

## 差分法のベクトル計算†

石黒 美佐子\*\* 難波 克光\*\*\*

差分法は、原子力分野の計算の一つである中性子の拡散と輸送問題において、楕円型偏微分方程式の数値解を得るために使用されてきた。この方面の大型原子力計算コード(プログラム)では、大部分の計算時間が差分近似に基づく計算に費される。そこで、差分計算が、最近のスーパーコンピュータを用いたベクトル計算によって高速化されることが望まれる。本論文では、差分法のベクトル計算解法について、FACOM 230-75 APUを用いて計算時間の比較を行った結果について示す。1次元問題では3対角連立方程式のベクトル計算用新解法を比較、多次元問題では、反復解法の計算時間を、係数の優対角性およびベクトル化手法の違いにより比較する。

### 1. まえがき

原子力分野の計算のうち、中性子の拡散と輸送問題は、中性子の従う定常状態のボルツマン方程式を解き、原子炉内の中性子束分布と実効増倍係数を求めることを目的とする。これらの問題には、原子炉の形状が複雑な場合には有限要素法が適用されることがあるが、計算時間については差分法にはかなわない<sup>1)</sup>。

たとえば、中性子拡散は、エネルギー群番号  $g=1, 2, \dots, G$  に関する楕円型偏微分方程式が連立したもので表される<sup>2)</sup>。

$$\begin{aligned} & -\nabla d_g(\mathbf{r})\nabla\phi_g(\mathbf{r}) + \sigma_{r,g}(\mathbf{r})\phi_g(\mathbf{r}) \\ & = \sum_{g'=1}^{g-1} \sigma_{s,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \\ & + \chi_g/\lambda \sum_{g'=1}^G (\nu\sigma_f)_{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \end{aligned} \quad (1.1)$$

境界条件は次式で与えられる。

$$a(\mathbf{r}_B)d_g(\mathbf{r}_B)\partial\phi_g(\mathbf{r}_B)/\partial\mathbf{n} + b(\mathbf{r}_B)\phi_g(\mathbf{r}_B) = 0 \quad (1.2)$$

ここで、 $\lambda$  は実効増倍係数、 $\phi_g(\mathbf{r})$  は空間点  $\mathbf{r}$  における第  $g$  群の中性子束の値を表す。 $a, b$  は境界上の点  $\mathbf{r}_B$  に依存する定数、 $\mathbf{n}$  は法線方向のベクトルを表す。 $\chi$  は  $g$  に依存する非負の定数、 $d$  は拡散係数、 $\sigma_r, \sigma_s, (\nu\sigma_f)$  は、反応断面積とよばれ  $g$  と  $\mathbf{r}$  に依存して定まる非負の定数である。一般に原子炉は、炉心部に燃料が置かれ、その周りに吸収剤、遮蔽剤が置かれる。原子炉は、組成材質(核種)によって、空間的にいくつかの領域に区分されモデル化される。核種ごとにまたは中性子のエネルギーにより核反応の起こりや

すさが異なり断面積は異なる。

(1.1)式は、空間離散化によって、 $G \cdot N$  元の固有値問題となる。

$$(M-U)\phi = (1/\lambda)F\phi \quad (1.3)$$

ここで、 $\phi$  は  $G \cdot N$  元ベクトル、 $M, U, F$  は  $G \cdot N$  元行列で次のように表される。

$$\begin{aligned} \phi & = \begin{bmatrix} \phi_1(\mathbf{r}) \\ \phi_2(\mathbf{r}) \\ \vdots \\ \phi_G(\mathbf{r}) \end{bmatrix}, \quad M = \begin{bmatrix} M_1 & & & \\ & M_2 & & 0 \\ & & \ddots & \\ 0 & & & M_G \end{bmatrix}, \\ U & = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ U_{21} & 0 & 0 & \dots & 0 \\ U_{31} & U_{32} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{G1} & U_{G2} & \dots & \dots & 0 \end{bmatrix}, \\ F & = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1G} \\ F_{21} & \cdot & \cdot & F_{2G} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ F_{G1} & F_{G2} & \dots & F_{GG} \end{bmatrix} \end{aligned} \quad (1.4)$$

$\phi_g(\mathbf{r})$  は  $N$  元ベクトル、 $U_{gg'} = \sigma_{r,g'}(\mathbf{r})$  および  $F_{gg'} = \chi_g(\nu\sigma_f)_{g'}(\mathbf{r})$  は  $N$  元対角行列である。

