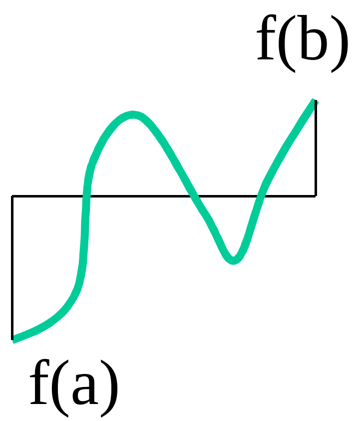
Lecture 4

- Finding roots

Finding roots: Find x such that f(x)=0 for given f.

Bisection method:

The intermediate value theorem states (the obvious):
if a continuous function changes sign within a given
interval, it has at least one root in such interval.

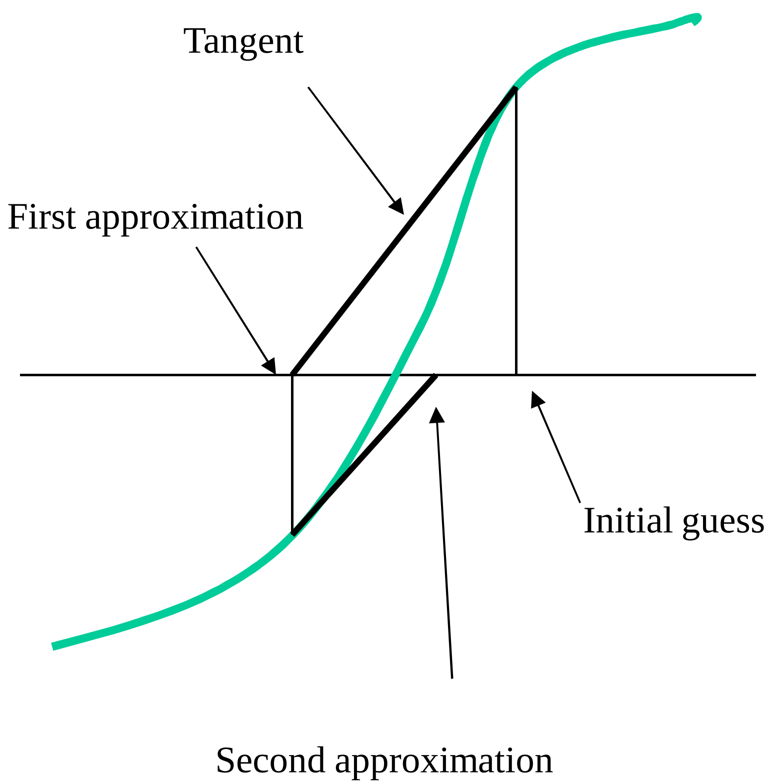


Suppose the function has one root in the interval.
Then if I partition the interval into two at the midpoint,
the root is either in the right or the left semi-intervals. To see where,
I just need to check if at the ends of the semi-intervals the sign of
the function is the same or not.

Iterating this procedure I can “bracket” the root within an interval
of ever increasing sharpness.

The method is not fast but is very robust: it always works.

Newton-Raphson method:



Start with a guess. Find the place where the tangent at your initial guess intersects the abscissa. This is your first approximation. If the function is continuous and you start reasonably close to the root, you will get a convergent procedure.

Alternatively, pick a point x_0 and,

$$f(x_R) \approx f(x_0) + (x_R - x_0) f'(x_0) = 0$$

Second derivatives small.

