Minimisation in 1D

How this is different to zero finding (and how it is the same)

Bracketing minima

Minimisation using first derivatives

Golden-section search

Jarrat's method

Brent's method (of minimisation)

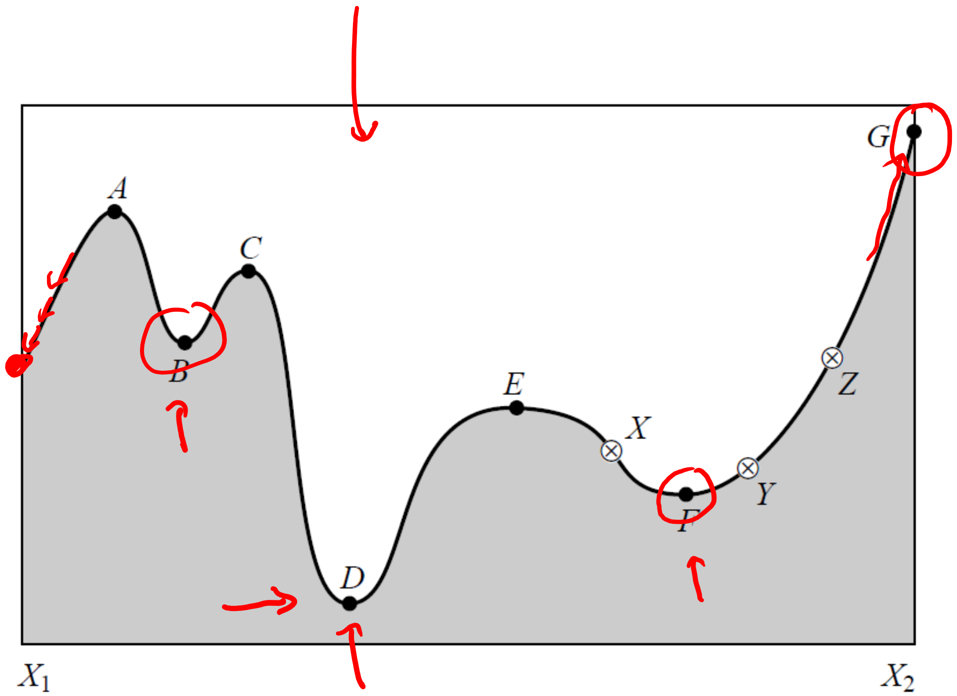
Minimisation is in general *more difficult* than zero finding.
It has all the problems and instabilities of root finding and none of the advantages.

(Side note: we will talk of *minimization,* but *this is the same problem as maximization).*

While finding a zero is pretty much unambiguous, when searching
For a minumum you might get:

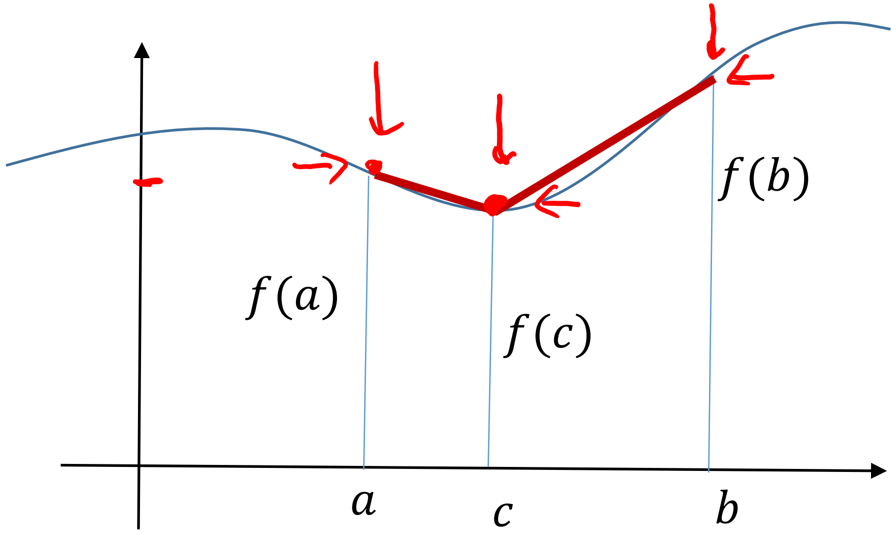
a) A local, not a global minimum (B, F)

b) A minimum/maximum on the edge of the domain (G) that is not the minimum that you want (D)



Bracketing minima

A minimum is numerically identified by the following pattern:



