

JAERI-M
86-019

軽水炉安全性解析コードRELAP5／
MOD2／CYCLE36のベクトル化

1986年2月

篠沢 尚久*・近藤 一也*・和田 善之**
石黒美佐子

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-M レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合せは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）
あて、お申しこしください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department
of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun,
Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1986

編集兼発行 日本原子力研究所
印 刷 日立高速印刷株式会社

軽水炉安全性解析コード RELAP 5 / MOD 2 / CYCLE 36 のベクトル化

日本原子力研究所東海研究所計算センター

篠沢 尚久^{*}・近藤 一也^{*}・和田 善之^{**}・石黒美佐子

(1986 年 1 月 30 日受理)

軽水炉安全性解析コード RELAP 5 / MOD 2 / CYCLE 36 をベクトル化した。その結果実用規模の過渡計算に対し、ベクトル計算機 VP - 100において、ベクトル化前のコードのスカラ－計算に比べ 3.5 倍程度の性能向上比を得た。ベクトル化においては、熱計算部分は、熱構造体及び熱メッシュの並列性を利用し、流体計算部分は、ボリューム及びジャンクションに関する並列性を利用した。本報告では、各サブルーチンごとのベクトル化の方法、ベクトル化によって得られた性能向上比、及び CDC 版から FACOM 版への変換によって必要となったビット処理関数のオーバーヘッドの減少方法について述べる。

東海研究所：茨城県那珂郡東海村白方字白根 2 の 4

*) 外来研究員（富士通）

**) 原子力データセンター

Vectorization of the LWR Transient Analysis Code
RELAP5/MOD2/CYCLE36

Nachisa SHINOZAWA*, Kazuya KONDOU*,
Yoshiyuki WADA** and Misako ISHIGURO
Computing Center, Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received January 30, 1986)

The LWR transient analysis code RELAP5/MOD2/CYCLE36 has been vectorized. The performance of the vectorized code in vector mode of the FACOM VP-100 was 3.5 times higher than that of original one in scalar mode for a real scale transient calculation. Subroutines for heat conduction were vectorized in terms of heat structures and heat meshes, while those for hydrodynamics were vectorized in terms of volumes and junctions. In this report, the vectorization method used for each of subroutines and its vectorization effect are described. A reduction of the overhead of bit operation functions which is introduced when the code was converted from the CDC to the FACOM is also described.

Keywords: RELAP5, Computer Code, Vectorization, Supercomputer, VP-100,
LWR LOCA

*) on leave from FUJITSU Ltd.

**) Nuclear Energy Data Center

目 次

1.はじめに	1
2.RELAP 5 / MOD 2 コードの概要	3
3.サンプルデータ	6
4.ビット操作ファンクションの改良	10
5.動的挙動解析	15
6.コードのベクトル化	25
6.1 基本的なベクトル化の方法	25
6.2 热伝導計算部分のベクトル化	25
6.3 流体力学計算部分のベクトル化	43
6.4 状態計算部分のベクトル化	66
6.5 その他のサブルーチン	76
7.ベクトル化による性能向上	78
8.ベクトル計算機による結果の妥当性	81
9.ベクトル化によるメモリーの増加	83
10.おわりに	85
謝辞	85
参考文献	86

CONTENTS

1. Introduction	1
2. Summary of the RELAP5/MOD2 code	3
3. Sample data	6
4. Reduction of bit operation functions	10
5. Dynamic behavior analysis for vectorization	15
6. Vectorization of the code	25
6.1 Basic strategy for vectorization	25
6.2 Vectorization of heat conduction subroutines	25
6.3 Vectorization of hydrodynamic subroutines	43
6.4 Vectorization of state relationship subroutines	66
6.5 Other subroutines	76
7. Performance improvement by vectorization	78
8. Numerical validity for the vector version	81
9. Increment of memory size by vectorization	83
10. Conclusion	85
Acknowledgements	85
References	86

1. はじめに

RELAP 5／MOD 2 コード¹⁾は、加圧水型原子炉（PWR）システムの過渡解析のためのコードであり、大破断時の冷却材喪失事故、プラント制御、燃料系の事故など、軽水炉（LWR）安全性に関する各種の PWR システムのシミュレーションに使用される。

RELAP 5 は、アイダホ国立研究所（INEL）の CDC CYBER - 176 計算機によって開発されたコードであり、MOD 0 から始まり、MOD 1, MOD 2 と順次その機能が拡張されてきている。それに伴ってコードも巨大化し、現在、実行ステートメント数約 60,000 ステップ、サブルーチン数 258 となっている。また、その処理時間も非常に長くかかるようになっている。

日本原子力研究所（原研）では、従来から RELAP 5 コード（CDC 版）を FACOM M シリーズ計算機で使用できるように変換し、これを原研版として利用してきた。そして、ベクトル計算機 FACOM VP - 100 の導入に際して RELAP 5／MOD 1 のベクトル化版が作成された。このベクトル化版は、ベクトル化前のスカラー版に比べて約 2.7 倍処理性能が向上した²⁾。原研では、VP - 100 の 2 台目が導入され、長時間計算はそのほとんどをベクトル計算機で処理する方針となっている。最近、RELAP 5 の利用が MOD 1 から MOD 2 に順次移行しているため、MOD 2 のベクトル化版も作成した。MOD 2 のベクトル化版は、ベクトル化前のスカラー版（オリジナル版）と比較して 3.0 から 3.5 倍の処理性能の向上を得た。

ベクトル化作業では、ベクトル化版コードをスカラー・モードで実行した結果と、ベクトル化前のコードの実行結果が完全に一致することを前提に進めた。その手順を以下に示す。

- (1) ビット操作ファンクションの改良
- (2) FORTUNE による挙動解析
- (3) 各サブルーチンのベクトル化

ビット操作ファンクションの改良、及びベクトル化による性能向上比の概要を Fig. 1.1 に示す。

本報告では、第 2 章で RELAP 5／MOD 2 の概要を示し、第 3 章でテストランに使用したサンプルデータの内容を示す。第 4 章から第 6 章は、ベクトル化作業の手順に従ってそれぞれ、ビット操作ファンクションの改良、FORTUNE による挙動解析、各サブルーチンのベクトル化について述べる。第 7 章は、ベクトル化によって得られた性能向上について、第 8 章は、結果の妥当性について、第 9 章は、ベクトル化によって増加したメモリサイズについて述べる。

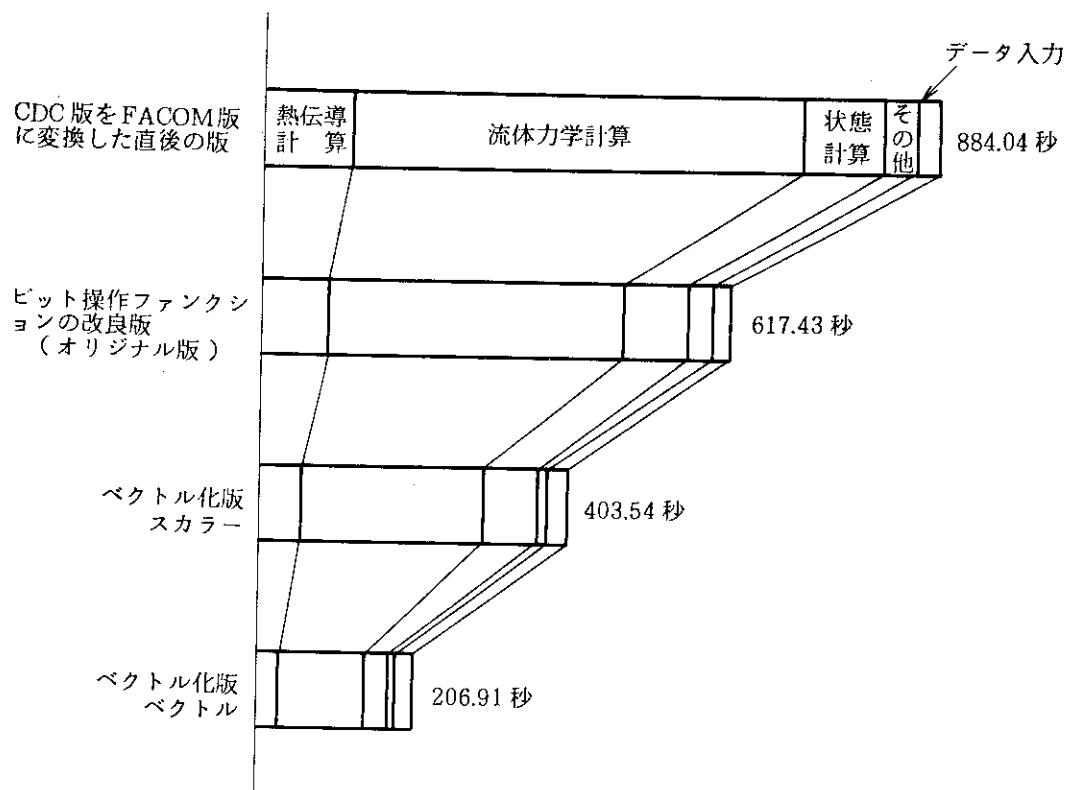


Fig. 1.1 Vectorization effect on the reduction of the cputimes for TYPPWR2 data

2. RELAP 5 / MOD 2 コードの概要

RELAP 5 / MOD 2 コードは、一次元二相流モデルに基づいて開発されたコードであり、気・液二相の他に非凝縮性ガスを含んだコンポーネントも扱うことが出来るようになっている。二流体非平衡モデルの基本方程式は、2つの質量保存の式、2つの運動量の式、及び2つのエネルギーの式の合計6個の式で表わされる。MOD 1との大きな違いは、エネルギーについて、MOD 1では気相、液相の混合体を考えていたのが、MOD 2では気相、液相を別々に扱っていることである。これに伴って変数が7個に増えている。MOD 2で使用している変数は、圧力(P)、速度(v_g, v_f)、内部エネルギー(U_g, U_f)、ボイド率(α_g)、と非凝縮性ガス質量比(X_n)である。このうちX_nは、定義がMOD 1と変わっており、非凝縮性ガスの質量と全気体質量の比である。他の主な変更点は、相間抗力の式の改良、新しい非平衡壁熱モデル、壁摩擦分割モデルの改良、蒸気発生モデルの改良、新しいコンポーネントの種類の追加などである。

2.1 基本方程式

RELAP 5 / MOD 2で使用している基本方程式は次の6個である。

質量保存の式：

$$\frac{\partial}{\partial t} (\alpha_g \rho_g) + \frac{1}{A} \frac{\partial}{\partial x} (\alpha_g \rho_g v_g A) = \Gamma_g \quad (2.1)$$

$$\frac{\partial}{\partial t} (\alpha_f \rho_f) + \frac{1}{A} \frac{\partial}{\partial x} (\alpha_f \rho_f v_f A) = -\Gamma_g \quad (2.2)$$

運動量の式：

$$\begin{aligned} & \alpha_g \rho_g A \frac{\partial}{\partial t} v_g + \frac{1}{2} \alpha_g \rho_g A \frac{\partial}{\partial x} v_g^2 \\ &= -\alpha_g A \frac{\partial P}{\partial x} + \alpha_g \rho_g B_x A - (\alpha_g \rho_g A) F_{WG} (v_g) + \Gamma_g A (v_{gI} - v_g) \\ & - (\alpha_g \rho_g A) F_{IG} (v_g - v_f) - C \alpha_g \alpha_f \rho A \frac{\partial}{\partial t} (v_g - v_f). \quad (2.3) \end{aligned}$$

$$\begin{aligned}
& \alpha_f \rho_f A \frac{\partial}{\partial t} v_f + \frac{1}{2} \alpha_f \rho_f A \frac{\partial}{\partial x} v_f^2 \\
& = -\alpha_f A \frac{\partial P}{\partial x} + \alpha_f \rho_f B_x A - (\alpha_f \rho_f A) FWF(v_f) - \Gamma_g A (v_{fI} - v_f) \\
& \quad - (\alpha_f \rho_f A) FIG(v_f - v_g) - C \alpha_f \alpha_g \rho A \frac{\partial}{\partial t} (v_f - v_g). \quad (2.4)
\end{aligned}$$

エネルギーの式

$$\begin{aligned}
& \frac{\partial}{\partial t} (\alpha_g \rho_g U_g) + \frac{1}{A} \frac{\partial}{\partial x} (\alpha_g \rho_g U_g v_g A) = -P \frac{\partial}{\partial t} \alpha_g \\
& - \frac{P}{A} \frac{\partial}{\partial x} (\alpha_g v_g A) + Q_{wg} + Q_{ig} + \Gamma_{ig} h_g^* + \Gamma_w h_f^* + DISS_g. \quad (2.5)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} (\alpha_f \rho_f U_f) + \frac{1}{A} \frac{\partial}{\partial x} (\alpha_f \rho_f U_f v_f A) = -P \frac{\partial}{\partial t} \alpha_f \\
& - \frac{P}{A} \frac{\partial}{\partial x} (\alpha_f v_f A) + Q_{wf} + Q_{if} - \Gamma_{ig} h_f^* - \Gamma_w h_f^* + DISS_f. \quad (2.6)
\end{aligned}$$

ここで、

A : 流路断面積

$\alpha_f = 1 - \alpha_g$

$\Gamma_g = -\Gamma_f$

B_x : x 軸方向の体積力

FWG, FWF : 蒸気及び液体に対する壁摩擦係数

FIG, FIF : 蒸気及び液体に対する二相間摩擦係数

C : 仮想質量係数

$\rho = \alpha_g \rho_g + \alpha_f \rho_f$

Q_{wg}, Q_{wf} : 蒸気及び液体に対する単位体積当りの壁熱伝導量

Q_{ig}, Q_{if} : 相間熱伝達項

$\Gamma_g = \Gamma_{ig} + \Gamma_w$

$DISS_g = \alpha_g \rho_g FWF v_g^2$: 気相におけるエネルギー消失項

$DISS_f = \alpha_f \rho_f FWF v_f^2$: 液相におけるエネルギー消失項

$DISS = DISS_g + DISS_f$

である。方程式中の添字 I は二相境界を意味している。

(2.3), (2.4) 式の仮想質量項について、MOD 1 のときには入っていた空間微分の項が消えて、非常に簡単になっている。この理由は、空間のノード付けが粗いため、微分が不正確なためである。

非凝縮性ガスを含むコンポーネントの場合、上記 6 個の式に他に次式が追加される。

$$\frac{\partial}{\partial t} (\alpha_g \rho_g X_n) + \frac{1}{A} \frac{\partial}{\partial x} (\alpha_g \rho_g X_n v_g A) = 0 . \quad (2.7)$$

また、(2.3)～(2.6)式の気相の各特性値は、蒸気と非凝縮性ガスの混合物に替えられる。

最終的には、これらを差分化した式について、他の変数を消去し、圧力のみの線形方程式として解いている。詳細は文献〔1〕を参照されたい。

3. サンプルデータ

ベクトル化の際使用したデータは、原研版へのコンバージョン時のテ스트ランで使用した PROB 2, TYPPWR 2 の 2 ケース及び、ROSA - IV 実験解析で実際に使用される LSTF データである。以下にその内容を示す。

(1) PROB 2

このデータは、RELAP 5 / MOD 1 のワークショップで使用されたもので一次系、二次系のループを持つ PWR を模擬したもので、ボリューム数 30、ジャンクション数 31、熱構造体数 6 である。Fig. 3.1 にノードダイヤグラムを示す。

(2) TYPPWR 2

これは、4 ループ（健全側は、3 ループをまとめて 1 つのループにしている。）の PWR プラント (ZION plant) の 2% (4 inch) コールドレグ破断を模擬したもので、ボリューム数 139、ジャンクション数 142、熱構造体数 83 である。Fig. 3.2 にノードダイヤグラムを示す。主な挙動は次に示す通りである。

- ① トランジエント開始後すぐに (0.01 秒後) 破断する。
- ② 加圧器 (コンポーネント 150 - ①) の圧力が 12.82 MPa 以下になるとスクラム (停止) 信号が発生する。
- ③ スクラム信号の 3.4 秒後にスクラムが開始する。
- ④ スクラム信号と同時に主循環ポンプの電源が止まる。(ポンプは慣性で回り続ける。)
- ⑤ スクラム信号と同時に S.G. フィードウォーター及びスチームラインのバルブが閉じ始める。フィードウォーターのバルブは 10 秒、スチームラインのバルブは 1 秒かかって線形に閉じる。
- ⑥ 加圧器の圧力が 12.6 MPa 以下になると、Safety Injection (S.I.) 信号が発生する。
- ⑦ S.I. 信号発生の 5 秒後に S.I. 及び charging が開始する。
- ⑧ スクラム信号発生の 14 秒後に補助給水が開始する。以下、S.G. のダウンカマー (コンポーネント 172) の水位に伴って自動的に on / off を繰り返す。

(3) LSTF 実験データ

これは、ROSA IV LSTF (3420 MW PWR の 1 / 48 モデル) によるコールドレグ 5% 破断実験を模擬したもので、ボリューム数 170、ジャンクション数 178、熱構造体数 166 となっており、Fig. 3.3 にそのノードダイヤグラムを示す。主な計算事象は次のとおりである。

- ① トランジエント開始直後破断する。
- ② 加圧器の圧力が 12.97 MPa に下がると、スクラムが始まる。
- ③ 3 秒後に二次系蒸気バルブが閉じる。
- ④ 7 秒後に二次系主給水系が止まる。
- ⑤ 35 秒後、炉心出力が低下する。
- ⑥ 37 秒後、二次補助給水系が動作開始する。

- ⑦ 加圧器の圧力が 12.27 MPa 以下になると、Safety Injection 信号が発生する。
- ⑧ S.I. 信号の 17 秒後に charging が始まる。
- ⑨ 22 秒後、高圧注入系が冷却水の注入を開始する。
- ⑩ 破断後 266 秒でコーストダウンが完了する。
- ⑪ 加圧器の圧力が 4.51 MPa 以下になると、アクチュエータが動き出す。
- ⑫ 加圧器の圧力が 1.29 MPa 以下になると、低圧注入系が冷却水注入を開始する。

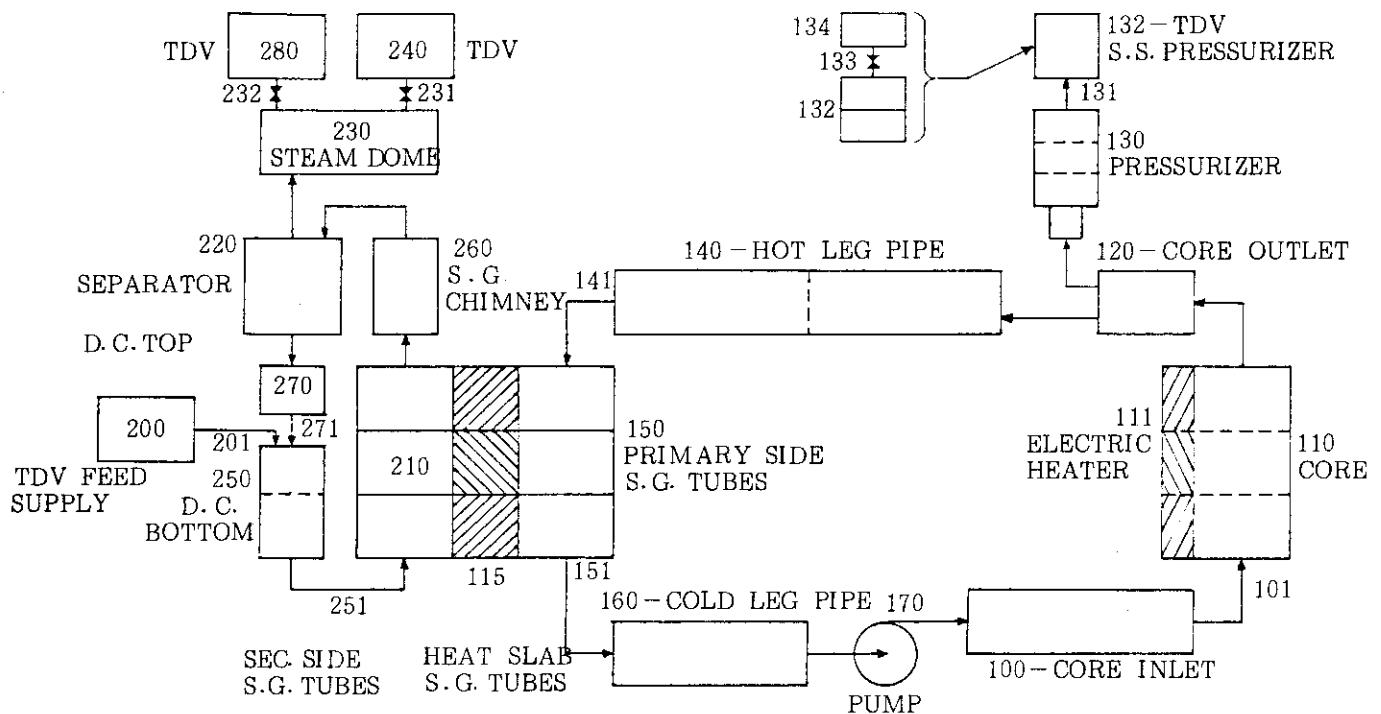


Fig. 3.1 PROB 2 noding diagram

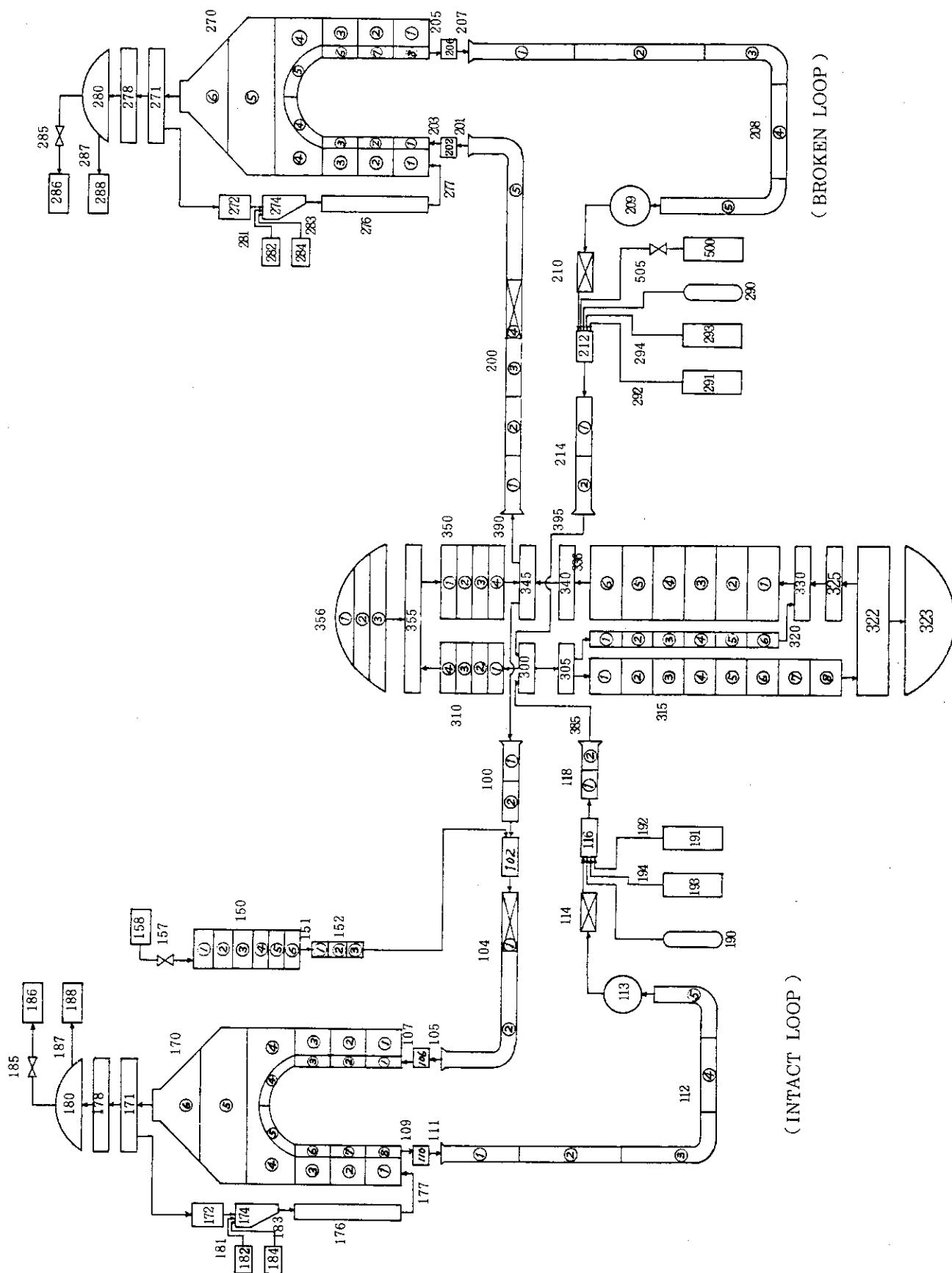


Fig. 3.2 Typical PWR model (TYPWPWR 2) nodding diagram

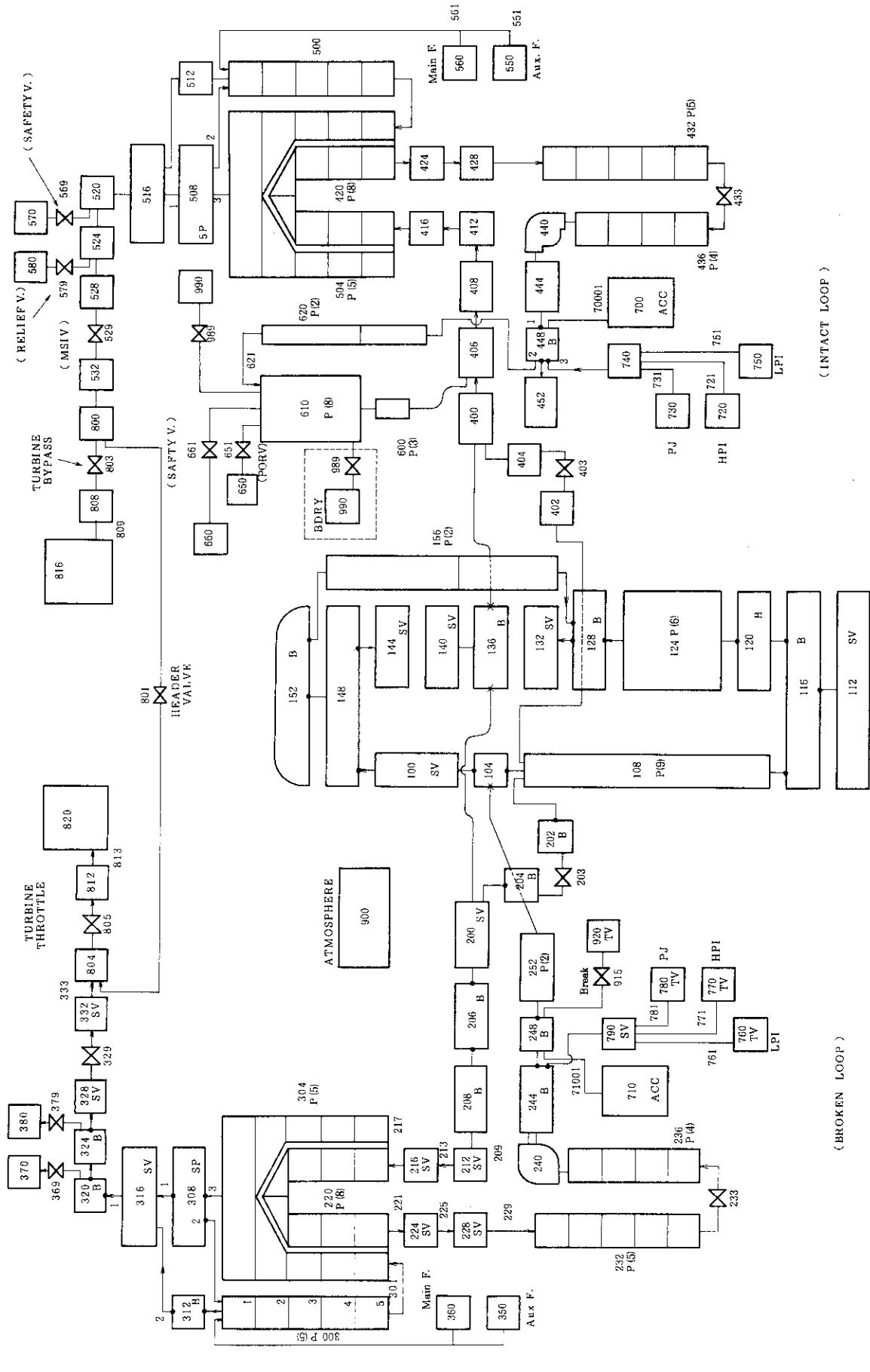


Fig. 3.3 LSTF noding diagram

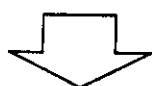
4. ビット操作ファンクションの改良

RELAP5/MOD2コードは、メモリーの節約のために1語60ビット中に複数のデータを詰込み(パック語)、ビット操作演算を用いて取出して使用している。CDC計算機では、これら詰込み、取出し操作は、論理演算子 .AND. , .OR. , .NOT. 及び、組込み関数SHIFT, MASK, XORを用いている。原研版では、同等の機能を持つファンクションを用意し、それらによって模擬している。また、60ビット整数演算についてもファンクション¥ADD, ¥SUBで対応している³⁾。このビット演算の変換は、ツールを用いて行われている。ビット操作ファンクションの実行回数は、Table 4.1に示すように非常に多く、CPU時間のかなりの部分を占めていると思われる。また、ビット操作ファンクションはある特定の組合せで使用される場合が多く、それらのほとんどはTable 4.2に示すように変更することによって、ファンクションの実行回数を減らすことができる。ここで、マスク演算については、Fig. 4.1に示すデータ文を用意し、これを用意するように変更する。

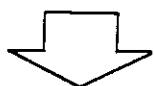
また、これらのパターンは単独で現われるだけでなく、複数組合わされている場合も多い。例3.1のような場合、複数の変更を組合せる事によりファンクションの実行回数を4分の1に減らすことができる。

(例3.1)

```
KP = ¥IDC( ¥SUB( ¥ADD( ¥AND( ¥NOT( ¥MASK(48)) ,
.      ¥SFT( IJ1(I),30)) , ¥DI(IXSOP)),¥B0001))
```



```
KP = ¥IDAND( ¥NOT( ¥MASK(48)) , ¥SFT( IJ1(I),30))
.      + IXSOP - ¥ID(¥B0001)
```



```
KP = ¥IDAND( ¥MKN48 , ¥SFT( IJ1(I),30)) + IXSOP - ¥I0001
```