$M_g = -\nabla d_g(\mathbf{r})\nabla + \sigma_{r,g}(\mathbf{r})$  は差分近似により得られる  $N$  元スティルチウス行列で、ある種のバンド行列となる<sup>2)</sup>。 $M_g$  は優対角行列で、優対角性  $\alpha$  を以下のように定義すると、 $\alpha$  はおよそ 1.2 から 2 の範囲の値をとる。

$$\alpha = \frac{(\text{対角要素の絶対値})}{(\text{対角外要素の絶対値の和})}$$

(1.3)式の左辺  $(M-U)$  は、正の対角要素をもち、 $\sigma_{r,g} \geq \sum_{g'=1}^{g-1} \sigma_{s,g'}$  であるという性質から、列に対し優対角となる。したがって、(1.3)式の解として、ユニークな正の固有ベクトル  $\phi$  が存在し、それに対応する固有値  $\lambda_0$  は、他の固有値の絶対値より大きいことが Birkhoff と Varga により示されている<sup>2)</sup>。

† Vectorization of Finite-Difference Method by MISAOKO ISHIGURO (Japan Atomic Energy Research Institute, Computing Center) and KATSUMI NANBA (Department of Scientific Technology, the University of Tsukuba).

\*\* 日本原子力研究所計算センター

\*\*\* 筑波大学理工学修士

(1.3)式で最大固有値を得るために、通常、累乘法 (power method) や Wielandt の反復法が使用される<sup>2)</sup>。累乘法では、(1.3)式から次の反復法となる。

$$(M-U)\phi^{(n)} = F\phi^{(n-1)} \quad (1.5)$$

$$\lambda^{(n)} = (F\phi^{(n)}, F\phi^{(n)}) / (F\phi^{(n)}, \phi^{(n-1)}) \quad (1.6)$$

$$\phi^{(n)} = (1/\lambda^{(n)})\phi^{(n)} \quad (1.7)$$

$\phi^{(n)}$  が定まれば、 $\lambda^{(n)}$  および  $\phi^{(n)}$  は陽に計算される。(1.5)式で  $\phi^{(n)}$  を求めることが問題となる。 $U$  が下三角行列であるから、(1.5)式は以下のように、 $G$  個の独立の  $N$  元の自己共役型連立方程式として書き下せる。

$$\begin{aligned} M_1\phi_1^{(n)}(\mathbf{r}) &= \sum_{g'} F_{1g'}\phi_{g'}^{(n-1)}(\mathbf{r}) \\ M_2\phi_2^{(n)}(\mathbf{r}) &= U_{21}\phi_1^{(n)}(\mathbf{r}) + \sum_{g'} F_{2g'}\phi_{g'}^{(n-1)}(\mathbf{r}) \\ &\vdots \\ M_G\phi_G^{(n)}(\mathbf{r}) &= \sum_{g'=1}^{G-1} U_{Gg'}\phi_{g'}^{(n)}(\mathbf{r}) + \sum_{g'} F_{Gg'}\phi_{g'}^{(n-1)}(\mathbf{r}) \end{aligned} \quad (1.8)$$

(1.8)式により  $\phi_g^{(n)}$  を求める過程自身はまた右辺を既知量として  $g=1, 2, \dots, G$  の順に  $N$  元連立1次方程式を解く問題となる。通常、おのおの独立にSOR法などの反復解法が用いられることから内側反復計算とよばれる。これに対し、(1.6)と(1.7)式によって  $\lambda^{(n)}$  や  $\phi^{(n)}$  を求める過程は外側反復計算とよばれる (図1)。

原子力コードでは、通常、各外側反復計算の初めに、Coarse Mesh rebalancing 法により、大まかなメッシュ分割 (たとえば、組成核種毎に空間を区分) で計算し、あらかじめ中性子束の初期値を改良しておく方法がよく用いられる<sup>3)</sup> が計算時間については無視できる。

中性子の拡散または輸送問題を差分法により計算す

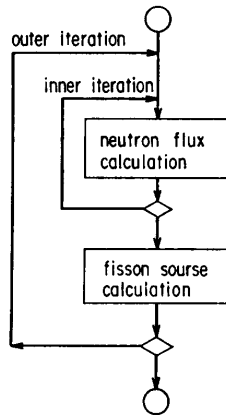


図1 拡散コードにおける二重反復計算

Fig. 1 Double loop iteration in a diffusion code.

表1 差分計算コードの二重反復計算のコスト (%)

Table 1 Time cost (%) of double loop iteration in finite-difference codes.

