

JAERI-M
92-142

原子力コードのベクトル化と改良

[MEUDAS 4, FORCE, STREAM V2.6,
HEATING 7-VP SCDAP/RELAP5
／MOD2.5, NBI 3 DGFN]

1992年9月

根本 俊行* 鈴木孝一郎* 磯辺 信雄* 町田 昌彦*
長内 誠志* 横川三津夫

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, 1992

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

原子力コードのベクトル化と改良

[MEUDAS4, FORCE, STREAM V2.6, HEATING7 - VP]
[SCDAP/RELAP5/MOD2.5, NBI3DGFN]

日本原子力研究所東海研究所情報システムセンター

根本 俊行*・鈴木孝一郎*・磯辺 信雄*

町田 昌彦*・長内 誠志*・横川三津夫

(1992年8月25日受理)

本報告はトカマク平衡及び局所MHD安定性解析コードMEUDAS4(CR版、及びFFT版)、磁界解析コードFORCE、3次元熱流体解析コードSTREAM V2.6、自由電子レーザーコードのベクトル化、及び3次元熱解析コードHEATING7-VP、炉心燃料損傷コードSCDAP/RELAP5/MOD2.5、イオンビーム軌道計算コードNBI3DGFNの改良について述べる。

オリジナル版のスカラモードに対するベクトル化版の速度向上は、MEUDAS4で2.3~4.9倍STREAM V2.6で1.9~5.4倍、FORCEで2.6~6.2倍、自由電子レーザーコードで1.9倍が得られた。また、コードの改良では、HEATING7-VPに対して解析領域設定の機能強化、SCDAP/RELAP5/MOD2.5に対してAE化を行った。

本報告書では、コードの概要、ベクトル化、及び整備の方法、計算結果の評価、及び機能強化のための改良内容について述べる。

Vectorization and Improvement of Nuclear Codes

[MEUDAS4, FORCE, STREAM V2.6, HEATING7-VP]
SCDAP/RELAP5/MOD2.5, NBI3DGFN]

Toshiyuki NEMOTO*, Koichiro SUZUKI*, Nobuo ISOBE*
Masahiko MACHIDA*, Seiji OSANAI* and Mitsuo YOKOKAWA

Computing and Information Systems Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received August 25, 1992)

Eight nuclear codes have been vectorized and modified to improve their performance. These codes are magnetic fluid equilibrium code MEUDAS4 (CR and FFT versions), the magnetic field analysis code FORCE, the three-dimensional heat fluid analysis code STREAM V2.6, the three-dimensional heat analysis code HEATING 7-VP, the severe accident transient analysis code SCDAP/RELAP 5/MOD 2.5 for light water reactors, the ion beam orbital analysis code NBI3DGFN, and a free electron laser analysis code.

The speedup ratios of the vectorized versions to the original ones in scalar mode are 2.3-4.9, 1.9-5.4, 2.6-6.2, and 1.9 for the MEUDAS4, STREAM, FORCE, and free electron laser analysis code, respectively. The definition method of the computational regions in the HEATING7-VP is improved. The SCDAP/RELAP5/MOD2.5 is modified to use extended memory regions of the computer.

In this report, outlines of the codes, techniques used in the vectorization and reorganization of the codes, verification of computed results, and improvement on the performance are presented.

* On leave from FUJITSU, Ltd

Keywords : Nuclear Codes, Vectorization, FORCE, MEUDAS4, STREAM
HEATING7-VP, SCDAP/RELAP5/MOD2.5, NBI3DGFN,
Free Electron Laser Code

目 次

1. はじめに	1
2. MEUDAS4のベクトル化	3
2.1 概 要	3
2.2 プログラム解析	3
2.3 ベクトル化の方針	3
2.4 ベクトル化効果	4
3. MEUDAS4 FFT 版のベクトル化	7
3.1 概 要	7
3.2 プログラム解析	7
3.3 ベクトル化の方針	7
3.4 ベクトル化効果	8
4. FORCE コードのベクトル化	13
4.1 概 要	13
4.2 基本方程式	13
4.3 動的挙動分析	14
4.4 ベクトル化手順	14
4.5 ベクトル化の結果	16
5. STREAM V2.6 のベクトル化	30
5.1 概 要	30
5.2 基礎方程式	30
5.3 境界条件	31
5.4 ベクトル化	32
5.5 I/O回数の削減	44
5.6 実 行	45
6. 自由電子レーザーコードのベクトル化と整備	48
6.1 コードと作業の概要	48
6.2 ベクトル化	48
6.3 リスタート機能の整備	52
6.4 ま と め	53
7. HEATING7 - VP の改良	54
7.1 概 要	54
7.2 物性領域定義の改良について	54
7.3 テスト計算	67
8. SCDAP/RELAP5 の AE 化	69

8. 1 概 要.....	69
8. 2 AE化のための調査.....	69
8. 3 AE化のための変更作業	71
8. 4 AE化のための変更作業後のロードモジュールについて.....	78
9. イオンビーム計算コードと磁場計算コードの結合.....	79
9. 1 概 要.....	79
9. 2 3次元磁場解析コード ELF／MAGICとの結合	79
9. 3 図化出力の改良：等高線プロット図の分割	81
9. 4 図化出力の改良：1次元図化出力の追加	82
9. 5 実行と評価.....	84
10. おわりに	108
謝 辞.....	108
参考文献.....	109
付録 VBIFD3,VFSINT 使用手引	111

Contents

1.	Introduction	1
2.	Vectorization of the MEUDAS4	3
2.1	Outline	3
2.2	Dynamic Analysis	3
2.3	Vectorization Method	3
2.4	Performance by Vectorization	4
3.	Vectorization of the MEUDAS4-FFT	7
3.1	Outline	7
3.2	Dynamic Analysis	7
3.3	Vectorization Method	7
3.4	Performance by Vectorization	8
4.	Vectorization of the FORCE	13
4.1	Outline	13
4.2	Basic Equations	13
4.3	Dynamic Analysis	14
4.4	Vectorization Method	14
4.5	Results by Vectorization	16
5.	Vectorization of the STREAM Version 2.6	30
5.1	Outline	30
5.2	Basic Equations	30
5.3	Boundary Conditions	31
5.4	Vectorization	32
5.5	Reduction of the Number of I/O	44
5.6	Results	45
6.	Vectorization and Reorganization of a Free Electron Laser Code	48
6.1	Outline	48
6.2	Vectorization	48
6.3	Restart Function	52
6.4	Conclusion	53
7.	Improvement of the HEATING7-VP	54
7.1	Outline	54
7.2	Definition of Physical Region	54
7.3	Results	67
8.	Adaptation of Address Extension Function to the SCDAP/RELAP5	69
8.1	Outline	69

8.2 Adaptability Investigation of Address Extension Function	69
8.3 Reorganization of the Code	71
8.4 Results	78
9. Inclusion of the Magnetic Field Calculation Code into Ion-beam Calculation Code	79
9.1 Outline	79
9.2 Inclusion of the Magnetic Field Analysis Code ELF/MAGIC	79
9.3 Modification of Graphic Function	81
9.4 Addition of the One-dimensional Graphic Function	82
9.5 Test Run and Results	84
10. Concluding Remarks	108
Acknowledgements	108
References	109
Appendix Usage of VBIFD3 and VFSINT	111

1. はじめに

情報システムセンター・外来研究員室（以降、外来研）では、スーパーコンピュータの有効利用のために現在精力的に原子力コードのベクトル化、及び改良を実施している。平成3年度は主に核融合コード、流体コード、安全性コード、自由電子レーザーシミュレーションコード等全部で8本のコードについてベクトル化、及び改良を行った。ベクトル化を行ったコードはトカマク平衡、及び局所MHD安定性解析コードMEUDAS4、同じくMEUDAS4 FFT版、コイルの電磁応力解析における電磁力計算コードFORCE、3次元熱流体解析コードSTREAM、自由電子レーザーコードである。また改良を行ったコードは、3次元熱解析コードHEATING7-VP、炉心燃料損傷解析コードSCDAP/RELAP5、イオンビーム軌道計算コードNBI3DGFNである。

MEUDAS4、及びMEUDAS4 FFT版コードは、日本原子力研究所（以降、原研）において既にベクトル化が行われていた。しかし、コード内部のSSL II（科学計算用サブルーチンライブラリ）サブルーチンの呼び出しを含むDOループがベクトル化されておらず、今回の作業ではSSL IIサブルーチンを含めたベクトル化を行った。これにより、VP-2600上におけるベクトル化計算では、スカラ計算と比較してMEUDAS4は約4.4倍、MEUDAS4 FFT版は約2.3倍～約4.9倍の高速化が計られた。

FORCEコードは、コード内部で使用しているSSLサブルーチンでCPU時間が多くかかっている。そのためSSLサブルーチンに相当する解法サブルーチンを新たに作成し、さらにベクトル化することによって高速化を計った。これによりVP-2600上において約2.6倍～約6.2倍に高速化された。

STREAMコードは、以前、原研において千葉らによってベクトル化済である。しかし、STREAMコードがバージョンアップ（改良）されたため、新たにベクトル化を行った。バージョンアップ版には旧版に用いられていたベクトル化手法をさらに改良して導入し、高速化を計った。これにより、VP-2600上において約1.9倍～約5.4倍の高速化が計られた。

自由電子レーザーコードは、基礎物理研究用コードである。実際のレーザー発振をシミュレーションするにはかなりの計算時間がかかり、高速化が望まれていた。また、長大計算のための入出力機能の改良も望まれていた。そこで今回はベクトル化を行うと同時に改良の提案も行った。

HEATING7-VPコードは対象物をいくつかのリージョン（メッシュ）に区切って熱解析を行うコードである。このリージョンは従来、原則的に四角形または立方体しか定義することができなかった。今回、更に精密な解析を行うために、リージョンを三角形または三角柱に区切れるように改良を行った。

SCDAP/RELAP5コードは、軽水炉における炉心燃料損傷のシミュレーションを行うコードである。解析対象物を大きくすると、コード自体が大きくなってしまい、原研のシステムではメモリ基本領域で動作させることは不可能であった。そのため、今回の改良でコードをAE化することにより、対象物を大きくすることが可能となった。

NBI3DGFNコードは、3次元磁場解析コード(GFUN)を内包しているが、計算ケースによっては、

GFUNでの計算結果が正しく収束しない場合があった。そこで磁場解析部分を他の磁場解析コードMAGICに置き換えることによって、正しく収束できるように改良を行った。
本報告書では、これらのベクトル化、及び改良方法等について詳しく述べる。

2. MEUDAS 4 のベクトル化

2.1 概要

MEUDAS 4 はトカマク平衡及び局所MHD安定性解析コードであり、磁気流体平衡方程式をダブル・サイクリック・リダクション法で解いている。MEUDAS 4 は、既にベクトル化チューニングが施されているが、磁気流体平衡方程式の計算においてSSL II⁽¹⁾ サブルーチンの呼出しを含むDOループがあり、この箇所におけるコストがプログラム全体に対して大きな割合を占めるため、SSL II サブルーチンを含んだベクトル化を行うこととなった。ベクトル化の対象となるSSL II サブルーチンは、2次元有限領域積分（不等間隔格子点入力、B-spline 2次元補間式）DBIFD3である。

2.2 プログラム解析

Table 2.1 に、動的挙動解析ツールFORTUNE⁽²⁾ を用いてMEUDAS 4 を解析した結果を示す。また、Fig. 2.1にDBIFD3とその下位ルーチンのTREE構造を、Fig. 2.2にLINTRCの概略を示す。DBIFD3とその下位ルーチンDUCAD1, DUBAS1, MGSSLの占めるコストの割合は全体の82%になっている。DBIFD3はサブルーチンLINTRCのDO 215のループ中の二箇所から呼ばれている。

2.3 ベクトル化の方針

DBIFD3はxy平面の格子点(x_i, y_j) ($x_1 < x_2 < \dots < x_n, y_1 < y_2 < \dots < y_m$)において関数値 $f_{ij} = f(x_i, y_j)$ が与えられたとき、点P(v_x, v_y)における補間値、偏微分値、あるいは領域 [$x_1 \leq x \leq v_x, y_1 \leq y \leq v_y$] 上の二重積分値を求めるサブルーチンである。MEUDAS 4 のサブルーチンLINTRCでは、DOループの中からDBIFD3を呼び出すことで、関数値 $f_{ij} = f(x_i, y_j)$ 上の複数個の点 $P_1(v_{x1}, v_{y1}) \sim P_N(v_{xN}, v_{yN})$ における x に関する偏微分値と y に関する偏微分値を求めている。DOループの回転数は偏微分値を求める点の数であり、今回のテストデータでは50である。

このDOループに含まれる偏微分計算のコストが非常に高いにもかかわらず、外部手続きDBIFD3の呼出しがあるためにベクトル化が妨げられている。従って、このDOループをサブルーチンDBIFD3、あるいはその下位ルーチンであるDUCAD1, DUBAS1に引き込み、点Pの座標(v_x, v_y)に依存する計算部分に対してDOループを再構成し、ベクトル化を可能にした。

ベクトル化にあたって、以下のような作業を要した。

① DOループに外部手続き呼出しがある場合、または大きく実行の流れが変わらるような制御文(GO TO 文)がある場合はDOループを分割し、外部手続き呼出し部分に関してはDOループの

