

二段階陽的有限要素法プログラミングのベクトル化手法の検討

VECTORIZATION TECHNIQUE FOR PROGRAM USING TWO-STEP EXPLICIT F.E.M CODE

○畑中 勝守^{*} 川原 睦人^{**} 梅津 剛^{***}

By Katsumori HATANAKA and Mutsuto KAWAHARA and Tsuyoshi UMETSU

In recent years, a plenty of supercomputers are builded and world widely used among scientists and engineers so that the processing speed can be considerably increased. This paper describes a new vectorizing technique for the Two-Step Explicit Finite Elements applied to the flow analysis. In this method, the coefficients are created element-wise and then the results such as velocity and water-elevation are piled up according to the compatibility condition. However, such piling up statements of the program will be an undesirable recursive sentence in vectorization. This is why the processing speed is not so fast even though large part of other sentences were vectorized except such sentence. In order to evade this recursive sentence, a new condition has been proposed for vectorization, that is referred to as the "mapping condition". Tuning the program by the improvement of using the mapping condition, the vector-ratio has been improved up to 99.9%.

1. はじめに

スーパーコンピュータの実用化に伴い、ベクトル処理向きのプログラム開発が盛んになっており、高い並列性を持ったアルゴリズムの重要性が高まって来ている。そこで本論では、自然水域の流れ解析に広く応用されている、二段階陽的有限要素法による浅水長波解析をとりあげ、そのスーパーコンピュータ上でのベクトル処理向きアルゴリズムの検討について報告する。

一般に、従来の汎用コンピュータを対象として作成されたプログラムをそのままベクトル化しても効果を得ることは少ないと言われている。これは、ベクトル処理がその性格上、プログラムコーディング上で多くの制約をもっているためで、数値計算用プログラムの様に複雑なループを持ったプログラムには、ほとんどベクトル化されないものもある。従って、スーパーコンピュータにはベクトル処理を目的としたプログラムが必要となってくるのである。さて、本論で検討する二段階陽的有限要素法プログラムは、並列処理可能なループをたくさん含んでおり、簡単なプログラムのチューニングでベクトル化率を90%近くにまで上昇させることができる。ところが、演算回数の最も多い、流速や水位変動量といった節点量を計算するループが部分的にしかベクトル化されないため、90%ちかいベクトル化率でありながら、計算速度がそれほど速くならない。この原因は、二段階陽的有限要素法のプログラムでは、この部分で各要素ごとに計算された節点量を全体系で重ね合せるといった計算をしているが、この重ね合せの部分がベクトル化の対象とならない回

^{*} 中央大学技術員 理工学部土木工学科

(〒112 東京都文京区春日1-13-27)

^{**} 工博 中央大学教授 理工学部土木工学科^{***} 中央大学大学院 理工学部土木工学科

☎ 03(817)1815

帰演算（ベクトル処理では、recurrenceと呼ばれている）になっているためである。従って、回帰演算を回避してこの部分を完全ベクトル化し並列性の高いアルゴリズムを完成することが、全体として二段階陽的有限要素法のプログラムのベクトル化には不可欠である。筆者らは、この重ね合せのループについてのアルゴリズムの改造を中心に、ベクトル化手法の検討を行ったので、ここに報告する。

2. 解析モデルについて

基礎方程式として、次の浅水長波方程式を用いる。

<運動方程式>

$$\frac{\partial u}{\partial t} + g \frac{\partial \eta}{\partial x} = 0 \quad (1)$$

$$\frac{\partial v}{\partial t} + g \frac{\partial \eta}{\partial y} = 0 \quad (2)$$

<連続方程式>

$$\frac{\partial \eta}{\partial t} + h \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 \quad (3)$$

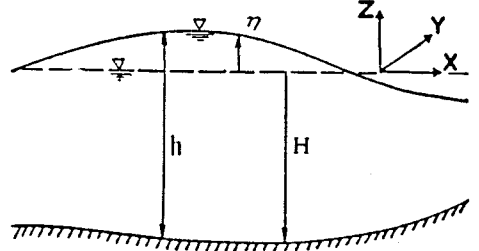


図-1. 座標系

ここに $h = H + \eta$ で、 u, v は流速、 η は水位上昇量、 H は水深、 g は重力加速度である。さて式(1)、(2)、(3)に重付残差法を適用し、空間方向には三角形一次の補間関数によるガラーキン法を、時間方向には二段階陽的有限要素法をそれぞれ用いて離散化を施すと、最終的に、次の様な陽的スキームが導かれる。すなわち、

