Metody Numeryczne
Zajęcia nr 3

Michał Bernardelli

Do zapamiętania: wskaźnik uwarunkowania macierzy, metoda iteracyjna, metoda Jacobiego, metoda Gaussa-Seidela, metoda Richardsona, macierz iteracji, zbieżność metod iteracyjnych, dobór optymalnego parametru metody Richardsona, preconditioner, metoda gradientów sprzężonych (CG), metoda gradientów sprzężonych z preconditionerem (PCG)

1 Metody iteracyjne

Wiele metod iteracyjnych dla układów równań liniowych $Ax = b$ można zapisać w postaci

$$Mx_{k+1} = Nx_k + b,$$

gdzie $A = M - N$. Przykładowo dla macierzy $A = -E + D - F$, gdzie $-E$ oznacza macierz z niezerowymi elementami pod, a $-F$ z niezerowymi elementami nad główną diagonalą D, mamy:

Metoda	M	N	$G = M^{-1}N$
Jacobi	D	$E + F$	$D^{-1}(E + F)$
Gauss-Seidel	$D - E$	F	$(D - E)^{-1}F$
Backward Gauss-Seidel	$D - F$	E	$(D - F)^{-1}E$
Richardson	$I\frac{1}{\tau}$	$\frac{1}{\tau}(I - \tau A)$	$I - \tau A$

Podstawowe pytanie stanowi lista warunków, jakie musi spełniać macierz iteracji $G = M^{-1}N$, aby metoda była zbieżna. Jeżeli przez $e_k = x_k - x^*$ oznaczymy błąd na k-tym kroku iteracji (gdzie x^* oznacza rozwiązanie układu równań), to

$$e_k = x_k - x^* = (Gx_{k-1} + M^{-1}b) - (Gx^* + M^{-1}b) = G(x_{k-1} - x^*) = Ge_{k-1} = G^k e_0.$$

Macierz G nazywamy macierzą iteracji. Można pokazać, że metoda iteracyjna dana przez macierz G jest zbieżna wtedy i tylko wtedy, gdy

$$\rho(G) = \max_i |\lambda_i(G)| < 1,$$

gdzie $\lambda_i(G)$ to wartości własne macierzy G.

Zadanie 1
Wykazać, że metoda iteracyjna Richardsona

$$x_{k+1} = x_k + \tau(b - Ax_k)$$

jest zbieżna dla

$$\tau \in \left(0, \frac{2}{\lambda_{\max}}\right).$$

Zadanie 2
Wykazać, że optymalny parametr τ dla metody iteracyjnej Richardsona wyraża się wzorem

$$\tau_{opt} = \frac{2}{\lambda_{\max} + \lambda_{\min}}.$$

Zadanie 3
Dany jest układ równań $Ax = b$ z macierzą

$$A = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 4 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 4 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 4 \end{bmatrix}$$

rozmiaru n oraz wektorem prawej strony złożonym z jedynek. Zbadać szybkość zbieżności metod

1. *Jacobiego*
$$Dx_{k+1} = (E + F)x_k + b,$$
2. *Richardsona (dla różnych τ, w tym optymalnego)*
$$x_{k+1} = x_k + \tau(b - Ax_k)$$
3. *Gaussa-Seidela*
$$(D - E)x_{k+1} = Fx_k + b,$$

do rozwiązania x^* tego układu równań w zależności od parametru $\alpha \in \mathbb{R}$ oraz rozmiaru macierzy $n \in \mathbb{N}$. Zwrócić uwagę na kryteria stopu.

Zadanie 4
Wykazać, że metoda Jacobiego dla macierzy ściśle diagonalnie dominujących jest zbieżna.

Zadanie 5
Wykazać, że metoda Gaussa-Seidel'a dla macierzy ściśle diagonalnie dominujących jest zbieżna.

Zadanie 6
Sprawdzić empirycznie, czy któraś z metod iteracyjnych:

1. *Jacobiego,*
2. *Gaussa-Seidel'a,*
3. *Richardsona,*
4. *PCG*

jest zbieżna dla macierzy Hilberta.

Zadanie 7

Dla macierzy trójdiagonalnej

$$A = \begin{bmatrix} \alpha & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & \alpha & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & \alpha & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & \alpha & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & \alpha \end{bmatrix},$$

różnych α oraz wektora prawej strony złożonego z jedynek zbadać szybkość zbieżności metody CG dla różnych macierzy preconditionera:

1. *macierzy jednostkowej,*
2. *Jacobiego,*
3. *Gaussa-Seidel'a,*
4. A^{-1}.

Zadanie 8

Napisać program rozwiązujący metodą eliminacji Gaussa (bez wyboru elementu głównego) układ równań (tzw. strzałka Wilkinsona):

$$\begin{pmatrix} a_1 & 0 & 0 & \cdots & 0 & b_1 \\ 0 & a_2 & 0 & \cdots & 0 & b_2 \\ 0 & 0 & a_3 & \cdots & 0 & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} \\ c_1 & c_2 & c_3 & \cdots & c_{n-1} & a_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}.$$

Jaki jest koszt działania algorytmu?