Numerical Analysis

First topic: solving nonlinear equations.

Question: what is a nonlinear equation? Come in two forms:

1) A rootfinding problem: find x st $f(x) = 0$

2) A fixed point problem: find x st $g(x) = x$

In both cases, $f: \mathbb{R} \to \mathbb{R}$ is nonlinear, so is g.
Clearly, if f or g is linear, these equations are trivial to solve.

Question: Can we solve these problems, in general?

... what does that even mean?

E.g. Find x st $ax^2 + bx + c = 0$.

A quadratic equation... what's the solution?

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

✓ We can solve it.

✓ We have the answer. Exact... Analytic.

Can we always find an analytic solution to a nonlinear equation? No. Many examples.

E.g. Can solve up to 4th order polynomial equations, but not higher.

You can definitely get roots of high degree polynoms.
So... this "no analytic solution" problem might be a bit of a red herring.

If we change our mindset from "need exact/analytic" solution to need some correct digits, things become much more tractable. After all, computers have finite memory, so it's not really possible to represent a solution to infinite anyway (at least not in all cases).

> Numerical analysis is the study of algorithms for approximating solutions to the problems of continuous math (and by extension: physics).

§ Newton's method

Let's say we want to solve:

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

So this means find $x^* \in \mathbb{R}$ satisfying the rootfinding equation. Or... just satisfies it approximately.

Recall: the Taylor expansion of f about x with increment h is:

$$f(x+h) = f(x) + f'(x)h + \cdots + \frac{h^n}{n!} f^{(n)}(x) + R_n$$

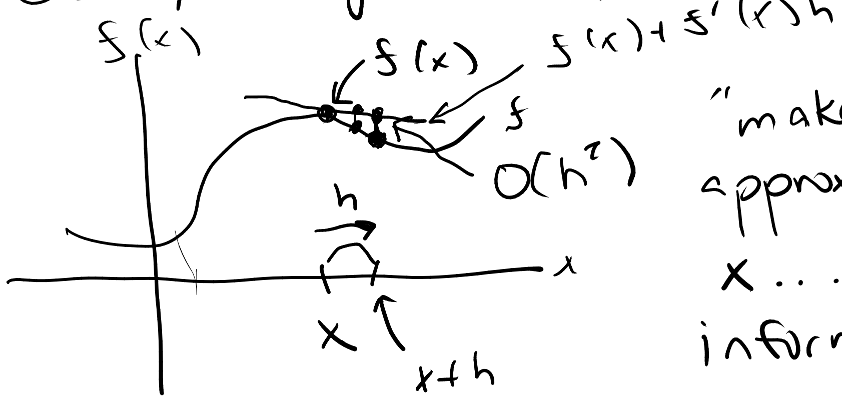
where the remainder R_n satisfies:

$$|R_n| = O(\underline{h^{n+1}}).$$

<u>Recall</u>, the <u>Landau notation</u> (or "big O" notation) $f(h) = O(g(h))$ means:

$$\lim_{h\to 0} \frac{|f(h)|}{|g(h)|} < \infty.$$

Idea w/ Taylor expansion:



"make a local polynomial approximation to f about x... using derivative information of f at x"

1st order T. poly.: $f(x) + f'(x)h + \underbrace{O(h^2)}$

tells you the <u>rate</u> at which the error goes to 0 as $h \to 0$

2nd order T. poly: $f(x) + f'(x)h + \frac{h^2}{2}f''(x) + O(h^3)$

This assumes some "regularity" on f: namely, derivatives need to exist, and they can't be too crazy. Or at least, the exact error depends on the $(n+1)$st derivative.

Let's try to develop an algorithm for finding x^* st $f(x^*) = 0$.

We will use a central idea in numerical analysis: relaxation. "Relax exact equation to get easier-to-solve approximate equation."

Let's replace x^* with $x_n + \Delta x_n$:

- introduce a sequence $x_0, x_1, \dots$
- goal: $x_n \to x^*$ as $n \to \infty$
- notation: $\Delta x_n = x_{n+1} - x_n$
- refer to x_n as the "iterate"

When we relax, we replace x^* w/ $x_n + \Delta x_n$.
Assuming tacitly that $x^* \approx x_n + \Delta x_n$.
(Should be true by def. as $n \to \infty$)

Where does that leave us? Let's see:

$$0 \overset{\text{Relaxation}}{=} f(x_n + \Delta x_n) \overset{\text{TE}}{=} f(x_n) + f'(x_n)\Delta x_n + O(\Delta x_n^2)$$

$$\approx \underbrace{f(x_n) + f'(x_n)\Delta x_n}_{\text{Truncate and approximate w/ 1st order Taylor poly.}}.$$

Altogether, we get:

$$0 = f(x_n) + f'(x_n)\Delta x_n, \quad n=0,1,\ldots$$

or:

$$\Delta x_n = -\frac{f(x_n)}{f'(x_n)}$$

or:

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.}$$

This is the Newton iteration.

Technically, if we have chosen x_0, then

$$\begin{cases} x_0 = \text{fixed} \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}, \quad n=0,1,\ldots \end{cases}$$

is Newton's method.

Notice: not too hard to see that:

$$x^* - x_n = O(\Delta x_n^2)$$

under certain circumstences. We'll come back to this. This tells you how fast NM converges... <u>if</u> it converges!