Because $f(b_n)(w-a_n) = -f(a_n)(b_n-w)$ by (*) or (**). So we have two similar triangles that meet at a vertex. Etc.
Or, compute directly.)

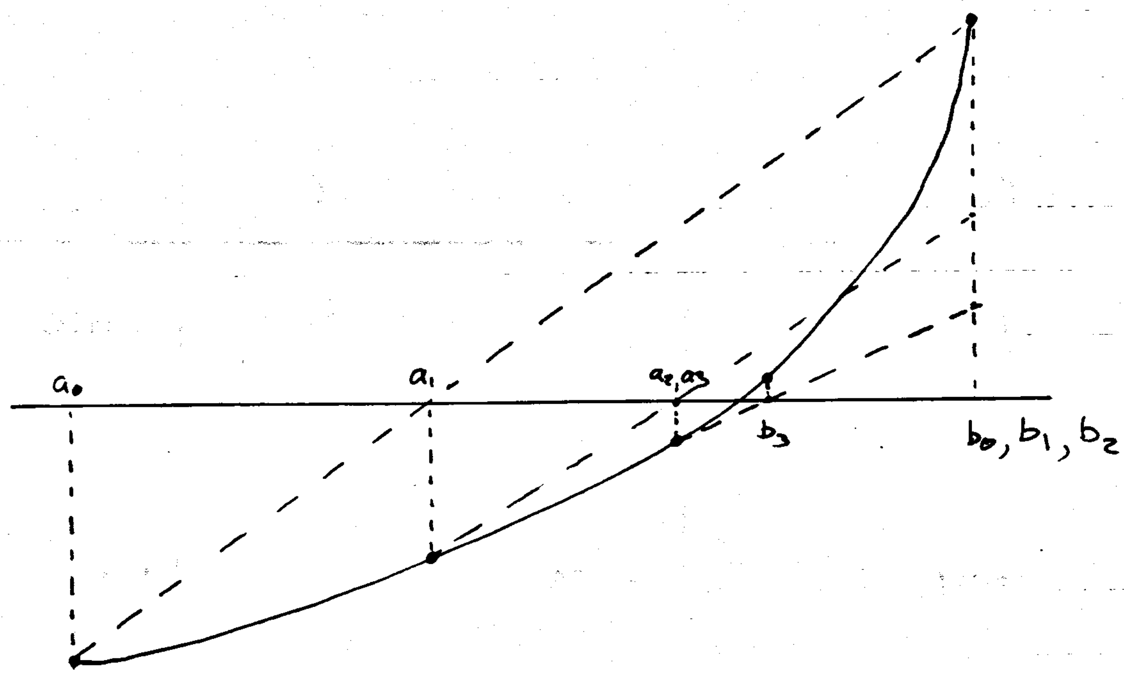
One bug: Convergence is one-sided with convex functions:



the new estimate always lies to the same side of the root. As a result, the bracketing interval only shrinks slowly (e.g., it is always of the form $[a_i, b_0]$).

One trick for avoiding this nastiness, is simply to repeatedly decrease the "apparent value of $|f|$" at the unchanging endpoint, until the root estimate switches sides.

E.g.:



In pseudo-code, we get: <u>Modified Regula Falsi</u>

$F \leftarrow f(a_0)$; $G \leftarrow f(b_0)$; $w_0 \leftarrow a_0$

For $n = 0, 1, 2, \ldots$ until termination conditions do:

$$w_{n+1} \leftarrow \frac{G a_n - F b_n}{G - F}$$

If $f(a_n) f(w_{n+1}) \le 0$

Then $a_{n+1} \leftarrow a_n$; $b_{n+1} \leftarrow w_{n+1}$; $G \leftarrow f(w_{n+1})$;

If $f(w_n) f(w_{n+1}) > 0$ then $F \leftarrow \frac{1}{2} F$;

Else $a_{n+1} \leftarrow w_{n+1}$; $b_{n+1} \leftarrow b_n$; $F \leftarrow f(w_{n+1})$;

If $f(w_n) f(w_{n+1}) > 0$ then $G \leftarrow \frac{1}{2} G$;

Observe that the interval $[a_{n+1}, b_{n+1}]$ brackets a root of f, given that $[a_n, b_n]$ brackets a root and f is well-behaved.

