Shooting Method

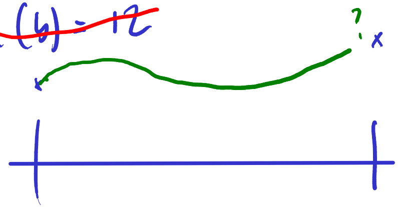
Idea: Want to make use of the fact that we can already solve IVPs.
Problem: Don't know *all* left BCs.

Demo: Shooting Method

$u''(x) = f(x)$

$u(a) = 15 \quad u'(a) = ?$

$\cancel{u(b) = 12}$



- What about systems?
- What are some downsides of this method?
- What's an alternative approach?

Finite Difference Method

$u''(x) + p(x)\,u'(x) = f(x)$

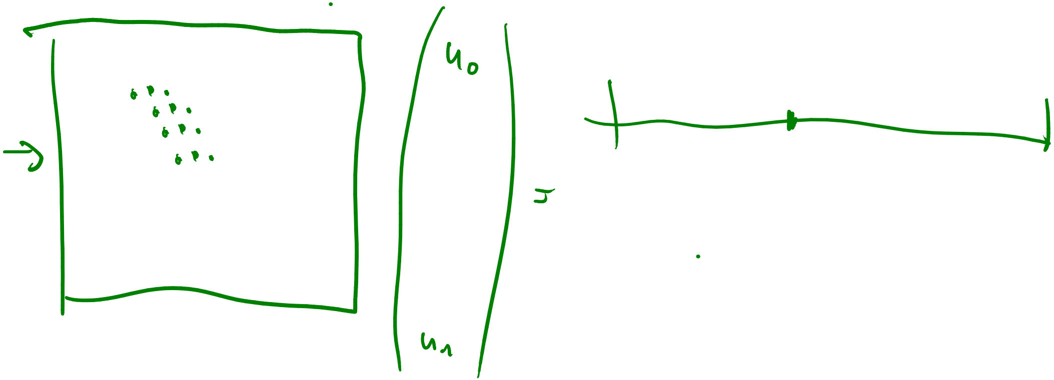
Idea: Replace u' and u'' with finite differences.
For example: second-order centered

$$u'(x) = \frac{u(x+h) - u(x-h)}{2h} + O(h^2)$$

$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2)$$

Demo: Finite differences

- What happens for a nonlinear ODE?



Collocation Method

Consider

$$(*)\begin{cases} y'(x) = f(y(x)), & \text{for all } x \\ g(y(a), y(b)) = 0. \end{cases}$$

(Scalar for simplicity–vector generalization is straightforward.)

- What can we do?

$$y(x) = \sum_{i=1}^{n} \alpha_i T_i(x)$$

solve

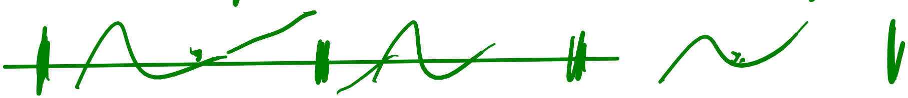
$$y'(x) = f(y)$$

$$\otimes \quad \sum \alpha_i T_i'(x) = f\Big(\sum_i \alpha_i T_i\Big)(x)$$

$$u''(x) = f(x)$$

Collocation: make $\otimes$ true at $(x_0, \dots, x_{n-1})$.

