

Idea algorytmu:

Macierz poszerzoną $[A | I]$ przekształca się do postaci $[I | V]$, gdzie A jest kwadratową macierzą do odwrócenia, I – macierzą jednostkową o wymiarach takich jak A . V jest macierzą odwrotną do A , tj. $V = A^{-1}$.

Podstawowe operacje algorytmu G-J realizowane są za pomocą przekształceń elementarnych:

1. Normalizacji wierszy (aby uzyskać jedynki na głównej przekątnej),
2. Zerowania elementów poza przekątną główną (aby dokończyć tworzenie macierzy jednostkowej po lewej stronie).

Oznaczenia:

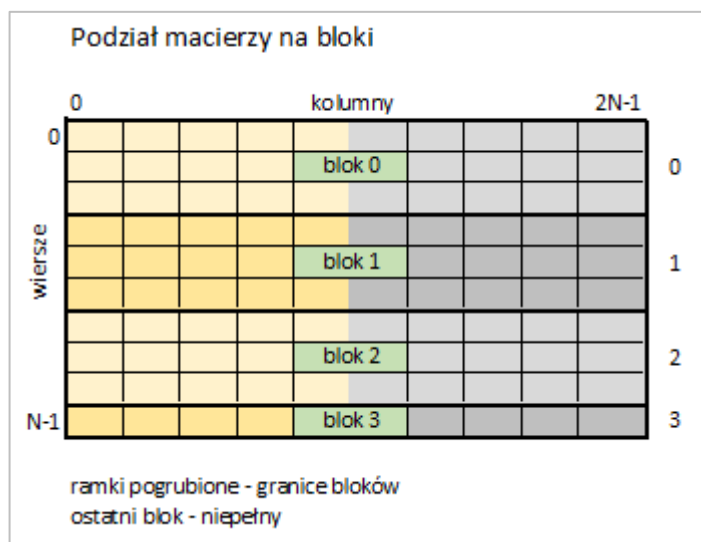
N - liczba wierszy odwracanej macierzy A

p - liczba procesów (MASTER=0 oraz procesy wykonawcze [dalej: *wykonawcy*]: W_1, \dots, W_{p-1} o identyfikatorach 1, 2, ..., $p-1$)

Organizacja pamięci i wymiana danych:

MASTER: posiada obszar na macierz rozszerzoną ($N \times 2N$).

Wirtualnie macierz podzielona jest na bloki po $b = N/p$ wierszy, każdy o $2N$ kolumnach oraz ewentualnie ostatni blok o liczbie wierszy mniejszej niż b .



Każdy z bloków będzie obsługiwany przez odrębny proces.

Wykonawcy W_x : posiadają bufor na dane dostarczanie przez MASTER i obszary robocze do wykonywania obliczeń.

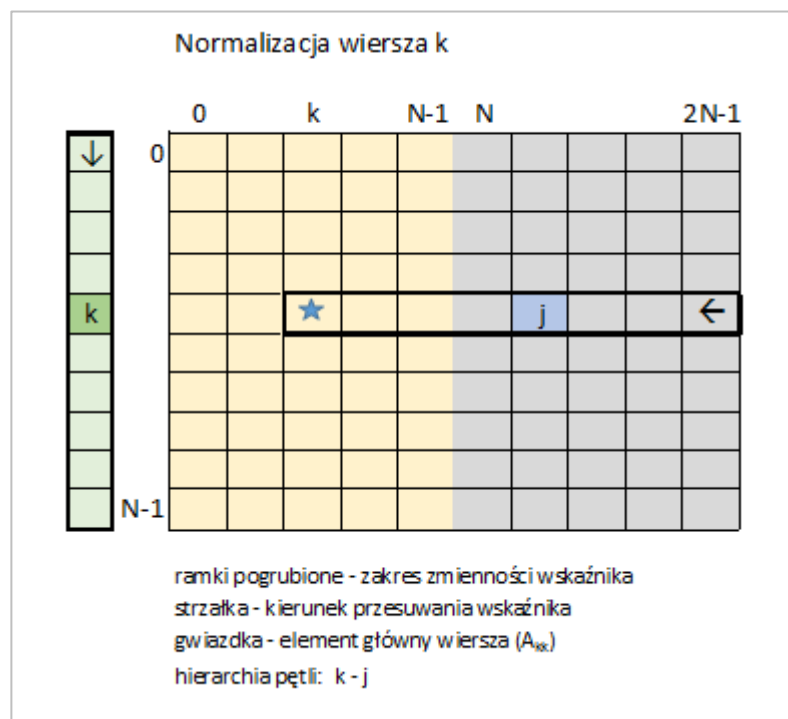
Każdy wykonawca otrzymuje pełny blok wierszy macierzy w celu wyzerowania elementów niediagonalnych. Wiersze należące do ostatniego, niepełnego bloku (jeśli istnieją) przetwarza MASTER.

