

行列乗算におけるストラッセンの方法の拡張

日立製作所 汎用コンピュータ事業部
後 保範 (Yasunori Ushiro)

1. 概要

$C=A \times B$ の実密行列乗算はストラッセンの方法¹⁾²⁾を適用すると、演算量が定義式に比較して、約 $7/8$ に減少することが知られている。ストラッセンの方法は行列 A, B, C をそれぞれ 2×2 に分割して適用するものである。行列乗算に 2 回または 3 回ストラッセンの方法を適用すれば、各行列 A, B, C はいずれも 4×4 及び 8×8 分割に対応し、演算量はそれぞれ約 $(7/8)^2, (7/8)^3$ に減少する。

本報告では、行列乗算 $C=A \times B$ において、より一般的に行列 A, B, C をそれぞれ $n \times 2$, $2 \times n$ 及び $n \times n$ に分割した場合の方法を提案する。ここで、 n は 2 以上の整数で $n=2$ のときはストラッセンの方法に一致する。本方法を実密行列の乗算に 1 回適用すると n が大きい場合、演算量が約 $3/4$ に減少する。大次元の密行列を係数とする連立一次方程式及び固有値の計算では、縦長行列 A と横長行列 B の行列乗算が多く発生するため、提案方式は効果がある。本方式もストラッセンの方法と同様に何回も繰り返し適用可能である。更に、行列乗算 $C=A^T \times A$ の様に結果が対称行列やエルミート行列になる場合も、直接にストラッセンの方法及び本方法を適用できることが判明した。その結果についても報告する。

実密行列や複素密行列に対してブロックガウス変換やハウスホルダー変換をする場合は、ストラッセンの方法を活用して演算量を削減できる。提案方式を適用すれば更に演算量が減少する。本報告では、これら密行列に対するブロックガウス変換において具体的に演算量の比較を示す。実際に計算機に適用する場合には演算量以外の要素の評価が必要であり、今回は計算機への適用結果は報告しない。

2. ストラッセンの方法による演算量減少の仕組み

提案方式を説明する前に、ストラッセンの方法による演算量削減の仕組みを説明する。ストラッセンの方法は下記の様な 2×2 分割したブロック行列に適用する。

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \times \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

ここで、各ブロック行列は正方は長方形列とする。この時、ブロック行列 C_{ij} の一般的な計算は下記の様になり、8 回の行列乗算と 4 回の行列加算で計算できる。

$$C_{ij} = A_{i1} \times B_{1j} + A_{i2} \times B_{2j} \quad (i, j=1, 2)$$

ストラッセンは以下の様に行列 C の計算を 7 回の行列乗算と 18 回の行列加減算で計算できることを示した。

$$\begin{array}{ll}
M1 = (A_{21} - A_{22}) \times B_{11} & M2 = A_{22} \times (B_{11} + B_{21}) \\
M3 = A_{11} \times (B_{12} + B_{22}) & M4 = (A_{12} - A_{11}) \times B_{22} \\
M5 = (A_{11} + A_{22}) \times (B_{11} + B_{22}) & M6 = (A_{12} + A_{22}) \times (B_{21} - B_{22}) \\
M7 = (A_{11} + A_{21}) \times (B_{12} - B_{11}) & \\
C_{11} = M5 + M6 - M2 + M4 & C_{12} = M3 + M4 \\
C_{21} = M1 + M2 & C_{22} = M5 + M7 + M1 - M3
\end{array}
\left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{中間ワーク} \\ \\ \\ \end{array}$$

$$\left. \begin{array}{l} C_{11} = M5 + M6 - M2 + M4 \\ C_{12} = M3 + M4 \\ C_{21} = M1 + M2 \\ C_{22} = M5 + M7 + M1 - M3 \end{array} \right\} \text{結果セット}$$

小行列のサイズを $m \times m$ とすると、小行列の乗算は $2m^3 - m^2$ 、加減算は m^2 の演算量となる。そのため、元の行列乗算の演算量は一般的方法では $16m^3 - 4m^2$ なのに対し、ストラッセンの方法では $14m^3 + 11m^2$ となり、 m が大きい場合は約 7/8 に演算量を削減することができる。