The maximization problem is equivalent to the minimization problem

$$\min f = \max(-f).$$

Whenever

$$f(c) < f(a) \text{ and } f(c) < f(b)$$

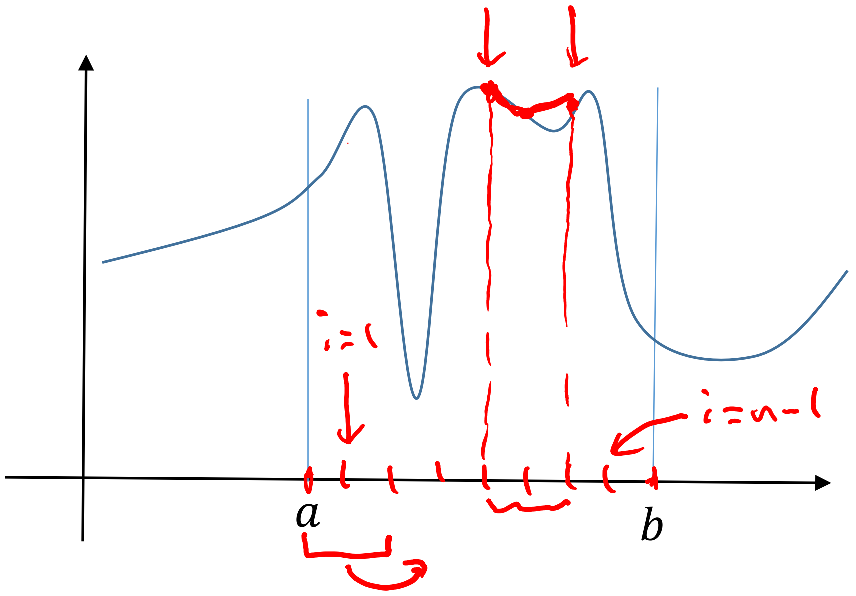
we have bracketed a minimum.

Warning: we must make sure that $a < c < b$ at all times for this to work!

Routine for bracketing minima

Algorithm:

