1. Wstęp

Przy różnych macierzach musimy zawszę sprawdzać jej typ. Może to nam ułatwić samo obliczenie, czas pracy algorytmu i wykorzystywaną pamięć. Jeśli mamy taką nierówność:

$$y = (B + uv^t)^{-1}b \; gdzie \; B + uv^t = A \text{ i znany wektor b}$$

w której złożoność obliczania faktoryzacji B mniejsza od złożoność obliczania faktoryzacji A, to najlepiej użyć wzór Shermana-Morrisona, który wygląda tak:

$$A_1^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

po podstawianiu, otrzymujemy taki wzór:

$$y = B^{-1}b - \frac{B^{-1}uv^TB^{-1}b}{1 + v^TB^{-1}u}$$

a żeby nie obliczać odwrotności B, możemy policzyć p i q:

$$z = B^{-1}b$$

$$q = B^{-1}u$$

w końcu otrzymuje wzór na y:

$$y = z - \frac{qv^Tz}{1 + v^Tq}$$

2. Zadanie

Trzeba rozwiązać równanie Ay = b dla N = 50:

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

$$oraz \; b = (5, \quad \cdots, \quad 5)^T$$

Trzeba wykorzystywać strukturę macierzy. Zapiszę ją jak $A = B + uv^T$, otrzymam taką macierz:

$$B = \begin{pmatrix} 9 & 7 & & & \\ & 9 & 7 & & \\ & & \ddots & \ddots & \\ & & & 9 & 7 \\ & & & & 9 \end{pmatrix}$$ i wektory $u = v = (1, \quad \cdots, \quad 1)^T$

Muszę policzyć z, q żeby otrzymać y. Złożoność obliczenia backward substitution wynosi O(n), tą metode będę używać dla z i q. Inne operacji będą w takiej samej złożoności.

3. Wyniki

$$y = \begin{pmatrix} 0.07525844089350037 \\ 0.07525904117533852 \\ 0.07525826938440369 \\ 0.07525926168703423 \\ 0.07525798586936636 \\ 0.07525962620636797 \\ \vdots \\ 0.13379325069654147 \end{pmatrix}$$

4. Podsumowanie

Jak i w zadaniu 3, trzeba uważać na strukturę macierzy i wiedzieć różne metody. W tym zadaniu, można zauważyć, że tą macierz można ”rozbić” na 2 wektory i wykorzystać algorytm Shermana-Morrisona. Zamiast zwykłej złożoności $O(n^3)$, zmniejszyli ją do $O(n)$ i zmniejszyli wykorzystywanie pamięci, bo nie chronimy niepotrzebnych zer.