Metody numeryczne

Uzupełnienie: Prewarunkowana metoda gradientów sprzężonych

P. F. Góra

https://zfs.fais.uj.edu.pl/pawel_gora

2025

“Prewarunkowana” (*preconditioned*) metoda gradientów sprzężonych

Załóżmy, że rozwiązujemy równanie

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,, \tag{1}$$

przy czym macierz $\mathbf{A}$ jest symetryczna, dodatnio określona i rzadka. Sugeruje to użycie (algebraicznej) metody gradientów sprzężonych. Jeśli jednak macierz $\mathbf{A}$ jest źle uwarunkowana, zbieżność może być bardzo wolna.

Spróbujmy przyspieszyć zbieżność odpowiednio modyfikując równanie (1) i algorytm gradientów sprzężonych, jednak tak, aby

- nie zmienić rozwiązania,
- macierz zmodyfikowanego układu pozostała symetryczna i dodatnio określona, aby można było zastosować metodę gradientów sprzężonych,

- macierz zmodyfikowanego układu pozostała rzadka, aby jeden krok iteracji był numerycznie tani,
- macierz zmodyfikowanego układu miała niski współczynnik uwarunkowania.

Czy to się w ogóle da zrobić? Okazuje się, że tak!

Postępujemy następująco: Niech $\mathbf{C} \in \mathbb{R}^{N\times N}$ będzie odwracalną macierzą symetryczną, rzeczywistą, dodatnio określoną. Wówczas $\widetilde{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$ też jest symetryczna, rzeczywista, dodatnio określona.

$$\mathbf{C}^{-1}\mathbf{A}\,\underbrace{\mathbf{C}^{-1}\mathbf{C}}_{\mathbb{I}}\,\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}\,, \tag{2a}$$

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}\,, \tag{2b}$$

gdzie $\widetilde{\mathbf{x}} = \mathbf{C}\mathbf{x}$, $\widetilde{\mathbf{b}} = \mathbf{C}^{-1}\mathbf{b}$. Do równania (2b) stosujemy teraz metodę gradientów sprzężonych.

W każdym kroku iteracji musimy obliczyć (tyldy, bo odnosi się to do “tyldowanego” układu (2b))

$$
\begin{aligned}
\alpha_k &= \frac{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}{\widetilde{\mathbf{p}}_k^T \widetilde{\mathbf{A}} \widetilde{\mathbf{p}}_k} = \frac{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}{\widetilde{\mathbf{p}}_k^T \mathbf{C}^{-1} \mathbf{A} \mathbf{C}^{-1} \widetilde{\mathbf{p}}_k}\,, && \text{(3a)} \\
\widetilde{\mathbf{r}}_{k+1} &= \widetilde{\mathbf{r}}_k - \alpha_k \widetilde{\mathbf{A}} \widetilde{\mathbf{p}}_k = \widetilde{\mathbf{r}}_k - \alpha_k \mathbf{C}^{-1} \mathbf{A} \mathbf{C}^{-1} \widetilde{\mathbf{p}}_k\,, && \text{(3b)} \\
\beta_k &= \frac{\widetilde{\mathbf{r}}_{k+1}^T \widetilde{\mathbf{r}}_{k+1}}{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}\,, && \text{(3c)} \\
\widetilde{\mathbf{p}}_{k+1} &= \widetilde{\mathbf{r}}_{k+1} + \beta_k \widetilde{\mathbf{p}}_k\,, && \text{(3d)} \\
\widetilde{\mathbf{x}}_{k+1} &= \widetilde{\mathbf{x}}_k + \alpha_k \widetilde{\mathbf{p}}_k\,. && \text{(3e)}
\end{aligned}
$$

Równania (3) zawierają jawne odniesienia do macierzy $\mathbf{C}^{-1}$, co nie jest zbyt wygodne. Łatwo się przekonać, iż za pomocą prostych przekształceń macierz tę można „usunąć”, tak, iż pozostaje tylko jedno jej nietrywialne wystąpienie. Zdefiniujmy mianowicie

$$\widetilde{\mathbf{r}}_k = \mathbf{C}^{-1}\mathbf{r}_k\,, \quad \widetilde{\mathbf{p}}_k = \mathbf{C}\mathbf{p}_k\,, \quad \widetilde{\mathbf{x}}_k = \mathbf{C}\mathbf{x}_k\,. \tag{4}$$

W tej sytuacji $\widetilde{\mathbf{r}}_k^T\widetilde{\mathbf{r}}_k = (\mathbf{C}^{-1}\mathbf{r}_k)^T\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T(\mathbf{C}^{-1})^T\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T\mathbf{C}^{-1}\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T(\mathbf{C}^{-1})^2\mathbf{r}_k$ etc.