ストラッセンの方法を分析すると、 C_{12} と C_{21} の計算には小行列の乗算がそれぞれ 2 回必要である。それに対し、 C_{11} と C_{22} の計算では、 C_{12} 及び C_{21} の計算のために作成した、中間ワークを活用すると、合計で 3 回の行列乗算で済む。ここで、行列乗算の回数を減少させている。この原理を活用すると、提案方式に拡張できる。

3. ストラッセンの方法から提案方法への拡張方法

ストラッセンの方法を図式化することを考える。結果の行列 C を

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

と、同一サイズの 4 行列にブロック化する。ストラッセンの方法では C_{12} と C_{21} を計算する中間ワーク行列を利用して、 C_{11} 及び C_{22} を 3 回の行列乗算で処理することであった。行列乗算回数を削減する仕組みをまとめると次のようになることが分かった。

(a) 行列乗算回数を削減できない C のブロックは 2 種類に分類できる。これを便宜的に T と L に記号化する。

(b) 上記 T と L を対角にする 2×2 ブロック行列乗算では、このブロックを構成する他の 2 個のブロック行列の計算において、行列乗算は 4 回から 3 回に削減できる。

これを、以下に 2×2 のブロックで図示する。ここで、 $\odot$ は行列演算が 1 回削減できるブロックを示す。 $\circ$ はそのブロックの相手ブロックを示す。

$$C = \begin{pmatrix} \odot & R \\ L & \circ \end{pmatrix}$$

以下、図 1 に 3×3 ブロック行列から 7×7 ブロック行列までを同様な形で図示する。この時、 $\times$ は一般的な行列乗算を使用することを示す。

図 1 から $n \times n$ ブロック行列は次の回数だけ行列乗算回数が減少することが分かる。

2, 4, 6 と n が偶数のときはそれぞれ 2, 5, 13 回と $[(n^2+2)/2 - n]$ 回減少

3, 5, 7 と n が奇数のときはそれぞれ 2, 8, 18 回と $[(n^2+1)/2 - n]$ 回減少

$$\begin{aligned}
C &= \begin{pmatrix} \odot & \times & T \\ \odot & T & \times \\ L & \bigcirc & \bigcirc \end{pmatrix} & C &= \begin{pmatrix} \odot & \odot & \odot & T \\ \odot & L & T & \bigcirc \\ \odot & T & L & \bigcirc \\ L & \bigcirc & \bigcirc & \bigcirc \end{pmatrix} & C &= \begin{pmatrix} \odot & \odot & \odot & \times & T \\ \odot & \odot & \odot & T & \times \\ \odot & L & T & \bigcirc & \bigcirc \\ \odot & T & L & \bigcirc & \bigcirc \\ L & \bigcirc & \bigcirc & \bigcirc & \bigcirc \end{pmatrix} \\
C &= \begin{pmatrix} \odot & \odot & \odot & \odot & \odot & T \\ \odot & \odot & \odot & L & T & \bigcirc \\ \odot & \odot & \odot & T & L & \bigcirc \\ \odot & L & T & \bigcirc & \bigcirc & \bigcirc \\ \odot & T & L & \bigcirc & \bigcirc & \bigcirc \\ L & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \end{pmatrix} & C &= \begin{pmatrix} \odot & \odot & \odot & \odot & \odot & \times & T \\ \odot & \odot & \odot & \odot & \odot & T & \times \\ \odot & \odot & \odot & L & T & \bigcirc & \bigcirc \\ \odot & \odot & \odot & T & L & \bigcirc & \bigcirc \\ \odot & L & T & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \odot & T & L & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ L & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \end{pmatrix}
\end{aligned}$$

図1 3×3 から 7×7 ブロック化での提案方式 (◎で行列乗算を 1 回削減)

行列乗算 $C=A \times B$ に上記方式を適用する場合、行列 C は $n \times n$ ブロック分割であるのに対し、行列 A は $n \times 2$ ブロック分割、行列 B は $2 \times n$ ブロック分割となる。 $n=4$ の場合の例を図 2 に示す。各ブロックが正方行列の場合、縦長行列と横長行列の乗算となる。

$$\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \\ A_{41} & A_{42} \end{pmatrix} \times \begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \end{pmatrix}$$