1. Divide the interval up into equal intervals
2. Test each triplet for a minimum

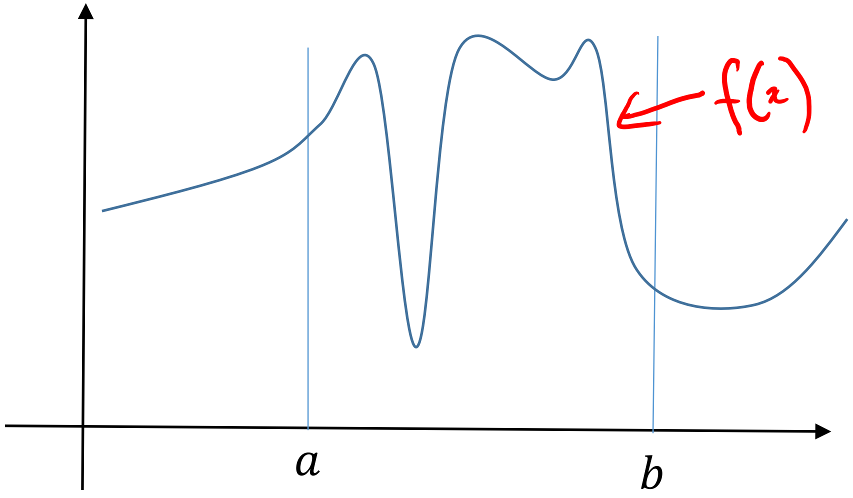


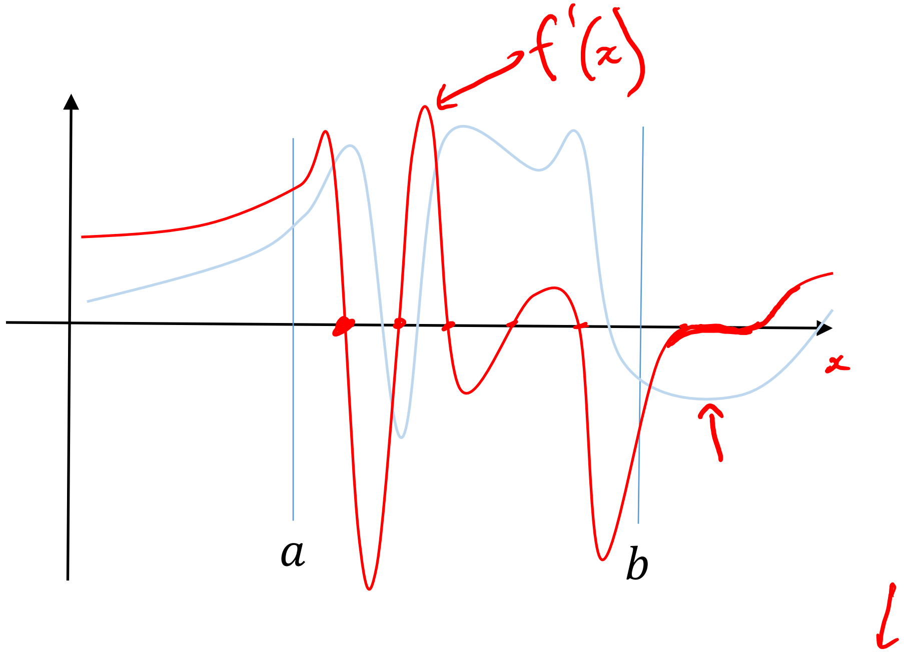
Pseudocode:

```
function Mbracket(a,b,N)
a = left point of interval
b = right point of interval
n = Number of segments

h = width of each segment = (b-a)/n

Loop from i = 1 to n-1
        x = h*i
        If f(x)<f(x-h) and f(x)<f(x+h)
                bracket found, store x-h, x+h in blist
        end if
end for loop
return blist
```

Minimisation from the derivative

Minima can be found by searching for zero crossings of the derivative





Advantages:

1. Fast, if close enough to the minimum

Disadvantages:

1. Unstable (more so than Newton's method)
2. Difficult to maintain a bracket
3. Fails if the *second derivative* is also zero

Algorithm:

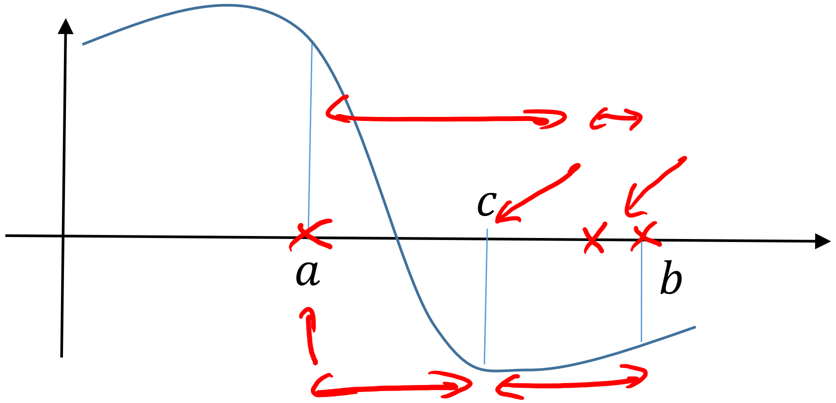
1. Create a function g(x) = df/dx
2. Find a minimum bracket for f
3. Find a zero bracket for g
4. Find the zeros of g, using Newton's method,

Brent's method etc., attempting to maintain the zero bracket. If a step fails, return to the previous step and rebracket the minimum of f.

Golden-section search

This is a stable but reasonably quick method for minimisation.

To understand how this works, let's go back to the bisection method.
Why do we choose the midpoint?



Fundamental law* of numerical search:
(a.k.a Murphy's Law of numerical search)

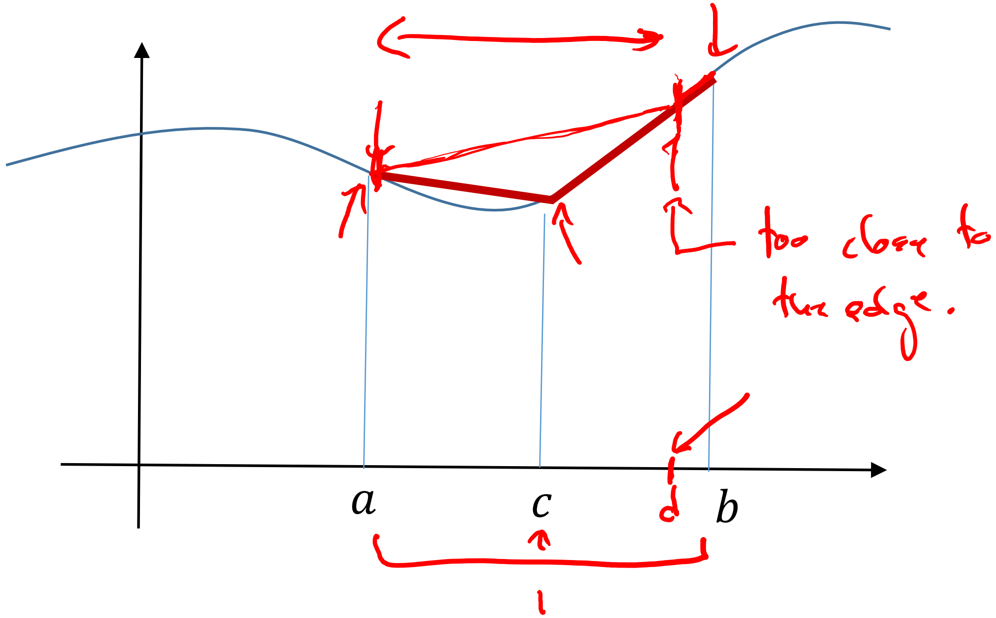
The thing you're looking for will try to be in the biggest interval of your search domain.