原子力コード名	問題	反復計算のタイムコスト (%)
CITATION	拡散	85.5
VENTURE	拡散	84.7
ADC	拡散	90.3
TWOTRAN	輸送	92.0
ANISN	輸送	86.5

る5件の代表的な大型原子力コードについて、計算実行時のプロフィールを、ソフトウェアツール FORTUNE<sup>4)</sup>を用いて測定した結果<sup>5)</sup>を表1に示す。計算時間の85~90%が二重反復計算に費やされていることがわかる。

二重反復計算のうち、内側 (反復) 計算は、計算時間上で重要な重みをもち、かつ解法による差違が現れる部分である。今後この論文では、(1.5)式のベクトル計算解法を問題にする。

差分法による反復計算では、あるメッシュ点における値は、同一群のその点に隣接する点の値を使って計算される。たとえば、2次元問題では、次の5点階差式となる。

$$\begin{aligned} \phi_{i,j}^{(m)} &= a_{i,j}\phi_{i,j}^{(m-1)} + b_{i,j}\phi_{i+1,j}^{(m-1)} + c_{i,j}\phi_{i,j-1}^{(m-1)} \\ &\quad + d_{i,j}\phi_{i-1,j}^{(m-1)} + e_{i,j}\phi_{i,j}^{(m-1)} \end{aligned} \quad (1.9)$$

$a_{ij} \sim e_{ij}$  は定数、 $m$  は内側反復回数

このことから(1.8)式の係数行列  $M_g$  は、規則性のあるある種の行列となる。 $N, M, L$  をおのおの  $x, y, z$  方向のメッシュ数とするなら、1次元問題は、 $N$  元の3対角行列 ( $e_i, d_i, f_i$ ) となる。

2次元問題は、(1.9)式から  $N \cdot M$  元の5対角行列で、その構造は、ブロック3対角行列 ( $E_i, D_i, F_i$ ) となる。

ここで、 $E_i, F_i$  は  $M$  元の対角行列、 $D_i$  は3対角行列である。

$$A = \begin{bmatrix} D_1 F_1 & & & & 0 \\ E_2 D_2 F_2 & & & & \\ & \ddots & & & \\ & & \ddots & & F_{N-1} \\ 0 & & & E_N D_N & \end{bmatrix} \quad (1.10)$$

3次元問題は、7点階差式の計算に帰着されるので、その係数行列は  $N \cdot M \cdot L$  元の7対角行列で、その構造は、二重にブロック化された3対角行列となりその対角要素は2次元問題の係数行列  $A$  と同じ形となる。

(1.8)式の解法は、計算速度の点で1次元問題では

直接解法が、多次元問題では反復解法が有利とされている。

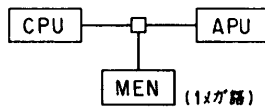
1次元問題における3対角方程式のベクトル計算解法としては、サイクリックリダクション法<sup>6)</sup>、リカーシブダブリング法<sup>7)</sup>、加速パラレルガウス法<sup>8)</sup>などの新解法が提案され、従来のガウスの消去法などとの計算時間の比較が並列プロセッサ方式のベクトル計算機 ILLIAC IV 上で実測、報告されている<sup>7),8)</sup>。本論文では、パイプライン方式のベクトル計算機 FACOM 230-75 APU (図2)<sup>9)</sup>を用いた数値実験によって、これらの新解法を含む8種の解法についての計算速度を比較する。

2次元問題では、通常、SOR法やSLOR法などの反復解法が用いられている。ベクトル計算機では、互いに依存するデータを同時に計算できないことから(図3)、ガウスの消去法や、最近のPreconditional CG法に基づく不完全LU分割法<sup>11)</sup>には効力を発揮できない。

一方、(1.9)式の5点階差式の計算を、 $j=1, 2, \dots, M$  について1回のベクトル計算により実行しようと

1. パイプライン方式の並列計算機

2. 構成



3. 1語は36ビット

(数値実験は単精度で計算)

4. プログラムは AP-FORTRAN<sup>10)</sup>

図2 FACOM 230-75 APU の特徴

Fig. 2 Feature of FACOM 230-75 APU.

1. ベクトル化できる例

```
DO 10 I = 1, N
10 A(I) = B(I) + C(I) * D(I)
INDEX I / 1, N /
A(I) = B(I) + C(I) * D(I)
```

2. ベクトル化できない例

```
DO 20 J = 2, N
20 B(J) = B(J) - E(J) * B(J-1)
```

図3 AP-FORTRAN によるプログラム

Fig. 3 Program by AP-FORTRAN.

表2 CRAY-1 による VENTURE サンプル問題の計算時間

Table 2 Computing time of VENTURE sample problems by CRAY-1.