図2 結果が 4×4 ブロック行列となる行列乗算

4. 提案方法の具体的計算法

結果が 4×4 ブロック行列となる提案方式の具体的計算法を図 3 に示す。ここで、先頭文字が L,M,T,U 及び P,Q,R は中間ワーク行列を示す。先頭文字が L,M の中間ワーク行列は図 1 の L に相当し、 C_{41}, C_{22} 及び C_{33} の計算に利用される。同様に T,U は図 1 の T に相当し、 C_{14}, C_{23} 及び C_{32} はこの中間ワークだけから計算される。先頭文字が P のワークは図 1 の ◎ と ○ の両方に相当する。Q と R はそれぞれ、図 1 の ◎ と ○ の片方に対応し、Q は $C_{11}, C_{21}, C_{31}, C_{12}$ 及び C_{13} の計算に R は $C_{42}, C_{43}, C_{24}, C_{34}$ 及び C_{44} の計算に使用される。行列乗算は中間ワーク行列の計算時にだけ現れ、合計 27 回である。この回数は一般的計算法の 32 回に対して 5 回減少している。行列加算の回数は中間ワーク計算で 42 回、結果行列 C の作成で 36 回の合計 78 回である。

$L1 = (A_{41} - A_{42}) \times B_{11}$	$L2 = (A_{21} - A_{22}) \times B_{12}$	$L3 = (A_{31} - A_{32}) \times B_{13}$
$M1 = A_{42} \times (B_{11} + B_{21})$	$M2 = A_{22} \times (B_{12} + B_{22})$	$M3 = A_{32} \times (B_{13} + B_{23})$
$T1 = A_{11} \times (B_{14} + B_{24})$	$T2 = A_{21} \times (B_{13} + B_{23})$	$T3 = A_{31} \times (B_{12} + B_{22})$
$U1 = (A_{12} - A_{11}) \times B_{24}$	$U2 = (A_{22} - A_{21}) \times B_{23}$	$U3 = (A_{32} - A_{31}) \times B_{22}$
$P1 = (A_{11} + A_{42}) \times (B_{11} + B_{24})$	$Q1 = (A_{12} + A_{42}) \times (B_{21} - B_{24})$	$R1 = (A_{11} + A_{41}) \times (B_{14} - B_{11})$
$P2 = (A_{21} + A_{42}) \times (B_{11} + B_{23})$	$Q2 = (A_{22} + A_{42}) \times (B_{21} - B_{23})$	$R2 = (A_{21} + A_{41}) \times (B_{13} - B_{11})$
$P3 = (A_{31} + A_{42}) \times (B_{11} + B_{22})$	$Q3 = (A_{32} + A_{42}) \times (B_{21} - B_{22})$	$R3 = (A_{31} + A_{41}) \times (B_{12} - B_{11})$
$P4 = (A_{11} + A_{22}) \times (B_{12} + B_{24})$	$Q4 = (A_{12} + A_{22}) \times (B_{22} - B_{24})$	$R4 = (A_{11} + A_{21}) \times (B_{14} - B_{12})$
$P5 = (A_{11} + A_{32}) \times (B_{13} + B_{24})$	$Q5 = (A_{12} + A_{32}) \times (B_{23} - B_{24})$	$R5 = (A_{11} + A_{31}) \times (B_{14} - B_{13})$

