

Rozwiązanie układów równań liniowych algebraicznych z macierzą gęstą

- Faktoryzacja macierzy
- LU faktoryzacja macierzy symetrycznej
- LL^T faktoryzacja macierzy symetrycznej dodatnio określonej
- Blokowa LL^T faktoryzacja macierzy symetrycznej dodatnio określonej

Test Gauss

Rozmiar zagadnienia: 2 400 równań

Charakterystyki komputera:

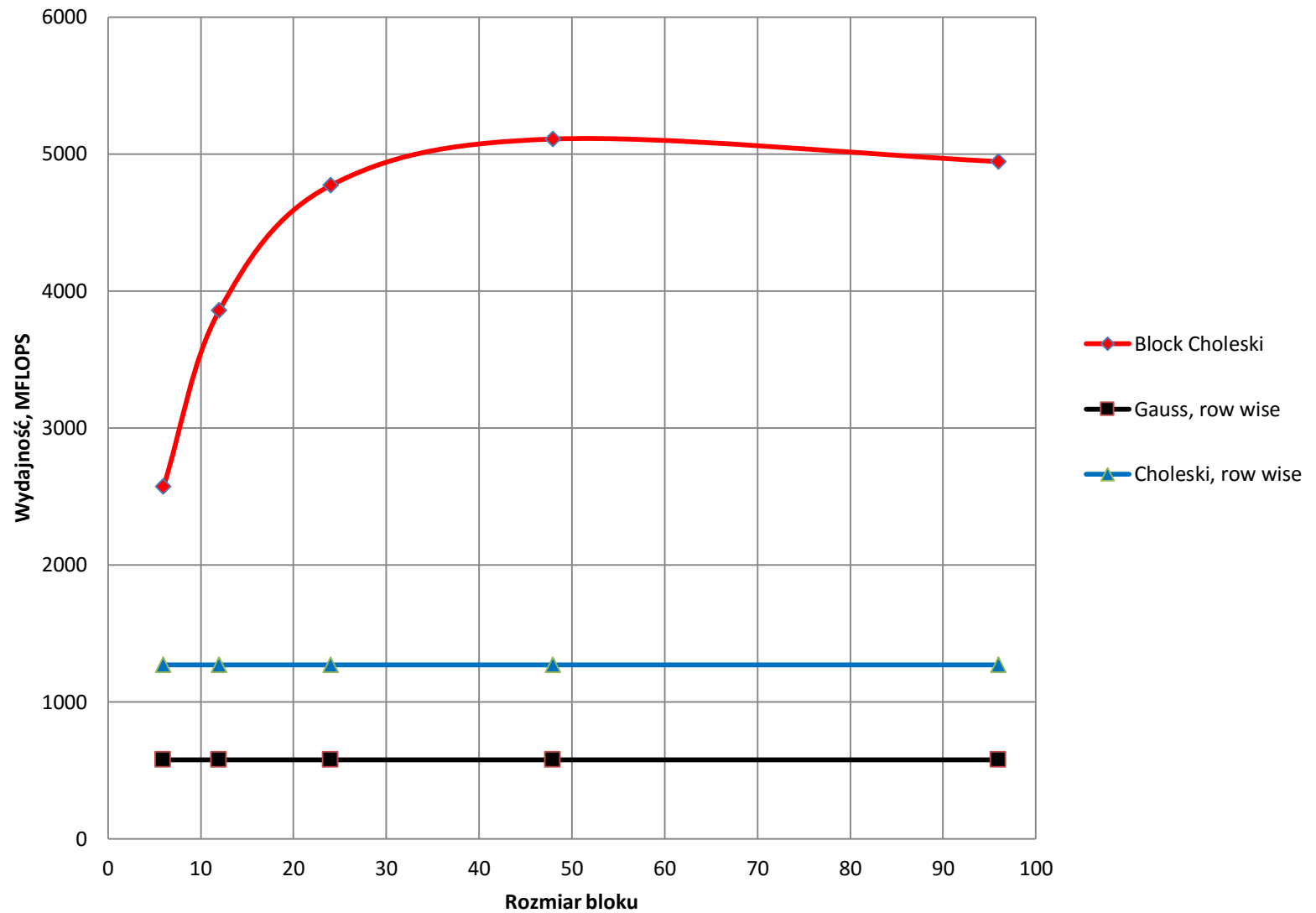
Procesor – Intel® Core™2 Quad CPU Q6600 @2.40 GHz