Wówczas równania (3) przechodzą w

$$
\begin{aligned}
\alpha_k &= \frac{\mathbf{r}_k^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}\,, && \text{(5a)}\\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k\,, && \text{(5b)}\\
\beta_k &= \frac{\mathbf{r}_{k+1}^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_{k+1}}{\mathbf{r}_k^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k}\,, && \text{(5c)}\\
\mathbf{p}_{k+1} &= \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k\,, && \text{(5d)}\\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k\,. && \text{(5e)}
\end{aligned}
$$

W powyższych równaniach rola macierzy $\mathbf{C}$ sprowadza się do obliczenia — *jeden raz w każdym kroku iteracji* — wyrażenia $\left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k$, co, jak wiadomo, robi się rozwiązując odpowiedni układ równań. Zdefiniujmy

$$\mathbf{M} = \mathbf{C}^2\,. \tag{6}$$

Macierz $\mathbf{M}$ należy rzecz jasna dobrać tak, aby równanie $\mathbf{M}\mathbf{z} = \mathbf{r}$ można było szybko rozwiązać.

Ostatecznie otrzymujemy następujący algorytm:

$$
\begin{array}{l}
\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1 \\
\text{rozwiąż } \mathbf{M}\mathbf{z}_1 = \mathbf{r}_1 \\
\mathbf{p}_1 = \mathbf{z}_1 \\
\textbf{while } \|\mathbf{r}_k\| > \varepsilon \\
\qquad \begin{aligned}
\alpha_k &= \frac{\mathbf{r}_k^T \mathbf{z}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
&\text{rozwiąż } \mathbf{M}\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \\
\beta_k &= \frac{\mathbf{r}_{k+1}^T \mathbf{z}_{k+1}}{\mathbf{r}_k^T \mathbf{z}_k} \\
\mathbf{p}_{k+1} &= \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k
\end{aligned} \\
\textbf{end}
\end{array}
\tag{7}
$$

Incomplete Cholesky preconditioner

Niech rozkład QR macierzy $\mathbf{C}$ ma postać $\mathbf{C} = \mathbf{Q}\mathbf{H}^T$, gdzie $\mathbf{Q}$ jest macierzą ortogonalną, $\mathbf{H}^T$ jest macierzą trójkątną górną. Zauważmy, że

$$\mathbf{M} = \mathbf{C}^2 = \mathbf{C}^T\mathbf{C} = \left(\mathbf{Q}\mathbf{H}^T\right)^T \mathbf{Q}\mathbf{H}^T = \mathbf{H}\mathbf{Q}^T\mathbf{Q}\mathbf{H}^T = \mathbf{H}\mathbf{H}^T, \tag{8}$$

a więc macierz $\mathbf{H}$ jest czynnikiem Cholesky'ego macierzy $\mathbf{M}$. Niech rozkład Cholesky'ego macierzy $\mathbf{A}$ ma postać $\mathbf{A} = \mathbf{G}\mathbf{G}^T$. *Przypuśćmy, iż $\mathbf{H} \simeq \mathbf{G}$.*

Wówczas

$$\widetilde{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1} = \left(\mathbf{C}^T\right)^{-1}\mathbf{A}\mathbf{C}^{-1} = \left(\left(\mathbf{Q}\mathbf{H}^T\right)^T\right)^{-1}\mathbf{A}\left(\mathbf{Q}\mathbf{H}^T\right)^{-1} = \left(\mathbf{H}\mathbf{Q}^T\right)^{-1}\mathbf{A}\left(\mathbf{H}^T\right)^{-1}\mathbf{Q}^T = \mathbf{Q}\,\underbrace{\mathbf{H}^{-1}\mathbf{G}}_{\simeq\mathbb{I}}\,\underbrace{\mathbf{G}^T\left(\mathbf{H}^T\right)^{-1}}_{\simeq\mathbb{I}}\mathbf{Q}^T \simeq \mathbf{Q}\mathbf{Q}^T = \mathbb{I}\,. \tag{9}$$

Ponieważ $\widetilde{\mathbf{A}} \simeq \mathbb{I}$, współczynnik uwarunkowania tej macierzy powinien być bliski jedności.

W tym momencie mamy spełnione następujące warunki zadane na stronie 2:

- Macierz $\widetilde{\mathbf{A}}$ jest symetryczna i dodatnio określona;
- Rozwiązania się nie zmieniły, gdyż algorytm (7) jest *formalnie* równoważny algorytmowi gradientów sprzężonych;
- Mamy nadzieję (☺), że macierz $\widetilde{\mathbf{A}}$ jest dobrze uwarunkowana, gdyż $\widetilde{\mathbf{A}} \simeq \mathbb{I}$.

