Obliczenia Naukowe

Lista V

Kacper Pieniążek, 236606

Wtorek TP 15:15

29 stycznia 2019

Spis treści

1 Opis problemu

Rozwiązania zadań uwzględniają następujące założenia na temat macierzy:

$$\mathbf{A} = \begin{pmatrix} A_1 & C_1 & 0 & 0 & 0 & \cdots & 0 \\ B_2 & A_2 & C_2 & 0 & 0 & \cdots & 0 \\ 0 & B_3 & A_3 & C_3 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & B_{v21} & A_{v-2} & C_{v-2} & 0 \\ 0 & \cdots & 0 & 0 & B_{v-1} & A_{v-1} & C_{v-1} \\ 0 & \cdots & 0 & 0 & 0 & B_v & A_v \end{pmatrix}$$

Macierz $A \in R^{n \times n}$ jest rzadką macierzą blokową, gdzie $v = \frac{n}{l}$ (dla $n \geq 4$ podzielnych przez $l \geq 2$). l jest rozmiarem wszystkich kwadratowych macierzy wewnętrznych A_i, B_i, C_i, 0.

- $A_i \in R^{l \times l}$, $i = 1, \cdots, v$ - kwadratowe macierze gęste
- $0 \in R^{l \times l}$ - kwadratowa macierz zerowa
- $B_i \in R^{l \times l}$, $i = 2, \cdots, v$ - kwadratowe macierze z niezerowymi dwoma ostatnimi kolumnam

$$B_i = \begin{pmatrix} 0 & \cdots & 0 & b^i_{1l-1} & b^i_{1l} \\ 0 & \cdots & 0 & b^i_{2l-1} & b^i_{2l} \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & b^i_{ll-1} & b^i_{ll} \end{pmatrix}$$

- item $C_i \in R^{l \times l}$, $i = 1, \cdots, v-1$ - kwadratowe macierze diagonalne

$$C_i = \begin{pmatrix} c^i_1 & 0 & 0 & \cdots & 0 \\ 0 & c^i_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & c^i_{l-1} & 0 \\ 0 & \cdots & 0 & 0 & c^i_l \end{pmatrix}$$

Macierz **A** jest macierzą rzadką, czyli posiada znaczną ilość elementów zerowych. Dokładniej, macierz posiada sadsadsadsadsadsada elementów niezerowych. Z tego względu, przechowywanie macierzy **A** odbywa się w strukturze **SparseMatrixCSC** z bilbioteki języka *Julia*. Zakładamy, że *CSC*, czyli format Compressed Sparse Column zapewnia dostęp do elementów macierzy w czasie stałym. Standardowe algorytmy wykorzystane w kolejnych zadaniach zostały odpowiednio zmodyfikowane, aby zoptymalizować działanie programów przy specyficznej postaci macierzy **A**.

2 Zadanie 1

2.1 Cel zadania

Celem zadania jest implementacja funkcji rozwiązującej układ $\mathbf{A}x = b$ metodą eliminacji Gaussa. W rozwiązaniu należy uwzględnić specyficzną postać macierzy **A**. Rozwiązanie uwzględnia dwa warianty algorytmu: bez wyboru elementu głównego oraz z częściowym wyborem elementu głównego.

2.2 Rozwiązanie

2.2.1 Metoda eliminacji Gaussa bez wyboru elementu głównego

W celu rozwiązania układu równań, metoda eliminacji Gaussa stopniowo eliminuje elementy kolejnych wierszy w taki sposób, aby zamiast $Ax = b$ doprowadzić do równania $Ux = b'$, gdzie U jest tzw. macierzą trójkątną górną. W tym celu wykorzystuje się operacje elementarne na macierzach. W kolejnych krokach i algorytmu, a_{ii} zeruje elementy a_{ki} $\forall k \in \{1, \cdots, m\}$, m - ilość rzędów macierzy. W ten sposób wszystkie elementy pod diagonalą macierzy są zerowe.

Warto zauważyć, że ten wariant narzuca warunek na elementy a_{ii} macierzy, tj. musza być to elementy niezerowe. Modyfikacja algorytmu dla pozostałych przypadków będzie omówiona w następnym punkcie.

Otrzymana macierz górna pozwala na łatwe rozwiązanie równania przez *podstawianie wstecz.* Ten krok może zostać zapisany następująco:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}}$$

.

Omówiona postać metody Gaussa może jednak zostać zoptymalizowana na potrzeby zadania, dzięki założeniom na temat danych macierzy.

Macierz **A** jest rzadką macierzą blokową, zatem wiele bloków posiada elementy zerowe, które nie wpływają na wynik algorytmu. Ilość operacji można znacznie zredukować. Ponieważ metoda tworzy macierz trójkątną górną, wiele elementów pod diagonalą nie wymaga wykonania żadnej pracy, a dzięki strukturze blokowej macierzy można zauważyć pewne zależności.