Cache - L1: 32 KB, L2:4096 KB

Pamięć: DDR2 800 MHz 4 GB

Metoda	FLOP	Czas, s	Wydajność, MFlops
Gauss (LU), wiersz po wiersze	$O(n^3/3)$	7.98	577
Gauss (LU), kolumna po kolumnie		8.71	529
Cholesky (LL^T), wiersz po wiersze		3.63	1 270
Cholesky (LL^T), kolumna po kolumnie		4.22	1 090
Block Cholesky (block 6)		1.79	2 573
Block Cholesky (block 12)		1.2	3 856
Block Cholesky (block 24)		0.97	4 770
Block Cholesky (block 48)		0.9	5 109
Block Cholesky (block 96)		0.93	4 944

Porównywanie wydajności dla różnych metod faktoryzacji



➤ **Pytanie: dla czego przy takiej samej ilości operacji zmiennoprzecinkowych różne metody demonstrują różną wydajność ?**

➤ Dla odpowiedzi na to pytanie rozważymy te algorytmy.

➤ Faktoryzacja macierzy – to jest rozkład macierzy na trójkątne mnożniki macierzowe. Jest zastosowana przy rozwiązywaniu układów równań liniowych algebraicznych metodami bezpośrednimi.

➤ Metody bezpośrednie – to są takie metody, które doprowadzają do rozwiązania dokładnego (w arytmetyce dokładnej) za skończoną ilość operacji.

➤ Cecha ta podstawowo różni ich od metod iteracyjnych, które doprowadzają do rozwiązania przybliżonego, przy czym ilość operacji dla metod iteracyjnych nie jest przewidywana.

Algorytm rozwiązywania układu równań liniowych algebraicznych

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

składa się z etapów:

1. Faktoryzacja macierzy: $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}; \quad \text{or} \quad \mathbf{A} = \mathbf{L} \cdot \mathbf{L}^T$
2. Podstawienie bezpośrednie: $\mathbf{L} \underbrace{\mathbf{Ux}}_{\mathbf{y}} = \mathbf{b} \rightarrow \mathbf{Ly} = \mathbf{b} \rightarrow \mathbf{y}$
3. Podstawienie wsteczne $\mathbf{Ux} = \mathbf{y} \rightarrow \mathbf{x}$

Złożoność obliczeniowa dla macierzy symetrycznej:

Faktoryzacja – $O(n^3/3)$ FLOP;

Podstawienia bezpośrednie i wsteczne - $O(n^2)$ FLOP

Podstawienie bezpośrednie:

$$\begin{pmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$y_1 = b_1 / \lambda_{11}$$