Back to our numerical example

i	$[a_i, b_i]$	F	G	w_i	$f(w_i)$
0	$[1, 2]$	-1	5	$1.166\overline{6}$	$-.5787$
1	$[1.166\overline{6}, 2]$	$-.5787$	5	1.2531	$-.2854$
2	$[1.2531, 2]$	$-.2854$	$\downarrow \times\frac{1}{2}$ 2.5	1.3296	$.02105$
3	$[1.2531, 1.3296]$	$-.2854$	$.02105$	1.3244	$-.00146$
4	$[1.3244, 1.3296]$	$-.00146$	$.02105$	1.3247	$-6.7\cdot10^{-6}$
6	$[1.3247164, 1.3247195]$	$-6.7\cdot10^{-6}$	$6.6\cdot10^{-6}$	1.32471796	$-9\cdot10^{-12}$

(I only showed all the digits at the last steps. — You should try this example with your code.)

$$x \approx \underbrace{1.32471}_{\text{significant digits}}8$$

$$\text{interval width} \approx 3.1\cdot10^{-6}$$

Note the slightly faster convergence than with bisection (6 vs 15 steps).

Termination Conditions

Unlike bisection, the root bracketing of the Modified Regula Falsi method may not give a small interval of convergence.

Consider $f(x) = (x - 1.3333333333)^3$ for instance, assuming a computer with 11 or fewer digits of accuracy.

In general, a routine will want to terminate on one of several conditions, say:

a) $|x_{n+1} - x_n|$ is "small"

b) $|f(x_n)|$ is "small"

c) n is "large"

N.B.: One may wish to measure (a) & (b) as relative errors, say:

a) $|x_{n+1} - x_n| \leq XTOL * |x_n|$

b) $|f(x_n)| \leq FTOL * F$

where F is an estimate of the magnitude of $f(x)$ near the root and XTOL & FTOL are some "tolerances".

Secant Method

This method is a modification of the Regula Falsi method which gives up root bracketing. The method need not converge, but when it converges, it converges quickly. However, because there is no root bracketing, one may not have a good idea how far from the true root the final estimate is.

Algorithm: x_{-1} & x_0 are input as data.

For $n = 0, 1, \ldots$ until termination conditions do:

$$x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})}$$

To reduce roundoff error use:

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Example again:

i	x_i	$f(x_i)$
-1	1	-1
0	2	5
1	$1.1666\overline{6}$	$-.5787$
2	1.253112	$-.28536$
3	1.337206	$.05388$
4	1.32385	$-.003698$
5	1.3247079	$-4.27 \cdot 10^{-5}$
6	$\underbrace{1.3247}_{}179$	$3.458 \cdot 10^{-8}$

"significant digits"
"accuracy" $\approx 1 \cdot 10^{-5}$ } based only on changes in x_n.

Newton's Method (Newton-Raphson)

In the Secant method we can write $x_{n+1} = \frac{f(x_n)}{f[x_n, x_{n-1}]}$

$f[x_n, x_{n-1}]$ is a first difference approximation to the first derivative of f. Passing to the continuous case, this suggest the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

This is Newton's method.
It requires knowledge of f'.
As with the Secant method, convergence is not guaranteed. For instance, if f' is near zero, the method can shoot off to infinity. Another bug is that the method can cycle. — However, when it converges, it converges very quickly. For this reason, one often uses the Newton or Secant methods to improve ("polish") an already good estimate of a root of f that one has obtained by other means.

Back to our example: $f(x) = x^3 - x - 1$, so $f'(x) = 3x^2 - 1$

Two different runs:

i	x_i	$f(x_i)$
0	1	-1
1	1.5	0.875
2	1.347826	0.10068
3	1.325200	0.002058
4	1.324718	$9.2 \cdot 10^{-7}$
5	1.324718	$1.86 \cdot 10^{-13}$

i	x_i	$f(x_i)$
0	2	5
1	1.54545	1.14576
2	1.35965	0.1537
3	1.325801	0.00462
4	1.324718	$4.65 \cdot 10^{-6}$
5	1.324718	$4.7 \cdot 10^{-12}$

Inverse Quadratic Interpolation for root finding
(cf NRiC pp. 267 ff.)