Table 4.1 Execution times of bit operation functions

<i>function name</i>	<i>nooptimize version</i>	<i>ORIGINAL version</i>
¥ID*	46,204,290	1,281,608
¥18*	2,101,336	2,101,253
¥ADD	7,226,058	1,976,241
¥AND	55,087,783	10,164,932
¥DI	9,247,087	3,998,088
¥EQ	2,339,444	2,339,444
¥GE	3,615,160	3,615,160
¥GT	106,390	106,390
¥IDAND**		44,922,768
¥LE	665,370	665,370
¥LT	15,901,274	15,901,274
¥MASK	55,107,205	0
¥NE	271,502	271,502
¥NOT	52,381,621	425,134
¥OR	3,043,068	3,043,068
¥SFT	56,898,352	56,898,352
¥SFT8	0	0
¥SUB	2,689,966	330
¥XOR	40,069	40,069
<i>TOTAL</i>	312,925,975	147,750,983

* ... FORTRANによる関数

** ... ビット操作ファンクションの最適化のために新たに追加したファンクション

Table 4.2 Restructured method of bit operation functions

<i>nooptimize version</i>	<i>ORIGINAL version</i>
$\$ID(\$ADD(a,b))$	$\$ID(a) + \$ID(b)$
$\$ID(\$SUB(a,b))$	$\$ID(a) - \$ID(b)$
$\$ID(\$AND(a,b))$	$\$IDAND(a,b)$
$\$ID(\$DI(i))$	i
$\$NOT(\$MASK(n))$	$\$MKNn^*$
$\$MASK(n)$	$\$MKPn^*$

*) DATA 文を用いる。

以上のようなビット操作関数の改良を行なった結果、ファンクションの実行回数は、Table 4.1に示すように全体で 52.8 % 削減され、実行時間も Table 4.3 に示すように全体で約 1.4 倍の性能向上となっている。コンバージョンによるビット操作のオーバーヘッドが、かなり改善されたことになる。この改良は、変換ツールを用いてほとんど機械的に行なうことができる。今後ビット操作を含むプログラムのコンバージョンを行なう際には、ビット操作ファンクションの改良を事前に行なうべきである。

ベクトル化は、このビット操作ファンクションの改良されたソースに対して行ない、以後これをオリジナル版と呼ぶこととする。また、ビット操作ファンクションの改良がなされていない版を最適化前版と呼ぶ。

```

*INCLUDE ￥MASKD
DATA ￥MKP1 /ZF8000000000000000/ , ￥MKN1 /Z07FFFFFFF/FFFF/
DATA ￥MKP2 /ZFC00000000000000/ , ￥MKN2 /Z03FFFFFFF/FFFF/
DATA ￥MKP3 /ZFE00000000000000/ , ￥MKN3 /Z01FFFFFFF/FFFF/
DATA ￥MKP4 /ZFF00000000000000/ , ￥MKN4 /Z00FFFFFFF/FFFF/
DATA ￥MKP5 /ZFF80000000000000/ , ￥MKN5 /Z007FFFFFFF/FFFF/
DATA ￥MKP6 /ZFFC000000000000/ , ￥MKN6 /Z003FFFFFFF/FFFF/
DATA ￥MKP7 /ZFFE000000000000/ , ￥MKN7 /Z001FFFFFFF/FFFF/
DATA ￥MKP8 /ZFFF000000000000/ , ￥MKN8 /Z000FFFFFFF/FFFF/
DATA ￥MKP9 /ZFFF800000000000/ , ￥MKN9 /Z0007FFFFFFF/FFFF/
DATA ￥MKP10 /ZFFFC000000000000/ , ￥MKN10 /Z0003FFFFFFF/FFFF/
DATA ￥MKP11 /ZFFFE000000000000/ , ￥MKN11 /Z0001FFFFFFF/FFFF/
DATA ￥MKP12 /ZFFFF000000000000/ , ￥MKN12 /Z0000FFFFFFF/FFFF/
DATA ￥MKP13 /ZFFFF800000000000/ , ￥MKN13 /Z00007FFFFFFF/FFFF/
DATA ￥MKP14 /ZFFFFC000000000000/ , ￥MKN14 /Z00003FFFFFFF/FFFF/
DATA ￥MKP15 /ZFFFFE000000000000/ , ￥MKN15 /Z00001FFFFFFF/FFFF/
DATA ￥MKP16 /ZFFFFF000000000000/ , ￥MKN16 /Z00000FFFFFFF/FFFF/
DATA ￥MKP17 /ZFFFFF800000000000/ , ￥MKN17 /Z000007FFFFFFF/FFFF/
DATA ￥MKP18 /ZFFFFFC000000000000/ , ￥MKN18 /Z000003FFFFFFF/FFFF/
DATA ￥MKP19 /ZFFFFFE000000000000/ , ￥MKN19 /Z000001FFFFFFF/FFFF/
DATA ￥MKP20 /ZFFFFFF000000000000/ , ￥MKN20 /Z000000FFFFFFF/FFFF/
DATA ￥MKP21 /ZFFFFFF800000000000/ , ￥MKN21 /Z0000007FFFFFFF/FFFF/
DATA ￥MKP22 /ZFFFFFFC000000000000/ , ￥MKN22 /Z0000003FFFFFFF/FFFF/
DATA ￥MKP23 /ZFFFFFE000000000000/ , ￥MKN23 /Z0000001FFFFFFF/FFFF/
DATA ￥MKP24 /ZFFFFFFF000000000000/ , ￥MKN24 /Z0000000FFFFFFF/FFFF/
DATA ￥MKP25 /ZFFFFFFF800000000000/ , ￥MKN25 /Z00000007FFFFFFF/FFFF/
DATA ￥MKP26 /ZFFFFFFF000000000000/ , ￥MKN26 /Z00000003FFFFFFF/FFFF/
DATA ￥MKP27 /ZFFFFFFFFFF0000000000/ , ￥MKN27 /Z00000001FFFFFFF/FFFF/
DATA ￥MKP28 /ZFFFFFFFFFF0000000000/ , ￥MKN28 /Z000000007FFFFFFF/FFFF/
DATA ￥MKP29 /ZFFFFFFFFFF8000000000/ , ￥MKN29 /Z0000000007FFFFFFF/FFFF/
DATA ￥MKP30 /ZFFFFFFFFFFC00000000/ , ￥MKN30 /Z0000000003FFFFFFF/FFFF/
DATA ￥MKP31 /ZFFFFFFFFFFE00000000/ , ￥MKN31 /Z000000001FFFFFFF/FFFF/
DATA ￥MKP32 /ZFFFFFFFFFFF00000000/ , ￥MKN32 /Z0000000000FFFFFFF/FFFF/
DATA ￥MKP33 /ZFFFFFFFFFF80000000/ , ￥MKN33 /Z00000000007FFFFFFF/FFFF/
DATA ￥MKP34 /ZFFFFFFFFFF00000000/ , ￥MKN34 /Z00000000003FFFFFFF/FFFF/
DATA ￥MKP35 /ZFFFFFFFFFFE00000000/ , ￥MKN35 /Z0000000001FFFFFFF/FFFF/
DATA ￥MKP36 /ZFFFFFFFFFFF00000000/ , ￥MKN36 /Z00000000000FFFFFFF/FFFF/
DATA ￥MKP37 /ZFFFFFFFFFFF80000000/ , ￥MKN37 /Z000000000007FFFFFFF/FFFF/
DATA ￥MKP38 /ZFFFFFFFFFFFC00000000/ , ￥MKN38 /Z000000000003FFFFFFF/FFFF/
DATA ￥MKP39 /ZFFFFFFFFFFFE00000000/ , ￥MKN39 /Z000000000001FFFFFFF/FFFF/
DATA ￥MKP40 /ZFFFFFFFFFFF00000000/ , ￥MKN40 /Z000000000000FFFFFFF/FFFF/
DATA ￥MKP41 /ZFFFFFFFFFFF80000000/ , ￥MKN41 /Z0000000000007FFFFFFF/FFFF/
DATA ￥MKP42 /ZFFFFFFFFFFFC00000000/ , ￥MKN42 /Z000000000003FFFFFFF/FFFF/
DATA ￥MKP43 /ZFFFFFFFFFFFE00000000/ , ￥MKN43 /Z000000000001FFFFFFF/FFFF/
DATA ￥MKP44 /ZFFFFFFFFFFF00000000/ , ￥MKN44 /Z000000000000FFFFFFF/FFFF/
DATA ￥MKP45 /ZFFFFFFFFFFF80000000/ , ￥MKN45 /Z0000000000007FFFFFFF/FFFF/
DATA ￥MKP46 /ZFFFFFFFFFFFC00000000/ , ￥MKN46 /Z000000000003FFFFFFF/FFFF/
DATA ￥MKP47 /ZFFFFFFFFFFFE00000000/ , ￥MKN47 /Z0000000000001FFFFFFF/FFFF/
DATA ￥MKP48 /ZFFFFFFFFFFF00000000/ , ￥MKN48 /Z0000000000000FFFFFFF/FFFF/
DATA ￥MKP49 /ZFFFFFFFFFFF80000000/ , ￥MKN49 /Z00000000000007FFFFFFF/FFFF/
DATA ￥MKP50 /ZFFFFFFFFFFFC00000000/ , ￥MKN50 /Z0000000000003FFFFFFF/FFFF/
DATA ￥MKP51 /ZFFFFFFFFFFFE00000000/ , ￥MKN51 /Z0000000000001FFFFFFF/FFFF/
DATA ￥MKP52 /ZFFFFFFFFFFF00000000/ , ￥MKN52 /Z0000000000000FFFFFFF/FFFF/
DATA ￥MKP53 /ZFFFFFFFFFFF80000000/ , ￥MKN53 /Z00000000000007FFFFFFF/FFFF/
DATA ￥MKP54 /ZFFFFFFFFFFFC00000000/ , ￥MKN54 /Z0000000000003FFFFFFF/FFFF/
DATA ￥MKP55 /ZFFFFFFFFFFF00000000/ , ￥MKN55 /Z00000000000001FFFFFFF/FFFF/
DATA ￥MKP56 /ZFFFFFFFFFFF00000000/ , ￥MKN56 /Z00000000000000FFFFFFF/FFFF/
DATA ￥MKP57 /ZFFFFFFFFFFF80000000/ , ￥MKN57 /Z0000000000000007FFFFFFF/FFFF/
DATA ￥MKP58 /ZFFFFFFFFFFFC00000000/ , ￥MKN58 /Z0000000000000003FFFFFFF/FFFF/
DATA ￥MKP59 /ZFFFFFFFFFFFE00000000/ , ￥MKN59 /Z0000000000000001FFFFFFF/FFFF/

```

Fig. 4. 1 Mask data

Table 4.3 Entry times and cpu time of CDC, nooptimize and ORIGINAL version

subroutine	CDC version			nooptimize version			ORIGINAL version		
	entries	time(sec)	fraction	entries	time(sec)	fraction	entries	time(sec)	fraction
DTSTEP	2203	7.552	0.0168	2671	31.294	0.0365	2671	23.838	0.0403
STATE(R)	74	3.048	0.0068	313	10.695	0.0125	313	8.546	0.0144
TRIP	2128	0.276	0.0006	2357	1.390	0.0016	2357	1.013	0.0017
HTADV	2202	37.237	0.0828	2670	63.839	0.0746	2670	45.115	0.0762
HTCOND	182766	41.474	0.0923	221610	48.731	0.0569	221610	43.024	0.0727
STATE(T)	2202	0.559	0.0012	2670	2.791	0.0033	2670	1.670	0.0028
JPROP(0)	2276	18.125	0.0403	2983	82.102	0.0959	2983	46.996	0.0794
VOLVEL	2202	19.837	0.0441	2670	71.925	0.0840	2670	37.941	0.0641
PHAINT	2202	70.080	0.1559	2670	116.322	0.1358	2670	87.394	0.1477
HZFLOW	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0
FWDrag	2202	20.291	0.0451	2670	22.006	0.0257	2670	21.401	0.0362
HLOSS	2202	5.122	0.0114	2670	8.035	0.0094	2670	6.900	0.0117
VEXPLT	2202	34.282	0.0763	2670	76.721	0.0896	2670	50.681	0.0856
JPROP(1)	2202	1.940	0.0043	2670	17.169	0.0201	2670	8.451	0.0143
PRESEQ	2504	23.605	0.0525	3284	59.857	0.0699	3284	36.100	0.0610
SYSSOL	2504	27.075	0.0602	3284	38.204	0.0446	3284	29.652	0.0501
VFINL	2202	13.998	0.0311	2670	58.896	0.0688	2670	31.854	0.0538
EQFINL	2202	25.477	0.0567	2670	43.717	0.0511	2670	29.682	0.0502
STATE(A)	2202	97.437	0.2167	2670	94.825	0.1107	2670	76.335	0.1290
RKIN	2128	1.397	0.0031	2357	3.123	0.0036	2357	2.146	0.0036
CONVAR	2128	0.758	0.0017	2357	4.658	0.0054	2357	2.987	0.0050
TOTAL		449.570			856.300			591.726	
JOB CPU					884.04			617.43	

5. 動的挙動解析

動的挙動解析ツールFORTUNE及びTOP 10によるTYPWR 2データでの各サブルーチンの実行回数及び実行コスト（スカラー計算における実行時間に比例する量）をFig.5.1に示す。ここでサブルーチン名の前に●印の付いているものがベクトル化の対象、○印の付いているものがスカラーハイド化の対象となったものである。次にFig.5.2に過渡解析時におけるサブルーチンの木構造と、各サブルーチン毎の実行コストの概略を示す。これらによると、実行コストは多数のサブルーチンに分散しており、最も実行コストの高いサブルーチンでも全体の11.4%しか占めず、比率が1%以上のサブルーチンが24個ある。サブルーチンの内容に関しては、方程式を解く部分は比較的コストが低く、方程式の係数を求めたり、計算した圧力から各状態量を求める部分のコストが高くなっている。このようなコードをベクトル化によりハイド化するためには、解法の変更よりもボリュームやジャンクションに対する計算の並列性を利用して数多くのサブルーチンをベクトル化する必要がある。ベクトル化の対象となるサブルーチンは40個で、ステートメント数は約10000行である。その他にスカラーレベルでの最適化を行ったサブルーチンは10個で、ステートメント数は約3500である。

NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	X	Q
I 0001	● PHAINT	1	1830	0	2670	529079641	11.4	1*****
I 0002	● STH2X6	1	686	0	6098	360837467	7.8	1*****
I 0003	● PRESEQ	1	976	0	3284	307516950	6.6	1*****
I 0004	● EQFINL	1	878	0	2670	297867709	6.4	1*****
I 0005	● VEXPLOF	1	938	0	2670	281810674	6.1	1*****
I 0006	● STH2X1	1	292	0	14754	245305154	5.3	1*****
I	STH2XB				0			
I 0007	○ PMINV3	1	46	0	352055	237687749	5.1	1*****
I 0008	● VOLVEL	1	597	0	2670	222501780	4.8	1*****
I 0009	● STATEP	1	986	0	366909	186752226	4.0	1*****
I 0010	● FWDRAG	1	551	0	2670	174600919	3.8	1*****
I 0011	● MADATA	1	408	0	222417	158409536	3.4	1*****
I 0012	● VLVELA	1	503	0	2983	14922758	3.2	1*****
I 0013	● HT1TOP	1	756	0	221610	13101506	2.8	1*****
I 0014	○ JCHOKE	1	1106	0	2670	110904428	2.4	1*****
I 0015	● VFINL	1	528	0	2670	92910761	2.0	1*****
I 0016	● PMINVF	1	144	0	3282	8490059	1.8	1*****
I 0017	● IAPACK	1	13	0	1126910	74735912	1.6	1*****
I	IATICK				0	2362723		
I 0018	● HLOSS	1	691	0	2670	7415668	1.6	1*****
I 0019	● PREDIB	1	412	0	78546	6807716	1.5	1*****
I 0020	● HTHETA	1	58	0	96120	63633608	1.4	1*****
I 0021	● STH2X0	1	32	0	560470	62212170	1.3	1*****
I 0022	● FIDS	1	141	0	373129	58188346	1.3	1*****
I 0023	● IPROP	1	460	0	10993	4885524	1.1	1*****
I 0024	● PSAIP0	1	96	0	271682	48357030	1.0	1***
I 0025	● VISCOL	1	74	0	367048	39227413	0.8	1***
I 0026	● HTRC1	1	480	0	340328	3499936	0.8	1***
I 0027	○ HZFLQW	1	738	0	10993	33840187	0.7	1***
I 0028	● HIFUB	1	156	0	280551	32591925	0.7	1***
I 0029	● STATE	1	620	0	2983	31716688	0.7	1***
I 0030	● PMINVW	1	156	0	3284	31666340	0.7	1***
I 0031	● DITIUS	1	259	0	206455	3029917	0.7	1**
I 0032	● CONDEN	1	297	0	54267	24582459	0.5	1**
I 0033	● INP10	1	120	0	0	20888130	0.5	1**
I 0034	● THCON	1	79	0	131			
I 0035	● HTD0V	1	417	0	739178	20669610	0.4	1*
I 0036	● RKIN	1	991	0	2670	18212070	0.4	1*
I 0037	● DTSEIP	1	775	0	2357	17072131	0.4	1*
I 0038	● CHFCAL	1	297	0	2671	16224359	0.4	1*
I 0039	● POLATL	1	124	0	785456	16158734	0.3	1*
I 0040	● VISCOG	1	67	0	369406	16025581	0.3	1*
I 0041	● *18	1	62	0	377212	15465692	0.3	1*
I 0042	● INPLNK	1	41	0	2101253	14708771	0.3	1*
I 0043	○ PSET	1	174	0	2654	12151918	0.3	1*
I 0044	● YID	1	64	0	456198	11729868	0.3	1*
I 0045	● INP	1	320	0	1281608	11534472	0.2	1*
I 0046	● PNINVR	1	57	2	2	10649019	0.2	1*
I 0047	● PACKER	1	384	0	3282	10514589	0.2	1*
I 0048	● HTCOND	1	402	0	2631	10476581	0.2	1*
I 0049	● FTBMDV	1	57	0	222417	9041401	0.2	1*
I 0050	● MOVER	1	354	0	197218	8705854	0.2	1*
I 0051	● MAJOUT	1	1817	0	2671	8383253	0.2	1*
					6	6645027	0.1	1*

Fig. 5.1 FORTUNE time cost of TYP PWR 2 data

1	0052	CONVAR	746	0	2357	4577294	0.1	1
1	0053	TIMSET	25	0	495482	4478209	0.1	1
1	0054	PSTDIB	1	395	0	2358	3509656	0.1
1	0055	DCVIC	1	227	0	5082	3356835	0.1
1	0056	*SIGNS	1	12	0	355578	316457	0.1
1	0057	O VALVE	1	929	0	2670	3063310	0.1
1	0058	SECOND	1	13	0	497841	2981046	0.1
1	0059	STACC	1	656	0	5966	283138	0.1
1	0060	O ACCUM	1	847	0	5340	274466	0.1
1	0061	STH2X3	1	438	0	5190	272613	0.1
1	0062	POLATS	1	123	0	76388	2381163	0.1
1	0063	INPUK	1	18	0	1669	212954	0.0
1	0064	O TSTATE	1	1022	0	2670	1573779	0.0
1	0065	PLTWRT	1	837	0	101	1384609	0.0
1	0066	O PMINUD	1	491	0	6	1204384	0.0
1	0067	FTBOUT	1	36	0	31	1134937	0.0
1	0068	FIBIN	1	77	0	1	1	1
1	0069	FIBCHK	1	34	0	1	1	1
1	0070	O TRIP	1	272	0	2357	864882	0.0
1	0071	I EDIT	1	892	0	1	74734	0.0
1	0072	O PUMP	1	485	0	5340	65516	0.0
1	0073	RHTCMP	1	1303	0	481	529952	0.0
1	0074	ALPACK	1	26	0	485	483485	0.0
1	0075	RPIPE	1	1655	0	23	468925	0.0
1	0076	FTBLCT	1	111	0	1532	455750	0.0
1	0077	O PUMP2	1	412	0	5342	400614	0.0
1	0078	HTSST	1	691	0	83	359823	0.0
1	0079	IHTCMP	1	1388	0	1	308078	0.0
1	0080	IELVTN	1	699	0	1	284419	0.0
1	0081	SCNREQ	1	2009	0	184	271196	0.0
1	0082	INVJT	1	422	0	1	266748	0.0
1	0083	RBRNCH	1	1594	0	15	253128	0.0
1	0084	RCONVR	1	1771	0	1	244689	0.0
1	0085	PMINV4	1	29	0	12	244710	0.0
1	0086	INPMOD	1	79	0	1669	242414	0.0
1	0087	HT1INP	1	937	0	16	195335	0.0
1	0088	HYDRO	1	168	0	2670	184230	0.0
1	0089	TSEISL	1	766	0	1	184460	0.0
1	0090	INPPCK	1	29	0	1537	162699	0.0
1	0091	TRAN	1	122	0	1	159985	0.0
1	0092	RTMDV	1	1204	0	14	151199	0.0
1	0093	UNIT	1	48	0	148	148066	0.0
1	0094	BUFFOT	1	69	0	143	148870	0.0
1	0095	SYSSOL	1	2	0	1	1	1
1	0096	MIREC	1	434	0	3284	139647	0.0
1	0097	IDFIND	1	145	0	1101	13150	0.0
1	0098	RSNGV	1	228	0	102	128642	0.0
1	0099	ISTATE	1	444	0	2190	119529	0.0
1	0100	SETNDF	1	812	0	16	11454	0.0
1	0101	IVLVEL	1	23	0	9344	112092	0.0
1	0102	REMIM	1	161	0	295	98134	0.0
1	0103	ICMPN1	1	840	0	10	85696	0.0
1	0104	WRPLID	1	986	0	11	81688	0.0
1	0105	IVLVEL	1	1464	0	1	67351	0.0
1	0106	REMIM	1	20	0	135	63367	0.0
1	0107	ICMPN1	1	409	0	1	52113	0.0
1	0108	WRPLID	1	28	0	2671	46410	0.0
1	0109	ICMPN1	1	757	0	1	45835	0.0
1	0110	WRPLID	1	906	0	1	42888	0.0

Fig. 5.1 continued

JAERI-M 86-019

1	1723	0	39743	0.0		
1	0109	4	447	0.0		
1	0110	87	0	39077	0.0	
1	0111	1245	0	2	37716	0.0
1	0112	RACCUM	1	2	34082	0.0
1	0113	TRANSET	1	2	33746	0.0
1	0114	RRKIN	1	1	31481	0.0
1	0115	RMADAT	1	1	30394	0.0
1	0116	ICMPF	1	1	27544	0.0
1	0117	DMPLST	1	1	25699	0.0
1	0118	ITRIP	1	1	24244	0.0
1	0119	FTBDEL	1	1	24049	0.0
1	0120	RPMND	1	1	20297	0.0
1	0121	FTBSV	1	1	18055	0.0
1	0122	FTBDSB	1	1	17374	0.0
1	0123	ICONVR	1	1	17269	0.0
1	0124	RGNIBL	1	1	15434	0.0
1	0125	HEADER	1	20	14566	0.0
1	0126	LJPROP	1	1	12779	0.0
1	0127	LCONIG	1	1	10678	0.0
1	0128	ISNGJ	1	1	10544	0.0
1	0129	IVELST	1	1	10481	0.0
1	0130	LOCF	1	1	10253	0.0
1	0131	RMIEDT	1	1	8251	0.0
1	0132	LCNFGS	1	1	7444	0.0
1	0133	RCOMIN	1	1	7351	0.0
1	0134	FTBSLK	1	1	5859	0.0
1	0135	SURLEN	1	1	4726	0.0
1	0136	IRKIN	1	1	4515	0.0
1	0137	RPMDC	1	2	4396	0.0
1	0138	FTBFLD	1	1	4354	0.0
1	0139	INPUT	1	2	4049	0.0
1	0140	LCORPN	1	1	3734	0.0
1	0141	IPUMP	1	1	3613	0.0
1	0142	RPUMP	1	2	2742	0.0
1	0143	RSTFEC	1	2	2513	0.0
1	0144	TRNFIN	1	1	2459	0.0
1	0145	IPIPE	1	1	2405	0.0
1	0146	RTSC	1	1	2167	0.0
1	0147	RNEWP	1	1	2130	0.0
1	0148	TCNSL	1	1	1770	0.0
1	0149	ATRIP	1	1	1677	0.0
1	0150	ITRSCH	1	1	1652	0.0
1	0151	STH2XJ	1	1	1034	0.0
1	0152	RRSTD	1	1	1022	0.0
1	0153	RSTFIN	1	1	1009	0.0
1	0154	YBUFFO	1	1	858	0.0
1	0155	FTBORG	1	0	0	0.0
1	0156	IRFLHT	1	1	400	0.0
1	0157	IMIEDT	1	1	556	0.0
1	0158	GNINAT	1	1	526	0.0
1	0159	LINES	1	2	430	0.0
1	0160	WRITEC	1	1	390	0.0
1	0161	READIC	1	1	384	0.0
1	0162	ISFDIS	1	1	332	0.0
1	0163	RPLTN	1	1	324	0.0
1	0164	YUNIT	1	1	292	0.0
1	0165	MCONST	1	2	214	0.0
1	0166	NFUNIT	1	2	144	0.0
1	0167	RPLIPS	1	1	139	0.0
1	0168	FTBINT	1	1	126	0.0
1	0169	IGNIBL	1	1	69	0.0

1	0170	RCARDS	1	174	0	0.0
1	0171	RPLTD	1	674	0	0.0
1	0172	RCOMPT	1	664	0	0.0
1	0173	RUSROR	1	241	0	0.0
1	0174	RELAPS	1	1859	0	0.0
1	0175	RPUUNIT	1	151	0	0.0
1	0176	RINTRY	1	320	0	0.0
1	0177	RMFIDS	1	205	0	0.0
1	0178	FTBFITB	1	35	0	0.0
1	0179	RNDNCN	1	145	0	0.0
1	0180	FTBMER	1	31	0	0.0
1	0181	INPB	1	36	0	0.0
1	0182	INXGET	1	27	0	0.0
1	0183	FTBEYP	1	77	0	0.0
1	0184	TRNCTL	1	24	0	0.0
1	0185	RS12PL	1	84	0	0.0
1	0186	FTBRDC	1	76	0	0.0
1	0187	PLQTHD	1	4	0	0.0
1	0188	RCRVSP	1	258	0	0.0
1	0189	SORPFR	1	275	0	0.0
1	0190	STRACE	1	10	0	0.0
1	0191	IPLDT	1	539	0	0.0
1	0192	STREFN	1	160	0	0.0
1	0193	SSTCHK	1	1422	0	0.0
1	0194	LENGTH	1	33	0	0.0
1	0195	MDAIA2	1	443	0	0.0
1	0196	RTURB	1	1753	0	0.0
1	0197	HTCSOL	1	90	0	0.0
1	0198	RPLDT	1	229	0	0.0
1	0199	STRIP	1	372	0	0.0
1	0200	SSTFIN	1	114	0	0.0
1	0201	TURBST	1	297	0	0.0
1	0202	IPLT2D	1	603	0	0.0
1	0203	HEADIN	1	8	0	0.0
1	0204	RPLDF	1	583	0	0.0
1	0205	GOTDR	1	6	0	0.0
1	0206	VIMPUL	1	1214	0	0.0
1	0207	GASIMC	1	75	0	0.0
1	0208	STSTEP	1	844	0	0.0
1	0209	HTRC2	1	474	0	0.0
1	0210	FTNPEQ	1	7	0	0.0
1	0211	RRESIF	1	722	0	0.0
1	0212	TPAHT	1	257	0	0.0
1	0213	SSSET	1	1594	0	0.0
1	0214	ICOMPT	1	428	0	0.0
1	0215	FTBPR1	1	117	0	0.0
1	0216	SSCTCL	1	24	0	0.0
1	0217	HELPHD	1	94	0	0.0
1	0218	F18OPN	1	144	0	0.0
1	0219	REEDIT	1	24	0	0.0
1	0220	UPLOAD	1	511	0	0.0
1	0221	RDEDT	1	24	0	0.0
1	0222	FTIMOV	1	7	0	0.0
1	0223	GACON	1	227	0	0.0
1	0224	NFSIZE	1	27	0	0.0
1	0225	INTERI	1	199	0	0.0
1	0226	RCDELT	1	373	0	0.0
1	0227	INPA6	1	20	0	0.0
1	0228	FTDERR	1	25	0	0.0
1	0229	QFMOVE	1	897	0	0.0
1	0230	QFRCH	1	762	0	0.0
1	0231	FTCCLS	1	74	0	0.0

Fig. 5.1 continued

Fig. 5. 1 continued

T R E E S T R U C T U R E		EXECUTIONS	COST (*10**3)	%	TOTAL COST (*10**3)	%
TRAN	--+--DTSTEP--+					
	+---MOVER --+					
	+---STATE ...					
	+---JPROP ...					
	+---VLVELA...					
	+---PLTREC					
	+---MIREC					
	+---MAJOUT					
	+---RSTREC					
	+---PLTWRT					
	+---INTERI					
	+---TRIP					
	+---TSTATE--+					
	+---POLATS					
	+---STH2X1					
	+---STH2X3					
	+---STH2X6					
	+---%STH2X2					
	+---%STH2XF					
	+---HTADV --+					
	+---QFMOVE--+					
	+---READEC					
	+---WRITEC					
	+---QFSRCH					
	+---HTRC2 --+					
	+---POLATL					
	+---HTRC1 ...					
	+---QFHTRC					
	+---HT2TDP--+					
	+---POLATL					
	+---MDATA2--+					
	+---GASTHC					
	+---READEC					
	+---HTCSOL					
	+---HT1TDP--+					
	+---POLATL					
	+---MDATA--+					
	+---GASTHC					
	+---GAPCON--+					
	+---CELMOD					
	+---CTHEXP					
	+---HTCOND--+					
	+---POLATL					
	+---HTRC1 --+					