Therefore

$$x_R = \frac{x_0 f'(x_0) - f(x_0)}{f'(x_0)} = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The method therefore consists on iterating

$$x_N = x_{N-1} - \frac{f(x_{N-1})}{f'(x_{N-1})}$$

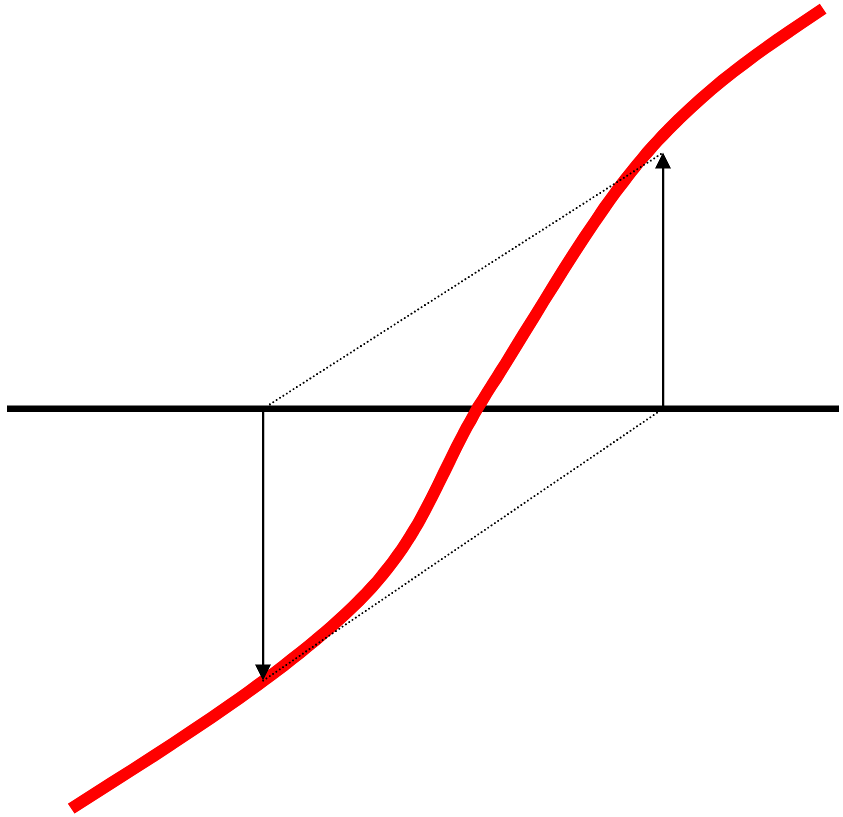
And one needs a “stopping criteria”, for instance,

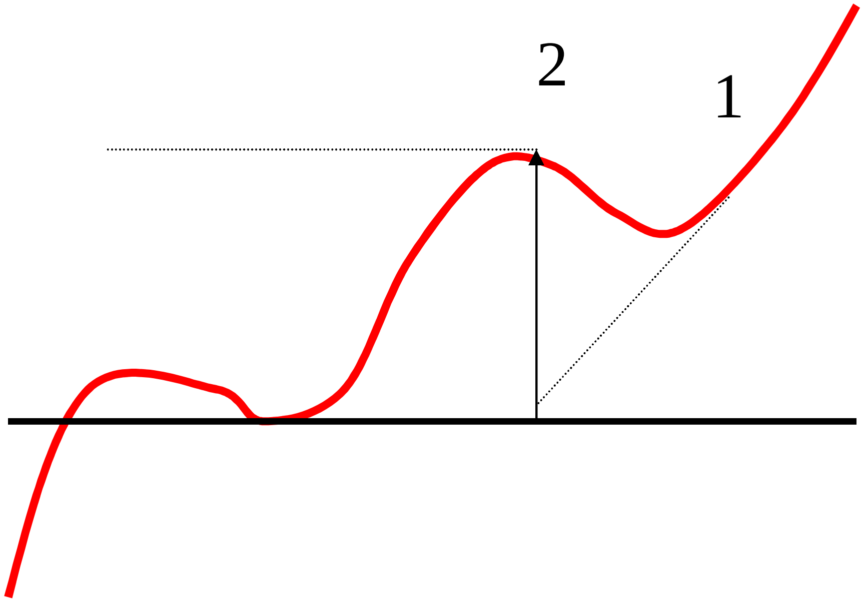
$$|x_N - x_{N-1}| < e \qquad \text{for a given e}$$

$$\text{or } \frac{|x_N - x_{N-1}|}{x_N} < e \qquad \text{or } f(\mathrm{x_N}) < e$$

All of these may have difficulties in practice...

