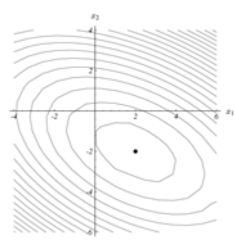
Iterative Linear System Solvers for Large, Sparse Matrices

Iterative Methods

Mostly for SPD systems

- conjugate gradient and its variants

$$
\begin{aligned}
Ax &= b \\
f(x) &= \frac{1}{2}x^T Ax - b^T x \\
\nabla f &= Ax - b = 0
\end{aligned}
$$



- minimize through successive line searches

Steepest Descent

Decrease function

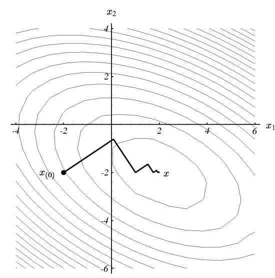
$$f(\underbrace{x^{(i)} + \alpha^{(i)} r^{(i)}}_{x^{(i+1)}})$$

- negative of gradient

$$-f'(x^{(i)}) = b - Ax^{(i)} = r^{(i)}$$

- how much?

$$\frac{\partial f}{\partial \alpha}\bigg|_{x^{(i+1)}} = 0 = -r^{(i+1)^T} r^{(i)}$$



Steepest Descent

How much?

$$r^{(i+1)^T} r^{(i)} = 0$$

$$\rightsquigarrow \quad (b - Ax^{(i+1)})^T r^{(i)} = 0$$

$$\rightsquigarrow \quad (b - Ax^{(i)})^T r^{(i)} = \alpha^{(i)} r^{(i)^T} A r^{(i)}$$

$$\rightsquigarrow \quad \alpha^{(i)} = \frac{r^{(i)^T} r^{(i)}}{r^{(i)^T} A r^{(i)}}$$

Steepest Descent

Algorithm

$$r^{(i)} = b - Ax^{(i)}$$

$$\alpha^{(i)} = \frac{r^{(i)T} r^{(i)}}{r^{(i)T} A r^{(i)}}$$

$$x^{(i+1)} = x^{(i)} + \alpha^{(i)} r^{(i)}$$

$$*(-A)$$

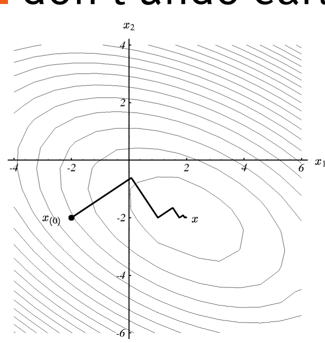
$$+b$$

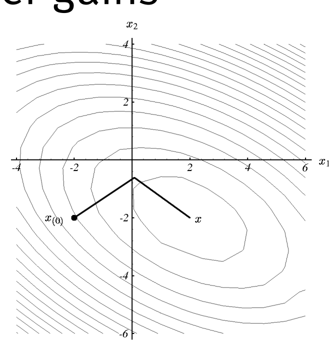
$$r^{(i+1)} = r^{(i)} - \alpha^{(i)} A r^{(i)}$$

Let's do Better

Conjugate directions

- don't undo earlier gains





Conjugate Gradient

$d_i^T A d_j = 0$

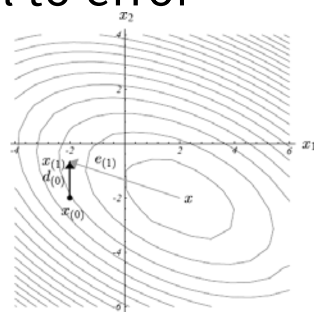
Directions A-orthogonal to error

$$x^{(i+1)} = x^{(i)} + \alpha^{(i)} d^{(i)}$$

$$\frac{\partial f}{\partial \alpha}\Big|_{x^{(i+1)}} = 0 \quad = \quad -r^{(i+1)^T} d^{(i)}$$

$$= \quad d^{(i)^T} A e^{(i+1)}$$

$$\alpha^{(i)} = -\frac{d^{(i)^T} A e^{(i)}}{d^{(i)^T} A d^{(i)}} = \frac{d^{(i)^T} r^{(i)}}{d^{(i)^T} A d^{(i)}}$$



Conjugate Gradient

Next direction?

- orthogonal to all others
- easy update:

$$d^{(i+1)} = r^{(i+1)} + \frac{r^{(i+1)^T} r^{(i+1)}}{r^{(i)^T} r^{(i)}} d^{(i)}$$

Algorithm

What happened to d? (see section 7.3 of Shewchuk)

$$d^{(0)} = r^{(0)} = b - Ax^{(0)}$$

$$\alpha^{(i)} = {r^{(i)}}^T r^{(i)} / {d^{(i)}}^T A d^{(i)}$$

$$x^{(i+1)} = x^{(i)} + \alpha^{(i)} d^{(i)}$$

$$r^{(i+1)} = r^{(i)} - \alpha^{(i)} A d^{(i)}$$

$$\beta^{(i+1)} = {r^{(i+1)}}^T r^{(i+1)} / {r^{(i)}}^T r^{(i)}$$

$$d^{(i+1)} = r^{(i+1)} + \beta^{(i+1)} d^{(i)}$$

Sweeps
- **matrix multiply**
- **dot product acc.**
- **update**
- **dot product acc.**
- **update**

Iterative Methods

What you'll need

- you supply Ax
- given topological datastructure
  - implement matrix multiplies
- stopping criterion: relative residual

Implementation

What lives where?

- degrees of freedom at vertices
  - need additional variables for solver
- matrix entries on edges

Matrix multiplies

- iterate over mesh!
- mesh is sparse matrix data structure

Iterative Methods

Other issues

- number of iterations proportional to κ (steepest descent), $\sqrt{\kappa}$ (CG)
- preconditioning
  - use approximate inverse
    - diagonal (Jacobi)
    - incomplete Cholesky, hierarchical,...

$\kappa(M^{-1}A) = 1$

Other System Types

SPD

- conjugate gradients
- Shewchuk, “CG w/o the Agony”

non-S PD & non-S non-PD

- bi-conjugate gradients
- see Numerical Recipes
  - http://www.nr.com/ (chapter 2.7)