Fig. 5.2 Tree structure and time cost

T R E E S T R U C T U R E		EXECUTIONS	COST (*10*3)	%	TOTAL COST	%
					(*10**3)	
TRAN	-- - HTADV --+--HT1TDP--+--HTCOND--+--HTRC1 --+--PSATPD	0	0	-	-	
	+--TPAWHT--+	0	0	-	-	
	+--DITTUS	206455	30240	0.7	-	
	+--PREDNB--+	78546	68077	1.5	-	
	+--PSATPD	271682	48357	1.0	-	
	+--CHFCAL	78546	16159	0.3	-	
	+--PSTDNB--+	5082	3510	0.1	-	
	+--STH2X3	5082	2673	0.1	-	
	+--THCON ...	5082	247	0.0	-	
	+--CONDEN	10164	416	0.0	-	
	+--VISC0G...	54267	24582	0.5	-	
	+--HYDRO --+	2670	3068	0.1	3068	0.1
	+--VALVE --+	0	0	-	-	
	+--STRACE	0	0	-	-	
	+--POLATL	0	0	-	-	
	+--POLATS	0	0	-	-	
	+--VOLVEL--+	2670	222502	4.8	222502	4.8
	+--HELPHD--+	0	0	-	-	
	+--HELPHD	0	0	-	-	
	+--HEADLN	0	0	-	-	
	+--XTIMEL	2670	529080	11.4	683494	14.7
	+--PHAINIT--+	0	0	-	-	
	+--PSATPD	96120	63634	1.4	-	
	+--HTHETA	373129	58188	1.3	-	
	+--FIDIS	280551	32592	0.7	-	
	+--HIFBUB	0	0	-	-	
	+--HELPHD...-	2670	174601	3.8	174601	3.8
	+--FWDRAG--+	0	0	-	-	
	+--HELPHD...-	2670	74157	1.6	74157	1.6
	+--HLLOSS	2670	281811	6.1	286276	6.2
	+--VEXPIT--+	0	0	-	-	
	+--HELPHD...-	5340	655	0.0	-	
	+--PUMP --+	5340	400	0.0	-	
	+--PUMP2 --+	10680	333	0.0	-	
	+--POLATS	10680	333	0.0	-	
	+--ACCUM --+	5340	2744	0.1	-	
	+--HELPHD...-	0	0	-	-	
	+--TURBST	0	0	-	-	
	+--JCHOKE--+	2670	110904	2.4	136260	2.9
	+--STH2X0	193422	21470	0.4	-	
	+--STH2X1	2820	1811	0.0	-	
	+--HELPHD...-	0	0	-	-	
	+--PSATPD	0	0	-	-	

Fig. 5.2 continued

T R E E S T R U C T U R E		EXECUTIONS	COST (*10**3)	%	TOTAL COST (*10**3)	%
TRAN	-- -HYDRO - -JCHOKE--+-STH2X3	0	0	-	-	-
	+--%STH2X2	0	0	-	-	-
+---JPROP	--+-STH2X6	132	75	0.0	38227	0.9
	+--HELPHD...	5340	21789	0.5	-	-
	+--STRACE	0	-	-	-	-
	+--HZFLOW--+	0	-	-	-	-
	+--HTHETA	5340	16438	0.3	-	-
+---VFINL	--+--PRESEQ--+	0	-	-	-	-
	+--HELPHD...	2670	92911	2.0	92911	2.0
	+--%TIMEI	3284	307517	6.6	307517	6.6
	+--SYSSOL--+	0	-	-	-	-
	+--READEC	3284	140	0.0	366490	7.9
	+--PMINV0--+	0	1204	0.0	-	-
	+--PMINV1--+	6	-	-	-	-
	+--PMINV2	0	-	-	-	-
	+--PMINV5	6060	73	0.0	-	-
	+--PMINV3	3402	239	0.0	-	-
	+--PMINV4	12	245	0.0	-	-
	+--WRITEC	0	-	-	-	-
	+--PMINUM	3284	31686	0.7	-	-
	+--PMINVR	3282	10515	0.2	-	-
	+--PMINVF--+	3282	84900	1.8	-	-
	+--PMINV0--+	0	-	-	-	-
	+--PMINV3	3378653	237449	5.1	-	-
	+--PMINUS	3281	39	0.0	-	-
	+--HELPHD...	0	-	-	-	-
+---EQFINL	+--HELPHD...	-	-	-	-	-
+---VIMPLT	+--HELPHD...	-	-	-	-	-
	+--READEC	0	-	-	-	-
	+--POMP	0	-	-	-	-
	+--ACUM	0	-	-	-	-
	+--TURBST	0	-	-	-	-
	+--HELPHD...	0	-	-	-	-
	+--JCHOKE...	0	-	-	-	-
	+--SYSSOL...	0	-	-	-	-
+---PIMPLT	+--HELPHD...	0	-	-	-	-
+---SIMPLT	+--HELPHD...	0	-	-	-	-
	+--HELPHD...	0	-	-	-	-

Fig. 5.2 continued

T R E E S T R U C T U R E		EXECUTIONS	COST (*10**3)	%	TOTAL COST	%
TRAN	-- -HYDRO	0	0	-	0	(*10**3)
	+--SIMPLT	0	0	-	0	
	+--PMINVM	0	0	-	0	
	+--STATE	2670	28389	0.6	836746	18.3
	+--STAC	5340	2534	0.1		
	+--STH2X6	5340	3037	0.1		
	+--STH2X1	10680	6860	0.2		
	+--STATEP	0	0	-		
	+--HELPHD	0	0	-		
	+--STEP	328410	167157	3.6		
	+--STH2X1	0	0	-		
	+--STH2X0	0	0	-		
	+--STH2X3	0	0	-		
	+--STH2X2	328410	201935	4.7		
	+--STH2X6	0	0	-		
	+--STH2XF	562224	319793	6.9		
	+--VISCOL	328410	35140	0.7		
	+--THCON	328410	36454	0.8		
	+--VISC0G	328410	13465	0.3		
	+--VLEVELA	0	0	-		
	+--HELPHD	0	0	-		
	+--STRACE	0	0	-		
	+--RKIN	2357	17072	0.4	18056	0.4
	+--POLATL	2357	102	0.0		
	+--POLATS	28284	882	0.0		
	+--CONVAR	2357	4577	0.1	4577	0.1
	+--POLATL	0	0	-		

Fig. 5.2 continued

6. コードのベクトル化

6.1 基本的なベクトル化の方法

熱伝導計算部分では熱構造体及び熱メッシュ、流体力学計算部分ではボリューム及びジャンクションに関して独立して計算できることを利用し、それらの長さをベクトル長としてベクトル化を行った。基本的な方法はMOD 1の場合²⁾と同様である。

6.1.1 IF 文及びサブルーチン CALL 文を含む DO ループの場合

IF 文が入れ子構造になっていたり、その中にサブルーチン CALL 文がある場合、IF 文で DO ループを区切り、インデックスリストを作り直し、リストベクトルによるベクトル化を行う。Fig. 6.1 にその例を示す。

6.1.2 ビット操作ファンクションを含む DO ループの場合

ビット操作ファンクションは、ほとんどアセンブラーで作られた外部手続きであるため、自動ベクトル化の対象からはずされている。DO ループ内の他の FORTRAN 文についても、コンパイラによる最適化がそこで途切れるため不利である。さらにコモン変数などの定義、引用関係がコンパイラには不明のため、DO ループそのものをスカラーとしてしまう場合もある。

パック語からビット操作ファンクションを通じて得られるデータの中には、コンポーネントの種類やジャンクションに接続しているコンポーネント番号のように、入力時に一度決定すれば以後二度と変わらないものがある。このようなデータは、サブルーチンの先頭で一度だけビット操作ファンクションを使用して取出し、作業用配列に貯えておき、DO ループの中では作業用配列を用いるように変更する。その結果、外部手続きの引用によるベクトル化に対する障害が除かれると共に、ファンクションの引用回数が相当減るという効果がある (Fig. 6.2)。

次に、アキュムレータフラグのように計算の進行に従って変わる可能性のあるデータについては、FORTRAN の組込み関数の中の整数データ用のビット操作関数 ISHFT, IAND, IOR を使用するように変更する。具体的には、Fig. 6.3, Fig. 6.4 のようにパック語の入っている 8 バイトデータについて EQUIVALENCE 文で整数配列と領域共有し、整数データ側に対して組込み関数を組合せることによってビット操作ファンクションの代用をする。その結果、ベクトル化可能となる。

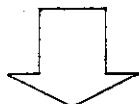
6.2 热伝導計算部分のベクトル化

流体ボリュームの壁を通しての熱伝達は、熱構造体を用いて計算される。熱構造体は、燃料、蒸気発生器を通しての熱の伝達、パイプ、格納容器などの壁を通しての熱伝達がモデル化できるようになっている。Fig. 6.5 にモデルのレイアウトを示す。一次元熱伝導方程式と境界条件

```

IF (ERX) GO TO 1010
SATT(I) = TSAT
SATHF(I) = HSUBFS
SATHG(I) = HSUBGS
DTDP(I) = TSAT*(VSUBGS-VSUBFS)/(HSUBGS-HSUBFS)
DTDUG(I) = 0.0DO
DTDXA(I) = 0.0DO
IGAS = 1
IF (VOIDG(I) .EQ. 0.0DO) IGAS = 0
ILIQ = 1
IF (VOIDG(I) .EQ. 1.0DO) ILIQ = 0
IF (ILIQ .EQ. 0) GO TO 125
UBAR = UF(I)
PROP(10) = TSAT
CALL STH2XF (LFA(FILNDX(6)),PROP,IP,ERX)
IF (ERX) GO TO 1010

```



```

DO 6120 I¥ = 1,IMAX¥3
I = IT¥3(I¥)
IF (ERT¥(I¥)) IERT¥(I¥) = 1
IF (IERT¥(I¥) .GT. 0) GO TO 6120
SATT(I) = ST¥(1,I¥)
SATHF(I) = ST¥(15,I¥)
SATHG(I) = ST¥(16,I¥)
DTDP(I) = ST¥(1,I¥)*(ST¥(12,I¥)-ST¥(11,I¥))
/(ST¥(16,I¥)-ST¥(15,I¥))
DTDUG(I) = 0.0DO
DTDXA(I) = 0.0DO
IF (VOIDG(I) .NE. 0.0DO) THEN
ICNT¥1 = ICNT¥1 + 1
ITBL¥1(ICNT¥1) = I¥
END IF
IF (VOIDG(I) .NE. 1.0DO) THEN
ICNT¥2 = ICNT¥2 + 1
ITBL¥2(ICNT¥2) = I¥
END IF
6120 CONTINUE
*VOCL LOOP,NOVREC
DO 6130 J¥ = 1,ICNT¥2
I¥ = ITBL¥2(J¥)
I = IT¥3(I¥)
PT¥(4,I¥) = UF(I)
PT¥(10,I¥) = ST¥(1,I¥)
6130 CONTINUE
CALL STH2VF (LFA(FILNDX(6)),PT¥,IP¥1,JP¥1,IP¥,ERT¥,ITBL¥2,ICNT¥2
,IECNT¥)
*VOCL LOOP,NOVREC
DO 6150 J¥ = 1,ICNT¥2
I¥ = ITBL¥2(J¥)
I = IT¥3(I¥)
IF (ERT¥(I¥)) IERT¥(I¥) = 1
IF (IERT¥(I¥) .GT. 0) GO TO 6150
:

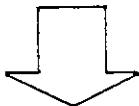
```

Fig. 6.1 An example of vectorization by list vector

```

K=IXVF
J=IXIPX
IXP=IXSOP-1
S DO 120 I=IJ,IJE,IJSKP
S   KP= YIDANDC *MKN48, YSFT(IJ1(I),30) + IXP
S   LP= YIDANDC *MKN48, YSFT(IJ2(I),30) + IXP
S   DVELFJ(J)=VFDPK(K)*SOURCP(KP)-VFDPL(K)*SOURCP(LP)
S   DVELGJ(J)=VGDPK(K)*SOURCP(KP)-VGDPD(K)*SOURCP(LP)
S   K=K+4
S   J = J + 26
S 120 CONTINUE

```



```

IF( YFIRST(3) ) THEN
  I¥ = 0
  DO 100 I = IJ , IJE , IJSKP
  I¥ = I¥ + 1
  *   KP(I¥) = IANDC YJ0001, IORC ISHFT(YIJ1(1,I),2 ,
  *   *   ISHFT(YIJ1(2,I),-30 ) ) + IXP
  *   *   LP(I¥) = IANDC YJ0001, IORC ISHFT(YIJ2(1,I),2 ,
  *   *   ISHFT(YIJ2(2,I),-30 ) ) + IXP
  100 CONTINUE
  IF( I¥ .GT. MAXJUN ) CALL THEEND( 'MAXJUN' , I¥ )
  YFIRST(3) = .FALSE.
  END IF
  :
  I¥=0
  K=IXVF
  J=IXIPX
*VOCL LOOP,NOVREC
  DO 108 I=IJ,IJE,IJSKP
  I¥ = I¥ + 1
  KP = YKP(I¥)
  LP = YLP(I¥)
  DVELFJ(J)=VFDPK(K)*SOURCP(KP)-VFDPL(K)*SOURCP(LP)
  DVELGJ(J)=VGDPK(K)*SOURCP(KP)-VGDPD(K)*SOURCP(LP)
  K = K + 4
  J = J + 26
  108 CONTINUE

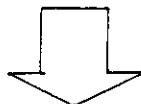
```

Fig. 6.2 Vectorization of sample DO - loop (1)

```

DO 100  I = IV, IVE, IVSKP
  IMAP(I) = AND($(SFT($(MKN1,53),IMAP(I))
  IF (VOIDGO(I) .GT. 0.0D0 .AND. VOIDG(I) .EQ. 0.0D0)
* IMAP(I) = OR($(SFT($(MKP1,53),IMAP(I))
  PO(I) = P(I)
  UFO(I) = UF(I)
  UGO(I) = UG(I)
  VOIDGO(I)= VOIDG(I)
  QUALAO(I)= QUALA(I)
  BORONO(I)= BORON(I)
  DOTMO(I) = DOTM(I)
  HIFO(I) = HIF(I)
  HIGO(I) = HIG(I)
100 CONTINUE

```



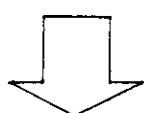
```

DATA           ¥J0003 / Z FFEF FFFF /
DATA           ¥J0004 / Z 0010 0000 /
:
INTEGER*4     *IJ2 (2,1) , *JC (2,1)
*             , *IMAP (2,1)
EQUIVALENCE  (IJ2 (1),*IJ2 (1,1)),(JC (1),*JC (1,1))
*             ,(IMAP (1),*IMAP (1,1))
:
V   DO 100  I = IV, IVE, IVSKP
V   *IMAP(1,I) = IAND($(J0003,*IMAP(1,I))
V   IF (VOIDGO(I) .GT. 0.0D0 .AND. VOIDG(I) .EQ. 0.0D0)
V   *IMAP(1,I) = IOR($(J0004,*IMAP(1,I))
V   PO(I) = P(I)
V   UFO(I) = UF(I)
V   UGO(I) = UG(I)
V   VOIDGO(I)= VOIDG(I)
V   QUALAO(I)= QUALA(I)
V   BORONO(I)= BORON(I)
V   DOTMO(I) = DOTM(I)
V   HIFO(I) = HIF(I)
V   HIGO(I) = HIG(I)
V 100 CONTINUE

```

Fig. 6.3 Vectorization of sample DO - loop (2)

```

IX=IXIPX
DO 2000 I =IJ,IJE,IJSKP
  FIDXUP(IX)=0.0DO
  IF ( YGE( YSFT(JC(I),2),    YB0001) ) GO TO 1990
    FORMFJ(I) = 0.0DO
    FORMGJ(I) = 0.0DO
  IF( YLTC( YSFT(IMREG(I),37), YB0001 ).OR. YLTC(IJ1(I), YB0001))
* GO TO 1990
    K      = YIDAND( YMKN43, YSFT(IJ1(I),18))
    L      = YIDAND( YMKN43, YSFT(IJ2(I),18))
    VDF   = VOIDFJ(I)
    VDG   = VOIDGJ(I)
    IF ( VELGJO(I) ) 7,6,8
6   IF ( VELFJO(I).GE.0.0DO ) GO TO 8
    :

    INTEGER*4   YJC    (2,1)           , YMREG(2,1)
    , YIJ1    (2,1)           , YIJ2    (2,1)
    EQUIVALENCE (JC    (1),YJC    (1,1)),(IMREG (1),YMREG(1,1))
    , (IJ1    (1),YIJ1   (1,1)),(IJ2    (1),YIJ2   (1,1))

    IX=IXIPX
    IY = 0
    JY = 0
*VOCL LOOP,NOVREC
M   DO 6000 I = IJ, IJE, IJSKP
    IY = IY + 1
V     FIDXUP(IX)=0.0DO
V     IF ( YSHFT(YJC(1,I),6) .GE. 0) GO TO 5990
V       FORMFJ(I) = 0.0DO
V       FORMGJ(I) = 0.0DO
V       IF( YSHFT(YMREG(2,I),9).LT.0 .OR. YIJ1(1,I).LT.0) GO TO 5990
V         K      = IAND( YJ0001, YSHFT(YIJ1(1,I),-10))
V         L      = IAND( YJ0001, YSHFT(YIJ2(1,I),-10))
V         IF ( VELGJO(I) ) 7,6,8
V   6   IF ( VELFJO(I).GE.0.0DO ) GO TO 8
    :

```

Fig. 6.4 Vectorization of sample DO - loop (3)

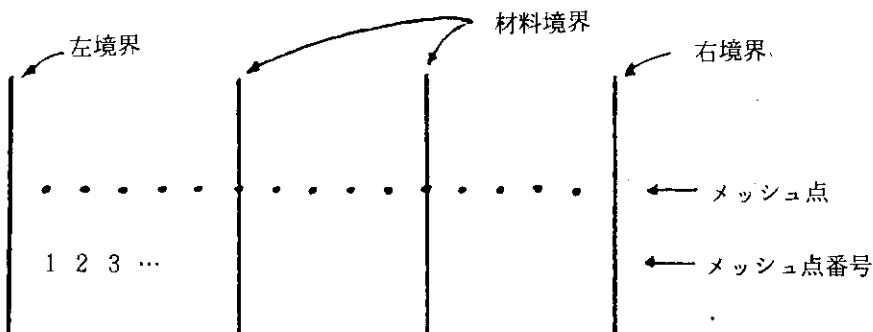


Fig. 6.5 Mesh point layout

の解法は、数値計算上は三重対角行列の解法に帰着し、ガウスの消去法によって解かれている。また、熱伝導率： κ 、熱容量係数： ρCp は、テーブル内挿又は関数計算によって求められる。熱計算部分の木構造及び過渡計算(TYPPW2)における実行回数と実行コストは、Fig.5.2のHTADV以下に示すとおりである。この中で今回ベクトル化の対象としたサブルーチンを、Table 6.1に示す。

Table 6.1 Names and functions of vectorized heat conduction subroutines

オリジナル版 でのルーチン名	ベクトル化版 でのルーチン名	内 容	主要ベクトル長
HTADV	HTADV	熱計算のアドバンスマントの制御	境 界 数
HT1TDP	HT1TDV	一次元熱伝導方程式の計算	熱構造体数、全メッシュ点数、その他
MADATA	MADATV	メッシュ点での熱伝導率及び熱容量の計算	全メッシュ間数
HTCOND	HTCONV	境界条件の計算	境 界 数
HTRC1	HTRC1V	相関式による熱伝達係数の計算	流体に接する境界数
DITTUS	DITTSV*	Dittus - Boelterの相関式を使用した強制冷却の計算	" の一部
CONDEN	CONDEV*	凝縮の熱伝達の計算	" "
PREDNB	PREDNV*	pre - DNB の計算	" "
PSTDNB	PSTDNV*	post - DNB の計算	" "
TPAWHT	TPAWHV*	気液混合状態での熱伝達の計算	" "
CHFCAL	CHFCAV*	臨界熱流束の計算	" "

* : ベクトル長が小さい場合のために全く同じ構造のスカラー版も用意する。

ベクトル化のために以下の方法を導入した。

- (1) 热計算部分は各熱構造体に関して独立に計算できるため、HTADVでHT1TDPを引用している箇所の外側にある熱構造体ループを、HT1TDP以下のサブルーチンに引込みベクトル化を行う。
- (2) オリジナルにメッシュ点又はメッシュ間にに関するDOループが存在しているブロックは、熱構造体ループとの二重DOループ構造となるが、ベクトル長を大きくするために可能な限りこれを一重化する。
- (3) 热伝導方程式を解く箇所は、ガウスの消去法を用いているためメッシュ点に関して再帰式となり、ベクトル化はできない。そこで、熱構造体ループとメッシュ点ループを入れ替えて熱構造体に関してベクトル化する。
- (4) 左右の境界に関する計算は、熱構造体ループと境界ループをまとめて一つのDOループにする。これによってベクトル長を2倍にすることができる。

(5) DO ループ内にある IF 文による選択のうち条件が計算途中でも不变なものは、あらかじめリストベクトルを作成しておきそれに従って計算する。

(6) 頻繁に使用するデータで不变なものは、あらかじめパック語から取出しておく。

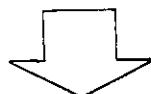
このため各種のリストベクトルを導入するが、そのうち主要なものは 2 つである。

(1) 二重ループの一重化のためのリストベクトル

メッシュ点又はメッシュ間数に関する熱構造体形状データ及び温度データは、Fig. 6.6 のような二次元配列 $\times \times \times \$(\ell, i)$ に格納される。このようなデータを一重ループで扱うために Table 6.2 に示す 3 種類のリストベクトルを導入する。なお二次元配列 $\times \times \times \$(\ell, i)$ は、一次元配列 $Z \times \times \times \$(j)$ と EQUIVALENCE 文で領域共有しておく。

(使用例)

```
DO 1 I = 1, *NH
    DO 1 L = 1, COLS$(I)
1      XXX$(L, I) = ....
```



```
DO 1 J = 1, *LCOLS
1      ZXXX$(LCOLS$(J)) = ....
```

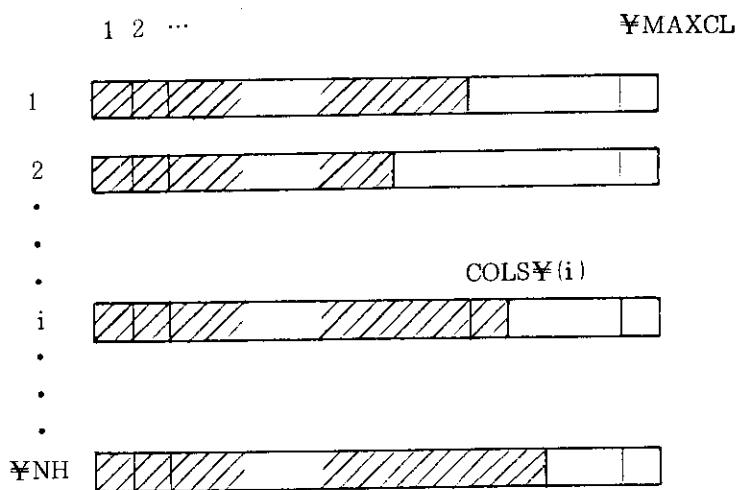


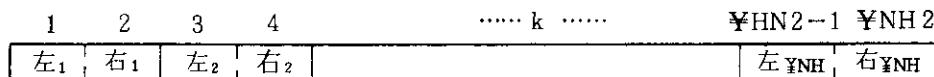
Fig. 6.6 Data structure of mesh point

Table 6.2 List vector names and contents regarding mesh point

名 前	内 容	有 効 長
LCOLS $\Psi(j)$	全メッシュ点の取り出し (ℓ, i) のとき $(i - 1) * \Psi_{MAXCL} + \ell$	$\Psi_{COLS} \equiv \sum_{i=1}^{\Psi_{NH}} \Psi(i)$
LNN $\Psi(j)$	メッシュ間の取り出し	$\Psi_{LNN} \equiv \sum_{i=1}^{\Psi_{NH}} \Psi(i)$ $= \Psi_{COLS} - \Psi_{NH}$
LIC $\Psi(j)$	内部メッシュ点の取り出し	$\Psi_{LIC} \equiv \Psi_{COLS} - 2 * \Psi_{NH}$

(2) 左右境界の一体化のためのリストベクトル

境界に関するデータは、Fig. 6.7 のように左右のデータを交互に並べた配列を使用する。このような構造にすることにより、オリジナルと同じ順序でデータを取扱うことができる。特にサブルーチン HTADV では、流体ボリュームへの熱輸送量を足し込んでいくためこの順序は精度上大きな問題になる。境界データを扱うために Table 6.3 に示す 4 種類のリストベクトルを導入する。



$$\Psi_{NH2} : \text{境界数} = \text{熱構造体数} (\Psi_{NH}) * 2$$

Fig. 6.7 Data structure of boundary

Table 6.3 List vector names and contents regarding boundary

名 前	内 容
LBI $\Psi(k)$	境界のインデックスを熱構造体のインデックスに変換
LBL $\Psi(k)$	境界メッシュ点取り出し
LBN $\Psi(k)$	境界メッシュ間取り出し
LBNO $\Psi(k)$	左右境界の識別 $\begin{cases} = 0 : \text{左} \\ = 1 : \text{右} \end{cases}$

以下に、各サブルーチンごとの具体的なベクトル化の方法について説明する。

6.2.1 HTADV

- (1) 熱計算で使用するデータのうち、計算途中で不変なデータ及びリストベクトルはすべてこのサブルーチンで1回だけ設定する。
- (2) サブルーチン HT1TDP を引用している箇所では、外側の DO ループを取除きサブルーチンの内部に熱構造体ループを組込む。
- (3) 流体ボリュームへの熱輸送量を計算する箇所では、1つの流体ボリュームに2つ以上の熱構造体が接続している場合は再帰的になる。従ってこのようなケースをあらかじめ検出しておき(変数 SAMEV)、計算方法を変更する。
- (4) 再冠水計算の箇所(サブルーチン QFMOVE 以下)は、今回はベクトル化を行わなかった。

6.2.2 HT1TDP

サブルーチン HT1TDP(ベクトル化版では HT1TDV)では全体を Fig. 6.8 のように11個のブロックに分割し、それぞれのブロックに関して以下のようなベクトル化を行う。

- (1) ブロック 1 では、初期設定を行うが、大部分のデータは既に初期設定されている。名前付きコモン HTRCOM内の変数を境界データとして以下のように配列化する。

$$\begin{aligned}
 & \left. \begin{array}{l} \cdot \text{HTCFL} \\ \cdot \text{HTCFR} \end{array} \right\} \longrightarrow \text{HTCFY}(k) \\
 & \left. \begin{array}{l} \cdot \text{HTCGL} \\ \cdot \text{HTCGR} \end{array} \right\} \longrightarrow \text{HTCGY}(k) \\
 & \left. \begin{array}{l} \cdot \text{QFGOL} \\ \cdot \text{QFGOR} \end{array} \right\} \longrightarrow \text{QFGOY}(k) \\
 & \cdot \text{QFFO} \longrightarrow \text{QFFOY}(k)
 \end{aligned}$$

ここで $k = 1, \dots, N_{\text{NH2}}$

- (2) ブロック 2 での圧力計算のために、以下の3ケースのリストベクトルをあらかじめ作成しておく。

- i) LGAP1Y : $NHIY(i) > 0$ かつ $GPINTY(i) > 0.0$
- ii) LGAP2Y : $NHIY(i) > 0$ かつ $GPINTY(i) \leq 0.0$
- iii) LGAP3Y : $NHIY(i) \leq 0$ かつ $GPINTY(i) > 0.0$

PGAS, PFLUIDは、コモン GAPVAR内の変数であり、以下のように配列化する。

$$\begin{aligned}
 & \cdot \text{PGAS} \longrightarrow \text{PGASY}(i) \\
 & \cdot \text{PFLUID} \longrightarrow \text{PFLUY}(i)
 \end{aligned}$$

- (3) サブルーチン FTBMOV によるデータの転送は、転送元、転送先のインデックスのリストベクトルをあらかじめ作成しておく(LFTY, LITY)。それぞれのリストベクトルの有効長(転送語数)は $LCOLS$ になる。このブロックで TEMBY(k), TEMIY(ℓ, i)も作成する。以下温度データを使用するときは、境界に関しては TEMBY, 全メッシュ点

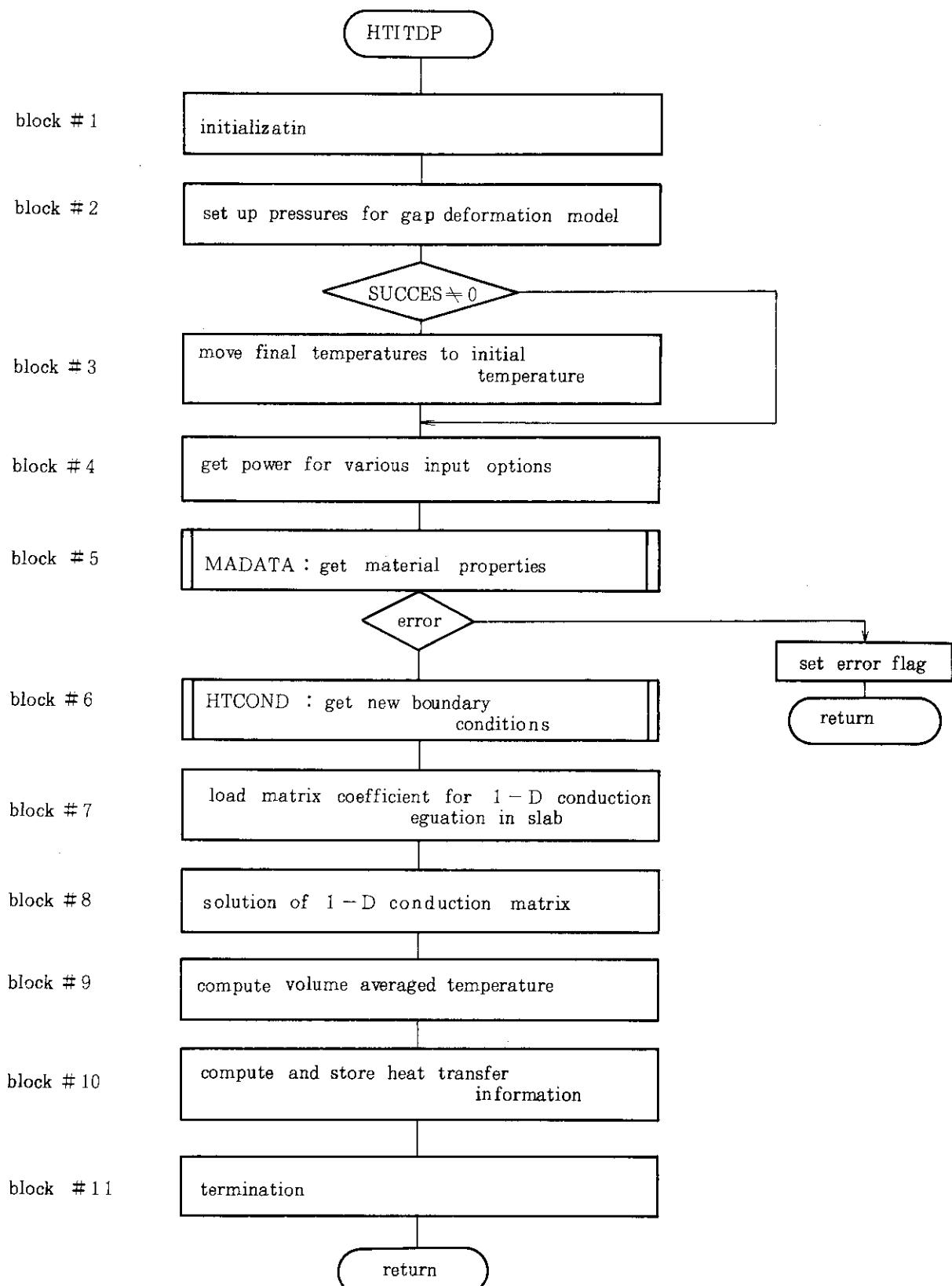


Fig. 6.8 Flow diagram of subroutine HT1TDP

に関してはTEMI¥を使用する。

(4) ブロック4でのパワー計算のために以下の4ケースのリストベクトルをあらかじめ作成しておく。

- i) LPOW 1¥ : STYP¥(i) = 0
- ii) LPOW 2¥ : $0 < \text{STYP}¥(i) < 100$
- iii) LPOW 3¥ : $100 \leq \text{STYP}¥(i) \leq 102$
- iv) LPOW 4¥ : $\text{STYP}¥(i) > 102$