引込みを行った。

- ② 上位手続きからD0ループの引込みを行った場合、既にその手続き内にD0ループがある場合は、引き込んだD0ループを最内ループになるように入替えを行った。
- ③ 点Pの座標(v_x, v_y)を格納する変数を配列化した。
- ④ ①、③の作業に伴い、DBIFD3で使用する作業配列を1次元から2次元に拡張した。また、新たな作業配列を追加した。
- ⑤ ①～④の作業で、プログラムを変更したルーチンは、名前を変更してオリジナルのルーチンと区別出来る様にした。プログラムを変更したルーチンと変更後の名前を以下に示す。

[プログラムを変更したルーチン]	[変更後の名前]
D F S I N T	⇒ V F S I N T
D U C A D 1	⇒ V E C C A D
D U B A S 1	⇒ V E C B A S, V C B A S 2

DUBAS1に関しては、点Pの座標における偏微分計算とある一つの離散点における偏微分計算の二通りの呼出し方があるためにVECBASとVCBAS2の2つのサブルーチンを作成した。前者については、前述のD0ループ引込みを行ったが、後者は点Pの座標に依存しないために作業配列の拡張に伴う修正のみを行った。但し、VCBAS2の呼出しはDBIFD3の入力パラメタに依存しており、MEUDAS4においては、VCBAS2が呼び出されるようなケースはない。

この様にして作成されたサブルーチンVBIFD3とその下位ルーチンの構造をFig. 2.3に示す。

2.4 ベクトル化効果

オリジナル版とベクトル化版の両プログラムを、VP2600のスカラモードとベクトルモードで実行したときのCPU時間、VU時間、及びベクトル化率をTable 2.2に示す。また、両者の計算結果は一致した。

今回のチューニングによる速度向上率をTable 2.3に示す。ベクトルモードで実行した場合、ベクトル化版では、オリジナル版に対して3.9倍の高速化が計れた。オリジナル版をスカラモードで実行した場合と比較すると、4.4倍の高速化が計れている。また、サブルーチンDBIFD3、VBIFD3で比較すると4.7倍の高速化が計れている。

今回作成したサブルーチンVBIFD3は、原研の他のユーザが使用可能な様に整備し、JSSL(原研版・科学用サブルーチンライブラリ)に登録した。付録にVBIFD3の使用手引を添付する。また、VBIFD3のソースプログラムは原研内での使用に限り公開することにしている。

Table 2.1 Cost of MEUDAS4

No.	ルーチン名	コスト (%)
1	DUBAS1	60.2
2	LINTRC	14.1
3	DUCAD1	12.8
4	DBIFD3	9.2
5	EQPDE	1.2
6	MGSSL	0.8

Table 2.2 CPU time for each version

プログラム	モード	CPU時間 (sec)	VU 時間 (sec)	ベクトル化率 (%)
オリジナル版	スカラ	108.87	—	—
	ベクトル	96.99	0.70	11.6
ベクトル化版	スカラ	110.29	—	—
	ベクトル	24.64	14.77	91.9

Table 2.3 Speedup of the vector version

比較対象	速度向上比
オリジナル版 / オリジナル版 スカラモード / ベクトルモード	1.12
オリジナル版 / ベクトル化版 スカラモード / ベクトルモード	4.42



Fig. 2.1 Tree structure of DBIFD3 and its slave routines

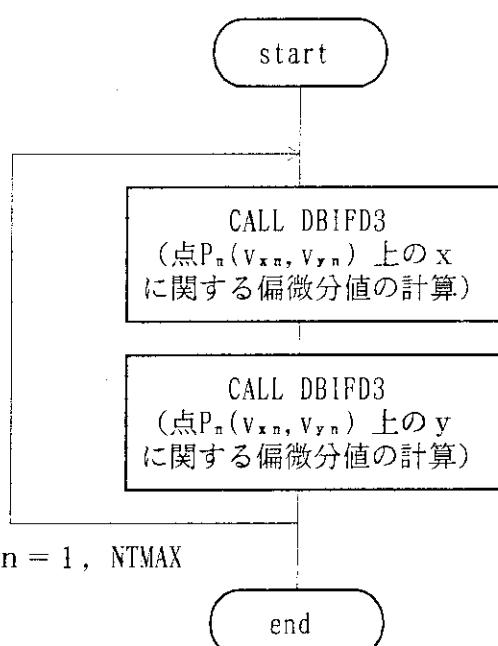


Fig. 2.2 Computation flow of LINTRC



Fig. 2.3 Tree structure of VBIFD3 and its slave routines

3. MEUDAS 4 FFT版のベクトル化

3.1 概要

MEUDAS 4 FFT版は、磁気流体平衡方程式を高速フーリエ変換（FFT）を用いて解いている。MEUDAS 4 FFT版では、高速フーリエ変換にSSL II⁽¹⁾ サブルーチンである離散型 sine 変換（台形公式、2 基底FFT）ルーチン DFSINT を使用している。DFSINT は DO ループの中から呼び出されており、DFSINT の占めるコストがプログラム全体に対し比較的高い。よって、DFSINT に呼び出し元であるサブルーチン GFFT1, PSIC1 の DO ループを引き込むことでベクトル化を行った。また、サブルーチン FAIC1 の DO ループの中にもサブルーチン CAL の呼び出しがあり、DFSINT の場合と同様に DO ループを CAL 中に引き込み、ベクトル化した。

3.2 プログラム解析

Table 3.1 に動的挙動解析ツール FORTUNE⁽²⁾ を用いて MEUDAS 4 FFT版を解析した結果を示す。DFSINT とその下位ルーチン DFSINM, DUPNR2 の占めるコストの割合は全体の 21 % になる。また、FAIC1 と FAIC1 中の DO ループから呼ばれるサブルーチン CAL の占めるコストの割合は全体の 12 % になっている。サブルーチン GFFT1, PAIC1, FAIC1 は何れもサブルーチン FACR1S から呼び出されている。FACR1S 以下のサブルーチンの TREE 構造を Fig. 3.1 に示す。また、Fig. 3.2 にサブルーチン DFSINT とその下位ルーチンの構造を示す。

3.3 ベクトル化の方針

3.3.1 サブルーチン DFSINT とその下位ルーチンのベクトル化

DFSINT は、周期 2π の奇関数 $x(t)$ の半周期を n 等分した n 個（但し、 $n = 2^k$ ）の標本からなる集合 $\{x_i\}$

$$x_i = x(\pi j/n), \quad j=0, 1, \dots, n-1$$

が与えられたとき、台形公式による離散型 sine 変換、又はその逆変換を FFT により行うものである。DFSINT とその下位ルーチンに含まれる DO ループは標本数 n を回転数としたものであるが、回帰演算が多いためにほとんどがベクトル化不可能である。DFSINT の呼び出しが、サブルーチン GFFT1, PSIC1, FFT1 の 3箇所である。何れの箇所においても、DO ループの中から DFSINT を呼び出すことで、複数組の標本集合に対して変換を行っている。よって、DFSINT の呼出しを含む DO ループを DFSINT とその下位ルーチンに引き込むことによってベクトル化を行う。但し、FFT1 では、DFSINT の呼び出しを含む DO ループの回転数が非常に小さく、ベクトル化の効果が期待出来ないので、DO ループ引込みは GFFT1 と PSIC1 についてのみ行った。GFFT1 と PSIC1 の構造を Fig. 3.3, Fig. 3.4 に示す。GFFT1 では、フーリエ係数を、PSIC1 ではフーリエ級数を

求めている。

ベクトル化にあたって、以下の作業を要した。

- ① DOループ中に外部手続き呼出しがある場合、または大きく実行の流れが変わる様な制御文(GO TO 文)がある場合はDOループを分割し、手続き呼出し部分に関してはDOループの引っ越しを行った。
- ② 上位手続きからDOループの引っ越しを行った場合、既にその手続き内にDOループがある場合は、引き込んだDOループが最内ループになるように入れ換えを行った。
- ③ 標本集合を格納する配列を2次元化した。
- ④ DOループの引っ越しによってベクトル化するあたり、より高速化が計れるようにGFFT1, PSIC Iにおいて標本を格納する配列の方向を入れ換えた。
- ⑤ 変換の対象となる標本集合の組を限定出来るように、範囲を示す引数を2つ加えた。
- ⑥ ①～⑤の作業で、プログラムを変更したルーチンは、名前を変更してオリジナルのルーチンと区別出来る様にした。プログラムを変更したルーチンと変更後の名前を以下に示す。

[プログラムを変更したルーチン]	[変更後の名前]
D F S I N T	⇒ V F S I N T
D U P N R 2	⇒ V U P N R 2
V U S I N M	⇒ V U S I N M

こうして作成したサブルーチンVFSINTとその下位ルーチンのTREE構造をFig. 3.5に示す。

3.3.2 サブルーチンCALのベクトル化

サブルーチンFAIC1では、外側に標本集合の組に関するDOループがあり、その中に幾つかのDOループとサブルーチンCALの呼出しがある。また、CAL中にもいくつかのDOループが存在するが、何れも回帰演算となるためにベクトル化が不可能である。

従って、FAIC1の標本集合の組に関するDOループをCALに引き込み、引き込んだDOループが最内ループになるようにDOループを入れ換えた。また、FAIC1においても同様にDOループの入れ換えを行った。この作業において、FAIC1の変数AKN2を配列化し、CALの配列A, Gを2次元化した。

3.4 ベクトル化効果

FFTの標本集合の組数が129, 513のときの、オリジナル版とベクトル化版でのCPU時間、VU時間をTable 3.2, Table 3.3に示す。時間測定は、VP2600のベクトルモードで行った。ベクトル化版ではオリジナル版に対して、2.3～2.9倍の高速化が計れている。また、オリジナル版をスカラモードで実行した場合と比較すると、4.9倍の高速化が計れている。また、今回のベクトル化作業において、オリジナル版とベクトル化版の計算結果は全て一致した。

今回作成したサブルーチンVFSINTは、原研の他のユーザにも使用可能な様に整備し、JSS Lライブラリに登録した。付録にVFSINTの使用手引を添付する。また、VFSINTのソースプログラムは原研内での使用に限り公開する。

Table 3.1 Cost of MEUDAS4-FFT

NO	ルーチン名	コスト (%)	ベクトル化率 (%)
1	F L X F U N	3 8 . 6	99.99
2	D U S I N M	1 5 . 4	7.47
3	E Q L I N	1 2 . 5	0.00
4	C A L	8 . 5	0.00
5	E Q R B P	5 . 3	99.98
6	F A I C 1	3 . 3	98.63
7	D F S I N T	3 . 2	0.00
8	E Q R C U	2 . 8	79.68
9	D U P N R 2	2 . 4	0.00
10	G R Z 1	1 . 4	96.99

Table 3.2 CPU time for each version
(The number of sets of samples is 129)

プログラム	実行モード	全 CPU時間 (sec)	体 VU 時間(sec)	FACR1S 以下のルーチンの CPU時間 (sec)
オリジナル版	スカラ	7.01	—	2.86
	ベクトル	3.43	0.41	2.14
ベクトル化版	ベクトル	1.43	0.34	0.14

Table 3.3 CPU time for each version
(The number of sets of samples is 513)

プログラム	実行モード	全 CPU時間 (sec)	体 VU 時間(sec)	FACR1S 以下のルーチンの CPU時間 (sec)
オリジナル版	スカラ	90.44	—	54.94
	ベクトル	53.54	3.17	36.58
ベクトル化版	ベクトル	18.33	2.01	1.27