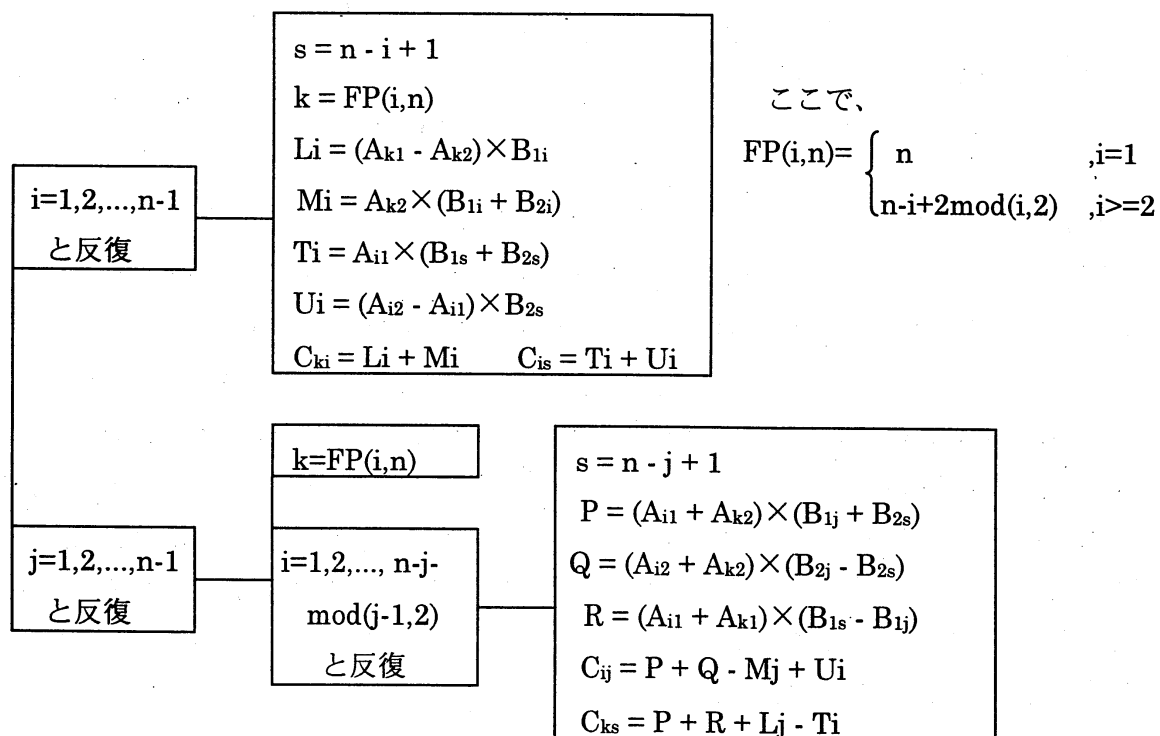
(a) 中間ワーク行列の計算

$C_{11} = P1 + Q1 - M1 + U1$	$C_{12} = P4 + Q4 - M2 + U1$	$C_{13} = P5 + Q5 - M3 + U1$	$C_{14} = T1 + U1$
$C_{21} = P2 + Q2 - M1 + U2$	$C_{22} = L2 + M2$	$C_{23} = T2 + U2$	$C_{24} = P4 + R4 + L2 - T1$
$C_{31} = P3 + Q3 - M1 + U3$	$C_{32} = T3 + U3$	$C_{33} = L3 + M3$	$C_{34} = P5 + R5 + L3 - T1$
$C_{41} = L1 + M1$	$C_{42} = P3 + R3 + L1 - T3$	$C_{43} = P2 + R2 + L1 - T2$	$C_{44} = P1 + R1 + L1 - T1$

(b) 結果 C の作成

図3 4×4 ブロック行列となる場合の具体的計算方法

更に一般化し、結果が $n \times n$ ブロック行列となる場合の具体的演算方式を図4に示す。但し、図4では n は偶数とする。 n が奇数の場合は図4を少し変更すれば良い。



この場合、 $C=A \times B$ の行列乗算は図 5 の様にブロック化表示される。

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ \vdots & \vdots \\ A_{n1} & A_{n2} \end{pmatrix} \times \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \end{pmatrix}$$

図 5 結果が $n \times n$ ブロック行列となる行列乗算

5. 対称行列への適用

$C=A^T \times A$ と結果が対称行列になる実密行列の行列乗算にも、ストラッセンの方法及び提案方式が直接適用できることが判明した。ここでは、 2×2 のブロック行列への適用方法を示す。結果が対称行列となる場合、結果は上三角行列に保存する。そのため、結果行列 C は上三角行列を 2×2 のブロック三角行列に分割する。 $C=A^T \times A$ における 2×2 のブロック行列化表示を図 6 に示す。

$$\begin{array}{|c|c|} \hline C_1 & C_2 \\ \hline \text{対称} & C_3 \\ \hline \end{array} = \begin{array}{|c|c|} \hline A_{11}^T & A_{21}^T \\ \hline A_{12}^T & A_{22}^T \\ \hline \end{array} \times \begin{array}{|c|c|} \hline A_{11} & A_{12} \\ \hline A_{21} & A_{22} \\ \hline \end{array}$$

