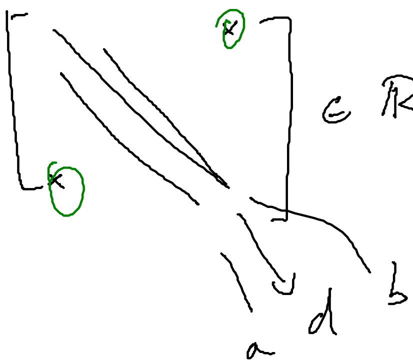
Iterative Methods for Linear Systems of Equations

$\underline{\underline{A}}\,\underline{x} = \underline{b}$, $\underline{\underline{A}}$ sparse: most of its entries are zeros

$A =$  $\in \mathbb{R}^{n \times n}$, $n = 10^6$

not stored!

but use alg to compute

$$\underline{w} := \underline{\underline{A}}\,\underline{v}$$

for $j = 1, 2, \dots, n$:

$$w_j = d_j \cdot v_j + a_j \cdot v_{j-1} + b_j\, v_{j+1}$$

if $j = 1$ or n : $w_j = w_j + x\, v_j$

$$\begin{bmatrix} 4 & 1 & & & 2 \\ 1 & 4 & 1 & & \\ & 1 & 4\ 1 & 0 & \\ & 0 & \ddots & \ddots & 1 \\ 2 & & & 1 & 4 \end{bmatrix}$$

Typically, $A = LU$ does not have L, U sparse!

Anyway, we need only an approximation of the exact solution $x \Longrightarrow$ we can use iterative methods.

(x_k) with $x_k \to x$

Idea: $r_k = b - Ax_k$ residual at step k

$$\left.\begin{array}{l} Ax_k = b - r_k \\ Ax = b \end{array}\right\} \Rightarrow A\underbrace{(x - x_k)}_{\text{error at step } k} = r_k$$

$A^{-1} \Rightarrow$

not possible!

$$\underbrace{x - x_k}_{\text{error or exact correction at step } k} = A^{-1} r_k$$

Idea: multiply with P^{-1} instead of A^{-1}

approximation of A^{-1}: $P^{-1} \simeq A^{-1}$

P^{-1} = easy to compute (cheap)

example $P = \mathrm{diag}(A) \Rightarrow$ Jacobi iteration

In general: $Px = Px \underbrace{- Ax + b}$ $\Leftrightarrow$ $Px = (P - A)x + b$

Iteration: $Px_{k+1} := (P - A)x_k + b$

$$\left.\begin{array}{l} A(x - x_k) = r_k \\ P^{-1} \; | \end{array}\right\} \Rightarrow P^{-1}A(x - x_k) = P^{-1} r_k$$

$$x_{k+1} := x_k + P^{-1} r_k$$

Ex 1) $P = \mathrm{diag}(A)$

2) $P := LU$ with $L, U \simeq$ factors of A,

e.g. incomplete LU decomposition

MATLAB: luinc ; cholinc

milu(inc), micholinc.

(modified, incomplete LU)

Does the error decreases ?

$$\left.\begin{array}{l}\boxed{P x_{k+1} = (P-A)x_k + b} \\ P x = (P-A)x + b\end{array}\right\} \Rightarrow P \underbrace{(x - x_{k+1})}_{e_{k+1}} = (P-A)\underbrace{(x-x_k)}_{e_k} + 0$$

$$\Rightarrow P e_{k+1} = (P-A) e_k \Rightarrow e_{k+1} = P^{-1}(P-A) e_k = \underbrace{(I - P^{-1}A)}_{M} e_k$$

$$e_{k+1} = M e_k = \dots = M^k e_0$$

$e_k \to 0$ only for $\rho(M) = \max |\lambda(M)| < 1$

imagine $M = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \Rightarrow M^k = \begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{bmatrix}$

<u>Remark</u> faster convergence for smaller $\rho(M)$!

$\Rightarrow$ idea of weighted Jacobi method

$D = \text{diag}(A)$

$M = I - \omega D^{-1} A$ with $\omega := \frac{2}{3}$

(makes $P = \frac{1}{\omega} D$)

$\hookrightarrow$ "preconditioner"

Now, suppose we do not use any preconditioner P

$P = I$ = identity matrix

<u>Note</u>

take $x_0 = 0$ $\leftarrow A^0 b$

$x_1 := b$ $\leftarrow A^0 b, A^1 b$

$x_2 := (I - A) b + b = b - Ab + b = 2b - Ab$

$x_3 := (I - A) x_2 + b = (I - A)(2b - Ab) + b = 3b - 3Ab + A^2 b$

$A^0 b, A^1 b, A^2 b$

$$x_\ell = \text{linear combination of } \underbrace{A^0 b, A^1 b, \dots, A^{\ell-1} b}_{\ell \text{ Vektors in } \mathbb{R}^n}$$

$$\Rightarrow x_\ell \in \mathcal{K}_\ell(A,b) := \operatorname{span}\{b, A^1 b, \dots, A^{\ell-1} b\}$$