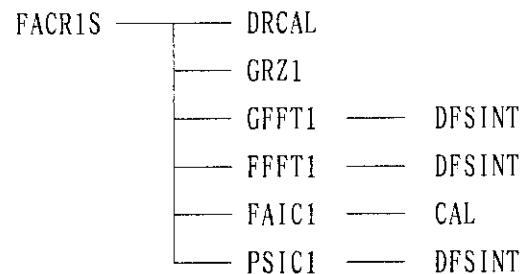


Fig. 3.1 Tree structure of FACR1S and its slave routines

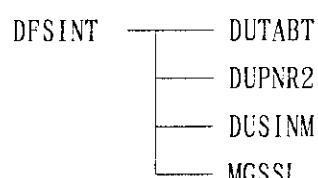


Fig. 3.2 Tree structure of DFSINT and its slave routines

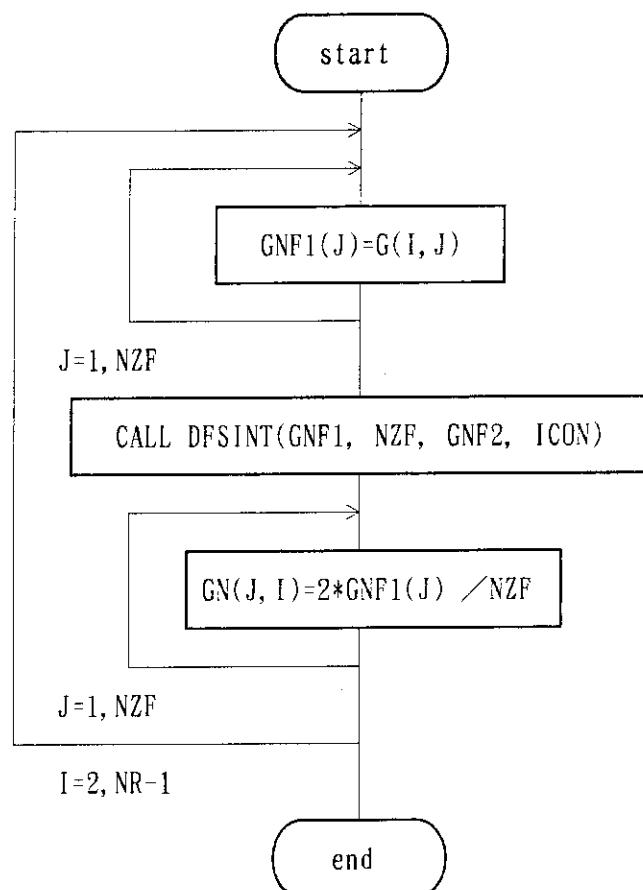


Fig. 3.3 Computation flow of subroutine GFIT1

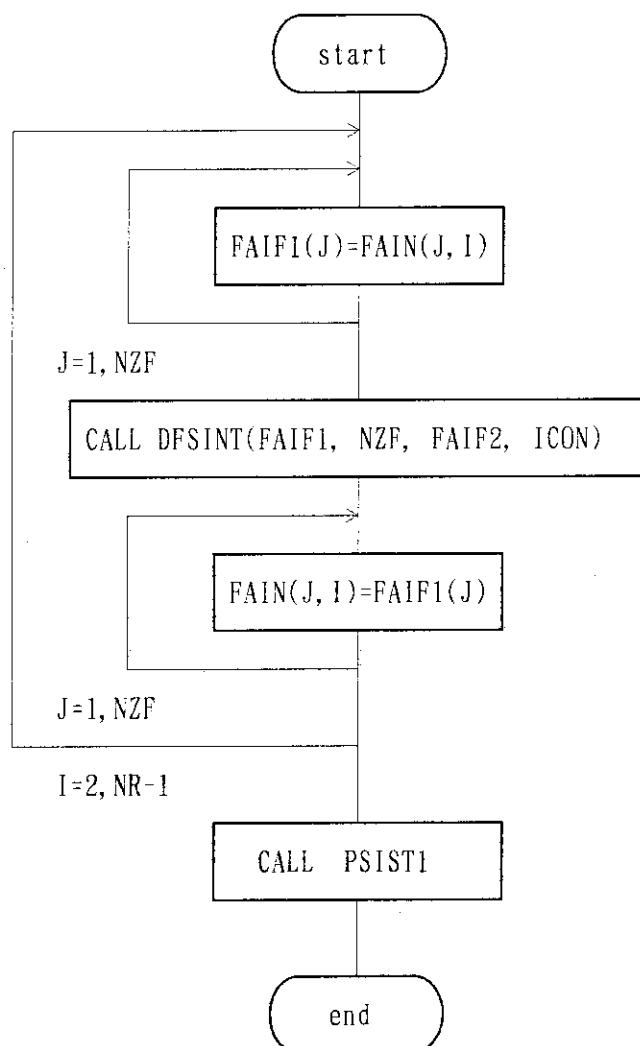


Fig. 3.4 Computation flow of subroutine PSIC1

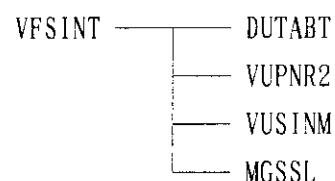


Fig. 3.5 Tree structure of VFSINT and its slave routines

4. FORCE コードのベクトル化

4.1 概要

磁界解析コードFORCE⁽³⁾はコイルの電磁力計算を主目的として開発されたコードである。具体的には、円弧と直線で作られる任意の形状のコイルが複数個、空間の任意の位置に置かれた場合に、空間内の任意の点の磁界を計算する。

FORCEは従来、主にTSS上での会話形式で使用されてきた。しかし、解析対象が大規模になってきたため計算時間が長くなり、TSSで実行することはCPU時間の制約上不可能になった。また、バッチ処理においてもCPU時間を多く使用することになった。これらの理由からベクトル化の必要が出てきた。本章ではFORCEのベクトル化作業について述べる。

4.2 基本方程式⁽⁴⁾

Fig. 4.1に示す体積電流を流す管Cを考え、点Mを通る微小電流を dI 、その微小距離を dl 、Mから測定点Pまでの距離を r とする。この全体積電流による点Pにおける磁束密度は真空の透磁率を μ_0 とすると、ビオーサバールの法則により(4.1)式で示される。

$$\vec{B} = \frac{\mu_0}{4\pi} \int_I dI \int_I \left[\vec{dl} \times \frac{\vec{r}}{r^3} \right] \quad (4.1)$$

ここで電流密度を \vec{J} とすると、 $dI \cdot \vec{dl} = \vec{J} \cdot d\vec{v}$ で示されるので

$$\vec{B} = \frac{\mu_0}{4\pi} \int_v \left[\vec{J} \times \frac{\vec{r}}{r^3} \right] d\vec{v} \quad (4.2)$$

となる関係式を得る。この式を各種の形状のコイルに応用し、磁束密度を求める。(4.2)式はベクトル・ポテンシアル \vec{A} を用いると(4.3)式の関係式が得られる。

$$\vec{B} = - \frac{\mu_0}{4\pi} \int_v \left[\vec{J} \times \text{grad} \left(\frac{1}{r} \right) \right] d\vec{v} = \frac{\mu_0}{4\pi} \text{rot} \int_v \frac{\vec{J}}{r} d\vec{v} = \text{rot} \vec{A} \quad (4.3)$$

ビオーサバルの法則は一般的には3次元空間における現象を示しているが、垂直無限長導体については無限長方向についての積分が収束するので2次元の問題として一般化できる。
 \vec{l} を2次元空間での測定点Pから電流へ向かう距離とすると、磁束密度 \vec{B} は(4.4)式で表される。

$$\vec{B} = \frac{\mu_0}{4\pi} \int_s \left[\frac{\vec{J} \times \vec{l}}{l^2} \right] ds \quad (4.4)$$

従って、垂直無限長導体による磁界の問題は2次元問題として扱うことが出来、その形状、分布などを比較的容易に検討できる。

4.3 動的挙動分析

実行時における計算時間の分布を調べるために、動的挙動解析ツール FORTUNEを使用して各サブルーチンについて調査を行った。FORTUNE の結果をTable 4.1 に示す。この図より、外部関数であるFRX、FRY、FRZ の3ルーチンが全体の約97%のコストを占めており、これらをベクトル化の対象とした。これらの関数はFig. 4.2に示されるように、サブルーチンSUMCでEXTERNAL宣言され、関数SUM、FUNC、GAUSS に次々に引き渡され、最終的に科学用サブルーチンライブラリSSL⁽⁵⁾ の積分値解法ルーチンGAS12D、GAS24D、GAS32Dに引数として渡されている。すなわちSSLサブルーチンライブラリ内部での関数値計算に使用されている。これらのSSLサブルーチンは一次元有限区間の積分値をガウスルジャンドル公式によって求めるものである。FORTUNE では解析対象のコード以外（外部のライブラリなど）は解析できないためコスト表には現れないが、実際にはSSLサブルーチンにもコストがかかっているはずである。SSLサブルーチンとFRX、FRY、FRZ をまとめてベクトル化を行うと効果が上がる。しかし、SSLはSSL IIと異なり、VP用のライブラリは用意されていない。また、SSLのソースは公開されていないため、これらのサブルーチンを作成することにした。

4.4 ベクトル化手順

4.4.1 積分ルーチンの作成

SSLのサブルーチンのベクトル化を行うことは著作権の問題があるため、同機能を持つサブルーチンを作成する。ガウスルジャンドル公式によって積分値を求めるには、分点と重みを求めるサブルーチンと、積分値を求めるルーチンの2種類のサブルーチンを作成する必要がある。

(1) 分点と重みを求めるサブルーチンGASLEG

ガウスルジャンドル公式にて積分値を求めるには、前もって(4.5)式における分点 x_i 及び重み w_i を求める必要がある。これらを求めるサブルーチンGASLEGは(株)オーム社「数値計算ハンドブック」⁽⁶⁾より引用した。

$$\int_a^b f(x)dx = \sum_{i=1}^n w_i f(x_i) \quad (4.5)$$

(2) 積分値計算ルーチンの作成及び外部関数の展開

(4.5)式を基に積分値を計算する。これも(株)オーム社「数値計算ハンドブック」より引用した。本コードにおいて、(4.5)式の $f(x)$ は外部関数 FRX、FRY、FRZ に相当する。これらは作成した積分値計算サブルーチンの中の D0 ループ内で参照される。D0 ループ内での外部参照は自動ベクトル化されない。そこで 3 つの積分値計算ルーチンを作成し、それぞれの D0 ループ内で FRX、FRY、FRZ を内部展開した。これらを順にサブルーチン GAUSSX(Fig. 4.3)、GAUSSY(Fig. 4.4)、GAUSSZ(Fig. 4.5)とした。

4.4.2 積分計算ルーチンの組み込み

新たに作成したサブルーチンを以下のように組み込んだ。

(1) 分点と重みを求めるサブルーチンGASLEGの組み込み

分点と重みは実行時に一度計算すれば良い。オリジナル版では、標準値設定ルーチンCLEAR とオプション入力ルーチンOPTI のどちらかで、積分点数が設定されている。従ってサブルーチン GASLEG はこれらのサブルーチンの中で呼ぶように変更し、分点と重みの計算をするようにした。変更を行ったサブルーチンCLEAR、及びOPTIのソースの一部をFig. 4.6、及びFig. 4.7 に示す。

(2) GAUSSX、GAUSSY、GAUSSZの組み込み

外部関数 FRX、FRY、FRZ と S S L サブルーチンの呼び出しの関係は、オリジナル版では Fig. 4.2 のようになっていた。これをベクトル化版では Fig. 4.8 のように変更を施した。主な変更点は、オリジナル版ではサブルーチンFUNCから外部関数 GAUSSまでは FRX、FRY、FRZ を EXTERNAL 文で宣言し、引数としてこれらの外部関数を直接渡していたが、ベクトル化版では FRX、FRY、FRZ のいずれかを使用するかのフラグIFR を設定し、そのフラグによって積分値計算ルーチンGAUSSX、GAUSSY、GAUSSZを選択するようにした。これらのサブルーチンのリストをFig. 4.9～Fig. 4.12 に示す。

4.5 ベクトル化の結果

4.5.1 サンプルデータによるベクトル化効果

ベクトル化の効果を測定するため、VP-2600 上でオリジナル版のスカラモード、及びベクトル化版のスカラモード、ベクトルモードでのそれぞれCPU 時間を測定した。この結果をTable 4.2 に示す。使用したサンプルデータは、計算点数が3000点のものである。サンプルデータを Fig. 4.13、実行JCL をFig. 4.14 に示す。尚、参考としてベクトル化版の各サブルーチンのコスト表をTable 4.3 に示す。Table 4.2 において

- オリジナル : オリジナルコードの全てのサブルーチンをスカラコンパイル(OPT(3))し、VP-2600 で実行したもの
- VP版スカラモード : ベクトル化版の全てのサブルーチンをスカラコンパイル(OPT(3))し、VP-2600で実行したもの
- VP版VPモード : ベクトル化版の全てのサブルーチンをベクトルコンパイルし、VP-2600 で実行したもの

である。コンパイラはFORTRAN77(V10L30) である。倍率は、オリジナルのCPU 時間をベクトル化版ベクトルモードのCPU 時間で割ったものと、ベクトル化版スカラモードのCPU 時間をベクトル化版ベクトルモードCPU の時間で割ったものをそれぞれ示している。

4.5.2 ベクトル化版による計算結果の妥当性

オリジナルとベクトル化版の計算結果を比較したところ、ほぼ値は一致している。ただし、以下のような不一致点が見られた。

$$\begin{array}{rcl} 0.0 & \Rightarrow & -0.0 \\ -0.0 & \Rightarrow & 0.0 \end{array}$$