Pathologies of Newton-Raphson:





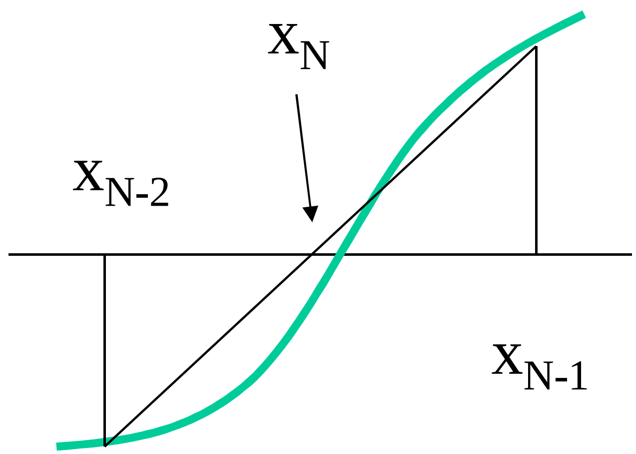
Another issue is that the method requires computing the derivative of the function. This might be hard to do analytically, so we might want to choose to do it numerically. Or to have a method that does not require it:

$$f'(x_{N-1}) \approx \frac{f(x_{N-1}) - f(x_{N-2})}{x_{N-1} - x_{N-2}}$$

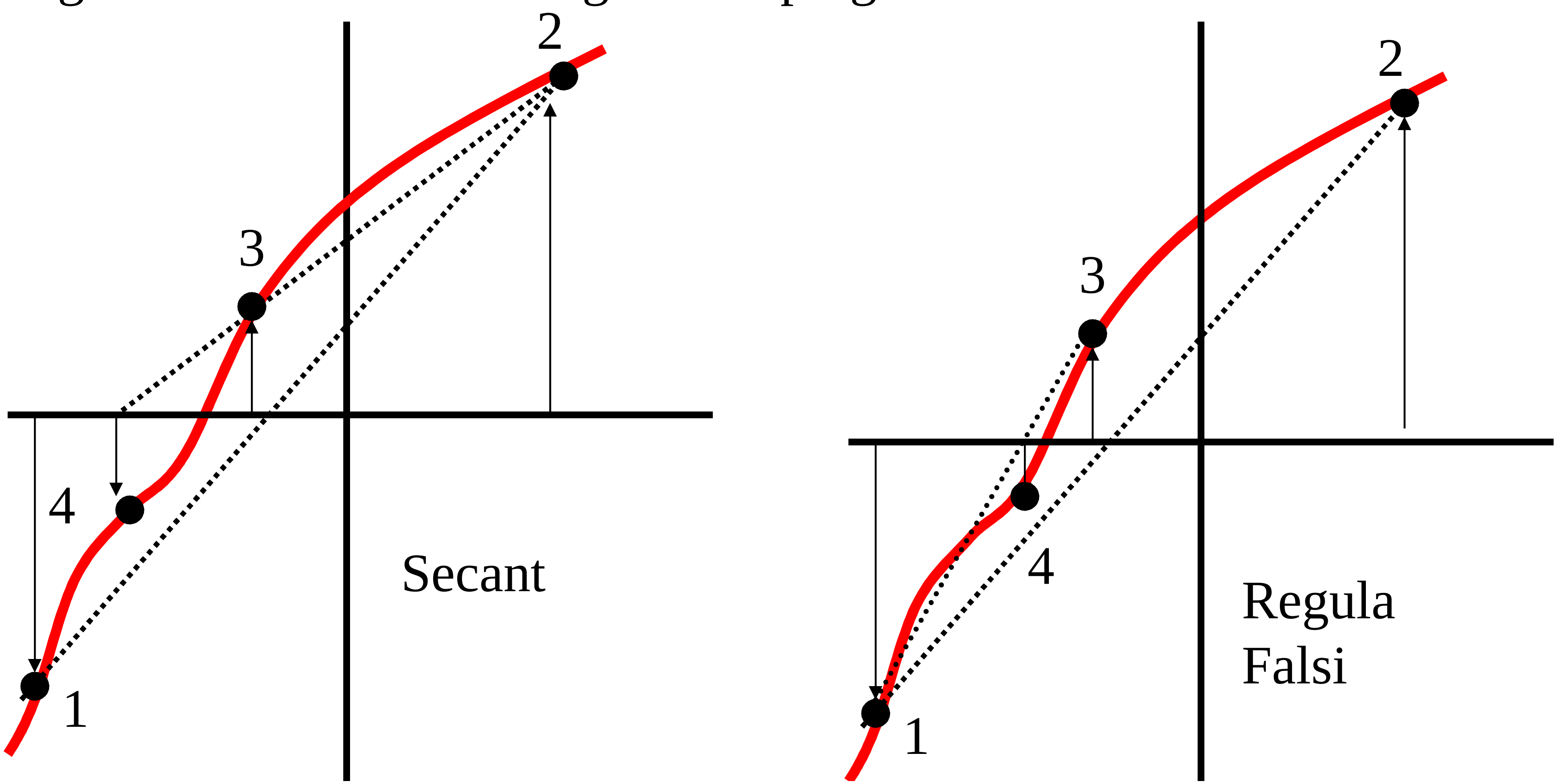
Combining with $$x_N = x_{N-1} - \frac{f(x_{N-1})}{f'(x_{N-1})}$$

