1

# Numerical Methods II

## Lecture 6 - Linear Iterative Methods

$Ax = b$ linear system $\dim = n \sim 10^8$

cannot store $A$: $n^2 >>$ computer

cannot factor $n^3 >>$ computer budget.

Soln: iterate $x_k \to x$ as $k \to \infty$

$k^{th}$ iterate

e.g. Elliptic PDE $\Delta u = f$ with boundary conditions

$$u_{ijk+1} + u_{ijk-1} + \cdots + u_{i-1jk} - 6u_{ijk} = \Delta x^2 f_{ijk}$$

(3-D discrete Laplace op.)

solve $\partial_t u = \Delta u - f$ until $u$ stops changing.

~~$u_{k+1} = (\text{discrete } \Delta) u_k - f$~~

$$u_{k+1} = u_k + \Delta t \underbrace{\left[\overbrace{(\text{discrete } \Delta)}^{-A} u_k - f\right]}_{\text{residual } r_k}$$

take $A = -$ discrete $\Delta$ > symmetric positive definite.

②

$b = -\Delta x^2 f$, $x = u$

(*) $x_{k+1} = x_k - s(A x_k - b)$

Find $s$ in (*) so that $x_k \to x$ as fast as possible.

Remark: may suppose $b=0$.

Set $y_k = x_k - x$, $Ax = b$

$$x_{k+1} - x = x_k - x - s\left(A(x_k - x) + b - b\right)$$

$$y_{k+1} = y_k - sAy_k$$

find $s$ so that $y_k \to 0$ as fast as possible. Now call $y_k$ $x_k$ & just set $b=0$ in (*).

If $A$ symm. & pos. def. diagonalise & apply component by component

$$w_{k+1} = w_k - s\lambda_j w_k$$

③

$\lambda_j$ = eigenvalue of $A$ $\quad A v_j = \lambda_j v_j$

$w_k = w_{k,j} \qquad x_k = \sum w_{k,j} v_j$

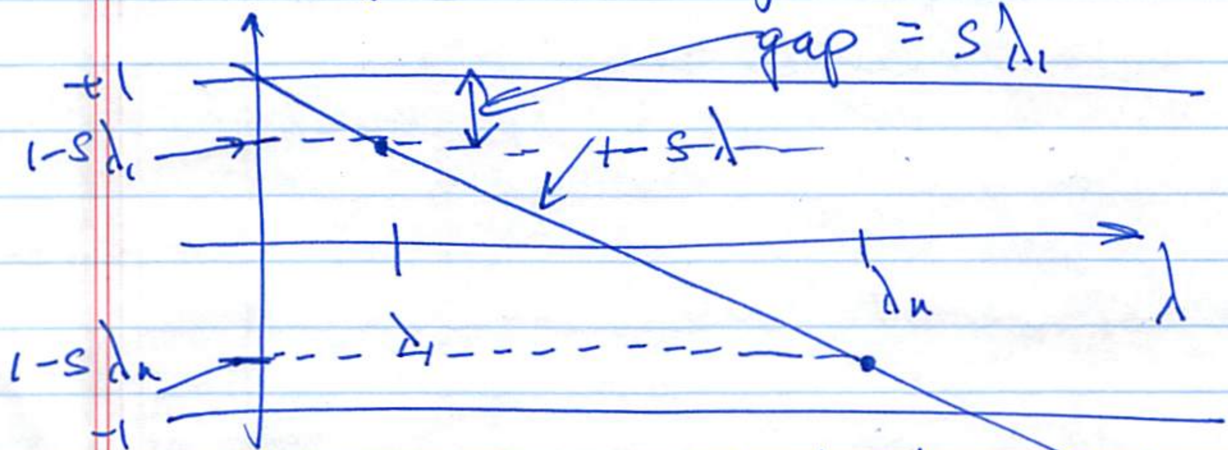
$0 < \lambda_1 \le \cdots \le \lambda_n$ spectral gap.

