

1

## Metoda gradientów sprzężonych

Liniowy układ równań  $Ax = b$

Macierz  $A$  jest symetryczna ( $A^T = A$ ) i dodatnio określona  
tj.  $\forall x \in \mathbb{R}^n, x \neq 0$  zachodzi  $x^T A x > 0$ .

Zagadnienie wyznaczenia rozwiązania  $x_*$  ( $Ax_* = b$ )  
jest równoważne zagadnieniu minimalizacji funkcji

$$Q(x) = \frac{1}{2} x^T A x - b^T x$$

Minimalizacja  $Q(x)$  w wybranym kierunku  $p$ :

$$q(\alpha) := Q(x - \alpha p), \alpha \in \mathbb{R}; x, p \in \mathbb{R}^n$$

$$q(\alpha) = \frac{1}{2} p^T A p \cdot \alpha^2 - p^T (Ax - b) \cdot \alpha + \frac{1}{2} x^T (Ax - 2b)$$

$$q_{\min} = q(\alpha_{\min}), \text{ gdzie } \alpha_{\min} = \frac{p^T (Ax - b)}{p^T A p} = - \frac{p^T r}{p^T A p}$$

$$\text{gdzie } r = b - Ax.$$

Kierunki sprzężone:  $\{p^0, p^1, \dots, p^{n-1}\}, n = \dim A$

$$(p^i)^T A p^j = \delta_{ij} = \begin{cases} 0, & i \neq j \\ \neq 0, & i = j \end{cases}$$

Uogólniona ortogonalność - dla macierzy symetrycznej i  
dodatnio określonej forma dwuliniowa  $(x, y)_A := x^T A y$   
spełnia aksjomaty iloczynu skalarnego.

**Twierdzenie 1:** Niech  $A$  - symetryczna i dodatnio określona,  
a  $\{p^0, p^1, \dots, p^{n-1}\}$  ( $n = \dim A$ ) są  $A$ -sprężone. Wówczas  
sekwencja wektorów

$$x^{m+1} = x^m - \alpha_m p^m \quad (x^0 - \text{dowolne})$$

$$\text{gdzie } \alpha_m = - (p^m)^T r^m / (p^m)^T A p^m, \quad r^m = b - Ax^m$$



2

zbiega do wektora  $x_*$  (rozwiązania  $Ax=b$ ) w co najwyżej  $n$  krokach.

Dowód:

Przemyślmy do pokazania, że wektor residualny  $r^m = b - Ax^m$  jest ortogonalny do wszystkich wektorów  $p^j$ ,  $j=0,1,\dots,n-1$ . Z definicji wektorów sprzężonych wynika jednak, że tworzą one układ liniowo niezależnych wektorów w liście równej normami przemiennej, czyli jej bazę. Wobec tego wektor  $r^m$  musi być wektorem zerowym! Tym samym  $x^m \equiv x_*$ .

Obliczmy:

$$\begin{aligned} (Ax^{m+1} - b)^T p^j &= (Ax^m - \alpha_m Ap^m - b)^T p^j = \\ &= (Ax^m - b)^T p^j - \alpha_m \underbrace{(p^m)^T}_{A^T} Ap^j \end{aligned}$$

Niech  $j < m$ . Wówczas  $(p^m)^T Ap^j = 0$  (z założenia) i mamy:

$$(Ax^{m+1} - b)^T p^j = (Ax^m - b)^T p^j$$

Gdy  $j = m$  mamy natomiast:

$$\begin{aligned} (Ax^{m+1} - b)^T p^m &= (Ax^m - b)^T p^m - \alpha_m \underbrace{(p^m)^T}_{\text{formuła dla } \alpha_m \dots} Ap^m = \\ &= -(r^m)^T p^m + (p^m)^T r^m = 0 \end{aligned}$$

Reasumując:

$$\begin{cases} (r^{m+1})^T p^j = (r^m)^T p^j \text{ dla } j=0,1,\dots,m-1 \\ (r^{m+1})^T p^m = 0 \end{cases}$$

Kładąc  $m = n-1$  mamy natychmiast koniec,  $x_n$ :

$$(r^n)^T p^j = (r^{n-1})^T p^j = \dots = (r^1)^T p^j = 0$$



(3)

dla  $j = 0, 1, \dots, n-1$ . Zatem  $r^n$  jest ortogonalny do  $p^0, p^1, \dots, p^{n-1}$  co dowodzi, że  $r^n \equiv 0$ .

### Algorytm gradientów sprzężonych (CGM)

START: wybierz dowolny  $x^0$ ; oblicz  $r^0 = b - Ax^0$  i  $(r^0)^T r^0$ ;  
połóż  $p^0 = r^0$ .

