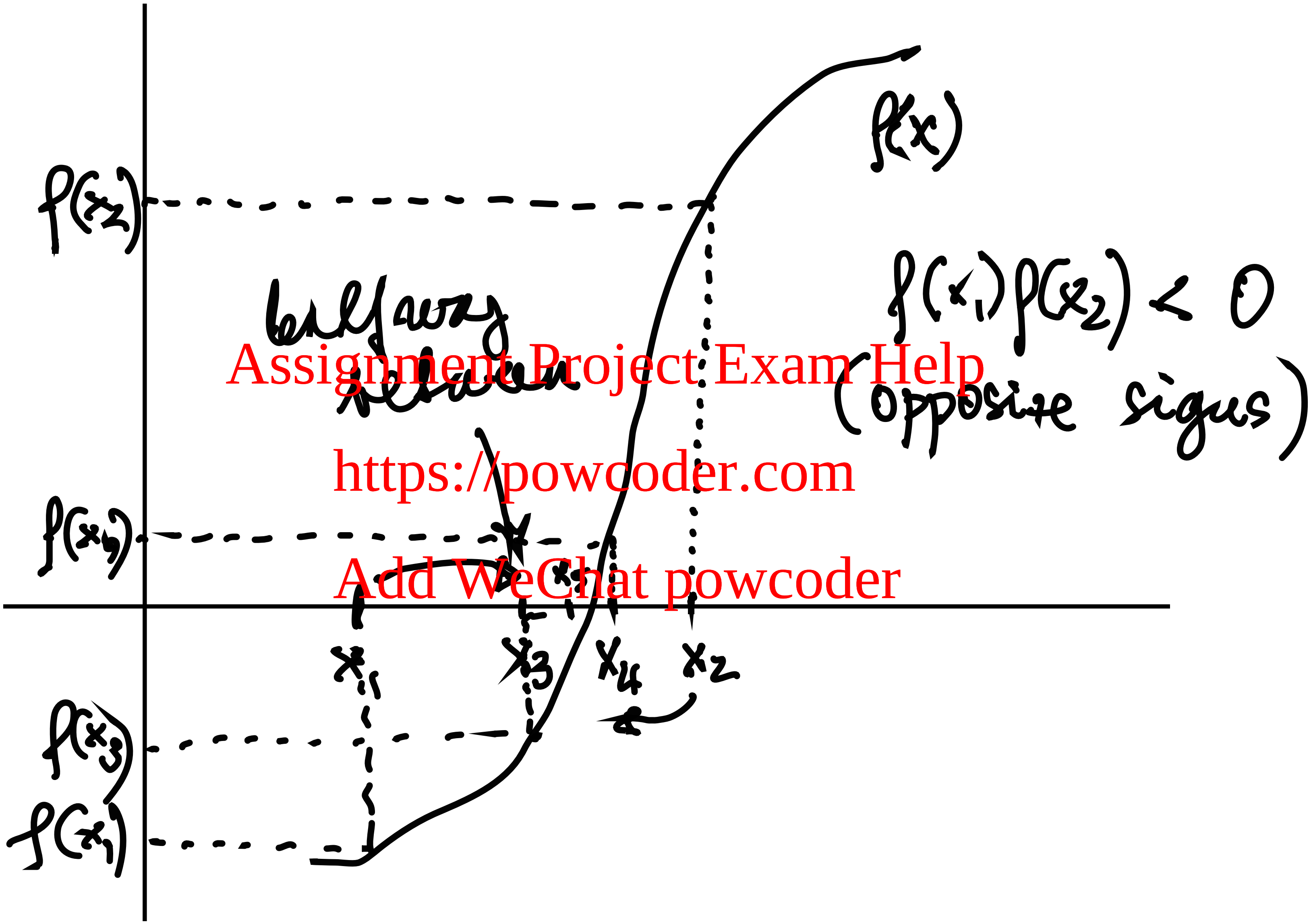
STAT 513/413: Lecture 18
Nonlinear equations (in one variable)

(Very classical)

Bisection: trapping the root in between



We always know a root is inside: it is a *safe* method

Some general remarks

Any method of this kind is always getting us *a* root

$$\text{some } x_0 \text{ such that } \quad f(x_0) = 0$$

So, unless it is *the* root (there is no other one), the method can yield a different one, if starting conditions are different

The overwhelming majority of methods (pretty much all we encounter here) are like this - and even worse (will see...)

