NUM 3

Agnieszka Rejek

1 Wstęp

Dla $A \in \mathbb{R}^{N \times N}$, $b \in \mathbb{R}^N$ $(N = 50)$ zadano równanie $Ay = b$. Wektor $b \equiv (5, \ldots, 5)^T$. Macierz jest zbudowana następująco:

$$\mathbf{A} = \begin{pmatrix} 10 & 8 & 1 & 1 & \ldots & 1 & 1 & 1 & 1 \\ 1 & 10 & 8 & 1 & \ldots & 1 & 1 & 1 & 1 \\ 1 & 1 & 10 & 8 & \ldots & 1 & 1 & 1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & 1 & 1 & \ldots & 1 & 10 & 8 & 1 \\ 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 10 & 8 \\ 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & 10 \end{pmatrix}$$

Zgodnie z poleceniem, został zaimplementowany odpowiedni algorytm obliczający rozwiązanie. Uwzględniono szczególną budowę macierzy A (opis w 2). Program został napisany w języku Python, a do sprawdzenia rozwiązania skorzystano z bibliotek numerycznych NumPy oraz SciPy.

2 Metoda

2.1

Jeżeli jesteśmy w stanie przedstawić znaną nam macierz (tutaj A) w formie $A' + uv^T$, gdzie u i v to odpowiednie wektory, możemy skorzystać ze wzoru Shermana-Morrisona:

$$(A' + uv^T)^{-1} = A'^{-1} - \frac{A'^{-1}uv^TA'^{-1}}{1 + v^TA^{-1}u} \tag{1}$$

Macierz A jest macierzą gęstą, jednak łatwo zauważyć, że gdyby zastąpić jedynki zerami, mielibyśmy do czynienia z macierzą wstęgową o szerokości pasma 2. Jeżeli zdefiniujemy wektory u i v $(N = 50)$, których wszystkie współczynniki są równe 1 $(u = v)$, ich iloczyn utworzy macierz, również w całości wypełnioną jedynkami. Wtedy aby uzyskać $A = A' + uv^T$ definiujemy macierz A' jako macierz wstęgową z odpowiednio wartościami 9 na diagonali i 7 na pozycji nad diagonalą.

2.2 Wzory

$$Ay = b \tag{2}$$

$$y = A^{-1}b \tag{3}$$

Korzystając ze wzoru Shermana-Morrisona (1):

$$y = (A' + uv^T)^{-1}b = A'^{-1}b - \frac{A'^{-1}uv^T A'^{-1}b}{1 + v^T A'^{-1}u} \tag{4}$$

Przyjmijmy $z = A'^{-1}b$ i $z' = A'^{-1}u$. Wtedy, ze wzoru (4) przechodzimy do:

$$y = z - \frac{z'(v^T z)}{1 + v^T z'} \tag{5}$$

Problem upraszcza się do rozwiązania dwóch równań:

$$z = A'^{-1}b \tag{6}$$

$$z' = A'^{-1}u \tag{7}$$

Macierz A' jest macierzą trójkątną górną, dlatego nie musimy przeprowadzać jej faktoryzacji LU, (rozkład składałby się z macierzy L z jedynkami na diagonali, i macierzy U równej macierzy A'). Równania (6) i (7) obliczamy przy użyciu metody backsubstitution. Następnie znając wektory z i z' obliczamy rozwiązanie równania (5).

2.3 Złożoność obliczeniowa

Gdyby rozwiązywać równanie (3) z macierzą gęstą, przy użyciu faktoryzacji LU, złożoność obliczeniowa wynosiłaby $O(N^3)$. Dzięki zastosowaniu wzoru Shermana-Morrisona i rozwiązywaniu równania z macierzą trójkątną górną A' złożoność wynosi $O(N)$. Oznacza to znaczne zwiększenie wydajności obliczeń.

3 Opis programu

W programie kolejno: są rozwiązywane, przy użyciu backsubstitution, równania (6) i (7), następnie obliczony wynik równania (5). Otrzymujemy wektor `y`. Następnie przy użyciu bibliotek NumPy (operacje na macierzach i wektorach) i SciPy (zaimplementowanie macierzy rzadkiej A') jest wyliczany wektor `y_for_check`, wprost z równania (3). Na końcu zostaje obliczony błąd, czyli różnica wektorów `y` i `y_for_check`.

Program po uruchomieniu najpierw drukuje wyniki z NumPy (drukuje też macierz A, żeby pokazać, że jest ona poprawnie zaimplementowana), następnie otrzymany przeze mnie wektor y, a na samym końcu różnicę pomiędzy wektorami.

4 Wyniki

Uzyskany przeze mnie wektor, różnił się od wektora wyliczonego przez NumPy na poszczególnych współczynnikach na 16 lub 17 miejscu po przecinku. Można to wytłumaczyć błędem precyzji. Wyniki wydają się być zadowalające.