$w_{k+1} = \mu_j w_k \qquad \mu_j = 1 - s\lambda_j$

Maximize the convergence rate by choosing $s$ to minimize

$$\max_j |\mu_j| = \max_j |1 - s\lambda_j|.$$

gap $= s\lambda_1$




optimal $s$ : $|1-s\lambda_1| = |1-s\lambda_n|$

$$1 - s\lambda_1 = -(1 - s\lambda_n)$$

$$2 = s(\lambda_n - \lambda_1)$$

$$⊛ \quad s = \frac{2}{\lambda_n - \lambda_1}$$

4

$$gap = s\lambda_1$$

$$= \frac{2\lambda_1}{\lambda_n - \lambda_1}$$

$$= \frac{2}{\frac{\lambda_n}{\lambda_1} - 1}$$

$$\boxed{gap = \frac{2}{cond(A) - 1}}$$

condition # $= \frac{\lambda_n}{\lambda_1}$

Conclusions:

1) if cond(A) is large (e.g. $\frac{1}{\Delta x^2}$) then even the optimal method converges slowly

2) To find the optimal s you need to know $\lambda_1$ and $\lambda_n$, not likely in practice. Adaptive method - try to find a good s as the algorithm goes.

## Variational formulation

$$Ax = b \iff \min \underbrace{\tfrac{1}{2} x^t A x - x^t b}_{F(x)}$$

$$\nabla F = Ax - b.$$

Gradient descent: $x_{k+1} = x_k - s \nabla F(x_k)$

$s$ = "~~the~~ learning rate".

$s$ too small ⇒ slow convergence

$s$ too large ⇒ instability.

## PDE methods: (finite difference only)

$$u_{k+1} = u_k + \frac{\Delta t}{\Delta x^2} \text{ discrete } \Delta u_k$$

$u_{k+1} = \bar{u}$ $\qquad$ $u_k = u$

$$\bar{u}_{ij} = u_{ij} - \frac{\Delta t}{\Delta x^2}\left(\bar{u}_{i+1,j} + \cdots + \bar{u}_{i,j-1} - 4\bar{u}_{ij}\right)$$

$$= \left(1 - 4\frac{\Delta t}{\Delta x^2}\right) u_{ij}$$

$$\frac{\Delta t}{\Delta x^2}\left(\bar{u}_{i+1,j} + \cdots + \bar{u}_{i,j-1}\right).$$

⑥

Choose $\Delta t$ so that

$$1 - 4\frac{\Delta t}{\Delta x^2} = 0.$$

(CFL max time step)

Larger $\Delta t$ makes a coefficient negative.

Get

$$\bar{u}_{ij} = \frac{1}{4}\left(u_{ij+1} + u_{ij-1} + u_{i+1j} + u_{i-1,j}\right)$$

= average of neighbors.

Jacobi iteration

(J) $$u_{k+1,ij} = \frac{1}{4}\left(u_{kij+1} + \cdots + u_{ki-1,j}\right)$$

Solve eqn $(i,j)$ for variable $(i,j)$.

---

Abstract version

$$\min_{x_j} F(x)$$

May or may not be stable for other problems.

Gauss - Seidel iteration

for $j = 1, \cdots, n$ ⟸ one "sweep" = 1 iteration.

replace $x_j$ with $\arg\min_{x_j} F(x)$

For each $j$, $F$ decreases ⟹ automatically stable.

⑦

For discrete Laplace in 2D:

$$\left.\begin{array}{l} \text{for } i=1,\dots m \\ \quad \text{for } j=1,\dots m \end{array}\right\} n=m^2$$

(GS) $$\bar{u}_{ij} = \frac{1}{4}\left(\bar{u}_{i-1,j} + \bar{u}_{i,j-1} + u_{i+1,j} + u_{i,j+1}\right).$$

Advantages over Jacobi

- variational $\Rightarrow$ stable + convergent for any problem with A symm + pos def.
- ~~on~~ one "vector" instead of 2

rumor: GS was discovered by accident when someone programmed Jacobi with only one vector.

<u>Disadvantage</u>: relies on the fact in this

example that it is easy to calculate $\partial_{x_j} F(x)$ w/o evaluating all of $F$.

This is often true, e.g. finite difference discretization.

It is often false, e.g. spectral discretization or integral equation.

<u>Over-relaxation</u>

for $j = 1, \ldots, n$

find $x_j^* = \arg\min_{x_j} F(x)$

$x_j \leftarrow x_j + \omega \left( x_j^* - x_j \right)$

$\omega = 1$ = Gauss Seidel

$\omega < 1$ = under-relaxation

$\omega > 1$ = over-relaxation = extrapolation.