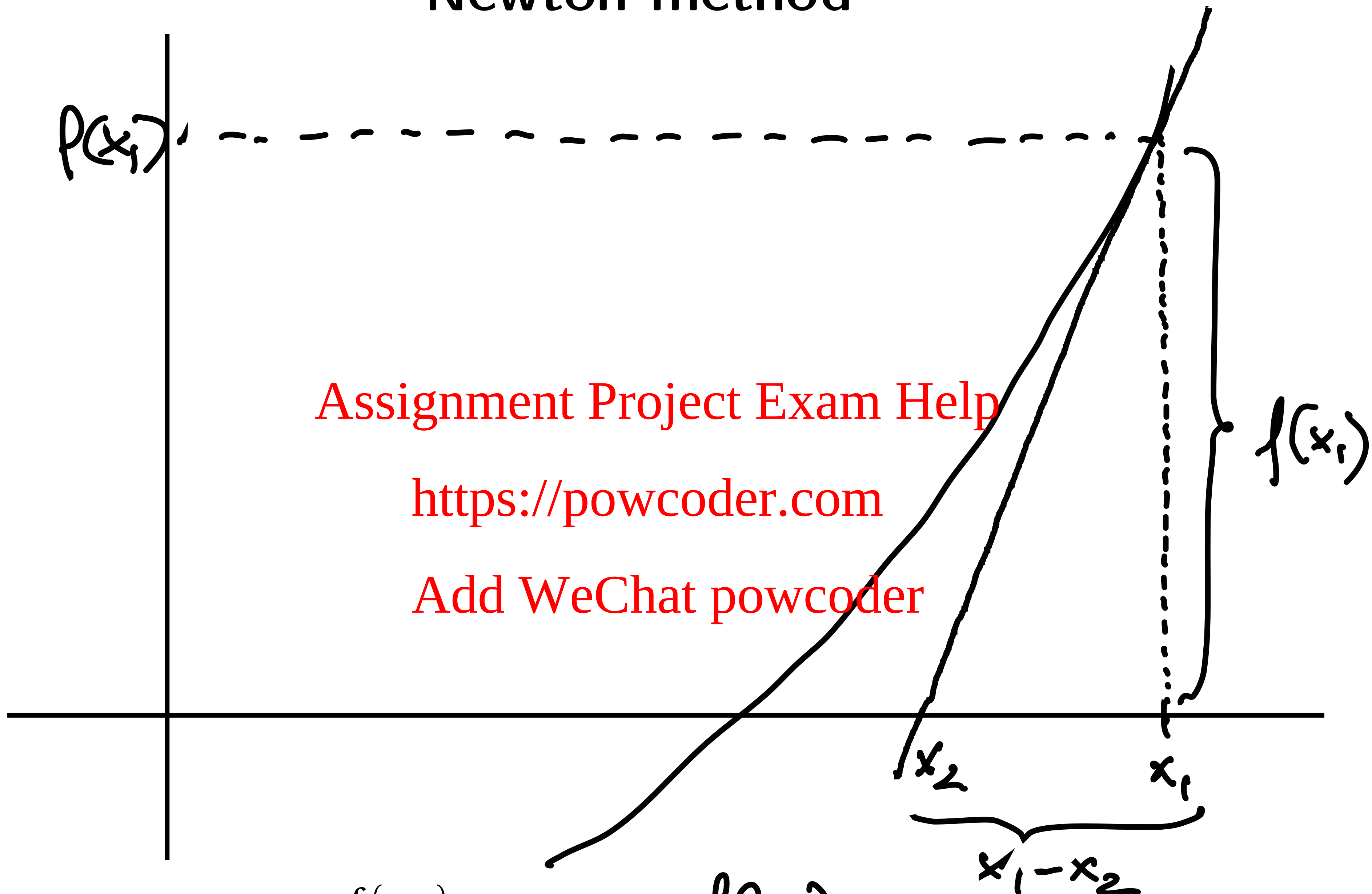
Bisection: always yields a root, but it is slow - linear convergence (the error is bounded by the length of the interval, which is multiplied by $1/2$ at every step)

However, it does not demand a lot:

- f does not have to be even continuous, that is, there may not be any x_0 such that $f(x_0) = 0$; and still, the method locates "the sign change at x_0"