ローカル変数POWINは、以下のように配列する。

$$\cdot \text{POWIN} \rightarrow \text{POWIN}¥(i)$$

(5) サブルーチンMADATAの内部に熱構造体ループを組込む。

(6) サブルーチンHTCONDの内部に熱構造体ループを組込む。

(7) 一次元熱伝導方程式を解くための領域を以下のように単純化する。

- HTE $\rightarrow \text{HTE}¥(\ell^*, i)$
- HTB $\rightarrow \text{HTB}¥(\ell, i)$
- HTF $\rightarrow \text{HTF}¥(\ell, i)$
- HTTC $\rightarrow \text{HTT}¥(\ell, i)$
- HTVHC $\rightarrow \text{HTV}¥(\ell, i)$

$$\text{ここで } \ell^* = 1, \text{ COLS}¥(i) + 1$$

$$\ell = 1, \text{ COLS}¥(i)$$

$$i = 1, \text{ } ¥NH$$

メッシュ間及び内部メッシュ点での値の計算は二重DOループになるが、リストベクトルLNN¥及びLIC¥を使用して一重ループにする。

境界での行列の係数の計算のために、以下の2ケースのリストベクトルをあらかじめ作成しておく。

- i) LFLG 1¥ : IFLG¥(k) = 1
- ii) LFLG 4¥ : IFLG¥(k) = 2又は3

ローカル変数GGO及びGGNを以下のように配列化する。

$$\left. \begin{array}{l} \cdot \text{GGO} \\ \cdot \text{GGN} \end{array} \right\} \rightarrow \text{GG}¥(k)$$

オリジナルプログラムでは係数行列の対称性($a_j = c_{j-1}$)を利用して係数aとcは同じ領域HTEを使用している。但し、壁面温度を陽に指定する境界条件(ITYPE = 1)では、 c_i 及び a_M が0になり対称性を壊している。そのためにHTEはメッシュ方向に1つだけ余分に領域が必要となる。ベクトル化版でも同じように領域を使用する。Fig. 6.9に領域の使用方法を示す。

$$\begin{array}{ccccccccc}
 & j = 1 & 2 & 3 & \cdots & M-1 & M & M+1 & (M = \text{COLS} \#(i)) \\
 \text{HTE} \# & \left\{ \begin{array}{cccccc} a_2 & a_3 & \cdots & a_{M-1} & & a_M \\ c_1 & c_2 & \cdots & c_{M-2} & c_{M-1} & \end{array} \right. \\
 \text{HTB} \# & b_1 & b_2 & b_3 & \cdots & & b_M \\
 \text{HTF} \# & d_1 & d_2 & d_3 & \cdots & & d_M
 \end{array}$$

但し一般形を $a_j T_{j-1} + b_j T_j + c_j T_{j+1} = d_j$ としたとき

Fig. 6.9 Use of area for coefficient matrix

- (8) ブロック 8 では、一次元熱伝導方程式をガウスの消去法を用いて解いているため、メッシュ方向にはベクトル化はできない。そのため、熱構造体ループに関してのみベクトル化する。但し、メッシュ数は熱構造体ごとに異なるため、単純に DO ループを入れることはできない。そこで、メッシュごとの熱構造体の配列 (NHM#) を作成し熱構造体ループの終値とする。熱構造体は、メッシュ数の多い順にソート（実際はインデックスソート。インデックスは NHS# に格納）しておく。外側のメッシュループの終値は、メッシュ間数の最大値 #MNN (NN#(i) の最大値) になる。この方法の概念図を Fig. 6.10 に示す。また、Fig. 6.11 にこのブロックの DO ループ入替え前後のプログラムを示す。
- (9) ブロック 9 では、AVWT#(l, i) と TEMF#(l, i) との内積を HTVATP(i) へ格納するため二重ループとなる。この二重ループを一重化すると完全に再帰的になるため、二重ループのままで内側のメッシュループのみベクトル化する。
- (10) 热輸送に関する諸量の設定のために、以下の 3 ケースのリストベクトルをあらかじめ作成しておく。
- i) LFLG 1# : IFLG#(k) = 1
 - ii) LFLG 2# : IFLG#(k) = 2
 - iii) LFLG 3# : IFLG#(k) = 3
- (11) 時刻、タイムステップ、壁温の設定に加えて、作業領域 (TEMF#) に設定した最終の温度値をオリジナルの領域に転送する。

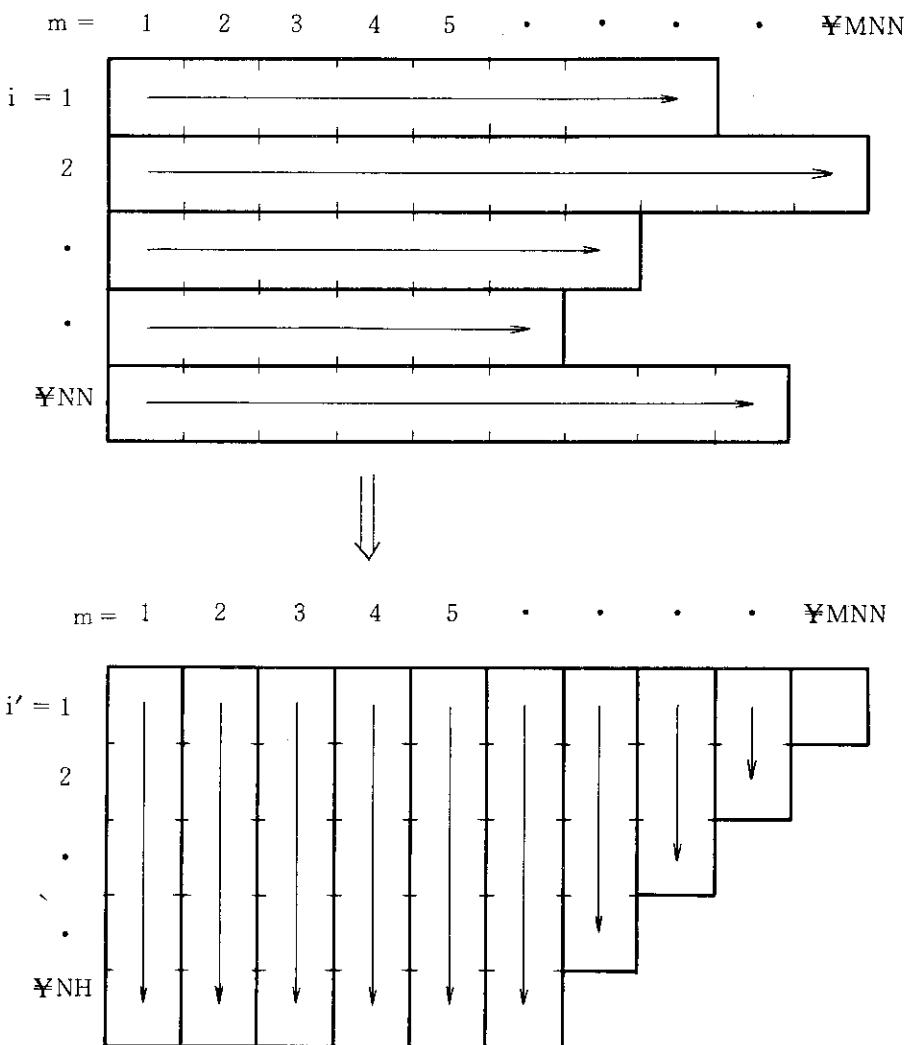


Fig. 6.10 Method for DO-loop interchange

6.2.3 MADATA

MADATAで使用する内挿データテーブルの大きさ、位置等のデータは、あらかじめHTADVで作成しておく。それらは、メッシュ間数と熱構造体数の二次元配列になるが、二重ループで使用することはないので、詰めた形で一次元化してある。

(1) 热伝導率の計算

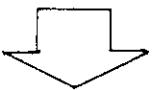
热伝導率フラグ (th.c.flag) 及びテーブル長 (t.length) の値によって、以下の4ケースのリストベクトルをあらかじめ作成しておく。

- i) LKSW1 ¥ : th.c.flag = 0かつ t.length ≥ 2
- ii) LKSW2 ¥ : th.c.flag = 0かつ t.length = 1
- iii) LKSW3 ¥ : th.c.flag = 1
- iv) LKSW4 ¥ : th.c.flag = 2

i)のケースでのテーブルサーチの始点は、前回の値は用いず毎回1より始める。iv)のケー

```

C
C FORWARD ELIMINATION
C
DO 3000 I = 1, *NH
    DEN = 1.000/HTB*(1,I)
    HTF*(1,I) = HTF*(1,I)*DEN
    HTE*(1,I) = HTE*(1,I)*DEN
3000 CONTINUE
C
DO 3220 I = 1, *NH
    DO 3200 M = 2, NN*(I)
        DEN = 1.000/( HTB*(M,I)
                        - HTE*(M,I)*HTE*(M-1,I) )
        HTF*(M,I) = ( HTF*(M,I)
                        - HTE*(M,I)*HTF*(M-1,I) )*DEN
        HTE*(M,I) = HTE*(M+1,I)*DEN
3200 CONTINUE
3220 CONTINUE
C
C BACK SUBSTITUTION
C
DO 3400 I = 1, *NH
    M = COLS*(I)
    TEMF*(M,I) = ( HTF*(M,I) - HTE*(M+1,I)*HTF*(M-1,I) )
                  / ( HTB*(M,I) - HTE*(M+1,I)*HTE*(M-1,I) )
3400 CONTINUE
C
DO 3620 I = 1, *NH
    DO 3600 M = NN*(I), 1, -1
        TEMF*(M,I) = HTF*(M,I)
                      - HTE*(M,I)*TEMF*(M+1,I)
3600 CONTINUE
3620 CONTINUE


C
C FORWARD ELIMINATION
C
DO 3000 I = 1, *NH
    DEN = 1.000/HTB*(1,I)
    HTF*(1,I) = HTF*(1,I)*DEN
    HTE*(1,I) = HTE*(1,I)*DEN
3000 CONTINUE
C
DO 3220 M = 2, *MNN
    DO 3200 I = 1, NHM*(M)
        DEN = 1.000/( HTB*(M,NHS*(I))
                        - HTE*(M,NHS*(I))*HTE*(M-1,NHS*(I)) )
        HTF*(M,NHS*(I)) = ( HTF*(M,NHS*(I))
                        - HTE*(M,NHS*(I))*HTF*(M-1,NHS*(I)) )*DEN
        HTE*(M,NHS*(I)) = HTE*(M+1,NHS*(I))*DEN
3200 CONTINUE
3220 CONTINUE
C
C BACK SUBSTITUTION
C
DO 3400 I = 1, *NH
    M = COLS*(I)
    TEMF*(M,I) = ( HTF*(M,I) - HTE*(M+1,I)*HTF*(M-1,I) )
                  / ( HTB*(M,I) - HTE*(M+1,I)*HTE*(M-1,I) )
3400 CONTINUE
C
DO 3620 M = *MNN, 1, -1
    DO 3600 I = 1, NHM*(M)
        TEMF*(M,NHS*(I)) = HTF*(M,NHS*(I))
                        - HTE*(M,NHS*(I))*TEMF*(M+1,NHS*(I))
3600 CONTINUE
3620 CONTINUE

```

Fig. 6.11 DO-loop interchange of Gaussian elimination

スでの下位ルーチン GASTHC 及び GAPCON は、以下の 2 つの理由によりベクトル化は行わない。

- ・このモデルは燃料棒を模擬した熱構造体のギャップ部分に使用するため、体系が大きくなつても数は増えない。
- ・これらのサブルーチンを完全にベクトル化するためには、[メッシュ数 (~ 200)] \times [熱構造体数 (~ 100)] \times [ガスの種類 (= 7)] の大きさの配列のデータをいくつか必要とする。

(2) 热容量係数の計算

热容量係数フラグ (v.h.c.flag) 及びテーブル長によって以下の 3 ケースのリストベクトルをあらかじめ作成しておく。

- i) LCSW1 : v.h.c.flag = 0かつ|t.length| ≥ 2
- ii) LCSW2 : v.h.c.flag = 0かつ t.length = 1
- iii) LCSW3 : v.h.c.flag = 1

i)のケースでは、テーブル長が負のときも温度のテーブルサーチを行う。

6.2.4 HTCOND

Fig. 6.12 にオリジナルサブルーチンの概略フローを示す。このフローよりわかるように分岐のキーとなるのは ITYPE という変数で、これは境界条件の型を指定するものであり、以下の意味を持つ。

- ITYPE = 0 : 対称または断熱近似

$$-k \frac{\partial T}{\partial x} = 0 \quad k : \text{熱伝導率}$$

- ITYPE = 1 : 対流

$$-k \frac{\partial T}{\partial x} = h (T - T_B)$$

シンク温度 T_B は接続している流体ボリュームの温度を使用。熱伝達係数 h は相関式 (HTRC1 を使用) より得る。

- ITYPE = 2 : 温度指定

$$T = \begin{cases} 0 \\ T_B \\ T_T(t) \end{cases}$$

壁温をボリュームの温度またはテーブルより得る。

- ITYPE = 3 : 温度指定

$$T = T_T(t)$$

壁温を時間の関数としてテーブルより得る。

- ITYPE = 4 : 伝熱

$$-k \frac{\partial T}{\partial x} = q_T(t)$$

熱流束 q を時間の関数としてテーブルより得る。

- ITYPE = 5 : 対流

$$-k \frac{\partial T}{\partial x} = h_T(t) (T - T_B)$$

熱伝達係数 h を時間の関数としてテーブルより得る。 T_B は流体ボリュームより得る。

- ITYPE = 6 : 対流

$$-k \frac{\partial T}{\partial x} = h_T(T) (T - T_B)$$

熱伝達係数 h を壁温の関数としてテーブルより得る。 T_B は流体ボリュームより得る。このサブルーチンの出力となっている IFLAG は、ITYPE の値より一意に決まるので一回だけ設定する。

- IFLAG = 1 : 壁温直接指定

ITYPE = 2 又は 3

- IFLAG = 2 : 伝熱

ITYPE = 0 又は 4

- IFLAG = 3 : 対流

ITYPE = 1 又は 5 又は 6

ITYPE (配列名 ITYPY) の値に応じて、以下の 4 ケースのリストベクトルをあらかじめ作成しておく。

- i) LTYP0¥ : ITYPY(k) = 0
- ii) LTYP1¥ : ITYPY(k) = 1
- iii) LTYP2¥ : ITYPY(k) = 2
- iv) LTYP3¥ : ITYPY(k) = 3, 4, 5, 6

ii)のケースではサブルーチン HTRC1 に DO ループを組込む。

6.2.5 HTRC1 及び下位ルーチン

Fig. 6.13 にオリジナルサブルーチンの概略フローを示す。ベクトル化では、このフローの前処理、主要計算、後処理それぞれの箇所にループ構造を組込む。DO ループはHTCOND で使用したリストベクトル LTYP1¥ で制御する。

(1) 前処理

コモン名 HTCOM の変数のうち、下記以外のものを配列化する。

- IV : BINX¥(k) を使用。
- IWRITE : 熱構造体に関係なく定まる。

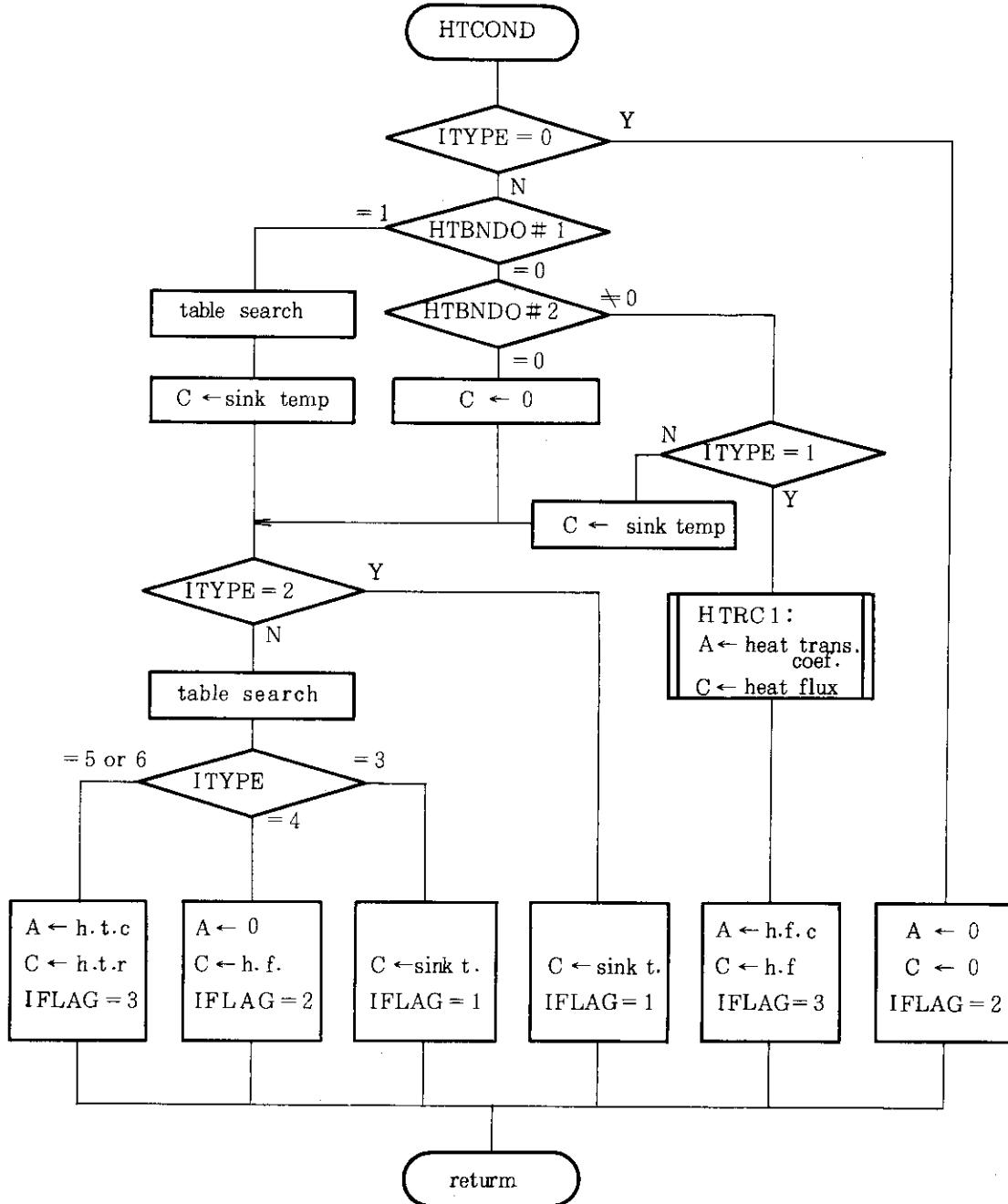


Fig. 6.12 Flow diagram of subroutine HTCOND

- LS : HINXB ¥(k) を使用。
- XEC : DATA 文で値が設定されている。参照のみ
- 未尾に L 又は R の付く変数 : 左右境界を考慮して配列化されている。

前処理では、配列化したデータの初期設定を行う。この箇所でサブルーチン PSATPD を引用しているが、このサブルーチンはベクトル化しない。

(2) 主要計算部分

この箇所では、諸量に応じて各相関計算サブルーチンに振分ける。このループでは各サブル

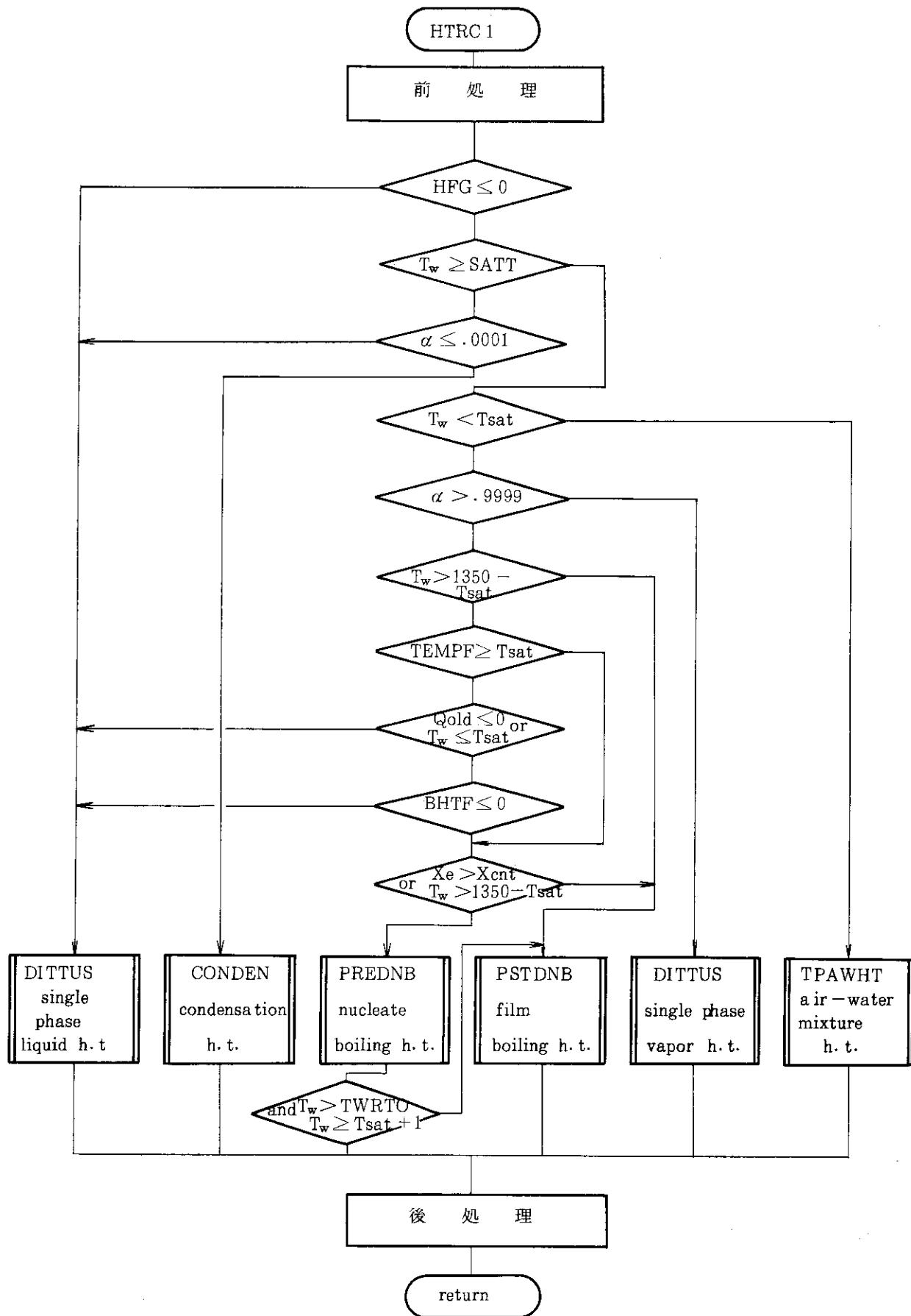


Fig. 6.13 Flow diagram of subroutine HTRC 1

ーチンを引用する代わりに、以下の5種類のリストベクトルを毎回設定する。

- i) LHTR1￥ : DITTUS を引用
- ii) LHTR2￥ : CONDEN を引用
- iii) LHTR3￥ : PREDNB を引用
- iv) LHTR4￥ : PSTDNB を引用
- v) LHTR5￥ : TPAWHT を引用

これらのリストベクトルに従って各サブルーチンを引用する。ここで、各ベクトル長が小さい場合を考え、全く同じサブルーチンのスカラー版を作成し、ベクトル長に応じてスカラー版又はベクトル版どちらかを呼ぶようにする。スカラー版のサブルーチン名は、ベクトル版のサブルーチン名の最後のVをSに置換えたものである。

(3) 後処理

配列データの値をオリジナルの領域に転送する。

(4) DITTUS

このサブルーチンは単純な構造なので全体を1つのDOループにする。

(5) CONDEN

中ほどにWRITE文があるため、それを避けて全体を複数のDOループ構造にする。

(6) PREDNB

前半はベクトル化可能であるが、後半は収束ループがあるためベクトル化できない。下位ルーチンのPSATPDはその場所に展開する。CHFCALはリストベクトルを作成してベクトル化する。

(7) PSTDNB

全体を複数のブロックに分け、それぞれDOループ構造にする。下位ルーチンのSTH2X3, THCON, VISCOGはベクトル化しない。

(8) TPAWHT

全体を複数のブロックに分け、それぞれDOループ構造にする。リストベクトルを作成して下位ルーチンDITTUSを引用する。

6.3 流体力学計算部分のベクトル化

Fig. 5.2のHYDRO以下に示す流体力学計算部分は、サブルーチンごとに独立した構造になっている。各サブルーチンには、ボリューム又はジャンクションに関するDOループがあり、これらをベクトル長としたベクトル化が可能である。

Table 6.4に今回ベクトル化の対象としたサブルーチンを示す。また、Table 6.5にベクトル化は行わなかったが、スカラーレベルでの最適化を行ったサブルーチンを示す。

6.3.1 ACCUM

ACCUMは、アキュムレータにおける質量移動、熱伝導によるエネルギーの移動、及び運動量の式により、液相の流速を計算するサブルーチンである。

このサブルーチン内にはDOループ構造が無く、また、上位ルーチンVEXPLTのコンポー

Table 6.4 Names and functions of vectorized hydrodynamic subroutines

オリジナル版でのルーチン名	ベクトル化版でのルーチン名	内容	主要ベクトル長
EQFINL	EQFINL	圧力、内部エネルギー、ボイド率、非凝縮ガス、ボロン濃度及び蒸気発生率の計算	ボリューム数 ジャンクション数
FIDIS	FIDISV	気泡又は水滴の相間抗力の計算	ボリューム数
FWDRAG	FWDRAG	壁摩擦係数の計算	ボリューム数
HIFBUB	HIFBUV	気泡流での液相熱伝導係数の計算	ボリューム数
HLOSS	HLOSS	圧力損失項の計算	ジャンクション数
HTHETA	HTHETV	水平流路における気・液の層の角度計算	ボリューム数
JPROP	JPROP	ジャンクションの各特性値の計算	ジャンクション数
PHAINT	PHAINT	相間摩擦と熱伝導の計算	ボリューム数 ジャンクション数
PRESEQ	PRESEQ	圧力の方程式の係数行列と定数ベクトルを求める	ボリューム数 ジャンクション数
SYSSOL	SYSSOL	直接法による圧力マトリックスの計算	マトリックスの非ゼロ要素数
VEXPLT	VEXPLT	相別の流速及び圧力の変化率の計算	ジャンクション数
VFINL	VFINL	新時刻の流速及び質量流量の計算	ジャンクション数
VLVELA	VLVELA	ジャンクション流速からボリューム流速を計算	ボリューム数
VOLVEL	VOLVEL	ジャンクション流速からボリューム流速を計算	ジャンクション数

Table 6.5 Names and functions of optimized hydrodynamic subroutines

ルーチン名	内容
ACCUM	アキュムレータにおける液相の流速を計算
HZFLOW	水平分離流における液滴のエントレインメントの計算
JCHOKE	臨界流の計算
PACKER	ウォータ・パッキング時の計算
PUMP	ポンプの圧力変化とトルクの計算
PUMP 2	ポンプのトルクとヘッドの内挿
PMINV 2	実数データの指数部取出し
PMINV 3	実数データの指数部取出し
PMINV 5	2の巾乗の計算
VALVE	バルブのふるまいの計算

ネットループを取込んでも、平均ベクトル長はわずか 2 である。従って、ベクトル化は行わずに、ビット操作ファンクションの変更のみを行う。

6.3.2 EQFINL

EQFINL は、新時刻における圧力、単位体積当りの相別内部エネルギー、ボイド率、非凝縮性ガス、ボロン濃度、及び蒸気発生率の計算を行う。

このサブルーチンは、ビット操作ファンクションを組込み関数に変更すれば、ベクトル処理が可能となる。ただし、ジャンクション・ループにおけるボロン濃度（変数名 BORON）の計算で隣接ボリュームのインデックス KXV 及び LTV を用いている部分が再帰演算となるためこの部分はスカラー処理とする。本サブルーチンの構造を Fig. 6.14 に示す。

6.3.3 FIDIS

FIDIS は、サブルーチン PHAINT から呼出され、仮引数 k の値によって気泡又は水滴の相間抗力を計算する。ただし、 $k = 0$ のときは気泡を、 $k = 1$ のときは水滴を扱う。

このサブルーチンには DO ループ構造が存在しないため、上位ルーチンの PHAINT からボリュームに関するループを取り込む。ただし、ベクトル長が小さい場合には、オリジナルのスカラーバージョンを呼び出す必要があるので、ベクトル化版のサブルーチン名は FIDISV とする。ベクトル化版の概略フローを Fig. 6.15 に示す。

6.3.4 FWDRAG

FWDRAG は、壁摩擦係数の計算を行う。本サブルーチンは Fig. 6.16 に示す様に、ボリュームに関するループで構成されている。

このサブルーチンのベクトル化版の構造は、基本的にはオリジナル構造と同一である。ただし、壁摩擦係数の良否のフラグ BADFW が、DO ループ内で定義され、かつ DO ループ外で参照される論理型スカラー変数であるため、ベクトル処理の障害となっている。従って、この DO ループ内の定義をベクトル処理が可能な型にする⁴⁾。

6.3.5 HIFBUB

HIFBUB は、気泡流における単位体積当りの液相熱伝導係数の計算を行う。オリジナルサブルーチンは、DO ループ構造を全く持たない。従って、このサブルーチンのベクトル化では、上位ルーチンの PHAINT からボリュームに関する DO ループを取り込んで、全ボリュームについて一度に計算する構造に変更する。（Fig. 6.17 参照）

