

プログラム: 3.1 (従来のLU分解法)

```

parameter (ix=l, iy=m, iz=n)
dimension (A(iy,iz,ix,5,5), B(iy,iz,ix,5,5),
c   C(iy,iz,ix,5,5), D(iy,iz,ix,5,5), E(iy,iz,ix,5,5))

do 3 i=1,ix
do 3 k=1,iz
do 3 j=1,iy

$$\bar{B}_{j,k}^i = B_{j,k}^i - A_{j,k}^i \times \bar{D}_{j,k}^{i-2}$$


$$\bar{C}_{j,k}^i = C_{j,k}^i - A_{j,k}^i \times \bar{E}_{j,k}^{i-2} - \bar{B}_{j,k}^i \times \bar{D}_{j,k}^{i-1}$$


$$\bar{D}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times (D_{j,k}^i - \bar{B}_{j,k}^i \times \bar{E}_{j,k}^{i-1})$$


$$\bar{E}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times E_{j,k}^i$$


$$\bar{b}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times (b_{j,k}^i - A_{j,k}^i \times \bar{b}_{j,k}^{i-2} - \bar{B}_{j,k}^i \times \bar{b}_{j,k}^{i-1})$$

3 continue
    
```

前進消去のプログラムを略記したのがプログラム 3.1¹ である。ここでは、 j 方向をベクトル化している。

3.3 ツイスト分解法の応用

従来のLU分解法に対して、行列の対角要素両端から中央の要素まで同時にLU分解を用いて前進消去し、そこで折り返して後退代入することができる。この分解法は、3重対角行列をはじめ、スパースな帯行列の構造に着目したときに、対角要素の両端から独立して計算する方法であり、“Alternative decomposition[64]” や “Twisted factorization(ツイスト分解)[63]” と呼ばれ、従来のLU分解と代数的に同値である。この分解法は、単一の線形方程式に対してCG法など反復解法の前処理の並列化に使用されている。

いま、3重対角行列 T を係数行列とする線形方程式

$$Tx = b, \quad T: N \times N \text{ 行列}, \quad x, b: N \text{ 次ベクトル} \tag{3.4}$$

の求解を考える。説明を簡単にするために、係数行列 $T = (t_{i,j})$ を対称な3重対角行列とし、 T の対角要素を a_i 、副対角要素を c_i と表記する。すなわち、

¹ 実際のプログラミングでは、 $i = 1, 2, (l-1), l$ に対しては場合分けが生じる。

\tilde{L} は図 3.1 のとおり, $(2 \text{ ブロック}) \times (2 \text{ ブロック})$ 行列であり,

- (a) L_{11} はサイズ $p \times p$ の下 2 重対角行列,
- (b) L_{22} はサイズ $(N - p) \times (N - p)$ の上 2 重対角行列,
- (c) L_{12} はサイズ $p \times (N - p)$ の零行列,
- (d) L_{21} は $(1, p)$ 要素を除いて零行列,
- (e) \tilde{D} は $\text{diag}(\tilde{L})$ である.

行列 T のサイズ N が偶数ならば, $p = \frac{1}{2}N$ であり, 奇数ならば, $p = \frac{1}{2}(N + 1)$ である.

そのアルゴリズムは以下のとおりである (N は偶数とする). ただし, インデックス外の要素の値は全て 0 とする.

アルゴリズム: 3.1 (ツイスト分解過程)

$$\bar{d}_i^{-1} = a_i - c_{i-1}^2 \bar{d}_{i-1}, \quad i = 1, 2, \dots, \frac{1}{2}N - 1, \quad (3.8)$$

$$\bar{d}_{N-i+1}^{-1} = a_{N-i+1} - c_{N-i+1}^2 \bar{d}_{N-i+2}, \quad i = 1, 2, \dots, \frac{1}{2}N - 1, \quad (3.9)$$

$$\bar{d}_{N/2}^{-1} = a_{N/2} - c_{N/2-1}^2 \bar{d}_{N/2-1}, \quad (3.10)$$

$$\bar{d}_{N/2+1}^{-1} = a_{N/2+1} - c_{N/2-1}^2 \bar{d}_{N/2+2} - c_{N/2}^2 \bar{d}_{N/2}. \quad (3.11)$$

このアルゴリズムにおいて, 式 (3.8)(3.9) は並列実行可能である. 式 (3.10) で表される $i = \frac{1}{2}N$ 番目の要素と式 (3.11) で表される $(\frac{1}{2}N + 1)$ 番目の要素だけは逐次計算する. したがって, L_{11} と L_{22} は, 式 (3.12)(3.13) のとおり独立して計算できる. 特に, N が奇数のときは, 対角要素の両端から分解してゆき, 行列の中央要素において 3×3 行列を逐次計算することになる. また, 係数行列 T が非対称の場合でも LU 分解を行えば良く, 上記アルゴリズムの計算過程は同様である.