- and all we need is to be able to evaluate f at each step

Newton method



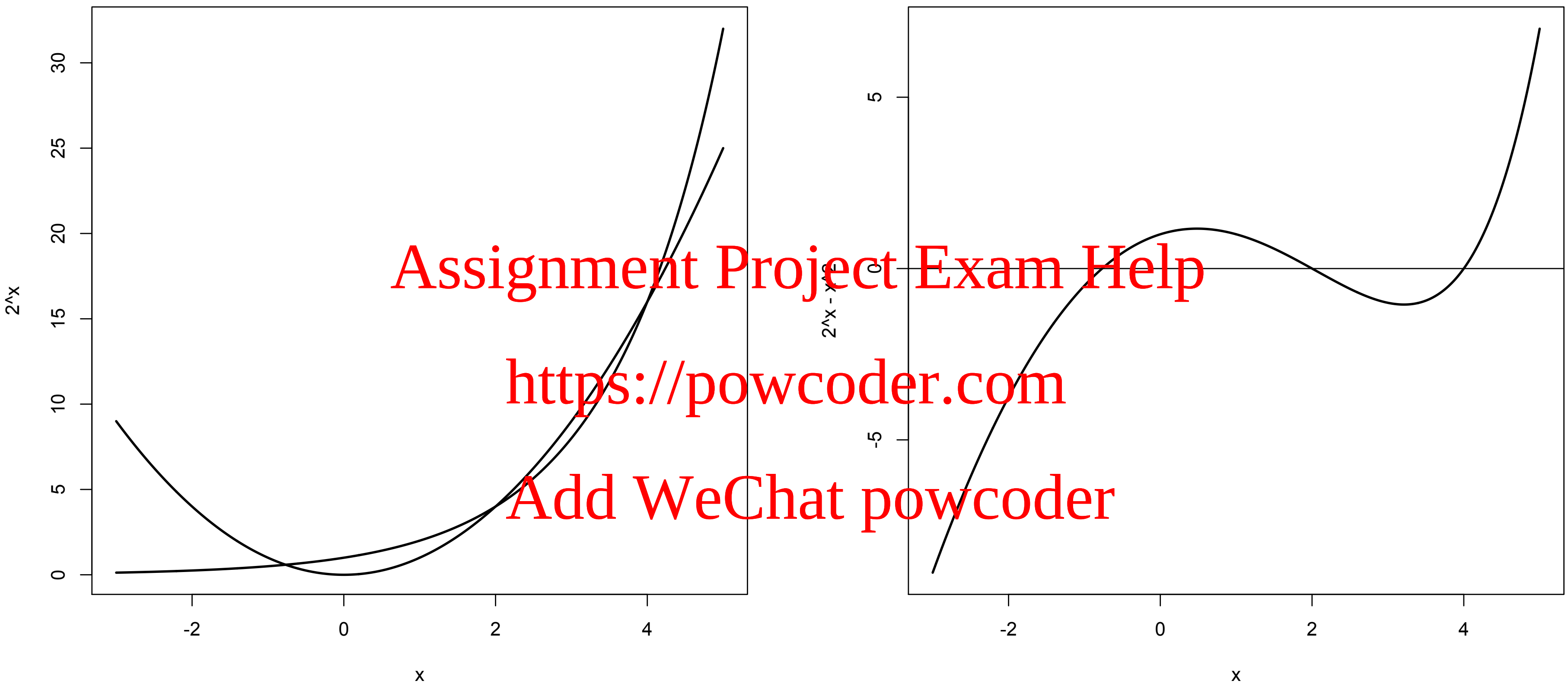
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \longleftarrow \quad \frac{f(x_1)}{x_1 - x_2} \approx f'(x_1)$$

A nice example

The roots of the equation $2^x = x^2$

```
> fu = function(z) x-(2^x-x^2)/(2^x*log(2)-2*x)
> z=0
> z=fu(z);z
[1] -1.442695
> z=fu(z);z
[1] -0.8970646
> z=fu(z);z
[1] -0.7734702
> z=fu(z);z
[1] -0.7666851
> z=fu(z);z
[1] -0.7666647
> z=fu(z);z
[1] -0.7666647
> z=fu(z);z
[1] -0.7666647
```