First step:

$$\bar{M}_{\alpha\beta} u_{\beta}^{n+1/2} = \tilde{M}_{\alpha\beta} u_{\beta}^n - \frac{\Delta t}{2} \left(-g H_{\alpha\beta}^x \eta_{\beta}^n + g B_{\alpha\gamma}^x W_{\gamma}^n \right) \quad (4)$$

$$\bar{M}_{\alpha\beta} v_{\beta}^{n+1/2} = \tilde{M}_{\alpha\beta} v_{\beta}^n - \frac{\Delta t}{2} \left(-g H_{\alpha\beta}^y \eta_{\beta}^n + g B_{\alpha\gamma}^y W_{\gamma}^n \right) \quad (5)$$

$$\bar{M}_{\alpha\beta} \eta_{\beta}^{n+1/2} = \tilde{M}_{\alpha\beta} \eta_{\beta}^n - \frac{\Delta t}{2} \left(-h H_{\alpha\beta}^x u_{\beta}^n - h H_{\alpha\beta}^y v_{\beta}^n + h B_{\alpha\gamma}^{\eta} W_{\gamma}^n \right) \quad (6)$$

Second step:

$$\bar{M}_{\alpha\beta} u_{\beta}^{n+1} = \tilde{M}_{\alpha\beta} u_{\beta}^n - \Delta t \left(-g H_{\alpha\beta}^x \eta_{\beta}^{n+1/2} + g B_{\alpha\gamma}^x W_{\gamma}^{n+1/2} \right) \quad (7)$$

$$\bar{M}_{\alpha\beta} v_{\beta}^{n+1} = \tilde{M}_{\alpha\beta} v_{\beta}^n - \Delta t \left(-g H_{\alpha\beta}^y \eta_{\beta}^{n+1/2} + g B_{\alpha\gamma}^y W_{\gamma}^{n+1/2} \right) \quad (8)$$

$$\bar{M}_{\alpha\beta} \eta_{\beta}^{n+1} = \tilde{M}_{\alpha\beta} \eta_{\beta}^n - \Delta t \left(-h H_{\alpha\beta}^x u_{\beta}^{n+1/2} - h H_{\alpha\beta}^y v_{\beta}^{n+1/2} + h B_{\alpha\gamma}^{\eta} W_{\gamma}^{n+1/2} \right) \quad (9)$$

ここに、 $M_{\alpha\beta}$ は質量行列で、 $\bar{M}_{\alpha\beta}$ は $M_{\alpha\beta}$ の集中化行列、 $\tilde{M}_{\alpha\beta}$ は配分行列であり、

$$\tilde{M}_{\alpha\beta} = (1 - e) M_{\alpha\beta} + e \bar{M}_{\alpha\beta} \quad (0 \leq e \leq 1) \quad (10)$$

なる関係がある。また、 e は混合比 (Selective Lumping Parameter) である。

3. 計算のアルゴリズム

3.1 従来のアルゴリズム

2節で示したスキームをプログラムにした場合のフローチャートを図-2に示す。ここで、演算回数の最も多いルーチンのFirst step, Second step の内、First stepに注目してみる。このルーチンでは、剛性行列を各要素ごとに作成し、求められた各節点量を適合条件に従い、全体系で自動的に重ね合せている。この部分のフローチャートを図-3に、プログラム例を図-4に示す。

