Non-Linear Equations

So far we have focussed on linear equations. This is because we can solve these equations in a finite # of steps!

$$\underset{\approx}{A}\,\underset{\sim}{x} = \underset{\sim}{b} \quad \text{is solved in } O(n^3) \text{ steps!}$$

Often, however, our physical problem is non-linear!

Example: Suppose we are trying to fit some reaction rate data as a $f^{\underline{n}}(T)$. If we have a single irreversible reaction:

$$K = a\, e^{-E_a/kT}$$

k ← Boltzmann's cst

E_a ← activation energy

a ← pre-exp. factor

If we have a series of measured rxn rates $K_i(T_i)$ we can solve

120

for a & E_a using <u>linear</u> regression:

Transform eq'n:

$$\ln K = \ln a - \frac{E_a}{K}\frac{1}{T}$$

$$\underset{b}{\ln K} = \underset{x_1}{\ln a} - \underset{x_2}{\frac{E_a}{K}}\frac{1}{T}$$

So $\min_{\underset{\sim}{x}} \|(\underset{\approx}{A}\underset{\sim}{x} - \underset{\sim}{b})\|_2$

where $b \equiv \begin{pmatrix} \ln K_1 \\ \vdots \\ \ln K_n \end{pmatrix}$, $\underset{\approx}{A} = \begin{pmatrix} 1 & 1/T_1 \\ \vdots & \vdots \\ 1 & 1/T_n \end{pmatrix}$

So you know how to solve this. Suppose we have a more complex problem in which a substance can follow <u>2</u> rxn pathways. In this case:

$$K = a_1 e^{-E_{a_1}/KT} + a_2 e^{-E_{a_2}/KT}$$

You can't linearize this!

We can still set it up as a least squares problem:

$$\min_{\underset{\sim}{x}} \| K_i - K(T_i) \|_2 \equiv S(\underset{\sim}{x})$$

where $\underset{\sim}{x} \equiv \begin{pmatrix} a_1 \\ Ea_1 \\ a_2 \\ Ea_2 \end{pmatrix}$

We can determine the minimum by requiring:

$$\frac{\partial S(\underset{\sim}{x})}{\partial \underset{\sim}{x}} = 0$$: 4 eq'ns w/ 4 unknowns

But this is a non-linear problem

Non-linear problems are nasty because:

1) it may not have a unique solution! You may have multiple roots, local minima, etc.
For linear problems (non-singular) you

are <u>guaranteed</u> a unique solution!

2) For linear systems, you can solve it in a finite # of steps ($O(n^3)$)
For non-linear problems there is <u>no</u> guarantee - sometimes algorithms will fail to converge on <u>any</u> soln!

Let's focus on the problem $f(x)=0$

(we'll generalize this to the system $\underset{\sim}{f}(\underset{\sim}{x})=0$ later)

Let x^* satisfy $f(x^*)=0$

We will say $\bar{x}$ "solves" $f(x)=0$

if:

$$|f(\bar{x})| \approx 0 \quad \text{or} \quad |\bar{x}-x^*| \approx 0$$

less than some set tolerance

we need this dual def$^{\underline{n}}$ because $f(x)$ may have a steep slope near x^*

The approaches to solving $f(x)=0$ are iterative - we obtain a series (sequence) of approximations to the solution:

$$x_1, x_2, x_3, \ldots, x_n$$

which hopefully converges on x^*!

How do we do this?

The simplest approach is the method of bisection

Suppose $f(x)$ is continuous over the interval

$$x \in [a, b]$$

and that $f(a)f(b) < 0$

Then we know that there is at least one root in this interval!

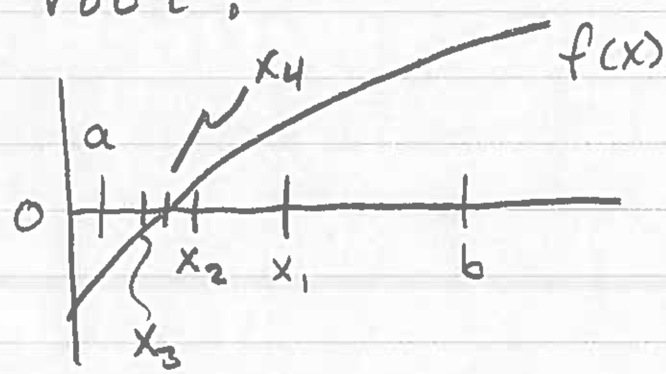
We can get within some interval of x^* in a finite number of steps!

