Sprawozdanie
Metody Numeryczne 2, laboratorium 1

Grzegorz Rozdzialik (D4, grupa lab. 2)

18 października 2016

1 Zadanie

Rozwiązywanie układu równań z macierzą trójdiagonalną w dziedzinie zespolonej metodą Gaussa-Seidla w tył (Backwards Gauss-Seidel).

Szukamy rozwiązania układu równań $Ax = b$, gdzie $A \in \mathbb{C}^{n \times n}$ (A - macierz trójdiagonalna, $\det A \neq 0$), $b \in \mathbb{C}^n$, $n \in \mathbb{N}$.

Układ $Ax = b$ można zapisać w postaci:

$$\begin{bmatrix} d_1 & u_1 & 0 & 0 & 0 & & \dots & & 0 \\ l_2 & d_2 & u_2 & 0 & 0 & & \dots & & 0 \\ 0 & l_3 & d_3 & u_3 & 0 & & \dots & & 0 \\ 0 & 0 & l_4 & d_4 & u_4 & & \dots & & 0 \\ & & & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \vdots & & 0 & l_{n-3} & d_{n-3} & u_{n-3} & 0 & 0 \\ & & & & 0 & l_{n-2} & d_{n-2} & u_{n-2} & 0 \\ & & & & & 0 & l_{n-1} & d_{n-1} & u_{n-1} \\ 0 & 0 & \dots\dots\dots\dots & & & & 0 & l_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \tag{1}$$

gdzie $l_i, d_i, u_i, b_i, x_i \in \mathbb{C}^n$ dla każdego $i = 1, \dots, n$.

2 Opis metody

Metoda Gaussa-Seidla w tył jest iteracyjną metodą rozwiązywania układów równań liniowych. Jest ona podobna do zwykłej metody Gaussa-Seidla, przy czym w tym przypadku obliczenia rozpoczynamy od ostatnich elementów wektora rozwiązań.

Mając przybliżenie początkowe $x^{(0)} \in \mathbb{C}^n$ tworzymy ciąg kolejnych przybliżeń rozwiązań $\left\{x^{(k)}\right\}$ $(k = 1, 2, \dots)$ taki, że

$$\lim_{k \to +\infty} x^{(k)} = x$$

Po wymnożeniu lewej strony układu (1) otrzymujemy następujący układ równań:

$$\begin{cases} d_1x_1 + u_1x_2 = b_1 \\ l_2x_1 + d_2x_2 + u_2x_3 = b_2 \\ l_3x_2 + d_3x_3 + u_3x_4 = b_3 \\ \vdots \\ l_ix_{i-1} + d_ix_i + u_ix_{i+1} = b_i \\ \vdots \\ l_{n-1}x_{n-2} + d_{n-1}x_{n-1} + u_{n-1}x_n = b_{n-1} \\ l_nx_{n-1} + d_nx_n = b_n \end{cases}$$

Następnie z i-tego równania wyliczamy:

$$x_i = \frac{b_i - l_ix_{i-1} - u_ix_{i+1}}{d_i} \qquad \text{dla } i = 2, \dots, n-1$$

Wyjątkiem są równania pierwsze oraz ostatnie, ponieważ nie istnieją współczynniki odpowiednio l_1 oraz u_n, stąd:

$$x_1 = \frac{b_1 - u_1x_2}{d_1}$$

$$x_n = \frac{b_n - l_nx_{n-1}}{d_n}$$

W k-tym kroku obliczamy przybliżenie $x^{(k+1)}$ w taki sposób, że najpierw obliczamy $x_n^{(k+1)}$, następnie zmniejszamy indeks dolny, obliczając kolejno $x_{n-1}^{(k+1)}, \ldots, x_1^{(k+1)}$.

Przy wyliczaniu $x_i^{(k+1)}$ używamy najbardziej aktualnych przybliżeń (zamiast $x_{i+1}^{(k)}$ używamy $x_{i+1}^{(k+1)}$, wyjątkiem jest $x_n^{(k)}$).

Należy oddzielnie przybliżać część rzeczywistą oraz część urojoną.

