11 PDEs and Sparse Linear Algebra

Each large topics, won't do them justice!

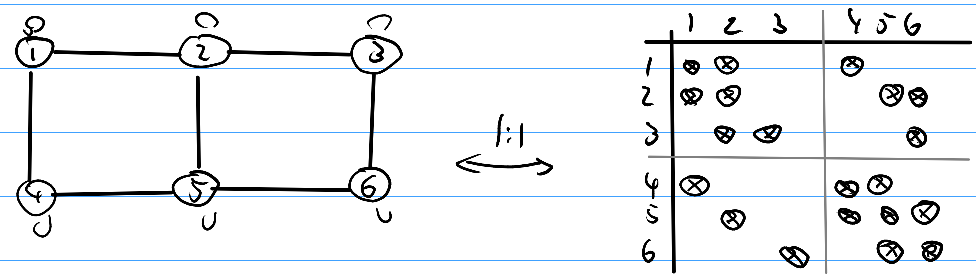
Entire classes at UIUC for each:

CS 555 → Numerical Methods for PDEs
CS 556 → Iterative and Multigrid Methods

Would love to see you in one of these classes!

11, Part 1 Solving sparse linear systems

Sparse matrices and graphs (and meshes)



Terminology: neighbor, edge

- Solving $Ax=b$ has been our bread and butter. So what's different now?

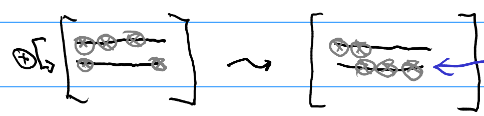
- How are we going to solve this?

- What's potentially problematic with Gaussian elimination?

Demo sparse fill (pt. 1)

- How do we fix this?

Understanding sparse fill

Graph equivalent of  worse!

→ Ordering! (also in demo)

Demo sparse fill (pt. 2)

Pro/con of sparse factorization:

- ⊕ robust
- ⊖ storage cost
- ⊖ computation cost

→ not viable for really large (3-D?) problems)

Idea: Don't try to factorize the matrix.
Try to iterate closer to a solution.

(11.1) "Stationary" iterative methods

Goal: Solve $Ax=b$ by iteration

Idea: Split $A = M - N$

what we're actually inverting in each step

$$M x_{k+1} = N x_k + b$$

"stationary": do the same thing every iteration

$$\rightsquigarrow \quad x_{k+1} = M^{-1} N x_k + M^{-1} b$$

A fixed point iteration.

When convergent? If contractive, i.e. $\rho(M^{-1}N) < 1$.

Why have the b term? What would we converge to without it?

O[Which M is easy to invert? Diagonal!

Jacobi method: $M = D$ $\qquad N = -(L+U)$

does not always converge (but does if diagonal is large enough)

$$x_{k+1} = D^{-1}\left(-(L+U)x_k + b\right)$$

$$(x_{k+1})_i = \frac{b_i - \sum_{j\neq i} a_{ij}(x_k)_j}{a_{ii}}$$

Demo Stationary methods

Gauss-Seidel: $M = D + L$ $\qquad N = -U$

SOR: $M = \frac{1}{\omega}D + L$ $\qquad N = \left(\frac{1}{\omega} - 1\right)D - U$

(11.2) Conjugate gradient method

Assume: A symmetric positive definite

Goal: Minimize $\varphi(x) = \frac{1}{2} x^T A x - x^T b \iff$ solve $Ax = b$

General form: $x_0 = \langle \text{starting vector} \rangle$

$x_{k+1} = x_k + \alpha_k s_k$ s_k: search directions

○ [Remaining questions: What's α? What's s_k?

What's α?

$$0 \overset{!}{=} \frac{\partial}{\partial \alpha} \varphi(x_k + \alpha_k s_k) \qquad \nabla \phi(x) = Ax - b = -r \quad \text{(residual)}$$
$$= \nabla \varphi(x_{k+1}) \cdot s_k = r_{k+1} \cdot s_k$$

○ [Choose α so that the next residual is $\perp$ to current search direction.

