Metody Numeryczne

Praca domowa 1.

Hubert Michalski hm438596

6 lutego 2024

Zadanie 1.1

Dla silnie diagonalnie dominującej macierzy $A \in R^{N\times N}$ w postaci *Hessenberga*, tzn. takiej, że $a_{ij} = 0$ dla $i > j + 1$:

$$\begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix}$$

sformułuj:

(a) algorytm wyznaczający kosztem $O(N^2)$ jej rozkład LU;

(b) algorytm wyznaczający dla zadanego $b \in R^N$ rozwiązanie x układu równań $Ax = b$ kosztem $O(N^2)$.

Rozwiązanie $a)$

Do rozkładu macierzy A można użyć zmodyfikowanego algorytmu $GEPP$. Warto jednak na początku zaznaczyć, że rozważana macierz A jest silnie diagonalnie dominująca tzn.

$$\forall_k |a_{k,k}| > \sum_{i\neq k} |a_{k,i}|$$

więc nie ma konieczności wykorzystywania zamiany wierszy. Gdyby jednak zadana macierz nie miała tej własności, to można by dodatkowo zaobserwować, że dla k-tego elementu diagonali mamy jedynie dwie możliwe wartości tzn. $a_{k,k}$ lub $a_{k+1,k}$ ponieważ wszystkie elementy $a_{i,k}$, dla $i > k + 1$ są zerami z definicji. Zatem gdyby $|a_{k+1,k}| > |a_{k,k}|$ to byśmy zmieniali wiersze k i $k + 1$. Teraz jednak ten krok może zostać pominięty.

W przypadku standardowego algorytmu $GEPP$ kolejnym krokiem byłoby wyznaczenie k-tej kolumny macierzy L, w tym celu podzielilibyśmy jej elementy pod diagonalą przez $a_{k,k}$. Jednak dla naszego specjalnego przypadku wystarczy jedynie zmodyfikować wartość $a_{k+1,k}$, ponieważ (ponownie z definicji) reszta elementów tej kolumny to zera.

Następnie należy zaktualizować pozostałą część macierzy, biorąc pod uwagę poprzednie modyfikacje. Z poprzedniego kroku wiadomo, że zmieniła się tylko wartość $a_{k+1,k}$, co oznacza, że wystarczy zaktualizować wyłącznie $k + 1$-szy wiersz macierzy A. Powyższe kroki powtarzamy dla $k = 1 : N - 1$.

Algorithm 1 LU decomposition for Hessenberg matrix

for k = 1 : $N - 1$ **do**
  $a_{k+1,k} \leftarrow a_{k+1,k}/a_{k,k}$
  for i = k + 1 : N **do**
    $a_{k+1,i} \leftarrow a_{k+1,i} - a_{k+1,k}a_{k,i}$

Szacowany koszt:

- liczba iteracji: $\mathcal{O}(N)$
- aktualizacja wiersza w k-tej iteracji: $\mathcal{O}(N)$

Ostatecznie otrzymujemy: $\mathcal{O}(N^2)$

Rozwiązanie $b)$

Aby wyznaczyć rozwiązanie zadanego układu równań skorzystajmy z wyżej wymienionego algorytmu do przedstawienia macierzy A jako iloczynu macierzy L oraz U. Dodatkowo wiemy, że równania z macierzami trójkątnymi można rozwiązać w czasie $\mathcal{O}(N^2)$. Wystarczy zatem dwukrotnie zastosować ten fakt do rozłożonej poprzednio macierzy i otrzymujemy rozwiązanie zadania:

Algorithm 2 Solve equation $Ax = b$ for Hessenberg matrix

1: $L, U \leftarrow \mathit{decompose}(A)$ // $\mathcal{O}(N^2)$ z poprzedniego zadania, dla uproszczenia zapisu jako dwie macierze lecz nie zmienia to złożoności rozwiązania
2: $y \leftarrow L^{-1}b$ // Rozwiąż $Ly = b$, czas $\mathcal{O}(N^2)$
3: $x \leftarrow U^{-1}y$ // Rozwiąż $Ux = y$, czas $\mathcal{O}(N^2)$

Każdy krok ma złożoność $\mathcal{O}(N^2)$ zatem złożoność całego algorytmu to $\mathcal{O}(N^2)$.