Zauważmy, że macierz A_1 jako jedyna ma dla pierwszych $l - 2$ kolumn wartości niezerowe, B_1 jest niezerowa tylko dla kolumn $l-1$ oraz l. Jeśli spojrzeć na następne l kolumn, pierwsze 2 kolumny mają element niezerowy najniżej w B_2, pozostałe $l - 2$ w A_2. Ten schemat powtarza się to samego końca macierzy, co pozwala nam na wyznaczenie najniżej położonego, niezerowego elementu w kolumnie k. Otrzymujemy wzór

$$last_k^{col} = min\{k + l - k\%l, n\}$$

Dodatkowo, aby pozbyć się elementów macierzy trójkątnej dolnej, można zauważyć zależnonści patrząc na wiersze macierzy A. W każdym wierszu po diagonali bloków C_i występują same zera, a przed tymi blokami znajdują się A_i, których elementy przekątnej tworzą również diagonalę macierzy A. Te elementy oddalone są o l od C_{ii}, a każdy element przesuwa się o 1 wraz z wierszami. Stąd, można wyprowadzić wzór na najdalszy element niezerowy macierzy dolnej dla rzędu k:

$$last_k^{row} = min\{k + l, n\}$$

Powyższe rozważanie pozwala na zmniejszenie ilości operacji algorytmu, ponieważ nie będzie zerować elementów, które w strukturze podanej macierzy A i tak już są zerowe.

Standardowa implementacja metody Gaussa ma złożoność $O(n^3)$, postać zmodyfikowana ma jedynie złożoność $O(n)$. Kroki w pętlach wewnętrznych wykonują się zależnie od l.

Podobnie, podstawianie wstecz jest złożoności $O(n^2)$, jednak znajomość ostatniej niezerowej wartości dla danego wiersza usprawnia metodę do $O(n)$.

2.2.2 Metoda eliminacji Gaussa z częściowym wyborem elementu głównego

W przypadku, kiedy w diagonali macierzy znajdują się elementy zerowe, metoda eliminacji Gaussa wymaga pewnych zmian do działania. Rozwiązaniem tego problemu jest częściowy wybór elementu głównego.

Częściowy wybór polega na zamianie wiersza o zerowym a_{ii} z wierszem, który posiada największy $|a_{ik}|\ \forall k$. Pozostała część algorytmu zachodzi bez zmian

Zamiana wierszy zaburza ograniczenia nałożone przy metodzie bez wyboru elementu głównego.

Podczas eliminowania współczynników z $l - 2$ pierwszych kolumn zerowe elementy mogą powstać co najwyżej do kolumny $2l$. Schemat ten powtarza się dla kolejnych wierszy, nakładamy więc nowe ograniczenia:

$$last_k^{col} = min\{k + 2 * l - k\%l, n\}$$

$$last_k^{row} = min\{k + 2 * l, n\}$$

2.3 Wyniki

n	x = A / b	wariant I	wariant II
16	4.996003610813204e-16	3.073929999430902e-15	3.5388358909926865e-16
10000	2.0872192862952942e-16	1.630018508005151e-14	2.9702906800821436e-16
50000	3.1316044147344984e-16	2.985915849811188e-14	4.1456863517183594e-16

n	x = A / b	wariant I	wariant II
16	0.487631 s	0.054179 s	0.177426 s
10000	0.573425 s	0.3004 s	0.389172 s
50000	0.573425 s	5.948235 s	6.266521 s

n	x = A / b	wariant I	wariant II
16	13.474 MiB	463.624 KiB	1.635 MiB
10000	83.949 MiB	16.087 MiB	26.802 MiB
50000	83.949 MiB	16.087 MiB	26.802 MiB

2.4 Wnioski

Zestawienie powyższych wyników pokazuje, że wariant I, tj. eliminacja Gaussa bez wyboru elementu głównego jest najmniej dokładna ze wszystkich testowanych sposobów. Eliminacja z częściowym wyborem utrzymuje dobrą dokładność, podobnie jak wbudowana funkcja języka Julia. Widać również, że wariant II zarówno wymaga więcej pamięci, jak i czasu do rozwiązania równania, niż podstawowy algorytm. Wzrost kosztów obliczeń jednak przyczynia się do zwrostu dokładności obliczeń oraz umożliwia obliczenia dla macierzy, dla których wariant I nie działa, na przykład dla macierzy z elementami zerowymi na diagonalnej.

Przy wykonaniu zadania bardzo przydatna okazała się obserwacja struktury macierzy oraz dostosowanie problemu do podanych danych; dzięki temu ilość operacji została zminimalizowana w stosunku do algorytmów dla macierzy gęstych.