3 Implementacja metody

Wprowadźmy następujące oznaczenia:

$$
\begin{aligned}
u_{r,j} &= \mathrm{Re}(u_j) & u_{i,j} &= \mathrm{Im}(u_j) \\
d_{r,j} &= \mathrm{Re}(d_j) & d_{i,j} &= \mathrm{Im}(d_j) \\
l_{r,j} &= \mathrm{Re}(l_j) & l_{i,j} &= \mathrm{Im}(l_j) \\
x_{r,j}^{(k)} &= \mathrm{Re}(x_j^{(k)}) & x_{i,j}^{(k)} &= \mathrm{Im}(x_j^{(k)}) \\
b_{r,j} &= \mathrm{Re}(b_j) & b_{i,j} &= \mathrm{Im}(b_j)
\end{aligned}
$$

dla $j = 1, \ldots, n$.

Poniższe wzory opisują sposób obliczenia kolejnego przybliżenia $x^{(k+1)}$:

$$x_{r,n}^{(k+1)} = \frac{b_{r,n} + d_{i,n}x_{i,n}^{(k)} - l_{r,n}x_{r,n-1}^{(k)} + l_{i,n}x_{i,n-1}^{(k)}}{d_{r,n}}$$

$$x_{i,n}^{(k+1)} = \frac{b_{i,n} - d_{i,n}x_{r,n}^{(k)} - l_{r,n}x_{i,n-1}^{(k)} - l_{i,n}x_{r,n-1}^{(k)}}{d_{r,n}}$$

dla $j = (n-1), \ldots, 2$

$$x_{r,j}^{(k+1)} = \frac{b_{r,j} + d_{i,j}x_{i,j}^{(k)} - l_{r,j}x_{r,j-1}^{(k)} + l_{i,j}x_{i,j-1}^{(k)} - u_{r,j}x_{r,j+1}^{(k+1)} + u_{i,j}x_{i,j+1}^{(k+1)}}{d_{r,j}}$$

$$x_{i,j}^{(k+1)} = \frac{b_{i,j} - d_{r,j}x_{i,j}^{(k)} - l_{r,j}x_{i,j-1}^{(k)} - l_{i,j}x_{r,j-1}^{(k)} - u_{i,j}x_{r,j+1}^{(k+1)} - u_{r,j}x_{i,j+1}^{(k+1)}}{d_{r,j}}$$

$$x_{r,1}^{(k+1)} = \frac{b_{r,1} + d_{i,1}x_{i,1}^{(k)} - u_{r,1}x_{r,2}^{(k+1)} + u_{i,1}x_{i,2}^{(k+1)}}{d_{r,1}}$$

$$x_{i,1}^{(k+1)} = \frac{b_{i,1} - d_{r,1}x_{i,1}^{(k)} - u_{i,1}x_{r,2}^{(k+1)} - u_{r,1}x_{i,2}^{(k+1)}}{d_{r,1}}$$

4 Warunek stopu

Z możliwych warunków stopu wybrałem warunek Gill'a. Obliczenia trwają dopóki spełniona jest nierówność

$$\|x^{(k+1)} - x^{(k)}\| \geqslant \varepsilon\|x^{(k)}\| + \delta$$

gdzie $\varepsilon, \delta > 0$.

Dodatkowo w implementacji nałożyłem restrykcję na maksymalną ilość iteracji. Po 10000 iteracjach obliczenia nie będą kontynuowane.

5 Poprawność algorytmu

W ogólności metoda Gaussa-Seidla, a więc i metoda Gaussa-Seidla w tył, jest zbieżna globalnie dla macierzy diagonalnie silnie dominujących (kolumnowo lub wierszowo) oraz macierzy dodatnio określonych.

Po przeprowadzeniu testów odkryłem, że zaimplementowany algorytm działa dobrze, jeżeli część rzeczywista elementów z diagonali dominuje w rzędzie lub kolumnie, a część urojona tego elementu jest znacznie mniejsza od części rzeczywistej.

Jeżeli część urojona elementów z diagonali jest porównywalna do części rzeczywistej, to algorytm daje złe wyniki.

6 Przykłady

We wszystkich przykładach $\varepsilon = \text{eps} \approx 2.22 * 10^{-16}$, $\delta = 0$. Rzędy błędu zostały obliczone na podstawie porównania rozwiązania z rozwiązaniem obliczonym przez funkcję *linsolve* dostępną w Matlabie.