Dla  $k = 0, 1, 2, \dots$ :

- 1)  $\alpha_k = -(r^k)^T r^k / (p^k)^T A p^k$ ;
- 2)  $x^{k+1} = x^k - \alpha_k p^k$ ;
- 3)  $r^{k+1} = r^k + \alpha_k A p^k$ ;
- 4) Jeżeli  $\|r^{k+1}\|_2^2 \equiv \sqrt{(r^{k+1})^T r^{k+1}} \leq \varepsilon$  to STOP;
- 5)  $\beta_k = (r^{k+1})^T r^{k+1} / (r^k)^T r^k$ ;
- 6)  $p^{k+1} = r^{k+1} + \beta_k p^k$ ;

Koniec.

- $\varepsilon$  - mała liczba dodatnia określająca kryterium stopu
- formuła dla  $\alpha_k$  jest inna niż „standardowa”, ale pokazujemy że jest jej równoważna.
- w praktycznej implementacji iloczyn  $A p^k$  jest obliczany raz na każdą iterację

Dowodnimy poprawność podanego algorytmu, przede wszystkim fakt, że sekwencja  $p^0, p^1, \dots$  to wektory  $A$ -sprężone.

Dowód przeprowadzimy dla algorytmu w nieco zmodyfikowanej postaci, a mianowicie:

- w kroku (1) użyjemy formuły  $\alpha_k = -(r^k)^T p^k / (p^k)^T A p^k$  (1')



(4)

- w kroku (5) zastosujemy formułę

$$(5') \quad \beta_k = - (p^k)^T A r^{k+1} / (p^k)^T A p^k$$

Na koniec pokażemy równoważność odpowiednich formuł.

**Twierdzenie 2:** Wektory  $p^k$ ,  $k=0,1,2,\dots$  generowane w algorytmie CGM spełniają w-mli

$$(p^k)^T A p^j = 0$$

dla  $0 \leq j < k$ ,  $k=1,2,\dots,n-1$ , oraz  $p^k \neq 0$  chyba, że  $x^k \equiv x_*$ . Podobnie,  $(r^k)^T r^j = 0$  dla  $0 \leq j < k$ ,  $k=1,2,\dots,n-1$ .

Dowód:

Obliczamy:

$$\begin{aligned} (p^j)^T r^{j+1} &\stackrel{(3)}{=} (p^j)^T (r^j + \alpha_j A p^j) = (p^j)^T r^j + \alpha_j (p^j)^T A p^j = \\ &\stackrel{(1')}{=} (p^j)^T r^j - \frac{(p^j)^T r^j}{(p^j)^T A p^j} (p^j)^T A p^j = 0, \quad j=0,1,2,\dots \quad (*) \end{aligned}$$

$$(p^j)^T A p^{j+1} \stackrel{(5')}{=} (p^j)^T A (r^{j+1} + \beta_j p^j) = (p^j)^T A r^{j+1} +$$

$$+ \beta_j (p^j)^T A p^j = (p^j)^T A r^{j+1} - \frac{(p^j)^T A r^{j+1}}{(p^j)^T A p^j} (p^j)^T A p^j = 0 \quad (**)$$

Dalej przeprowadimy rozumowanie indukcyjne. Założymy zatem, że sformułowane w tezie twierdzenia warunki ortogonalności są spełnione dla pewnego  $k < n-1$ . Pokażemy (wykorzystując  $(*)$  i  $(**)$ ), że zachodzą one również dla  $k+1$ .



⑤

Dla  $k=1$  mamy na podstawie (\*):

$$(r^0, r^1) = (p^0, r^1) = 0$$

a na podstawie (\*\*)  $(p^0)^T A p^1 = 0$ .

Niech teraz  $j < k$ . Wówczas:

Wtedy teraz  $j \leq k$ . Wówczas:

$$(r_j)^T r^{k+1} \underset{\textcircled{3}}{=} (r_j)^T (r^k + \alpha_k A p^k) = (r_j)^T r^k + \alpha_k \overbrace{(r_j)^T A p^k}^{(p^k)^T A r_j} =$$

$$\stackrel{\textcircled{6}}{=} (r^j)^T r^k + \alpha_k (p^k)^T A (p^j - \beta_{j-1} p^{j-1}) =$$

$$= \underbrace{(r^j)^T r^k}_{\uparrow} + \alpha_k \underbrace{(p^k)^T A p^j}_{\uparrow} - \alpha_k \beta_{j-1} \underbrace{(p^k)^T A p^{j-1}}_{\uparrow}$$

$L \equiv 0$  na mocy założenia indukcyjnego!