原因是 0に近い値で切り捨てられている部分の誤差と考えられるが、これは許容範囲であるよってベクトル化版の計算結果は妥当であると言える。

Table 4.1 Dynamic behavior of the original version of FORCE code

I	I	NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	X	0.....1.....2.....3.....4.....
I	I	00001	FRZ	1	34	0	27851904	0.8487785E+10	44.8	I.....
I	I	00002	FRX	1	19	0	27851904	0.50989E+10	26.9	I.....
I	I	00003	FRY	1	20	0	25343426	0.468883E+10	24.7	I.....
I	I	00004	GAUSS	1	53	0	3376968	0.118174E+09	0.6	I.....
I	I	00005	VOLUME	1	180	0	540672	0.116644E+09	0.6	I.....
I	I	00006	FL	1	35	0	2760924	0.109216E+09	0.6	I.....
I	I	00007	FUNC	1	26	0	510080	0.107770E+09	0.6	I.....
I	I	00008	TRANS2	1	16	0	234432	0.479475E+08	0.3	I.....
I	I	00009	SUMC	1	59	0	2312	0.454219E+08	0.2	I.....
I	I	0010	AXYZCK	1	36	0	690231	0.282999E+08	0.2	I.....
I	I	0011	SUM	1	253	0	56096	0.173248E+08	0.1	I.....
I	I	0012	TWHICH	1	252	0	2112	0.134194E+08	0.1	I.....
I	I	0013	TFIELD	1	262	0	1	0.122024E+08	0.1	I.....
I	I	0014	VFORCE	1	563	0	56096	0.645104E+07	0.0	I.....
I	I	0015	LGCONV	1	115	0	1	0.643799E+07	0.0	I.....
I	I	0016	VPOWER	1	502	0	67584	0.432538E+07	0.0	I.....
I	I	0017	F	1	20	0	2112	0.425378E+07	0.0	I.....
I	I	0018	ATCOIL	1	441	0	33792	0.170655E+07	0.0	I.....
I	I	0019	PFIELD	1	137	0	1	0.161294E+07	0.0	I.....
I	I	0020	TABLE	1	184	0	2112	0.145306E+07	0.0	I.....
I	I	0021	SUML	1	20	0	1	0.471534E+06	0.0	I.....
I	I	0022	REST	1	95	0	4008	0.438208E+06	0.0	I.....
I	I	0023	PWHICH	1	117	0	500	0.393247E+06	0.0	I.....
I	I	0024	CHOP	1	446	0	1	0.358723E+06	0.0	I.....
I	I	0025	APCOIL	1	259	0	1800	0.199800E+06	0.0	I.....
I	I	0026	LGCONV	1	126	0	2112	0.138504E+06	0.0	I.....
I	I	0027	FDN	1	553	0	1	0.126960E+06	0.0	I.....
I	I	0028	PCO1	1	293	0	13	0.936110E+05	0.0	I.....
I	I	0029	ECHO	1	69	0	1	0.840560E+05	0.0	I.....
I	I	0030	TFCO	1	533	0	1	0.800470E+05	0.0	I.....
I	I	0031	PROLOG	1	93	0	1	0.29810E+05	0.0	I.....
I	I	0032	TCOI	1	689	0	16	0.219440E+05	0.0	I.....
I	I	0033	AUTELM	1	36	0	1	0.902900E+04	0.0	I.....
I	I	0034	AFORCE	1	523	0	1	0.503500E+04	0.0	I.....
I	I	0035	FINI	1	110	0	31	0.240300E+04	0.0	I.....
I	I	0036	PROC2	1	216	0	1	0.160000E+04	0.0	I.....
I	I	0037	WATCH	1	28	0	4	0.127400E+04	0.0	I.....
I	I	0038	ATDIV	1	93	0	20	0.109600E+04	0.0	I.....
I	I	0039	INTWI	1	74	0	2	0.31400E+03	0.0	I.....
I	I	0040	PROC1	1	216	0	1	0.790000E+02	0.0	I.....
I	I	0041	CLEAR	1	140	0	1	0.740000E+02	0.0	I.....
I	I	0042	TITLE	1	87	0	1	0.240000E+02	0.0	I.....
I	I	0043	ENTCOM	1	16	0	1	0.220000E+02	0.0	I.....
I	I	0044	ENVRTI	1	29	0	1	0.200000E+01	0.0	I.....
I	I	0045	MAIN	1	85	0	1	0.180000E+01	0.0	I.....
I	I	0046	OPTI	1	417	0	0			I.....
I	I	0047	GRAP	1	166	0	0			I.....
I	I	0048	SPOT	1	303	0	0			I.....
I	I	0049	HELP	1	81	0	0			I.....
I	I	0050	PASS	1	176	0	0			I.....
I	I	0051	TRANS3	1	39	0	0			I.....
I	I	0052	INTNO	1	73	0	0			I.....
I	I	0053	WFEMAS	1	90	0	0			I.....
I	I	0054	FLIST	1	252	0	0			I.....
I	I	0055	GSHAPE	1	129	0	0			I.....
I	I	0056	PROC3	1	216	0	0			I.....

Table 4.2 CPU time of the original and vectorized versions

項目		積分点数	1 2	2 4	3 2
CPU 時間 (秒)	オリジナル		2' 57"11	5' 32"63	7' 23"29
	VP版スカラモード		2' 21"79	4' 26"37	5' 49"52
	VP版VPモード (VU時間)		1' 09"12 (50"54)	1' 10"43 (51"92)	1' 11"42 (52"93)
使用領域 (KB)	オリジナル		7948	7948	7948
	VP版スカラモード		7964	7964	7964
	VP版VPモード		8108	8108	8108
I/O 回数	オリジナル		217	217	217
	VP版スカラモード		217	217	217
	VP版VPモード		220	219	219
倍率	オリジナル/VP版VPモード		2.56	4.72	6.21
	VP版スカラモード/ VP版VPモード		2.05	3.78	4.89
ベクトル化率(%)			86.9	93.1	94.7

Table 4.3 Dynamic behavior of the vectorized version of FORCE code

NO.	ROUTINE	UNITS	LINES	ERR.	EXECUTIONS	COST	X	0.....1.....2.....3.....4.....
1 0001	GAUSSZ	1	48	0	1160496	0.802524E+10	45.4	*****
1 0002	GAUSSX	1	34	0	1160496	0.4468724E+10	26.5	*****
1 0003	GAUSSY	1	35	0	1055976	0.429943E+10	24.3	*****
1 0004	VOLUME	1	180	0	0.116648E+09	0.7	1	*****
1 0005	GAUSS	1	41	0	3376968	0.111140E+09	0.6	1
1 0006	FL	1	35	0	540672	0.109146E+09	0.6	1
1 0007	FUNC	1	30	0	27660924	0.1088226E+09	0.6	1
1 0008	TRANS2	1	16	0	510080	0.47947E+08	0.3	1
1 0009	SUMC	1	66	0	234432	0.461121E+08	0.3	1
1 0010	AXYZCK	1	36	0	2312	0.430856E+08	0.2	1
1 0011	SUM	1	29	0	690231	0.312706E+08	0.2	1
1 0012	TWHICH	1	252	0	56096	0.123348E+08	0.1	1
1 0013	TFIELD	1	262	0	2112	0.134196E+08	0.1	1
1 0014	VFORCE	1	567	0	1	0.121120E+08	0.1	1
1 0015	LGCONV	1	115	0	56096	0.645104E+07	0.0	1
1 0016	VPOWER	1	502	0	67581	0.643399E+07	0.0	1
1 0017	F	1	20	0	1	0.432358E+07	0.0	1
1 0018	ATCOIL	1	448	0	1	0.428519E+07	0.0	1
1 0019	PFIELD	1	137	0	2112	0.170505E+07	0.0	1
1 0020	TABLE	1	184	0	1	0.16129E+07	0.0	1
1 0021	SUML	1	20	0	33792	0.145306E+07	0.0	1
1 0022	REST	1	95	0	1	0.471134E+06	0.0	1
1 0023	PWHICH	1	117	0	4008	0.435208E+06	0.0	1
1 0024	CHOP	1	46	0	500	0.393247E+06	0.0	1
1 0025	APCOIL	1	243	0	1	0.369127E+06	0.0	1
1 0026	LGCONV	1	126	0	1800	0.199300E+06	0.0	1
1 0027	FDEN	1	53	0	2112	0.138304E+06	0.0	1
1 0028	PCOI	1	295	0	15	0.126362E+06	0.0	1
1 0029	ECHO	1	69	0	13	0.936110E+05	0.0	1
1 0030	TCFO	1	533	0	1	0.840160E+05	0.0	1
1 0031	PROLOG	1	93	0	1	0.800470E+05	0.0	1
1 0032	GASLEG	1	91	0	1	0.359865E+05	0.0	1
1 0033	TCOI	1	689	0	1	0.28910E+05	0.0	1
1 0034	AUTELM	1	36	0	16	0.219440E+05	0.0	1
1 0035	AFORCE	1	523	0	1	0.902000E+04	0.0	1
1 0036	FINI	1	110	0	1	0.503200E+04	0.0	1
1 0037	PROC2	1	216	0	31	0.240300E+04	0.0	1
1 0038	WATCH	1	28	0	1	0.160300E+04	0.0	1
1 0039	ATDIV	1	93	0	4	0.127400E+04	0.0	1
1 0040	INTWI	1	74	0	20	0.109400E+04	0.0	1
1 0041	PROC1	1	216	0	2	0.311000E+03	0.0	1
1 0042	CLEAR	1	145	0	1	0.810000E+02	0.0	1
1 0043	TITLE	1	87	0	1	0.740000E+02	0.0	1
1 0044	ENTCOM	1	16	0	1	0.240000E+02	0.0	1
1 0045	ENVRTMT	1	29	0	1	0.220000E+02	0.0	1
1 0046	MAIN	1	85	0	1	0.200000E+01	0.0	1
1 0047	GCNTR	1	187	0	0	0	0	1
1 0048	INTNO	1	73	0	0	0	0	1
1 0049	GRAP	1	166	0	0	0	0	1
1 0050	PROC3	1	216	0	0	0	0	1
1 0051	JUNGLE	1	389	0	0	0	0	1
1 0052	GHIST	1	336	0	0	0	0	1
1 0053	RECO	1	79	0	0	0	0	1
1 0054	HELP	1	81	0	0	0	0	1
1 0055	OPTI	1	422	0	0	0	0	1
1 0056	FRY	1	20	0	0	0	0	1

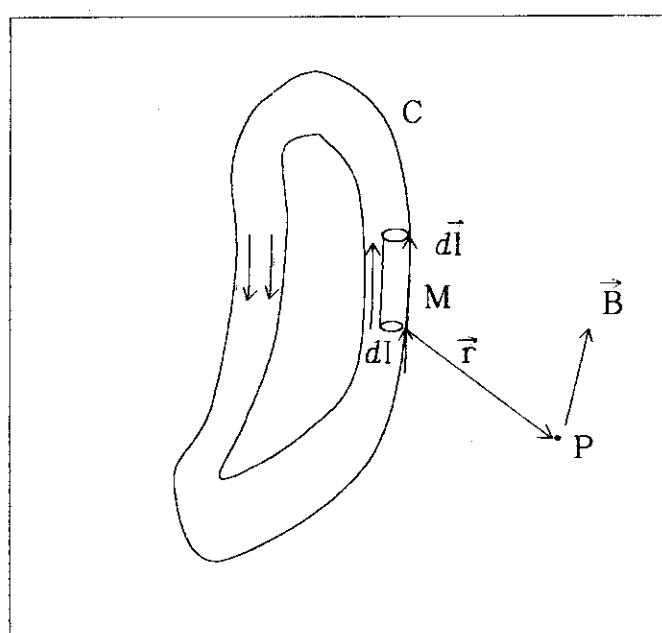


Fig. 4.1 Biot-Savart law by volume current

```

SUBROUTINE SUMC (X, Y, Z, FS, FE, R1, R2, ZS, ZE, CD, BXX, BYY, BZZ)
  EXTERNAL FRX, FRY, FRZ

  BX= CD*SUM(FRX, F1, F2, R0, R1, R2, Z1, Z2)
  BY= CD*SUM(FRY, F1, F2, R0, R1, R2, Z1, Z2)
  BZ=-CD*SUM(FRZ, F1, F2, R0, R1, R2, Z1, Z2)

  RETURN
  END

REAL FUNCTION SUM*8 (FUN, F1, F2, R0, R1, R2, Z1, Z2)
  EXTERNAL FUN

  SUM= FUNC (F1, F2, R0, R2, Z2, FUN)
  *      -FUNC (F1, F2, R0, R1, Z2, FUN)
  *      -FUNC (F1, F2, R0, R2, Z1, FUN)
  *      +FUNC (F1, F2, R0, R1, Z1, FUN)

  RETURN
  END

REAL FUNCTION FUNC*8 (F1, F2, R0, R, Z, FON)
  EXTERNAL FON

10 S      =GAUSS(FON, F1, ZERO)
  FUNC =GAUSS(FON, ZERO, F2)+S

20 FUNC =GAUSS(FON, F1, F2)

  RETURN
  END

REAL FUNCTION GAUSS*8 (F, A, B)
  EXTERNAL F

  IF(IGAUSS.EQ.12) CALL GAS12D(A, B, F, GAUSSV)
  IF(IGAUSS.EQ.24) CALL GAS24D(A, B, F, GAUSSV)
  IF(IGAUSS.EQ.32) CALL GAS32D(A, B, F, GAUSSV)
  GAUSS=GAUSSV
  RETURN
  END

```

Fig. 4.2 Calling procedure of the SSL subroutines GAS12D, GAS24D, GAS32D
in original version

```

C
C      SUBROUTINE GAUSSX(A,B,X,W,N,S,IRC)
C
C      IMPLICIT REAL*8(A-H,O-Z)
COMMON /LABS/R0,R,Z
DIMENSION X(1),W(1)
PARAMETER(ZERO=0.0D0)
PARAMETER(HALF=0.5D0)
C
IF(N.LT.1.OR.A.GE.B) THEN
  IRC=100
  RETURN
ENDIF
IRC=0
T1=(B-A)*HALF
T2=(B+A)*HALF
S=ZERO
DO 1000 I=1,N
  X1=T1*X(I)+T2
  CF=COS(X1)
  RT=SQRT(Z*Z+R*R-2.0*R*R0*CF+R0*R0)
  AL=R-R0*CF+RT
  IF(AL.LE.0.0) AL=1.0E-14
  FRX=CF*(RT+R0*CF*LOG(AL))
  V=FRX*W(I)
  S=S+V
1000 CONTINUE
S=S*T1
RETURN
END

