Example

Consider $\sqrt{x^2+1}-1$ for small values of $|x|$.

Instead we can make computer compute

$$\left(\sqrt{x^2+1}-1\right)\left(\frac{\sqrt{x^2+1}+1}{\sqrt{x^2+1}+1}\right)=\frac{x^2+1-1}{\sqrt{x^2+1}+1}=\frac{x^2}{\sqrt{x^2+1}+1}$$

Theorem on Loss of Precision

If $x>y>0$ are machine numbers s.t.

$$2^{-q}\le 1-\frac{y}{x}\le 2^{-p}$$

$1-\frac{y}{x}=\frac{x}{x}-\frac{y}{x}=\frac{x-y}{x}$

then at most q and at least p significant bits are are lost in computing $x-y$.

Example

Consider $x-\sin x$ for small $|x|$.

$$x-\sin x = x-\left(x-\frac{x^3}{3!}+\frac{x^5}{5!}-\cdots\right)=\frac{x^3}{3!}-\frac{x^5}{5!}+\frac{x^7}{7!}-\cdots$$

$1-\frac{\sin x}{x}\ge\frac{1}{2}=2^{-1}$ for $1.9\le|x|$

So for $|x|\ge 1.9$, at most 1 significant bit is lost.

for $|x|\le 1.9$ using Taylor series approx. with 7 terms gives an $|\text{error}|\le 10^{-9}$

Chp 3. Solution of non-linear equations

Q) How to find x satisfying $g(x)=h(x)$?

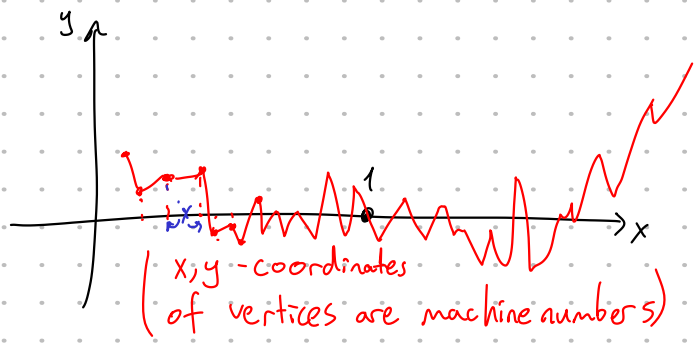
(or equivalently) find zeros of $f(x)$ (where $f(x)=g(x)-h(x)$)?

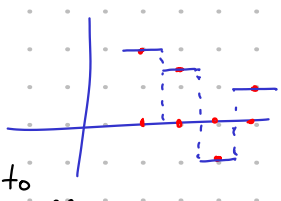
If $f(x)$ is a polynomial, zeros of f = roots of f.

Consider $f(x)=^{\circledast} x^4-4x^3+6x^2-4x+1$

Note that $f(x)=(x-1)^4$ so $f(1)=0$ is the only zero.

If we use $\circledast$ to compute f near 1 with a computer we get



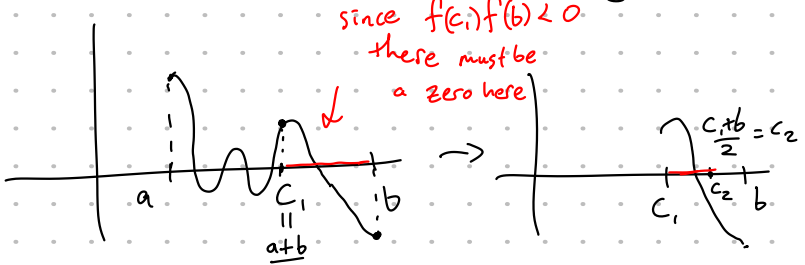


due to round off errors

To a computer it seems like there are many sign changes (and therefore 0s) near 1.

All 3.1 Bisection (Interval Halving) Method

If $f:[a,b]\to\mathbb{R}$ is cont. and $f(a)f(b)<0$, then $\exists\, r\in[a,b]$ s.t. $f(r)=0$ by the IVT.



$f(b)<0<f(a)$ $\quad 0<f(c_1)$ $\quad f(b)<0<f(c_1)$ $\quad f(c_2)<0$

After each step, the interval which contains r shrinks by a factor of 2.

We cannot expect r to be an endpoint of one of those intervals or even that $f(f(r))=0$.

So when do we stop?

We stop when at least one of the 3 conditions is true:

1) Number of steps exceeds a threshold value M
2) The error $|r-c_n|<\delta$ where δ is an acceptable threshold
3) The value $|f(c_n)|<\varepsilon$ where ε is an acceptable threshold.

Rmk 1: Actually we compute midpoint of the interval $[a,b]$ as $c=a+\frac{b-a}{2}$.

It is possible to come up with a,b s.t. $\frac{a+b}{2}\notin[a,b]$ when evaluated using finitely many digits on specific systems:

e.g. In a 3 digit chopped decimal system, set

$a=0.981$ $\quad b=0.983$

then $a+b=1.964$ and $\frac{a+b}{2}=0.980$.

The general principle is

new approx. = old approx. + small correction.