Picturing it

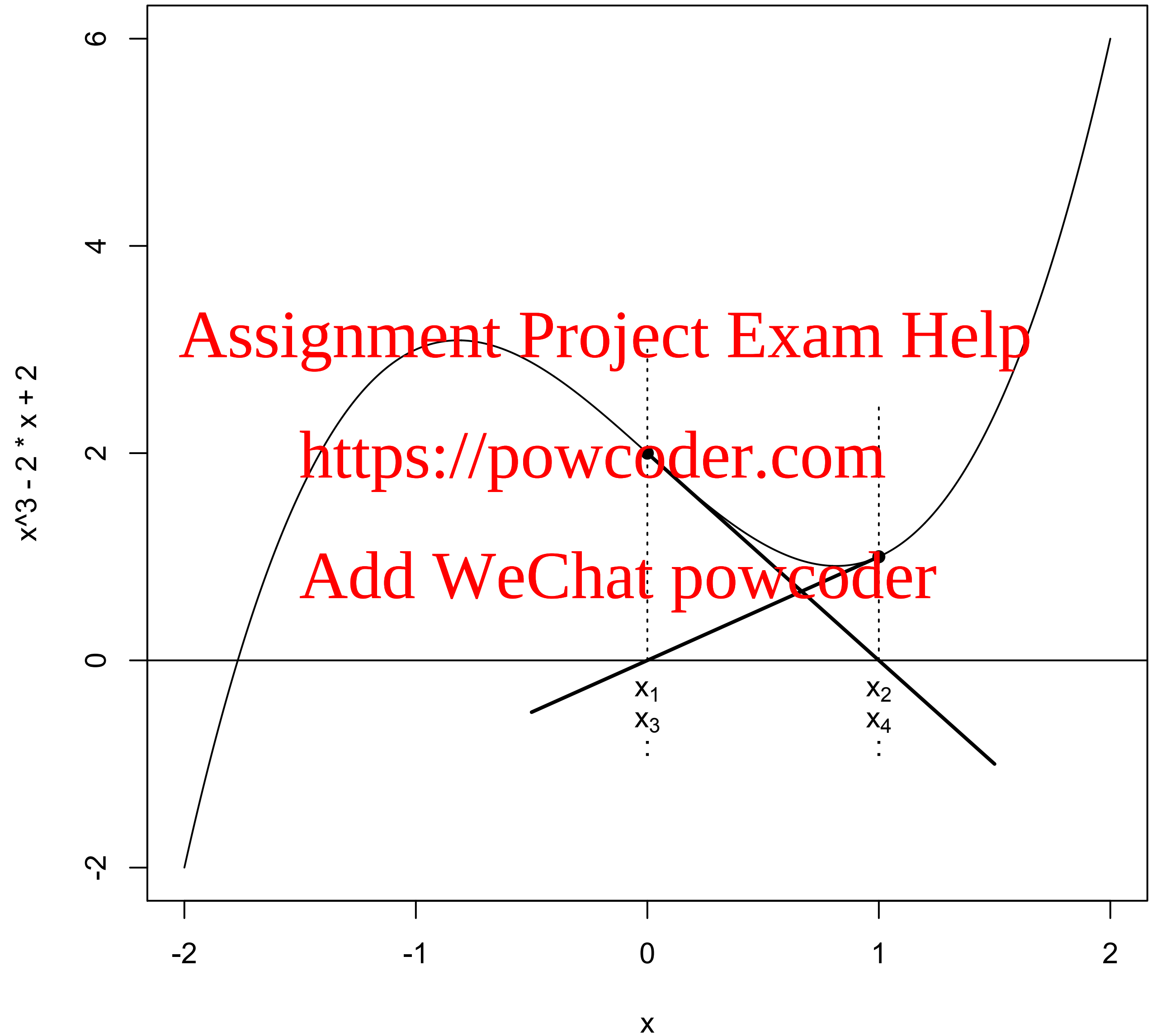


A not that nice example

The roots of the equation $x^3 = 2x - 2$

```
fun <- function(x) x^3-2*x+2
nwt <- function(x) x - fun(x)/(3*x^2-2)
> x=0
> x=nwt(x);x
[1] 1
> x=nwt(x);x
[1] 0
> x=nwt(x);x
[1] 1
> x=nwt(x);x
[1] 0
> x=nwt(x);x
[1] 1
> x=nwt(x);x
[1] 0
...
```

The picture



Newton method: properties

It is one of the “even worse” methods: it may not converge

However, in many well-behaved situations it does

And it does quickly: quadratic convergence
(if $f'(x_0) \neq 0$, and the starting point is “close enough”)