```

Fig. 4.3 Subroutine GAUSSX

```

C
C      SUBROUTINE GAUSSY(A,B,X,W,N,S,IRC)
C
C      IMPLICIT REAL*8(A-H,O-Z)
COMMON /LABS/R0,R,Z
DIMENSION X(1),W(1)
PARAMETER(ZERO=0.0D0)
PARAMETER(HALF=0.5D0)
C
IF(N.LT.1.OR.A.GE.B) THEN
  IRC=100
  RETURN
ENDIF
IRC=0
T1=(B-A)*HALF
T2=(B+A)*HALF
S=ZERO
DO 1000 I=1,N
  X1=T1*X(I)+T2
  CF=COS(X1)
  SF=SIN(X1)
  RT=SQRT(Z*Z+R*R-2.0*R*R0*CF+R0*R0)
  AL=R-R0*CF+RT
  IF(AL.LE.0.0) AL=1.0E-14
  FRY=SF*(RT+R0*CF*LOG(AL))
  V=FRY*W(I)
  S=S+V
1000 CONTINUE
S=S*T1
RETURN
END

```

Fig. 4.4 Subroutine GAUSSY

```

OPTIONS XOPT(NOAMOVE)
C
C      SUBROUTINE GAUSSZ(A,B,X,W,N,S,IRC)
C
C      IMPLICIT REAL*8(A-H,O-Z)
*INCLUDE *PAI*
COMMON /LABS/R0,R,Z
DIMENSION X(1),W(1)
PARAMETER(ZERO=0.0D0)
PARAMETER(HALF=0.5D0)
C
IF(N.LT.1.OR.A.GE.B) THEN
  IRC=100
  RETURN
ENDIF
IRC=0
T1=(B-A)*HALF
T2=(B+A)*HALF
S=ZERO
DO 1000 I=1,N
  X1=T1*X(I)+T2
  ZZ=ABS(Z)
  IF(ZZ.LE.1.E-5) GO TO 20
  CF=COS(X1)
  SF=SIN(X1)
  RT=SQRT(Z*Z+R*R-2.0*R*R0*CF+R0*R0)
  AL=(RT-ZZ)/(RT+ZZ)
  IF(AL.LE.0.0) AL=1.0E-14
  BL=R-R0*CF+RT
  AT=R0*SF*RT
  IF(ABS(AT).LE.1.E-5) GO TO 40
  GO TO 30
20  FRZ=0.0
  GO TO 50
30  FRZ=Z*( LOG(BL)+R0/ZZ*(0.5D0*CF*LOG(AL)-SF*ATAN(ZZ*(R-R0*CF)/AT)))
  GO TO 50
40  FRZ=Z*( LOG(BL)+R0/ZZ*(0.5D0*CF*LOG(AL)-SF*0.5D0*PI) )
50  CONTINUE
  V=FRZ*W(I)
  S=S+V
1000 CONTINUE
S=S*T1
RETURN
END

```

Fig. 4.5 Subroutine GAUSSZ

```

SUBROUTINE CLEAR
COMMON /ABSTBL/XABS(32),WABS(32)
.
.
.
IGAUSS=24
.
.
.
C***** ABS CALCULATION ROUTINE BY N. ISOBE 91/02/04 ***
CALL GASLEG(IGAUSS,XABS,WABS,IRC)
*****
.
.
.
RETURN
END

```

Fig. 4.6 Modified subroutine CLEAR

```

SUBROUTINE OPTI(CARD)
COMMON /ABSTBL/XABS(32),WABS(32)
.
.
.
READ(WCH(1:LCH) , *) IGAUSS
.
.
.
30 CONTINUE
C***** ABS CALCULATION ROUTINE BY N. ISOBE 91/02/04 ***
CALL GASLEG(IGAUSS,XABS,WABS,IRC)
*****
.
.
.
RETURN
END

```

Fig. 4.7 Modified subroutine OPTI

```
SUBROUTINE SUMC (X, Y, Z, FS, FE, R1, R2, ZS, ZE, CD, BXX, BYY, BZZ)
```

```
IFR=1  
BX= CD*SUM(IFR, F1, F2, R0, R1, R2, Z1, Z2)
```

```
IFR=2  
BY= CD*SUM(IFR, F1, F2, R0, R1, R2, Z1, Z2)
```

```
IFR=3  
BZ=-CD*SUM(IFR, F1, F2, R0, R1, R2, Z1, Z2)
```

```
RETURN  
END
```

```
REAL FUNCTION SUM*8 (IFUN, F1, F2, R0, R1, R2, Z1, Z2)
```

```
SUM= FUNC (F1, F2, R0, R2, Z2, IFUN)  
* -FUNC (F1, F2, R0, R1, Z2, IFUN)  
* -FUNC (F1, F2, R0, R2, Z1, IFUN)  
* +FUNC (F1, F2, R0, R1, Z1, IFUN)
```

```
RETURN  
END
```

```
REAL FUNCTION FUNC*8 (F1, F2, R0, R, Z, IFON)
```

```
10 S =GAUSS(IFON, F1, ZERO)  
FUNC =GAUSS(IFON, ZERO, F2)+S
```

```
20 FUNC =GAUSS(IFON, F1, F2)
```

```
RETURN  
END
```

```
REAL FUNCTION GAUSS*8 (IFUN, A, B)
```

```
IF (IFUN.EQ.1) CALL GAUSSX(A, B, XABS, WABS, IGAUSS, GAUSSV, IRC1)  
IF (IFUN.EQ.2) CALL GAUSSY(A, B, XABS, WABS, IGAUSS, GAUSSV, IRC1)  
IF (IFUN.EQ.3) CALL GAUSSZ(A, B, XABS, WABS, IGAUSS, GAUSSV, IRC1)  
GAUSS=GAUSSV  
RETURN  
END
```

Fig. 4.8 Calling procedure of the Gauss-Legendre subroutines
in vectorized version

Fig. 4.9 Modified subroutine SUMC

```

*****
C
C   DATE : FEB/1977
*****
CTY   FUNCTION  SUM  (FUN,F1,F2,RO,R1,R2,Z1,Z2)
CCC  OREAL FUNCTION  SUM*B (FUN,F1,F2,RO,R1,R2,Z1,Z2)
      OREAL FUNCTION  SUM*B (IFUN,F1,F2,RO,R1,R2,Z1,Z2)
      IMPLICIT REAL*B (A-H,O-Z)
CCC  OEXTERNAL FUN
CTY   S= FUNC (F1,F2,RO,R2,Z2,FUN)
CTY   S=-FUNC (F1,F2,RO,R1,Z2,FUN)+S
CTY   S=-FUNC (F1,F2,RO,R2,Z1,FUN)+S
CTY   S= FUNC (F1,F2,RO,R1,Z1,FUN)+S
CTY
CTY   SUM=S
CCC  OSUM= FUNC (F1,F2,RO,R2,Z2,FUN)
CCC  *     -FUNC (F1,F2,RO,R1,Z2,FUN)
CCC  *     -FUNC (F1,F2,RO,R2,Z1,FUN)
CCC  *     +FUNC (F1,F2,RO,R1,Z1,FUN)
C
OSUM= FUNC (F1,F2,RO,R2,Z2,IFUN)
*     -FUNC (F1,F2,RO,R1,Z2,IFUN)
*     -FUNC (F1,F2,RO,R2,Z1,IFUN)
*     +FUNC (F1,F2,RO,R1,Z1,IFUN)
C
      RETURN
      END

```

Fig. 4.10 Modified subroutine SUM

```

*****
C
C   DATE      : JAN/27/1977
*****
CTY   FUNCTION  FUNC (F1,F2,RO,R,Z,FON)
CCC  OREAL FUNCTION  FUNC*B (F1,F2,RO,R,Z,FON)
      OREAL FUNCTION  FUNC*B (F1,F2,RO,R,Z,IFON)
      IMPLICIT REAL*B (A-H,O-Z)
      COMMON /LABS/RRO,RR,ZZ
CCCC OEXTERNAL FON
ODATA ZERO / 0.0D /
      RRO=RO
      ZZ =Z
      RR =R
      IF(ABS(F1).LE.5.E-5) F1=0.
      IF(ABS(F2).LE.5.E-5) F2=0.
      IF(F1*F2) 10,20,20
CTY10 S    =GAUSS(FON,F1,0.)
CTY   FUNC =GAUSS(FON,0.,F2)+S
CCC100S  =GAUSS(FON,F1,ZERO)
CCC  OFUNC =GAUSS(FON,ZERO,F2)+S
100S  =GAUSS(IFON,F1,ZERO)
OFUNC =GAUSS(IFON,ZERO,F2)+S
      RETURN
CCC20 FUNC =GAUSS(FON,F1,F2)
20 FUNC =GAUSS(IFON,F1,F2)
      RETURN
      END

```

Fig. 4.11 Modified subroutine FUNC

```

*****
C   FUNCTION : GAUSS INTEGRATION
C   PARAMETER(S) :
C     NAME      TYPE    I/O
C     F          R*8     I      FUNCTION TO BE INTEGRATE
C     A          R*8     I      START
C     B          R*8     I      END
C   DATE      : JAN/7/1986
C   AUTHER    : Y.OHGANE (K.C.S)
*****
CTY   FUNCTION  GAUSS (F,A,B)
CCC   OREAL FUNCTION  GAUSS*8 (F,A,B)
      OREAL FUNCTION  GAUSS*8 (IFUN,A,B)
      IMPLICIT REAL*8 (A-H,O-Z)
*INCLUDE *LABE*
COMMON /ABSTBL/ XABS(32),WABS(32)
C   DIMENSION X(300000),W(300000)
CCC   OEXTERNAL F
      OEXTERNAL FRX,FRY,FRZ
CTY   IF(IGAUSS.EQ.12) CALL GAS12D(A,B,F,GAUSS)
CTY   IF(IGAUSS.EQ.24) CALL GAS24D(A,B,F,GAUSS)
CTY   IF(IGAUSS.EQ.32) CALL GAS32D(A,B,F,GAUSS)
CCCC  OIF(IGAUSS.EQ.12) CALL GAS12D(A,B,F,GAUSSV)
CCCC  OIF(IGAUSS.EQ.24) CALL GAS24D(A,B,F,GAUSSV)
CCCC  OIF(IGAUSS.EQ.32) CALL GAS32D(A,B,F,GAUSSV)
CCCC  CALL GASLEG(IGAUSS,XX,WW,IRC)
      IF (IFUN.EQ.1) CALL GAUSSX(A,B,XABS,WABS,IGAUSS,GAUSSV,IRC1)
      IF (IFUN.EQ.2) CALL GAUSSY(A,B,XABS,WABS,IGAUSS,GAUSSV,IRC1)
      IF (IFUN.EQ.3) CALL GAUSSZ(A,B,XABS,WABS,IGAUSS,GAUSSV,IRC1)
OGAUSS=GAUSSV
RETURN
END

```

Fig. 4.12 Modified subroutine GAUSS

```

;-----;
; TEST OF FORCE ----- TEST4A <VER.6>
;-----;
; MODE : EOB1 MODE
;-----;
; WT : 891.0(MM)
; WR : 301.0(MM)
;-----;
TCOIL 1 W 891.0 301.0 ORIGIN 0.0 0.0 0.0 0.0 J 33.9
      ARC CENTER -7508.000 0.000 R 18008.000 A 0.00 13.60
      ARC CENTER 5454.071 3135.856 R 4672.000 A 13.60 90.00
      ARC CENTER 5454.071 3135.856 R 2874.000 A 90.00 180.00
      BAR S 2580.000 4933.856 E 2580.000 -4933.856
      ARC CENTER 5454.071 -4933.856 R 2874.000 A 180.00 270.00
      ARC CENTER 5454.071 -3135.856 R 4672.000 A 270.00 346.40
      ARC CENTER -7508.000 0.000 R 18008.000 A 346.40 360.00
END
TFCOPY 15 NO 1 1 DTH 22.5
;
PCOIL 1 W 590.0 1848.0 R 1755.0 Z 950.0 M -22.48
PCOIL 2 W 590.0 1848.0 R 1755.0 Z 2850.0 M -22.48
PCOIL 3 W 590.0 1848.0 R 1755.0 Z 4750.0 M -1.42
PCOIL 4 W 590.0 1848.0 R 1755.0 Z 6650.0 M -1.42
PCOIL 5 W 900.0 900.0 R 3900.0 Z 9000.0 M 13.14
PCOIL 6 W 500.0 1500.0 R 11500.0 Z 6000.0 M -9.94
PCOIL 7 W 500.0 900.0 R 11500.0 Z 3000.0 M -5.37
PCOIL 8 W 500.0 900.0 R 11500.0 Z -3000.0 M -5.37
PCOIL 9 W 500.0 1500.0 R 11500.0 Z -6000.0 M -9.94
PCOIL 10 W 900.0 900.0 R 3900.0 Z -9000.0 M 13.14
PCOIL 11 W 590.0 1848.0 R 1755.0 Z -6650.0 M -1.42
PCOIL 12 W 590.0 1848.0 R 1755.0 Z -4750.0 M -1.42
PCOIL 13 W 590.0 1848.0 R 1755.0 Z -2850.0 M -22.48
PCOIL 14 W 590.0 1848.0 R 1755.0 Z -950.0 M -22.48
PCOIL 15 W 2150.0 4773.0 R 6000.0 Z 0.0 M 28.00
;
AFORCE T 1 2 P 1 2 IPERI
ATCOIL NAR 4 NAW 3 NATH 28 NBR 4 NBW 3 NBD 10
APCOIL NPR 3 NPZ 3 NPTH 15 ANGS 45. ANGE 135.
AEND
;SHAPE A
;
TITL ' TEST OF FORCE ----- TEST4A <VER.6> '