Podobnie, dla  $j=k$  mamy:

$$\textcircled{6} \quad (r^k)^T r^{k+1} = (p^k - \beta_{k-1} p^{k-1})^T r^{k+1} = \underbrace{(p^k)^T r^{k+1}}_{L \equiv 0 \text{ to } (*)} -$$

$$\begin{aligned} -\beta_{k-1} (p^{k-1})^T r^{k+1} &= -\beta_{k-1} (p^{k-1})^T (r^k + \alpha_k A p^k) = \\ &\quad \uparrow \\ &\quad \textcircled{3} \end{aligned}$$

$$= -\beta_{k-1} \underbrace{(p^{k-1})^T r^k}_{\hookrightarrow \equiv 0 \text{ by } (*)} - \beta_{k-1} \alpha_k \underbrace{(p^{k-1})^T A p^k}_{\hookrightarrow \equiv 0 \text{ by } (**)} = 0$$

Trym samym udowodniliśmy prawdziwość 2-giej części  
tezy Twierdzenia 2 dla  $k+1$ .



⑥

Dalej: dla  $j < k$  mamy

$$(p^j)^T A p^{k+1} \stackrel{\text{⑥}}{=} (p^j)^T A (r^{k+1} + \beta_k p^k) = (p^j)^T A r^{k+1} +$$

$$+ \beta_k \underbrace{(p^j)^T A p^k}_{\substack{\hookrightarrow \equiv 0 \text{ z} \\ \text{założenia indukcyjnego}}} = (p^j)^T A r^{k+1} \stackrel{\substack{\uparrow \\ A=A^T}}{=} (r^{k+1})^T A p^j =$$

$$\stackrel{\text{③}}{=} \frac{1}{\alpha_j} (r^{k+1})^T (r^{j+1} - r^j) = \frac{1}{\alpha_j} \underbrace{(r^{j+1})^T r^{k+1}}_{L=0} - \frac{1}{\alpha_j} \underbrace{(r^j)^T r^{k+1}}_{0 \Leftarrow} = 0$$

dopiero co udowodniliśmy!

Powyższy rachunek jest poprawny tylko, gdy  $\alpha_j \neq 0$  - do czego jeszcze wrócimy. Analogiczna równość dla  $j=k$  czyli  $(p^k)^T A p^{k+1} = 0$  to po prostu równość (\*\*).

Tym samym udowodniliśmy prawdziwość 1-szej części tezy TWIERDZENIA 2 dla  $k+1$ .

Udowodnimy teraz, że wektory  $p^k$  porostają niezerowe tak długo, jak ściśle rozwiązanie nie jest znalezione. Założymy, że dla pewnego  $m < n$ , wektor  $p^m \equiv 0$ . Wówczas z formuły w punkcie ⑥ algorytmu CGM wynika, że:

$$\begin{aligned} 0 &= (p^m)^T p^m \stackrel{\text{⑥}}{=} (r^m + \beta_{m-1} p^{m-1})^T (r^m + \beta_{m-1} p^{m-1}) = \\ &= (r^m)^T r^m + 2\beta_{m-1} \underbrace{(r^m)^T p^{m-1}}_{L=0 \text{ bo (*)}} + \beta_{m-1}^2 (p^{m-1})^T p^{m-1} \geq (r^m)^T r^m \end{aligned}$$

Stąd  $r^m = b - A x^m = 0 \Rightarrow x^m \equiv x_*$ . Z drugiej strony, tak długo jak  $r^m \neq 0$  mamy  $\|p^m\|_2^2 \geq \|r^m\|_2^2$  więc oczywiście  $p^m \neq 0$ !



(7)

Wróćmy do kwestii warunku  $\alpha_j \neq 0$ . Mamy

$$\begin{aligned} (r^j)^T p^j &= (r^j)^T (r^j + \beta_{j-1} p^{j-1}) = (r^j)^T r^j + \\ &+ \beta_{j-1} \underbrace{(r^j)^T p^{j-1}}_{(p^{j-1})^T r^j = 0 \text{ bo } (*)} = (r^j)^T r^j \quad (***) \\ (p^{j-1})^T r^j &= 0 \text{ bo } (*) \end{aligned}$$

Dlatego równoważną (i foltywnie stosowaną w praktyce) formułą dla  $\alpha_j$  jest

$$\alpha_j = -(r^j)^T r^j / (p^j)^T A p^j$$

Jeśli zatem  $\alpha_j = 0$  to  $r^j \equiv 0$  i  $x^j \equiv x_*$ , tj. proces iteracyjny zatrzymał się de facto w poprzedniej iteracji!

