Conjugate Gradient Algorithm (M. Hestenes, E. Stiefel)

Assume linear system $Ax = b$,

$A \in \mathbb{R}^{n \times n}$ symmetric $A^T = A$,

pos def $x^T A x > 0 \quad \forall \; 0 \neq x \in \mathbb{R}^n$

Viewpoint A (Direct Method)

- Two vectors $u, v \in \mathbb{R}^n$ are conjugate if

$$u^T A v = 0 \qquad u^T v = 0$$

- These properties of A allow to define an inner product on $\mathbb{R}^n$

$$u^T A v = (u, v)_A = (Au, v)$$
$$= (u, A^T v)$$
$$= (u, Av)$$

- Two vectors are conjugate iff they are orthogonal wrt this inner product

$$(u, v)_A = 0 \iff u \perp v$$

- Let $S = \{s_1, \dots, s_n\}$ be a set of n mutually conjugate vectors wrt A

$\Leftrightarrow s_i^T A s_j = 0 \quad \forall \; i \neq j$

$\Rightarrow S$ forms a basis of $\mathbb{R}^n$

$$\Rightarrow x_* = \sum_{i=1}^{n} \alpha_i s_i \Rightarrow \underbrace{A x_*}_{= b} = \sum_{i=1}^{n} \alpha_i A s_i$$

$$s_j^T b = s_j^T A x_* = \sum_{i=1}^{n} \alpha_i s_j^T A s_i = \sum_{i=1}^{n} \alpha_i (s_j, s_i)_A$$

$$= \alpha_j (s_j, s_j)_A$$

$$\Rightarrow \alpha_j = \frac{(s_j, b)}{(s_j, s_j)_A}$$

Direct Method

Find a sequence of n conj directions and compute α_j

Lizbeth

assume A diag $\begin{bmatrix} a_1 & & \\ & a_2 & \\ & & a_n \end{bmatrix}$, $a_n > 0$

$$\Rightarrow x^T A x = [x_1, \dots, x_n] A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} a_n x_n^2 = F(x)$$

n-dimensional analogue of a parabola. What do the extrema of $F(x)$?

$$\nabla F(x) \stackrel{!}{=} 0$$

$$= \begin{bmatrix} 2a_1 x_1 \\ \vdots \\ 2a_n x_n \end{bmatrix} \Rightarrow x = 0$$

Viewpoint B (Iterative Methods)

The solution x_* is the unique minimizer of the quadratic function

$$\boxed{f(x) = \frac{1}{2} x^T A x - x^T b}, \quad x, b \in \mathbb{R}^n$$

$\exists$ unique minimizer

$\Leftrightarrow$ Hessian matrix is symmetric, positive definite

$$H(f(x)) = A$$

That the minimizer solves the initial problem follows from 1st derivative

$$\nabla f(x) = \frac{1}{2}(A^T + A)x - b = Ax - b \stackrel{!}{=} 0$$

This suggests

$$s_0 = -\nabla f(x)\big|_{x=x_0}$$

$$\Rightarrow s_0 = b - Ax_0$$

and the other vectors will be conjugate to this gradient

$\Rightarrow r_k = b - Ax_k$ with to $r_k = -\nabla f(x)|_{x=x_k}$

Remark

A gradient descent algo since movement in r_k direction

But we insist that these directions s_k must be conjugate to eachother

Requirements

Next search direction built out of current residual and all previous search dimensions

Constraint

Conjugation $\Leftrightarrow$ orthonormal - type

$\Rightarrow s_k = r_k - \sum_{i<k} \frac{(r_k, s_i)_A}{(s_i, s_i)_A} \cdot s_i$

$\Rightarrow$

$$x_{k+1} = x_k + \alpha s_k$$

$$\alpha_k = \frac{s_k^T (b - A x_k)}{s_k^T A s_k}$$

$$= \frac{s_k^T r_k}{s_k^T A s_k}$$

$$= \frac{(s_k, r_k)}{(s_k, s_k)_A}$$

Proof

$$f(x_{k+1}) = f(x_k + \alpha_k s_k) = g(\alpha_k)$$

$$g'(\alpha_k) = 0$$

$$= \frac{1}{2}(x_k + \alpha_k s_k)^T A (x_k + \alpha_k s_k) - (x_k + \alpha_k s_k)^T b$$

$$= f(x_k) + \alpha_k^2 \frac{1}{2} s_k^T A s_k - \alpha_k s_k^T b + \alpha_k \underbrace{\left(\frac{1}{2} x_k^T A s_k + \frac{1}{2} s_k^T A x_k\right)}_{= \alpha_k s_k^T A x_k}$$

$$\Rightarrow g'(\alpha_k) = \alpha_k s_k^T A s_k - s_k^T (b - A x_k)$$
$$\stackrel{!}{=} 0$$

Reduce computational complexity by making use of base properties of our system

for $i \neq j$ $\quad r_i \perp r_j \iff r_i^T r_j = 0$

A orthogonality $\quad s_i \perp s_j \iff s_i^T A s_j = 0$

$\Rightarrow$ s_i and r_i span the same Krylov subspace where r_i form an orthonormal basis wrt std inner product and s_i wrt inner product induced by A

$\Rightarrow$ x_k are projections of x onto Krylov subspace.