(9)

Motivation: The iterates seem to be moving at constant speed:

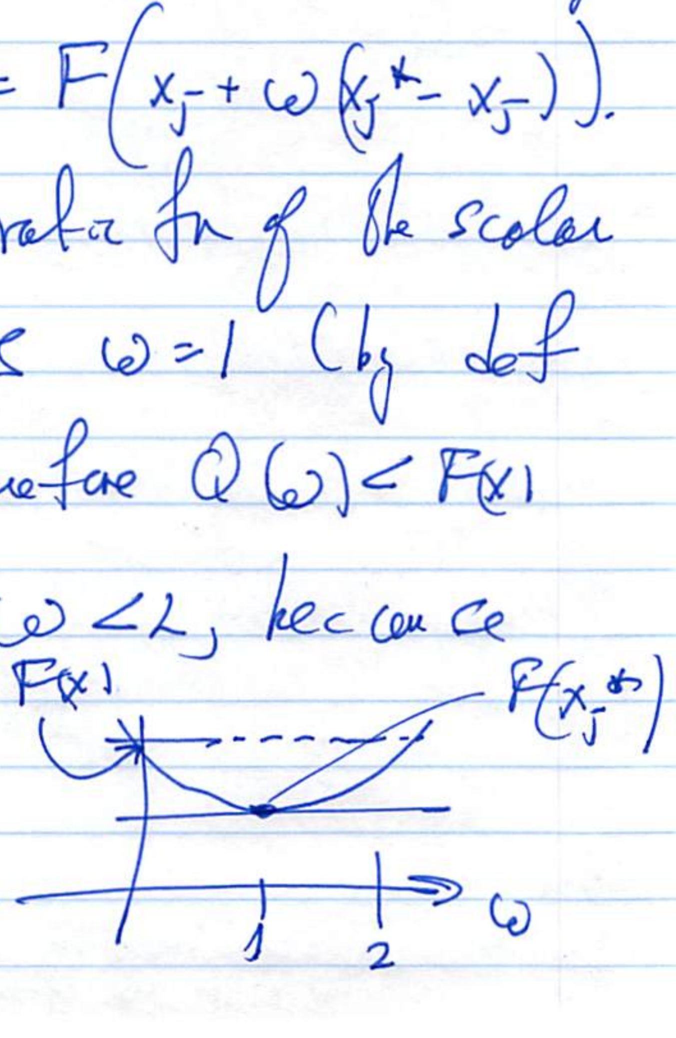
$$x_{k+1} - x_k \cong x_k - x_{k-1}$$

You should get there faster by extrapolation.

Theory: For linear problems, quadratic F, $0 < \omega < 2$ is stable.

$\omega \leq 0$ or $\omega \geq 2$ is non-convergent.

Pf: define $Q(\omega) = F\left(x_j + \omega(x_j^* - x_j)\right)$.
This is a quadratic fn of the scalar $\omega$. The min is $\omega = 1$ (by def of $x_j^*$). Therefore $Q(\omega) < F(x)$ only if $0 < \omega < 2$, because it is a parabola



## Definitions

Preconditioner: an operation (linear) to help solve $Ax=b$ that involves applying a matrix $M \neq A$. e.g. Gauss Seidel. Often very problem specific e.g. multigrid. $MA$ or $M^{-1}A$ = preconditioned matrix.

Iterative method: Find $x$ using only matrix application ($y \to Ay$) and linear combinations $y_1, y_2 \to ay_1 + by_2$

$y_1^t y_2$ (inner product).

$n$ stages of an iterative method produces a polynomial in $A$.

Problem: Find polynomials $P_n(A)$ with $P_n(0) = I$ so that $P_n(A)x$ is

11

small when $n$ is large.