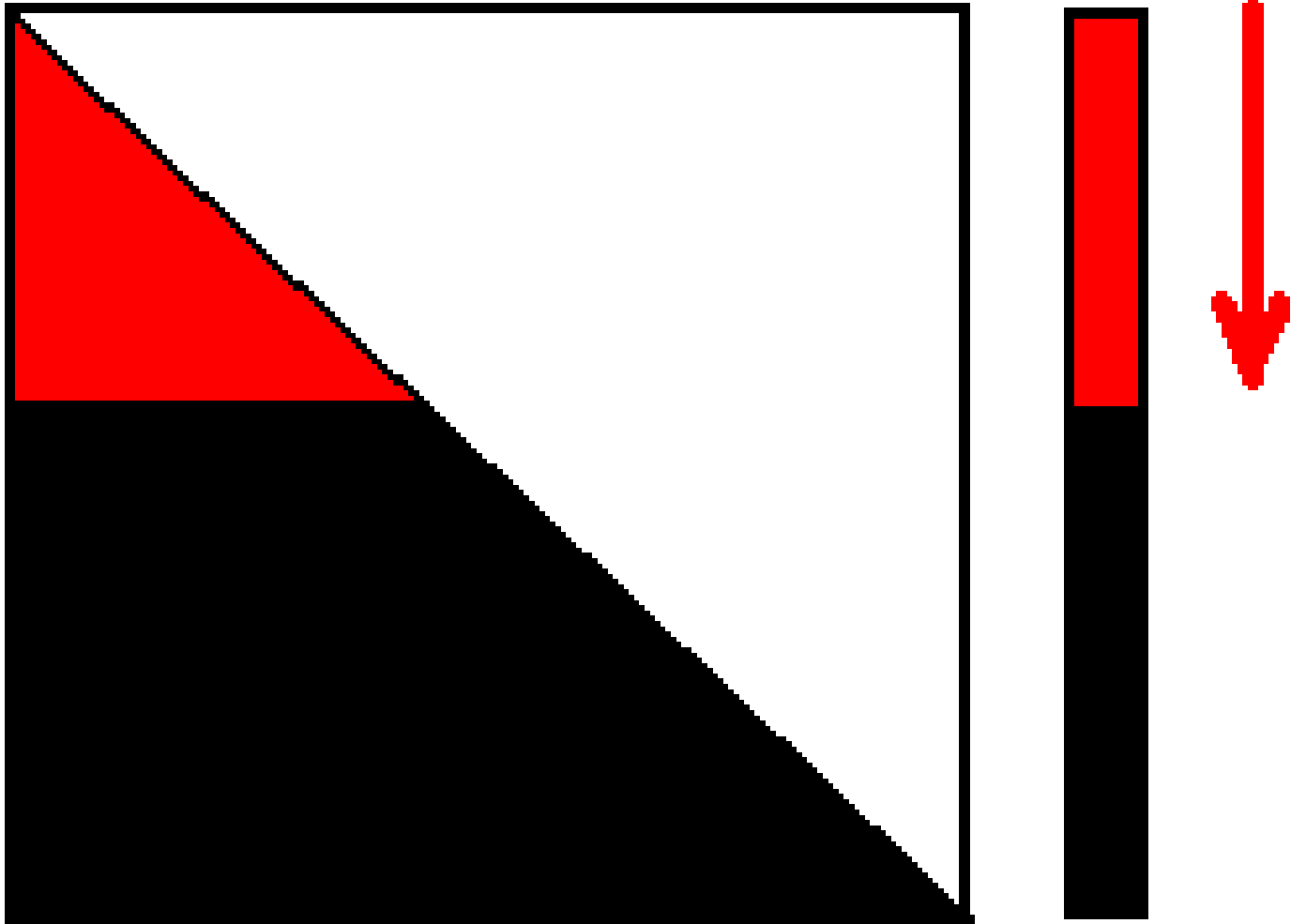
$$\lambda_{21}y_1 + \lambda_{22}y_2 = b_2 \Rightarrow y_2 = (b_2 - \lambda_{21}y_1) / \lambda_{22}$$

$$\lambda_{31}y_1 + \lambda_{32}y_2 + \lambda_{33}y_3 = b_3 \Rightarrow y_3 = (b_3 - \lambda_{31}y_1 - \lambda_{32}y_2) / \lambda_{33}$$

.....

$$y_k = \left(b_k - \sum_{s=1}^{k-1} \lambda_{k,s} y_s \right) / \lambda_{k,k}$$

Podstawienie bezpośrednie:



Podstawienie wsteczne:

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$u_{33}x_3 = y_3 \Rightarrow x_3$$

