Metody Numeryczne – Zadanie NUM4

Bartosz Bochniak

Wstęp

Polecenie

Zadana jest macierz:

$$A = \begin{pmatrix} 5 & 3 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 1 & 5 & 3 & 1 & \cdots & 1 & 1 & 1 & 1 \\ 1 & 1 & 5 & 3 & \cdots & 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \cdots & 1 & 5 & 3 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 5 & 3 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 5 \end{pmatrix}$$

oraz wektor $\boldsymbol{b} \equiv (2, \dots, 2)^T$. Macierz **A** ma liczby 5 na diagonali, 3 na pierwszej pozycji nad diagonalą, a pozostałe elementy są równe 1. Wymiar macierzy ustalamy na N = 120.

- Rozwiąż numerycznie równanie **Ay = b**, stosując odpowiednią metodę. Uwaga: algorytm należy go zaimplementować samodzielnie.
- Sprawdź swój wynik przy użyciu procedur bibliotecznych lub pakietów algebry komputerowe.
- Potraktuj N jako zmienną i zmierz czas potrzebny do uzyskania rozwiązania w funkcji N . Wynik przedstaw na wykresie. Czy wykres jest zgodny z oczekiwaniami?

Rozwiązanie

Równanie **Ay** = **b** można zapisać w formie **y** = $\mathbf{A}^{-1}\mathbf{b}$. Dla macierzy pełnych obliczanie odwrotności macierzy jest bardzo czasochłonną operacją ($O(n^3)$), dlatego użyty zostanie wzór Shermana-Morrisona na mniej kosztowne obliczenie odwrotności A. Jest to wzór znacznie upraszczający liczenie odwrotności macierzy dla macierzy o formie:

$$A = A' + uv^T$$

Przy założeniu, że wiemy jaka jest odwrotność macierzy A', oraz u i v to są znane wektory. Zatem, wzór Shermana-Morrisona przyjmuje postać:

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

Wzór ten pozwala obliczyć odwrotność macierzy **A** w czasie liniowym, dla rozłożenia **A**, gdzie:

$$A = A' + uu^T, \;\; dla: \qquad A' = \begin{pmatrix} 4 & 2 & & & \\ & 4 & 2 & & \\ & & \ddots & \ddots & \\ & & & 4 & 2 \\ & & & & 4 \end{pmatrix}, \qquad u = (1, 1, \dots, 1)^T$$

Podstawiając powyższy wzór do równania **y** = **A**$^{-1}$**b**, dostajemy:

$$y = (A' + uu^T)^{-1}b \rightarrow y = A'^{-1}b - \frac{A'^{-1}u * u^T A'^{-1}b}{1 + u^T A'^{-1}u}$$

Definiujemy pomocne wektory b_s, oraz u_s:

$$b_s = A'^{-1}b \qquad u_s = A'^{-1}u \quad \rightarrow \quad A'b_s = b \qquad A'u_s = u$$

Oba powyższe wektory można policzyć także w czasie O(n) za pomocą metody backward substitution, gdyż macierz A' jest macierzą trójkątną górną. Po wstawieniu danych wektorów otrzymany wzór to:

$$y = b_s - \frac{u_s * u^T b_s}{1 + u^T u_s}$$

Można dalej uprościć ten wzór poprzez zapisanie wyrazów $u^T b_s \;\; oraz \;\; u^T u_s$ jako skalarów, które da się obliczyć (jako suma iloczynów i-tych elementów tych wektorów dla $i \in \{1, \dots, N\}$) w złożoności O(n). Zatem podstawiając:

$$\alpha = u^T u_s \quad \beta = u^T b_s \quad \mu = \frac{\beta}{1 + \alpha}$$

Ostatecznie dostajemy wzór:

$$y = b_s - \mu * u_s$$

Całkowita złożoność wszystkich powyższych operacji składa się do złożoności O(n).

Implementacja

Dla wydajności pamięciowej macierz A definiowana jest jako dwie diagonale macierzy A', oraz wektor u. Po zaimplementowaniu powyższego algorytmu, wynik równania zostaje obliczone także za pomocą funkcji biblioteki `Eigen`, po czym oba wyniki są porównywane. Do tego, za pomocą aplikacji `gnuplot`, mierzymy czas działania naszego algorytmu w zależności od rozmiaru macierzy N, po czym kreowany jest wykres czasu od rozmiaru, by potwierdzić złożoność całego programu jako O(n).