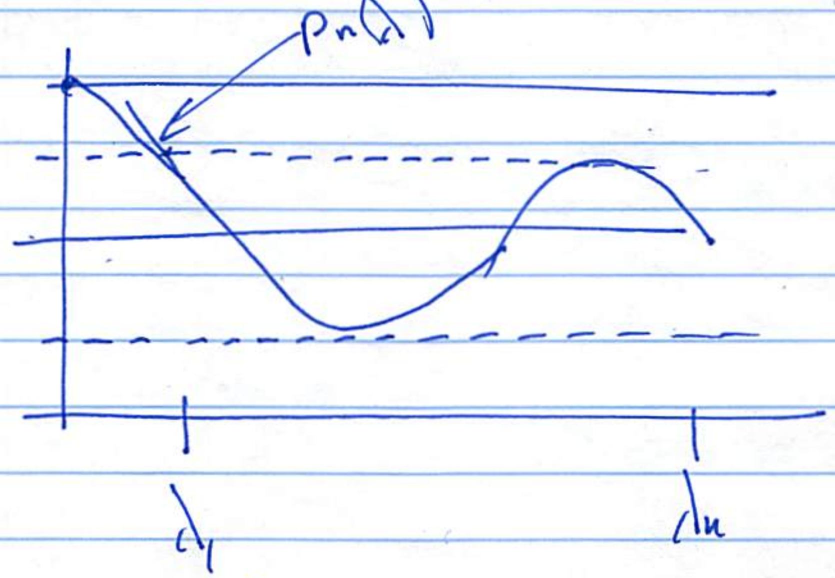
Diagonalize

$w \to P_n(\lambda) w$

Optimal method

$$\min_{\lambda \in \mathrm{spec}(A)} \to \max p_n(\lambda)$$

$p_n$ deg $n$
$p_n(0) = 1$

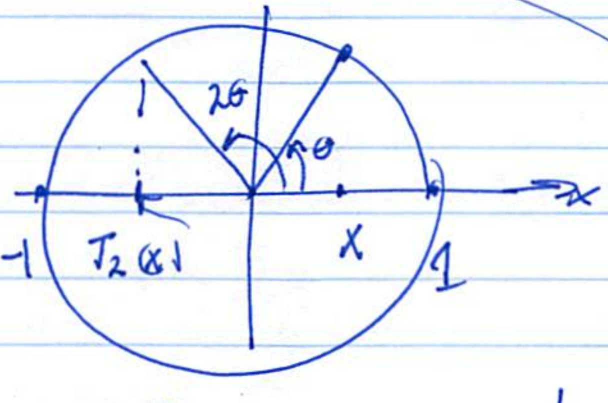



Answer: Chebychev polynomial.

Digression on ~~Chebychev~~ Chebyshev polynomial.

$T_n(x)$ = poly degree $n$ in $x$.

$T_n(x) = \cos(n\theta)$ , $x = \cos(\theta)$




<u>Proposition:</u> if $-1 \le x \le 1$ then $-1 \le T_n(x) \le 1$.

Recurrence relation

$$T_{n+1}(x) = \cos(n\theta + \theta)$$

$$= \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta)$$

$$T_{n-1}(x) = \cos(n\theta - \theta)$$

$$= \cos(n\theta)\cos(-\theta) - \sin(n\theta)\sin(-\theta)$$

$$\cos(\theta) \quad + \cancel{\cos}\sin(n\theta)\sin(\theta)$$

$$T_{n+1}(x) + T_{n-1}(x) = 2x\,T_n(x)$$

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x)$$ – recurrence relation

General version: $p_{n+1}(x) = x\,p_n(x) - a_n p_n(x) - b_n p_{n-1}(x)$

13

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x \cdot x - 1 = 2x^2 - 1$$

$$T_3(x) = 2x(2x^2 - 1) - x$$

$$= 4x^3 - 3x$$

$$T_4(x) = 2x(4x^3 - 3x) - 2x^2 + 1$$

$$= 8x^4 - 8x^2 + 1$$

$$T_5(x) = 2x(8x^4 - 8x^2 + 1) - 4x^3 + 3x$$

$$= 16x^5 - 20x^3 + 5x$$

$$T_6(x) = 2x(16x^5 - 20x^3 + 5x) - 8x^4 + 8x^2 - 1$$

$$= 32x^6 - 48x^4 + 18x^2 - 1$$

Notice: · $T_n(x)$ poly degree $n$

· $2^{n-1}x^n$ + lower order (deg $n-2$)

· even $n \Rightarrow$ all even powers

odd $n \Rightarrow$ all odd powers

(14)

- $T_n(1) = 1$, $T_n(-1) = -1$, $T_n(m_j) = \pm 1$

  $j = 0, \cdots, n$

Theorem: If ① $p_n(x) = 2^{n-1}x^n +$ lower order

② $|p_n(x)| \le 1$ for $-1 \le x \le 1$

Then $p_n(x) = \pm T_n(x)$.