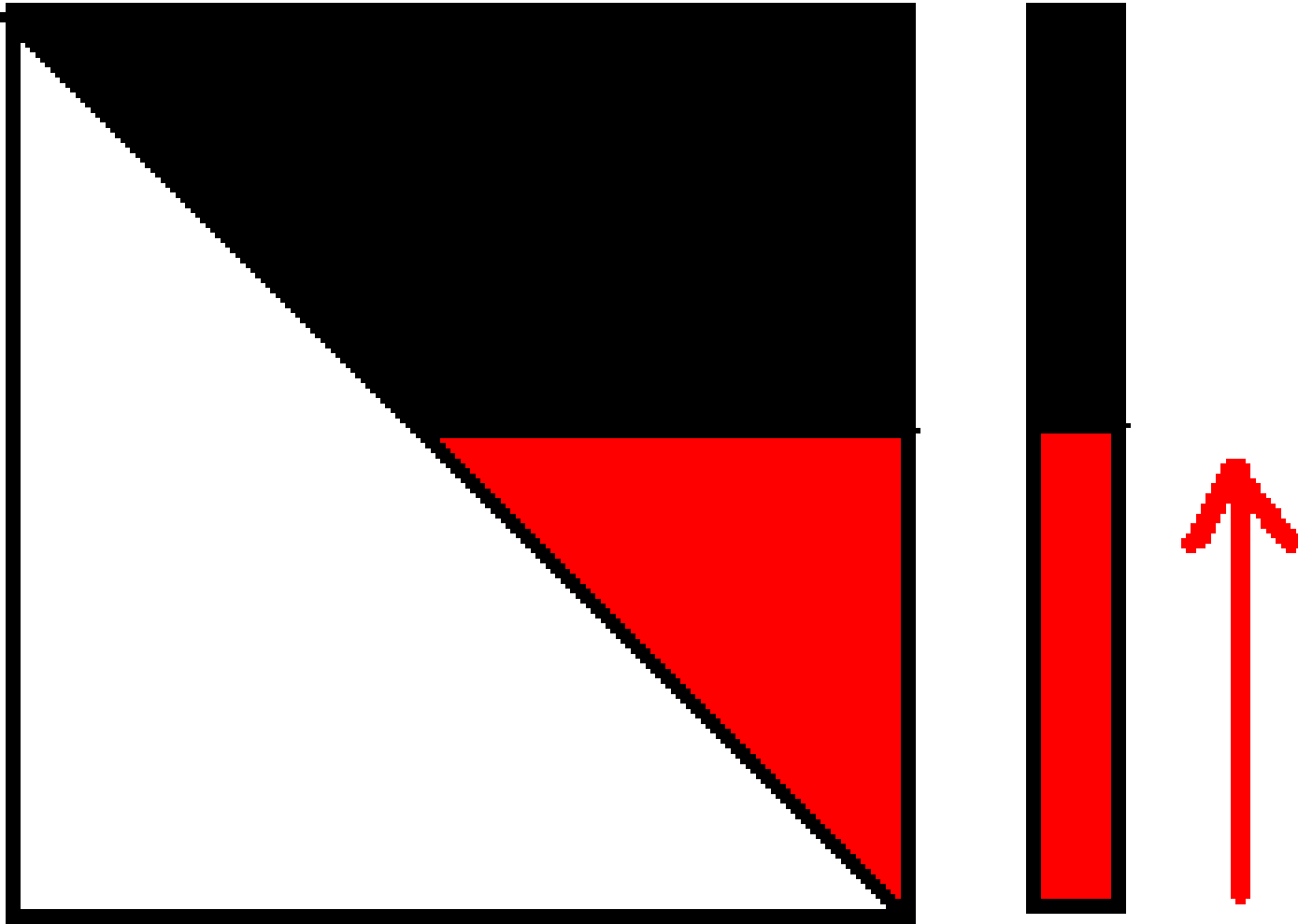
$$u_{22}x_2 + u_{23}x_3 = y_2 \Rightarrow x_2$$

$$u_{11}x_1 + u_{12}x_2 + u_{13}x_3 = y_1 \Rightarrow x_1$$

.....

$$x_k = \left(y_k - \sum_{s=k+1}^n u_{k,s} x_s \right) / u_{k,k}$$

Podstawienie wsteczne:



$k = 1, 2, \dots, n$

$piv1 = 1.0 / a_{k,k}$

$i = k + 1, \dots, n$

$piv = a_{i,k} \cdot piv1; \quad // \lambda_{i,k} = a_{i,k} / a_{kk}$

$j = i, \dots, n$

$a_{i,j} = a_{i,j} - \lambda_{i,k} \cdot a_{j,k}$

end j loop

end i loop

end k loop

Algorytm LL^T

FLOP = $O(n^3/3)$

$$l_{11} = \sqrt{a_{11}}$$

$i = 2, \dots, n$

$j = 1, 2, \dots, i-1$

$S = 0.0;$

$k = 1, 2, \dots, j-1$

$S = S + a_{i,k} \cdot a_{j,k}$

end k loop

$$l_{i,j} = (a_{i,j} - S) / a_{j,j}; \quad // l_{i,j} = \left(a_{i,j} - \sum_{k=1}^{j-1} a_{i,k} \cdot a_{j,k} \right) / a_{j,j}$$

end j loop

$S = 0.0;$

$k = 1, 2, \dots, i-1$

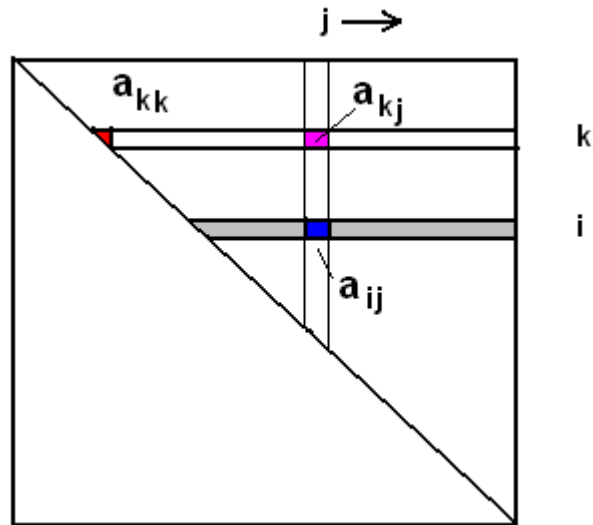
$S = S + a_{i,k} \cdot a_{i,k};$

end k loop

$$l_{i,i} = \sqrt{a_{i,i} - S}; \quad // l_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} a_{i,k}^2}$$