Given data a, b, c & $f(a), f(b), f(c)$,
fit a quadratic polynomial in y, i.e., $x = p_2(y)$.
Now interpolate at $y = 0$ to generate a new estimate of the root.

Using our old interpolation results, let's derive the formula.
Let's use:

$$y_0 = f(b) \quad \text{so} \quad f^{-1}(y_0) = b$$
$$y_1 = f(a) \quad \text{so} \quad f^{-1}(y_1) = a$$
$$y_2 = f(c) \quad \text{so} \quad f^{-1}(y_2) = c$$

Then
$$p_2(y) = f^{-1}[y_0] + f^{-1}[y_0, y_1](y - y_0) + f^{-1}[y_0, y_1, y_2] \cdot (y - y_0)(y - y_1)$$

So
$$p_2(y) = b + \frac{b-a}{f(b)-f(a)}\left(y - f(b)\right) + \frac{\frac{b-a}{f(b)-f(a)} - \frac{a-c}{f(a)-f(c)}}{f(b) - f(c)}\left(y - f(b)\right)\left(y - f(a)\right)$$

And
$$p_2(0) = b + \frac{b-a}{f(b)-f(a)} \cdot \left(-f(b)\right) + \frac{\frac{b-a}{f(b)-f(a)} - \frac{a-c}{f(a)-f(c)}}{f(b) - f(c)} f(b) f(a)$$

And so, our new root estimate is given by

$$x = b + \frac{P}{Q}$$

where P & Q are appropriate formulas.

This method works well near a root if $f(x)$ is well-behaved. However, the method can grow unstable if $Q \approx 0$.

One can guard against this by combining three ideas: root bracketing
bisection
inverse quadratic interpolation.

This combo comprises Brent's Method. (see NRiC)

Specifically, if the correction term $\frac{P}{Q}$ is too large, that is, if it falls outside of the current bracketing interval, then Brent's method simply performs a bisection step.
For that matter, even if $\frac{P}{Q}$ falls inside the bracketing interval, if the new estimate does not collapse the bracketing interval fast enough, then Brent's method executes a bisection step.

In short, Brent's method combines a fast higher-order method with a sure-fire method.

We said that Newton's method converges if f is well-behaved and if the initial guess is near the root. Just what does that mean? — Here is one answer.

Applicability Theorem

Let $f \in C^2([a,b])$, and suppose f satisfies the following conditions:

1) $f(a)f(b) < 0$
2) $f'(x) \neq 0$ for all $x \in [a,b]$
3) $f''(x)$ is either non-negative everywhere on $[a,b]$ or non-positive everywhere on $[a,b]$.
4) At the endpoints

$$\left|\frac{f(a)}{f'(a)}\right| < b-a \qquad \left|\frac{f(b)}{f'(b)}\right| < b-a$$

Then Newton's method converges to the unique root of $f(x)$ in $[a,b]$ for any initial guess $x_0 \in [a,b]$.

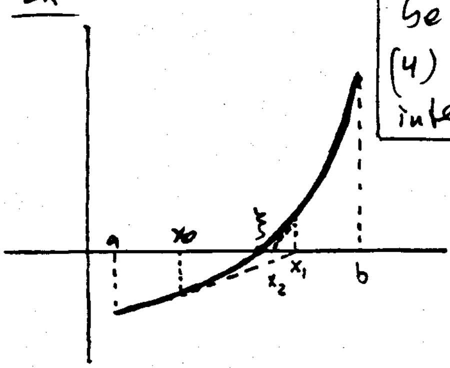
(Note: These same conditions ensure that the secant method converges.)

Comments: (1) & (2) ensure that there is exactly one zero in $[a,b]$

(3) implies that f is either concave from above or concave from below. (2) & (3) ensure that f' is monotone, pos or neg, on $[a,b]$.

(4) says that the tangent to the curve of f at either endpoint intersects the x-axis within the interval $[a,b]$.

Ex



In the example, $f''(x) \geq 0$, $f(a) < 0$, $f(b) > 0$.
ξ is the true root.
Once $x_i > \xi$ all $x_{i+k} > \xi$ and decrease monotonically to ξ. If $x_0 < \xi$, then $x_1 > \xi$.