```

Fig. 4.13 Sample input data

```

        TWC(6 4 8 0 4) CLASS(2)
//FORCE EXEC GOLM,LM='JXXXX.FORCE#B.V6VP'
//FT05FO01 DD *
REST
FINI N
/*
//FT06FO01 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=1370)
/*
/*
/* P R I N T   F I L E
//FT66FO01 DD DSN=J9127.@FORCE.FT66@T,DISP=(NEW,CATLG),
//           UNIT=TSSWK,
//           SPACE=(TRK,(20,5),RLSE)
//FT77FO01 DD DUMMY
//**FT77FO01 DD DSN=J3403.@FORCE.FT77,DISP=(NEW,CATLG),
//**           UNIT=TSSWK,
//**           SPACE=(TRK,(10,9),RLSE)
/*
/*
/* P R I N T   F I L E   O F   A C L O S S
//FT78FO01 DD DUMMY
//**FT78FO01 DD DSN=J3403.@FORCE.FT78,DISP=(NEW,CATLG),
//**           UNIT=TSSWK,
//**           SPACE=(TRK,(10,5),RLSE)
/*
/*
/* S A V E   F I L E
//FT07FO01 DD DUMMY
/*
/*
/* R E - S T A R T   F I L E
//FT10FO01 DD DSN=J9127.FORCE.DATA(SAMPLE),DISP=SHR
/*
/*
/* C O M M A N D   S T O C K   F I L E
//FT11FO01 DD DUMMY
/*
/*
/* F E M A S   S T A N D E R D   F I L E
//FT15FO01 DD DUMMY
//**FT15FO01 DD DSN=J3403.@FORCE.FT15,DISP=SHR
/*
/*
/* F I E L D   S T A N D E R D   F I L E
//FT55FO01 DD DUMMY
/*
/* A R G U S   W O R K   F I L E S
//FT96FO01 DD UNIT=WK10,SPACE=(TRK,(50,10)),DISP=(NEW,DELETE)
//FT97FO01 DD UNIT=WK10,SPACE=(TRK,(50,10)),DISP=(NEW,DELETE)
//FT98FO01 DD UNIT=WK10,SPACE=(3200,(200,1)),DISP=(NEW,DELETE),
//           DCB=(RECFM=F,BLKSIZE=3200,DSORG=DA,BUFNO=1)
//FT99FO01 DD UNIT=WK10,SPACE=(TRK,(100,10)),DISP=(NEW,DELETE)
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=137,BLKSIZE=1370)
//MPTHSGT DD DSN=SYS1.KPATNLIB,DISP=SHR
//GDFILE DD SYSOUT=E,OUTLIM=60000,DEST=LOCAL
*/
/*
/*

```

Fig. 4.14 JCL for execution

5. STREAM V2.6 のベクトル化

5.1 概要

本章では、原研が開発を進めている、高温工学試験研究炉の安全審査用基礎資料の評価解析に使用されている3次元熱流体解析コード‘STREAM’のバージョン・アップ版であるSTREAM V2.6 のベクトル化作業とその効果についての報告をする。

STREAMの旧バージョンは、以前に当研究所内でベクトル化された経緯がある⁽¹⁾。新バージョンは前回のベクトル化手法が一部採用されているため、今回はそのルーチンの更なるベクトル化が主な作業となった。

5.2 基礎方程式

STREAMが扱う式は、流体質量の保存、流体運動量の保存、流体一固体のエネルギーの保存、乱流エネルギーと乱流消失率の移流拡散、生成消滅、並びに拡散物質の保存を表している。但し、STREAM2.6 では圧縮性流体の計算は行えないので、非圧縮性流体のみについて述べる。

(1) 質量の保存式

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (5.1)$$

(2) 運動量の保存式 ($i = 1 \sim 3$)

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} \\ = - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \nu \frac{\partial u_i}{\partial x_j} - g \beta (T - T_0) \end{aligned} \quad (5.2)$$

(3) エネルギーの保存式

$$\frac{\partial \rho C_p T}{\partial t} + \frac{\partial u_i \rho C_p T}{\partial x_i} = \frac{\partial}{\partial x_i} K \frac{\partial T}{\partial x_i} + q \quad (5.3)$$

(4) 乱流エネルギー・乱流消失率の式 ($k - \varepsilon$ 方程式)

$$\frac{\partial k}{\partial t} + \frac{\partial u_i k}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\nu_t}{\sigma k} \frac{\partial k}{\partial x_i} \right) + G_s + G_T - \varepsilon \quad (5.4)$$

$$\begin{aligned} \frac{\partial \varepsilon}{\partial t} + \frac{\partial u_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\nu_t}{\sigma k} \frac{\partial \varepsilon}{\partial x_i} \right) \\ + C_1 \frac{\varepsilon}{k} (G_s + G_T) (1 + C_3 R_1) - C_2 \frac{\varepsilon^2}{k} \end{aligned} \quad (5.5)$$

(5) 拡散物質の保存式

$$\frac{\partial C}{\partial t} + \frac{\partial u_i C}{\partial x_i} = - \frac{\partial}{\partial x_i} D_m \frac{\partial C}{\partial x_i} + \dot{d} \quad (5.6)$$

(6) 状態方程式

$$\rho = \text{一定} \quad (5.7)$$

これらの式に現れる変数の意味は次の通りである。

x_i	位置座標	(m)
u_i	x_i 方向の流体速度	(m/s)
t	時間	(s)
ρ	流体か固体の密度	(kg/m ³)
p	流体の圧力	(N/m ²)
ν	動粘性係数	(m ² /s)
	($\nu_l + \nu_t$: 分子動粘性係数+渦動粘性係数)	
μ	粘性係数	(kg/m·s)
	($\mu_l + \mu_t$: 分子粘性係数+渦粘性係数)	
σ_{ij}	応力テンソル	
	$\left(= \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] - \left[p + \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] \right)$	
H	比エンタルピー	(J/kg)
g_i	加速度	(m/s ²)
β	体膨張率	(l/°C)
T	流体か固体の温度	(°C)
T_0	流体の基準温度	(°C)
C_p	定圧比熱	(J/kg·°C)
K	熱伝導率	(J/m·s·°C)
q	発熱量	(J/m ³ ·s)
k	乱流エネルギー	(m ² /s ²)
ε	乱流消失率	(m ² /s ³)
C	拡散物質の濃度	(l)
D_m	拡散係数	(m ² /s)
\dot{d}	拡散物質の発生項	(l/s)
R	ガス定数	(J/kg·°K)

5.3 境界条件

STREAMでは、境界について特別な指定を行わなければ自動的に次の条件が設定される。

- ① 流体と固体、流体と境界外が接する面は流体の流入流出は無い。
- ② 流体と固体、流体と境界外が接する面は滑り壁(Free Slip Wall)である。
- ③ 流体と境界外、固体と境界外が接する面は断熱壁(Adiabatic Wall)である。
流体と固体が接する面は熱伝導をする。
- ④ 流体と固体、流体と境界外が接する面は物質拡散が無い。

これらの条件は壁面でしばしば見られるもので、ユーザがこれらの条件を指定する手間を省く。特別の条件は、次にあげるFlux境界条件と非Flux境界条件である。

(1) Flux境界条件

境界面から流体が流入流出する場合、流体質量に伴って移動するものとして、質量、運動量、エネルギー、及び拡散物質があげられる。これは運動方程式やエネルギー方程式などの、次式の対流項（移流項）によるものである。

$$-\int \phi u_j n_j ds \quad (5.8)$$

ここで、 ϕ ; ρ , ρu_j , $\rho C_p t$, C

u_j ; 検査面（境界面）の流速

n_j ; 境界面に外向きに立てた法線ベクトルの方向余弦

S ; 積分領域

この式から明らかに、境界面を貫く流速成分 u_j が与えられれば、面を通り抜ける質量 ρ 、運動量 ρu_j 、エネルギー H ($\rho C_p t$)、拡散物質濃度 C を同時に指定する必要がある。ただし、非圧縮性流体の場合は質量密度 ρ の値は一定であるから、とりあげて指定する必要はなく、運動量、エネルギー、拡散物質濃度については指定が必要である。しかし、移流項の差分化には上流（風上）差分が用いられているため、流入の場合は境界を通過する運動量の値は領域外の値を指定する必要があるが、流出の場合は運動量とエネルギーが外部へ流れるため、特に指定する必要は無い。

(2) 非Flux境界条件

非Flux境界条件は、質量の流れにより運ばれるものではないため、質量保存式には直接は影響を与えないが、境界面を通して、運動量、エネルギー、拡散物質の保存式に影響を及ぼす。発熱や拡散物質の項に対してはソース項として組み入れられるようになっているので、ここでの説明は省略する。

5.4 ベクトル化

ベクトル化作業は以下の手順で行った。

- i) 動的挙動解析ツールの FORTUNE⁽²⁾、ANALYZER⁽³⁾ を用いてオリジナル版 STREAM 2.6 の動的解析を行い、計算コストの高いルーチンを選び出す。
- ii) 計算コストの高いルーチンについて、ベクトル化可能であるかどうか調べ、可能であるものについてベクトル化作業を進める。
- iii) オリジナル版とベクトル化版との計算結果を比較する。

動的挙動解析ツールからの情報を元にベクトル化の可能性について検討を行った結果、オリジナル版で最もコストの高い部分は M I C C G 法および I L U C R 法の作業ファイル入力部分であり、これは内部処理することで計算コストを下げるとともにベクトル計算向きとした。

連立一次方程式の解法 (M I C C G 法、I L U C R 法、S O R 法) 部分はデータの参照に再帰性があるために、このままではベクトル計算できない。従って hyper plane 方式によるリストを作成しベクトル化を可能とした。

その他FUNCTIONの文関数化、DOループの下位ルーチンへの引き込み、多重ループの一重化等の手法を用いてベクトル化を行い、計算結果はオリジナル版とベクトル化版で完全に一致した。

5.4.1 hyper plane方式⁽⁷⁾

MICCG法、ILUCR法、SOR法では、そのDOループ構造が再帰的であるためにベクトル化には向きである。本作業ではこれらのループを、hyper Plane方式によるリストベクトルを用いることによってベクトル計算向きとした。

hyper plane方式を利用した場合の利点として、「平均ベクトル長が長くできる」、「オリジナル版と基本的な演算順序が同じであるため計算結果は一致する」等があげられる。逆にリスト領域を使用するため、主記憶領域が増加するのが短所である。

—修正ルーチン—

CDECMP,	CLUBBB,	DECOMX,	DECOMY,	HYPLAN,
HYPLA2,	HYPLA3,	LDLTPX,	LDLTPY,	PROCES,
RDECMP,	RLUBBB			

5.4.2 DO文の書き下し

多重DOループで最内DOループのベクトル長が短い場合、ベクトル演算効率は低くなる。外側DOループのベクトル長が充分に長ければ、最内ループを書き下すことによって外側のDOループがベクトル化できるので効率的となる。Fig. 5.1 はその修正例である。オリジナル版では文番号110 に関するDOループのみベクトル化されるがベクトル長が短く、外側のDOループの計算コストが高いためにベクトル化効率が悪い。修正版では、このDOループを書き下すことによって文番号100 のIに関するベクトル化ができるため、ベクトル長の増大、計算密度の点で効率が良い。

—修正ルーチン—

COSOUU,	COSOWW,	PKSORC
---------	---------	--------

5.4.3 作業配列の導入

DOループ内にサブルーチンの呼び出しがある場合、そのDOループはベクトル化できない。この場合作業配列を導入し、下位ルーチンにDOループを引き込むことでベクトル化できる。Fig. 5.2 はその修正例である。

修正ルーチン

COSOUU,	COSOWW,	GAUSSE,	G I J CYL,	G I J SOC,
PKSCYL,	PKSORC,	TRIAUU,	TRIAWW	

5.4.4 外部手続きのオンライン展開

5.4.3と同じケースで、下位ルーチンが単純な構造の場合は上位ルーチンに展開してベクトル化を行う。Fig. 5.3 はこの例であり、下位のルーチンが変数に単純に代入するだけなので、書き下して展開する。尚、ここでは、上位ルーチンのDOループの引き込みをしてベクトル化している。

修正ルーチン

TRIAUU,	TRIAWW,	I S C T R M
---------	---------	-------------

展開したルーチン

TRI SEP,	S E T D I M
----------	-------------

5.4.5 ベクトル化不適命令

計算型GOTO文はベクトル化不適命令であり、このままではベクトル化できない。これはブロックIF文に書き直すことでベクトル化できる。Fig. 5.4 にその例を示す。

修正ルーチン

BDFORK,	O P N F L X,	C O R D I S,	C O R A R E,	C O R C E L
---------	--------------	--------------	--------------	-------------

5.4.6 多重ループの一重化

多重ループは、原則として最内ループしかベクトル化の対象にならない。そのため、最内ループのベクトル長が長い場合を除いてはベクトル化効率は悪くなる。そこで、あらかじめ多重ループに含まれるループの数だけリストベクトルを作成して一重化する。Fig. 5.5、Fig. 5.6にその例を示す。