Wynik

Dla N = 120:

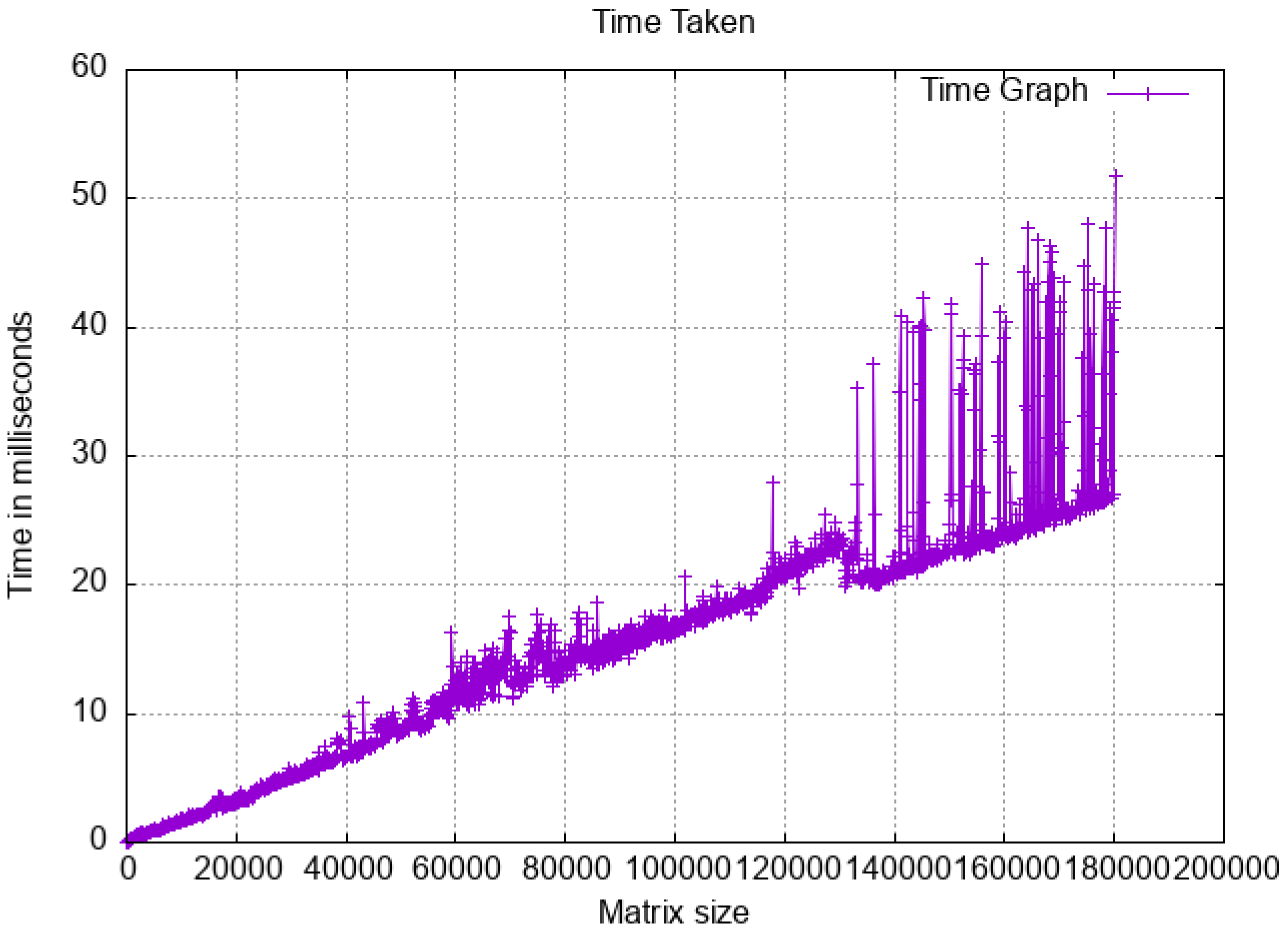
```
Wynik za pomoca wzoru Shermanna-Morrisona:

(0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158312, 0.0158311, 0.0158313, 0.0158309,
0.0158316, 0.0158302, 0.0158331, 0.0158273, 0.0158389,
0.0158157, 0.0158621, 0.0157693, 0.0159548, 0.0155838,
0.0163259, 0.0148417, 0.01781, 0.0118734, 0.0237467)
```

```
Wynik za pomoca biblioteki Eigen:

(0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158311, 0.0158311, 0.0158311, 0.0158311,
0.0158311, 0.0158312, 0.0158311, 0.0158313, 0.0158309,
0.0158316, 0.0158302, 0.0158331, 0.0158273, 0.0158389,
0.0158157, 0.0158621, 0.0157693, 0.0159548, 0.0155838,
0.0163259, 0.0148417, 0.01781, 0.0118734, 0.0237467)
```

Wykres czasu wykonania algorytmu od rozmiaru macierzy:



Powyższy wykres przyjmuje formę przybliżoną do funkcji liniowej, co potwierdza złożoność obliczeniową O(n).