

Politechnika Poznańska	
Sprawozdanie nr 2	Krystian Baran 145000
Metody iteracyjne rozwiązywania układów równań	13/06/2021

Cel ćwiczenia:

Celem ćwiczenia było zapoznanie się z metodami iteracyjnymi rozwiązywania układów równań liniowych i porównanie ich czasów działania dla dużych i rzadkich macierzy współczynników.

Kod sprawozdania:

```

clc, clear, format long;
warning('off','all');

A = gallery("moler", 100);
disp("cond")
cond(A)
N = length(A);
X0 = zeros(N,1);
B = A*ones(N,1);
tol1 = 10^(-3);
tol2 = 10^(-6);

disp("Jacobi")
[X, N, err] = JacobiMethod(A,transpose(B), X0, 1000, tol1);
disp(N)
disp(err)
[X, N, err] = JacobiMethod(A,transpose(B), X0, 1000, tol2);
disp(N)
disp(err)

disp("Gauss Seidel")
[X, N, err] = GaussSeidelMethod(A,transpose(B),X0, 1000, tol1);
disp(N)
disp(err)
[X, N, err] = GaussSeidelMethod(A,transpose(B),X0, 1000, tol2);
disp(N)
disp(err)

disp("SOR")
[X, N, err] = SORMethod(A,transpose(B),X0, 1000, tol1);
disp(N)
disp(err)
[X, N, err] = SORMethod(A,transpose(B),X0, 1000, tol2);
disp(N)
disp(err)

disp("Fast Fall")
[X, N, err] = fastFallMethod(A,B, X0, 1000, tol1);
disp(N)

```

```

disp(err)
[X, N, err] = fastFallMethod(A,B, X0, 1000, tol2);
disp(N)
disp(err)

disp("Conj Gradient")
[X, N, err] = ConjugateGradientMethod(A,B,X0, 1000, tol1);
disp(N)
disp(err)
[X, N, err] = ConjugateGradientMethod(A,B,X0, 1000, tol2);
disp(N)
disp(err)

```

Przebieg ćwiczenia:

Do ćwiczenia została wybrana macierz z dostępna w GNU Octave pod **galley(„moler”, n)**. Jest to macierz symetryczna i dodatnio określona macierz Molera. To, że macierz jest symetryczna i dodatnio określona jest warunkiem zbieżności niektórych metod, zatem tego typu macierz jest odpowiednia do tego rozważania.

Przedstawiono poniżej macierz wymiaru 4×4 i przykładowe obliczenia każdą z metod. Obliczenia zostały wykonane dla maksymalnej liczby iteracji 1000 i dla dwóch tolerancji odpowiednio 10^{-3} i 10^{-6} .

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 3 & 1 \\ -1 & 0 & 1 & 4 \end{bmatrix}$$

Dla tej macierzy współczynnik uwarunkowania obliczony funkcją jest :

$$\text{cond}(A) = 153.5275998653943$$

Macierz wyrazów wolnych została wygenerowana tak, żeby rozwiązanie równania było wektorem jedynek. Macierz ta przedstawiono także poniżej.

$$B = \begin{bmatrix} -2 \\ 1 \\ 3 \\ 4 \end{bmatrix}$$

Dla metody Jacobiego uzyskano następujące wyniki posortowane według tolerancji:

$$x_1 = \begin{bmatrix} 7.076641936998668e+53 \\ -3.125773931399195e+53 \\ -2.721029535341054e+53 \\ -2.163830412030273e+53 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 7.076641936998668e+53 \\ -3.125773931399195e+53 \\ -2.721029535341054e+53 \\ -2.163830412030273e+53 \end{bmatrix}$$

Dla obu tolerancji zostało wykonane maksymalna ilość kroków. Widzimy, że nie został uzyskany pożądany wynik, zatem metoda Jacobiego nie jest przydatna to tej macierzy. Normy błędu zostały zapisane w tabeli poniżej:

	Norma Euklidesowa	Norma Maksimum
--	-------------------	----------------

x_1	2.860740069651245e+54	1.630148462102018e+54
x_2	2.860740069651245e+54	1.630148462102018e+54

