Metoda SOR

1 Opis metody

Metoda SOR (*Successive Overrelaxation*) jest jedną z metod służących do iteracyjnego (przybliżonego) rozwiązywania układów równań postaci

$$Ax = y, \tag{1}$$

gdzie $A \in M_{n\times n}$, $y \in \mathbb{R}^n$, $n \in \mathbb{N}$ są dane a szukanym jest $x \in \mathbb{R}^n$. Metoda opiera się na przedstawieniu macierzy A w postaci

$$A = \left(\frac{1}{\omega}D + L\right) + \left(D + U - \frac{1}{\omega}D\right)$$

gdzie macierze D, L i U są odpowiednio: daigonalną, dolną trójkątną i górną trójkątną składową macierzy A, zaś $\omega > 0$ jest parametrem metody. Zauważmy, że układ równań (1) można wówczas przedstawić w postaci

$$\left(\frac{1}{\omega}D + L\right)x = \left(\frac{1}{\omega}D - D - U\right)x + y, \tag{2}$$

lub równoważnie

$$(D + \omega L)\,x = (D - \omega(D + U))\,x + \omega y. \tag{3}$$

Wzór (3) stanowi podstawę do zdefiniowania metody SOR. Polega ona na wyznaczeniu ciągu wektorów $x^{(0)}, x^{(1)}, x^{(2)}, ...$, gdzie $x^{(0)}$ jest ustalonym wektorem początkowym, natomiast pozostałe wektory spełniają zależność:

$$(D + \omega L)\,x^{(k+1)} = (D - \omega(D + U))\,x^{(k)} + \omega y, \quad k = 1, 2, ... \tag{4}$$

Następujące twierdzenie mówi o zbieżności metody SOR.

Twierdzenie 1 *Jeżeli macierz A jest symetryczna i dodatnio określona, to ciąg $\{x^{(k)}\}_{k=0}^{\infty}$ zdefiniowany wzorem (4) jest zbieżny do rozwiązania układu równań (1) dla dowolnego wektora początkowego $x^{(0)}$ i dowolnego $\omega \in (0, 2)$.*

2 Opis zadania

Zadanie polega na napisaniu programu rozwiązującego metodą SOR układ równań (1), w którym macierz A jest macierzą rzadką (większość jej elementów jest równych 0) mającą postać wielodiagonalną. Oznacza to, że elementy niezerowe mogą występować tylko na diagonali oraz na pewnej liczbie "wstęg" skupionych wzdłuż diagonali. Zakładamy przy tym, że powyżej jak i poniżej diagonali znajduje się taka sama liczba niezerowych "wstęg". Przykładowa macierz A może wyglądać następująco:

$$A = \begin{bmatrix} c_1 & b_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_1 & c_2 & b_2 & a_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ e_1 & d_2 & c_3 & b_3 & a_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & e_2 & d_3 & c_4 & b_4 & a_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_3 & d_4 & c_5 & b_5 & a_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & e_4 & d_5 & c_6 & b_6 & a_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_5 & d_6 & c_7 & b_7 & a_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & e_6 & d_7 & c_8 & b_8 & a_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_7 & d_8 & c_9 & b_9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e_8 & d_9 & c_{10} \end{bmatrix} \tag{5}$$

Dane wejściowe zapisane są w następującym formacie:

N
M
wstęga pierwsza
wstęga druga
...
wstęga M-ta
prawa strona równania
wektor początkowy
ω
L

gdzie N oznacza rozmiar macierzy, M liczbę niezerowych "wstęg", ω jest parametrem metody, a L oznacza liczbę iteracji, którą mamy wykonać. Przykładowo plik zawierający następujące dane:

10
5
$a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8$
$b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9$
$c_1\ c_2\ c_3\ c_4\ c_5\ c_6\ c_7\ c_8\ c_9\ c_{10}$
$d_1\ d_2\ d_3\ d_4\ d_5\ d_6\ d_7\ d_8\ d_9$
$e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7\ e_8$
$y_1\ y_2\ y_3\ y_4\ y_5\ y_6\ y_7\ y_8\ y_9\ y_{10}$
$x_1^0\ x_2^0\ x_3^0\ x_4^0\ x_5^0\ x_6^0\ x_7^0\ x_8^0\ x_9^0\ x_{10}^0$
1.5
4

stanowić będzie zbiór wejściowy dla zadania polegającego na wykonaniu 4 iteracji metody SOR z parametrem $\omega = 1.5$ dla układu równań (1) z macierzą A określoną wzorem (5), prawą stroną $y = (y_1, y_2, ..., y_{10})$ i wektorem początkowym $x^0 = (x_1^0, x_2^0, ..., x_{10}^0)$. Jako dane wyjściowe program ma zwrócić wektor końcowy (w tym przypadku x^4 gdyż są 4 iteracje), a więc przykładowy zbiór wyjsciowy powinien wyglądać następująco:

x_1^4 x_2^4 x_3^4 x_4^4 x_5^4 x_6^4 x_7^4 x_8^4 x_9^4 x_{10}^4

przy czym wyniki należy wypisać używając notacji naukowej uwzględniającej 16 miejsc po przecinku.

3 Przykłady

Poniżej podajemy dwa przykłady konkretnych danych wejściowych oraz odpowiadających im wyników realizacji programu.

Przykład 1

Dane wejściowe (plik *1.in*):

7
5
1 2 1 2 1
2 -1 3 1 3 -1
5 6 7 8 9 10 11
2 -1 3 1 3 -1
1 2 1 2 1
8 9 11 16 15 14 11
2 3 2 3 2 3 2
1.5
1
Dane wyjściowe (plik *1.out*):

$-1.0000000000000000e+000$ $2.5000000000000000e-001$
$-7.3214285714285721e-001$ $3.1808035714285765e-001$
$-2.6432291666666696e-001$ $9.2352120535714288e-001$
$6.6197874391233791e-001$

Przykład 2

Dane wejściowe (plik *2.in*):

10
3
1 -1 2 1 3 -1 1 2 2
8 9 10 9 8 5 10 9 7 5
2 3 1 2 1 2 3 1 2
7 -8 5 -7 3 -5 7 -5 4 -3

2 1 2 1 2 1 2 1 2 1
1.3
5

Dane wyjściowe (plik *2.out*):

$1.0105797917402293e+000 \quad -9.6616275117973560e-001$
$9.5001966298893947e-001 \quad -9.8262499373334167e-001$
$1.0460264364569309e+000 \quad -9.7058548895514807e-001$
$9.8014935398247816e-001 \quad -1.1039531515056584e+000$
$1.0218890063497770e+000 \quad -1.0067887181977542e+000$

Uwagi:

1. W obu przykładach wyniki są wypisywane wierszami a linia została złamana z powodu braku miejsca w wierszu.

2. Z uwagi na błędy zaokrągleń wyniki będą porównywane z pewnym dopuszczalnym błędem zarówno bezwzględnym ε_a (zazwyczaj około 10^{-25}), jak i względnym ε_r (zazwyczaj około 10^{-8}). Dokładnie, jeżeli oczekiwany wynik to x a program zwróci $\bar{x}$, to wynik będzie odrzucony, jeżeli $|x-\bar{x}| > \varepsilon_a$ oraz $|\frac{x-\bar{x}}{x}| > \varepsilon_r$ dla $x \neq 0$.