$$x_N = x_{N-1} - \frac{f(x_{N-1})(x_{N-1} - x_{N-2})}{f(x_{N-1}) - f(x_{N-2})}$$

"Secant method", geometrically,



A variation of this method is the **regula falsi** method. In the secant method one computes the secant of successive points, whereas in regula falsi one checks that the two points have functional values that are opposite in sign. If the last two approximations have the same sign, the method digs back into previous estimates to get a different sign. This has the advantage of keeping the root bracketed.



Rates of convergence:

Error in i-th iteration with respect to the correct root.

Bisection method: $\varepsilon_{i+1}=\varepsilon_i/2$ Therefore $\lim_{n\to\infty}\dfrac{\varepsilon_{n+1}}{\varepsilon_n}=\dfrac{1}{2}$

"Converges linearly"

Newton's method:

$$x_N = x_{N-1}-\frac{f(x_{N-1})}{f'(x_{N-1})} \Rightarrow \varepsilon_N=\varepsilon_{N-1}-\frac{f(x_{N-1})}{f'(x_{N-1})}$$

$$f(x_{N-1}) \approx f(x_R)+(x_{N-1}-x_R)f'(x_R)+\frac{(x_{N-1}-x_R)^2}{2}f''(x_R)$$

And since $x_{N-1}-x_R=\varepsilon_{N-1}$ If we substitute, we get,

$$\varepsilon_N=-\frac{(\varepsilon_{N-1})^2}{2}\frac{f''(x_R)}{f'(x_R)}$$

So convergence is quadratic.

```
function rtsafe(funcd,x1,x2,xacc)
     parameter (maxit=100)
     call funcd(x1,fl,df)
     call funcd(x2,fh,df)
     if(fl*fh.ge.0.) pause 'root must be bracketed'
     if(fl.lt.0.)then
       xl=x1
       xh=x2
     else
       xh=x1
       xl=x2
       swap=fl
       fl=fh
       fh=swap
     endif
     rtsafe=.5*(x1+x2)
     dxold=abs(x2-x1)
     dx=dxold
     call funcd(rtsafe,f,df)
     do 11 j=1,maxit
       if(((rtsafe-xh)*df-f)*((rtsafe-xl)*df-f).ge.0.
*          .or. abs(2.*f).gt.abs(dxold*df) ) then
     dxold=dx
         dx=0.5*(xh-xl)
         rtsafe=xl+dx
         if(xl.eq.rtsafe)return
       else
         dxold=dx
         dx=f/df
         temp=rtsafe
         rtsafe=rtsafe-dx
         if(temp.eq.rtsafe)return
       endif
       if(abs(dx).lt.xacc) return
       call funcd(rtsafe,f,df)
       if(f.lt.0.) then
         xl=rtsafe
         fl=f
       else
         xh=rtsafe
         fh=f
       endif
11   continue
     pause 'rtsafe exceeding maximum
            iterations'
     return
     end
```

Order interval so f(a)<0

If Newton-Raphson out of range, bisect

Newton-Raphson

Check

Routine that combines bisection and Newton-Raphson to enjoy the advantages of both.