Krylov-space generated by $\underline{A}$ and $\underline{b}$

<u>Idea</u> Choose x_ℓ to be a "good" combination of $b, Ab, \dots, A^{\ell-1} b$

1) choose x_ℓ such that $r_\ell \perp \mathcal{K}_\ell \Rightarrow$ CG

2) choose x_ℓ such that $\|r_\ell\| = \min!$ GMRES, MINRES

3) in case A is not symmetric,

$$r_\ell \perp \mathcal{K}_\ell(\underline{A}^T) \Rightarrow \text{BiCG}$$

BiCGstab

4) $\|e_\ell\| = \min! \Rightarrow$ SYMMLQ

<u>Note</u> $\underline{A}$ symmetric, use CG, because short recurrences fast algorithms!

$$Ax = b \qquad \mathcal{K}_\ell(A,b) = \operatorname{span}\{b, Ab, A^2 b, \dots, A^{\ell-1} b\}$$

<u>Example</u> $A = \begin{bmatrix} 1 & & & 0 \\ & 2 & & \\ & & 3 & \\ 0 & & & 4 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

$$K_4 := \begin{bmatrix} b & Ab & A^2 b & A^3 b \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$$

$$\operatorname{cond}(K_4^T K_4) \simeq 10^6, \quad \operatorname{cond} K_4 \simeq 10^3 \gg 1$$

Hence $b, Ab, \ldots$ are not well suited for computations

$\Rightarrow$ orthogonalise in order to work with orthogonal basis!

"CG" $\Rightarrow$ construct an orthogonal basis of $\mathcal{K}_\ell(A, r_0)$ and call it $\{r_0, r_1, \ldots, r_{\ell-1}\}$

construct an A-orthogonal basis of $\mathcal{K}_\ell(A, r_0)$ and call it $\{p_1, p_2, \ldots, p_\ell\}$

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

How is this done?

$$Ax = b \iff x = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\, J(x) \text{ with } J(x) = \frac{1}{2} x^T A x - b^T x$$

Take $x^{(0)} = 0$.

$$x^{(\ell)} := \underset{x \in \mathcal{K}_\ell(A, r_0)}{\operatorname{argmin}}\, J(x)$$

If we have $p_1, p_2, \ldots, p_\ell$ an A-orthogonal basis

search for $x^{(\ell)} := \gamma_1 \underline{p}_1 + \ldots + \gamma_\ell \underline{p}_\ell$ that minimizes $J(x)$

$$\begin{bmatrix} p_1^T A p_1 & \ldots & p_1^T A p_\ell \\ \vdots & & \vdots \\ p_\ell^T A p_1 & \ldots & p_\ell^T A p_\ell \end{bmatrix} \underline{\gamma} = \begin{bmatrix} p_1^T r \\ \vdots \\ p_\ell^T r \end{bmatrix} \Rightarrow$$

$$\underline{\gamma} = \left[p_1^T r / p_1^T A p_1 \;\; \ldots\ldots \;\; p_\ell^T r / p_\ell^T A p_\ell \right]^T$$

CG:

$$p_1 = r_0 = b - A x^{(0)}$$

for $j = 1, 2, 3, \dots, \ell$:

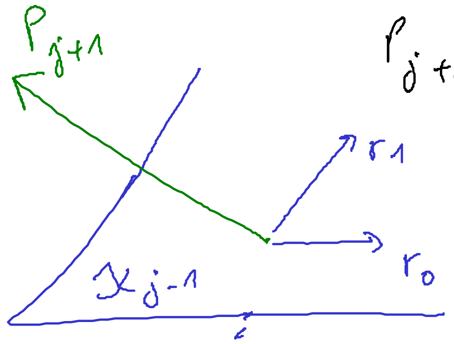
$$x^{(j)} = x^{(j-1)} + \underbrace{\frac{p_j^T r_{j-1}}{p_j^T A p_j}}_{=\gamma_j} p_j$$

$r_j \perp \mathcal{K}_{j-1}$

$$r_j = r_{j-1} - \frac{p_j^T r_{j-1}}{p_j^T A p_j} A p_j$$

p_{j+1} A-orth. to $p_1, \dots, p_j$ $(\mathcal{K}_{j-1})$

$$p_{j+1} = r_j - \frac{(A p_j)^T r_j}{p_j^T A p_j} p_j$$



→ faster than $\frac{\kappa(A) - 1}{\kappa(A) + 1}$

Convergence:

$$\|x - x^{(\ell)}\|_A \le 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^{\ell} \|x - x^{(0)}\|_A$$

$$\|v\|_A^2 := v^T A v$$

Note: Improve the method by using a preconditioner P, such that $\kappa(P^{-1} A)$ is smaller!