A good reason for choosing the midpoint is that, as far as we know, the root is *equally likely to be in each half of the interval.*

Can we do the same for finding the minimum?

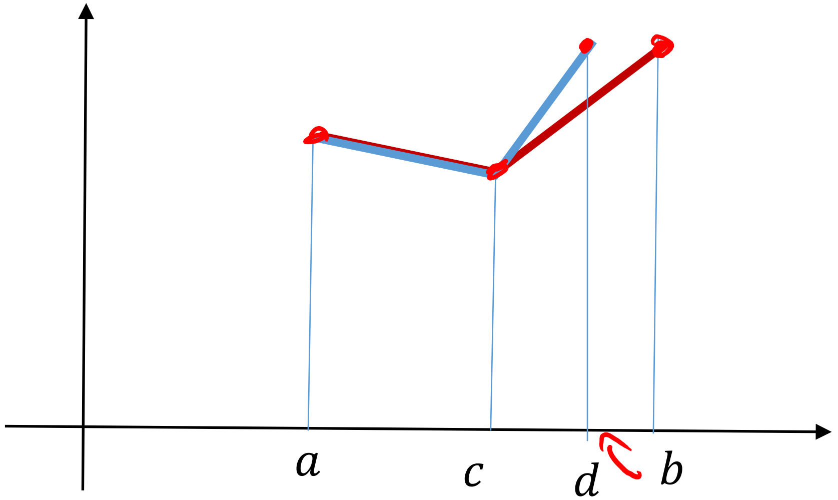
*not a law.

What happens if we *start with a bracketed* minimum.
Which point should we choose next?

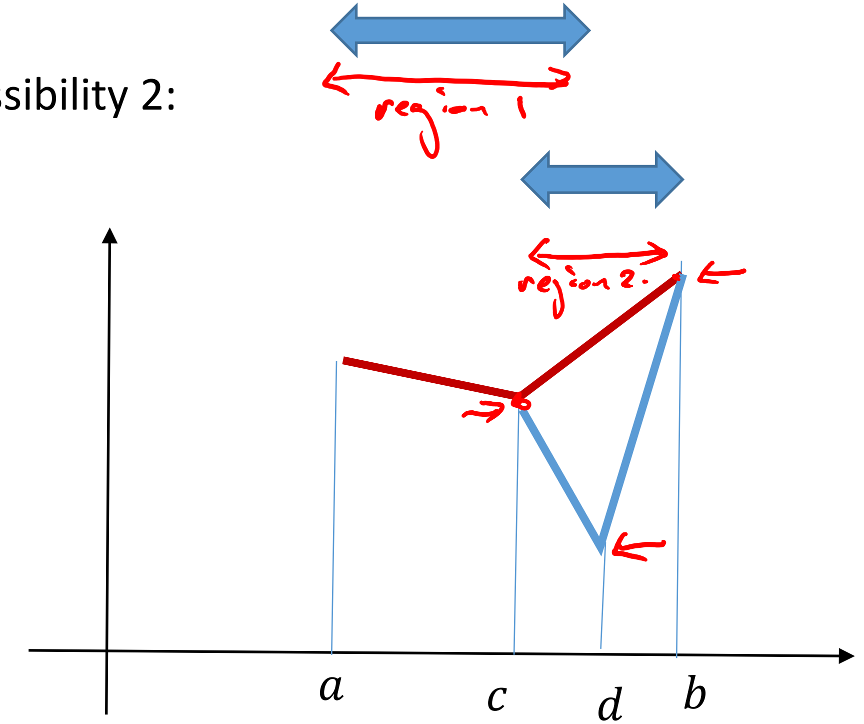


We would like to choose the next point d such that the two regions for the *new* bracket are equal in length.