Układ 1 $n = 5$

Elementy na diagonali są znacznie większe od elementów na dwóch innych pasmach. Dodatkowo nie zawierają części urojonej.

$$\begin{bmatrix} 61354 & 44+59\,\mathrm{i} & 0 & 0 & 0 \\ 60+88\,\mathrm{i} & 83844 & 54+27\,\mathrm{i} & 0 & 0 \\ 0 & 84+40\,\mathrm{i} & 89650 & 4+42\,\mathrm{i} & 0 \\ 0 & 0 & 81+84\,\mathrm{i} & 48721 & 55+50\,\mathrm{i} \\ 0 & 0 & 0 & 80+63\,\mathrm{i} & 25148 \end{bmatrix} x = \begin{bmatrix} 20+62\,\mathrm{i} \\ 38+10\,\mathrm{i} \\ 71+37\,\mathrm{i} \\ 2+50\,\mathrm{i} \\ 28+32\,\mathrm{i} \end{bmatrix}$$

Znalezione rozwiązanie:

$$x = \begin{bmatrix} 0.000325764927490767 + 0.00101000840330470i \\ 0.000453671976952820 + 0.000117676750075580i \\ 0.000792072530469199 + 0.000411492905043098i \\ 4.04854673760366e-05 + 0.00102061129748172i \\ 0.00111583663410018 + 0.00126911883695549i \end{bmatrix}$$

Liczba iteracji: 9
Rząd błędu: -6
Czas działania: 0.290354ms

Układ 2 $n = 4$
Elementy na diagonali są znacznie większe od elementów na dwóch innych pasmach. Ich część urojona jest porównywalna z rzeczywistą.

$$\begin{bmatrix} 12339 + 18978\,\mathrm{i} & 22 + 91\,\mathrm{i} & 0 & 0 \\ 39 + 74\,\mathrm{i} & 17852 + 19484\,\mathrm{i} & 60 + 64\,\mathrm{i} & 0 \\ 0 & 95 + 35\,\mathrm{i} & 18710 + 12993\,\mathrm{i} & 24 + 14\,\mathrm{i} \\ 0 & 0 & 94 + 34\,\mathrm{i} & 15539 + 14995\,\mathrm{i} \end{bmatrix} x = \begin{bmatrix} 79 + 40\,\mathrm{i} \\ 80 + 56\,\mathrm{i} \\ 35 + 70\,\mathrm{i} \\ 33 + 83\,\mathrm{i} \end{bmatrix}$$

Znalezione rozwiązanie: **brak** - przekroczono maksymalną liczbę iteracji.
Czas działania: 97.203925ms

Układ 3 $n = 4$
Elementy na diagonali są porównywalne z elementami na dwóch innych pasmach. Ich część urojona jest porównywalna z rzeczywistą.

$$\begin{bmatrix} 19 + 53\,\mathrm{i} & 43 + 19\,\mathrm{i} & 0 & 0 \\ 71 + 11\,\mathrm{i} & 76\,\mathrm{i} & 44 + 50\,\mathrm{i} & 0 \\ 0 & 48 + 54\,\mathrm{i} & 21 + 65\,\mathrm{i} & 48 + 88\,\mathrm{i} \\ 0 & 0 & 28 + 55\,\mathrm{i} & 54 + 68\,\mathrm{i} \end{bmatrix} x = \begin{bmatrix} 83 + 32\,\mathrm{i} \\ 69 + 16\,\mathrm{i} \\ 71 + 74\,\mathrm{i} \\ 18 + 57\,\mathrm{i} \end{bmatrix}$$

Znalezione rozwiązanie: **brak** - przekroczono maksymalną liczbę iteracji.
Czas działania: 94.710700ms

Układ 4 $n = 5$
Elementy na diagonali są porównywalne z elementami na dwóch innych pasmach. Nie mają części urojonej.

$$\begin{bmatrix} 95 & 16 + 7\,\mathrm{i} & 0 & 0 & 0 \\ 17 + 72\,\mathrm{i} & 35 & 27 + 88\,\mathrm{i} & 0 & 0 \\ 0 & 83 + 93\,\mathrm{i} & 5 & 34 + 15\,\mathrm{i} & 0 \\ 0 & 0 & 3 + 56\,\mathrm{i} & 93 & 20 + 80\,\mathrm{i} \\ 0 & 0 & 0 & 1 + 83\,\mathrm{i} & 78 \end{bmatrix} x = \begin{bmatrix} 36 + 89\,\mathrm{i} \\ 30 + 41\,\mathrm{i} \\ 68 + 50\,\mathrm{i} \\ 5 + 36\,\mathrm{i} \\ 33 + 99\,\mathrm{i} \end{bmatrix}$$