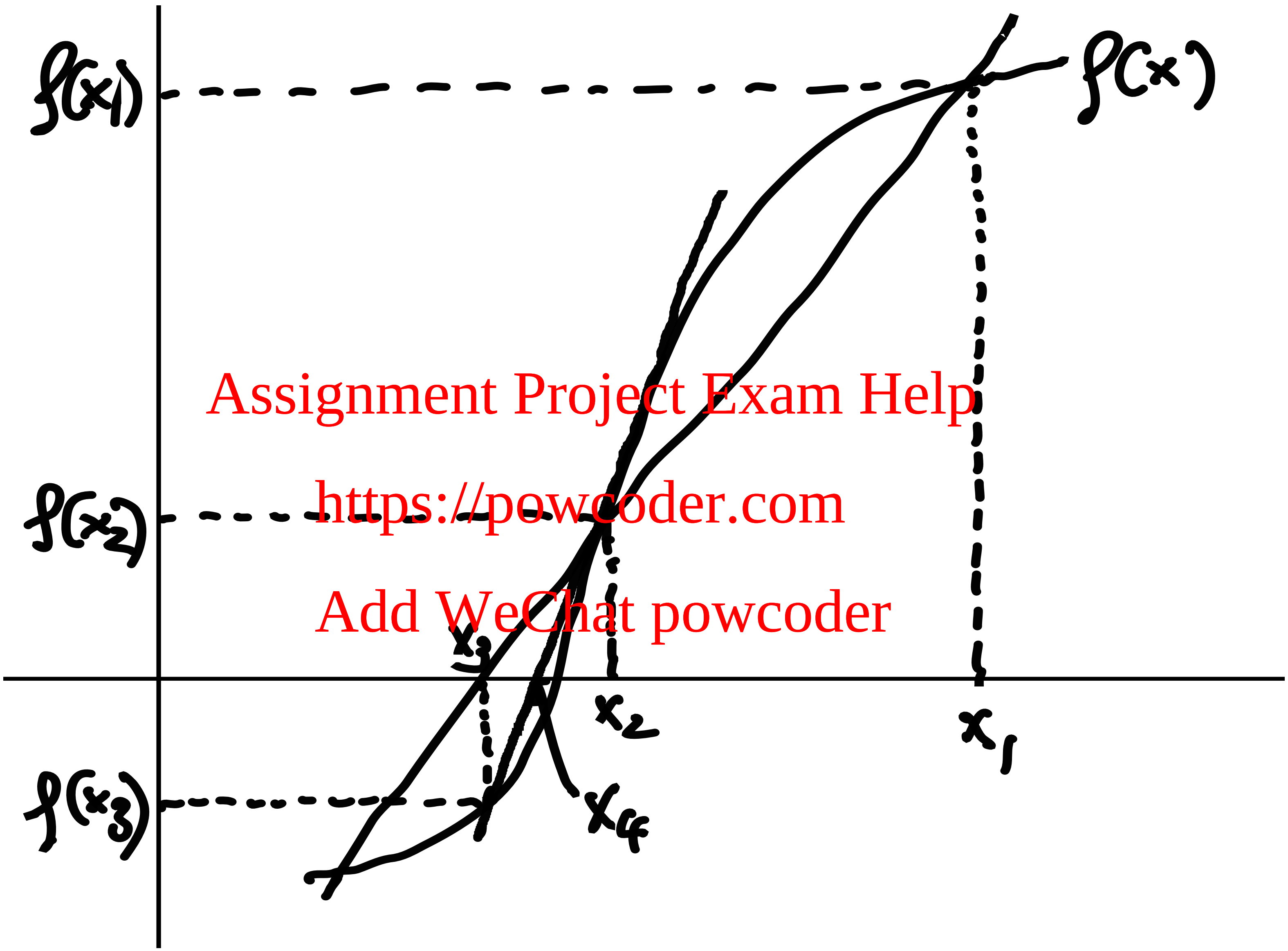
It is also more demanding: apparently

- f does have to be not merely continuous, but differentiable
- and at each step, we should be able to evaluate not only f, but also its *derivative*

Despite all of this, the Newton method (sometimes called Newton-Raphson) is *very important* - we can say that the most important method we look at. It is in a sense central to everything that comes later. We will introduce it multidimensional generalization, but it all comes from the univariate version.