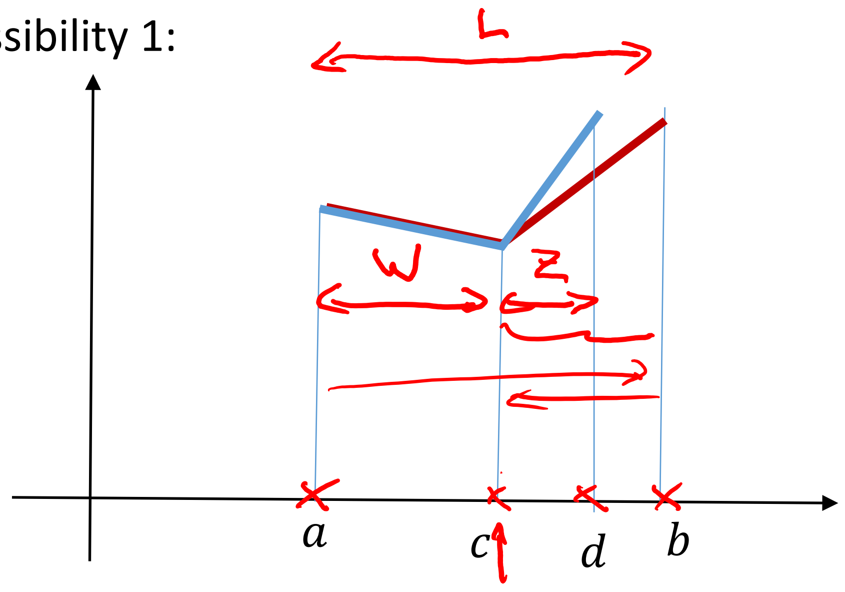
Possibility 1:

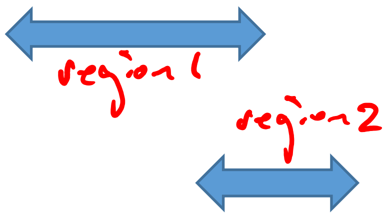


Possibility 2:

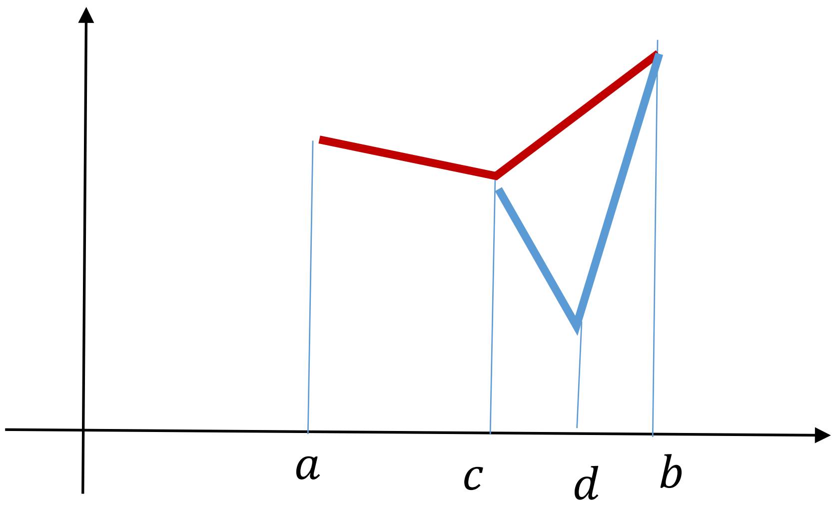


Possibility 1:





Possibility 2:



We want:

① $|\text{Region } 1| = |\text{Region } 2|$

② Fraction of the new point to be the same on each iteration.

①:
$$W + Z = L - W \Rightarrow \underline{Z = L - 2W.}$$

②:
$$\frac{W}{L} = \frac{Z}{L-W}$$

$$\Rightarrow \frac{W}{L} = \frac{L-2W}{L-W} = \frac{1-2\frac{W}{L}}{1-\frac{W}{L}}.$$