サンプル問題	メッシュ数 Column, Row, Plane	オリジナルプログラム (スカラー計算)		ベクトル化プログラム		
		全 CPU 時間(秒)	繰返回数	全 CPU 時間(秒)	比率 (%)	繰返回数
問題1 3次元 X-Y-Z 形状 2群	9 × 9 × 5 = 405	2.61	33	1.38	53.0	37
	18 × 9 × 5 = 810	5.31	36	2.48	46.7	48
	36 × 9 × 5 = 1620	8.92	31	3.47	38.9	42
問題2 2次元 R-Z 形状 2群	58 × 50 = 2900	24.98	46	12.12	48.5	81
	116 × 50 = 5800	54.29	44	27.39	50.5	95
問題3 3次元 X-Y-Z 形状 3群	13 × 13 × 18 = 3042	28.22	36	12.43	44.0	41
	26 × 13 × 18 = 6084	66.52	44			収束せず
	52 × 13 × 18 = 12168	122.29	41	62.20	50.9	80

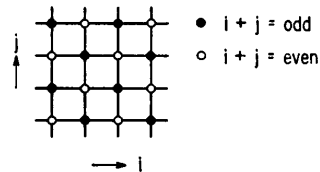


図4 奇数-偶数交互法

Fig. 4 Odd-even ordering method.

するならば、 $\phi_{i,j-1}^{(m)}$  の代わりに  $\phi_{i,j-1}^{(m-1)}$  を用いて計算しなければならなくなり、収束の遅れを招くことになる。ベクトル計算による反復回数の増加は、致命的欠点である。事実、CRAY-1 を利用した VENTURE コードのテスト計算<sup>5)</sup>で収束の劣化が顕著に現れており、ベクトル計算による速度向上が反復回数の増加で半分相殺されている(表2)。本論文では、このような収束の劣化を防ぐ odd-even SOR 法(図4)をとりあげ、ブロック LU 分解法などの直接解法との計算時間比較、係数の優対角性と反復回数の関係などが明らかにされる。

さらに3次元問題では、ベクトル計算の実現手法によって反復回数と計算時間がどう違うかを調査する。ベクトル計算の立上りに要する時間を考慮すると、あまりベクトル長の短いものは、速度向上の効果が期待できない。x-y 平面上のメッシュを1回の(odd-even法では交互に2回)ベクトル計算により実行するなど反復解法における8種の手法を比較する。

2. 3対角連立方程式のベクトル計算

2.1 解法とデータ

従来の数値解法、(1)ガウスの消去法(GET)、(2)

ヤコビの反復法 (JC), (3) チェビセフ加速法 (CH), (4) 共役傾斜法 (CG) に加えて次の並列計算解法について実験する.

(5) パラレルガウス法 (PG)<sup>8)</sup>

ガウスの消去法の LDR 分解の 3 ステップを反復計算によりベクトル化する.

(6) 加速パラレルガウス法 (APG)<sup>8)</sup>

PG 法の反復計算を, 数個おきのデータをベクトル化することにより収束を速める.

(7) リカーシブダブリング法 (RD)<sup>7)</sup>

LU 分解の前進消去, 後退代入でベクトル計算できるように数回のステップをまとめて解く.

(8) サイクリックリダクション法 (CR)<sup>6)</sup>

3 対角行列の奇数行と列を消去し, 再び 3 対角行列を得る. この変換を繰り返すうちに, 次数が低下し最後に対角行列として一部の解を得, 後退代入によりすべての解を得る.

反復法の打ち切り相対誤差を  $10^{-5}$  とする.

数値計算に用いた係数行列は, 優対角性の強いもの, 弱いものを取り混ぜて 10 ケースから成る. そのなかには計算精度の比較を目的とした Givens の行列 ( $\alpha \approx 1$ , 極端に優対角性が弱い) が含まれる.

## 2.2 計算結果

(1) F 75 APU による計算時間の全体的な比較は図 5, スカラ逐次計算との比較は表 3 で示される. CR 法が, 従来の GET 法よりも 6 倍も速いことがわかる.