尚、本サブルーチンはベクトル長により、オリジナル版とベクトル化版を使い分ける必要があるため、ベクトル化版のサブルーチン名を HIFBUV とする。

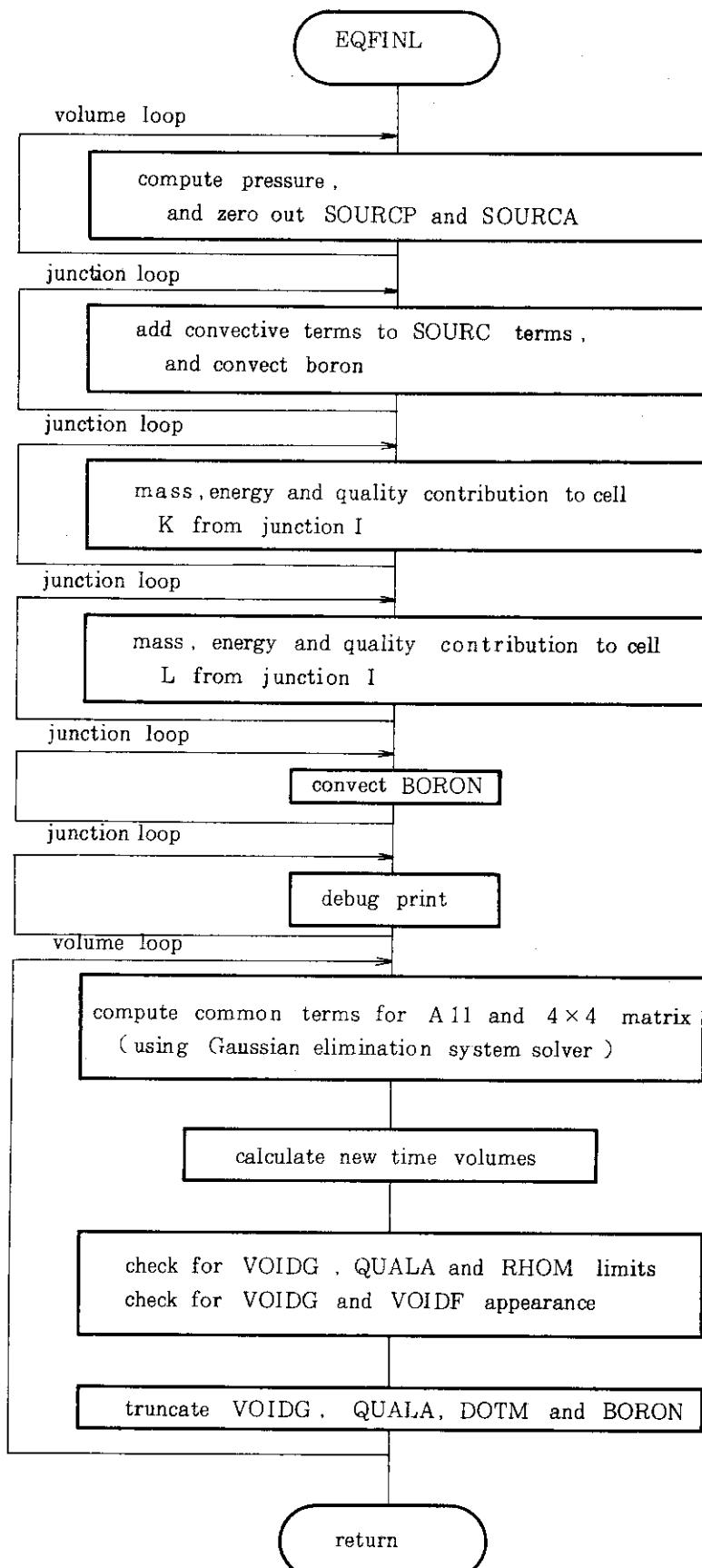


Fig. 6.14 Flow diagram of subroutine EQFINL

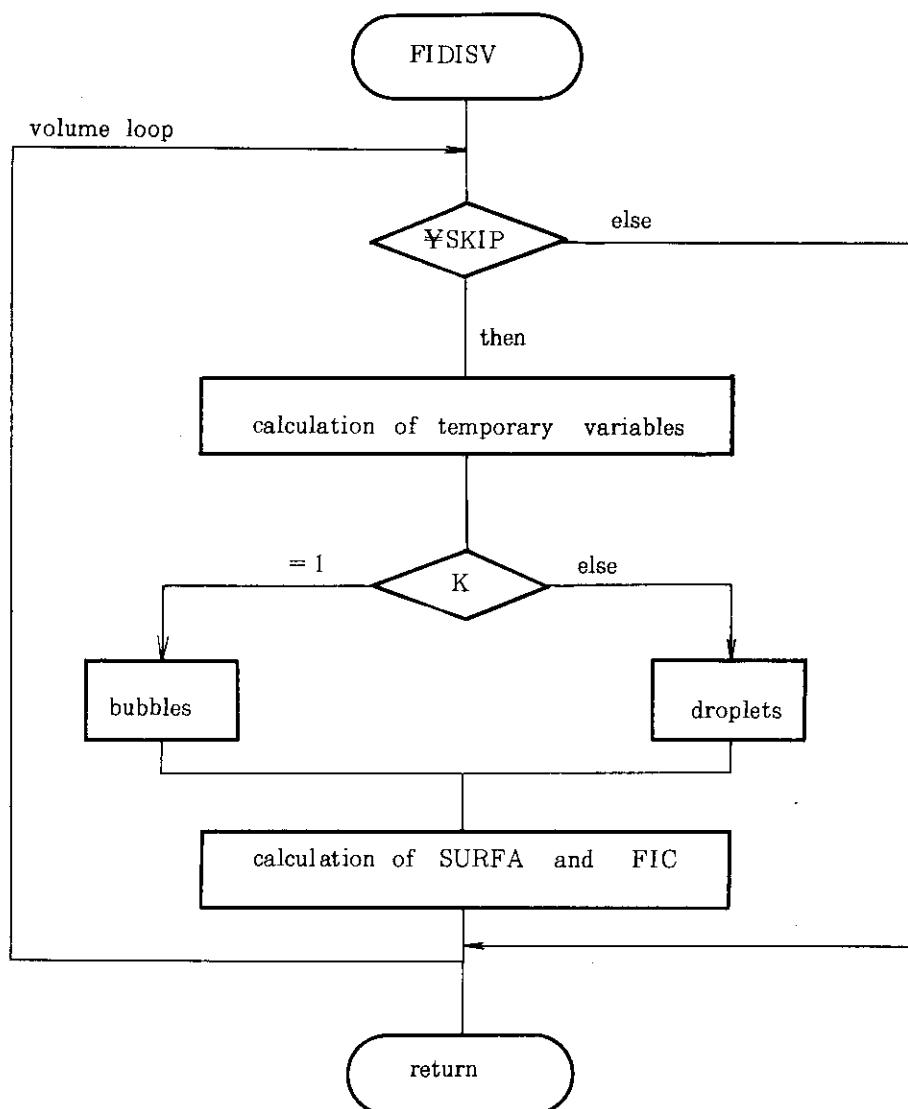


Fig. 6.15 Flow diagram of subroutine FIDIS in vector version

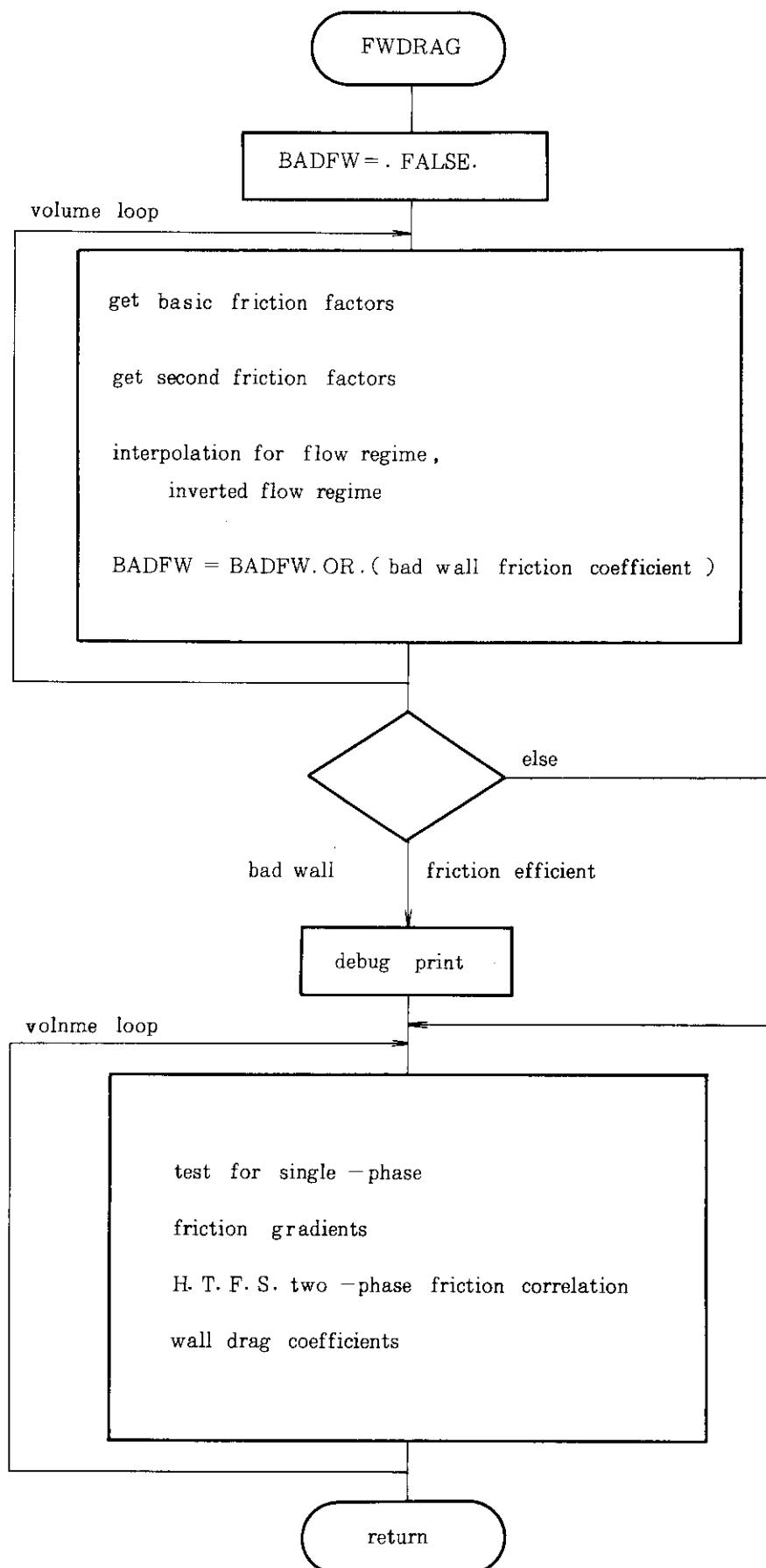


Fig. 6.16 Flow diagram of subroutine FWDRAG

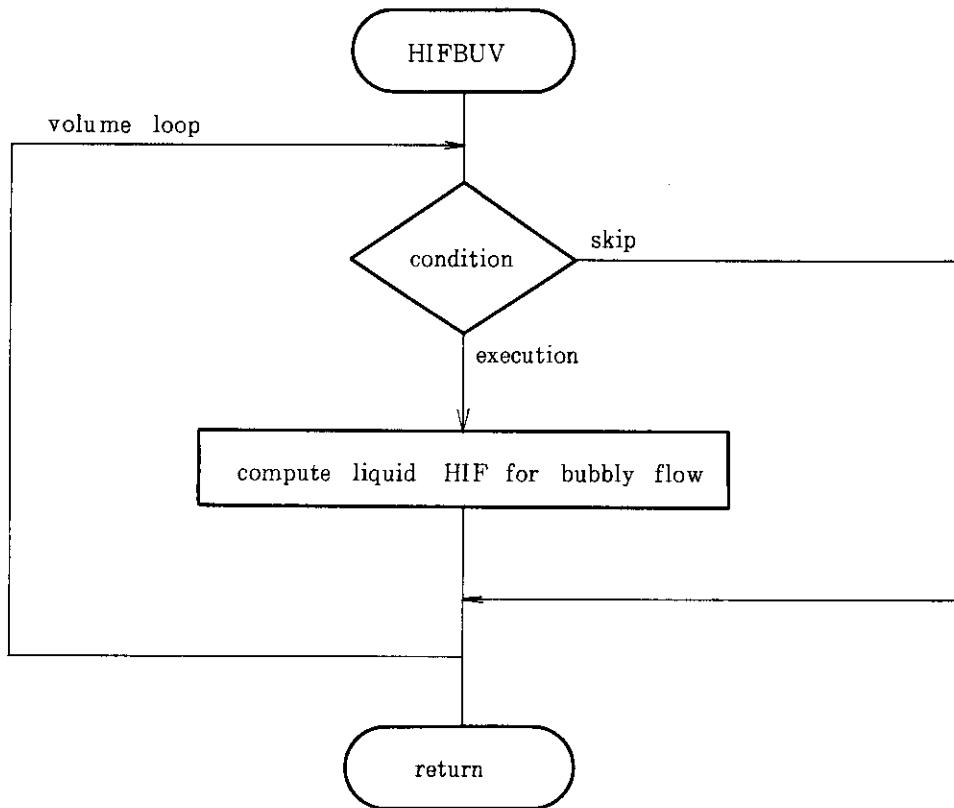


Fig. 6.17 Flow diagram of subroutine HIFBUV in vector version

6.3.6 HLOSS

HLOSSは、絞り部や流路の拡大部、縮小部における圧力損失項を計算するサブルーチンである。FORTUNE 実行結果によると、実際に計算対象となるジャンクションは非常に少く、先頭の IF 文でほとんどがはずされている。そこでサブルーチン全体を 2 つの DO ループに分け、接続ボリュームのインデックスを計算する部分についてのみベクトル化する。このループの中で、計算対象となるジャンクションインデックスを集め、リストベクトルを作成する。圧力損失項を計算する部分は、このリストベクトルによる DO ループとする。この DO ループは、ベクトル長が小さいのでスカラーのままとする。Fig. 6.18 に概略フローを示す。

6.3.7 HTHETA

水平な流路における低速流は、気相と液相の各々の密度差による浮力によって層を成す。HTHETAは、この層の角度を求めるファンクション・ルーチンであり、PHAINITの下位ルーチンである。オリジナルでは、二分割法による収束計算を行う DO ループ（最大ベクトル長が 16 及び 17）が存在するが、再帰演算のためベクトル処理は行えない。

本ルーチンのベクトル化では、ファンクション・ルーチンをサブルーチンに変更し PHAINIT

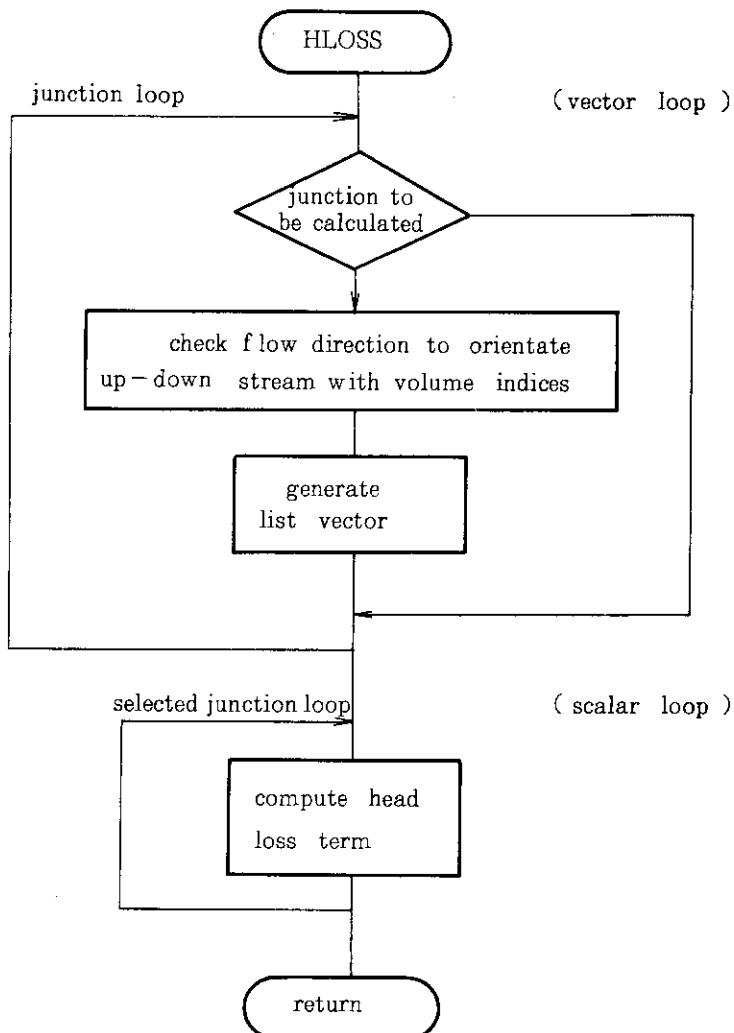


Fig. 6.18 Flow diagram of subroutine HLOSS

からボリュームに関するDO ループを取込む。このボリュームについてベクトル化を行う。収束ループは陽に書下しを行う。(Fig. 6.19 参照)

尚、本ルーチンはPHAINIT以外からも呼出されるので、ベクトル化版のサブルーチン名はHTHETVとする。

6.3.8 HZFLOW

HZFLOWは、水平分離流における液滴のエントレインメントの計算を行うサブルーチンである。ベクトル長が小さくベクトル化による効果が期待できないのでスカラー最適化を行うのみとする。

6.3.9 JCCHOKE

JCCHOKEは臨界流の計算を行うサブルーチンであるが、音速値の定義、引用関係が再帰的であるためベクトル化不可能である。よって、スカラー最適化のみとする。

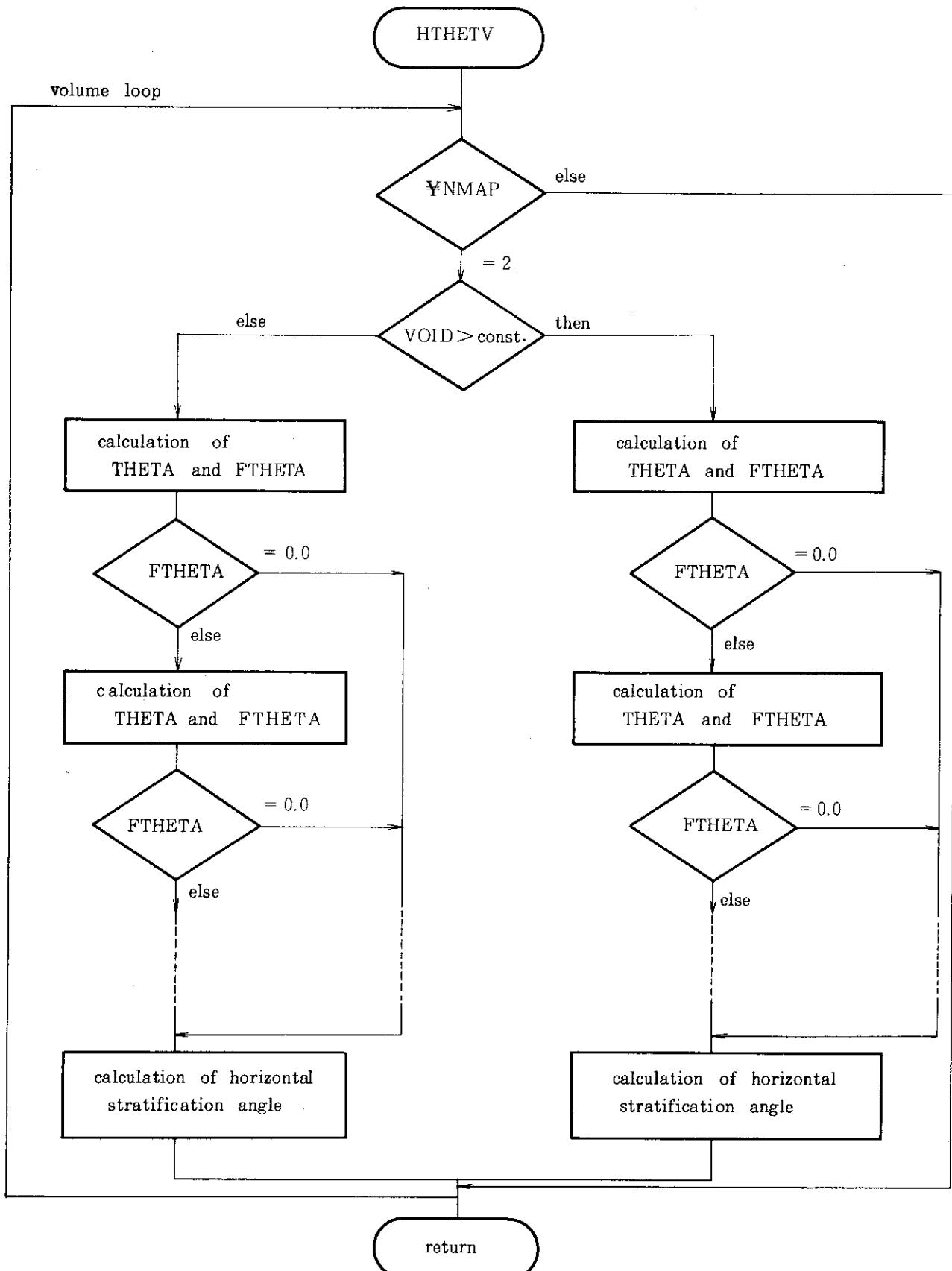


Fig. 6.19 Flow diagram of subroutine HTHETA
in vector version

6.3.10 JPROP

JPROPは、ジャンクションの各特性値を計算するサブルーチンである。全体がコンポーネントとジャンクションに関する二重ループとなっていて、内側のDOループの繰返し数は1～3程度である。コンポーネントに関するデータのインデックスを作業用配列に格納しておき、それを使用するように変更する。これによって一重ループにすることができ、ベクトル長も大きくなる(Fig. 6.20参照)。

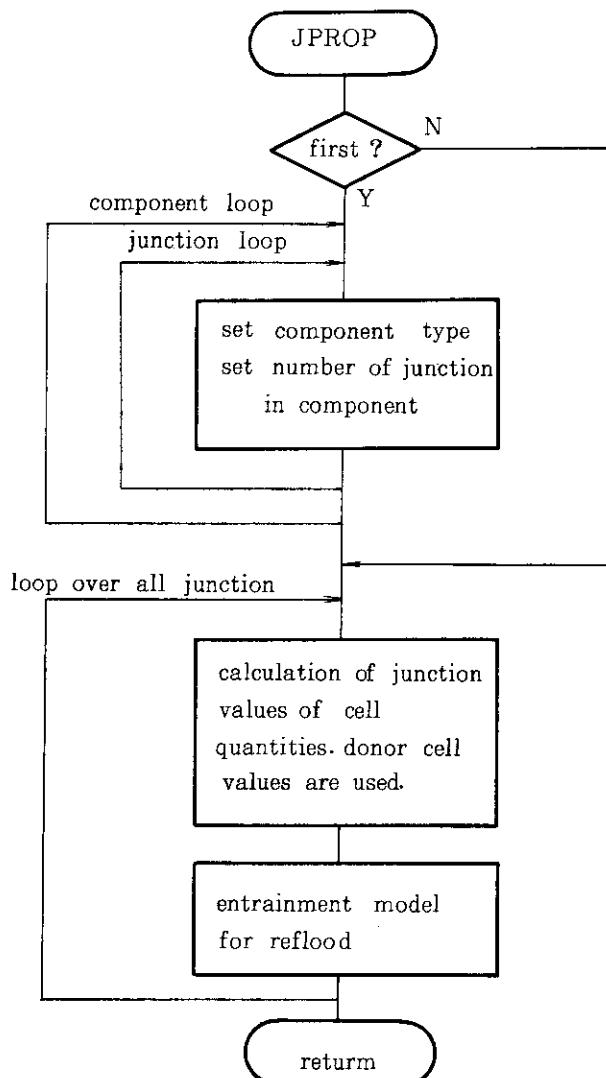


Fig. 6.20 Flow diagram of subroutine JPROP

6.3.11 PACKER

ウォータ・パッキングにおける計算を行うサブルーチンであるが、ベクトル長が小さいのでスカラー計算のままとする。

6.3.12 PHAINT

PHAINTは、RELAP5/MOD1におけるFIDRAG及びMDOTを統合したサブルーチンであり、相間摩擦と熱伝導の計算を行う。オリジナルでは、約1500行の比較的大型なサブルーチンであり、前半はボリュームに関するDOループ、後半はジャンクションに関するDOループで構成されている。このボリューム・ループ内では、下位ルーチン(PSATPD, HTHETA, FIDIS, HIFBUB)の呼出しと接続ジャンクションに関するDOループを含む。概略をFig. 6.21(a)に示す。

本サブルーチンは、ボリューム及びジャンクションの各DOループについてベクトル化する。まず、DOループに含まれる実行文が非常に多いため、これらを適当な計算ブロックに分割する。さらに、ボリュームに関するDOループでは、下位ルーチンの呼出しの前後と接続ジャンクションの二重DOループの前後でDOループを分割する。接続ジャンクションに関するDOループのベクトル長は平均2と非常に小さいので、二重DOループの内側と外側を入れ替える。このためのインデックスは、初めに1度だけ計算して保存しておく。また、このボリューム・ループの内、特にベクトル長の小さい部分(wet wall correlation)はスカラー処理とする。Fig. 6.21(b)にベクトル化版の構造を示す。

6.3.13 PRESEQ

PRESEQでは、独立変数を圧力のみとした方程式の係数行列と定数ベクトルを作成する。これには、液相と気相の単位体積当たりの内部エネルギー、ボイド率及び非凝縮性ガスの各変数を消去するための、質量保存とエネルギーの方程式を使用している。オリジナルサブルーチンの構造をFig. 6.22(a)に示す。

本サブルーチンのベクトル化は、前半のボリュームに関するDOループでは、ビット操作ファンクションを組込み関数に変更するだけでベクトル処理が可能となる。後半のコンポーネントに関するDOループは、内側のジャンクションに関するDOループとの二重構造を一重化する。ただし、各コンポーネント毎のジャンクションのインデックスが再帰的となるため、係数行列(COEFP)の計算部分はスカラー処理とする。Fig. 6.22(b)にベクトル化版の構造を示す。

6.3.14 PUMP, PUMP 2

PUMPは、ポンプの圧力変化とトルクの計算を行う。本サブルーチンには、DOループが存在しない。また、上位ルーチンであるVEXPLTの、コンポーネントに関するDOループの実効ベクトル長が2である。従って、本サブルーチンのベクトル化は行わず、ビット操作ファンクションを組込み関数に変更するのみとする。

PUMP 2は、PUMPの下位ルーチンであり、ポンプのトルク及びヘッドの内挿を行う。PUMPと同様に、ビット操作ファンクションの変更のみを行う。

6.3.15 SYSSOL 及び下位ルーチン

圧力マトリックスを直接法で解くサブルーチンであり、基本的にはベクトル化不可能である。解法を間接法などに変更することによってベクトル化することも考えられるが、計算精度、収