Dla metody Gaussa Seidela uzyskano następujące wyniki:

$$x_1 = \begin{bmatrix} 9.99999999999929e-01 \\ 9.99999999999964e-01 \\ 9.99999999999980e-01 \\ 9.99999999999987e-01 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 9.99999999999929e-01 \\ 9.99999999999964e-01 \\ 9.99999999999980e-01 \\ 9.99999999999987e-01 \end{bmatrix}$$

Wyniki dla obu tolerancji są takie same i są w przybliżeniu poprawnie, natomiast została użyta maksymalna liczba kroków dla obu tolerancji. Poniżej przedstawiono normy błędów:

	Norma Euklidesowa	Norma Maksimum
x_1	9.165151389911680e+00	1.400000000000000e+01
x_2	9.165151389911680e+00	1.400000000000000e+01

Dla metody SOR Omega uzyskano następujące wyniki:

$$x_1 = \begin{bmatrix} 9.795207513735043e-01 \\ 9.897603756867521e-01 \\ 9.944832797512497e-01 \\ 9.962593679055637e-01 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 9.999791846130899e-01 \\ 9.999895923065449e-01 \\ 9.999943927304880e-01 \\ 9.999961979706506e-01 \end{bmatrix}$$

Wyniki są mniej dokładne niż dla poprzedniej metody, natomiast została wykonana mniejsza liczba kroków. Liczba kroków wraz z normami zapisano w tabeli poniżej:

	N	Norma Euklidesowa	Norma Maksimum
x_1	206	1.000186885723105e-03	9.822719700611771e-04
x_2	489	1.016603557686786e-06	9.983945936653527e-07

Dla metody szybkiego spadku uzyskano następujące wyniki:

$$x_1 = \begin{bmatrix} 9.623440301937382e-01 \\ 9.808081489715292e-01 \\ 9.895278276719417e-01 \\ 9.929910458722796e-01 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 9.999619847065191e-01 \\ 9.999806250150229e-01 \\ 9.999894278993083e-01 \\ 9.999929241644943e-01 \end{bmatrix}$$

Widzimy, że wyniki mają mniejszą dokładność niż dla poprzedniej metody, natomiast dla pierwszej tolerancji zostało wykonane mniej kroków niż dla poprzedniej metody, dla drugiej tolerancji wykonano ich więcej. Tabela z krokami i normami błędów zapisana poniżej.

	N	Norma Euklidesowa	Norma Maksimum
x_1	201	1.677479306945879e-03	9.829923220121550e-04
x_2	579	1.693486278405312e-06	9.923723065519141e-07

Dla metody sprzężonych gradientów uzyskano następujące wyniki:

$$x_1 = \begin{bmatrix} 9.999999999999938e-01 \\ 1.000000000000001e+00 \\ 1.000000000000008e+00 \\ 1.000000000000012e+00 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 9.999999999999938e-01 \\ 1.000000000000001e+00 \\ 1.000000000000008e+00 \\ 1.000000000000012e+00 \end{bmatrix}$$

Możemy zauważyć, że uzyskane wyniki mają lepszą dokładność niż poprzednie metody, a także, że została wykonana znacznie mniejsza liczba kroków do uzyskania wyniku. Jak poprzednio liczba kroków i norma błędów zapisana w tabeli poniżej.

	N	Norma Euklidesowa	Norma Maksimum
x_1	4	8.208560743427264e-14	1.818181818181991e-01
x_2	4	8.208560743427264e-14	1.818181818181991e-01

W następnym kroku przeszliśmy do badania liczby kroków dla coraz większych macierzy. Wymiary badane są: 50, 100, 200, 400, 800, 1600. Liczbę kroków i normy błędów dla każdego wymiaru zapisano w osobnych tabelach.

cond	2.132606752E+17					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	1000	1.86292411E+04	4.04250000E+04	1000	1.86292411E+04	4.04250000E+04
SOR	576	2.35754733E-03	9.98544802E-04	1000	2.26995513E-06	1.08552672E-06
FastFall	826	1.34853897E-03	9.98650035E-04	1000	3.69363165E-04	2.80681665E-04
Conj	13	2.79726514E-03	5.44167508E-04	19	2.70158999E-06	9.36592943E-07