Pf: ( ... )

Consequence of $T_n$ (not theorem):

The monomial basis is ill conditioned.

A basis is well conditioned if the new element is not close to a linear combination of old elements.

e.g. ortho-normal basis.

But $|x^n - 2^{-(n-1)}(\text{l.o.})| \le 2^{-(n-1)}$

on $[-1, 1]$

so $x^n$ is very close to a linear combination of $1, x, \cdots, x^{n-1}$.

<u>Chebyshev acceleration</u>: Find a polynomial $p_n(\lambda)$ with

$$p_n(0) = 1$$

$$\max\{|p_n(\lambda)| \quad \lambda_1 \le \lambda \le \lambda_n\} = \min$$

Construct $p_n(\lambda)$ from $T_n(x)$.

<u>Idea</u>: $T_n(x)$ is the "smallest" polynomial

$2^{n-1}x^n +$ lower order on $[-1, 1]$.

$T_n(x)$ is the fastest growing degree $n$ polynomial outside the interval $[-1, 1]$.

Take $p_n(\lambda) = c_n T_n(x(\lambda))$

where $\quad x(\lambda) = a\lambda + b$

$$\lambda_1 \to 1, \quad \lambda_n \to -1$$

$$c_n = \frac{1}{T_n(x(0))}$$ so that $p_n(0) = 1$.

Since $[\lambda_1, \lambda_n] \to [1, -1]$ and $0$ is outside $[\lambda_1, \lambda_n]$, we know $x(0)$ is outside of $[1, -1]$. Actually, $0 \to x(0) > 1$. Therefore $T_n(x(0))$ is large - as large as possible among polynomials

$$|S_n(x)| \le 1 \quad -1 \le x \le 1.$$

This makes $c_n$ as small as possible

<u>Proposition</u> if $|S_n(x)| \le 1$ for $-1 \le x \le 1$ and $S_n$ degree $n$, then $T_n(x) \ge S_n(x)$ for all $x > 1$. If $T_n(x) = S_n(x)$ for any $x > 1$, then $T_n = S_n$ exactly.

<u>Proof</u> $T_n(x)$ on $[-1, 1]$ starts at $\pm 1$ when $x = -1$ and goes back and forth to $\pm 1$.

(17)

This means $T_n(y_k) = S_n(y_k)$ for lots of $y_k \in [-1, 1]$. $T_n(1) = 1$ and $S_n(1) \le 1$. If $S_n(x) > T_n(x)$ for $x > 1$, then $T_n = S_n$ too many times. (You do the counting).
QED

The map

$$x(\lambda_1) = 1 \qquad x(\lambda_n) = -1 \quad \text{(orientation reversing)}$$

$$1 = a\lambda_1 + b$$
$$-1 = a\lambda_n + b$$

$$b = x(0) = \frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}$$

$$\kappa = \operatorname{cond}(A) = \frac{\lambda_n}{\lambda_1}$$

$\lambda_n = \kappa \cdot \lambda_1$ $\qquad \kappa$ = large number.

$$b = \frac{\kappa\lambda_1 + \lambda_1}{\kappa\lambda_1 - \lambda_1} = \frac{\kappa+1}{\kappa-1} = \frac{\kappa-1}{\kappa-1} + \frac{2}{\kappa-1}$$

$$b = 1 + \frac{2}{\kappa - 1}$$

(18)

$b = 1 + \varepsilon$

$T_n(1+\varepsilon) = \cos(n\theta)$

$\cos(\theta) = 1 + \varepsilon \qquad \varepsilon > 0$

$\theta$ near $0$ if $\varepsilon$ small.

