18.335 / 6.7310 Introduction to Numerical Methods

Course Overview

- handouts, syllabus, psets, etc.
  - Canvas + Piazza + Gradescope
  - github.com/mitmath/18335

Numerical analysis:

Fast algorithms for approximately solving the problem of "continuous math" $\Big\}$ math + time + accuracy

Example: nonlinear equations

$f: \mathbb{R} \to \mathbb{R}$, f is continuous

Find $f(x) = 0$

- f is a general polynomial of degree ≥ 5:

  no root formula using $+, -, \times, \div, \sqrt[n]{\ }$

- Goal: approximately find the root using only $+, -, \times, \div$

- <u>Naive method: Bisection method</u>

  Intermediate value theorem:

  $f(a) f(b) < 0 \Rightarrow \exists$ root $x^* \in [a, b]$

$a_0 = a.\quad b_0 = b$

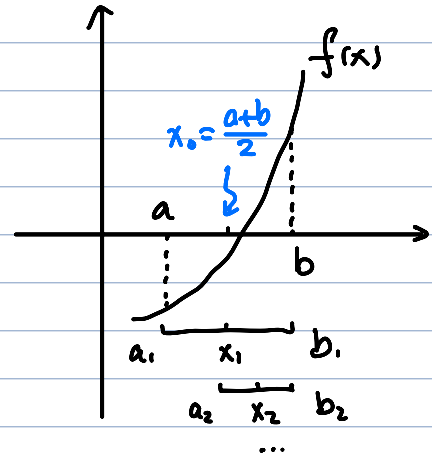
For $n = 0, 1, 2, \dots$

$$x_n = \frac{a_n + b_n}{2}$$

if $\mathrm{sign}(f(x_n)) = \mathrm{sign}(f(a_n))$

$a_{n+1} = x_n\ ,\quad b_{n+1} = b_n$

else $a_{n+1} = a_n\ ,\quad b_{n+1} = x_n$



Error analysis:

$$e_n := |x_n - x^*| \le \frac{1}{2}(b_n - a_n) = \frac{b-a}{2^n} \to 0 \quad \text{as} \quad n \to +\infty$$

$e_{n+1} \approx \frac{1}{2} e_n$ Linear convergence

Can we do it faster? (Have only access to $f(x)$)

Yes. One possible method:

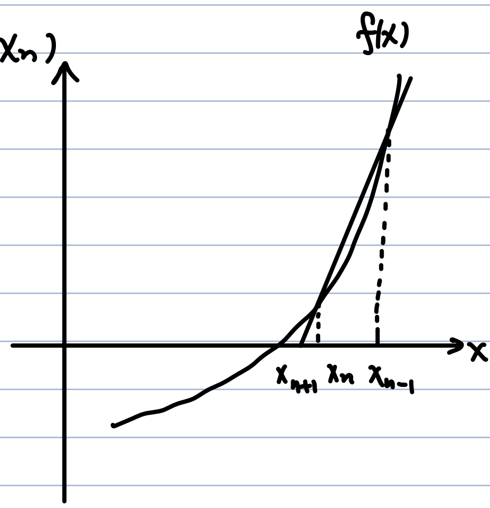
- <u>Secant method</u>

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

$$\approx f(x_n) + \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_n)$$

$$= 0$$

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



Error analysis:

$$e_{n+1} := x_{n+1} - x^* = (x_n - x^*) - \frac{f(x_n)[(x_n - x^*) - (x_{n-1} - x^*)]}{f(x_n) - f(x_{n-1})}$$

$$= e_n - \frac{f(x_n)(e_n - e_{n-1})}{f(x_n) - f(x_{n-1})}$$

$$= \frac{f(x_n)e_{n-1} - f(x_{n-1})e_n}{f(x_n) - f(x_{n-1})}$$

$$= \underbrace{\frac{f(x_n)e_n^{-1} - f(x_{n-1})e_{n-1}^{-1}}{f(x_n) - f(x_{n-1})}}_{\text{claim : } C_n \approx C} e_n e_{n-1}$$