図 6 $C=A^T \times A$ における 2×2 のブロック行列化表示

対称行列の演算量削減方式は下記の様にストラッセンの方法を変更して計算できる。主な変更点は、中間ワーク行列の計算時に上三角部分だけ計算すること及び C_3 が下三角行列のため、行列の転置が必要となる。更に行列加算回数を減少させるため、 $M6, M7$ が対称行列となるよう加減算の符号を変更した。ここで、 $U\{\}$ は上三角行列部分の計算を示す。

$$\begin{array}{ll} M1 = U\{(A_{12}^T - A_{22}^T) \times A_{11}\} & M2 = U\{A_{22}^T \times (A_{11} + A_{21})\} \\ M3 = U\{A_{11}^T \times (A_{12} + A_{22})\} & M4 = U\{(A_{11}^T - A_{21}^T) \times A_{22}\} \\ M5 = U\{(A_{11}^T + A_{22}^T) \times (A_{11} - A_{22})\} & M6 = U\{(A_{21}^T + A_{22}^T) \times (A_{21} + A_{22})\} \\ M7 = U\{(A_{11}^T + A_{12}^T) \times (A_{11} + A_{12})\} & \\ C_1 = M5 + M6 - M2 + M4 & C_2 = M3 - M4 \\ C_3 = M1^T + M2^T & C_4 = M7 - M5 - M1 - M3 \end{array} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{中間ワ-ク} \\ \\ \text{結果セ-ット} \end{array}$$

行列サイズが大き場合は対称行列でも、一般的計算方法に比較して演算量が約 $7/8$ に減少する。もちろん、結果の三角行列ブロック数が $n \times n$ となる提案方式にも適用でき、 n が大きく行列サイズも大きい場合は演算量は約 $3/4$ に減少する。これは、対称密行列を係数とする連立一次方程式の計算や固有値計算に役立つ。

6. 複素行列及びエルミート行列への適用

複素密行列及びエルミート密行列の場合は、それぞれ実密行列及び実対称密行列の場合と同様に提案方式で演算量を削減できる。更に、この場合は 1×1 ブロックの行列乗算そのものに図 7 の方式が適用でき、同次元数の実係数行列乗算の回数が 4 回から 3 に削減できる。そのため、複素行列の方が実行列より更に演算量の削減効果が大きくなる。

$(C_1 + C_2i) = (A_1 + A_2i) \times (B_1 + B_2i) \quad , \quad i \text{ は虚数単位を示す。}$		
$C_1 = A_1 \times B_1 - A_2 \times B_2$ $C_2 = A_1 \times B_2 + A_2 \times B_1$	$\left. \begin{array}{l} \\ \end{array} \right\}$	定義式による計算 (行列乗算は 4 回)
$P = A_1 \times (B_1 + B_2)$ $Q = (A_1 + A_2) \times B_2$ $R = A_2 \times (B_1 - B_2)$		
$C_1 = P - Q$ $C_2 = Q + R$	$\left. \begin{array}{l} \\ \\ \end{array} \right\}$	演算削減方式 (行列乗算は 3 回)
(a) 複素行列乗算の演算量削減		

$(C_1 + C_2i) = (A_1 - A_2i)^T \times (A_1 + A_2i) \quad , \quad i \text{ は虚数単位を示す。}$		
$C_1 = A_1^T \times A_1 + A_2^T \times A_2$ $C_2 = A_1^T \times A_2 - A_2^T \times A_1$	$\left. \begin{array}{l} \\ \end{array} \right\}$	定義式による計算 (行列乗算は 4 回)
$P = A_1^T \times (A_1 - A_2)$ $Q = (A_1^T + A_2^T) \times A_2$ $R = A_2^T \times (A_1 + A_2)$		
$C_1 = P + Q$ $C_2 = Q - R$	$\left. \begin{array}{l} \\ \\ \end{array} \right\}$	演算削減方式 (行列乗算は 3 回)
(b) エルミート行列乗算の演算量削減		