$\cos(\theta) = 1 - \frac{1}{2}\theta^2$

$1 - \frac{1}{2}\theta^2 = 1 + \varepsilon$

$\theta^2 = -2\varepsilon$

$\theta = i\sqrt{2\varepsilon}$

$$\cos(n\theta) = \frac{e^{in\theta} + e^{-in\theta}}{2}$$

$$= \frac{\exp(-n\sqrt{2\varepsilon}) + \exp(n\cdot\sqrt{2\varepsilon})}{2}$$

for large n, have

$$C_n \cancel{\varepsilon} = \frac{1}{T_n(1+\varepsilon)} = \frac{2}{e^{n\sqrt{2\varepsilon}}}$$

$$\varepsilon = \frac{2}{\kappa - 1} \qquad \sqrt{2\varepsilon} \cong \sqrt{\frac{4}{\kappa}} = \frac{2}{\sqrt{\kappa}}$$

19

$$p_n(\lambda) = \frac{1}{C_n} T_n(x(\lambda))$$

has

$$|p_n(\lambda)| \le \frac{1}{C_n} \cdot |T_n(x(\lambda))| \le \frac{1}{C_n}$$

of $|T_n| \le 1$ if $-1 \le x \le 1$.

$$|p_n(\lambda)| \lesssim e^{-\frac{2n}{\sqrt{\kappa}}}$$

convergence rate with $\frac{1}{\sqrt{\kappa}}$

instead of $\frac{1}{\kappa}$.

Chebyshev acceleration algorithm

1) Decide $n$ = # of iterations

2) get $\lambda_1, \lambda_n$

3) find $x_{1n}, \dots, x_{nn}$ = zeros of $T_n(x)$

$$T_n(x_{kn}) = 0$$

4) $x(\lambda_{kn}) = x_{kn}$ $\qquad \lambda_{kn} = \dfrac{x_{kn} - b}{a}$

(20)

$$s_{kn} = \frac{1}{\lambda_{kn}}$$

$$x_{k+1} = x_k - s_{kn} (\underbrace{A x_k - b}_{r_k = \text{residual}})$$

$\boxed{\text{not } b \text{ of } x = a\lambda + b}$

---

Conjugate gradients

1) do not need to know $\lambda_1$, $\lambda_n$, # of iterations in advance

2) optimal polynomial for the A you have, not just uniform spectrum in $[\lambda_1, \lambda_n]$

3) Based on orthogonality - 3 term recurrence relation.

(21)

Reference for Conjugate gradients

Numerical Optimization Jorge Nocedal, Steve Wright

Notation for Conjugate gradients.

$d$ = dim. of space

$n$ = iterate $\quad x_n = n^{th}$ iterate

$k$ = earlier iterate

$x_* =$ answer $\quad Ax_* = b$

$$x_* = \arg\min \overbrace{\tfrac{1}{2} x^t A x - x^t b}^{F(x)}$$

A-norm $\quad \|x\|_A = \sqrt{x^t A x}$

Search directions $p_0, p_1, \cdots$

Krylov space $\quad K_n = \operatorname{span}\{p_0, \cdots, p_n\}$

conjugate = orthogonal in the A-norm

$$p_j^t A p_k = 0 \text{ if } j \neq k$$

Claim $\quad x_n = \arg\min_{x \in K_n + x_0} \|x - x_*\|_A.$

residual $= \nabla F = Ax - b = r$

$\alpha_n$ = step size at iteration $n$

(22)

"line search"
= minimize in a search direction

$$x_{n+1} = x_n + \alpha_n p_n$$

$$\alpha_n = \arg\min F(x_n + \alpha p_n)$$

$$= \arg\min \cancel{\|x_n + \alpha p_n\|_A} \; \|x_n + \alpha p_n - x_*\|_A$$

Pf: $F(x_n + \alpha p_n)$

$$= \frac{1}{2}(x_n + \alpha p_n)^t A (x_n + \alpha p_n) - (x_n + \alpha p_n)^t b$$

$$= \cancel{\frac{1}{2} x_n^t A x_n} + \alpha p_n^t A x_n + \frac{1}{2}\alpha^2 p_n^t A p_n$$