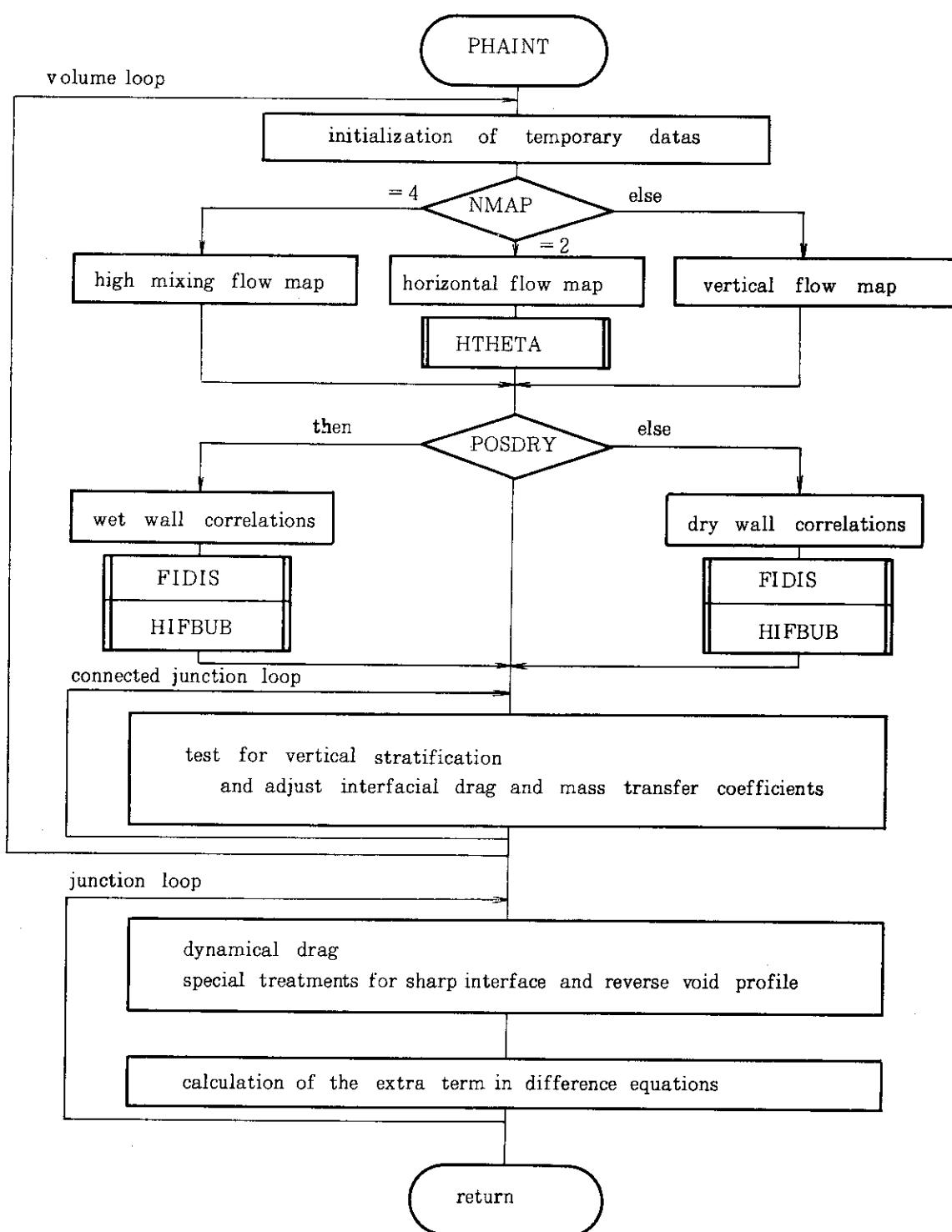


Fig. 6.21(a) Flow diagram of subroutine PHAINT in original version

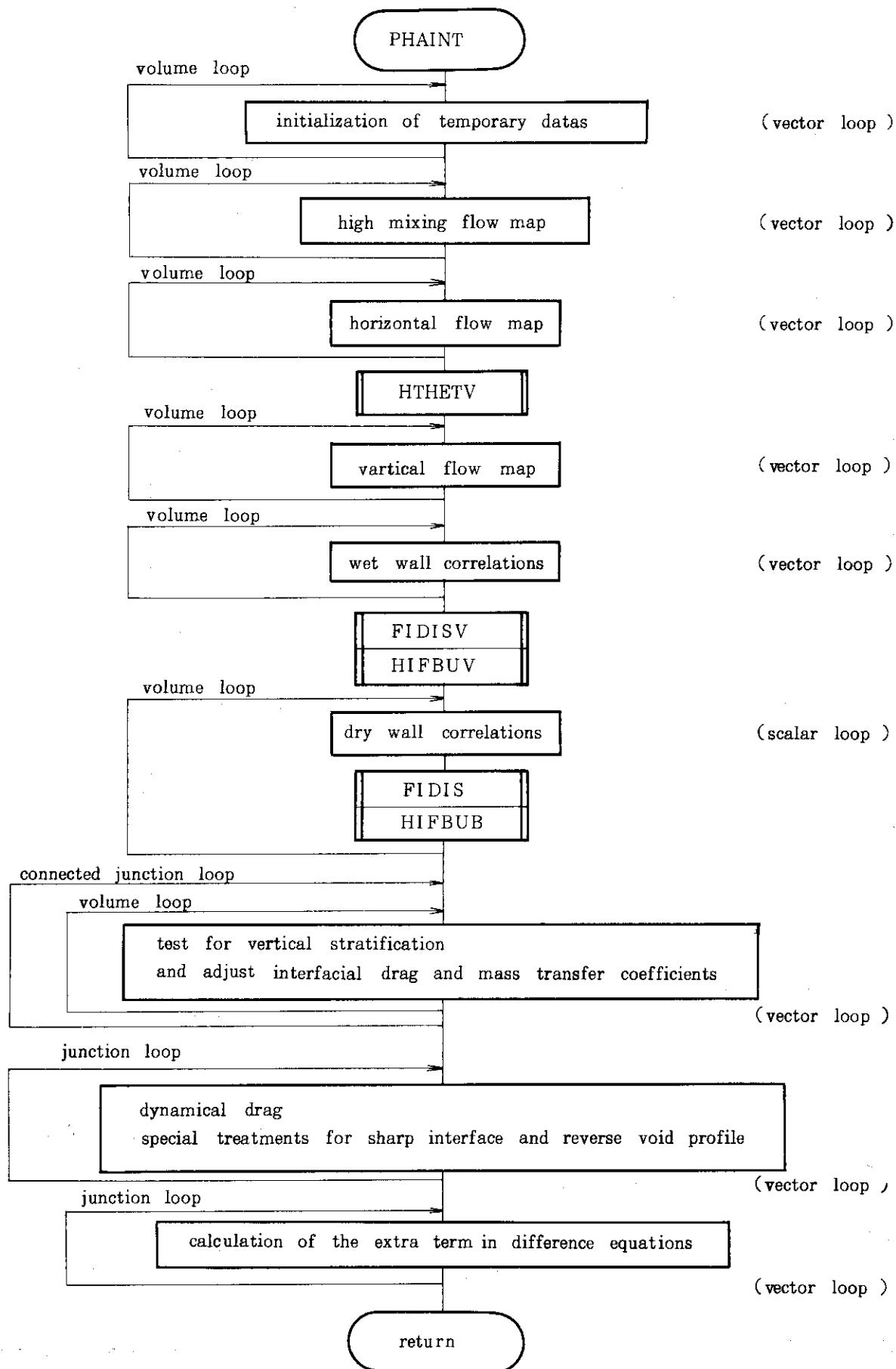


Fig. 6.21(b) Flow diagram of subroutine PHAINT in vector version

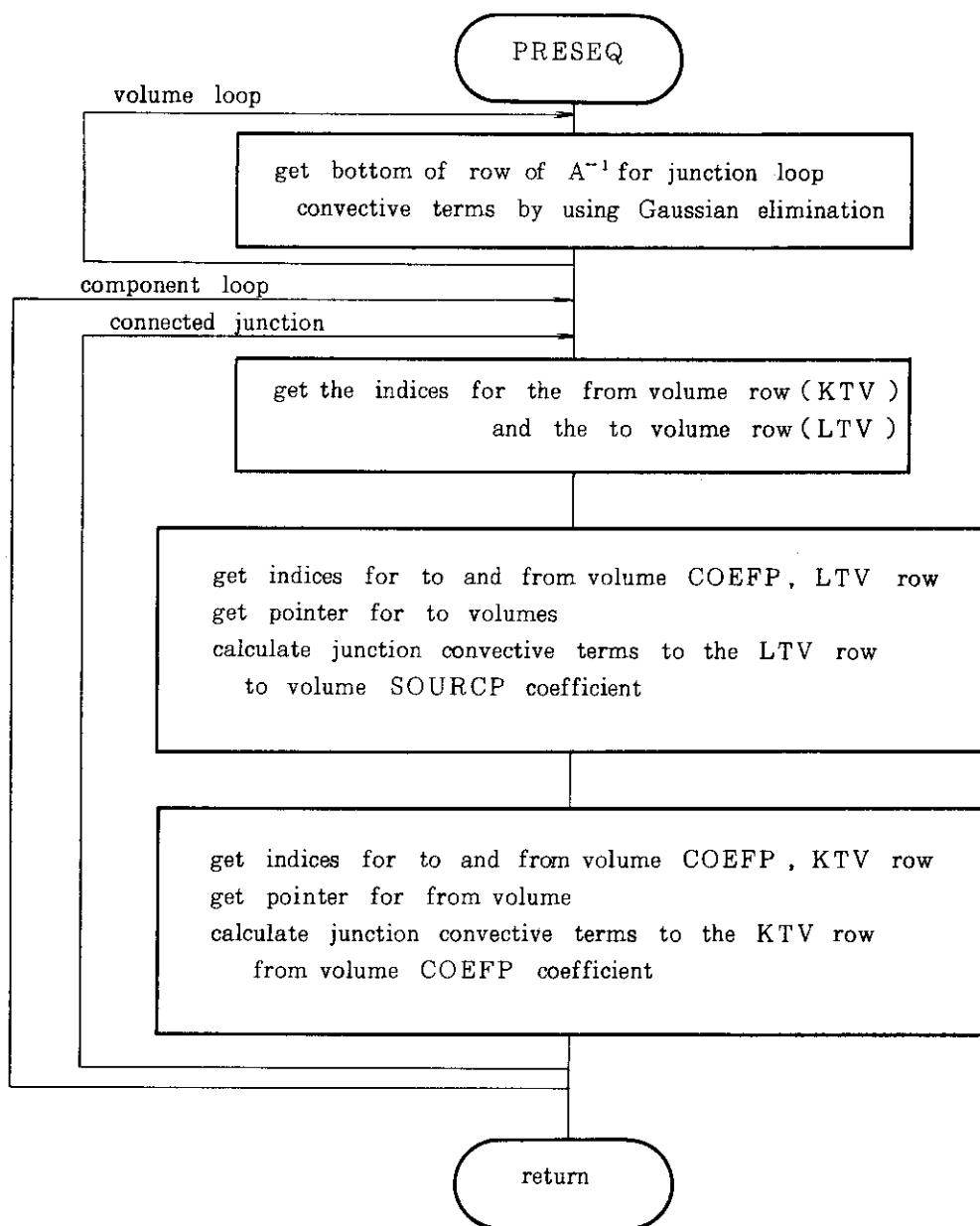


Fig. 6.22(a) Flow diagram of subroutine PRESEQ
in original version

束性等の検討を行っていないので、今回は変更は行わない。インデックス設定など、前処理部分のみベクトル化を行い、その他の部分は、スカラ-最適化を行う。

(1) PMINV 2 及びPMINV 3

実数データの指数部を取出すルーチンで定義は

$$y = \lceil \log_2 |a| + 1 \rceil$$

である。正確な値を求めるため対数計算とせず、実際に 2 の乗算（又は除算）を繰返すことでの指数値を求めている。これを高速化のために計算方法を変更する。組込み関数 ISHFT を用い

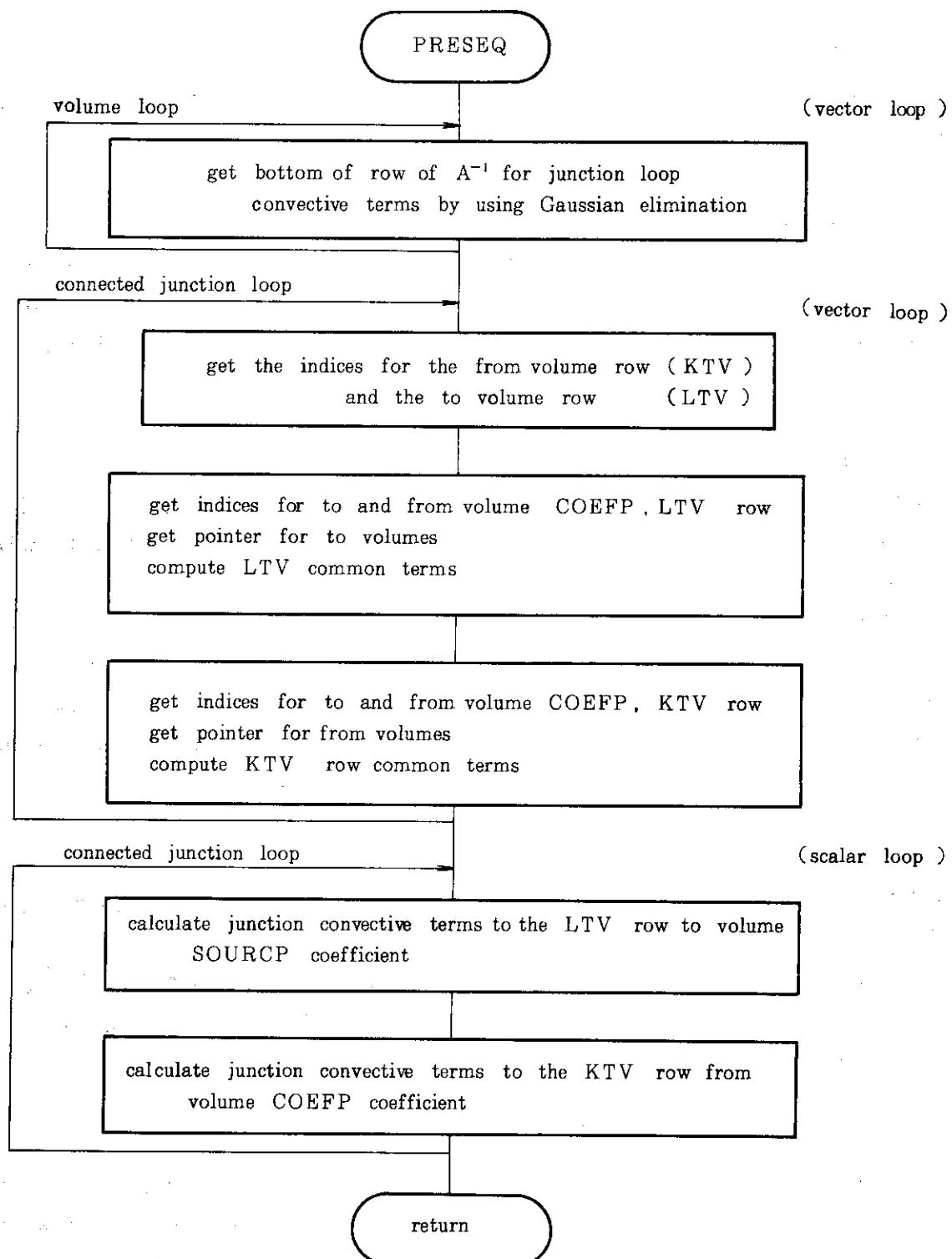


Fig. 6.22(b) Flow diagram of subroutine PRESEQ
in vector version

て16進指数部を取出し4倍する。次に仮数部の値に従って0から3の値を加える。以上の操作で繰返し計算なしに同じ値を取出すことができる。

(2) PMINV5

2の巾乗を計算するファンクションで、非常に簡単なプログラムであるので上位ルーチンに展開する。

6.3.16 VALVE

VALVEでは、バルブのふるまいを計算する。本サブルーチンは、平均ベクトル長97のコンポーネントに関するDOループで構成されているが、処理対象となるコンポーネントがバルブのみであるため実効ベクトル長は4である。従って、ベクトル化は行わずビット操作ファンクションの組込み関数への変更のみを行う。

6.3.17 VEXPLT

VEXPLTでは、相別の流速及び圧力変化率を計算する。また、旧時刻の対流項の計算も行う。オリジナルのサブルーチンはFig. 6.23(a)に示すように、コンポーネントに関するDOループで構成されている。

コンポーネント・ループは、内側にジャンクションに関するDOループを持つため、この二重ループを一重化してベクトル化する。また、下位ルーチンの呼出しや、入出力文等のベクトル処理が行えない文の前後でDOループを分割してベクトル化を行う。ただし、最後の壁摩擦損失の計算部分が再帰演算となるためスカラー処理とする。ベクトル化の構造を、Fig. 6.23(b)に示す。

6.3.18 VFINL

VFINLは、サブルーチンPRESEQ及びSYSSOLの呼出しと、新時刻の流束及び質量流量を計算する。オリジナルの構造をFig. 6.24に示す。

本サブルーチンのベクトル化版の構造は、基本的にはオリジナルの構造と同じである。ただし、ジャンクションに関するデータの退避処理の部分が二重DOループになっているため、内側のDOループを陽に書下す。

6.3.19 VLVELA

ジャンクション流速からボリューム流速を求めるサブルーチンで、外側にボリュームループ、内側にそのボリュームに接続しているジャンクションに関するDOループの二重DOループ構造になっている。内側のDOループの平均ベクトル長は2と非常に小さいので、2つのDOループの順序を入れ替えることを考える。オリジナル版の構造をFig. 6.25(a)に示す。接続ジャンクションの数は、ボリュームごとに異っておりそのままでは入替えはできない。そこで、Fig. 6.26に示す二次元配列を用意し、この中にボリューム別の接続ジャンクションのインデックスを格納する。二重DOループの外側を接続ジャンクション数の最大値、内側をボリューム数によるDOループとすることによって、ボリューム数をベクトル長としたベクトル化を行う(Fig. 6.25

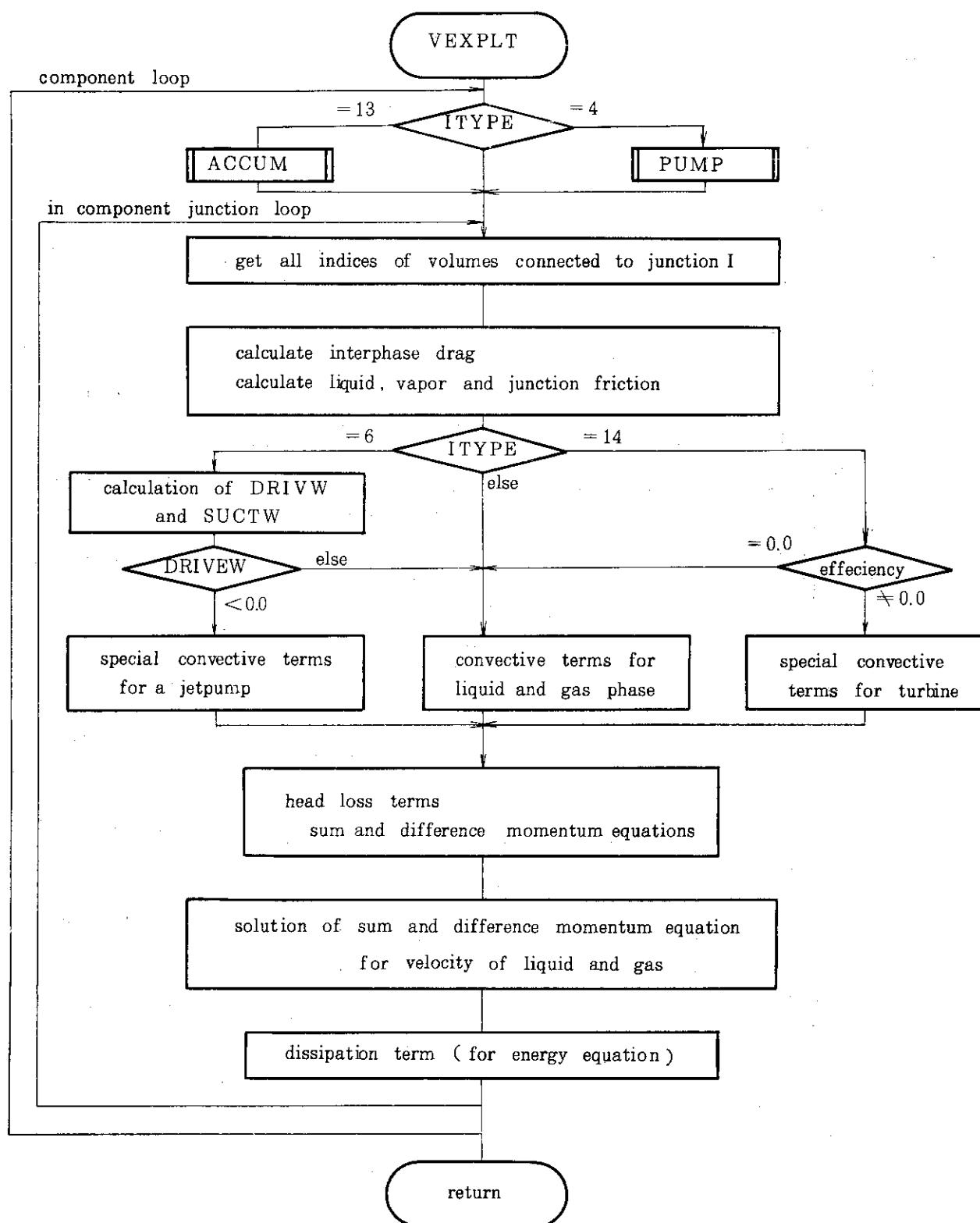


Fig. 6.23(a) Flow diagram of subroutine VEXPLT
in original version

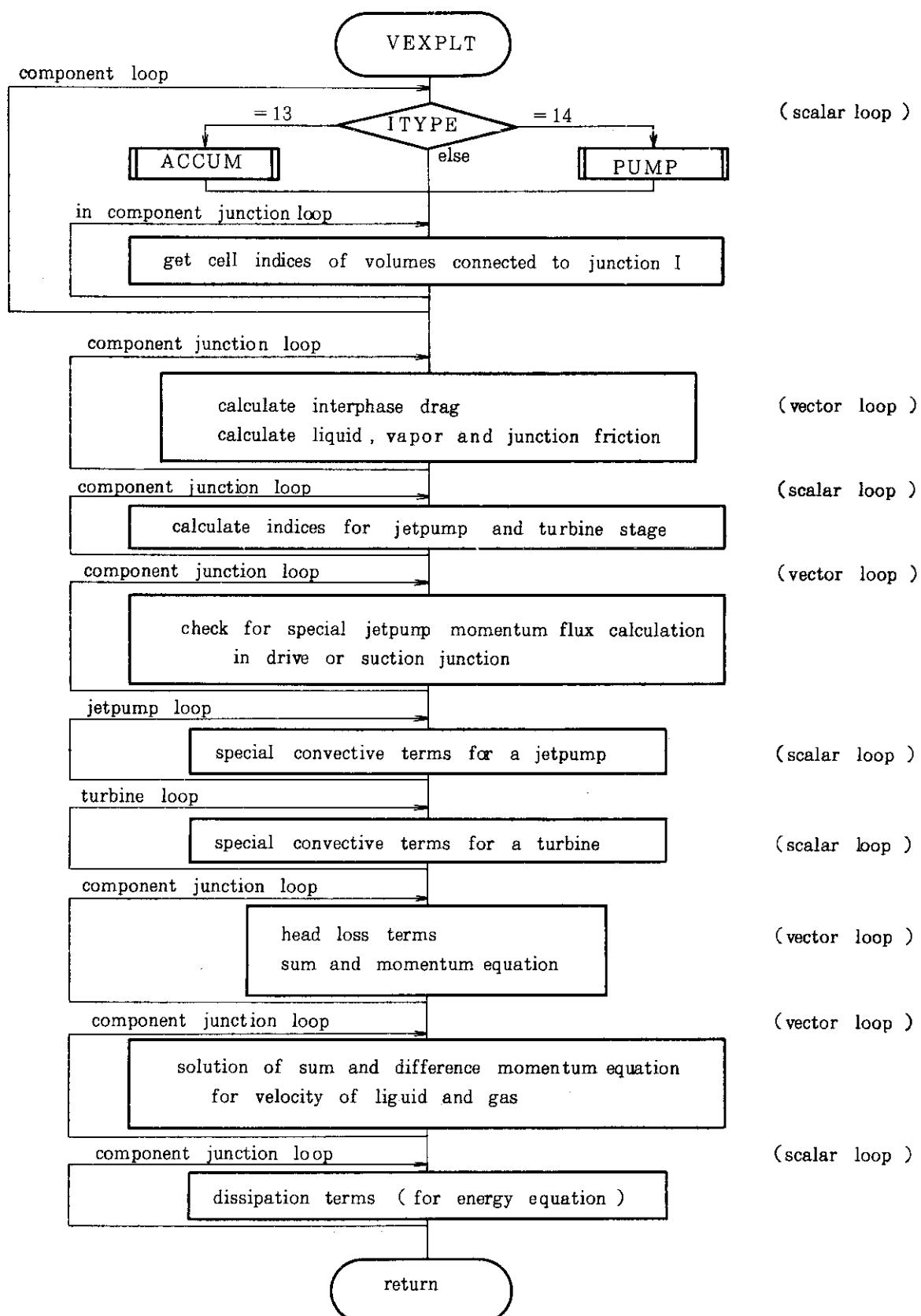


Fig. 6.23(b) Flow diagram of subroutine VEXPLT
in vector version

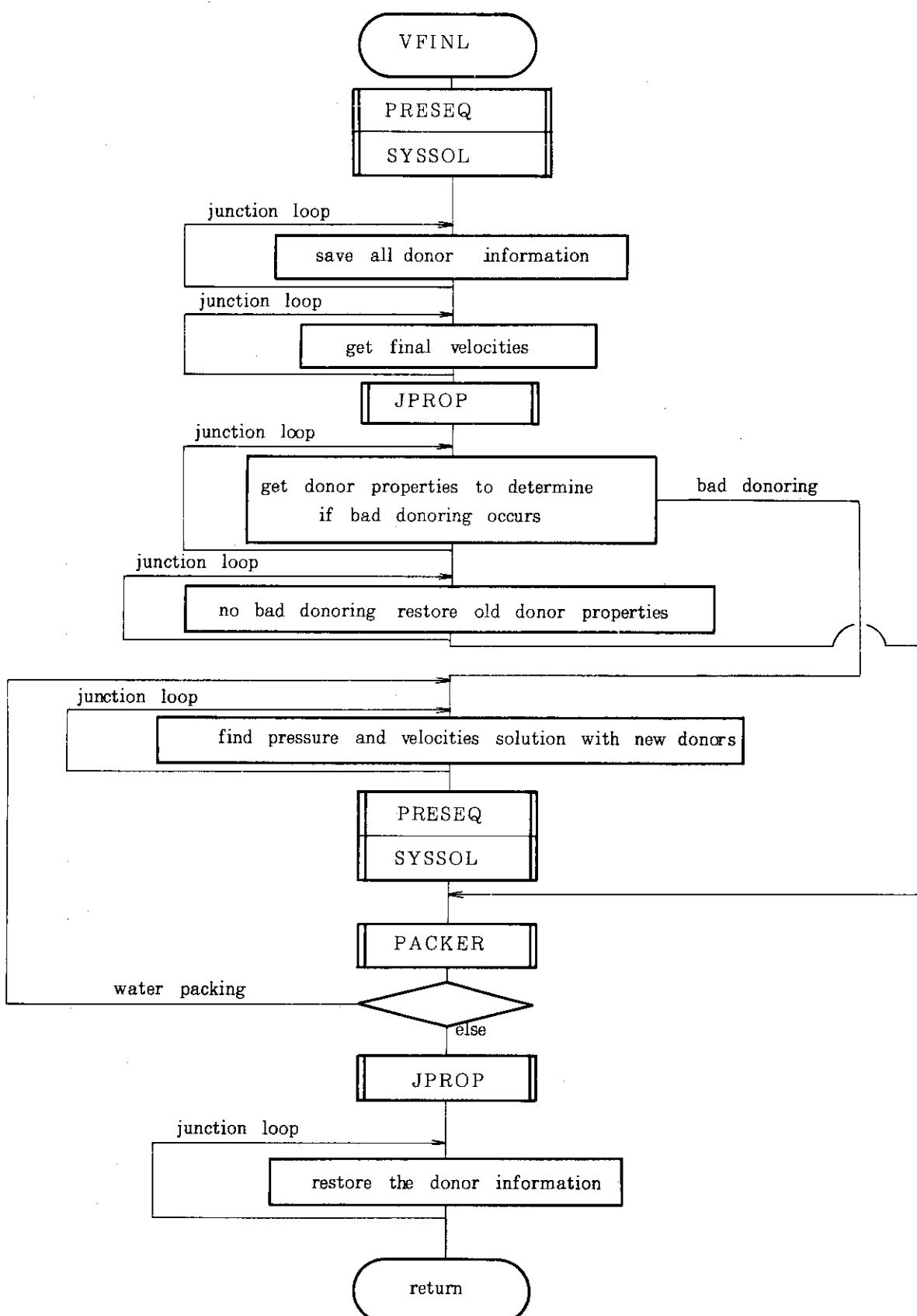


Fig. 6.24 Flow diagram of subroutine VFINL

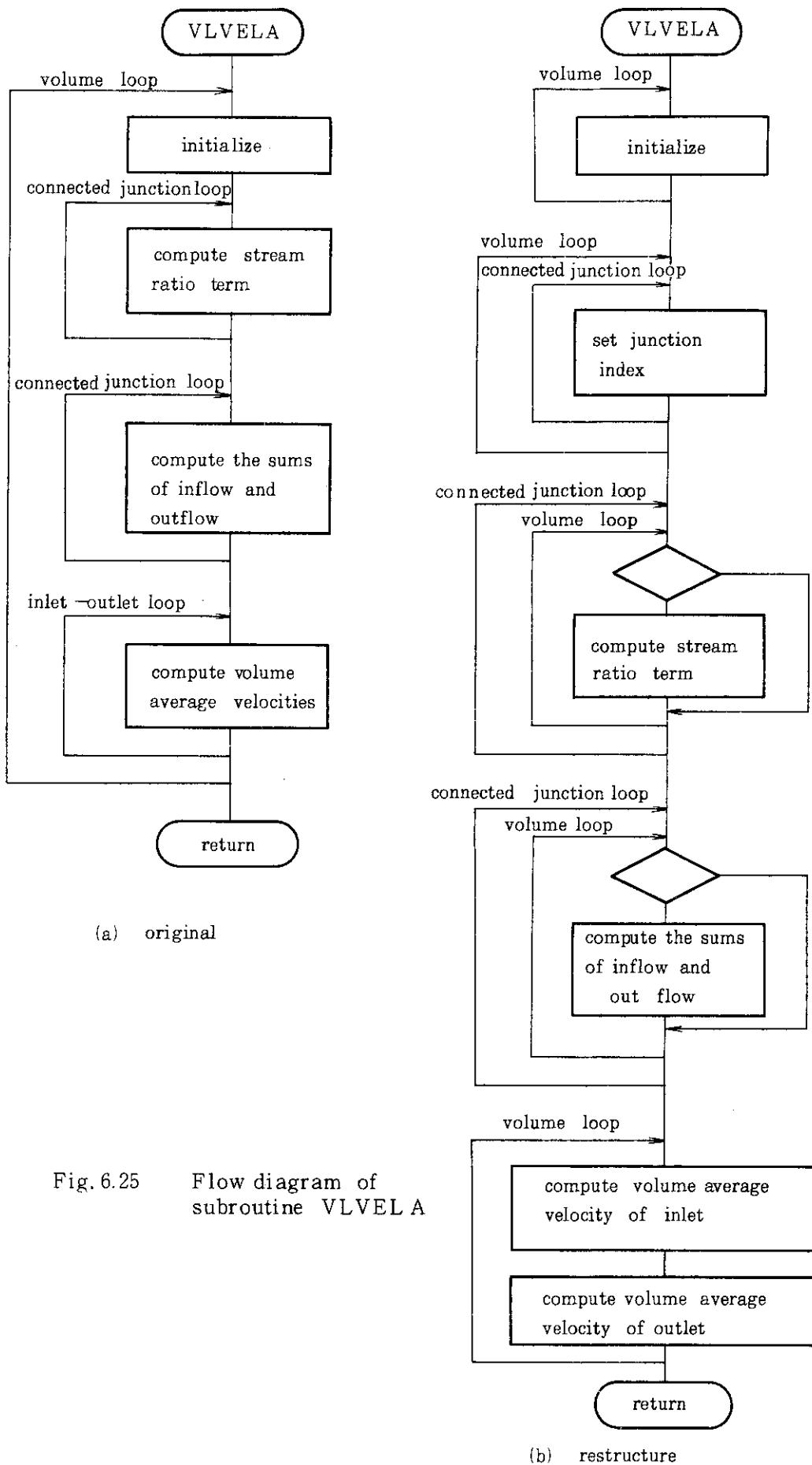


Fig. 6.25 Flow diagram of subroutine VLVELA

(b) 参照)。

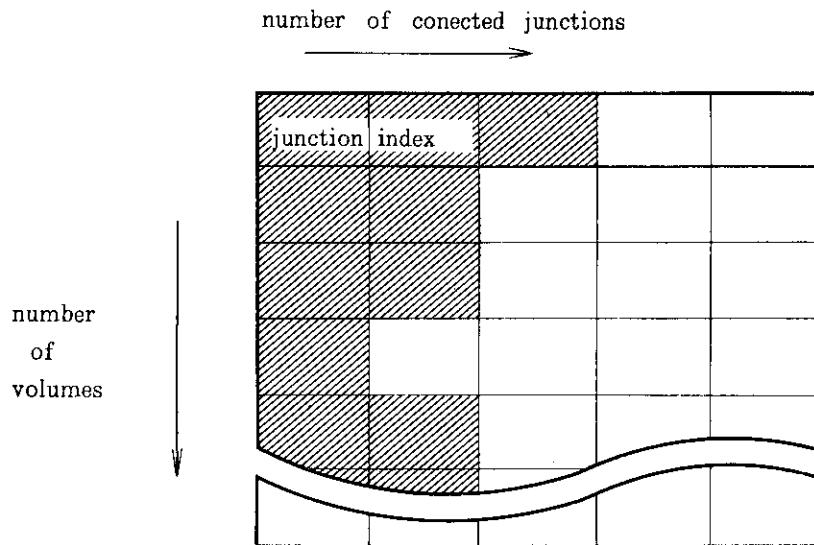


Fig. 6.26 Tabular form of connected junctions index

6.3.20 VOLVEL

VOLVELは、ジャンクションにおける流速の平均化により、ボリュームにおける流速を計算する。本サブルーチンは、外側にボリュームに関するDOループ、内側に接続ジャンクション・ループの二重ループ構造になっている。この二重DOループを一重化してベクトル化を行う。ただし、接続ジャンクションのインデックスが再帰的になる部分はスカラー処理とする。