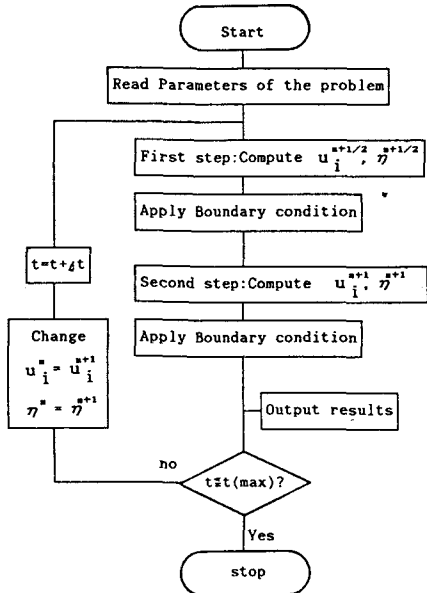


図-2. 二段階的有限要素法によるプログラムのフローチャート

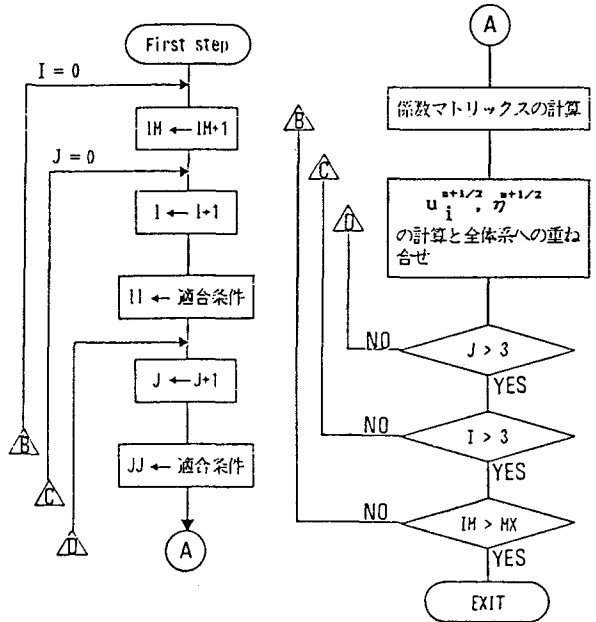


図-3. First-stepのフローチャート

```

S      DO 2000 IM=1,MX
S      DO 100 I=1,3
S      II=IE(I,IM)
      DO 100 J=1,3
      JJ=IE(J,IM)
      IF (I.EQ.J) THEN
      RM=RMI(I,IM)
      ELSE
      RM=RMI(J,IM)
      END IF
      UUH(II)=UUH(II)+RM*UU1(JJ)
      &      -DT2*(-HX(I,IM))*ET1(JJ)*GG
      VVH(II)=VVH(II)+RM*VV1(JJ)
      &      -DT2*(-HY(I,IM))*ET1(JJ)*GG
      ETH(II)=ETH(II)+RM*ET1(JJ)
      &      -DT2*(-HX(I,IM)*UU1(JJ)-HY(I,IM)*VV1(JJ))*H3(IM)
100 CONTINUE

```

図-4. First-stepのプログラム例

このアルゴリズムは、従来の汎用コンピュータ上のスカラー処理で、できるだけ高速に計算するように作成されたもので、演算回数が節約できるよう工夫されている。さてここで、図-4のプログラムの棒①に注目してみると、ここが回帰演算になっており、スーパーコンピュータでは全くベクトル化されることが分かる。また、この計算部分で全計算の8割以上を占める計算を行っているため、先に述べたように、この部分に改造を施し、アルゴリズムの並列性を高める必要があるのである。

3. 2 改良されたアルゴリズム

まず、新たな条件として、適合条件と相対する条件（以下、マッピングと呼ぶ）を作成し、これを用いて計算することを検討した。この条件は、各節点がどの要素の節点として結合しているかを判断し、これをアレイに格納する。そしてこのアレイを、要素ごとに求められた節点量を全体系で重ね合わせる際の添字として利用しようというものである。そこで、簡単な例として図-5の様な三角形三節点の有限要素分割に対するマッピングのアレイを考えてみる。まず各節点に対して、8個のアレイを用意する。さて、図-5の第5節点に着目すると、この節点が4つの要素に関わっている事が分る。従って、第5節点のアレイの数は4になり、残りのアレイには、ダミーをインプットする。他の節点に対しても同様に考えれば、表-2に示す条件が得られる。この条件を利用して図-3のアルゴリズムを改良したものが、図-6に示すフローチャートである。そして、図-7がこれをもとに作成されたプログラム例である。マッピングの条件は図-7の枠②の部分で使用されており、全てのループがベクトル化されている。以上のように、本手法によるプログラムの改良により、重ね合せの部分の回帰演算を回避することが可能となった。

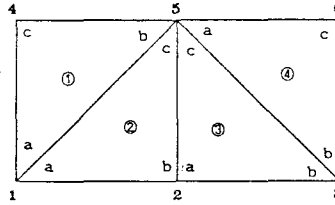


図-5. 有限要素分割図

	a	b	c
①	1	5	4
②	1	2	5
③	2	3	5
④	5	3	6

表-1. 適合条件

	I	II	III	IV	...	X
1	(①, a)	(②, a)	(*, *)	(*, *)	...	(*, *)
2	(②, b)	(③, a)	(*, *)	(*, *)	...	(*, *)
3	(③, b)	(④, b)	(*, *)	(*, *)	...	(*, *)
4	(①, c)	(*, *)	(*, *)	(*, *)	...	(*, *)
5	(①, b)	(②, c)	(③, c)	(④, a)	...	(*, *)
6	(④, c)	(*, *)	(*, *)	(*, *)	...	(*, *)