Pozostaje tylko zagwarantowanie, że jeżeli $\mathbf{A}$ jest rzadka, to także czynnik Cholesky'ego macierzy $\mathbf{M} = \mathbf{H}\mathbf{H}^T$ jest rzadki.

Niepełny rozkład Cholesky'ego — algorytm w wersji *GAXPY*

```
for  k = 1:N
     H_kk = A_kk
     for  j = 1:k-1
          H_kk = H_kk - H_kj^2
     end
     H_kk = sqrt(H_kk)
     for  l = k+1:N
          H_lk = A_lk
          if  A_lk ≠ 0
               for  j = 1:k-1
                    H_lk = H_lk - H_lj H_kj
               end
               H_lk = H_lk / H_kk
          endif
     end
end
```

Uwagi

- Ponieważ $\mathbf{A}$ jest rzadka, powyższy algorytm na obliczanie przybliżonego czynnika Cholesky'ego wykonuje się szybko. Wykonuje się go *tylko raz*.
- Równanie $\mathbf{M}\mathbf{z} = \mathbf{r}$ rozwiązuje się szybko, gdyż znamy czynnik Cholesky'ego $\mathbf{M} = \mathbf{H}\mathbf{H}^T$.
- Obliczone $\mathbf{H}$ jest rzadkie, a zatem równanie $\mathbf{M}\mathbf{z} = \mathbf{r}$ rozwiązuje się szczególnie szybko.
- Mamy nadzieję, że macierz $\tilde{\mathbf{A}}$ ma współczynnik uwarunkowania bliski jedności, a zatem nie potrzeba wielu iteracji (7).

Przykład — macierz pasmowa z pustymi diagonalami

Rozważmy macierz o następującej strukturze:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & b_3 & 0 & c_5 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & a_2 & 0 & b_4 & 0 & c_6 & 0 & 0 & 0 & 0 & \dots \\ b_3 & 0 & a_3 & 0 & b_5 & 0 & c_7 & 0 & 0 & 0 & \dots \\ 0 & b_4 & 0 & a_4 & 0 & b_6 & 0 & c_8 & 0 & 0 & \dots \\ c_5 & 0 & b_5 & 0 & a_5 & 0 & b_7 & 0 & c_9 & 0 & \dots \\ 0 & c_6 & 0 & b_6 & 0 & a_6 & 0 & b_8 & 0 & c_{10} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{10}$$

Macierz ta jest symetryczna, zakładamy też, że jest dodatnio określona.

Niepełny czynnik Cholesky'ego macierzy (10) ma postać

$$\mathbf{H} = \begin{bmatrix} p_1 & & & & & & \\ 0 & p_2 & & & & & \\ q_3 & 0 & p_3 & & & & \\ 0 & q_4 & 0 & p_4 & & & \\ r_5 & 0 & q_5 & 0 & p_5 & & \\ 0 & r_6 & 0 & q_6 & 0 & p_6 & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{11}$$

(W pełnym czynniku Cholesky'ego macierzy (10) zera leżące w (11) *pomiędzy* diagonalą "p" a diagonalną "r" znikłyby — w ogólności mogłyby tam znajdować się jakieś niezerowe liczby.)

Zgodnie z podanym algorytmem, elementy ciągów $\{p_k\}$, $\{q_k\}$, $\{r_k\}$ wyliczamy z następujących wzorów:

$$
\begin{array}{rcl@{\qquad}rcl}
p_1 & = & \sqrt{a_1}, & p_2 & = & \sqrt{a_2}, \\
q_3 & = & b_3/p_1, & q_4 & = & b_4/p_2, \\
r_5 & = & c_5/p_1, & r_6 & = & c_6/p_2, \\
p_3 & = & \sqrt{a_3 - q_3^2}, & p_4 & = & \sqrt{a_4 - q_4^2}, \\
q_5 & = & (b_5 - r_5 q_3)/p_3, & q_6 & = & (b_6 - r_6 q_4)/p_4, \\
r_7 & = & c_7/p_3, & r_8 & = & c_7/p_4, \\
p_5 & = & \sqrt{a_5 - q_5^2 - r_5^2}, & p_6 & = & \sqrt{a_6 - q_5^6 - r_6^2}, \\
q_7 & = & (b_7 - r_7 q_5)/p_5, & q_8 & = & (b_8 - r_8 q_6)/p_6, \\
r_9 & = & c_9/p_5, & r_{10} & = & c_{10}/p_6, \\
p_7 & = & \sqrt{a_7 - q_7^2 - r_7^2}, & p_8 & = & \sqrt{a_8 - q_8^6 - r_8^2}, \\
q_9 & = & (b_9 - r_9 q_7)/p_7, & q_{10} & = & (b_{10} - r_{10} q_8)/p_8, \\
r_{11} & = & c_{11}/p_7, & r_{12} & = & c_{12}/p_8, \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}
$$