係数行列 T が, 式 (3.7) のとおり分解されたとき, 線形方程式 (3.4) の解 x は,

$$\begin{aligned} \tilde{L}y &= b, \\ \tilde{D}\tilde{L}^T x &= y \end{aligned}$$

を解いて求められる. 実際のアルゴリズムは, 次のとおりである.

アルゴリズム: 3.2 (ツイスト分解による求解過程)

【前進代入】

$$\check{y}_i = b_i - c_{i-1}\check{y}_{i-1}, \quad i = 1, 2, \dots, \frac{1}{2}N - 1, \quad (3.12)$$

$$\check{y}_{N-i+1} = b_{N-i+1} - c_{N-i+1}\check{y}_{N-i+2}, \quad i = 1, 2, \dots, \frac{1}{2}N - 1, \quad (3.13)$$

本研究では、この分解法を複数の線形方程式を解くための直接法に応用し、ベクトル計算機の計算効率を改善するような前処理を施した解法へとさらに改良した。一般に、ベクトル計算機を用いて問題を解く場合、ベクトル長を十分に長くすると計算機性能を引き出すことができる [10]。計算機性能が飽和するに至らない程度のベクトル長においては、その長さを2倍にすることで計算効率の向上が期待できる。そこで本研究では、このベクトル計算機の一般的なアーキテクチャ特性を念頭に、式 (3.20) の両方向からの分解 $\hat{P}_{j,k}\hat{Q}_{j,k}$ に対して、中央から上下に二分割し、下半分の要素を 180 度回転してベクトル化方向の要素へ組み込むことでベクトル長を倍増させた。本研究の中では、この解法を Rotated Alternative LU 分解法 (以下、Rotated ALU 分解法と略記) と呼ぶことにする。詳細は以下のとおりである。

図 3.2 は、 $\hat{P}_{1,1}$ に対する Alternative decomposition (左) と Rotated ALU 分解 (右) をイメージで表現したものである。図中の実線、点線は左右の図で各々対応しており、矢印は LU 分解の順序を表している。Rotated ALU 分解では、 $\hat{P}_{1,1}$ を上下に分割し²、下半分の要素を図 3.2 のとおり 180 度回転して $j = 1, j = 2$ とする。 $\hat{Q}_{1,1}$ についても同様である。

行列群 $\hat{P}_{j,k}$ (ここで、 $j = 1, \dots, m; k = 1, \dots, n$) に対する Rotated ALU 分解では、上半分の要素だけを並べて $j = 1, \dots, m$ を形成し、下半分の要素を 180 度回転して $j = (m + 1), \dots, 2m$ を形成し、 j 方向をベクトル化する。これを $k = 1, \dots, n$ について行う。行列群 $\hat{Q}_{j,k}$ についても同様である。

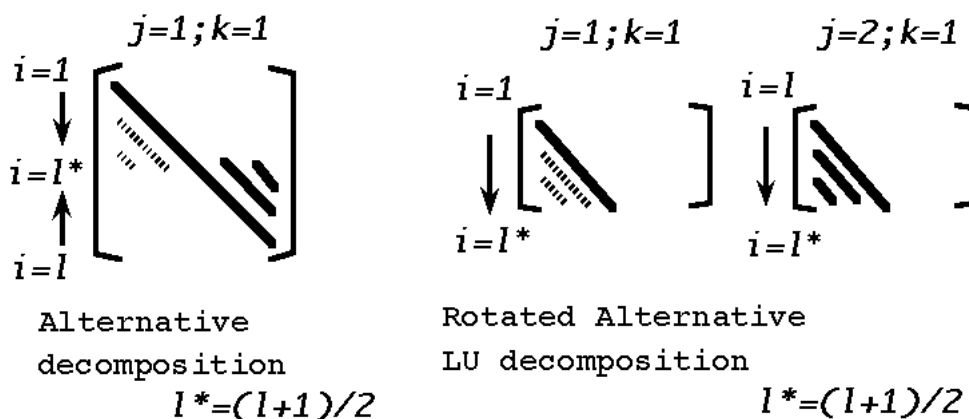


図 3.2: 前進消去における Alternative decomposition (左) と Rotated ALU 分解 (右) のイメージ。

この行列に対する LU 分解は、Alternative decomposition による分解と等しい演算を行う。プログラム 3.1 と同様に前進消去のプログラム例を次に示す。