Fragment kodu realizujący algorytm

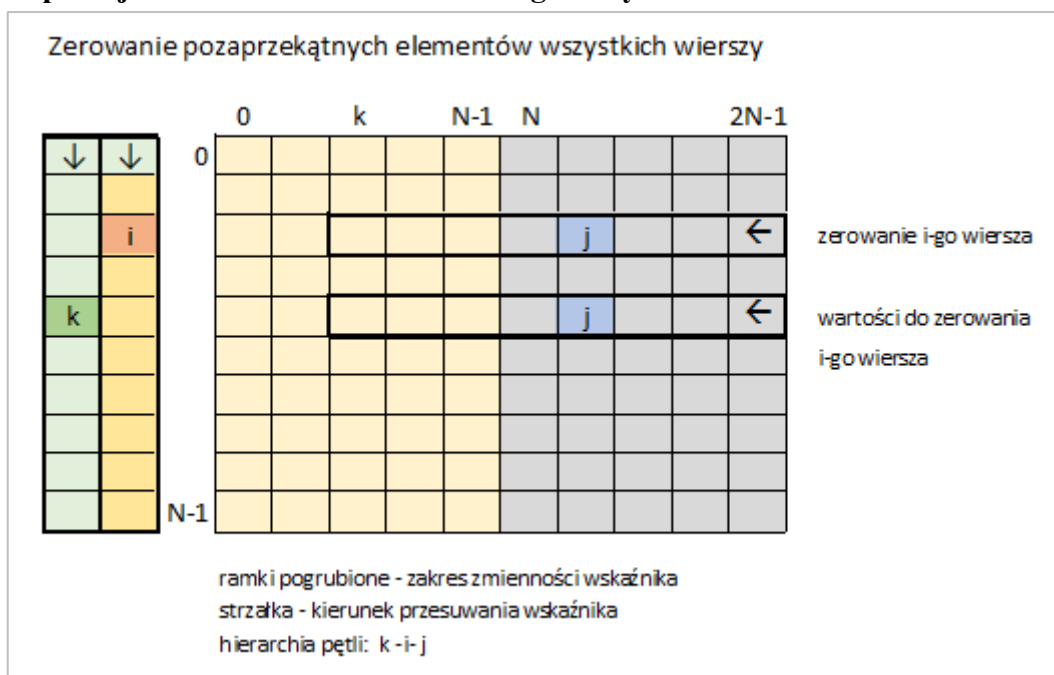
```
int i,j,k;

/* Transformacja [A | I] do postaci [I | V]
   N - liczba wierszy;
   M=2*N - liczba kolumn macierzy poszerzonej
*/
for (k=0; k<N; k++)          // k=nr wiersza
{
    // normalizacja wiersza
    for (j=M-1; j>=k; j--)    // j=nr kolumny
        A[k][j] = A[k][j] / A[k][k];
    // usuwanie zer poza przekatna
    for (i=0; i<N; i++)       // i=nr wiersza
        if (i != k)
            for (j=M-1; j>=k; j--)    // j=nr kolumny
                A[i][j] = A[i][j] - A[i][k]*A[k][j];
}
```

Schemat operacji normalizacji:



Schemat operacji zerowanie elementów niediagonalnych:



Aktywności:

Definicja: *prawa część wiersza* – elementy w kolumnach od k do $2N-1$ tego wiersza; pogrubiona ramka na rysunku)

MASTER:

1. MASTER: Wprowadza i kontroluje poprawność parametrów wywołania.
2. MASTER: Wczytuje liczbę N wierszy oraz współczynniki odwracanej macierzy i tworzy macierz poszerzoną o odpowiednią macierz jednostkową.
3. MASTER: Ustala liczbę p dostępnych procesów,
4. MASTER: Oblicza wielkość b bloku (liczbę wierszy na proces),
5. MASTER: Przesyła do wykonawców liczbę N wierszy macierzy poszerzonej,
6. MASTER: Do każdego wykonawcy przesyła nr początkowego i końcowego wiersza przypisanego mu bloku (odpowiednio: $nrp[w]$ i $nrk[w]$, gdzie w jest identyfikatorem wykonawcy). Wykonawcą zerowania w niepełnym bloku jest MASTER ($w=0$),
7. MASTER: Uruchamia pętlę główną ($k=0, \dots, N-1$),
8. Dla danego k :
 - a. MASTER: Wykonuje normalizację wiersza k ,
 - b. MASTER: Przesyła do wykonawców prawą część wiersza k ,
 - c. MASTER: Przesyła do wykonawców prawe części wierszy należących do przypisanych im bloków,
 - d. WYKONAWCY: na podstawie danych: $w, N, nrp[w], nrk[w]$ oraz prawych części: wiersza k oraz wierszy swojego bloku wykonują zerowanie elementów niediagonalnych,
 - e. WYKONAWCY: zwracają zmodyfikowane prawe części wierszy swojego bloku procesowi MASTER,
 - f. MASTER: umieszcza otrzymane zaktualizowane prawe części w globalnej macierzy poszerzonej,
 - g. MASTER: wykonuje zerowanie w niepełnym bloku (o ile jest konieczne) i aktualizuje odpowiednie prawe części wierszy w globalnej macierzy poszerzonej,
9. MASTER: Po zakończeniu realizacji pętli głównej czeka na niedostarczone komunikaty,
10. MASTER: Wyprowadza wyniki odwracania macierzy i kończy program.

Zamieszczony fragment kodu opracowano na podstawie

[1] http://we.pb.edu.pl/~jforenc/2006_2007_02/INF-2007-Wykl-07-Zaoczne-4na1.pdf