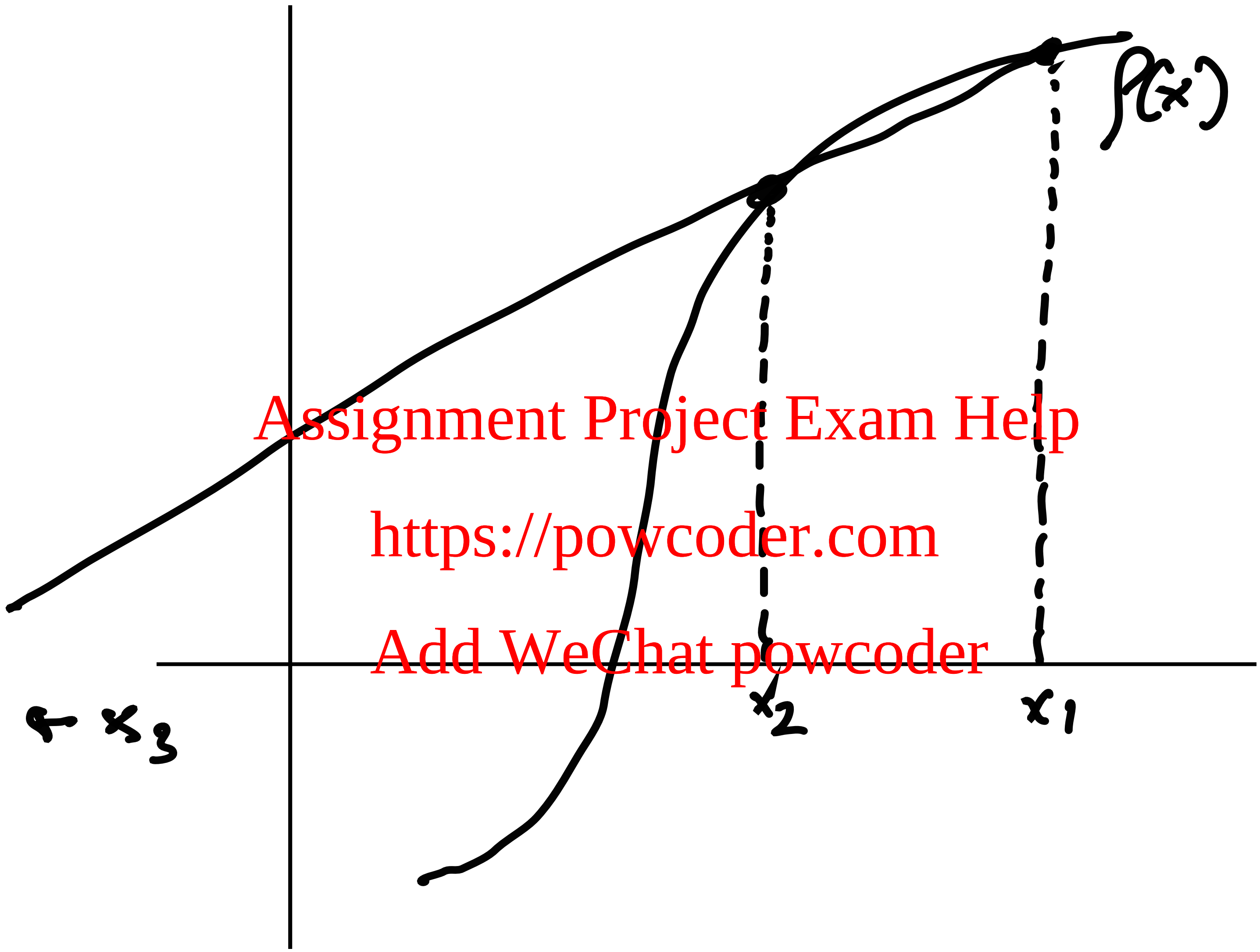
Now, a few other methods

Secant method: nice picture



$$\frac{f(x_1)}{x_1 - x_3} = \frac{f(x_2)}{x_2 - x_3}$$

Secant method: not that nice picture



$$x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)} = \frac{x_2 f(x_1) - x_1 f(x_2)}{f(x_1) - f(x_2)}$$

Secant, and others

Secant can be seen as a modification of the Newton method

- in which the derivative of f is approximated by a (divided) difference: thus it does not have to be evaluated!