### KONIEC DOWODU

Pokażemy jeszcze, że każdy  $\beta_k$  można obliczyć jako stosunek kwadratów norm kolejnych wektorów residualnych (formuła (5) w oryginalnym sformułowaniu CGM).

Otóż mamy

$$\begin{aligned} (r^{k+1})^T p^{k+1} &= (r^k + \alpha_k A p^k)^T p^{k+1} \stackrel{(**)}{=} (r^k)^T p^{k+1} = \\ &= (r^k)^T (r^{k+1} + \beta_k p^k) = \underbrace{(r^k)^T r^{k+1}}_{L \equiv 0} + \beta_k (r^k)^T p^k \end{aligned}$$

Zatem:

$$\beta_k = \frac{(r^{k+1})^T p^{k+1}}{(r^k)^T p^k} \stackrel{(***)}{=} \frac{(r^{k+1})^T r^{k+1}}{(r^k)^T r^k} \equiv \frac{\|r^{k+1}\|_2^2}{\|r^k\|_2^2}$$



8

## Zbiorność metody gradientów sprzężonych

① Zdefiniujemy dla  $k=0, 1, 2, \dots, n-1$  podprzestrzenie w  $\mathbb{R}^n$ :

$$S_k = \{x^0\} \oplus \text{span}\{p^0, Ap^0, A^2p^0, \dots, A^k p^0\}$$

Elementy  $S_k$  to takie wektory w  $\mathbb{R}^n$ , które są sumami  $x^0$  i dowolnych kombinacji liniowych wektorów  $\{p^0, Ap^0, \dots, A^k p^0\}$ .

Przestrzeń liniową  $\text{span}\{p^0, Ap^0, \dots, A^k p^0\}$  nazywamy przestrzenią Kryłowa (wymiar  $k+1$ ) dla wektora  $p^0$ . Prowadzi się, że  $x^{k+1}$  w CGM minimalizuje w  $S_k$  normę:

$$\|x_* - x\|_A := \sqrt{(x_* - x)^T A (x_* - x)} \quad \text{tj.}$$

$$\|x_* - x^{k+1}\|_A = \min_{x \in S_k} \|x_* - x\|_A \quad (x_* - \text{dokładne rozwiązanie})$$

② Mają miejsce oszacowania

$$\|x_* - x^k\|_A \leq 2 a^k \xrightarrow{\text{potęga } a!} \|x_* - x^0\|_A \quad \text{i}$$

$$\|x_* - x^k\|_2 \leq 2\sqrt{\kappa} a^k \|x_* - x^0\|_A$$

gdzie  $a = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$ ,  $\kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$  - wsp. uwarunkowania macierzy  $A$ .

Wniosek: gdy  $\kappa \gg 1$ , liczba  $a$  jest bliska jedności i zbieżność będzie powolna. Potrzebny jest zabieg zwany PRECONDITIONING!

③ Zachodzi ważne twierdzenie...

**Twierdzenie 3:** Jeżeli macierz  $A$  posiada  $m$  ( $m \leq n$ ) różnych wartości własnych to metoda CGM daje (teoretycznie) ściśle rozwiązanie nie później niż po  $m$  iteracjach.

Dowody: np. w „Introduction to parallel and vector solution of linear systems” by James Ortega, Plenum Press, 1988.



9

## Preconditioning (skalowanie, poprawa uwarunkowania)

Jeżeli macierz oryginalnego układu  $AX=b$  jest źle uwarunkowana ( $\kappa(A) \gg 1$ ) stosujemy preconditioning.

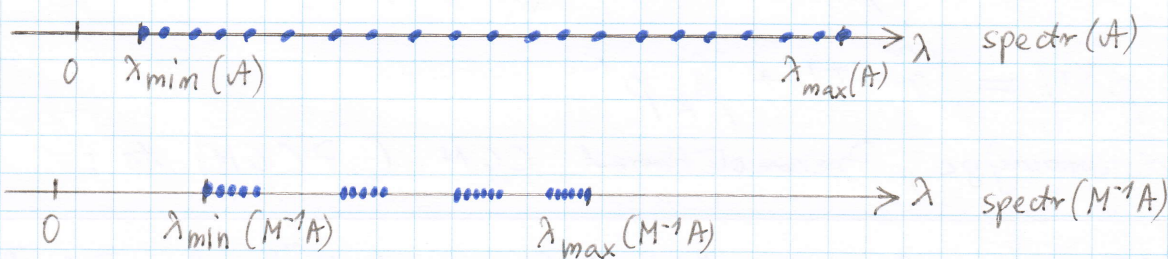
Formalnie, polega to na rozwiązaniu „zastępczego” układu postaci

$$M^{-1}AX = M^{-1}b$$

gdzie macierz  $M$  jest „podobna” do  $A$ , ale rozwiązanie układu z  $M$  jest dużo łatwiejsze niż układu z  $A$ .