² プログラミングでは、テクニックとして l を奇数に取る。偶数の場合は、二分割した後の i にダミー領域を設けて奇数にする。

プログラム: 3.2 (Rotated ALU 分解法)

```

parameter (ix=(l + 1)/2, iy=2m, iz=n)
dimension (A(iy,iz,ix,5,5), B(iy,iz,ix,5,5),
c   C(iy,iz,ix,5,5), D(iy,iz,ix,5,5), E(iy,iz,ix,5,5))

do 4 i=1,ix
do 4 k=1,iz
do 4 j=1,iy

$$\bar{B}_{j,k}^i = B_{j,k}^i - A_{j,k}^i \times \bar{D}_{j,k}^{i-2}$$


$$\bar{C}_{j,k}^i = C_{j,k}^i - A_{j,k}^i \times \bar{E}_{j,k}^{i-2} - \bar{B}_{j,k}^i \times \bar{D}_{j,k}^{i-1}$$


$$\bar{D}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times (D_{j,k}^i - \bar{B}_{j,k}^i \times \bar{E}_{j,k}^{i-1})$$


$$\bar{E}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times E_{j,k}^i$$


$$\bar{b}_{j,k}^i = (\bar{C}_{j,k}^i)^{-1} \times (b_{j,k}^i - A_{j,k}^i \times \bar{b}_{j,k}^{i-2} - B_{j,k}^i \times \bar{b}_{j,k}^{i-1})$$

4 continue
    
```

3.5 数値実験

ここでは、従来のLU分解法とRotated ALU分解法とを比較した結果を示す。実験では、流体問題で扱うサイズに基づいた、様々なベクトル長に対する計算時間を測定した。

3.5.1 実験内容

係数行列 $\hat{F}_{j,k}$ を構成する小行列は、Frank行列を基にLU分解できるように作成した。真の解 $x_{j,k}$ は、乱数を用いて作成した。右辺項 $b_{j,k}$ は、 $\hat{F}_{j,k}$ 、 $x_{j,k}$ と式(3.1)を用いて作成した。また、様々なベクトル長に対しても、全要素数 $(l \times m \times n)$ は一定となるようにした。LU分解する i 方向の要素数は固定し $(l = 63)$ 、それ以外の方向は $m \times n = 2400$ (一定) となるように m, n を変化させた(表3.1)。本実験では、 j 方向をベクトル化した。

表 3.1: 実験モデル.

l	63	63	63	63	63	63	63	63
m	40	60	80	120	160	240	300	480
n	60	40	30	20	15	10	8	5

時間計測は、従来のLU分解とRotated ALU分解による計算部分に対して実施した。

計算機は、富士通 VPP500(IEEE アーキテクチャ) の 1PE を用いた。

3.5.2 実験結果

計算時間を、図 3.3 に示す。横軸は j 方向の要素数 (m) を表し、縦軸は計算時間 [$msec$] を表しており、表 3.1 の各サイズにおける計算時間をプロットした。この時間を基に計算時間比をグラフにしたのが、図 3.4 であり、その値も記載した。

これらから、従来の LU 分解法に比べて Rotated ALU 分解法では、最高 30%程度まで計算時間が短縮したことが分かる。ここで、両解法の違いによる演算量の増減は無く、ベクトル化率も同じであるため、ベクトル長による計算時間の短縮と考えられる。