図 7 複素行列乗算及びエルミート行列乗算の演算量削減方法

7. 演算量の評価

(1) 行列乗算の演算数の算出

行列乗算 $C=A \times B$ で、 A, B, C をそれぞれ $n \times 2$ 、 $2 \times n$ 及び $n \times n$ ブロック行列に分割した提案方式の演算量を算出する。対称やエルミート行列の場合は $C=A^T \times A$ の行列乗算とする。 $n=2$ のときはストラッセンの方法となる。

A, B, C の行列サイズをそれぞれ $P \times R$ 、 $R \times P$ 、 $P \times P$ とする。 P は n の倍数で R は 2 の倍数とする。 A, B, C のブロック行列のサイズは $p \times r$ 、 $r \times p$ 、 $p \times p$ で次の様になる。

$$p = P/n, \quad r = R/2$$

表 1 に分割数 n とブロック行列のサイズ p, r による行列乗算の演算量計算式を示す。ブロック行列の乗算、 A と B の加減算及び C の加減算の回数をそれぞれ α, β, γ とし、それぞれの演算量を a, b, c とする。表 1 の記号 s は図 7 の演算削減方法を使用した場合の演算量である。 n が偶数の方が奇数の場合より効率が良い。

表1 提案方式の演算量計算式

内容	記号		実行列	対称行列	複素行列	エルミート行列
行列計算回数 (偶数 n)	乗算	α	$3n^2/2+n-1$			
	A,B 加算	β	$3n^2-2n+2$	n^2	$3n^2-2n+2$	n^2
	C 加算	γ	$3n^2-4n+4$			
行列計算回数 (奇数 n)	乗算	α	$(3n^2-1)/2+n$			
	A,B 加算	β	$3n^2-2n-3$	n^2-4	$3n^2-2n-3$	n^2-4
	C 加算	γ	$3n^2-4n+2$			
各計算の演算量	乗算 定義	a	$p^2(2r-1)$	$p(p+1)(2r-1)/2$	$2p^2(4r-1)$	$p(p+1)(4r-1)$
	乗算 提案	s	a	a	$p(6pr+3r-p)$	$p(6pr+10r-p-1)/2$
	A,B 加算	b	pr	pr	$2pr$	$2pr$
	C 加算	c	p^2	$p(p+1)/2$	$2p^2$	$p(p+1)$
総演算量	定義		$n^2(2a+c)$			
	提案		$\alpha s + \beta b + \gamma c$			

注) 偶数 n では図1のL,Tブロック数は $2n-2$ で◎,○ブロック数は n^2-2n+2

奇数 n では図1のL,Tブロック数は $2n-3$ で◎,○ブロック数は n^2-2n+1

ブロック化行列の行列乗算回数に着目すれば、分割数 n が 2,3,4,5,6, の場合定義方式ではそれぞれ 8,18,32,50,72 となるのに、提案方式では 7,16,27,42,59 となる。

(2) 行列乗算での演算量の比較

実密行列の乗算 $C=A \times B$ で演算量を、定義式と提案方式で比較する。A,B,C の行列はそれぞれ縦長、横長及び正方行列とし、サイズはそれぞれ以下の通りとする。これは、連立一次方程式や固有値の計算において、 α のサイズでブロック化すると発生する。

$$A: 2520 \times \alpha \quad B: \alpha \times 2520 \quad C: 2520 \times 2520$$

提案方式では行列 A,B,C をそれぞれ $n \times 2, 2 \times n$ 及び $n \times n$ に分割する。 n を 2 にするとストラッセンの方法となる。行列 C のサイズを 2520 としたのは、2 から 10 の最小公倍数で比較検討を容易にするためである。図8に α は 20,50 及び 200 で分割数 n が 2 から 10 とし、定義式に対する演算量の削減率を示す。