Macierze niesymetryczne

Jeżeli w równaniu

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{12}$$

macierz $\mathbf{A}$ nie jest symetryczna i dodatnio określona, sytuacja się komplikuje. Zakładając, że $\det \mathbf{A} \neq 0$, równanie (12) możemy "zsymetryzować" na dwa sposoby.
CGNR:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}\,, \tag{13}$$

lub CGNE:

$$\begin{aligned} \mathbf{A}\mathbf{A}^T\mathbf{y} &= \mathbf{b}\,, && \text{(14a)} \\ \mathbf{x} &= \mathbf{A}^T\mathbf{y}\,. && \text{(14b)} \end{aligned}$$

Do dwu powyższych równań formalnie rzecz biorąc *można* używać metody gradientów sprzężonych. Trzeba jednak pamiętać, że nawet jeśli macierz $\mathbf{A}$ jest rzadka, macierze $\mathbf{A}^T\mathbf{A}$, $\mathbf{A}\mathbf{A}^T$ nie muszą być rzadkie, a co gorsza, ich współczynnik uwarunkowania jest kwadratem współczynnika uwarunkowania macierzy wyjściowej.

Alternatywnie, zamiast "symetryzować" macierz, można zmodyfikować algorytm, tak aby zamiast dwu, generował on *cztery* ciągi wektorów. Należy jednak pamiętać, że dla wielu typów macierzy taki algorytm bywa bardzo wolno zbieżny, a niekiedy nawet dochodzi do kompletnej stagnacji przed uzyskaniem rozwiązania:

Metoda gradientów bi-sprzężonych (*Bi-Conjugate Gradients*, Bi-CG)

$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1$, $\mathbf{p}_1 = \mathbf{r}_1$, $\bar{\mathbf{r}}_1 \neq 0$ dowolny, $\bar{\mathbf{p}}_1 = \bar{\mathbf{r}}_1$

while $\|\mathbf{r}_k\| > \varepsilon$

$$
\begin{aligned}
\alpha_k &= \frac{\bar{\mathbf{r}}_k^T \mathbf{r}_k}{\bar{\mathbf{p}}_k^T \mathbf{A}\mathbf{p}_k} \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
\bar{\mathbf{r}}_{k+1} &= \bar{\mathbf{r}}_k - \alpha_k \mathbf{A}^T \bar{\mathbf{p}}_k \\
\beta_k &= \frac{\bar{\mathbf{r}}_{k+1}^T \mathbf{r}_{k+1}}{\bar{\mathbf{r}}_k^T \mathbf{r}_k} \\
\mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \\
\bar{\mathbf{p}}_{k+1} &= \bar{\mathbf{r}}_{k+1} + \beta_k \bar{\mathbf{p}}_k \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k
\end{aligned}
\tag{15}
$$

end

Wektory wygenerowane w algorytmie (15) spełniają następujące relacje:

$$\bar{\mathbf{r}}_i^T \mathbf{r}_j = \mathbf{r}_i^T \bar{\mathbf{r}}_j = 0\,,\ i > j\,, \tag{16a}$$
$$\bar{\mathbf{r}}_i^T \mathbf{p}_j = \mathbf{r}_i^T \bar{\mathbf{p}}_j = 0\,,\ i > j\,, \tag{16b}$$
$$\bar{\mathbf{p}}_i^T \mathbf{A} \mathbf{p}_j = \mathbf{p}_i^T \mathbf{A}^T \bar{\mathbf{p}}_j = 0\,,\ i > j\,. \tag{16c}$$

Jeżeli w algorytmie (15) weźmiemy $\bar{\mathbf{r}}_1 = \mathbf{A}\mathbf{r}_1$, we wszystkich krokach zachodzić będzie $\bar{\mathbf{r}}_k = \mathbf{A}\mathbf{r}_k$ oraz $\bar{\mathbf{p}}_k = \mathbf{A}\mathbf{p}_k$. Jest to wersja przydatna dla rozwiązywania układów równań z macierzami symetrycznymi, ale nieokreślonymi dodatnio. Jest to przy okazji szczególny wariant algorytmu GMRES (*generalised minimum residual*), formalnie odpowiadającego minimalizacj funkcjonału

$$\Phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2\,. \tag{17}$$