Let $\gamma = \frac{W}{L}$. Then $\underline{\gamma = \frac{1-2\gamma}{1-\gamma}.}$

$$\gamma = \frac{1-2\gamma}{1-\gamma}$$

re-arrange and solve:

$$\gamma(1-\gamma) = 1-2\gamma$$

$$\gamma - \gamma^2 = 1-2\gamma$$

$$\gamma^2 - 3\gamma + 1 = 0.$$

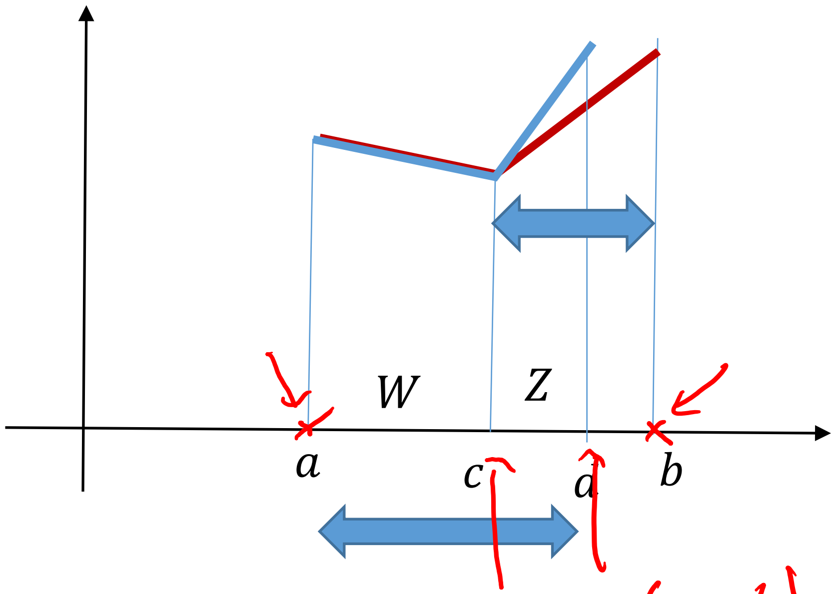
Solve: $\gamma = \frac{1}{2}\left(3 \pm \sqrt{9-4}\right)$

$$= \frac{1}{2}\left(3 \pm \sqrt{5}\right)$$

Rewrite: we can show that (exercise)

$$\gamma = 1 - \frac{1}{\phi}$$

where $\phi = \frac{1}{2}\left(1+\sqrt{5}\right)$ is golden ratio



$$c = a + W = \underbrace{a + L\left(1 - \frac{1}{\phi}\right)}$$

$$d = a + W + Z$$

$$= a + W + L - 2W$$

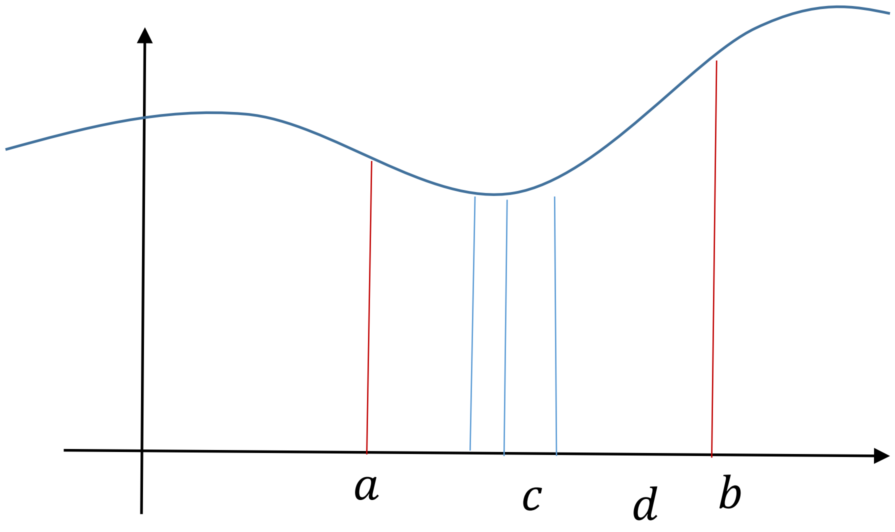
$$= a + L - W$$

$$= a + L - L\left(1 - \frac{1}{\phi}\right)$$

$$= \underbrace{a + L\frac{1}{\phi}}.$$

The best ratio to pick for the "midpoint" is $1/\phi$ along the bracketed interval, where ϕ is the golden ratio

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618\ldots$$



Golden-section search:

1. Start with a bracketed minimum [a,b]

2. Pick two new points
   c = b - (b-a)/ ϕ
   d = a + (b-a) /ϕ

3. If f(c) < f(d)
   minimum is now bracketed by a and d:
   a unchanged, b = d

Else If f(d)<f(c)