図8より n が偶数の方が奇数の場合より演算量削減効果が大で、 n が 4 以上なら分割数が大きくなるほど削減効果が増大することが分かる。

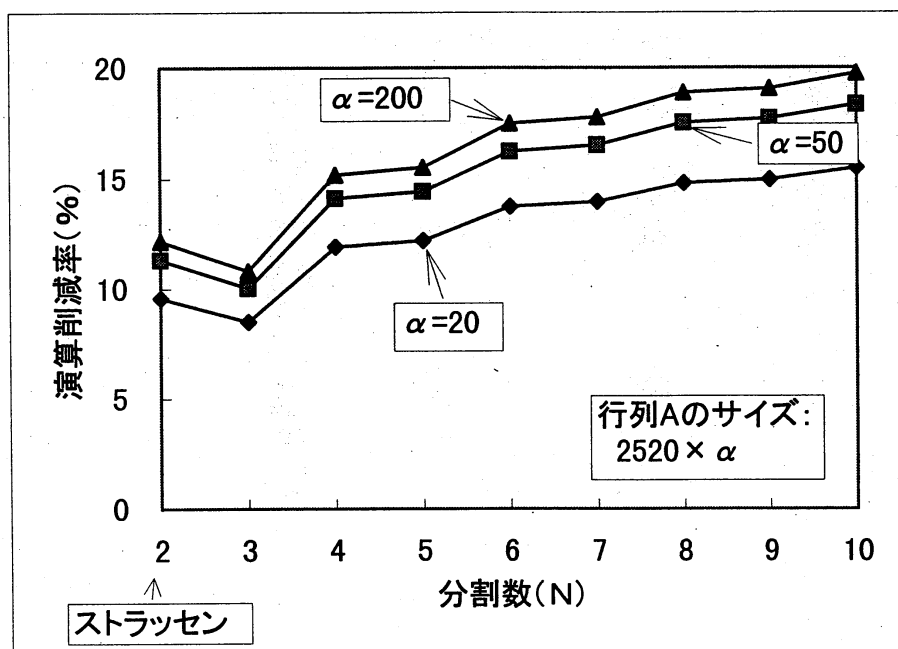


図8 実密行列乗算における提案方式の演算量削減率の比較

(3) 連立一次方程式(三角分解)での演算量の比較

提案方式を連立一次方程式の計算即ち、三角分解に適用することを考える。三角分解への適用では、 α のサイズでブロック化しながら計算するものとする。実及び複素係数の三角分解、即ちブロックガウス法の1ブロック段の消去の概念図を図9に示す。図10には対称又はエルミート行列のブロック三角分解の概念図を示す。行列のサイズはすべて2000次元とし、ブロック化サイズ α は80とした。図11に実行列と複素行列で提案方式を使用した場合の、一般的なガウス消去法に対する演算量の削減率を示す。図12に対称行列とエルミート行列の場合の削減効果を示す。分割数の1-1はブロック化しない場合を、2-2はストラッセンの方法を2度適用したことを示す。対称やエルミート行列の場合は5章の工夫をし、複素数やエルミート行列の場合は図7の工夫をした複合効果である。分割数の10-8は結果の行列を 10×10 と 8×8 に分割し、2度提案方式を適用することを示す。