Fig. 6.27(a)にオリジナルの、又Fig. 6.27(b)にベクトル化版の構造を示す。

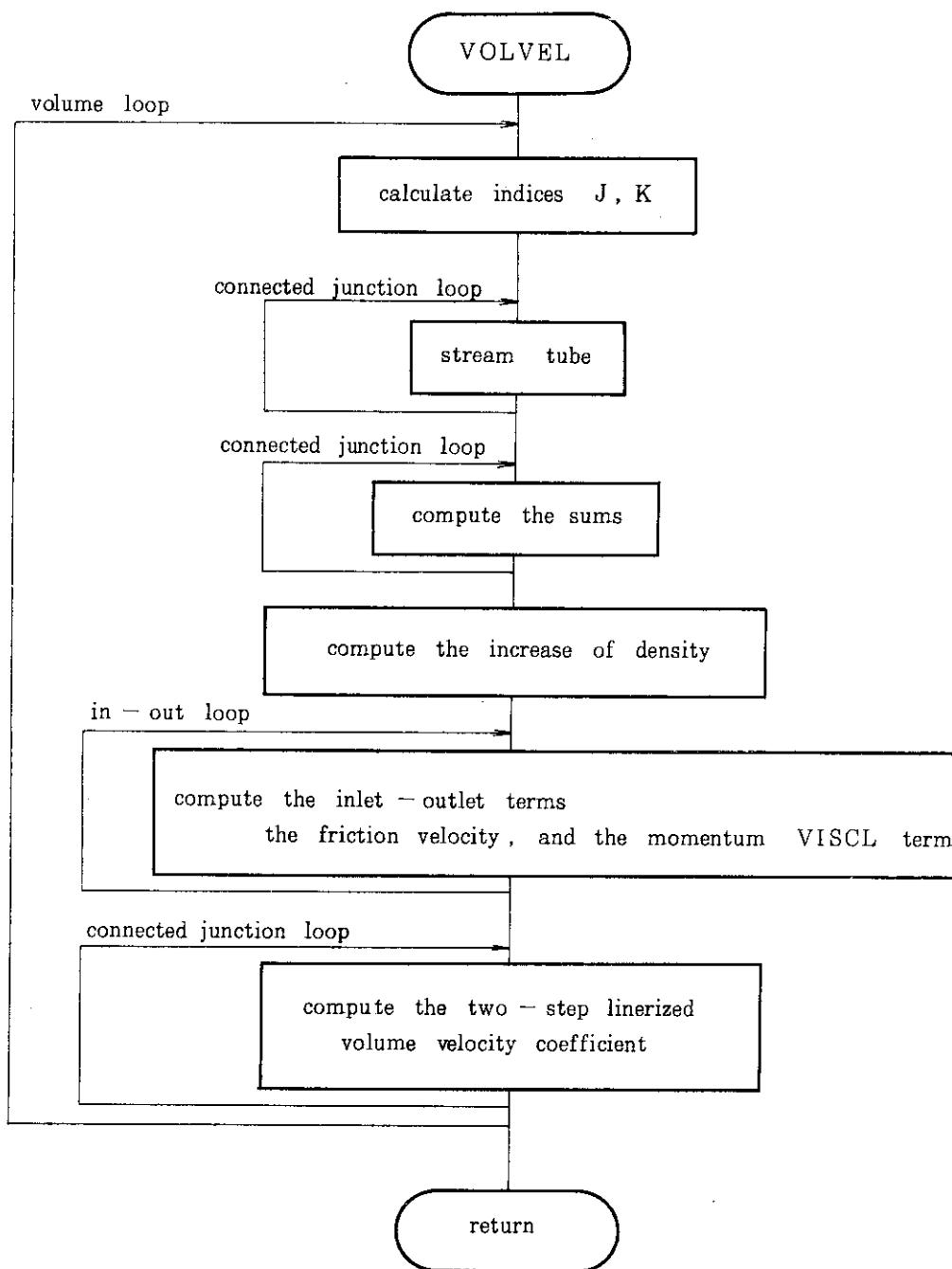


Fig. 6.27(a) Flow diagram of subroutine VOLVEL
in original version

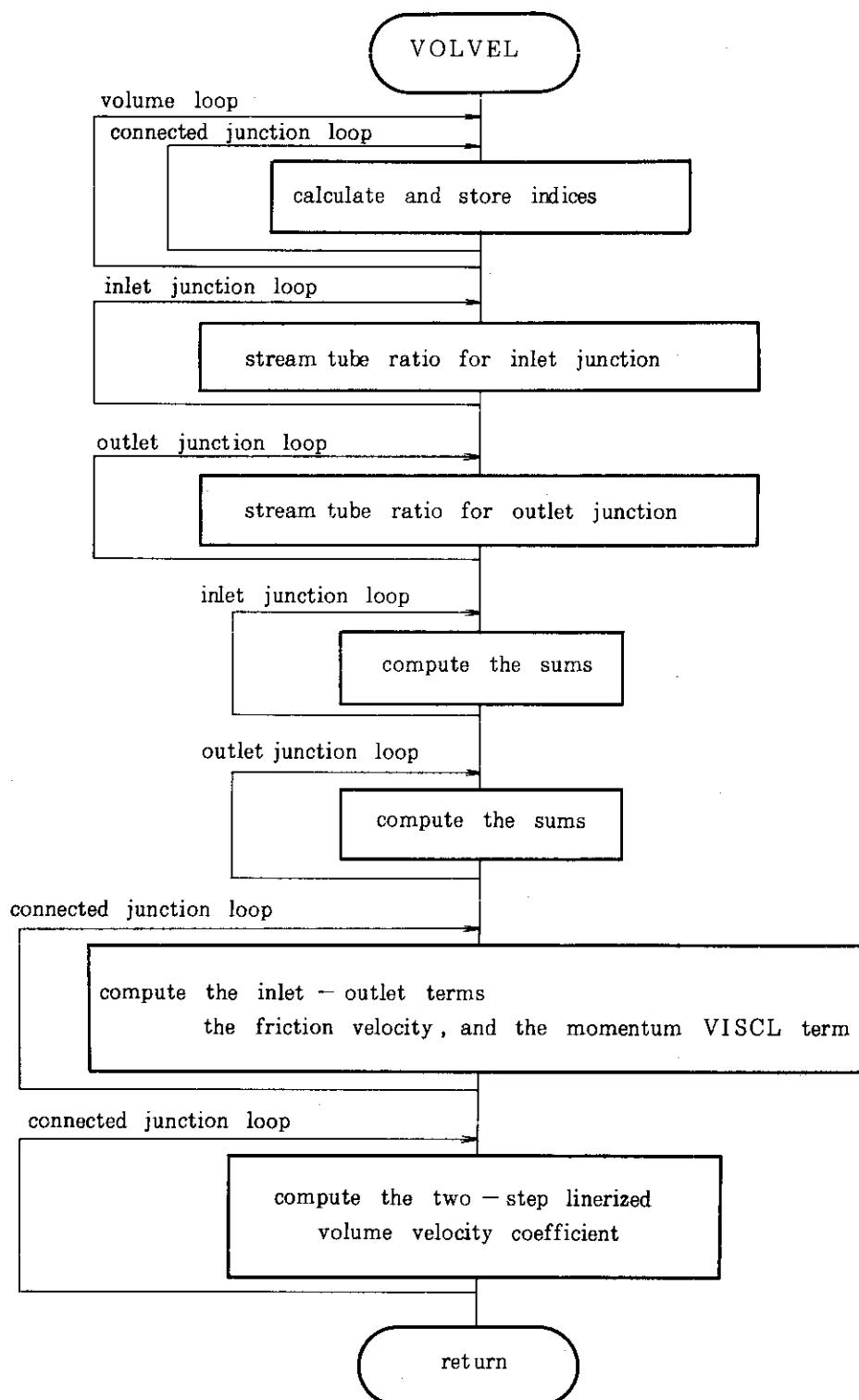


Fig. 6.27(b) Flow diagram of subroutine VOLVEL
in vector version

6.4 状態計算部分のベクトル化

Fig. 5.2 の STATE 以下に示す部分は、圧力 (P), エネルギー (U_f, U_g) を入力として密度など各状態値、導関数などを求めるサブルーチンである。密度導関数、温度導関数は、蒸気表から得られた κ, β, C_p などを用いて計算される¹⁾。全体のループ構造は、サブルーチン STATE に全コンポーネント、ボリュームに関する DO ループがあり、そこで状態によって各ボリュームによって引用するサブルーチンは制御される。下位ルーチン STATEP においても条件に応じて蒸気表検索ルーチン STH2X などを引用している。ベクトル化の方法は、ボリュームの状態、種類に従ってボリュームインデックスを集めたリストベクトルを作成し、下位ルーチンに送込みそのリストベクトルによるベクトル化を行う。Table 6.6 にベクトル化したルーチン名とその内容を示す。

Table 6.6 Names and functions of vectorized state relationship subroutines

オリジナル版でのルーチン名	ベクトル化版でのルーチン名	内容
STATE	STATE	各コンポーネントの状態式の評価を制御する。mass error の計算を行う。
STATEP	STATEV	time advanced volume についての各状態量、導関数を計算する。
STH2X0	STH2V0	温度を入力し飽和圧力を計算する。
STH2X1	STH2V1	温度とクオリティを入力し飽和状態における各特性値を計算する。
STH2XB	STH2VB	温度、圧力、クオリティを入力し飽和状態における各特性値を計算する。
STH2X2	STH2V2	圧力とクオリティを入力し飽和状態における各特性値を計算する。
STH2X6	STH2V6	圧力及び内部エネルギーを入力し、各特性値を計算する。
STH2XF	STH2VF	" "
VISCOG	VISCGV	水蒸気の粘度を計算する。
VISCOL	VISCLV	水の粘度を計算する。
THCON	THCONV	飽和状態の熱伝導率を計算する。
POLATL	POLALV	与えられたテーブルデータによる内挿計算を行う。

以上のサブルーチンはすべてボリュームについてベクトル化されている。

6.4.1 STATE

STATEは、ボリュームの種類、状態によってサブルーチンSTACCを引用するか、STATEPを引用するかを制御し、次に流体系ごとのmass error、全体のmass errorを計算する。下位ルーチンをベクトル化するために、ボリュームをその状態によって

- (1) アキュムレータボリューム
- (2) 中に空気を含むボリューム
- (3) 空気を含まないボリューム

の3種類に分け、それぞれボリュームインデックスによるリストベクトルを作成する。(1)はSTACCを引用するボリュームであるが、該当するボリューム数は非常に少ないのでベクトル化せず、このリストベクトルによるDOループを作成し中でSTACCを引用するようとする。(2)、(3)はSTATEPを引用するボリュームであり、STATEPはこのリストベクトルでベクトル化する。

mass error計算ループは、ビット操作ファンクションを取除き、接続ジャンクションのインデックスを作業用配列に貯え、それを使用するように変更することでベクトル化できる。

Fig. 6.28(a), (b)にその概略を示す。

6.4.2 STATEP

STATEPは、ほとんど独立した2つの部分

- (1) 空気を含んだボリュームについて計算する部分
- (2) 空気を含まないボリュームについて計算する部分

から成立っている。該当するボリュームのインデックスリストは、上位ルーチンSTATEで作られるため、ここでは基本的には2つの大きなDOループができることになる(Fig. 5.29参照)。(1)のDOループは、ボリューム数が少いためベクトル化せずそのまま残す。(2)のDOループについてベクトル化を行う。ベクトル化の方法は、下位ルーチンへのCALL文の前後でDOループを区切り、条件に従ってリストベクトルの再構成を行い新しいリストベクトルによるベクトル化を行う。DOループは、相別(気相、液相)、状態別(過熱、過冷却、など)に細かく分かれしていく。Fig. 6.30に概略を示す。

6.4.3 STH2X1 (STH2XB , STH2X2)

STH2X1は、温度、クオリティーなどを入力として、蒸気表データを内挿することにより、飽和状態における各特性値を求めるサブルーチンである。本サブルーチンは、STH2X6、STH2X0(後述)と共にNRTS Environmental Subroutinesの中の1本である。本サブルーチンは、MOD1のものとまったく同じであるので、MOD1をベクトル化した時のものをそのまま使用する。ただし、ベクトル化のために新設されている変数名、配列名は、MOD2の他のサブルーチンに合わせて変更する。ベクトル化の方法は、上位ルーチン(STATEP)から送られて来るリストベクトルにもとづく。

DOループは、蒸気表インデックスサーチ部分と、内挿計算部分の2つに分かれ、前者はスカラ-, 後者をベクトル計算とする。Fig. 6.31にその概略を示す。

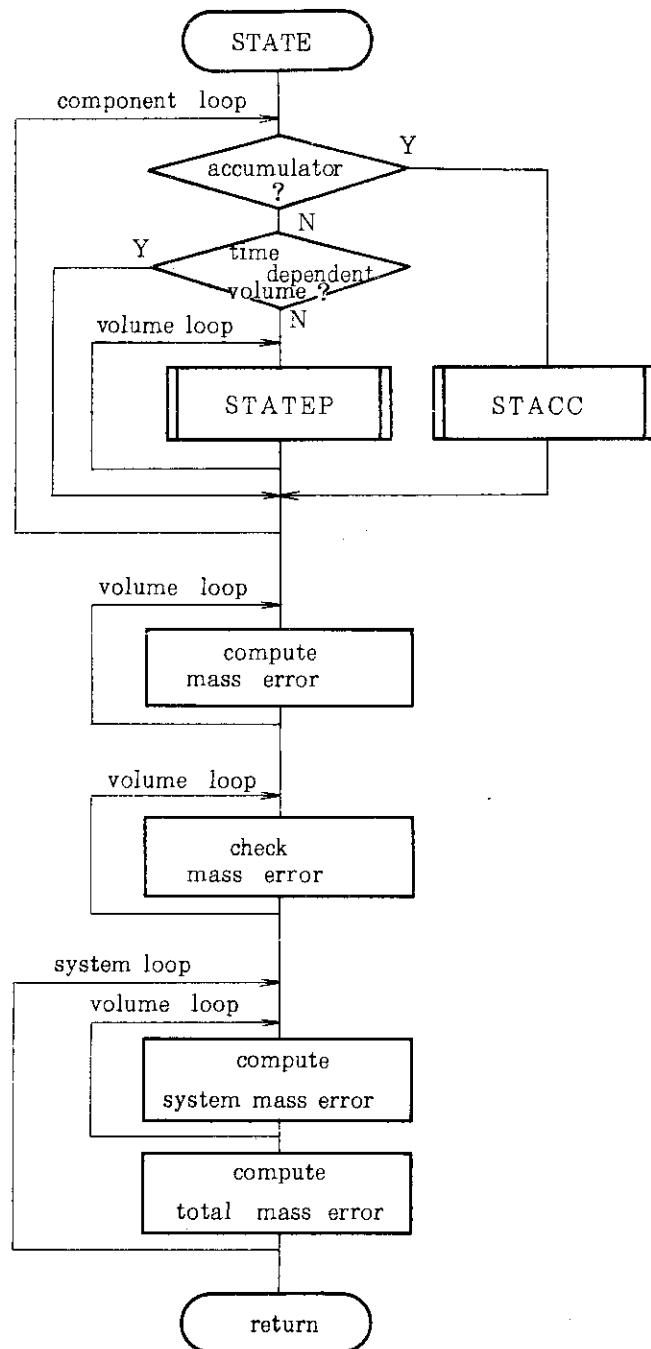


Fig. 6.28(a) Flow diagram of subroutine STATE
in original version

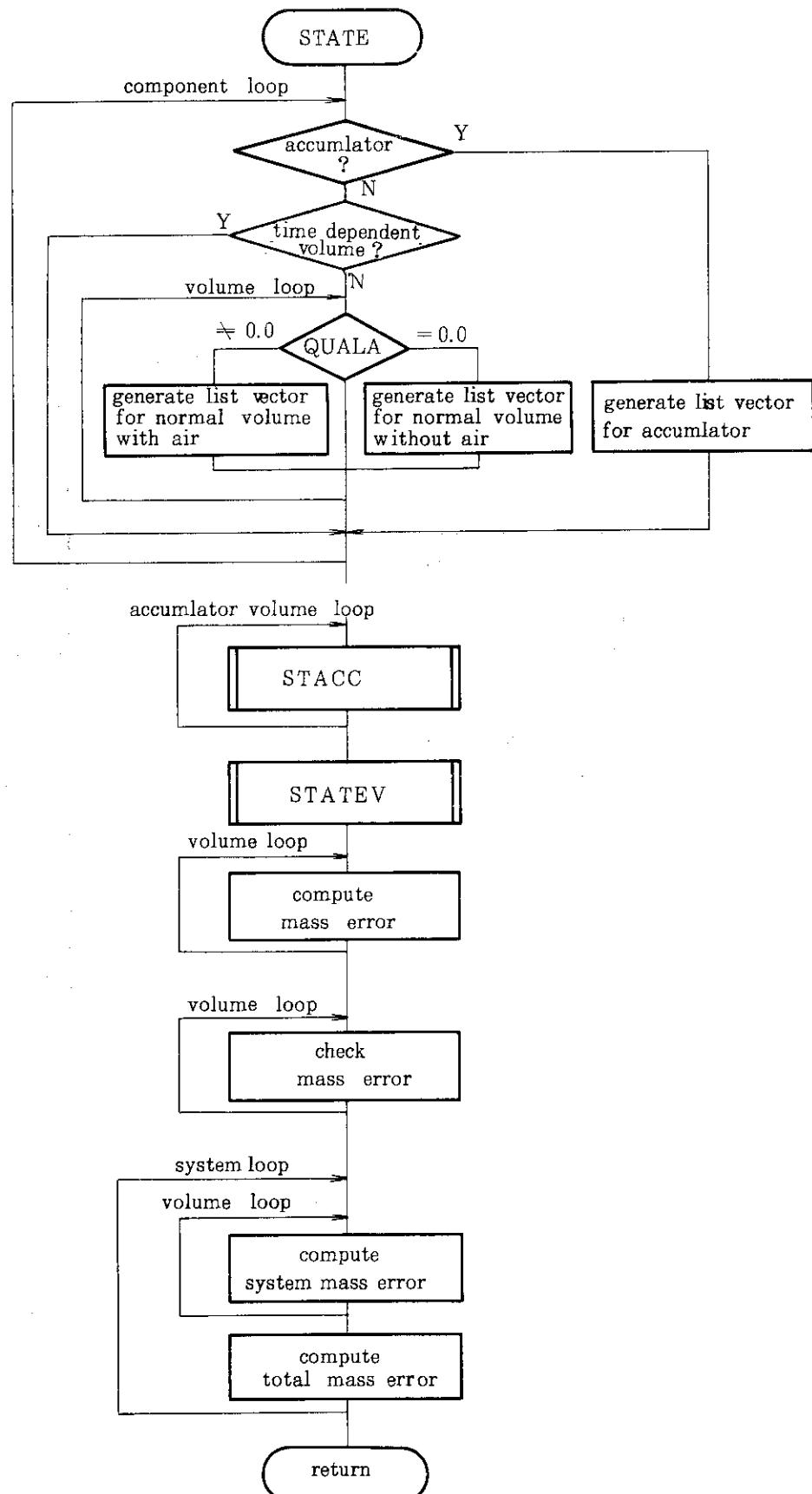


Fig. 6.28(b) Flow diagram of subroutine STATE
in vector version

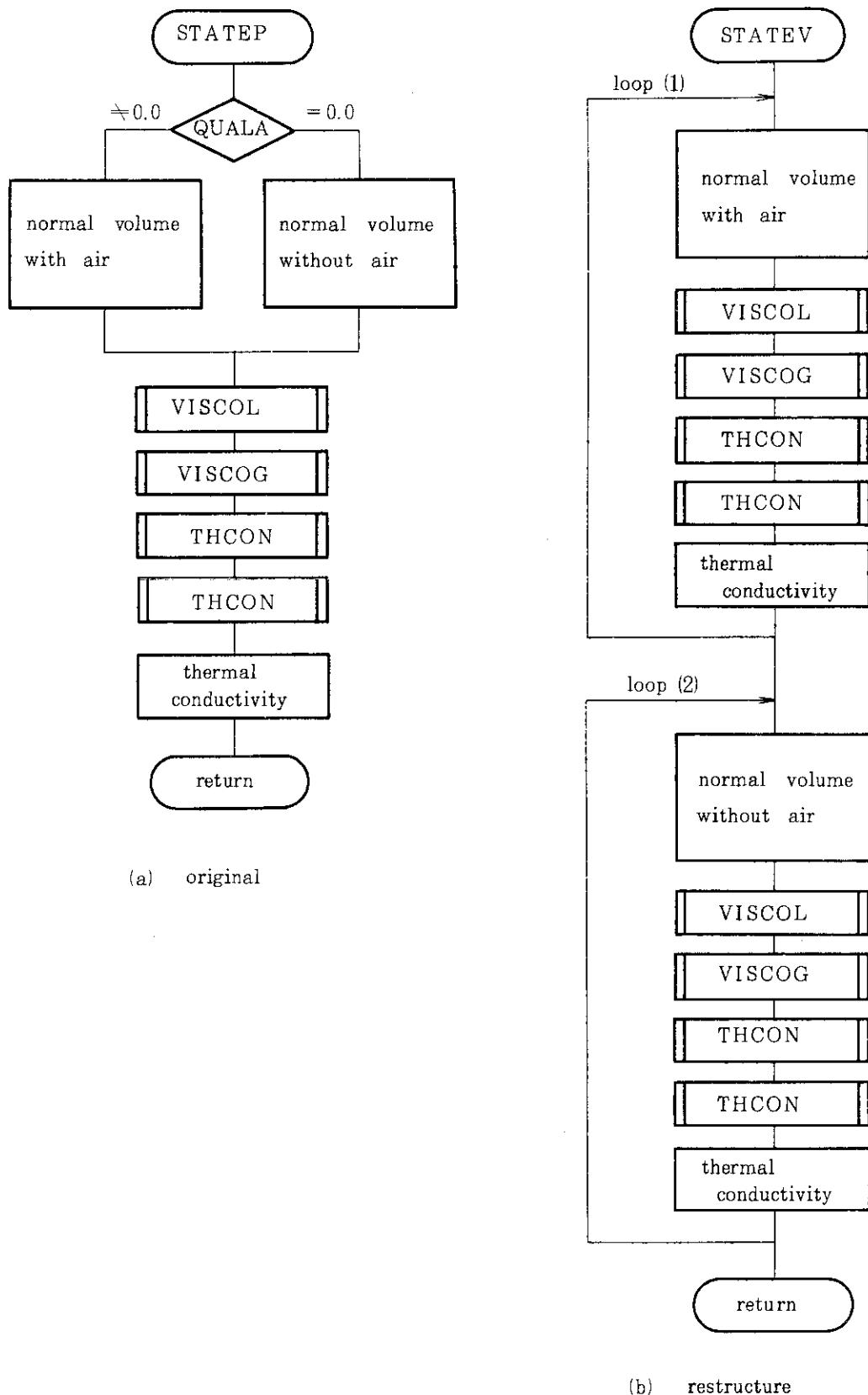


Fig. 6.29 Flow diagram of subroutine STATEP

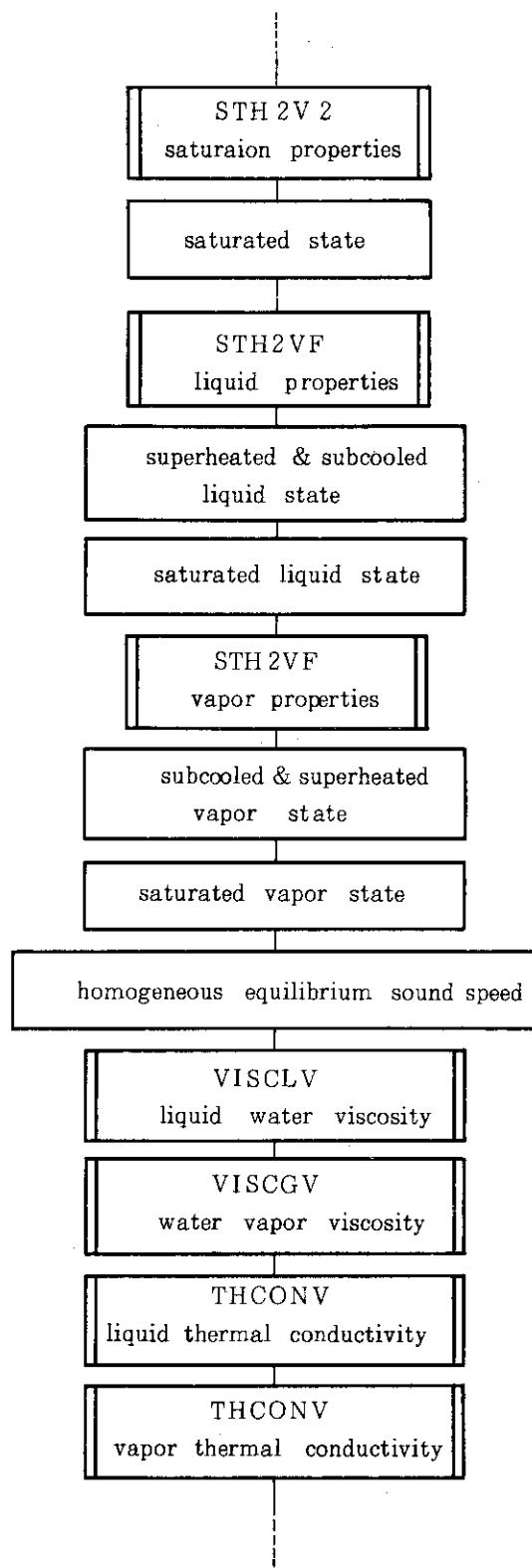


Fig. 6.30 Block diagram of vectorized loop

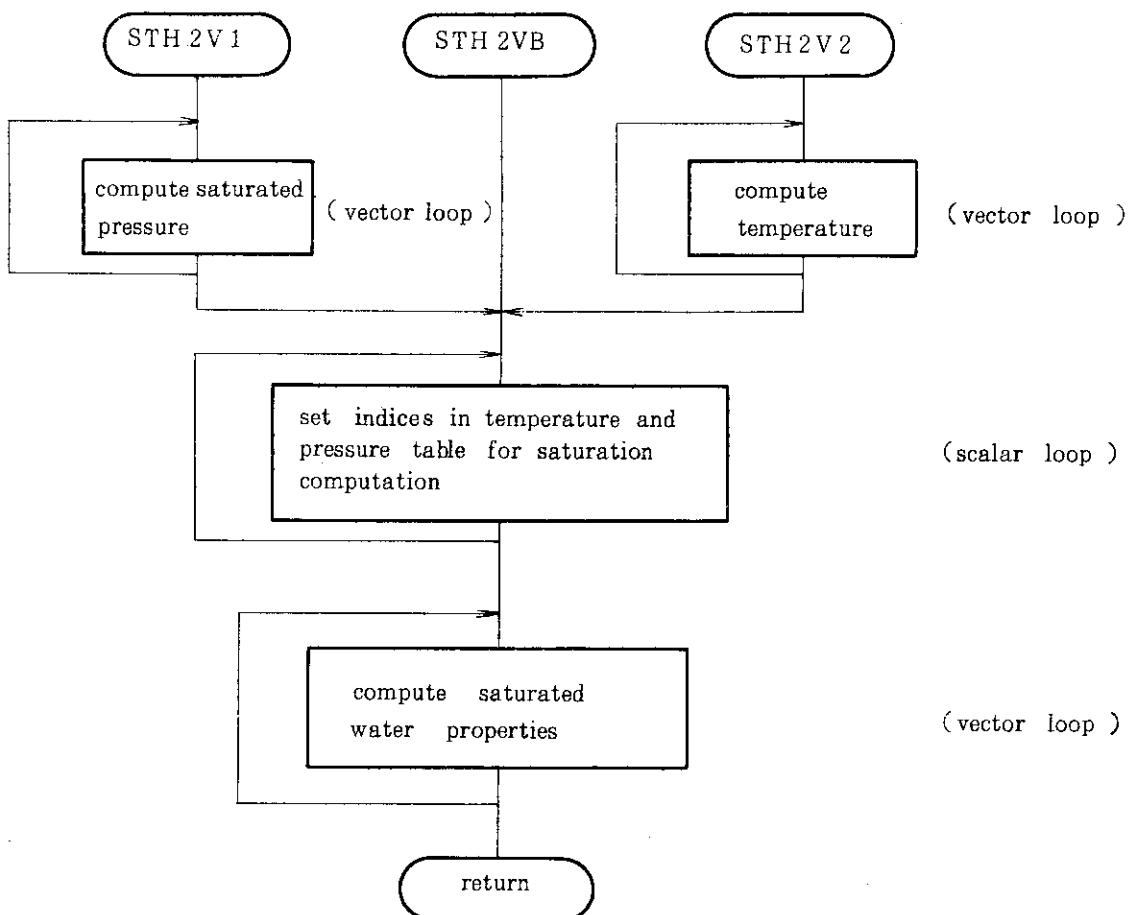


Fig. 6.31 Flow diagram of subroutine STH2X1

6.4.4 STH2X6 (STH2XF)

STH2X6もSTH2X1と同じく蒸気表検索ルーチンであり、MOD1のそれと同一であるためMOD1時のものを流用する。Fig. 6.32に概略フローを示す。このブロックごとにベクトル化されることになる。各ブロックのボリュームインデックスは、それ以前のブロック内で再構成する。

6.4.5 VISCOG, VISCOL, STH2X0

VISCOG, VISCOLは、本来気相、液相それぞれの粘度を計算するファンクションであるが、ベクトル化のためにサブルーチンに変更し、計算ボリュームのすべてを一度に計算するようとする。