„Podobna” do  $A$  oznacza - w idealnej sytuacji - dwie rzeczy:

- $\kappa(M^{-1}A) \ll \kappa(A)$
- widmo macierzy  $M^{-1}A$  ma strukturę „wysepek” tj. wygląda jak „rozburzone” widmo macierzy o niewielkiej liczbie różnych wartości własnych, vide obrazek



Jak to działa w CGM?

Po pierwsze  $M^{-1}A$  musi być nadal SPD. Zwykle  $M$  jest też SPD.

Zdefiniujemy zatem nowy iloczyn skalarny  $(v, w)_M := v^T M w$

Wtedy

$$(M^{-1}A v, v)_M = v^T \underbrace{A^T}_{A} \underbrace{(M^{-1})^T}_{M^{-1}} M v = v^T A v > 0 \text{ dla } v \neq 0.$$



(10)

Okażuje się, że zmodyfikowany CGM można sformułować niemał jak oryginalny modulo dodatkowy krok... Mamy miałośc:

$$\tilde{r}^k := M^{-1}b - M^{-1}Ax^k = M^{-1}(b - Ax^k) = M^{-1}r^k$$

$$\Downarrow$$

$$M\tilde{r}^k = r^k$$

$$\tilde{\alpha}_k = (\tilde{r}^k, \tilde{r}^k)_M / (\underbrace{p^k, M^{-1}Ap^k}_M)_M = (\tilde{r}^k)_M \tilde{r}^k / (p^k)^T A p^k =$$

$$(p^k)^T M M^{-1} A p^k = (p^k)^T A p^k$$

$$= (\tilde{r}^k) r^k / (p^k)^T A p^k$$

$$x^{k+1} = x^k + \tilde{\alpha}_k p^k$$

$$\tilde{r}^{k+1} = \tilde{r}^k - \tilde{\alpha}_k M^{-1} A p^k \quad / \cdot M$$

$$\Downarrow$$

$$r^{k+1} = r^k - \tilde{\alpha}_k A p^k$$

$$\tilde{\beta}_k = \frac{(\tilde{r}^{k+1}, \tilde{r}^{k+1})_M}{(\tilde{r}^k, \tilde{r}^k)_M} = \frac{(\tilde{r}^{k+1})^T r^{k+1}}{(\tilde{r}^k)^T r^k}$$

$$p^{k+1} = \tilde{r}^{k+1} + \tilde{\beta}_k p^k.$$

Podsumowując, Preconditioned CGM (PCGM) to:

START: wybierz  $x^0$ ; oblicz  $r^0 = b - Ax^0$ ; rozwiąż  $M\tilde{r}^0 = r^0$ ; połóż  $p^0 = \tilde{r}^0$ .

DLA  $k=1, 2, \dots$ :

- 1)  $\tilde{\alpha}_k = -(\tilde{r}^k)^T r^k / (p^k)^T A p^k$ ;
- 2)  $x^{k+1} = x^k + \tilde{\alpha}_k p^k$ ;
- 3)  $r^{k+1} = r^k + \tilde{\alpha}_k A p^k$ ;
- 4) sprawdź kryterium zbieżności  $\rightarrow$  jeśli spełnione to STOP
- 5) rozwiąż  $M\tilde{r}^{k+1} = r^{k+1}$ ;
- 6)  $\tilde{\beta}_k = (\tilde{r}^{k+1})^T r^{k+1} / (\tilde{r}^k)^T r^k$ ;
- 7)  $p^{k+1} = \tilde{r}^{k+1} + \tilde{\beta}_k p^k$ ;

KONIEC.



## Metody preconditioningu - niektóre pomysły

- 1) „Rozciąganie” układu  $M\tilde{r} = r$  polega de facto na wykonaniu niewielkiej liczby (kilku) iteracji SSOR (symetryczna nadrelaksacja) zastosowanych do układu  $A\tilde{r} = r$ . Potrzebny jest „tuning” - im więcej iteracji tym szybciej zewrotny proces PCGM zbiega się w mniejszej liczbie iteracji, ale za to każda z nich trwa dłużej.
- 2) Stosujemy do macierzy  $A$  niepełną faktoryzację Choleskiego (fakt. Choleskiego to metoda rozkładu macierzy symetrycznej i dodatnio określonej na ilorzyrn postaci  $A = LL^T$ , gdzie  $L$  jest macierzą dolną trójkątną). „Niepełność” faktoryzacji polega na wyznaczeniu - zamiast dokładnej i - na ogół bardziej niż  $A$  upetnionej macierzy  $L$  - macierzy  $\tilde{L}$  takiej, że:
  - $\tilde{L}\tilde{L}^T \approx A$
  - upetnionej macierzy  $\tilde{L}$  jest takie same lub zbliżone do upetnienia macierzy  $A$