→ not a sparse matrix → $O(n^3)$ solve cost



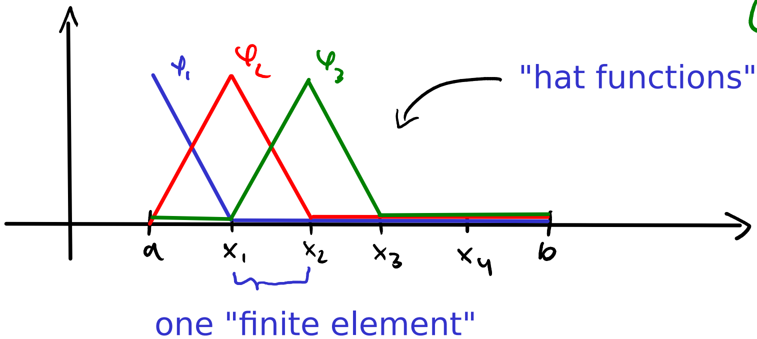
Galerkin/Finite Element Method

$$u''(x) = f(x), \qquad u(a) = u(b) = 0.$$

$$\int_a^b u'' \varphi = \int_a^b f \varphi$$

$$-\int u' \varphi' = \int_a^b f \varphi$$

Problem with collocation: Big dense matrix.
Idea: Use piecewise basis. Maybe it'll be sparse.



- What's the problem with that?

11 Partial Differential Equations and Sparse Linear Algebra

Remark: Both PDEs and Sparse Linear Algebra are big topics. Will only scratch the surface here. Want to know more?

- CS555 $\rightarrow$ Numerical Methods for PDEs ✓
- CS556 $\rightarrow$ Iterative and Multigrid Methods ✓

We would love to see you there! :)

11.1 Sparse Linear Algebra

Solving Sparse Linear Systems

Solving $A\boldsymbol{x} = \boldsymbol{b}$ has been our bread and butter.

Typical approach: Use factorization (like LU or Cholesky)
Why is this problematic?

Demo: Sparse Matrix Factorizations and "Fill-In"

Idea: Don't factorize, iterate.

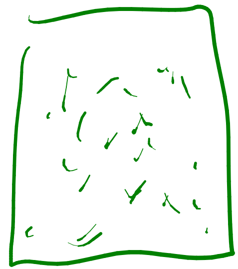
'Stationary' Iterative Methods

Idea: Invert only part of the matrix in each iteration. Split

$$A = M - N,$$

where M is the part that we are actually inverting.

A



- When do these methods converge?
- What could we choose for M (so that it's easy to invert)?

$$Ax = b$$

$$M - N$$

$$Mx_{k+1} = Nx_k + b$$

$$x_{k+1} = M^{-1}(Nx_k + b) \qquad \rho(M^{-1}N) < 1$$

$$|\lambda(M^{-1}N)| > 1$$

Demo: Stationary Methods

Hestenes and Stiefel

Conjugate Gradient Method

Assume A is symmetric positive definite.

Idea: View solving $A\boldsymbol{x} = \boldsymbol{b}$ as an optimization problem.

$$\text{Minimize} \quad \varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b} \quad \Leftrightarrow \quad \text{Solve} \quad A\boldsymbol{x} = \boldsymbol{b}.$$

$f(x) = x^2 + (y-3)^2$

Observe $-\nabla\varphi(\boldsymbol{x}) = \boldsymbol{b} - A\boldsymbol{x} = \boldsymbol{r}$ (residual).

Use an iterative procedure ($\boldsymbol{s}_k$ is the search direction):

$$\begin{aligned} \boldsymbol{x}_0 &= \langle\text{starting vector}\rangle \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k, \end{aligned}$$

- What should we choose for α_k (assuming we know $\boldsymbol{s}_k$)?
- What should we choose for $\boldsymbol{s}_k$?

$$0 = \frac{\partial}{\partial \alpha} \varphi(x_k + \alpha_k s_k) = \underbrace{\nabla \varphi(x_{k+1})} \cdot s_k$$

$$= r_{k+1} \cdot s_k$$

$$r_{k+1} = r_k + \alpha A s_k$$

$$0 = s_k^T r_{k+1} = s_k^T r_k + \alpha s_k^T A s_k \quad \rightsquigarrow \quad \alpha_k = \frac{s_k^T r_k}{s_k^T A s_k} = -\frac{s_k^T A e_k}{s^T A s_k}$$

$$e_k = x_k - x^* \qquad r_k = -A e_k$$

$s_1 =$ (direction of steepest desc) v_0

$$e_0 = x_0 - x^* = \sum_i \underbrace{\delta_i}_{-\alpha_k} s_i$$

$$s_i^T A s_j = 0 \quad (\text{if } i \neq j)$$

Demo: Conjugate Gradient Method

11.2 PDEs