修正前 Fig. 5.5 におけるDO 100のループは I, J, K の三重ループであるが（ベクトル長は各々 IM, JM, KM）、最内ループ中に GOTO 文があるので、GOTO 文で飛び出さない時の I, J, K の値をリストベクトル LI, LJ, LK を各自作成した。Fig. 5.6 以降のループでは L のみの一重ループとなり、ベクトル長 L は、

$$L = IM \times JM \times KM - (\text{GOTO 文で飛ぶ回数})$$

となり充分に長くなるため、ベクトル化効率は著しく向上する。

修正ルーチン

COSOEE, COSOKK, COSOTT, COSOUU, COSOWW, NOSHST

5.4.7 D O ループの分割

これまでにベクトル化したルーチンでも、IF文を多く含む等の理由からベクトル化の効率が悪い場合がある。このような場合はループの回転毎に同じようなIF文が実行されるため処理速度の向上はあまり望めない。そこで5.4.6の後、Lの1重ループ内に含まれるIFセンテンスの部分を1ヶのDOループにし、単純代入文はそれだけで1ヶのDOループに分割した。このように分割することによりDOループの個数が増えるので、オーバーヘッドが余計にかかることになるが、ベクトル長が充分に長いのでそのオーバーヘッドを差し引いても効果的である。Fig. 5.6 に例を示す。

修正ルーチン

COSOEE, COSOKK, COSOTT, COSOUU, COSOWW, NOSHST

5.4.8 F U N C T I O N の文関数化

プログラム内で使用されているFUNCTIONの中で、文関数化できるものを文関数化した。STREAMの旧バージョンでは、FUNCTIONを文関数化した経緯があり、STREAM V2.6ではそれが生かされている。今回は文関数化できる関数を文関数化することにより呼び出しオーバーヘッドの削減を計った。Fig. 5.7 に例を示す。

修正ルーチン

USNEWT

展開したFUNCTION

DFUNCU, FUNCUU, DISCEN

```

DO 100 K=1, KM
DO 100 J=1, JM
DO 100 I=2, IM
.....
COF (1, I, J, K) =CONX1 (L)+DIFX1 (L)
COF (2, I, J, K) =CONX2 (L)+DIFX2 (L)
COF (3, I, J, K) =CONY1 (L)+DIFY1 (L)
COF (4, I, J, K) =CONY2 (L)+DIFY2 (L)
COF (5, I, J, K) =CONZ1 (L)+DIFZ1 (L)
COF (6, I, J, K) =CONZ2 (L)+DIFZ2 (L)
COF (7, I, J, K) =(RTUU1+RTUU2)*VOL (L)

V      DO 110 L=1, 6
V      COF (7, I, J, K)=COF (7, I, J, K)+COF (L, I, J, K)
V 110 CONTINUE
.....
100 CONTINUE

DO 100 K=1, KM
DO 100 J=1, JM
V      DO 100 I=2, IM
.....
V      COF (1, I, J, K) =CONX1 (L)+DIFX1 (L)
V      COF (2, I, J, K) =CONX2 (L)+DIFX2 (L)
V      COF (3, I, J, K) =CONY1 (L)+DIFY1 (L)
V      COF (4, I, J, K) =CONY2 (L)+DIFY2 (L)
V      COF (5, I, J, K) =CONZ1 (L)+DIFZ1 (L)
V      COF (6, I, J, K) =CONZ2 (L)+DIFZ2 (L)
V      COF (7, I, J, K) =(RTUU1+RTUU2)*VOL (L)

V      COF (7, I, J, K)=COF (7, I, J, K)+COF (1, I, J, K)+COF (2, I, J, K)
&           +COF (3, I, J, K)+COF (4, I, J, K)+COF (5, I, J, K)+COF (6, I, J, K)
.....
V 100 CONTINUE

```

Fig. 5.1 Vectorization by expansion of the inner DO loop

```

SUBROUTINE XXXXXX (IM, JM, KM, IMP, JMP, KMP, MAT, MGEO
*           , X, Y, Z, OPNX, OPNY, OPNZ, OPNV, U, V, W, T, FS, TK, TE, EVS
*           , GSD, GTD, F, IOPC1, ISW)
*           .
*           .
DO 100 K=1, KM
DO 100 J=1, JM
DO 100 I=1, IM
*           .
*           .
CALL YYYYYY (IM, JM, KM, IMP, JMP, KMP, X, Y, Z, OPNX, OPNY, OPNZ, OPNV
*           , T, EVS, I, J, K, GK, G11, G22, G33, G12, G23, G31)
*           .
*           .
100 CONTINUE
RETURN
END

SUBROUTINE YYYYYY (IM, JM, KM, IMP, JMP, KMP, X, Y, Z, OPNX, OPNY, OPNZ, OPNV
*           , T, EVS, I, J, K, GK, G11, G22, G33, G12, G23, G31)
*           .
*           .
SGT=0.9
IF (IGRV.EQ.0) THEN
  TH=0.5*(Y(J)+Y(J+1))
  GRX=FUNGR1 (GX, GY, TH)
  GRY=FUNGR2 (GX, GY, TH)
*           .
*           .
RETURN
END

```

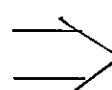


Fig. 5.2 Usage of the working area (continued)

```

SUBROUTINE XXXXXX (IM, JM, KM, IMP, JMP, KMP, MAT, MGEO
*           , X, Y, Z, OPNX, OPNY, OPNZ, OPNV, U, V, W, T, FS, TK, TE, EVS
*           , GSD, GTD, F, IOPC1, ISW
*.....*
*.....*
CALL YYYYYY (IM, JM, KM, IMP, JMP, KMP, X, Y, Z, OPNX, OPNY, OPNZ, OPNV
*           , T, EVS, I, J, K
&           , GK, G11, G22, G33, G12, G23, G31
&           , LI, LJ, LK, LMAX)
*.....*
RETURN
END

SUBROUTINE YYYYYY (IM, JM, KM, IMP, JMP, KMP, X, Y, Z, OPNX, OPNY, OPNZ, OPNV
&           , T, EVS, I, J, K, GK, G11, G22, G33, G12, G23, G31
&           , LI, LJ, LK, LMAX)
*.....*
V   DO 1000 L=1, LMAX
V     I=LI (L)
V     J=LJ (L)
V     K=LK (L)
V
V     SGT=0.9
V     IF (IGRV.EQ.0) THEN
V       TH=0.5*(Y(J)+Y(J+1))
V       GRX=FUNGR1 (GX, GY, TH)
V       GRY=FUNGR2 (GX, GY, TH)
V
V 1000 CONTINUE

```

Fig. 5.2 Usage of the working area

```
SUBROUTINE XXXXXX (IM, JM, KM, MMM, OOO, I, J, K, XXXX)
.....
```

```
IF (OPNX (I-1, J, K) .LE. 0.0) THEN
  CALL YYYYYY (MMM (I-1, J, K), M1A, M1B)
  IF (M1A. EQ. 11) THEN
    XXXX=AA
.....
```

```
SUBROUTINE YYYYYY (MMM, M1A, M1B)
M1A=MOD (MMM, 100)
M1B=(MMM-M1A)/100
RETURN
END
```



```
SUBROUTINE XXXXXX (IM, JM, KM, MMM, OOO, I, J, K, XXXX)
.....
```

```
V DO 100 L=1, LMAX
V IF (OPNX (I-1, J, K) .LE. 0.0) THEN
V   M1A=MOD (MMM (I-1, J, K), 100)
V   M1B=(MMM (I-1, J, K)-M1A)/100
V   IF (M1A. EQ. 11) THEN
V     XXXX (L)=0.0
.....
```

```
V 100 CONTINUE
```

Fig. 5.3 Vectorization by expansion of the slave routine
into the DO loop

```
DO 110 K=K1,K2
DO 110 J=J1,J2
DO 110 I=I1,I2
.....
GO TO (1,2,3),MENF
1 CONTINUE
.....
GO TO 100
2 CONTINUE
.....
GO TO 100
3 CONTINUE
.....
100 CONTINUE
110 CONTINUE

DO 110 K=K1,K2
DO 110 J=J1,J2
V DO 110 I=I1,I2
.....
V IF ((MENF.LE.1).OR.(MENF.GT.3)) THEN
.....
V ELSEIF (MENF.EQ.2) THEN
.....
V ELSE
.....
V 110 CONTINUE
```

Fig. 5.4 Modified loop of the computed GOTO statement

```

DO 100 K=1,KM
C
DZ2=Z(K+1)-Z(K)
IF(K.NE.1) THEN
  DZ1=Z(K)-Z(K-1)
ELSE
  DZ1=1.0E+20
ENDIF
IF(K.NE.KM) THEN
  DZ3=Z(K+2)-Z(K+1)
ELSE
  DZ3=1.0E+20
ENDIF
C
DO 100 J=1,JM
C
DY2=Y(J+1)-Y(J)
IF(J.NE.1) THEN
  DY1=Y(J)-Y(J-1)
ELSE
  DY1=1.0E+20
ENDIF
IF(J.NE.JM) THEN
  DY3=Y(J+2)-Y(J+1)
ELSE
  DY3=1.0E+20
ENDIF
IF(COC.GT.0.0.AND.J.EQ.1) DY1=Y(JM+1)-Y(JM)
IF(COC.GT.0.0.AND.J.EQ.JM) DY3=Y(2)-Y(1)
C
DO 100 I=1,IM
C
IF(OPNV(I,J,K).EQ.0.0)          GO TO 10
IF(ISWSD.GE.1.AND.CKK.NE.1.E+20
* .AND.FS(I,J,K).GT.0.0) GO TO 10
IF(ISWSD.GE.1.AND.CKK.EQ.1.E+20
* .AND.FS(I,J,K).GE.TOFS) GO TO 10
C
DX2=X(I+1)-X(I)
IF(I.NE.1) THEN
  DX1=X(I)-X(I-1)
ELSE
  .....
C
GO TO 100
C ...
10 CONTINUE
COF(1,I,J,K)=0.0
COF(2,I,J,K)=0.0
COF(3,I,J,K)=0.0
COF(4,I,J,K)=0.0
COF(5,I,J,K)=0.0
COF(6,I,J,K)=0.0
COF(7,I,J,K)=0.0
SOC(I,J,K)=0.0
C ...
100 CONTINUE

```



Fig. 5.5 Original loop structure in the subroutine

(continued on Fig. 5.6)

```

      DO 1000 K=1,KM
      DO 1000 J=1,JM
V     DO 1000 I=1,IM

V     IF((OPNV(I,J,K).EQ.0.0) .OR.
&       (ISWSD.GE.1 .AND. CKK.NE.1.E+20
&        .AND. FS(I,J,K).GT.0.0 ) .OR.
&       (ISWSD.GE.1 .AND. CKK.EQ.1.E+20
&        .AND. FS(I,J,K).GE.TOFS)) THEN
V       COF(1,I,J,K)=0.0
V       COF(2,I,J,K)=0.0
V       COF(3,I,J,K)=0.0
V       COF(4,I,J,K)=0.0
V       COF(5,I,J,K)=0.0
V       COF(6,I,J,K)=0.0
V       COF(7,I,J,K)=0.0
V       SOC(I,J,K) =0.0
V     ELSE
V       L=L+1
V       LI(L)=I
V       LJ(L)=J
V       LK(L)=K
V     ENDIF
V1000 CONTINUE
      LMAX=L

V     DO 1010 L=1,LMAX
V       I=LI(L)
V       J=LJ(L)
V       K=LK(L)

V       DZ2(L)=Z(K+1)-Z(K)
V       IF(K.NE.1) THEN
V         DZ1(L)=Z(K)-Z(K-1)
V       ELSE
V         DZ1(L)=1.0E+20
V       ENDIF
V       IF(K.NE.KM) THEN
V         DZ3(L)=Z(K+2)-Z(K+1)
      .....