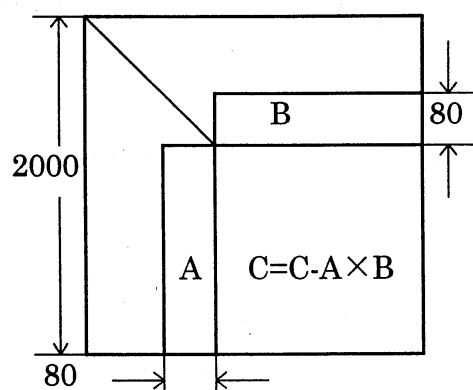


図9 実及び複素行列の三角分解

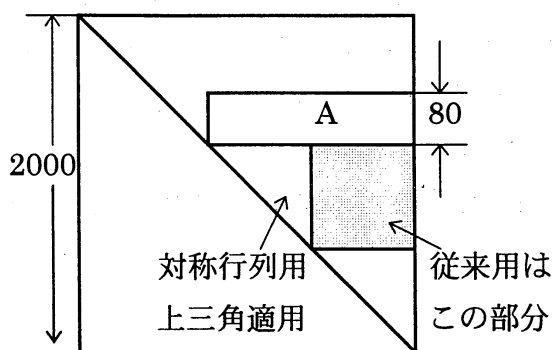


図10 対称及エルミート行列の三角分解

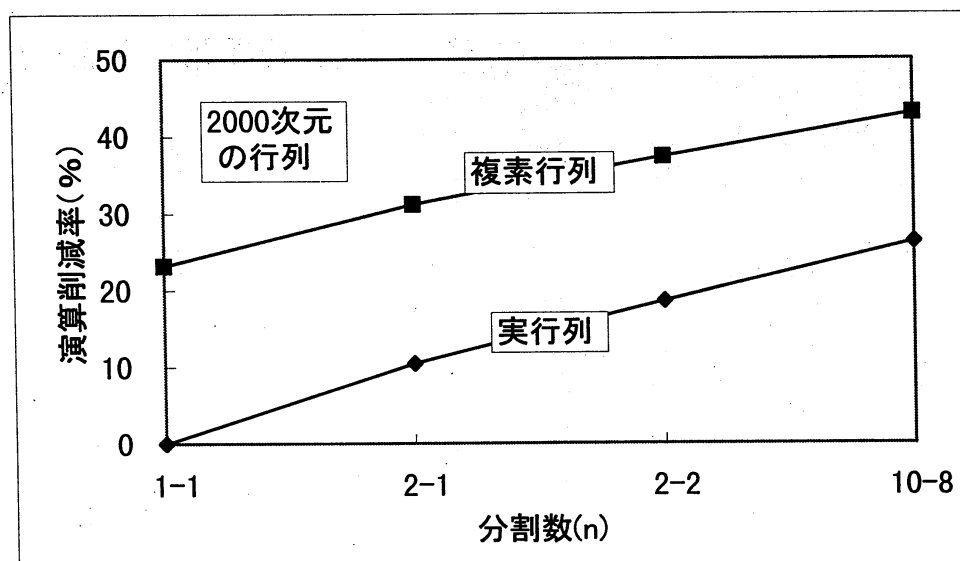


図11 実及び複素行列の三角分解での提案方式の演算量削減効果

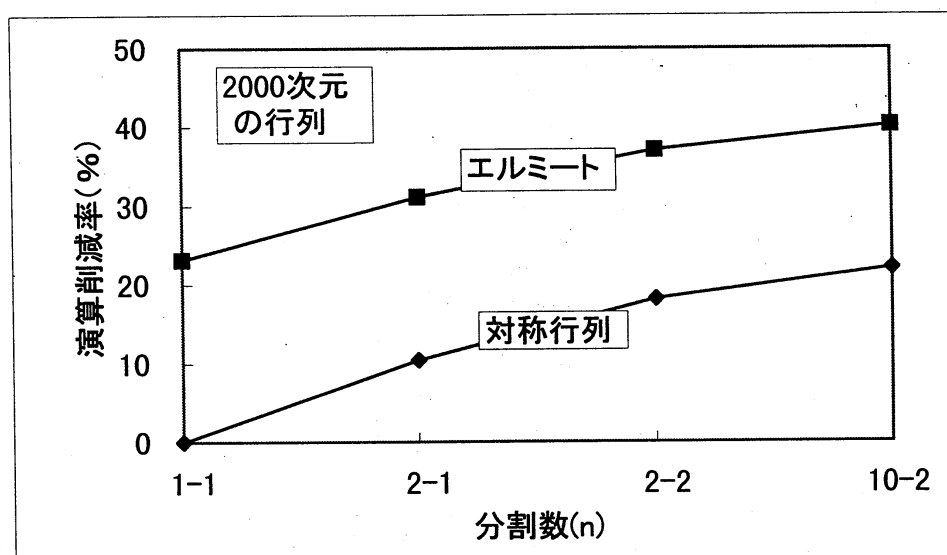


図12 対称及びエルミート行列の三角分解での提案方式の演算量削減効果

8. 結言

提案方式は、密行列の乗算でストラッセンの方法より更に、演算量削減効果が大きいことが判明した。行列の三角分解においても同様に効果大である。更に、対称及びエルミート行列にも直接適用できる方式を考案した。今後は、実際の計算機プログラムに本方式を適用してその効果と丸め誤差の影響を確認する。今後、多くの人々が連立一次方程式の計算や固有値計算に本方式を活用されて効果を発揮されるものと期待する。

9. 参考文献

- 1) V. Strassen(1968). "Gaussian Elimination is Not Optimal", Numer. Math. 13, 354-356
- 2) Gene H. Golb (1993) "Matrix Computation", Johns Hopkins University Press, 32-34