If CG starts with $x_0 = 0$, then

$$x_k = \underset{y \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ (x-y)^T A (x-y) \;\middle|\; y \in \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\} \right\}$$

Conjugate Gradient Algorithm

(For real, symmetric, pos def A)

$r_0 = b - Ax_0$

$s_0 = r_0$

FOR $k = 0, 1, \ldots$

$$\alpha_k = \frac{r_k^T r_k}{s_k^T A s_k}$$

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

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

BREAK LOOP if r_{k+1} is sufficiently small

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$s_{k+1} = r_{k+1} + \beta_k s_k$$

END

return x_{k+1}

Preconditioning Strategies

what if $A \neq A^T$

$Ax = b$ $\quad A \neq A^T$ $\quad \text{span}\{b, Ab, \dots\}$

$A^T A x = A^T b$

$\rightsquigarrow Bx = c$ $\quad \text{span}\{c, Bc, \dots\}$

$1 \leq \text{cond}(A) < \text{cond}(B)$

$\text{cond}(A \cdot A^T) = \text{cond}(A^2) \approx (\text{cond}(A))^2$

Overview

$Ax = b$ $\quad A = A^T$, pos def

$\Rightarrow$ solve w CG Method

$$\underbrace{M^{-1}Ax}_{\tilde{A}} = \underbrace{M^{-1}b}_{\tilde{b}}$$

$$A \longmapsto M^{-1}A$$

$$\{\lambda_A\} \longmapsto \{\lambda_{M^{-1}A}\}$$

$$\underbrace{M^{-1}AM^{-T}}_{\tilde{A}} \cdot \underbrace{M^T x}_{\tilde{x}} = \underbrace{M^{-1}b}_{\tilde{b}}$$

$$\tilde{A}\tilde{x} = \tilde{b}$$

$\Rightarrow \tilde{x} \Rightarrow x = M^{-T}\tilde{x}$

Note Convergence of CG Method

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n, \quad \kappa = \text{cond}(A)$$

Conjugate Gradients Algorithm

(For real, symmetric, pos def A)

$r_0 = b - Ax_0$

$s_0 = r_0$

FOR $k = 0, 1, \ldots$

$$\alpha_k = \frac{r_k^T r_k}{s_k^T A s_k}$$

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

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

BREAK LOOP if r_{k+1} is sufficiently small

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$s_{k+1} = r_{k+1} + \beta_k s_k$$

END

return x_{k+1}

A symmetric

$$A = \frac{1}{2}(L + L^T) \qquad A_{ij} = A_{ji}$$

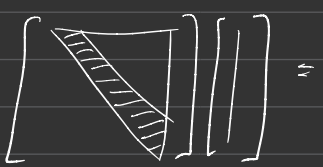
matrix vector calculation

$x \longmapsto Ax$

$T_A(x)$

Band structured A

$N \times N$



$T_A(w)$:

```
for i = rows
    v_i = 0
    for j = columns
        v_i = v_i + A_ij w_j
    end
end
```

$O(n^2)$

```
for i = rows
    v_i = 0
    for j = columns
        if A_ij ≠ 0 then
            v_i = v_i + A_ij w_j
        end
    end
end
```

$O(\#nz)$ — number of non zero entries

Lets be specific

$$A = \begin{bmatrix} 7 & 4 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 3 & 0 & 8 \\ 1 & 6 & 2 & 0 \end{bmatrix}$$

std 7400 0050 0308 1620

16 double precision

new storage scheme

data	7	4	5	3	8	1	6	2
row	1	1	2	3	3	4	4	4
column	1	2	3	2	4	1	2	3

can you convert a bad domain for a PDE problem into a nice domain & solve a nice domain/mesh & solve with

O(#nz) ops and O(#nz) storage

```
for i = rows
    v_i = 0
end
for k = 1 ... data size
    v[row[k]] = v[row[v]] + data[k] · w[col[k]]
end
```

PDE Poisson problem

N+1 => N-1 interior points for which u(x) unknown

$$-\Delta u = f \quad \text{on } \Omega$$

$$u = 0 \quad \text{on } \partial\Omega$$

Laplacian: $\Delta = \sum_{d=1}^{3} \frac{\partial^2}{\partial x_d^2}$, f: source

if $f = 0$ => Poisson problem => Laplace eq

in $d=1$ $\Delta = \frac{\partial^2}{\partial x^2}$, $\Delta u(x) = \frac{\partial^2}{\partial x^2} u(x)$

Taylor tells us: $\boxed{g(x+h) = g(x) + g'(x)h + O(h^2)}$

$\Omega = [0,1]$

$\partial\Omega = \{0,1\}$ => $u(0) = u(1) = 0$

$$\frac{\partial}{\partial x} g(x) = g'(x) + \frac{1}{h}\left(g(x+h) - g(x)\right) + O(h^2)$$

$$\frac{\partial}{\partial x} g'(x) = \frac{1}{h^2}\left[g(x+h) - 2g(x) + g(x-h)\right] + O(h^2)$$

$$\frac{-1}{h^2}\begin{bmatrix} \ddots & 0 & 0 \\ 0 & \begin{matrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{matrix} & 0 \\ 0 & 0 & \ddots \end{bmatrix}\begin{bmatrix} \vdots \\ u(x-h) \\ u(x) \\ u(x+h) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ f(x-h) \\ f(x) \\ f(x+h) \\ \vdots \end{bmatrix}$$