(2) 反復解法の計算時間 (反復回数) と精度は,

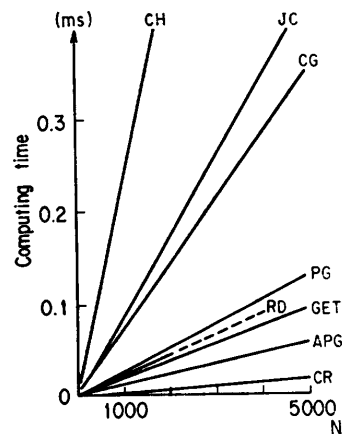


図 5 FACOM 230-75 APU における計算時間. 1次元問題, 乱数データ ( $\alpha=1.25 \sim 2.5$ )

Fig. 5 Computing time on the FACOM 230-75 APU. Random data ( $\alpha=1.25 \sim 2.5$ ) for one-dimensional problem.

表 3 3対角方程式の計算時間の比較

Table 3 Comparison of the computing times for tridiagonal system of equations.

解法	ベクトル計算 (ミリ秒)	逐次計算 (ミリ秒)	F 75 CPU F 75 APU
CR (サイクリックリダクション)	3	24	8.0
APG (加速パラレルガウス)	12	53	4.4
GET (ガウス消去法)	18	9	0.5
RD (リカーシブダブリング)	20	182	9.1
PG (パラレルガウス)	27	39	1.4
CG (共役傾斜)	72	715	10.0
JC (ヤコビ)	81	180	2.2
CH (チェビセフ加速)	235	276	1.2

$N=1,000$

Random data  $\alpha=1.25 \sim 2.5$

係数行列の優対角性 ( $\alpha$ ) に依存する. APG 法では,  $\alpha$  が 1 に近づくにつれて反復回数が急激に増加し, 妥当な時間内に計算が完了しなくなる.

(3) RD 法は, 指数部が 15 ビットもある ILLIAC 法に用いられた解法で, F 75 APU のように指数部が短い (8 ビット) 計算機では, オーバフローに悩まされ,  $\alpha$  が 1 に近い場合には解が得られない. また表 3 の逐次計算の時間を見てわかるように, 解法そのものの計算量の多さから, CR 法に比べて計算時間を要する面もある.

(4) Givens のデータのように,  $\alpha \approx 1$  の場合には, CR 法と GET 法以外では, 誤差が  $10^{-2}$  以上 (単精度計算) と大きくなり使用に耐えない.

結局, 計算速度, 精度の面で CR 法が有効であるが, CR 法は, 主記憶量が  $3N \log_2 N$  語と大きいのが難点である.

## 3. 2次元差分問題

### 3.1 解法とデータ

(1.10) 式のブロック 3 対角行列を係数にもつ方程式について, F 75 APU を用いて次の 5 種の解法について実験した.

(1) ガウスの消去法 (Gauss)

バンド幅  $(2M+1)$  のバンド行列と見なし, 掃出しを行う. ベクトル計算は  $(M+1)$  個の  $y$  方向要素に対して適用される.

(2) ブロック LU 分解法 (BLU)

ブロック 3 対角行列をブロック  $L$  行列, ブロック  $U$  行列に分解する. LU 分解自身は  $x$  方向に対して逐次的に行われるが, 各分解ステップにおける  $y$  方向の行列間演算は, AP-FORTRAN の行列特殊関数<sup>10)</sup>を使用してベクトル計算される.

(3) ブロックサイクリックリダクション法 (BCR)<sup>6)</sup>

ブロック3対角行列に対し、3対角行列用のCR法を適用する。行列間演算をすべてベクトル計算する。

(4) 通常のSOR法

(1.9)式の差分計算で $x-y$ 平面上の全メッシュを1回のベクトル計算により実行する。

(5) odd-even SOR法

図4で示すように、 $i+j=odd$ ,  $i+j=even$ となるメッシュを交互にベクトル計算する。

相対誤差は $10^{-5}$ とし、計算時間としては最適過大緩和因子を用いた時のものを取りあげる。

計算に用いたデータは、乱数データで優対角性が強いもの( $\alpha=1.25\sim 2.5$ )、弱いもの( $\alpha=1.04$ )、定数対称データ $(-1, 16, -1)$ ( $\alpha=4$ )、 $(-1, 4, -1)$ ( $\alpha=1$ )計4種から成る。ここで、 $(-1, 16, -1)$ は(1.10)式で与えた行列形式で、対角要素はすべて16、その他の要素は $-1$ をとる。

### 3.2 計算結果

3種の直接解法は、いずれも $y$ 方向メッシュをベクトル計算することが計算速度に効いていることから、ブロック数を固定し( $N=30$ )ブロックのサイズ( $M$ )を変化させた場合の計算時間を見ることにする。

(1) 図6は、優対角性がきわめて強いデータ $(-1, 16, -1)$ と最も弱いデータ $(-1, 4, -1)$ を対比させ、図示したものである。BLU法やGauss法などの直接法は、解がもし得られたなら優対角性による計算時間の差はない。