Rates of Convergence

An important part of understanding and using numerical techniques is estimating errors and convergence. — In the case of bisection we saw that each step added one bit of accuracy, and we said that bisection had linear convergence. More generally, we have the following definition

Def. Let $x_0, x_1, \ldots$ be a sequence which converges to a number ξ. Let $\varepsilon_n = \xi - x_n$.
Suppose there exists a power p and a constant $C \neq 0$ such that

$$\lim_{n\to\infty} \frac{|\varepsilon_{n+1}|}{|\varepsilon_n|^p} = C.$$

Then p is called the order of convergence of the sequence and C is called the asymptotic error constant.

For instance, if $p = 2$, we say that the order of convergence is two, or that the sequence has quadratic convergence. Asymptotically,

$$|\varepsilon_{n+1}| \longrightarrow C|\varepsilon_n|^2$$ for some non-zero constant C,

In terms of bit accuracy:

$$\log_2|\varepsilon_{n+1}| \sim 2\log_2|\varepsilon_n| + \log_2 C$$

So, effectively the number of bits of accuracy doubles on each step. — That's why quadratic convergence is much faster than linear convergence.

Newton's Method – Quadratic Convergence

As an example, let's look at the error terms for Newton's method, and show that the method converges quadratically. A standard trick is to use Taylor's formula.

Let's look at the function $h(x) = \frac{f(x)}{f'(x)}$.

If we assume that f is sufficiently smooth and that $f'(x)$ is non-zero over some interval of interest, then h will also be sufficiently smooth for us to apply Taylor's formula. (By the way, these assumptions are reasonable given that we are looking at a region in which Newton's method converges.)

Let us expand $h(x)$ about the root ξ of f. Taylor tells us that:

$$h(\xi+\varepsilon) = h(\xi) + \varepsilon h'(\xi) + \frac{\varepsilon^2}{2} h''(\xi) + \cdots$$

for arbitrary ε in some neighborhood of zero

Now let's compute h' & h''

$$h'(x) = \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2}$$

$$= 1 - \frac{f(x)f''(x)}{[f'(x)]^2}$$

$$h''(x) = -\frac{[f'(x)]^3 f''(x) + [f'(x)]^2 f(x) f'''(x) - 2f'(x)f(x)[f''(x)]^2}{[f'(x)]^4}$$

$$= -\frac{[f'(x)]^2 f''(x) + f(x)f'(x)f'''(x) - 2f(x)[f''(x)]^2}{[f'(x)]^3}$$

So, $h'(\xi) = 1$

$$h''(\xi) = -\frac{f''(\xi)}{f'(\xi)}$$

and then

$$(*) \qquad h(\xi+\varepsilon) = \varepsilon - \frac{1}{2}\varepsilon^2 \frac{f''(\xi)}{f'(\xi)} + \cdots$$

Now let's look at Newton's method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Recall $\varepsilon_n = \xi - x_n$, so $x_n = \xi - \varepsilon_n$.
By $(*)$:

$$\varepsilon_{n+1} = \varepsilon_n + h(\xi - \varepsilon_n)$$

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

The previous formula establishes the quadratic convergence of Newton's method. Specifically,

$$|\varepsilon_{n+1}| \sim C \, |\varepsilon_n|^2$$

where $C = \frac{1}{2} \frac{f''(\xi)}{f'(\xi)}$

Notice how this result depends on a non-zero first derivative and a non-zero second derivative at the root ξ. Newton's method can still converge even if $f'(\xi) = 0$, but the rate of convergence will be slower. What if $f'(\xi) \neq 0$ but $f''(\xi) = 0$?

Secant Method

A more complicated argument shows that the order of convergence for the Secant method is

$$p = \frac{1}{2} + \frac{\sqrt{5}}{2} \doteq 1.618 \quad \text{(golden mean)}$$

Tradeoff: Newton's method has a slightly better convergence rate than the Secant method, but requires explicit evaluation of f'.

Systems of Equations — Higher-dimensional Zeros

It is generally difficult to solve several simultaneous non-linear equations. See §9.6 of NRiC. However, suppose one has a "pretty good" neighborhood estimate of a simultaneous root. Then the higher-dimensional analogue of Newton's method is useful:

Suppose we have a sufficiently smooth mapping $\vec{f}: \mathbb{R}^n \to \mathbb{R}^n$, written as $\vec{f}(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$.

