Non-linear equations

Here we try to solve some equation $f(x) = 0$ for x.

General principle

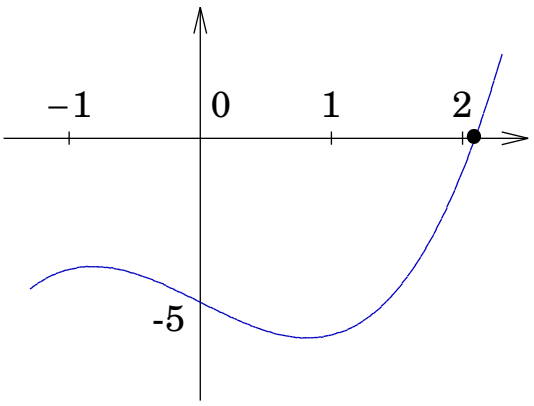
The more we know about f, the more powerful the methods we can use, and the more efficient they can be.

However, these methods tend to be harder to implement and to be much more fragile *especially* when used wrongly.

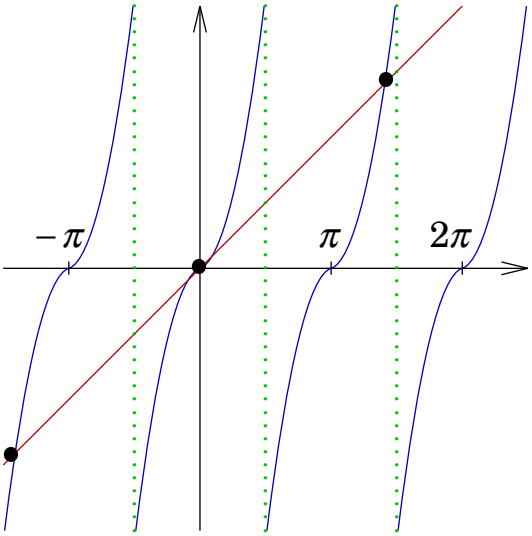
We use the following example equations:

(a) [Wallis's equation] $x^3 - 2x - 5 = 0$;
[solution: $x \approx 2.094552$]

(b) $\tan x = x$.
[solutions: $x = 0, \pm 4.4934, \pm 7.725, \ldots$; large roots are close to $(n + \frac{1}{2})\pi$].



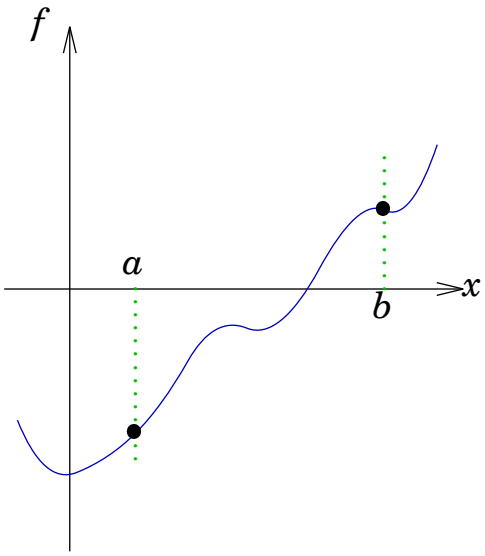
(a)



(b)

Bisection

If f is continuous, find a, b such that $f(a) < 0$, $f(b) \geqslant 0$. Then there must be an x between a and b such that $f(x) = 0$.



To localise x, set $c = \frac{1}{2}(a+b)$; then if $f(c) < 0$ replace a by c else replace b by c. Repeat until $|b-a|$ is 'small enough'.

Notes

- If f is not continuous, then we can't say anything about roots.
- We could require $f(b) > 0$; but then $f(c) = 0$ is a 'special case'.
- $|b-a|$ is halved each time round, and therefore so is the error in assuming $x = c$ as the answer.

Pro

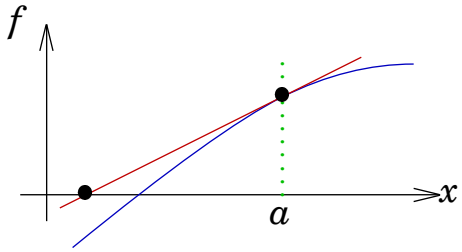
- Very simple to write a computer program for
- Guaranteed to converge

Con

- Needs two starting values [how do we get these?]
- Convergence is always slow

Newton-Raphson method

Find an approximate value, a for the root. Draw the tangent to $f(x)$ at $x = a$; where this tangent crosses the x-axis is taken to be a better approximation.



To find this point, call it c, we have two expressions for the slope of the tangent:

$$\frac{f(a)}{a-c} = f'(a),$$

or

$$c = a - \frac{f(a)}{f'(a)}.$$

Replace a by c and repeat.

Notes

Pro

- 'Self starting'; a can be anywhere. [But it is better if it is near the root.]
- Usually *very* efficient [as we'll see later] near the root.
- Easy to program—but see below.
- If you're prepared to do complex arithmetic, can be used to find complex roots of equations.

Con

- Does not necessarily converge at all. If you are unlucky, $f'(a)$ may be small, even zero, throwing you off to darkest regions, where f may not even be defined. Even without that, it can skitter around and not make progress.
- Or may, having been thrown off a little, converge, but to the 'wrong' root.
- Needs f to be differentiable, as well as continuous. Needs you to *know* the derivative [what if f is derived from experimental data?].
- Needs you to be able to write the derivative inside any computer program. Maple can do this automatically, but in C, Java, Pascal, Basic you have to supply it. Also, for most functions, f' is a much messier function than f, so some [not all] of the efficiency is lost.

Convergence

How do we know when to stop going round Newton-Raphson? Suppose the exact root [not usually known!] is b. There are two usual criteria:

(a) The value of $|a-b|$ that we have found is 'small', say $< 10^{-6}$. We judge this by comparing successive values of a.
(b) The value of $|f(a)|$ that we have found is 'small', say $< 10^{-6}$.

You cannot usually expect to satisfy both of these! Real computer programs must also check for being stuck in a loop.

- If f' is small, then f is small even when $|a-b|$ is large. Note that this happens especially if b is a repeated root of $f(x) = 0$. [Try using Newton-Raphson to find the triple root at $x = 0$ of $\tan x = x$; note that $|\tan x - x| \leqslant \frac{1}{3}10^{-6}$ when $|x| < 0.01$.]

- If f' is large, then f is large even when $|a-b|$ is small.