$(-1, 16, -1)$ では、2種のSOR法は反復回数が少なく、直接法に比べて計算時間はきわめて短い。ブロックのサイズが大きくなるにつれて、その差が著しくなってくる。

$(-1, 4, -1)$ では、SOR法は反復回数の増加のために時間がかかり、通常のSOR法では精度の条件を満足する解が得られない場合が多い。odd-even SOR法でさえ、計算時間はGauss法やBLU法より長い。BCR法は精度を満足する解が得られない。

(2) 図6および図7は、通常のSOR法とodd-even SOR法の計算時間の差は、優対角性が弱くなる( $\alpha$ が1に近づく)ほど大きくなることを示している。現実の中性子拡散計算では、図7の2種のデータの中間を動く。

(3) 最適過大緩和因子 $\omega_{opt}$ は、表4で示すようにodd-even法では優対角性が強いほど1に近づき、

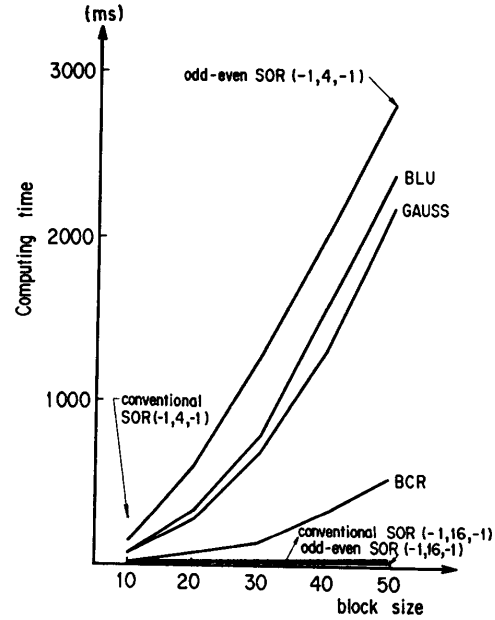


図6 FACOM 230-75 APUにおける計算時間。  
2次元問題、対称データ、ブロック数=30

Fig. 6 Computing time on the FACOM 230-75 APU. Symmetrical data for two-dimensional problem, number of blocks=30.

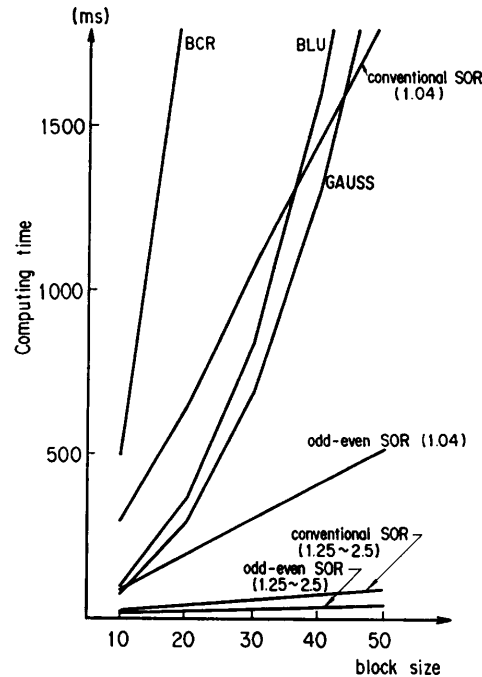


図7 FACOM 230-75 APUにおける計算時間。  
2次元問題、乱数データ、ブロック数=30

Fig. 7 Computing time on the FACOM 230-75 APU. Random data for two-dimensional problem, number of blocks=30.

表 4 SOR 法における最適過大緩和因子 ( $\omega_{opt}$ )  
Table 4 Optimum overrelaxation factor ( $\omega_{opt}$ ) in SOR method.

Size	Data (-1, 16, -1)		(-1, 4, -1)		乱数 $\alpha=1.25\sim 2.5$		乱数 $\alpha=1.04$	
	conv.	odd-even	conv.	odd-even	conv.	odd-even	conv.	odd-even
30x30	1.10	1.00	un-solvable	1.97	1.00	1.22	1.00	1.87