Taylor says that for small $\Delta\vec{x}$ we can expand each of the f_i as:

$$f_i(\vec{x} + \Delta\vec{x}) = f_i(\vec{x}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} \Delta x_j + \cdots$$

where $\vec{x} \in \mathbb{R}^n$, $\Delta\vec{x} \in \mathbb{R}^n$

$\Delta\vec{x} = (\Delta x_1, \ldots, \Delta x_n)$

So, suppose our current best estimate of a root of $\vec{f}$ is $\vec{x}^{(k)}$. We wish to compute $\Delta\vec{x}$ such that $\vec{x}^{(k)} + \Delta\vec{x}$ is a root of $\vec{f}$. Then for each i:

$$0 = f_i(\vec{x}^{(k)}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} \Delta x_j + \cdots$$

So, we have an update rule: $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \Delta\vec{x}$, where $\Delta\vec{x}$ satisfies the matrix equation

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \Delta\vec{x} = - \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix}$$

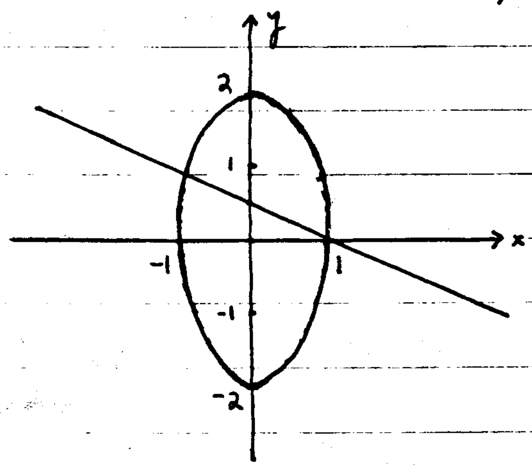
In other words, if $J(x^{(k)})$ is the Jacobian of f at $x = x^{(k)}$:

$$\boxed{\vec{x}^{(k+1)} = \vec{x}^{(k)} - J^{-1}(x^{(k)})\,\vec{f}(x^{(k)}).}$$

Example. Let's find the simultaneous zeros of the system

$$f(x,y) = 4x^2 + y^2 - 4$$
$$g(x,y) = 2y + x - 1$$

Now, of course, this is simple enough for us to solve explicitly. The simultaneous zeros are given by the intersection of a line with an ellipse:



ellipse: $x^2 + \left(\frac{y}{2}\right)^2 = 1$

line: $y = -\frac{1}{2}x + \frac{1}{2}$

Clearly the two intersection points are $(1,0)$ and $\left(-\frac{15}{17}, \frac{16}{17}\right)$.

Now, let's apply Newton-Raphson:

Observe that $\frac{\partial f}{\partial x} = 8x \qquad \frac{\partial f}{\partial y} = 2y$

$$\frac{\partial g}{\partial x} = 1 \qquad \frac{\partial g}{\partial y} = 2$$

So let's write the Jacobian as $J(x,y) = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x & 2y \\ 1 & 2 \end{pmatrix}$

Suppose our initial estimate of a root is right on target:

(i) $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Then we wish to solve $\begin{pmatrix} 8 & 0 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

so $\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

(ii) $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{-15}{17} \\ \frac{16}{17} \end{pmatrix}$

Then we wish to solve $\begin{pmatrix} \frac{-120}{17} & \frac{32}{17} \\ 1 & 2 \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

so $\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

In short, if we're at a zero, the method does indeed remain there.