We divide the interval in half and test to see which is satisfied:

$$f(a)f(m) < 0 \quad \underline{\text{or}} \quad f(m)f(b) < 0$$

where $m = \frac{a+b}{2}$

We just keep the half that has the root!



How fast does this converge?

Each iteration divides the interval in half, the midpoint is an estimate of x^*, thus:

$|m - x^*|$ is cut in half (approximately) at each iteration

The error is less than 10^{-7} of the original interval after about 24 iterations.

Let's take e_i as the error at the i^{th} iteration (e.g., $m - x^*$)

Then for bisection:

$$\frac{|e_{i+1}|}{|e_i|} \approx \frac{1}{2} \quad \text{on average}$$

In general, a method is said to converge at a rate r if:

$$\lim_{i\to\infty} \frac{|e_{i+1}|}{|e_i|^r} = C$$

In general, $|e_i| \ll 1$ so we want r to be large (>1) and C to be small.

If C is too large the method won't converge!

If $r = 1$ the method is <u>linear</u>

If $r = 2$ the method is <u>quadratic</u>

If $r > 1$ the method is <u>super linear</u>

For bisection, $r = 1$ (linear convergence)

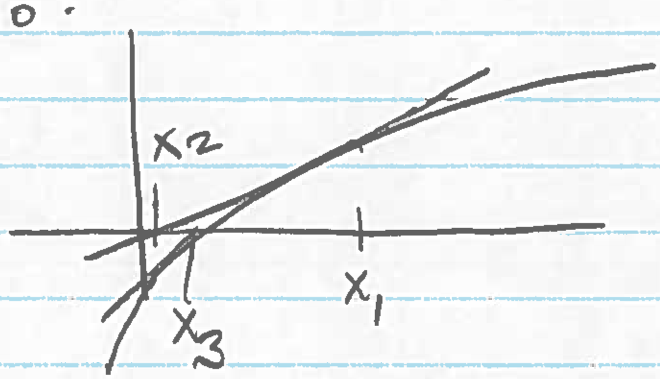
Let's look at a method with <u>faster</u> convergence: Newton's Method

In this method we compute <u>both</u> $f(x_i)$ and $f'(x_i)$

We fit the function with a tangent line (e.g., truncate the Taylor series after the linear term) and find its root.

This gives us our next guess!

So:



How do we get x_{i+1}?

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \ldots$$

truncate here

So set:

$$0 = f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Let's use this method to calculate $\sqrt{2}$:

This is equivalent to the root of

$$f(x) = x^2 - 2 = 0$$

Thus $f'(x) = 2x$

and $\dfrac{f(x)}{f'(x)} = \dfrac{x^2-2}{2x}$

so $x_{i+1} = x_i - \dfrac{x_i^2 - 2}{2x_i}$

We start at $x_0 = 1$

Thus $x_1 = 1 - \dfrac{1-2}{2} = 1.5$

$$x_2 = 1.5 - \frac{(1.5)^2 - 2}{3} = 1\tfrac{5}{12} = 1.41666\ldots$$

$$x_3 = 1\tfrac{5}{12} - \frac{(1\frac{5}{12})^2 - 2}{2\,(1\frac{5}{12})} = \frac{577}{408}$$

$$= 1.4142157\ldots$$

vs. exact sol'n $x^* = 1.4142136\ldots$

How fast did this converge?

$e_0 = 0.4142136$
$e_1 = 0.085786$
$e_2 = 0.002453038$
$e_3 = 2.138 \times 10^{-6}$

much faster than linear convergence!

Actually, it's quadratic:

$$\frac{e_1}{e_0^2} = 0.5$$

$$\frac{e_2}{e_1^2} = 0.333$$

$$\frac{e_3}{e_2^2} = 0.355\ldots$$

Let's look at this <u>in general</u>

we have:

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2} f''(\xi)(x - x_i)^2$$

Let $x = x^*$