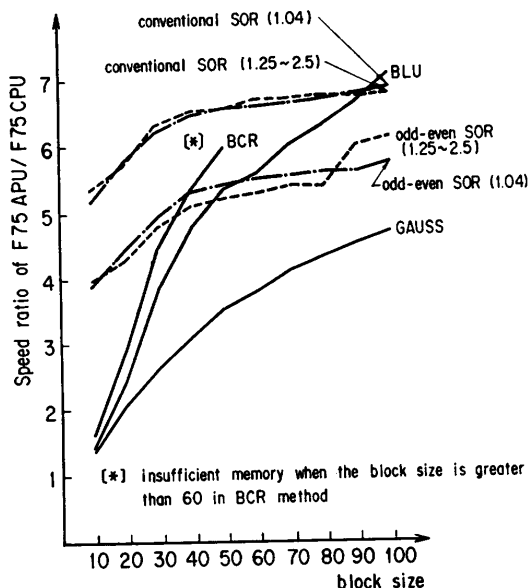


図 8 FACOM 230-75 APU と CPU の計算。  
2次元問題, 乱数データ, ブロック数=20

Fig. 8 Computing speed ratio of FACOM 230-75 APU/CPU. Random data for two-dimensional problem, number of blocks=20.

弱い場合には2に近い値となる。 $\omega_{opt}$ の値はブロックのサイズにあまり影響されない。30x20, 30x40の場合もほぼ同じ値となる。

(4) 図8は、F75 CPU と F75 APU の計算時間比を示す。直接法よりも SOR 法のほうが全体としてベクトル計算の効果が上がる。odd-even SOR 法で4~5倍の速度向上となる。

(5) BCR 法や BLU 法は、リダクションまたは分解の各ステップで行列の逆転が必要となり、計算時間を費やす。BCR 法については、この行列の逆転のために1次元問題を解く場合の CR 法の有利さが相殺される。

結局、現実の計算コードのメッシュ数、行列の優対角の割合を考慮すると odd-even SOR 法が優れている。

表 5 3次元問題における内側反復回数(最初の外側反復)  
Table 5 Number of inner iterations (first outer iteration) for three-dimensional problem.

data	method	$\omega$																			
		0.8	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	ADN0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	
F-1	SOR scalar	53	43	34	27	38	51	-													
	parallel (i)	70	61	53	64	-															
	(i,j)	80	71	-																	
	(i,j) odd-even	51	41	32	24	33	82	-													
	(i,j,k) odd-even	49	38	28	19	25	63	-													
	SLOR scalar	50	43	36	30	35	67	-													
	parallel (k)	60	53	47	49	90	-														
	(j,k)	70	63	78	-																
	(j,k) odd-even	44	36	29	22	32	51	-													
	modified SLOR (u)	40	33	27	34	40	50	-													
k+j=u	37	23	33	43	53	63	73	82	92												
ADI																					
F-2	SOR scalar	8	7	8	10	13	16	21	30	50											
	parallel (i)	9	8	10	14	20	31	78	-												
	(i,j)	11	10	13	19	42	-														
	(i,j) odd-even	8	7	8	10	12	16	22	29	50											
	(i,j,k) odd-even	7	6	8	10	12	15	20	28	45											
	SLOR scalar	8	7	9	11	14	18	26	60	-											
	parallel (k)	9	8	10	12	17	28	98	-												
	(j,k)	10	10	13	19	69	-														
	(j,k) odd-even	8	6	8	10	13	18	25	46	-											
	modified SLOR (u)	7	7	8	10	13	17	22	32	54											
k+j=u	14	6	7	8	10	11	13	14	16												
ADI																					
F-3	SOR scalar	13	11	10	10	13	18	22	34	62											
	parallel (i)	15	14	13	19	26	69	-													
	(i,j)	18	16	21	61	-															
	(i,j) odd-even	12	10	8	11	12	16	21	28	50											
	(i,j,k) odd-even	12	10	7	9	11	15	20	28	44											
	SLOR scalar	13	11	10	11	16	23	32	83	-											
	parallel (k)	15	13	12	16	27	93	-													
	(j,k)	17	15	21	42	-															
	(j,k) odd-even	12	10	8	11	12	17	28	67	-											
	modified SLOR (u)	9	10	11	15	19	22	42	60	-											
k+j=u	11	16	8	9	11	14	16	18	20	22											
ADI																					

— optimum, - diverge

### 4. 3次元差分問題

3次元差分問題に関しては、文献12)の中で触れられているので、数値計算結果を表5に示すに留める。

平面上のメッシュをベクトル計算の対象とする odd-even SOR 法および odd-even SLOR 法が、反復計算に要する計算ステップ数が少なく、かつ反復回数が少ない点で有利である。しかしながら、SLOR 法はある1方向(たとえば、x方向)に関しては、逐次的にしか計算できない解法なので、原子力コードで適用する場合には、プログラムの大幅な書直しが必要となる。