Suppose now we start near one of the zeros, say at

$$\begin{pmatrix} x^{(0)} \\ y^{(0)} \end{pmatrix} = \begin{pmatrix} 1+\varepsilon \\ 0 \end{pmatrix} \quad \text{, with } |\varepsilon| \text{ small.}$$

Then we wish to solve

$$\begin{pmatrix} 8+8\varepsilon & 0 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} 8\varepsilon + 4\varepsilon^2 \\ \varepsilon \end{pmatrix}$$

so

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -\dfrac{\varepsilon + \frac{1}{2}\varepsilon^2}{1+\varepsilon} \\ -\dfrac{\frac{1}{4}\varepsilon^2}{1+\varepsilon} \end{pmatrix}$$

So, our next estimate is
$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} x^{(0)} \\ y^{(0)} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$
$$= \begin{pmatrix} \dfrac{1+\varepsilon+\frac{1}{2}\varepsilon^2}{1+\varepsilon} \\ \dfrac{-\frac{1}{4}\varepsilon^2}{1+\varepsilon} \end{pmatrix}$$

Observe that $g\left(x^{(1)}, y^{(1)}\right) = 0$

This is because Newton's method, being a derivative-based method, finds zeros of linear functions in a single step. Once on the line $g(x,y)=0$, the second row of the Jacobian ensures that the system remains on this line. Thus, from there on out, the search behaves like 1D Newton.

Let's investigate the convergence rate:

We can write
$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} 1 + \dfrac{\frac{1}{2}\varepsilon^2}{1+\varepsilon} \\ -\frac{1}{2}\cdot\dfrac{\frac{1}{2}\varepsilon^2}{1+\varepsilon} \end{pmatrix} = \begin{pmatrix} 1+\delta \\ -\frac{1}{2}\delta \end{pmatrix}$$
with δ small.

To compute our next estimate we must solve
$$\begin{pmatrix} 8+8\delta & -\delta \\ 1 & 2 \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\begin{pmatrix} 8\delta + \frac{17}{4}\delta^2 \\ 0 \end{pmatrix}$$

so
$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} -\dfrac{16\delta + \frac{17}{2}\delta^2}{16+17\delta} \\ \dfrac{8\delta + \frac{17}{4}\delta^2}{16+17\delta} \end{pmatrix}$$

Observe that $\Delta x = -2\Delta y$, as we previously noted.
Observe further that $\text{sign}(\Delta y) = \text{sign}(\delta) \cdot \text{sign}(8+\frac{17}{4}\delta) \cdot \text{sign}(16+17\delta)$

So, <u>if $\delta > 0$ (& hence $x^{(k)} > 1$)</u>, then Newton's method moves leftward up the line $y = -\frac{1}{2}x + \frac{1}{2}$ toward the root $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

<u>If $-\frac{16}{17} < \delta < 0$ (& hence $\frac{1}{17} < x^{(k)} < 1$)</u>, then Newton's method moves rightward down the line toward the root $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

<u>If $-\frac{32}{17} < \delta < -\frac{16}{17}$ (& hence $-\frac{15}{17} < x^{(k)} < \frac{1}{17}$)</u>, then Newton's method again moves leftward up the line, but this time toward the root $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{15}{17} \\ \frac{16}{17} \end{pmatrix}$.

<u>Finally, if $\delta < -\frac{32}{17}$ (& hence $x^{(k)} < -\frac{15}{17}$)</u>, then Newton's method converges on $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{15}{17} \\ \frac{16}{17} \end{pmatrix}$ from above.

In order to verify the rate of convergence, let's compute $\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix}$ from $\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix}$:

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$= \begin{pmatrix} 1 + \dfrac{\frac{17}{2}\delta^2}{16+17\delta} \\ \dfrac{-\frac{17}{4}\delta^2}{16+17\delta} \end{pmatrix}$$

Observe that we can write

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} 1+\hat{\delta} \\ -\frac{1}{2}\hat{\delta} \end{pmatrix}$$

which has the same form as the expression for $\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix}$
but

$$\hat{\delta} = \frac{\frac{17}{2}\delta^2}{16+17\delta}.$$

For small δ this says that $\hat{\delta}$ is approximately proportional to δ^2, which illustrated the quadratic convergence of Newton's method.

<u>One final note:</u> For some points (x,y), the Jacobian $J(x,y)$ is singular, so we can't solve for $\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$. The analogy in 1D is that $f'(x)=0$, so we can't compute $-\frac{f(x)}{f'(x)}$.

The set of points at which the Jacobian is singular is given by the line $y = 8x$.
This includes the origin and the point $\left(\frac{1}{17}, \frac{8}{17}\right)$.
(The latter point is a non-root at which the convergence direction in the analysis on p. 24 is undefined.)