W etapie preconditioningu roziszuujemy układ ( $M = \tilde{L}\tilde{L}^T$ )

$$\tilde{L}\tilde{L}^T\tilde{r} = r$$

czyli układ dwóch układów z macierzami trójkątnymi

$$\begin{aligned}\tilde{L}s &= r \Rightarrow s \\ \tilde{L}^T\tilde{r} &= s \Rightarrow \tilde{r}\end{aligned}$$



3) zastosowaniu aproksymacji  $A^{-1}$  otrzymanej metodą tzw. rozszczepienia regularnego. Polega ono na zapisaniu macierzy  $A$  w postaci:

$$A = P - Q = P(I - P^{-1}Q)$$

przy czym macierz  $P$  jest nonsingularna i taka, że promień spektralny  $\rho(P^{-1}Q) < 1$ .

Wtedy:  $A^{-1} = (I - P^{-1}Q)^{-1}P^{-1} = \underbrace{[I + P^{-1}Q + (P^{-1}Q)^2 + \dots]}_{\text{macierzy "szereg geometryczny"}} P^{-1}$

Jako preconditioner można przyjąć

$$\tilde{r} = \underbrace{[P^{-1} + P^{-1}QP^{-1} + P^{-1}QP^{-1}QP^{-1}]}_{\text{3 pierwsze składniki rozwinięcia}} \cdot r$$

→ można obliczyć rekurencyjnie (jak?! ) i "nie drogo" przy założeniu, że rozkładanie układu z macierzą  $P$  jest tanie (diagonalna, dolna/górna trójkątna itp.)

→ tak ... (ogólniej dla  $n$  składników rozwinięcia ...)

$$s = r$$

for  $k = 1 : m-1$  do

1) solve  $P\tilde{r} = s$ ;

2)  $s = r + Q\tilde{r}$ ;

end;

solve  $P\tilde{r} = s$ ;



(13)

Skalowana metoda gradientów sprzężonych (preconditioned conjugate gradient method - PCGM) - INACZEJ (after Ortega)

Originalny problem  $Ax = b$   $\text{cond}(A) \gg 1$  - wolna zbieżność!

Przekształcamy...

$$CAx = Cb \quad \det C \neq 0$$

Wprowadzamy nowy wektor nieznanych  $\hat{x}$  taki, że  $C^T \hat{x} = x$

Mamy układ:

$$CAC^T \hat{x} = \hat{b}, \quad \hat{b} = Cb$$

Idealny preconditioner  $C$ :  $CAC^T = I \Rightarrow A = C^{-1}C^{-T} = (C^T C)^{-1}$   
czyli  $C^T C = A^{-1}$  (np.  $C = L^T$  - z rozkładu Choleskiego  $A^{-1}$ )

Zdefiniujemy  $M = (C^T C)^{-1}$ . W przypadku idealnego preconditionera  $M = A$ ! W praktyce  $C$  powinno być takie, żeby  $M$  było dobrą aproksymacją macierzy  $A$ .

Zapiszmy „zwykły” CGM dla układu  $\hat{A}\hat{x} = \hat{b}$ ,  $\hat{A} = CAC^T$

START: wybierz  $\hat{x}^0$ , oblicz  $\hat{r}^0 = \hat{b} - \hat{A}\hat{x}^0$ , położ  $\hat{p}^0 = \hat{r}^0$

DLA  $k = 0, 1, 2, \dots$ :

$$\begin{aligned} \hat{\alpha}_k &= -(\hat{r}^k, \hat{r}^k) / (\hat{p}^k, \hat{A}\hat{p}^k) \\ \hat{x}^{k+1} &= \hat{x}^k + \hat{\alpha}_k \hat{p}^k \\ \hat{r}^{k+1} &= \hat{r}^k + \hat{\alpha}_k \hat{A}\hat{p}^k \rightarrow \text{sprawdź zbieżność} \\ \hat{\beta}_k &= (\hat{r}^{k+1}, \hat{r}^{k+1}) / (\hat{r}^k, \hat{r}^k) \\ \hat{p}^{k+1} &= \hat{r}^{k+1} + \hat{\beta}_k \hat{p}^k \end{aligned}$$

Pokażemy, jak przepisać ten algorytm w terminach oryginalnych danych.