### 5. まとめ

ベクトル計算による速度向上倍率は、計算機のアーキテクチャやプログラム作成の仕方次第で多少の違いは出るものと思われるが、2~4章で得た結果は、一応の

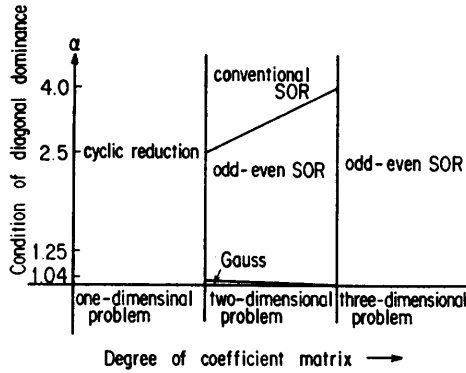


図9 有限差分計算における最適の方法

Fig. 9 The most feasible method in finite-difference calculations.

傾向を示すものとして有効である。

優対角性 ( $\alpha$ ) との関係で、最適の方式を図にまとめると図9のようになる。

最近の preconditional CG 法のような直接解法が、原子力分野で種々な問題について安定した解が、従来の SOR 法などより速い計算時間で得られる保証があれば、原子力コードのなかですでに使用されていてしるべきであろう。逐次解法で有望とされる解法をベクトル計算向きに改良することも有効である。

ベクトル計算については、差分法のようにごくありふれた問題でも定性的なことがまだわかっていないことが多い。大型計算プログラムで使用される数値計算アルゴリズムのベクトル処理に関しては、今後の経験の蓄積が重要である。

**謝辞** 数値計算にあたり、3対角方程式の解法では、原研の原田、藤井、藤村、中村の各氏に各種解法を分担いただきましたことを感謝する。

### 参考文献

- 1) Estiot, J. C. et al.: Comparison of Finite-Differences and Finite Elements in the Case of a Large Fast Power Reactor, *Ann. Nucl. Energy*, Vol. 8, pp. 609-620 (1981).
- 2) Wachspress, E. L.: *Iterative Solution of Elliptic Systems*, pp. 61-88, Prentice-Hall, Inc.,

New Jersey (1977).

- 3) Lathrop, K. D. and Brinkley, F. W.: *TWO-TRAN-II, LA-4848-MS*, pp. 12-14, Oak Ridge National Laboratory (1973).
- 4) 富士通: FORTRAN HE 使用手引書, pp. 160-170 (1979).
- 5) 鎌田 稔他: CRAY-1 と FACOM 230-75 APU による拡散コード VENTURE のベクトル演算化, JAERI-M 82-019 (1982).
- 6) Heller, D. E.: Some Aspects of the Cyclic Reduction Algorithm for Block Tridiagonal Linear Systems, *SIAM J. Num. Anal.*, Vol. 13, No. 4, pp. 484-496 (1976).
- 7) Stone, H. S.: An Efficient Parallel Algorithm for the Solution of a Tridiagonal System of Equations, *JACM*, Vol. 20, No. 1, pp. 27-38 (1973).
- 8) Heller, D. E. et al.: Accelerated Iterative Methods for the Solution of Tridiagonal Systems on Parallel Computers, *JACM*, Vol. 23, No. 4, pp. 636-654 (1976).
- 9) 三輪 修他: FACOM 230-75 アレイプロセッサシステム, *FUJITSU*, Vol. 29, No. 1, pp. 93-128 (1978).
- 10) 富士通: AP-FORTRAN 文法書 (1977).
- 11) Meijerink, J. A. and Van der Vorst, H. A.: Guidelines for the Usage of Incomplete Decompositions in Solving Sets of Linear Equations as They Occur in Practical Problems, *J. Comput. Phys.*, Vol. 44, No. 1, pp. 134-155 (1981).
- 12) Ishiguro, M. and Koshi, Y.: Vectorization for Solving the Neutron Diffusion Equation—Some Numerical Experiments, *Nucl. Sci. Eng.*, Vol. 80, No. 2, pp. 322-328 (1982).
- 13) Varga, R. S. (渋谷訳): 大型行列の反復解法, pp. 25-117, サイエンス社, 東京 (1972).
- 14) 石黒美佐子, 原田裕夫他: 線形 3 重対角方程式の並列計算, JAERI-M 9703 (1981).
- 15) 難波克光: あるパイプライン方式ベクトル計算機による五重対角連立方程式の数値実験, 筑波大学修士論文 (1982.3).

(昭和 57 年 2 月 24 日受付)

(昭和 57 年 6 月 15 日採録)