- (1) VLSCOGは、通常の計算の他に臨界圧力を越えるようなボリュームについては、VISCOLを引用している。該当するボリュームのインデックスを集めリストベクトルを作直しそれでベクトル化する。Fig. 6.33に概略を示す。
- (2) VISCOLは、粘度の計算の前に温度を調べ臨界温度以下のボリュームについては、サブルーチン STH2X0を引用し飽和圧力を求めなおしている。変更方法は、STH2X0を引用

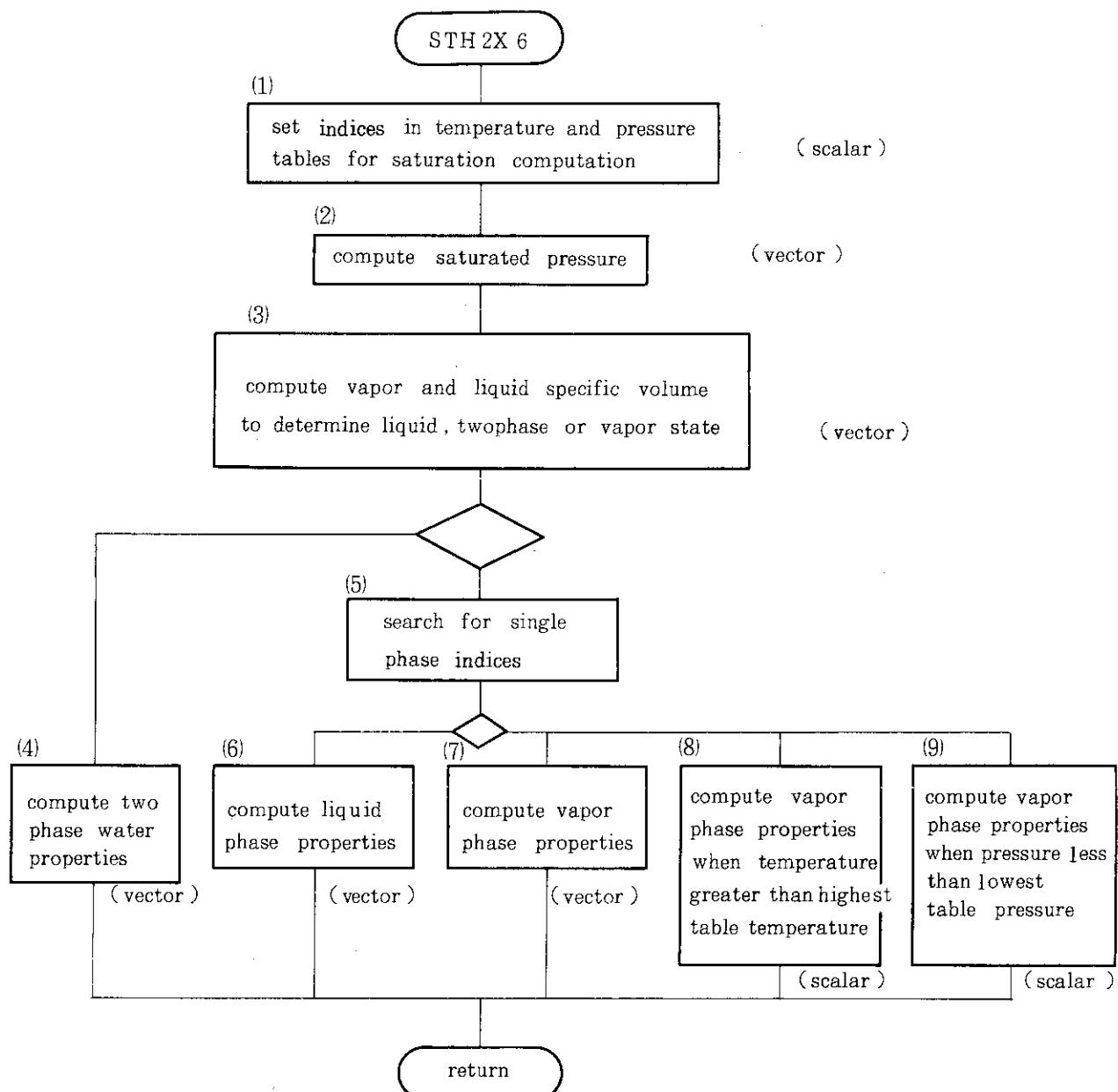


Fig. 6.32 Flow diagram of subroutine STH 2X 6

すべきボリュームのインデックスを集めリストベクトルを作直す。粘度の計算は、与えられたリストベクトルによって一度に計算する。Fig. 6.34に概略を示す。

(3) STH 2X 0 は、単純構造であるので全体を 1 つの DO ループとしてベクトル化する。

6.4.6 THCON, POLATL

熱伝導率を計算するサブルーチン THCON は、仮引数 N によって計算方法がまったく異なる。N = 1 の場合、テーブル内挿によって計算するが、実際の内挿計算は、サブルーチン POLATL が引用され、液相が飽和した状態の計算が行われる。N = 2 の場合、飽和又は過熱状態にある

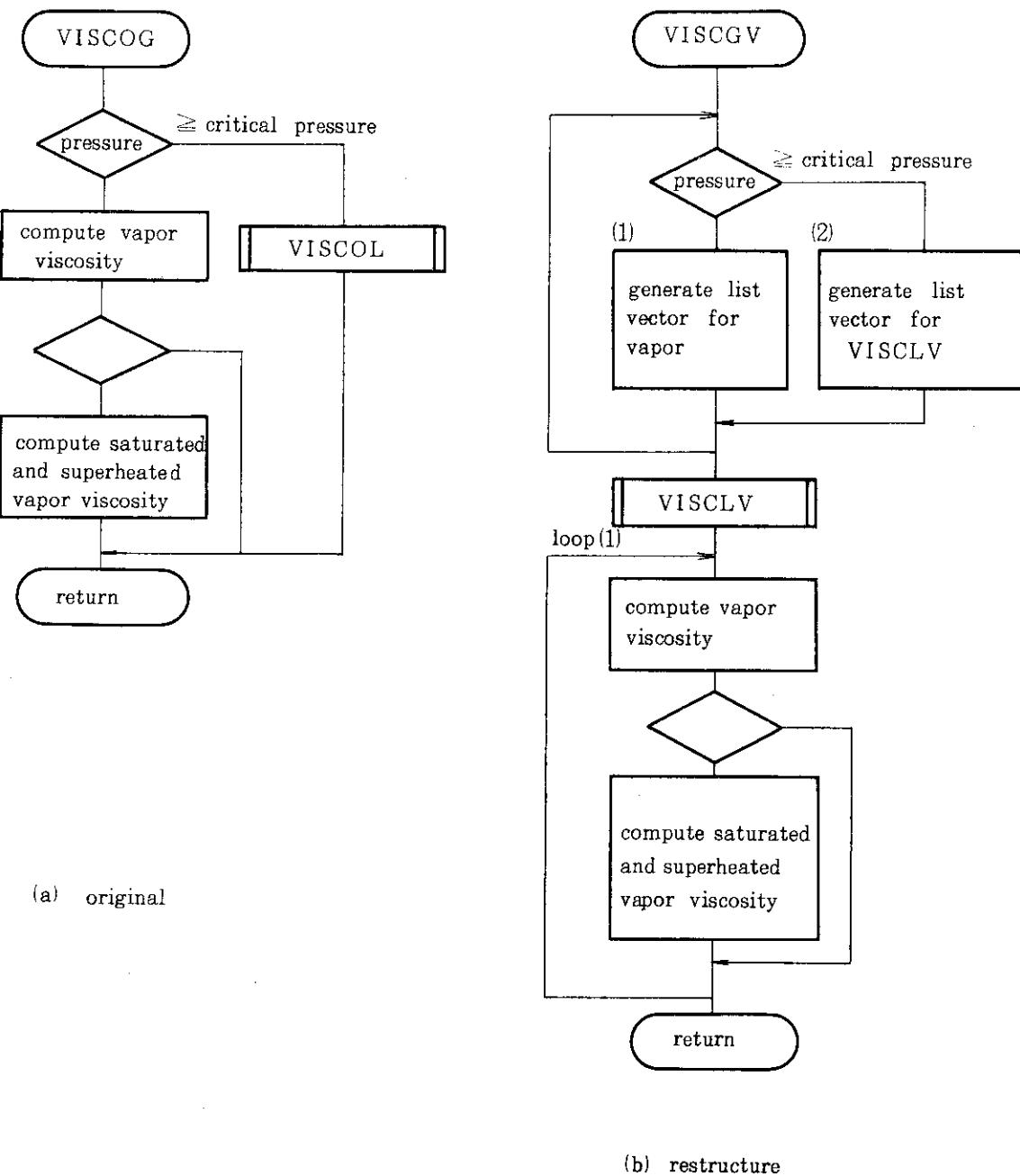


Fig. 6.33 Flow diagram of subprogram VISCOG

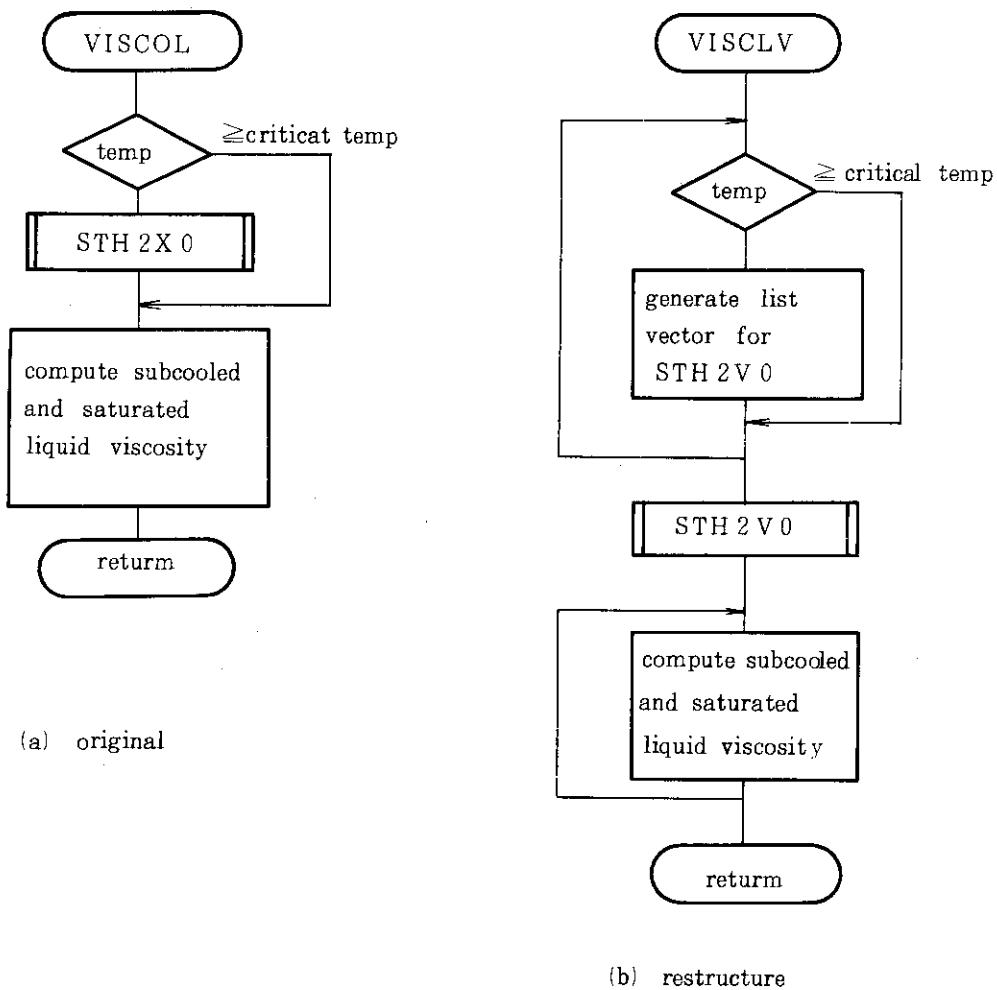


Fig. 6.34 Flow diagram of subprogram VISCOL

気相の場合の計算が行われ、計算ブロック全体をDO ループで囲む。

内挿ルーチンPOLATLは、テーブルインデックスサーチループと内挿計算ループの2つのDO ループに分け、前者をスカラー、後者をベクトル演算とする。Fig. 6.35, Fig. 6.36にその概略を示す。

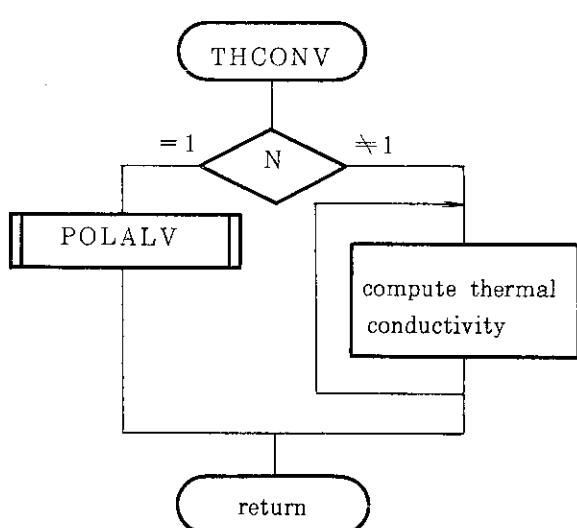


Fig. 6.35 Flow diagram of subprogram
THCONV

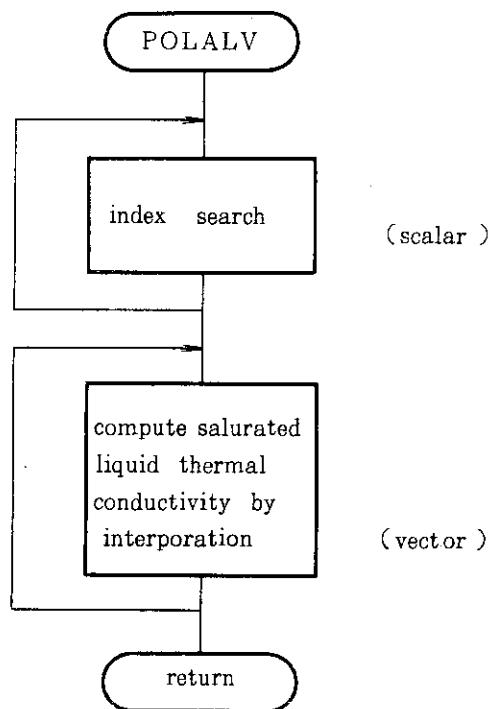


Fig. 6.36 Flow diagram of subroutine
POLATL

6.5 その他のサブルーチン

熱伝導計算、流体力学計算及び状態計算以外のサブルーチンで、ベクトル化又はスカラーレベルでの最適化を行ったものを Table 6.7 及び Table 6.8 に示す。

Table 6.7 Names and functions of vectorized other subroutines

オリジナル版でのルーチン名	ベクトル化版でのルーチン名	内容	主要ベクトル長
DTSTEP	DTSTEP	過渡計算時における出力の制御	ボリューム数 ジャンクション数
MOVER	MOVER	時間きざみ巾の制御	ボリューム数 ジャンクション数

Table 6.8 Names and functions of optimized other subroutines

ルーチン名	内容
CONVAR	コンポーネント制御変数の計算
RKIN	炉心の動特性の模擬
TRIP	トリップ状態の判定
TSTATE	時間依存ボリューム及びジャンクションの状態計算

6.5.1 DTSTEP, MOVER

DTSTEP は、過渡計算時における時間きざみ巾のコントロール、minor edit, major editなどの出力を制御するサブルーチンである。下位ルーチン MOVER は、計算値が妥当な値である場合、計算値の更新を行い次の時間ステップへ進む。また計算値が妥当でない場合、時間きざみを半分にし、計算値を前回の時間ステップの値に戻し、再度状態計算をやりなおす。この 2 つのサブルーチンは、ビット操作ファンクションを組込み関数等に変更するだけでベクトル化は完了する。再計算時に引用する STATE, JPROP, VLVELA は、ベクトル化されたものを使用する。

6.5.2 CONVAR

CONVAR は、コンポーネントの制御変数の計算を行う。このサブルーチンは、外側にコントロール・コンポーネントに関する DO ループ（平均ベクトル長 22）、内側にコントロール変数に関する DO ループ（平均ベクトル長 6）の二重 DO ループ構造となっている。

しかし、最内ループの演算密度が非常に低く、さらに平均ベクトル長も小さいため、ベクトル化による性能向上があまり期待できない。従って、本サブルーチンではベクトル化は行わず、ビット操作ファンクションを組込み関数に変更するのみとする。

6.5.3 RKIN

RKIN は、炉心の動特性を模擬するサブルーチンであるが、ベクトル長が小さいため、スカラ-最適化のみを行う。

6.5.4 TRIP

TRIP は、トリップの判定を行い、トリップ状態である場合、その時間をセットする。本サブルーチンにはトリップ数の DO ループが存在するが、平均ベクトル長が 12 ないし 8 と小さく、さらに DO ループ内の演算密度も低い。従って、ビット操作ファンクションの組込み関数への変更のみを行い、ベクトル化は行わない。

6.5.5 TSTATE

TSTATE は、時間依存ボリューム及びジャンクションの状態を求めるサブルーチンである。本サブルーチンには、時間依存ボリュームに関する DO ループが存在するが、ベクトル長が平均 2 と小さく演算密度も低い。従って、ベクトル化は行わずビット操作ファンクションの組込み関数への変更のみを行う。

7. ベクトル化による性能向上

前章の書替えを行った後、スカラー計算及びベクトル計算を行い、そのCPU時間を、
TYPPWR2, LSTF 5%破断についてそれぞれTable 7.1, Table 7.2に示す。ここで、
PROB 2は問題が小さく実際の運用とは離れているため、処理性能の比較からは除外した。この
結果、ベクトル化版のベクトル計算は、オリジナル版のスカラー計算と比較して3.0～3.5
倍の性能向上、ベクトル化版のスカラー計算と比較して2.0～2.2倍の性能向上となっている。
さらに最適化前の版のスカラー計算と比較すれば、4.3倍の性能向上である。MOD1のベクト
ル化ではベクトル化による性能向上は、2.8倍程度であったのに比べると、かなり性能が向上
している。この主な理由は、熱伝導計算部分の境界条件計算サブルーチン群が、前回はまったく
ベクトル化されていなかったのに対し、今回はそれらがすべてベクトル化されたためである。
特に、LSTF 5%破断データでは熱計算部分のCPU時間の割合が高くなっている。ベクトル長
となる熱構造体の数が166と多いのと相まって、高い性能向上比となっている。

Table 7.1 Vector length, vectorizable ratio, and cpu time of three versions for TYPPWR 2 data

subroutine name	vector length	vector ratio	ORIGINAL version scalar mode		VECTOR version scalar mode		VECTOR version vector mode		
			entries	cpu	cpu	ratio	entries	cpu	ratio
D T S T E P	139	0.70	2671	23.838	6.477	3.680	2687	3.686	6.467
S T A T E (R)	40-139	0.91	313	8.546	7.572	1.129	321	3.295	2.594
T R I P		0.0	2357	1.013	0.229	4.424	2365	0.224	4.522
H T A D V	139,166	0.64	2670	45.115	28.173	1.601	2686	15.095	2.989
H T C O N D	39	0.71	221610	43.024	26.744	1.609	2686	13.931	3.088
S T A T E (T)		0.0	2670	1.670	0.910	1.835	2686	0.934	1.788
J P R O P (0)	142	0.99	2983	46.996	17.178	2.736	3007	7.529	6.242
V O L V E L	139,144	0.65	2670	37.941	14.935	2.540	2686	7.372	5.147
P H A I N T	139,142	0.98	2670	87.394	66.275	1.319	2686	27.934	3.129
H Z F L O W		0.0	0	0.0	0.0	0.0	0	0.0	0.0
F W D R A G	139	0.99	2670	21.401	18.203	1.176	2686	3.846	5.564
H L O S S	28	0.06	2670	6.900	4.530	1.523	2686	4.861	1.419
V E X P L T	127,139	0.90	2670	50.681	36.286	1.397	2686	16.582	3.056
J P R O P (1)	142	0.99	2670	8.451	1.952	4.329	2686	2.150	3.931
P R E S E Q	139,194	0.92	3284	36.100	22.851	1.580	3295	5.989	6.028
S Y S S O L		0.21	3284	29.652	23.249	1.275	3295	21.653	1.369
V F I N L	142	0.84	2670	31.854	11.605	2.745	2686	8.191	3.889
E Q F I N L	139,142	0.95	2670	29.682	20.956	1.416	2686	6.695	4.433
S T A T E (A)	40-139	0.91	2670	76.335	67.240	1.135	2686	28.823	2.648
R K I N	30	0.85	2357	2.146	1.930	1.112	2365	1.656	1.296
C O N V A R		0.0	2357	2.987	0.847	3.527	2365	0.820	3.264
T O T A L		0.82		591.726	378.142	1.565		181.266	2.984
J O B C P U				10' 17.43	6' 43.54	1.530		3' 26.91	2.984

Table 7.2 Vector length, vectorizable ratio, and cpu time of three versions for LSTF 5% break analysis data

subroutine name	vector length	vector ratio	ORIGINAL version scalar mode		VECTOR version scalar mode		VECTOR version vector mode		
			entries	cpu	cpu	ratio	entries	cpu	ratio
D T S T E P	168	0.67	15070	156.797	38.577	4.065	14483	18.266	8.584
S T A T E (R)	50-168	0.90	1772	58.760	52.473	1.120	1546	19.187	3.062
T R I P		0.0	13297	15.169	2.872	5.282	12936	2.813	5.392
H T A D V	168,332	0.72	15069	503.784	312.432	1.612	14482	122.541	4.111
H T C O N D	102	0.69	2501454	563.846	386.026	1.461	14482	173.290	3.254
S T A T E (T)		0.0	15069	9.789	5.447	1.797	14482	5.329	1.837
J P R O P (0)	175	0.99	16841	342.537	127.937	2.677	16028	54.080	6.334
V O L V E L	167,179	0.65	15069	269.438	108.799	2.476	14482	52.959	5.088
P H A I N T	168,175	0.97	15069	608.184	479.456	1.268	14482	181.179	3.357
H Z F L O W		0.0	0	0.0	0.0	0.0	0	0.0	0.0
F W D R A G	168	0.99	15069	151.994	132.111	1.151	14482	22.195	6.848
H L O S S	175	0.07	15069	45.934	29.689	1.547	14482	29.927	1.535
V E X P L T	149,168	0.89	15069	315.557	224.293	1.407	14482	86.826	3.634
J P R O P (1)	175	0.99	15069	68.627	16.064	4.272	14482	14.363	4.778
P R E S E Q	168,175	0.92	24406	327.645	205.997	1.591	23773	50.964	6.429
S Y S S O L		0.23	24406	270.934	213.985	1.266	23773	193.681	1.399
V F I N L	175	0.85	15069	252.609	90.496	2.791	14482	56.166	4.498
E Q F I N L	168,175	0.94	15069	208.418	148.605	1.402	14482	41.731	4.994
S T A T E (A)	50-168	0.90	15069	518.306	457.773	1.132	14482	185.190	2.799
R K I N		0.0	0	0.0	0.0	0.0	0	0.0	0.0
C O N V A R		0.0	13297	27.855	7.750	3.594	12936	7.268	3.833
T O T A L		0.80		4716.183	3040.782	1.551		1317.955	3.578
J O B C P U				79' 43.27	51' 47.23	1.539		23' 05.14	3.453

8. ベクトル計算機による結果の妥当性

ベクトル化版コードを、スカラー・モード及びベクトル・モードで実行したときの結果を Fig. 8.1, Fig. 8.2 に示す。これらによると、スカラー・モードで実行した場合はオリジナル版と完全に一致するが、ベクトル・モードで実行した場合は、わずかな差異が見られる。また、LSTF 5% 破断データにおいては、アキュムレータの動作開始後に挙動が激しくなり、それに伴ってスカラー・モードとベクトル・モードによる結果の差も大きくなっている。この差の原因として、ベクトル化のミス及び、計算機の差が考えられる。計算機の差については MOD 1 の時に調査がなされており、以下のものがわかっている。

- (1) ベクトルコンパイラがパイプラインの有効利用のために、文、式の評価順序を変更する。
- (2) 総和計算において、ベクトル化のために足込むデータの順序を入れ替える。
- (3) SQRT, SINなど組込み関数の値が、スカラーとベクトルで若干異なる。

これらによる値の差は、8 バイト演算で 10 進 18 衔中下 1 衔以下というわずかなものであるが、RELAP 5 自身が持っている数値に対する敏感性により誤差が累積し、スクラムなどの発生タイミングにずれが生じる。以後、これらが累積され大きな差となって現われてくる。

従って、このスカラー・モードとベクトル・モードでの計算結果の差がベクトル化のミスによるものか、計算機の差によるものかを区別する必要がある。(1), (2)については、演算順序をスカラーと同じにするコンパイルオプションである NOADVANCED (NOEVL) を指定し、また(3)についても、スカラー組込み関数を用いて計算したところ、性能はかなり低下したがオリジナル版をスカラー計算した結果と全く同じものを得た。以上のチェックにより、ベクトル化版のベクトル化によるミスではなく、スカラーモードとベクトルモードによる計算順序の違いが計算結果の差として表われていることが判る。

9. ベクトル化によるメモリーの増加

ベクトル化作業は、当初三人でサブルーチンごとに別々に行ったため、追加した作業用配列はサブルーチン別に独立していた。その結果、実行時に必要なリージョンサイズは、オリジナル版では約 2.8 MByte であったが、ベクトル化版では約 5 MByte となった。これを実際の運用に先立って、領域の縮小を計った。

各配列を保存配列と、真の作業配列に分け、この作業配列を Table 9.1 に示すように同一のコモン領域に設定した。この結果リージョンサイズは 4.4 MByte にまで縮小された。原研の運用上、この程度の縮小では実質的な有効性はあまりないが、他のコードではかなり有効であると思われる。

Table 9.1 Names and sizes of named common block
added by vectorization

(単位: Byte)

インクルード メンバー名	使 用 サブルーチン名	保 存 領 域		作 業 領 域 サ イ ズ		
		コ モ ン ブ ロ ッ ク 名	サ イ ズ	コ モ ン ブ ロ ッ ク 名 ¥CMWK1	コ モ ン ブ ロ ッ ク 名 ¥CMWK2	コ モ ン ブ ロ ッ ク 名 ¥CMWK3
¥EQFINL	EQFINL			9,604		
¥GAPVAV	HT1TDV, MADATV			25,200		3,200
¥HLOSS	HLOSS			160,000	118,400	
¥HTRCOV	熱伝導計算部分					
¥HTSCRV	HT1TDV, MADATV					
¥HTSRCV	熱伝導計算部分	¥CMHTS	393,680			
¥HTVP 1	"	¥HTVP 1	116			
¥JPROP	JPROP	¥CMJPR	3,600			
¥MOVER	MOVER	¥CMMOV	1,200			
¥MTBLSV	HTADV, MADATV	¥CMMTB	224,000			
¥PHAIN1	PHAIN1	¥CMPHA	15,600	168,000		
¥PHAIN2	PHAIN1, HIFBUB				12,000	
¥PMINV1	PMINVD, PMINV1			96,000		
¥POLALV	POLALV	¥CMPLL	1,200			
¥POLASV	POLASV	¥CMPLS	1,200			
¥PRESEQ	PRESEQ	¥CMPRE	3,600	21,600		
¥STH2V1	STH2V1				3,600	
¥STH2V6	STH2V6				73,200	
¥THCONV	THCONV	¥CMTHC	1,200		1,200	
¥VEXPLT	VEXPLT			123,600		
¥VFINL	VFINL	¥CMVFI	2,400			
¥VISCGV	VISCGV				2,400	
¥VISCLV	VISCLV					7,200
¥VLVELA	VLVELA			62,400		
¥VMINV3	VMINV3				48,000	
¥VOLVEL	VOLVEL	¥CMVOL	14,400	4,800		
¥WKSTAT	STATE, STATEV	¥CMSTA	21,708	170,400		
保存・作業領域別合計			683,904	170,400	118,400	7,200
総合計 979,904 Byte						

10. おわりに

軽水炉安全性解析コードRELAP5／MOD2／CYCLE36のベクトル化が完了し、その結果3.0～3.5倍の性能向上を得た。原子炉安全性計算の分野で、ある程度の成果が得られた数少ない例のうちの一つである。本コードは、今後原研において相当量の使用が見込まれ、ベクトル化版は計算機の負担の軽減、処理効率の向上、研究の促進に役立つものと思われる。

RELAP5は、現在もエラー修正が続けられており、原研でも修正作業が行われる予定である。同時に、このベクトル化版も修正する必要があり、その際に本報告が参考になるであろう。

ベクトル化作業は、MOD1の時の20人・月に比べてMOD2では10人・月とかなり早く終了している。これは、前回のベクトル化の方針を踏襲したことと、ベクトル化作業の経験がかなり役立った為である。ベクトル化作業は、ノウハウの蓄積が非常に重要であり、特にRELAP5のようなコードは、プログラム構造、データ構造に特徴があり、それを理解したうえで作業を行う必要がある。今後、同様な作業ができたとき、文献[3]と共に利用されれば幸いである。

近々、VPコンパイラもレベルアップされる予定であり、最適なベクトル化技法も変わる可能性がある。その場合は、他のベクトル化の方法もためしてみる必要があろう。性能向上比が、現在より高いものが得られる可能性は充分ある。

謝 辞

計算センター室長平川隆氏、室長代理浅井清氏、および安全工学第1研究室長田坂完二氏には、本報告を書く機会を与えて頂くと共に、報告書全般に関する御指導を受けました。心より感謝します。

安全工学第1研究室熊丸博滋氏、小泉安郎氏、株NDCの大槻真人氏およびフランスエネルギー庁のC. Chauliac氏には、入力データを提供して頂くと共に計算結果に対して貴重な助言を頂きました。感謝します。

助原子力データセンターの須藤高史氏には本報告作成にあたり、数々の御教示を頂きました。心より感謝します。

計算センター藤井実氏および富士通株の牧野光弘氏には、本リポートの投稿にあたり記述方法等の御指導をいただきました。感謝します。

参考文献

- (1) V. H. Ransom, R. J. Wagner : RELAP 5/MOD 2 Code Manual Volume 1, EGG-SAAM - 6377, INEL (April 1984).
- (2) M. Ishiguro, et al. : VECTORIZATION OF TRANSIENT ANALYSIS CODE RELAP 5/MOD 1 AND ITS EFFECT, JAERI-M 85-040, (March 1985).
- (3) 奈良岡賢逸, 他: 軽水炉安全解析コード RELAP 5/MOD 2/CY 21 の FACOM M-380 システムへの変換, 私信
- (4) 徳永康男, 原田裕夫, 石黒美佐子: 原子力コードにおける数値解法とそのベクトル化, JAERI-M 85-143 第4章 (September 1985).