(*, *): Dummy

表-2. マッピングの条件

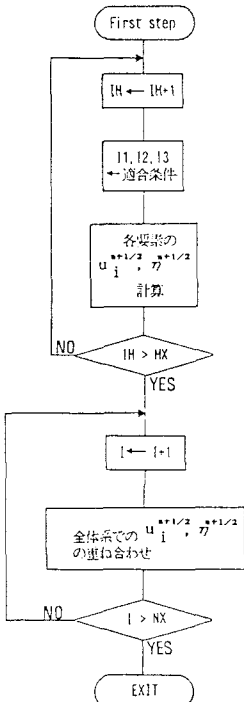


図-6. 改良されたFirst-stepのフローチャート

```

V      DO 2000 IM=1,MX
V      I1=IE(1,IM)
V      I2=IE(2,IM)
V      I3=IE(3,IM)
C
V      UUU(IM,1) = RMII(IM) * UU1(I1)
&      -DT2*(-HX(1,IM))*ET1(I1)*GG
&      + RMIJ(IM) * UU1(I2)
&      -DT2*(-HX(2,IM))*ET1(I2)*GG
&      + RMIJ(IM) * UU1(I3)
&      -DT2*(-HX(3,IM))*ET1(I3)*GG
V
V      EEE(IM,3) = RMIJ(IM) * ET1(I1)
&      -DT2*(-HX(1,IM))*UU1(I1)-HY(1,IM)*VV1(I1)*H3(IM)
&      + RMIJ(IM) * ET1(I2)
&      -DT2*(-HX(2,IM))*UU1(I2)-HY(2,IM)*VV1(I2)*H3(IM)
&      + RMII(IM) * ET1(I3)
&      -DT2*(-HX(3,IM))*UU1(I3)-HY(3,IM)*VV1(I3)*H3(IM)
C
V 2000 CONTINUE
C
V      DO 100 I=1,NX
V      UUH(I) = UUU(KE(I,1),LE(I,1)) + UUU(KE(I,2),LE(I,2))
&      + UUU(KE(I,3),LE(I,3)) + UUU(KE(I,4),LE(I,4))
&      + UUU(KE(I,5),LE(I,5)) + UUU(KE(I,6),LE(I,6))
&      + UUU(KE(I,7),LE(I,7)) + UUU(KE(I,8),LE(I,8))
V
V      ETH(1) = EEE(KE(I,1),LE(I,1)) + EEE(KE(I,2),LE(I,2))
&      + EEE(KE(I,3),LE(I,3)) + EEE(KE(I,4),LE(I,4))
&      + EEE(KE(I,5),LE(I,5)) + EEE(KE(I,6),LE(I,6))
&      + EEE(KE(I,7),LE(I,7)) + EEE(KE(I,8),LE(I,8))
V 100 CONTINUE

```

図-7. 改良されたFirst-stepのプログラム例

4. 解析例

解析例として、比較的小規模な有限要素分割の解析（解析例 a）を 3 台のスーパーコンピュータ（FACOM VP-30、HITAC S-810/20、HITAC S-820/80）を用いて、また、大規模計算の一例として、総要素数 3 万の有限要素分割による津波解析（解析例 b）を FACOM VP-30 を用いて、それぞれ行った。以下に、各々の解析での計算時間、ベクトル化率、スカラー処理とベクトル処理の速度比等の実測結果を示す。

<解析例 a>

東京湾の浅水長波流れ解析・・・総要素数 = 1,387、総節点数 = 770、総計算回数 = 2,000

プログラム	機種名	ベクトル化率
改良前	VP-30	94.7%
	S-810	89.2%
	S-820	89.3%
改良後	VP-30	99.9%
	S-810	99.9%
	S-820	99.9%

表-3. 解析例 a, ベクトル化率

プログラム	機種名	計 算 時 間 sec.		対スカラー 速度比
		スカラー処理	ベクトル処理	
改良前	VP-30	214.0	53.0	4.0
	S-810	131.0	35.4	3.7
	S-820	59.0	12.2	4.8
改良後	VP-30	260.0	26.5	9.8
	S-810	160.0	23.1	6.9
	S-820	63.0	7.5	8.4

表-4. 解析例 a, 実測結果

改良前のベクトル化率が 90% 前後であるのに対して、本手法による改良後のプログラムのベクトル化率は、3 台のスーパーコンピュータとも、99.9% と非常に高いものとなっている。

※ VP-30: FACOM VP-30
S-810: HITAC S-810/20
S-820: HITAC S-820/80

一方、スカラー処理とベクトル処理の計算速度比は、改良前のプログラムが 4 倍前後であるのに対し、改良後のプログラムは、7~10 倍となり、ほぼ満足のいく高速化が得られた。