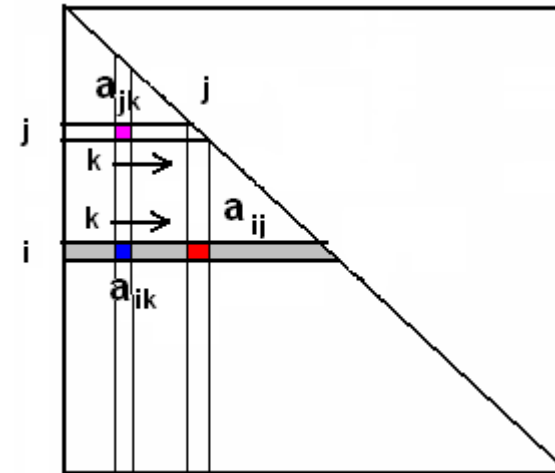
end i loop

LU



$$a_{ij} \leftarrow a_{ij} + \alpha \cdot a_{kj}$$

LL^T



$$s \leftarrow s + a_{ik} \cdot a_{jk}$$

Odczyt: 2 (a_{ij} – wolny, a_{kj} – szybki)

Zapis: 1 (a_{ij} – wolny)

Odczyt: 2 (a_{ik} – szybki, a_{jk} – wolny)

s - jest umieszczone w rejestrze

Algorytm Choleskiego (LL^T) w pętli wewnętrznej zawiera o 1 przesłanie mniej od algorytmu Gaussa (LU) i ma tylko jeden wolny odczyt! (Dla tego i działa szybszej).