Znalezione rozwiązanie: **brak** - przekroczono maksymalną liczbę iteracji.
Czas działania: 101.725476ms

Układ 5 $n = 5$

Elementy na diagonali są większe od elementów na dwóch innych pasmach. Nie mają części urojonej.

$$\begin{bmatrix} 317 & 15+66\,\mathrm{i} & 0 & 0 & 0 \\ 19+4\,\mathrm{i} & 363 & 67+41\,\mathrm{i} & 0 & 0 \\ 0 & 4+80\,\mathrm{i} & 337 & 90+66\,\mathrm{i} & 0 \\ 0 & 0 & 61+39\,\mathrm{i} & 394 & 68+40\,\mathrm{i} \\ 0 & 0 & 0 & 73+6\,\mathrm{i} & 258 \end{bmatrix} x = \begin{bmatrix} 62+86\,\mathrm{i} \\ 16+43\,\mathrm{i} \\ 33+58\,\mathrm{i} \\ 27+13\,\mathrm{i} \\ 54+58\,\mathrm{i} \end{bmatrix}$$

Znalezione rozwiązanie:

$$x = \begin{bmatrix} 0.210003815360937 + 0.259969953085646i \\ 0.0367475440953636 + 0.0776124641037519i \\ 0.0937110364191769 + 0.160199771371148i \\ 0.0603979474198726 - 0.0309935213767811i \\ 0.191492204302669 + 0.232171082852658i \end{bmatrix}$$

Liczba iteracji: 19
Rząd błędu: -1
Czas działania: 0.722562ms

Układ 6 $n = 100$

Elementy na diagonali porównywalne z elementami na dwóch innych pasmach (zarówno część rzeczywista, jak i urojona).

	BGS	linsolve
Rząd błędu	*brak*	4
Czas działania	901.49ms	0.43ms
Ilość iteracji	*nie dotyczy*	*nie dotyczy*

Układ 7 $n = 100$

Elementy na diagonali są większe (około ośmiokrotnie) od elementów na dwóch innych pasmach. Nie mają części urojonej.

	BGS	linsolve
Rząd błędu	1	4
Czas działania	0.26ms	0.45ms
Ilość iteracji	30	*nie dotyczy*

Układ 8 $n = 100$

Elementy na diagonali są znacznie większe (około pięćdziesięciokrotnie) od elementów na dwóch innych pasmach. Ich część urojona jest porównywalna z częściami urojonymi elementów z innych pasm.

	BGS	linsolve
Rząd błędu	1	4
Czas działania	0.11ms	0.37ms
Ilość iteracji	12	*nie dotyczy*

Układ 9 $n = 100$

Elementy na diagonali są znacznie większe (około pięćdziesięciokrotnie) od elementów na dwóch innych pasmach, zarówno część urojona, jak i rzeczywista.

	BGS	linsolve
Rząd błędu	*brak*	4
Czas działania	748.37ms	0.41ms
Ilość iteracji	10000	*nie dotyczy*

7 Wnioski

1. Aby algorytm zadziałał, część rzeczywista elementów na diagonali musi być większa od części urojonej tychże elementów. Dodatkowo macierz powinna być diagonalnie dominująca (wierszowo lub kolumnowo).

2. Im elementy z głównej diagonali są większe od elementów z pozostałych dwóch pasm, tym dokładność algorytmu jest lepsza. Można to zauważyć porównując układ 1, w którym elementy na diagonali były znacznie większe od pozostałych, oraz układ 5, gdzie elementy z diagonali były maksymalnie pięciokrotnie większe.

3. Wraz ze wzrostem ilość równań (a więc i niewiadomych) metoda Gaussa-Seidla w tył staje się szybsza w porównaniu ze zwykłą funkcją *linsolve* (przykład: układ 7 oraz 8), o ile tylko metoda jest zbieżna.

4. Metoda Gaussa-Seidla w tył działa znacznie dłużej od *linsolve* dla dużych macierzy, jeżeli metoda nie jest zbieżna (przykład: układ 9).