Wyniki dla n = 50

cond	4.823418150E+17					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	1000	1.49061881E+05	3.28350000E+05	1000	1.49061881E+05	3.28350000E+05
SOR	1000	5.95783019E-02	1.59862674E-02	1000	5.95783019E-02	1.59862674E-02
FastFall	1000	1.68573559E-01	8.37115268E-02	1000	1.68573559E-01	8.37115268E-02
Conj	21	5.75464175E-04	4.80077227E-04	32	4.92524762E-07	4.80077227E-04

Wyniki dla n = 100

cond	5.094155864E+16					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	1000	1.19255095E+06	2.64670000E+06	1000	1.19255095E+06	2.64670000E+06
SOR	1000	4.57794847E+00	6.92904604E-01	1000	4.57794847E+00	6.92904604E-01
FastFall	1000	2.26326073E+00	9.57483833E-01	1000	2.26326073E+00	9.57483833E-01
Conj	32	1.20653311E-02	8.89913806E-04	53	1.67078228E-06	8.60398186E-07

Wyniki dla n = 200

cond	9.694890144E+16					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	1000	9.54051949E+06	2.12534000E+07	1000	9.54051949E+06	2.12534000E+07
SOR	1000	7.04596505E+01	6.94635798E+00	1000	7.04596505E+01	6.94635798E+00
FastFall	1000	1.49648078E+01	4.67522879E+00	1000	1.49648078E+01	4.67522879E+00
Conj	55	1.06959965E-02	6.64261174E-04	90	1.70620454E-05	7.02835921E-07

Wyniki dla n = 400

cond	2.664818070E+16					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	1000	7.63243792E+07	1.70346800E+08	1000	7.63243792E+07	1.70346800E+08
SOR	1000	6.50844346E+02	4.37105058E+01	1000	6.50844346E+02	4.37105058E+01
FastFall	1000	8.74761371E+01	1.95685935E+01	1000	8.74761371E+01	1.95685935E+01
Conj	97	5.12205090E-03	7.68431638E-04	163	1.72448159E-05	9.54329803E-07

Wyniki dla n = 800

cond	1.676692082E+16					
Metoda	tol1			tol2		
	n	Norm2	Norm Max	n	Norm2	Norm Max
Jacobi	1000	NaN	NaN	1000	NaN	NaN
Gauss	-	-	-	-	-	-
SOR	-	-	-	-	-	-
FastFall	-	-	-	-	-	-
Conj	173	1.15641555E-02	8.83091776E-04	305	1.00424856E-05	7.93589882E-07

Wyniki dla n = 1600

Możemy zauważyć, że współczynnik uwarunkowania macierzy zmniejsza się ze zwiększeniem rozmiaru badanej macierzy, natomiast ze zwiększeniem rozmiaru dużo metod przestaje być skutecznych. Metoda Jacobiego przestała być skuteczna od $n = 50$, natomiast każda inna metoda, z wyjątkiem metody sprzężonych gradientów, przestała być skuteczna od $n = 100$. Przy czym to, że metoda nie jest już skuteczna, gdy jedna z dwóch przedstawionych norm jest większa niż żądana dokładność.

Metoda sprzężonych gradientów okazała się najbardziej skuteczna dla tego typu macierzy, ponieważ, nawet przy dużym rozmiarze macierzy współczynników, metoda wykonała mniej niż połowę dozwolonych kroków.

Dla $n = 1600$ czas obliczania innych metod był zbyt duży, żeby otrzymać wynik w przeznaczonym czasie, zatem jedynie obliczono dla ostatniej metody i, ponieważ metoda Jacobiego nie oddawała

dla każdego wymiaru te same wyniki, także takie wyniki przepisano dla tego rozmiaru. Możemy się spodziewać, że dla reszty metod norma błędu rosłaby aż do osiągnięcia „Inf” lub „NaN”.

Wnioski:

Na podstawie badanych wyników widzimy że dla danej macierzy ważny jest dobór metody iteracyjnej którą chcemy rozwiązać układ i, że dla badanej macierzy, metoda Jacobiego okazała się najmniej skuteczna w przeciwieństwie do metody sprzężonych gradientów, która okazała się najbardziej skuteczna.