$e_k = x_k - x^*$

$r_k = -Ae_k$

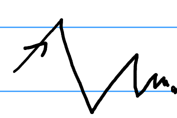
$$r_{k+1} = r_k + \alpha_k A s_k$$
$$0 \overset{!}{=} s_k^T r_{k+1} = s_k^T r_k + s_k^T \alpha_k A s_k \leadsto \alpha_k = \frac{s_k^T r_k}{s_k^T A s_k} = -\frac{s_k^T A e_k}{s^T A s_k} \quad (**)$$

What's s_k?

$$s_k \overset{?}{=} -r_k = \overbrace{-\nabla \varphi(x_k)}^{②}$$

① ○ [Do we know this method? Is it any good?

○ [Yes, steepest descent. And, no.



○ [Problem: overshoots, uses same search directions over and over.

Terminology: x, y are "A-orthogonal" or "conjugate" iff $x^T A y = 0$

Idea: Require $s_i^T A s_j = 0$ if $i \neq j$. (why A-orthogonal? Want to relate to (xx).)

Then $e_0 = x_0 - x^* = \sum_i \delta_i s_i$ ⊗

○ [I.e. entire initial error is lin. comb. of search directions!

Want to understand ⊗ → Find δ_i's.

$$s_k^T A e_0 = \sum_i \delta_i s_k^T A s_i \overset{s_k \text{ conj.}}{=} \delta_k s_k^T A s_k$$

$$\rightsquigarrow \quad \delta_k = \frac{s_k^T A e_0}{s_k^T A s_k} \underset{s_k \text{ conj.}}{=} \frac{s_k^T A \left(e_0 + \sum_{i=1}^{k-1} \alpha_i s_i\right)}{s_k^T A s_k} = \frac{s_k^T A e_k}{s_k^T A s_k} = -\alpha_k!$$

○ [What's happening here? "Zap" one of n error components every step.

○ [Guaranteed done after n steps!

○ [Where do the search directions come from?

Idea 2: Can build those conjugate directions from Krylov space using three-term Lanczos-type iteration

→ cheap!

11 part 2 **Partial Differential Equations**

→ increasing laziness :)

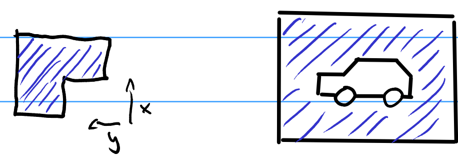
Notation: $\frac{\partial}{\partial x} u = \partial_x u = u_x$

PDE: Equation with multiple partial derivatives, e.g.

$$\partial_x^2 u + \partial_y^2 u = 0$$ → solution must depend on x, y

Examples: Waves, EM, diffusion

- If multiple spatial derivatives, possibly nontrivial geometry:

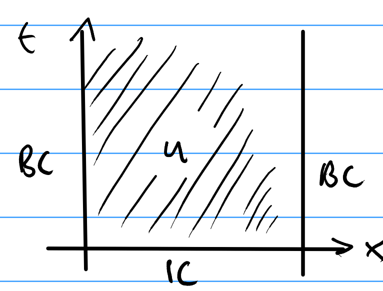


with boundary conditions (Dirichlet, Neumann, etc.)

- Sometimes, one variable is time-like (rather than space-like)

Q [What makes a variable time-like?

- Causality
- One-sided BCs (ICs)
- No geometry (in t)



- Time-dep. PDEs give rise to a steady-state PDE

$$u_t = f(u_x, u_y, u_{xx}, u_{yy})$$

$$0 = f(u_x, u_y, u_{xx}, u_{yy})$$

- Idea: Discretize spatial derivatives first
  → large system of ODEs ("semidiscrete system")
  → use ODE solver from Ch 9
  (implicit/explicit...)

Demo: time-dependent PDEs

- Notation: "Laplacian"

$$\Delta u = \operatorname{div}\operatorname{grad} u = \nabla\cdot(\nabla u) = \overbrace{\underbrace{\underbrace{u_{xx}}_{1D} + u_{yy}}_{2D} + u_{zz}}^{3D}$$

Three main types of PDEs:

- hyperbolic (wave-like) — conserves energy
  - → first order $u_t = f(u_x)$
  - → second-order wave eqn. $u_{tt} = \Delta u$
    - equivalent: $u_t = v_x$, $v_t = u_x$
    - → kind of first order

- parabolic (heat-like) → $u_t = \Delta u$
  - dissipates energy

- elliptic (steady-state) → $\Delta u = 0$ (no time)
  - of heat and wave
  - → pure BVP, results in (sparse) linear system
  - similar to 1D BVPs
  - same methods apply (FD, Galerkin, etc.)