check: $f(x_n) \approx f'(x^*)e_n + \frac{1}{2}f''(x^*)e_n^2$

$f(x_{n-1}) \approx f'(x^*)e_{n-1} + \frac{1}{2}f''(x^*)e_{n-1}^2$

$$C_n = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \cdot \boxed{\frac{f(x_n)e_n^{-1} - f(x_{n-1})e_{n-1}^{-1}}{x_n - x_{n-1}}}$$

$$\boxed{\approx \frac{1}{2}f''(x^*)\underbrace{\frac{e_n - e_{n-1}}{x_n - x_{n-1}}}_{=1}}$$

$$\approx \frac{1}{f'(x^*)} \cdot \frac{1}{2}f''(x^*) =: C$$

We arrive at

$$e_{n+1} = C\, e_n e_{n-1}$$

Let $e_{n+1} \approx A e_n^p \quad (p > 1)$

$$Ae_n^p \approx c\, e_n \left(\frac{e_n}{A}\right)^{1/p} = \frac{c}{A^{1/p}} e_n^{1+\frac{1}{p}}$$

$$\Rightarrow A^{1+1/p} = c\, e_n^{1+1/p-p}$$

Since $e_n \to 0$ as $n \to +\infty$,

we should have $1 + \frac{1}{p} - p = 0 \Rightarrow p = \frac{1+\sqrt{5}}{2} \approx 1.618$

That is $e_{n+1} \approx c^{0.618} e_n^{1.618}$ faster than bisection

super linear convergence $\square$

most fundamental part

numerical analysis $\begin{cases} \text{nonlinear equations} \\ \text{data interpolation/approximation} \\ \text{numerical integration/differentiation} \\ \text{solution of ODE's/PDE's} \\ \text{etc.} \end{cases}$

numerical linear algebra (NLA)

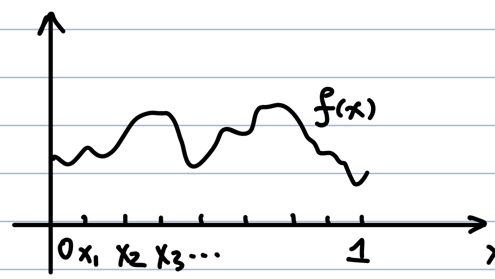
Example: Numerical methods for PDEs

a) Poisson equations

$$\begin{cases} \Delta u := \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f \\ u|_{\partial\Omega} = 0 \end{cases}$$

In 1D, $\dfrac{d^2 u}{dx^2} = f$

$u(0) = u(1) = 0$



Discretization:

$u(x_i) \approx u_i$

$f(x_i) \approx f_i$

$h = x_{i+1} - x_i$

$$\frac{d^2 u}{dx^2}(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$\Longrightarrow \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & & \\ & & & & 1 \\ & & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

Solve large linear system

$$Au = f$$

b) Schrödinger Equation

$$-\Delta u + Vu = Eu$$

u: wave function

V: potential

E: Quantized energy
(eigenvalues)

· Discretization : (1D)

$$\frac{1}{h^2}\begin{bmatrix} 2+V_1h^2 & -1 & & \\ -1 & 2+V_2h^2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2+V_nh^2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = E\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

$$Au = Eu$$

Solve large eigenvalue problems

NLA

Outline

fundamentals of numerical analysis

Floating-point arithmetic
matrix norms
backward error analysis
matrix factorization, etc.

Direct method: Gauss / Cholesky
Iterative method: Jacobi / G-S.
etc.

Given $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$

Solve $Ax = b$

Given $A \in \mathbb{R}^{n\times n}$

Find all (λ, x)
s.t. $Ax = \lambda x$

Power method
QR iteration

Krylov subspace methods

Arnoldi
GMRES
CG / BiCG, etc.

Other topics: Nonlinear optimization
Monte Carlo sampling
Randomized algorithm etc.