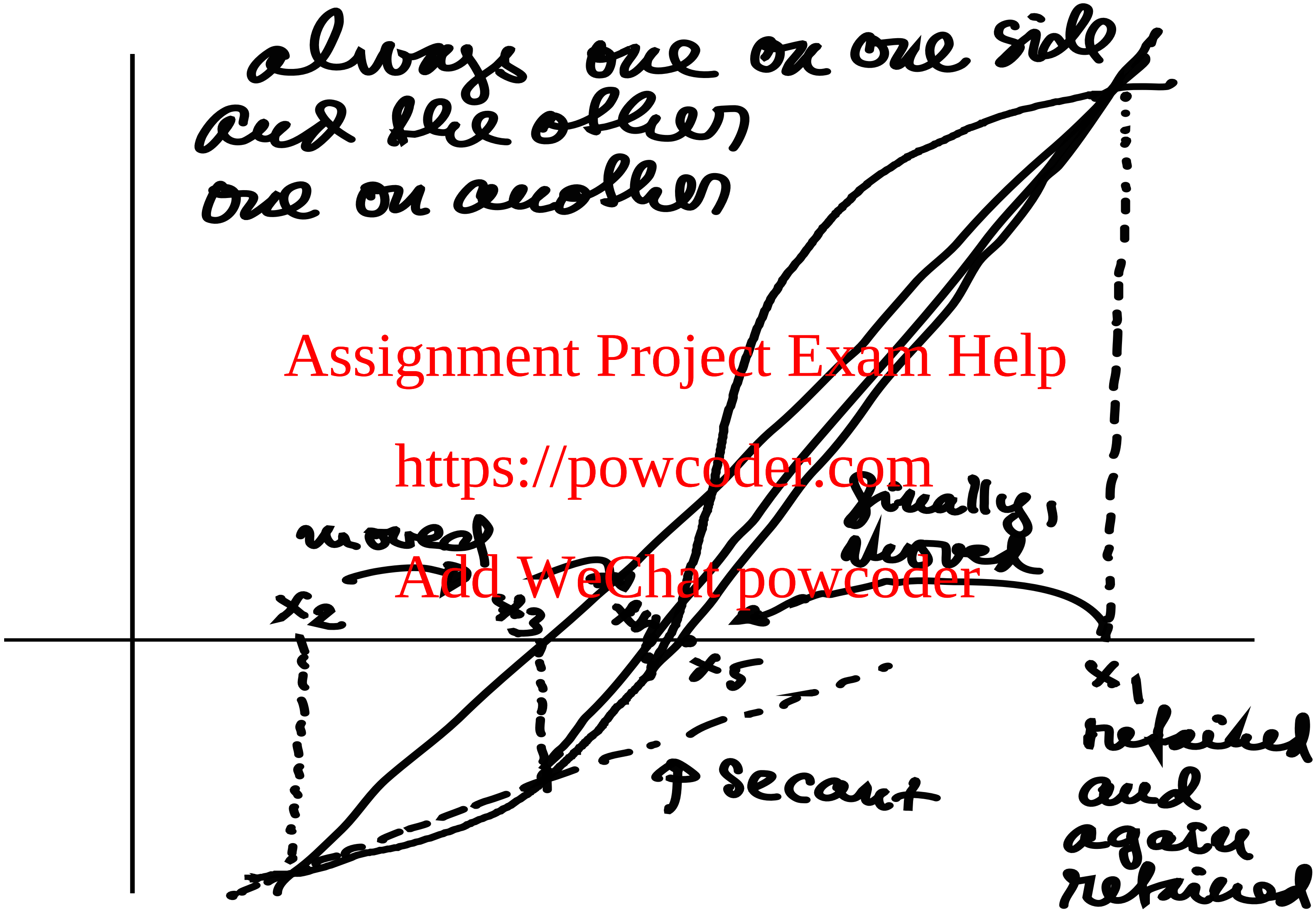
- and f does not have to be differentiable, only (perhaps) continuous

And finally, it has still *superlinear* convergence: worse than quadratic, but better than linear (yes, and in well-behaved situations: $f'(x_0) \neq 0$ etc.)

Neither Newton nor secant is safe: none of them guarantees a root.