8 Skrypt testujący

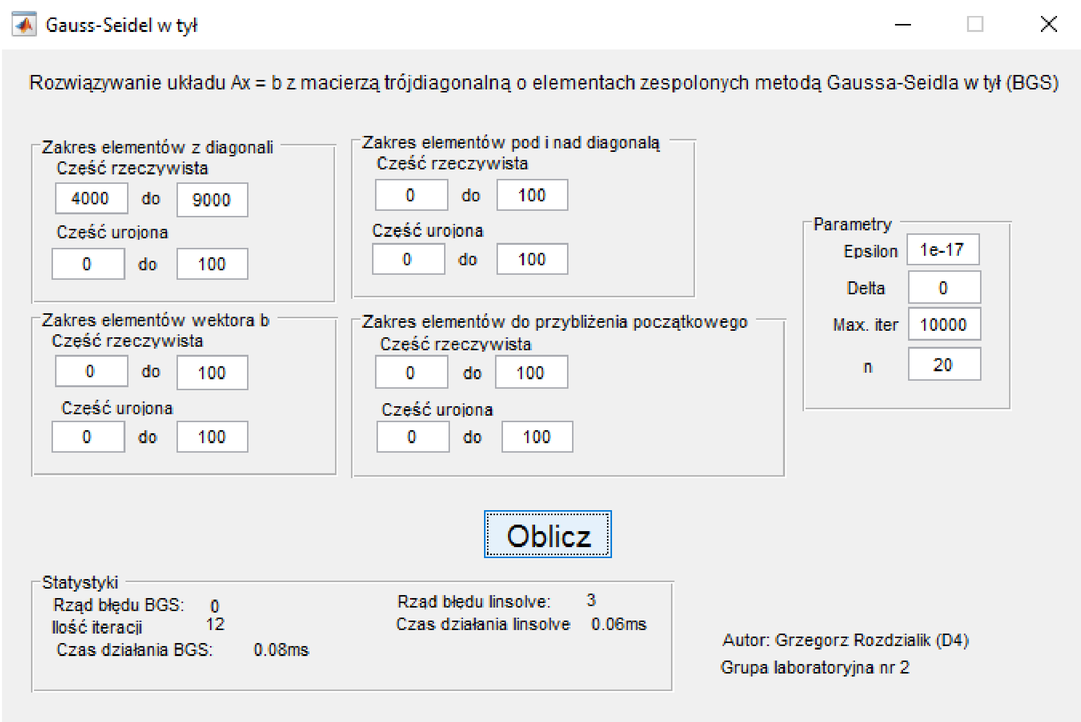
Do metody został dołączony skrypt testujący, pozwalający na wygenerowanie losowych wartości elementów i sprawdzenie poprawności działania metody. Możliwe parametry to ustawienie oddzielnie zakresu części rzeczywistej i części urojonej dla:

- rozmiaru układu równań n
- elementów na diagonali macierzy A
- elementów poza diagonalą macierzy A
- elementów wektora b
- elementów wektora przybliżenia początkowego $x^{(0)}$
- parametrów ε, δ
- maksymalnej liczby iteracji

Skrypt pokazuje rząd błędu (porównując najpierw z rozwiązaniem obliczonym przez *linsolve*, następnie jako $\|Ax-b\|$), ilość iteracji oraz czas potrzebny na obliczenia.

9 Interfejs graficzny

Do metody został dołączony interfejs graficzny. Znajduje się on w pliku *bgsGUI.fig*. Pozwala na zmianę zakresu losowanych elementów z podziałem na elementy na diagonali macierzy A, poza diagonalą (ale nadal w macierzy A), wektora b oraz wektora przybliżenia początkowego $x^{(0)}$. Dodatkowo możliwe jest zmienienie rozmiaru n układu równań oraz dostosowanie parametrów stopu ε, δ oraz maksymalnej liczby iteracji.



Interfejs graficzny jest graficzną wersją skryptu testującego *testScript.m* i pokazuje następujące wyniki:

- rząd błędu rozwiązania obliczonego za pomocą *linsolve* dostępnego w Matlabie oraz czas działania tych obliczeń (w milisekundach)
- rząd błędu liczony względem wektora b ($\|Ax - b\|$)
- ilość iteracji
- czas działania metody Gaussa-Seidla w tył (w milisekundach)

10 Bibliografia

- Metoda Gaussa-Seidla - Wikipedia, wolna encyklopedia
- P. Tatjewski *Metody numeryczne*, Warszawa: Oficyna Wydawnicza PW, 2013