Minimum is now bracketed by c and b
   b unchanged
   a = c

4. Repeat

The Golden-section search is

1. Slow (like bisection, it has a *linear* convergence)

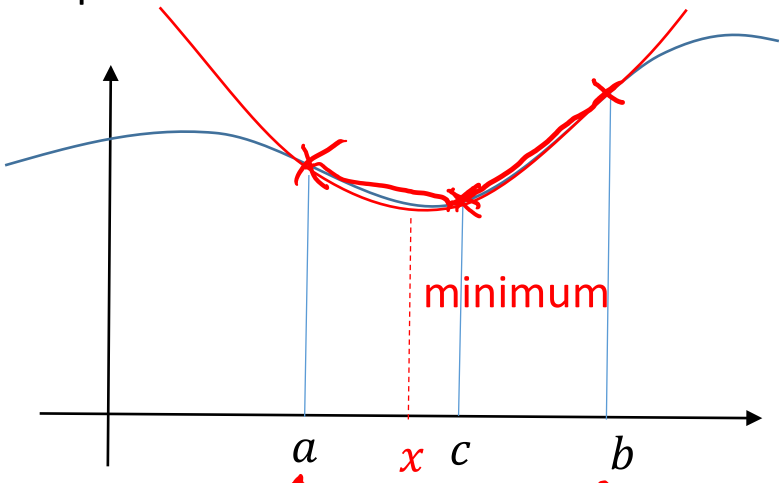
2. Robust

Is there another method that is equally robust but has superlinear convergence?

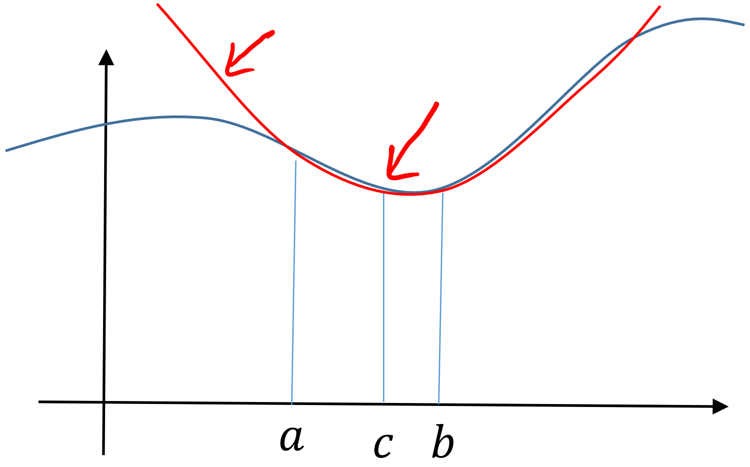
Jarratt's method

The idea: use *parabolic interpolation* to find the minimum.

Step 1



Step 2



Jarratt's method:

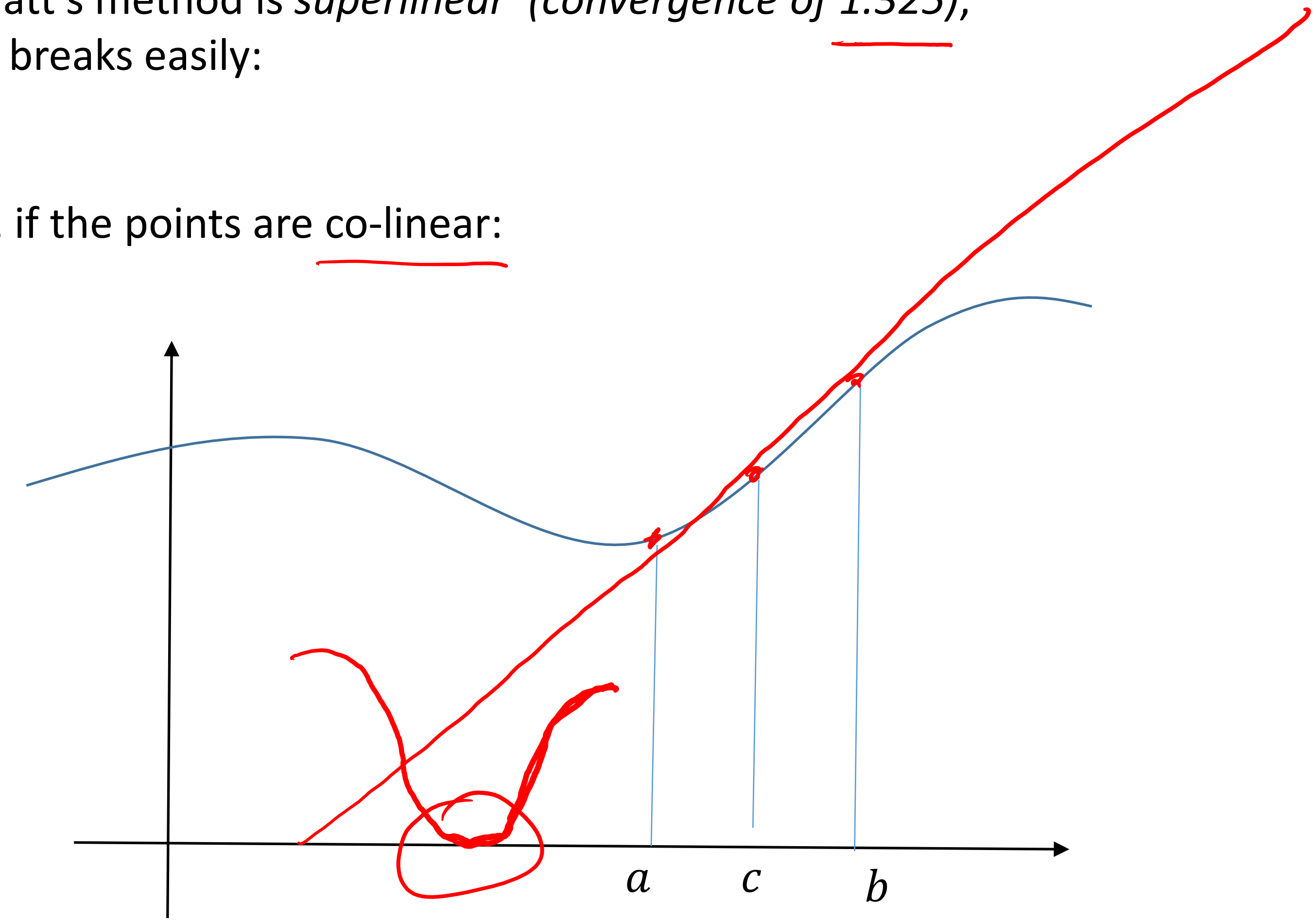
1. Start with a bracketing triplet a,b,c
2. Find the minimum of an interpolating parabola
3. Set a = whichever of a,b is closest.
   b = c
   c = x
4. Repeat from 1.