<解析例 b>

十勝沖地震の津波伝播解析・・・総要素数 = 32,382、総節点数 = 16,544

プログラム	ベクトル化率
オリジナル	0.4%
改良 1	88.9%
改良 2	99.9%

表-5. 解析例 b, ベクトル化率

プログラム	処理 モード	計 算 時 間 sec.			1000step tv/t (%)	1000step 対スカラー 速度比
		100step	300step	1000step		
オリジナル	SCALAR	227.87	659.23	2171.69	33.5%	5.17 倍
改良 1	SCALAR	208.13	597.92	1954.92		
	VECTOR	53.61	125.36	378.44	88.2%	13.18 倍
改良 2	SCALAR	261.55	745.79	2452.74		
	VECTOR	39.81	71.70	186.16		

※ 改良 1: インライン展開のみ
改良 2: 本手法による改良

表-6. 解析例 b, 実測結果

※ $tv/t = \frac{VPU-TIME}{CPU-TIME}$

この解析では、FACOM VP-30 を使用し、オリジナルのプログラムを改良して行く過程でのベクトル化の効率の変化を示す。まず、プログラムを原形のままベクトル化したところ、ベクトル化率は0.4% (表-5) と、非常に低い数値であった。そこで、重ね合わせのループをインライン展開し (改良1) ベクトル化率を計ると88.9%とかなり高い数値にすることができた。しかし、表-6のように、計算速度はさほど速くなかった。次に、本手法によるアルゴリズムを用いた改良 (改良2) では、ベクトル化率が99.9%と解析例aと同様に非常に高いベクトル化率が得られた。また、ベクトル化率があがったせいで計算速度も向上し、1000stepの計算では対スカラー処理との速度比で13.18倍、全CPU-TIMEに占めるVPU-TIMEの割合いでは、88.2%という結果が得られた。また図-8から分かるように、改良2によるプログラムのベクトル化不可能領域は、改良1によるプログラムに比べて大幅に減少させることができた。

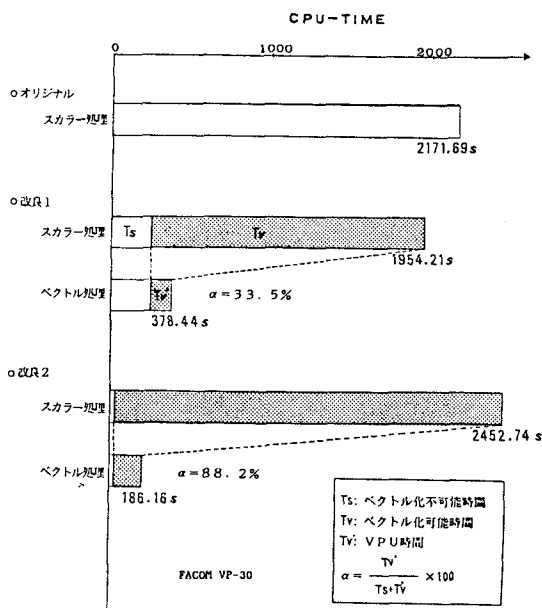


図-8. 解析例b, 1000stepのときのスカラー時間とベクトル時間

5. おわりに

二段階陽的有限要素法プログラミングのベクトル化手法について検討を行った。解析例の実測結果から、本手法によるアルゴリズムが、ベクトル処理向きの高い並列性を持ったアルゴリズムであることが確認でき、満足のいく計算速度の向上が得られた。また、スーパーコンピュータは機種ごとの性格の違いが大きく、機種の性格に合ったコーディングをする必要があると言われているが、本手法によるアルゴリズムを用いて作成したプログラムは、特別なコーディングのテクニックを用いた訳ではなく、機種ごとの性格の違いも考慮していないにもかかわらず、3台のスーパーコンピュータで十分な高速化が得られた。このことから、本手法のアルゴリズムが二段階陽的有限要素法プログラムのベクトル化に有効であり、計算機の性格にさほど左右されないものであることが分かる。今後は、この手法が他の有限要素法プログラムに対しても適用可能と思われるため、これについての検討を行っていきたいと考えている。

参考文献

- 1) 川原睦人 : 有限要素法流体解析、日科技連出版、1984
- 2) 村田健郎 他 : スーパーコンピュータ/科学技術計算への適用、丸善株式会社、1985
- 3) 日本物理学会 : スーパーコンピュータ、培風館、1985
- 4) M.Kinoshita and M.Kawahara : Two-step Explicit Finite Element Method for Tsunami Analysis Considering Open Boundary Condition、NUMETA87、Vol.2-135、1987

(1988年1月22日受付)