Umieszczenie danych w pamięci

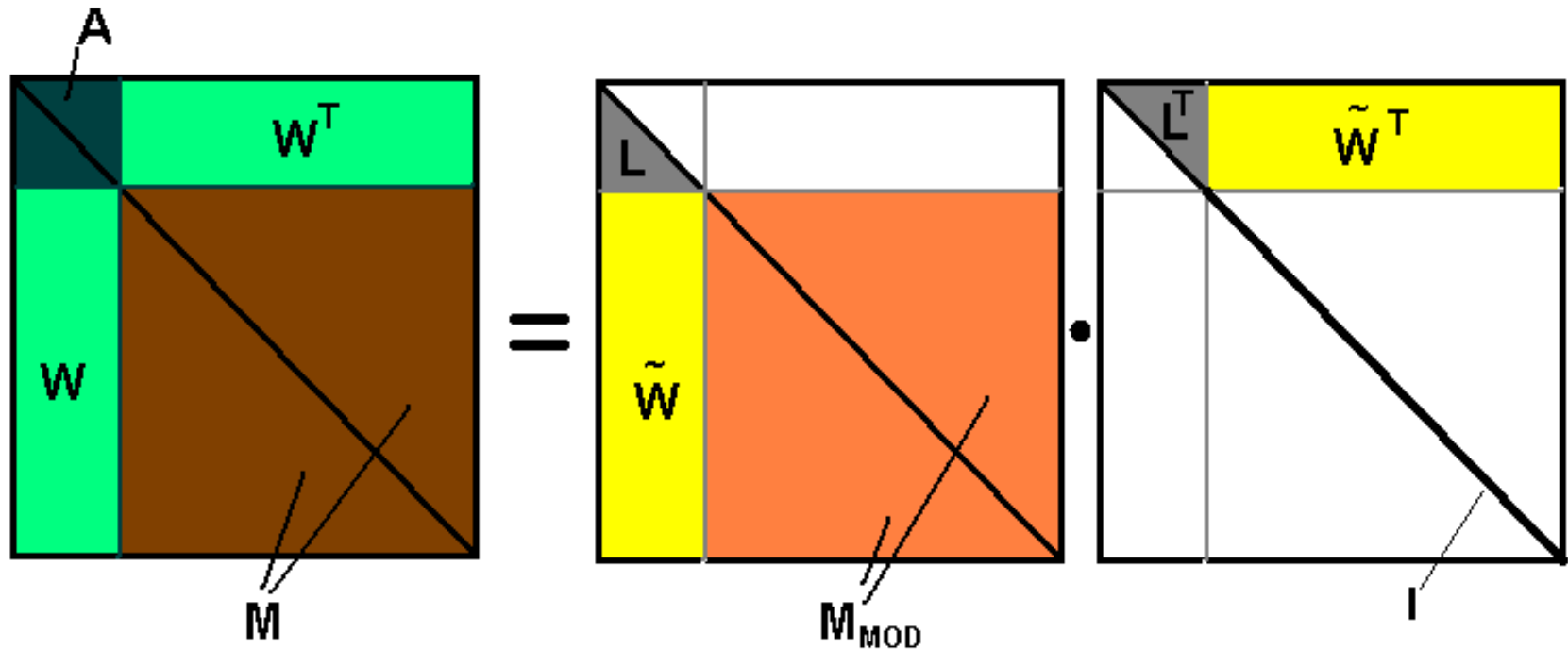
LU

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ & 6 & 7 & 8 & 9 \\ & & 10 & 11 & 12 \\ & & & 13 & 14 \\ & & & & 15 \end{pmatrix}$$

LL^T

$$\begin{pmatrix} 1 & & & & \\ 2 & 3 & & & \\ 4 & 5 & 6 & & \\ 7 & 8 & 9 & 10 & \\ 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

Metoda blokowa LL^T



1. $A = L \cdot L^T$
2. $W^T = L \cdot \tilde{W}^T + 0 \cdot I \Rightarrow L \cdot \tilde{W}^T = W^T \Rightarrow \tilde{W}^T$
3. $M = \tilde{W} \cdot \tilde{W}^T + M_{MOD} \cdot I \Rightarrow M_{MOD} = M - \tilde{W} \cdot \tilde{W}^T$

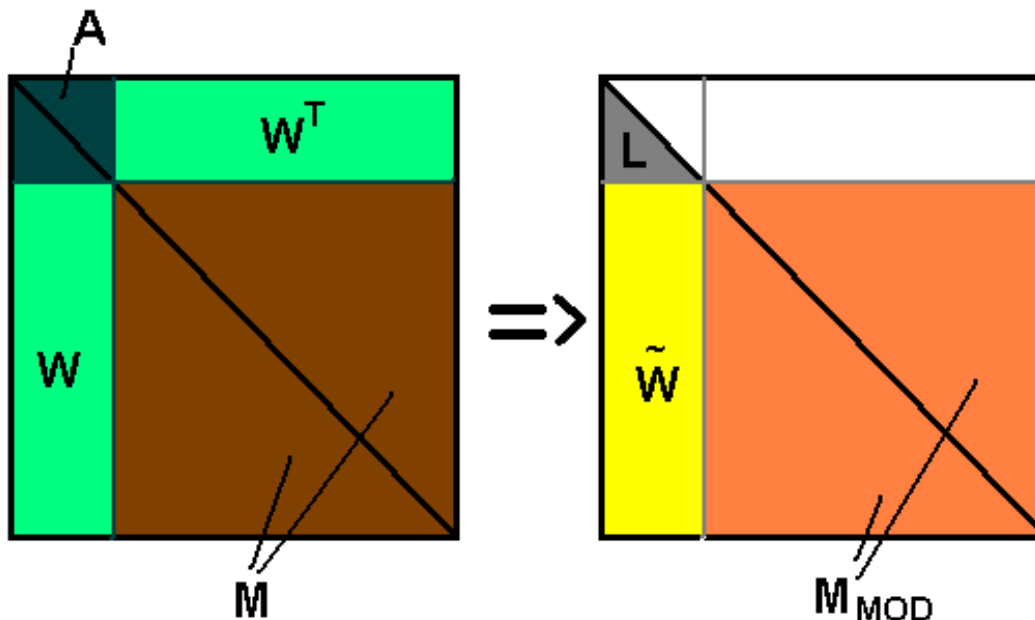
Dopełnienie Schura: $M_{MOD} = M - \tilde{W} \cdot \tilde{W}^T$

Metoda blokowa LL^T

FLOP $= O(n^3/3)$

- Podział na bloki nadaje możliwość na drugim i trzecim etapach utrzymać poziom BLAS-III – zastąpić procedury mnożenia macierzy przez wektor (BLAS-I – BLAS - II) procedurami mnożenia macierzy przez macierz.

Wynik kolejnego kroku:



Powtarzając tę procedurę dla podmacierzy M_{MOD} , dokonujemy faktoryzacji macierzy