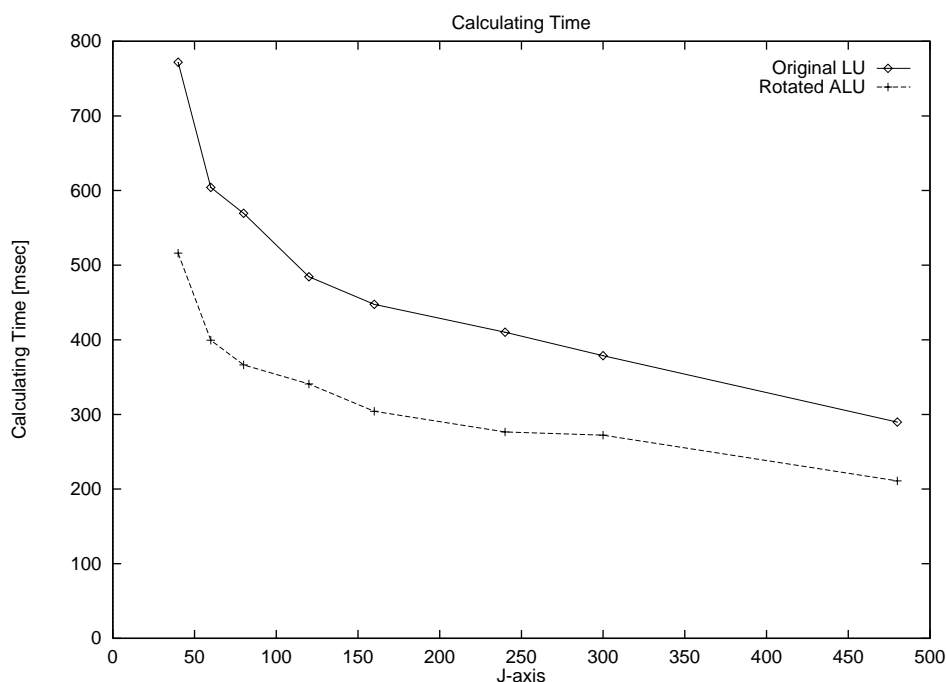


図 3.3: 従来の LU 分解法と Rotated ALU 分解法との計算時間の比較.

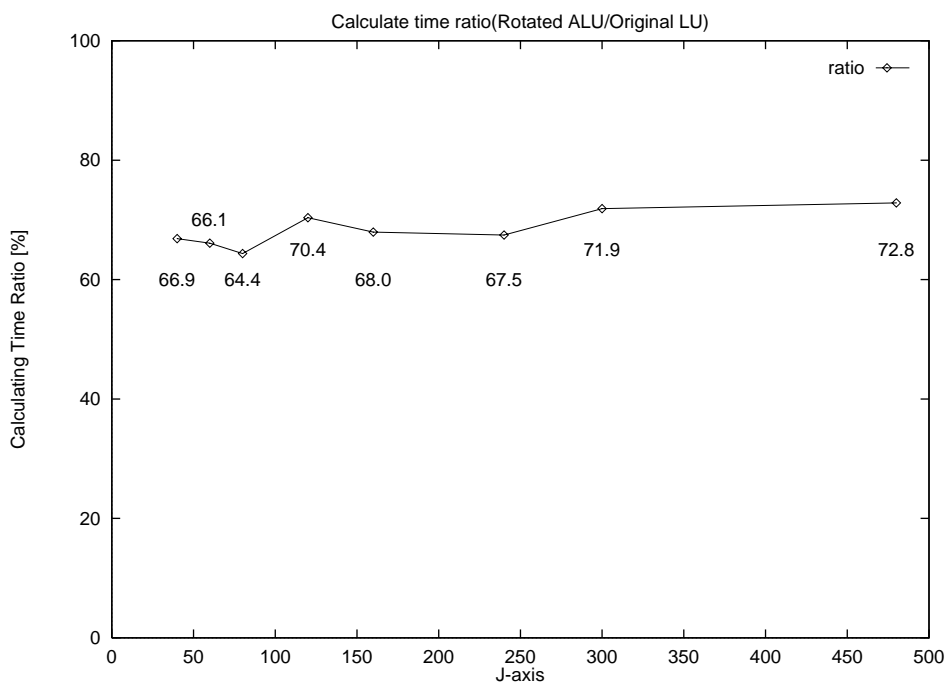


図 3.4: 従来の LU 分解法と Rotated ALU 分解法との計算時間比.

解の精度に関して，単精度で計算を行ったところ，表 3.1 の全てのモデルに対して，両アルゴリズムの誤差のオーダーは式 (3.21) に示すとおりであり，このテスト問題において同等の精度であることが確認された．これは，IEEE アーキテクチャの単精度におけるマシンイプシロン $\varepsilon_M = O(10^{-7})$ と比較しても十分な精度である [41] ．

$$\frac{\|\mathbf{x}_{j,k} - \bar{\mathbf{x}}_{j,k}\|_2}{\|\mathbf{x}_{j,k}\|_2} \sim O(10^{-6}) \tag{3.21}$$

$\mathbf{x}_{j,k}$: 真の解 , $\bar{\mathbf{x}}_{j,k}$: 数値解

3.6 まとめ

ブロック 5 重対角行列群を係数行列とする大規模な複数のシステムに対して，Rotated ALU 分解法は，従来の LU 分解法に比べると，ベクトル計算機の性能をさらに引き出すアルゴリズムであり，計算精度も保持していると言える．

流体の問題に限らず他の問題についても，同様の行列群に対してベクトル計算機で解く場合は，Rotated ALU 分解法を用いることを提案する．

この成果は文献 [25] として掲載された．