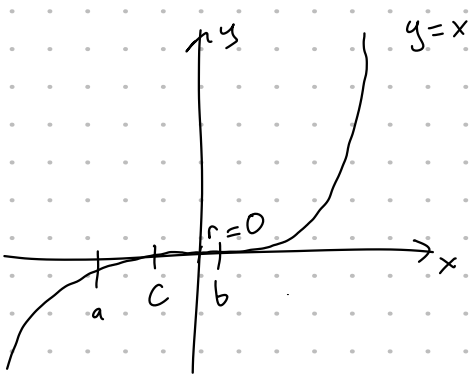
Rmk 2: Instead of checking $f(a)f(b) < 0$,

check $\text{sign}(f(a)) \neq \text{sign}(f(b))$

$10^{-50} \cdot (-10^{-50})$

$= -10^{-100} \simeq 0$

Failures:



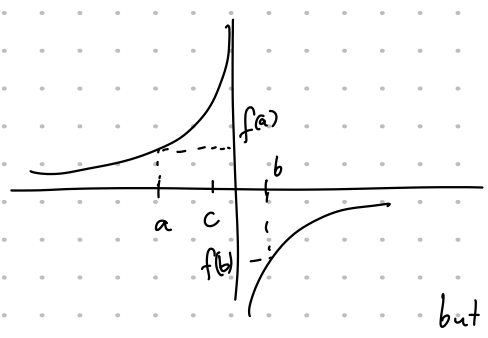
$y = x^3$ (function is very close to 0 near r)

e.g. when $f'(r) = 0$

$|f(c)| < \varepsilon$

but $|r - c| > \delta$

r may not be determined to a high precision



$|b-a| < \delta$ but $|f(c)| > \varepsilon$

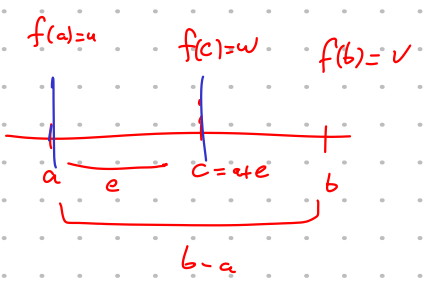
Of course here f is not continuous but in general it might be hard to verify that.

Pseudo code:

input $f, a, b, M, \delta, \varepsilon$

$u \leftarrow f(a)$
$v \leftarrow f(b)$
$e \leftarrow b - a$

output a, b, u, v
if $\text{sign}(u) = \text{sign}(v)$ then stop
for $k = 1$ to M do
  $e \leftarrow e/2$
  $c \leftarrow a + e$
  $w \leftarrow f(c)$
  output k, c, w, e
  if $|e| < \delta$ or $|w| < \varepsilon$ then stop
  if $\text{sign}(w) \neq \text{sign}(u)$ then
    $b \leftarrow c$
    $v \leftarrow w$
  else
    $a \leftarrow c$
    $u \leftarrow w$
  end if
end do



Error Analysis

Let the intervals be $[a_0, b_0], [a_1, b_1], \ldots$

then $a_0 \leq a_1 \leq a_2 \leq \cdots \leq b_0$

$a_0 \leq \cdots \leq b_2 \leq b_1 \leq b_0$

and $b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n)$.

Thus, a_n and b_n converges and $b_n - a_n = 2^{-n}(b_0 - a_0)$

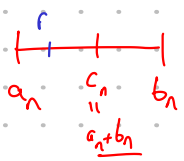
$$\lim_{n\to\infty} b_n - \lim_{n\to\infty} a_n = \lim_{n\to\infty} 2^{-n}(b_0 - a_0) = 0$$

So $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n =: r$

Then $0 \geq f(a_n) f(b_n) \Rightarrow 0 \geq \lim_{n\to\infty} f(a_n) f(b_n) = (f(r))^2$

and $f(r) = 0$.

If we stop the process at $[a_n, b_n]$ then the best approx is the mid point $c_n = \frac{a_n + b_n}{2}$ and



$$|\text{error}| = |r - c_n| \leq \frac{1}{2}(b_n - a_n) = 2^{-(n+1)}(b_0 - a_0)$$

Theorem on Bisection Method

$|\text{error}| = |r - c_n| \leq 2^{-(n+1)}(b_0 - a_0)$

Example Suppose we use the bisection method on $[50, 63]$. How many steps should be taken to compute a root with relative accuracy of 10^{-12}?

We want

$$\frac{|r - c_n|}{|r|} \leq 10^{-12} \quad \text{since} \quad 50 \leq r \leq 63$$

$$\frac{1}{|r|} \leq \frac{1}{50} \quad \text{so} \quad \frac{|r - c_n|}{|r|} \leq \frac{|r - c_n|}{50} \leq 10^{-12} \quad \text{(we want)}$$

or $|r - c_n| \leq 5 \cdot 10^{-11}$

By the above thm, $|r - c_n| \leq 2^{-(n+1)}(b_0 - a_0) = 2^{-(n+1)}(63 - 50)$

$= 2^{-(n+1)}(13)$

So if $n \geq 37$ we have $|r - c_n| \leq 2^{-(n+1)}(13) \leq 5 \cdot 10^{-11}$

and $\left|\frac{r - c_n}{r}\right| \leq 10^{-12}$