Przeanalizujemy pierwszy krok algorytmu ( $k=0$ )



Na starcie mamy:  $\hat{r}^0 = \hat{b} - \hat{A} \hat{x}^0 = Cb - CAC^T C^{-T} x^0 =$   
 $= C(b - Ax^0) = Cr^0$

Mamy dalej  $\hat{p}^0 = \hat{r}^0$ . Zdefiniujmy  $p^0 = C^T \hat{p}^0 \Rightarrow \hat{p}^0 = C^{-T} p^0$

Stąd:

$$(\hat{p}^0, \hat{A} \hat{p}^0) = (C^{-T} p^0)^T C A \underbrace{C^T C^{-T}}_I p^0 = (p^0)^T \underbrace{C^{-1} C}_I A p^0 =$$

$$= (p^0)^T A p^0$$

oraz.  $(\hat{r}^0, \hat{r}^0) = (Cr^0)^T (Cr^0) = (r^0)^T C^T C r^0 =$   
 $= (C^T C r^0)^T r^0$

Oznaczmy  $\tilde{r}^0 = C^T C r^0$

Wówczas  $\hat{\alpha}_0 = -(\tilde{r}^0, r^0) / (p^0, A p^0)$

Mnożymy wyrażenie dla  $\hat{x}^1$  ( $k=0$ ) przez  $C^T$  i odejmujemy

$$x^1 = x^0 - \hat{\alpha}_0 p^0$$

Wyrażenie dla  $\hat{r}^1$  ( $k=0$ ) mnożymy przez  $C^{-1}$  i odejmujemy

$$r^1 = r^0 + \hat{\alpha}_0 C^{-1} \underbrace{C A C^T}_{\hat{A}} \underbrace{\hat{p}^0}_{C^{-T} p^0} = r^0 + \hat{\alpha}_0 A p^0$$

Dalej, zauważmy, że:

$$(\hat{r}^i, \hat{r}^i) = (Cr^i)^T Cr^i = (\tilde{r}^i)^T r^i ; i=0,1$$

zatem:

$$\hat{\beta}_0 = (\tilde{r}^1, r^1) / (\tilde{r}^0, r^0)$$

Wreszcie mnożymy wyrażenie na  $\hat{p}^1$  przez  $C^T$  i odejmujemy

$$p^1 = C^T C r^1 + \hat{\beta}_0 p^0 = \tilde{r}^1 + \hat{\beta}_0 p^0$$



Jak wiadac macierz  $C$  i  $C^T$  pojalnijsz sja wyksztumac w formie iloczynu  $C^T C = M^{-1}$ . Oznacza to, ze deleroc pojalnia sja mnozenie przez  $C^T C$  musimy de facto rozuznac wktad liniowy z macierza  $M$ !

Orletyczna forma PCGM to:

START: wybierz  $x^0$ ; potoz  $r^0 = b - Ax^0$ ; rozuznij  $M\tilde{r}^0 = r^0$ ;  
potoz  $p^0 = \tilde{r}^0$

DLA  $k = 0, 1, 2, \dots$ :

$$\alpha_k = -(\tilde{r}^k, r^k) / (p^k, Ap^k)$$

$$x^{k+1} = x^k - \alpha_k p^k$$

$$r^{k+1} = r^k + \alpha_k Ap^k \rightarrow \text{test zbieznosci}$$

$$M\tilde{r}^{k+1} = r^{k+1} \rightarrow \tilde{r}^{k+1} - \text{preconditioner!}$$

$$\beta_k = (\tilde{r}^{k+1}, r^{k+1}) / (\tilde{r}^k, r^k)$$

$$p^{k+1} = \tilde{r}^{k+1} + \beta_k p^k$$

Co to znaczy, ze macierz  $M$  aproksymuje macierz  $A$ ?

Sluono  $\hat{A} = CAC^T$  to

$$C^T \hat{A} C^{-T} = C^T C \underbrace{A C^T C^{-T}}_I = M^{-1} A.$$

Zatem macierz  $\hat{A} \sim M^{-1} A$  (podobna!), i

$$\text{cond}(\hat{A}) = \frac{\lambda_{\max}(M^{-1}A)}{\lambda_{\min}(M^{-1}A)}$$



## Preconditioning metodą obciętych szeregów Neumanna

TW: Niech  $A$  - macierz nieosobliwa,  $A = P - Q$  to taki rozkład, że macierz  $P$  jest nieosobliwa, a promień spektralny macierzy  $P^{-1}Q$  jest mniejszy od jedności. Wówczas:

$$A^{-1} = \left( \sum_{k=0}^{\infty} \mathcal{U}^k \right) P^{-1}, \quad \mathcal{U} = P^{-1}Q$$