There is, however, a modification of secant that does that

Regula falsi: modified secant (safe)



Start with two points on the opposite sides, $f(x_1)f(x_2) < 0$; and then drop x_1 if $f(x_2)f(x_3) < 0$, otherwise drop x_2 instead

Enough of dimension one

Would be fun to draw some more pictures, but: in the world of modern computational power, this has a limited appeal (only when such a method is used repeatedly in a more complex one)

Regula falsi: fine, but at times again the convergence is only linear

Its improvement, still safe, but better convergence: Illinois method

But: those safe methods are pretty much possible only in dim 1

And all those methods have been already invented and implemented

In R: function `uniroot`

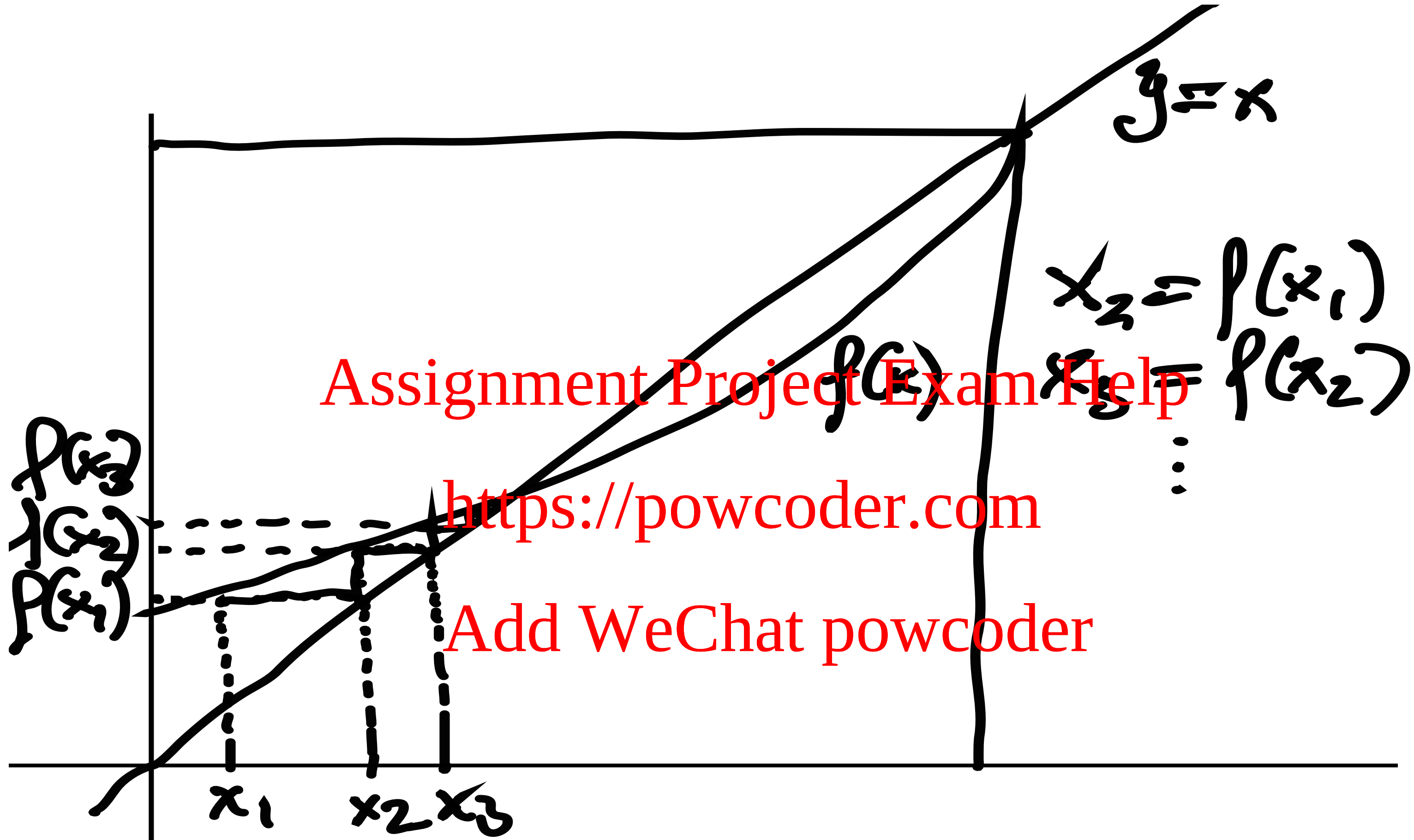
Brent's method `zeroin`, improving earlier Dekker's method

Combination of secant and bisection, using also quadratic instead of linear interpolation, and some other improvements

(Adopted - and slightly improved - first by MATLAB)

And what is it all good for, anyway?

Fixed points



$x_0 = f(x_0)$ $\quad |f'(x_0)| < 1$