Formula for the minimum point x of a parabola through points a,b,c:

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

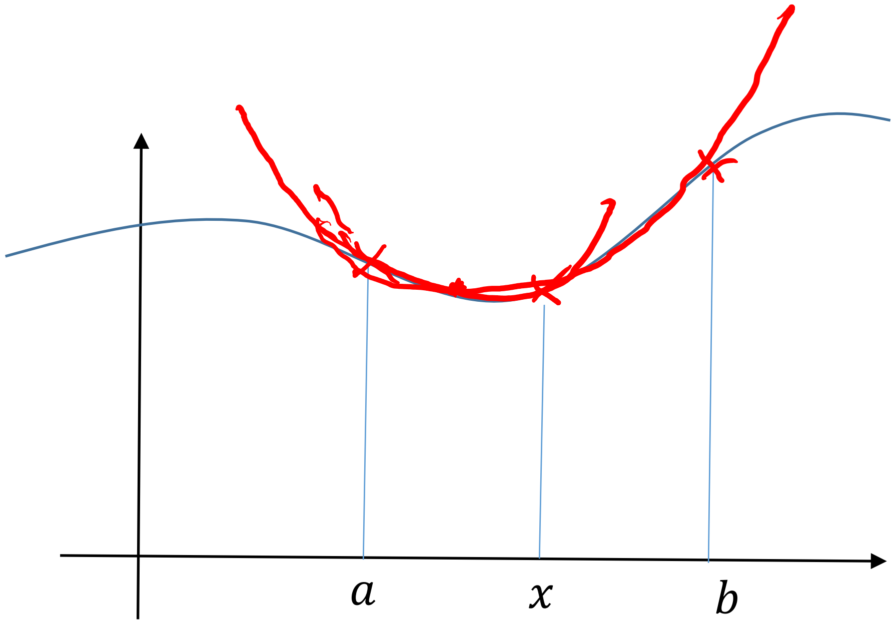
Jarratt's method is *superlinear (convergence of 1.325),*
but breaks easily:

e.g. if the points are co-linear:



<u>Brent's method (of minimisation)</u>

The idea: combines *parabolic interpolation with* a golden search



Brent's method algorithm:

1. Start with three points that bracket the minimum
2. Do a *parabolic interpolation*
3. *Repeat 2 while this is going well**
4. If it's not, do one step of a golden search, then go back to step 2.

*a) the midpoint is within the brackets, and
b) the three points are not co-linear, and
c) The current step is smaller than *half the second-last step*

Brent's method of minimisation combines robustness with speed, and is the "gold standard" for 1D function minimisation.