indep. of $\alpha$

$$- \cancel{x_n^t b} - \alpha p_n^t b$$

minimize over $\alpha$, set $\partial_\alpha \cdots = 0$

$$p_n^t A x_n - p_n^t b + \alpha_n p_n^t A p_n = 0$$

$$\cancel{\alpha_n = \alpha} = p_n^t r_n + \alpha_n p_n^t A p_n$$

$$\boxed{\alpha_n = -\frac{p_n^t r_n}{p_n^t A p_n}}$$ "optimal" step

$$\cancel{\frac{1}{2}\|x_n + \alpha p_n\|_A^2 = \frac{1}{2}(x_n + \alpha p_n)^t A}$$

$$A x_* = b$$

23

$$\frac{1}{2}(x_n + \alpha p_n - x_*)^t A (x_n + \alpha p_n - x_*)$$

$$= \frac{1}{2}(x_n + \alpha p_n)^t A (x_n + \alpha p_n)$$

$$- (x_n + \alpha p_n)^t \underbrace{A x_*}_{b}$$

$$+ \frac{1}{2} x_*^t A x_*$$

$= F(x_n + \alpha p_n) +$ something independent of anything

<u>Lemma</u> (Pythagoras): if $x_0$

$$x_n = \arg\min_{x \in x_0 + K_n} \|x - x_*\|_A$$

and $p_{n+1} \perp K_n$ ($p_{n+1}^t A y = 0$, $f\ y \in K_n$)

and $K_{n+1} = \text{span}\{K_n, p_{n+1}\}$

and $x_{n+1} = \arg\min_{x \in K_{n+1} + x_0} \|x - x_*\|_A$

then $x_{k+1} = x_k + \alpha_k p_{k+1}$

<u>Proof</u>: Obvious (Pythagoras)

~~Consequence: $r_k$ =~~

<u>Summary</u>: suppose $p_j^t A p_k = 0$ $j \neq k$

$p_j \neq 0$

$$x_{n+1} = x_n + \alpha_n p_n$$

$$\alpha_n = -\frac{p_n^t A r_n}{p_n^t A p_n}$$

$$r_n = A x_n - b$$

Then ① $x_n = \underset{x \in x_0 + K_n}{\arg\min} F(x)$

② $r_n$ perp to $K_n$ $r_n^t g = 0$

of $g \in K_n$ $r_n^t p_k = 0$ $k < n$

<u>Pf</u> ① is a consequence of the lemma

② is a consequence of ①.

Conjugate gradients, make $p_{n+1}$ orthogonal to $K_n$ by orthogonalization (DUH!)

Fact: orthogonalize against $p_n$ only

The rest are automatic

$$x_{n+1} = x_n + \alpha_n p_n$$

$$r_{n+1} = A x_n - b$$

$$p_{n+1} = r_{n+1} - \beta_n p_n$$ (orthogonalize against $p_n$ only)

choose $\beta_n$ by

$$p_n^t A p_{n+1} = 0$$

$$p_n^t A r_{n+1} - \beta_n p_n^t A p_n = 0$$

$$\beta_n = \frac{p_n^t A r_{n+1}}{p_n^t A p_n}$$

Proposition: this $p_{n+1}$ also satisfies

$$p_k^t A p_{n+1} = 0 \quad \text{for } 0 \le k \le n-1$$

Proof: $p_k^t A r_{n+1} - \beta_k \cancel{p_k^t A p_n}$

since $p_k$ are A-orthogonal "by induction".

(26)

Lemma: ① $K_n = \text{span}(p_0, A p_0, \ldots, A^n p_0)$, $r_0 = A x_0 - b$

② $r_n \in K_n$

Pf induction:

$p_{n+1} = r_{n+1} - \beta_n p_n$

so if $p_{n+1} \in K_{n+1}$ then $r_{n+1} \in K_{n+1}$

and conversely.

Now: $r_{n+1} = A x_{n+1} - b$

$= A(x_n + \alpha_n p_n) - b$

$= A x_n - b - \alpha_n A p_n$

(*) $\boxed{r_{n+1} = r_n - \alpha_n A p_n}$

This shows $r_{n+1} \in K_{n+1}$.

It also shows you can update $r_{n+1}$ w/o doing another matrix/vector product

Proof of Proposition:

27

$$p_k^t A r_{n+1} = (A p_k)^t r_{n+1}$$

But $A p_k \in K_{k+1}$ and $k+1 < n+1$

so $A p_k \perp r_{n+1}$.