Rozważmy macierz:

$$M = P(I + \mathcal{U} + \mathcal{U}^2 + \dots + \mathcal{U}^{m-1})^{-1}$$

$$M^{-1} = (I + \mathcal{U} + \mathcal{U}^2 + \dots + \mathcal{U}^{m-1}) P^{-1}$$

które są przybliżeniami, odpowiednio, macierzy  $A$  i  $A^{-1}$

Jako preconditioner możemy użyć macierzy  $M$ , czyli mamy

$$\tilde{r} = M^{-1}r = (I + \mathcal{U} + \mathcal{U}^2 + \dots + \mathcal{U}^{m-1}) P^{-1}r \quad (*)$$

Obliczenie  $(*)$  może być obliczone następująco:

$$\begin{cases} r^0 = 0; \\ \text{dla } i = 0, 1, \dots, m-1 \\ \text{rozwiąż } Pr^{i+1} = Qr^i + r; \\ \text{położ } \tilde{r} = r^m; \end{cases}$$

W szczególności gdy  $P = D$  a  $Q = -(L+U)$ , powyższy proces to po prostu  $m$  iteracji metody Jacobiego zastosowanej do układu  $A\tilde{r} = r$ . Stosując tę technikę preconditioningu wykonujemy wewnątrz każdej iteracji PCGM  $m$  iteracji metody Jacobiego przyjmując wyniki ostatniej iteracji jako  $\tilde{r}^{k+1}$ .

W roli preconditionera nie można użyć „zwykłej” metody GS lub SOR. Powód: implikowana tymi metodami macierz  $M$  nie jest symetryczna.

(ani dla  $m=1$ , ani  $m>1$ ). Właściwą metodą jest natomiast SSOR (lub - w szczególności - SGS).



Zadachi umne tverdenie :

TW: Niech  $A = P - Q$  bde macarz symetryczn i dodatnio okreilona, oraz niech  $P$  bde symetryczna i niesobliwa. Wówczas macarz  $R = (I + \mathcal{H} + \mathcal{H}^2 + \dots + \mathcal{H}^{m-1})P^{-1}$ , gdzie  $\mathcal{H} = P^{-1}Q$  spetnia warunki:

- 1) jeżeli liuba  $m$  jst nieparzysta, to  $R$  jst dodatnio okreilona wtedy i tylko wtedy, gdy  $P$  jst dodatnio okreilona
- 2) jeżeli liuba  $m$  jst parzysta to macarz  $R$  jst dodatnio okreilona wtedy i tylko wtedy, gdy dodatnio okreilona jst macarz  $P + Q$ .

Dowód: J. Ortega: Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, NY 1988.

- Wniosek :
- preconditioner Jacobiego bde działat poprawnie dla nieparzystych wartosci  $m$
  - preconditioner SSOR bde działat poprawnie dla dowolnej wartosci  $m$ .

W obu przypadkach  $P$  powinna byc dodatnio okreilona.

TW: Jeżeli macarz  $A$  jst SDO oraz  $\omega \in (0, 2)$  to macarz  $R$  i  $M$  podchodzacy od metody SSOR sđ dodatnio okreilone dla dowolnego  $m$ .

$$\text{Ponieważ } M^{-1}A = (I + \mathcal{H} + \mathcal{H}^2 + \dots + \mathcal{H}^{m-1})P^{-1}(P - Q) = \\ = (I + \mathcal{H} + \mathcal{H}^2 + \dots + \mathcal{H}^{m-1})(I - \mathcal{H}) = I - \mathcal{H}^m$$

$$\text{to, } \text{cond}(\hat{A}) = \text{cond}(M^{-1}A) = \frac{\lambda_{\max}(M^{-1}A)}{\lambda_{\min}(M^{-1}A)} = \frac{1 - \lambda_1^m}{1 - \lambda_m^m}$$

gdzie  $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m$  to wartosci wne macierzy  $\mathcal{H}$  (pamiętamy, że  $\rho(\mathcal{H}) < 1$ , w.c. w tym przypadku (SSOR)  $\rho(\mathcal{H}) \in (0, 1)$ )



Tw: Wiedzi  $A$ -macierz SDO i  $0 < \omega < 2$ . Wtedy

$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m$  - wartości własne macierzy  $\hat{A}$  podlegającej od metody SSOR. Wówczas  $\lambda_1 \geq 0$  i dla  $m$ -krotkowego preconditionera SSOR współczynnik uwarunkowania

$$\text{cond}(\hat{A}) = \frac{1 - \lambda_1^m}{1 - \lambda_m^m}$$

małyje monotoniornie ze wzrostem  $m$ .

Dowód: J. Ortega w monografii jak wyżej.

Uwaga: własność ta nie ma miejsca dla preconditionera Jacobi'ego.