Systems of non-linear equations:

Are very hard.

One is essentially looking in a multidimensional space for intersections of surfaces, with a priori no clue as to where they may happen and no guarantee that one found everything.

We can write formulae very similar to the ones we introduced here for many variables and equations.

$$F_i(\vec{x} + d\vec{x}) = F_i(\vec{x}) + J \bullet d\vec{x} + O(\vec{x}^{\,2}) \qquad J_{ik} = \frac{\partial F_i}{\partial x_k}$$

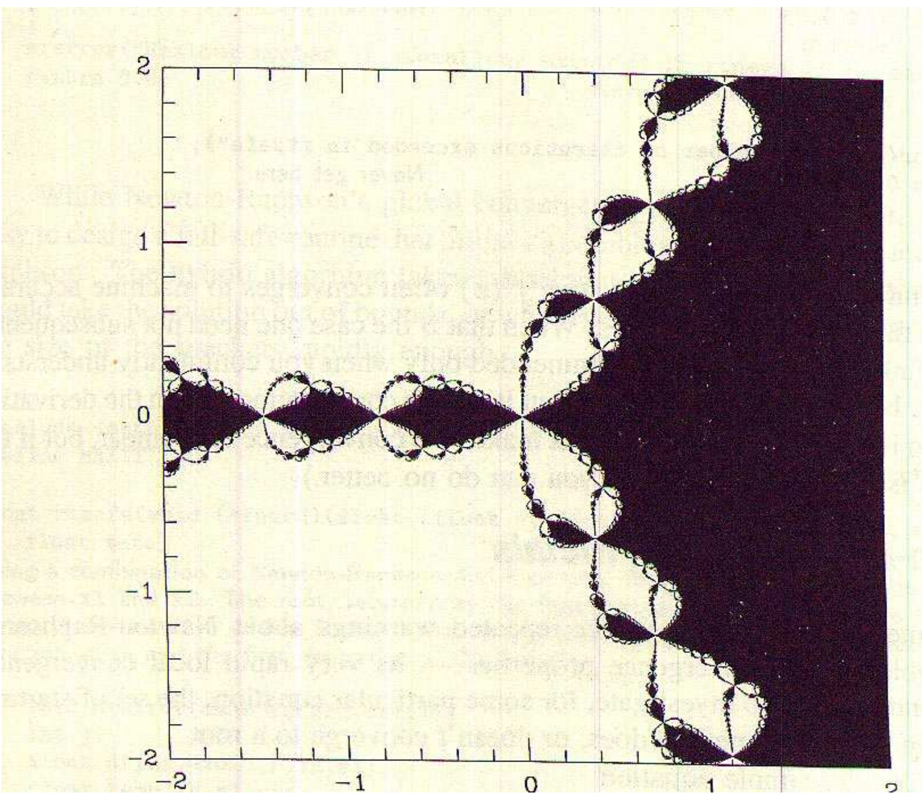
$$d\vec{x} = -J^{-1}F$$ Newton-Raphson.

It is usually better to combine this method with a minimization idea, since the latter is very efficient to implement (see Numerical Recipes)

$$f = \sum_i F_i F_i$$

Chaos and Newton-Raphson:

Suppose one considers the equation $z^3-1=0$. The root is at $z=1$. Suppose one now applies Newton-Raphson to find this root and asks the question "which points close to $z=1$ converge to that root" And suppose one poses the problem in the complex plane. The answer:



How could this happen?
The method has an instability if $f'=0$! At such points a small deviation in the guess can yield wildly differing next approximations. Chaos!

Summary

- Bisection always works, but may require many tries.
- Newton-Raphson sometimes fails, and possible pathologies are many.