```

Fig. 5.6 Modified loop structure in the subroutine

```
SUBROUTINE XXXXXX(U, Y, MU, US)
.....
F30 =FUNCUU(Y, MU, US30, K, A) -U
.....
```

```
FUNCTION FUNCUU(Y, MU, US)
COMMON/LGLW/K, A
REAL K
REAL MU
FUNCUU=1.0/K* ALOG (US*Y/MU)+A
FUNCUU=FUNCUU*US
RETURN
END
```



```
SUBROUTINE XXXXXX(U, Y, MU, US)
.....
```

```
*INCLUDE FUNCUU
FUNCUU(AAAY, AAMU, AAUS, AAAK, AAAA)
= (1.0/AAAK*LOG (AAUS*AAAY/AAMU) + AAAA)*AAUS
.....
```

```
F30 =FUNCUU(Y, MU, US30, K, A) -U
.....
```

Fig. 5.7 Vectorization by using the statement FUNCTION definition

5.5 I/O回数の削減

オリジナル版では機番11(FT11F001)に一時データセットを定義して作業用ファイルとして使用している。このファイルは、前ステップの計算結果を保存して置くための中間ファイルであり、実行終了後は削除されるが、I/O回数が多くなるため、ジョブ実行時に経過時間が多くなるという欠点が存在する。

FACOM VP-2600にはこのような中間ワークファイルに対してVIO/F入出力機能⁽¹¹⁾が備わっており、この機能を利用するとI/O回数を大幅に減らすことができる。

Table 5.1にその効果を示す。ここで、

I/O : I/O回数(回)

CPU : CPU時間(秒)

Memory : 主記憶領域(KB)

Brgn : 基本域使用量 Ergn : 拡張域使用量

Elapsed : JOBの実行から終了までの経過時間

である。Table 5.1は、比較的小さいデータを用い、作業用ファイルを使用した場合、およびVIO/F使用した場合の実行結果である。経過時間は他のジョブの混み具合によって影響が出るが、今回のテスト時は双方とも同等の環境であった。

本機能の適用によってI/O回数を大幅に減らした他、経過時間を短縮することができた。

Table 5.1 Comparison of I/O Counts and Elapsed Time

	I/O (Counts)	CPU (s)	Memory (KB) Brgn	Ergn	Elapsed (s)
Original Vector (Work)	1300	6.66	296	10980	49.76
Original Vector (VIO/F)	46	6.39	312	15268	18.69

5. 6 実行

今回使用したデータは非圧縮性流体を直角座標系で計算する 5 ケースである。

5. 6. 1 入力データ

Table 5. 2 にデータの内容を示す。

X, Y, Z 方向の領域区分	IM×JM×KM
計算終了サイクル	C Y C L
方程式の選択	E Q U A
U	X 方向流速方程式計算
V	Y 方向流速方程式計算
W	Z 方向流速方程式計算
P	圧力計算
T	温度計算
k	乱流計算
E	乱流消失率計算
C	拡散物質の濃度計算

※ これらの方程式を解く場合は解法を示した。

Table 5. 2 Contents of Solver Input Data

DATA CASE	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5
IM×JM×KM	50×13×10	50×13×10	50×13×10	28×13×10	120×50× 1
C Y C L	100	100	20	20	50
U	JACOBI 法	ILUCR 法	ILUCR 法	---	JACOBI 法
V	JACOBI 法	ILUCR 法	ILUCR 法	---	JACOBI 法
W	JACOBI 法	ILUCR 法	ILUCR 法	---	---
P	MICCG 法	MICCG 法	MICCG 法	---	MICCG 法
E Q U A	T	ILUCR 法	ILUCR 法	ILUCR 法	---
	k	ILUCR 法	ILUCR 法	---	---
	E	ILUCR 法	ILUCR 法	---	---
	C	ILUCR 法	ILUCR 法	---	---

5.6.2 ベクトル化効果と考察

STREAMは単精度計算の場合、入力データで三角柱要素の配置指定をしたりすると収束しにくくなり、結果としてCPU 時間とI/O が増大する恐れがある。そのため、今回のベクトル化では全ての実数変数を倍精度化して計算した。Table 5.3 は5.6.1 の入力データを用い、FORTRAN77VP V10 及び、FORTRAN77EX/VP V12で翻訳し実行した結果である。CASE4 のようなベクトル長の短いものに関しては効果小だったが、CASE1,2 のような場合は対オリジナルベクタとの比較では最大5.37倍を示した。

メモリ使用割合はオリジナルベクタに対し、VIO/P を使用した状態で約70%ほど増加したが、原研のclass6のような拡張域200MB 使用可能なクラスにおいて約150,000 メッシュの計算が可能である。class9のような512MB まで使用可能なクラスにおいて、約400,000 メッシュの計算が可能となる。しかし、実際にはSTREAMを用いて研究をする場合、512MB まで使用するクラスは夜間の許可ジョブクラスであり、あまり実用的ではない。そこでメモリ使用量を縮小する手段として、

- ①VIO/P の代わりに一時データセットを用いる。
- ②計算精度を要求しない変数に関して4byte 実数型変数を用いる。

の2通りが考えられる。

①はメッシュ数が大きくなった場合、I/O の増大による経過時間の増加を招くが、メモリ節約には効果がある。次に②について説明すると、冒頭で述べた理由から、今回のテストランでは、ジョブ制御文上であらかじめDOUBLE指定して翻訳してある。よって、実数型変数に関してのメモリは全て2倍割り当てられている計算となる。そこで②を施すことは、倍精度不要な実数型変数に関してはメモリを半分に縮小できるため、精度を要求する変数が少ないコードの場合には有効である。しかし、実際にこれを施すには計算精度の必要な変数を抽出し、それらをソースコード上で陽に8byte 実数型で宣言してやる必要がある。

Table. 5. 3 Vectrization Effect for STREAM

INPUT DATA STREAM V2. 6	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	Compiler Version and Compaile Option etc.
Original Scalar	1' 50. 04"	1' 58. 21"	50. 54"	19. 92"	31. 51"	FORTRAN77 Ver. 10
Original Vector (VU time)	2' 06. 03" (58. 03")	2' 12. 69" (58. 33")	44. 33" (13. 70")	20. 05" (7. 97")	23. 26" (0. 78")	AE, OPTIMIZE(3), ALC, DOUBLE, XOPT(NOAMOVE)
Original VecV12 (VU time)	2' 01. 82" (52. 16")	2' 09. 45" (52. 41")	49. 65" (11. 10")	23. 42" (6. 16")	25. 06" (0. 18")	FORTRAN77/VP Ver. 10
Tuning VectorV10 * (VU time)	24. 49" (7. 14")	26. 10" (7. 80")	16. 33" (3. 67")	9. 35" (2. 40")	14. 54" (0. 60")	FORTRAN77EX/VP Ver. 12
OrgSc / TuV10	4. 50	4. 53	3. 10	2. 13	2. 16	AE, ALIGNC, STATIS,
OrgV10 / TuV10	5. 15	5. 08	2. 72	2. 14	1. 59	DOUBLE,
OrgV12 / TuV10	4. 97	4. 96	3. 04	2. 51	1. 72	XOPT(NOPREEX, NOEVAL)
Tuning VectorV12 * (VU time)	23. 49" (6. 47")	25. 20" (7. 16")	18. 75" (2. 78")	11. 81" (1. 40")	13. 44" (0. 54")	Original Vector V10, OrgV12 ; Original Vector V10, OrgV12 ; Tuning Vector V12
OrgSc / TuV12	4. 69	4. 69	2. 70	1. 69	2. 34	TUV10 ; Tuning Vector V10, TUV12 ; Tuning Vector V12
OrgV10 / TuV12	5. 37	5. 27	2. 36	1. 70	1. 73	
OrgV12 / TuV12	5. 19	5. 14	2. 65	1. 98	1. 87	

6. 自由電子レーザーコードのベクトル化と整備

6.1 コードと作業の概要

日本原子力研究所物理部において、開発中である自由電子レーザーの設計に資するため大型計算機用1次元計算コードが開発されている。

本コードは、自由電子レーザーの発振過程をシミュレーションするコードであり、基礎方程式は、レーザーの平板型アンジュレータ中の電磁波增幅の1次元理論より導出される電子の運動方程式と光子場のMaxwell 方程式の2つである^(14, 15)。これらの方程式は、単純に差分化され数値積分が行われる^(11, 12)。求める物理量は、電子パルスがアンジュレータ中を往復する間のレーザー場の時間及び空間変化、光学利得の時間変化、出力光のスペクトル分布である。コードは、物理量を計算する部分とその量を図形出力するための2つにわけられる。

現在プログラムの大枠は出来ており、レーザー光がミラー間を往復する回数が小さい場合は計算可能である。しかし、実際のレーザー発振をシミュレーションするには、往復回数を数百回にまで増やす必要がある。そこで、長大計算を高速にしかも効率良く行う必要性から、ベクトル化と長大計算のためのデータの入出力機能の整備を行った。但し、整備についてはコードがまだ完成していないため、方法についてのみ記す。

6.2 ベクトル化

6.2.1 コードの構成とコスト分析

ここでは、物理量を計算する部分の解析結果についてのみ述べる。コードの構成をFig. 6.1に、FORTUNE によるコスト解析結果をTable 6.1 に示す⁽²⁾。

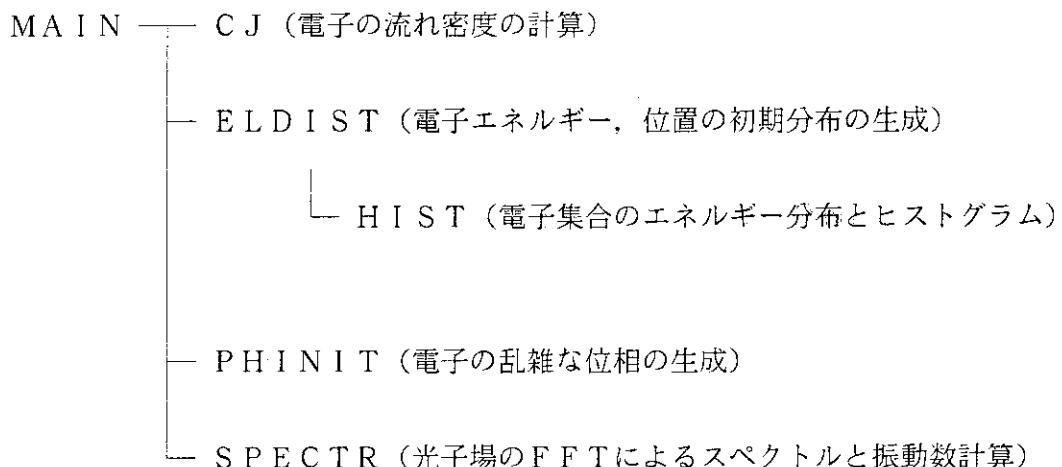


Fig. 6.1 Tree Structure and contents of subroutines

Table 6.1 Time cost distribution of each subroutine
by using FORTUNE

サブルーチン名	実行回数	コスト比率(%)
MAIN	1	96.2
CJ	1142070	2.9
HIST	60012	0.6
ELDISI	6	0.2
SPECTR	1	0.2
PHINIT	6	—

6.2.2 ベクトル化方法

Table 6.1 からコストはMAINルーチンにのみ偏っていることがわかる。従って、MAINルーチンのみベクトル化の対象とする。次にFORTUNE によるMAINルーチンの各DOループのコスト解析の結果をTable 6.2 に示す。

Table 6.2 Time cost distribution of each do loop in MAIN routine
by using FORTUNE

ループNo.	ベクトル長	コスト(%)
3050	10000	74.19%
3060	4	18.30%
3030	32768	6.54%

Table 6.2 より 3050、3060、3030 の DO ループのコストが高いことが分かる。ここで、1重 DO ループ 3030 は、十分ベクトル化されており、ベクトル化の対象となる DO ループは、3060、3050 からなる 2重 DO ループである。この部分のオリジナル版のコンパイルリストを Fig. 6.2 に示す。

この 2重ループはレーザー発振に寄与する全ての電子の、エネルギー、位相、及び各電子位置での光子場の値を計算する部分である。外側の DO ループは番号づけられた各電子に関するループであり、内側の DO ループは、各電子が相互作用する光子場の位置に関するループである。Fig. 6.2 に見るようく、内側の DO ループは既にベクトル化されているが、外側の DO ループは、ベクトル処理とスカラ処理の混合演算になっている。この原因は、内側ループのインデックスの始点と終点の値 (MM, MP) が外側の DO ループで計算されており、外側のループインデックスに依存する変数になっているためである。

そこで、Fig. 6.3 のように j 番目の電子の相互作用する最低、最大の場の点 (MM, MP) を各電子の番号 j に関する配列とし、2重 DO ループを配列化した光子場の点を計算する部分と電子と相互作用する光子場を計算する部分の 2つに DO ループを分割した。分割した 2つのループは双方ともベクトル化された。

```

M      DO 3050 J = 1 ,NELEC2
V          M=IDINT((ZETA(J)-OSHIFT+ZOPTF)/DZOPT)+1
M          MP=IDINT((ZETA(J)-OSHIFT+ZOPTF-XLHF)/DZOPT)+1
M          MM=IDINT((ZETA(J)-OSHIFT+ZOPTF+XLHF)/DZOPT)+2
-----
V      DO 3060 N = MM, MP
V          SUMSIN(N)=SUMSIN(N)+DSIN(PHASE)
V          SUMCOS(N)=SUMCOS(N)+DCOS(PHASE)
V      3060  CONTINUE
-----
V      3050 CONTINUE

```

Fig. 6.2 Original loop structures and compile information

```

V      DO 3051 J = 1 ,NELEC2
V      M=IDINT((ZETA(J)-OSHIFT+ZOPTF)/DZOPT)+1
V      MP(J)=IDINT((ZETA(J)-OSHIFT+ZOPTF-XLHF)/DZOPT)+1
V      MM(J)=IDINT((ZETA(J)-OSHIFT+ZOPTF+XLHF)/DZOPT)+2
-----
V      3051 CONTINUE
C
V      DO 3052 J=1 ,NELEC2
V      DO 3060 N = MM(J),MP(J)
V          SUMSIN(N)=SUMSIN(N)+DSIN(PHASE)
V          SUMCOS(N)=SUMCOS(N)+DCOS(PHASE)
V      3060    CONTINUE
V      3052    CONTINUE

```

Fig. 6.3 Improvement on loop structure and compile information

6.2.3 結果

以上の改良により、得られる計算時間とオリジナルコードの計算時間の比較をTable 6.3 に示す。

ベクトル化版ベクトルモードとオリジナル版スカラモードの速度比は1.88、ベクトル化版スカラモードとは1.24である。ベクトル化版のベクトル化率は71%である。また、いずれも計算結果は全て一致した。

Table 6.3 The comparison of CPU time

	オリジナル版 スカラモード	オリジナル版 ベクトルモード	ベクトル化版 スカラモード	ベクトル化版 ベクトルモード
C P U	6' 28	4' 16	8' 02	3' 35
V U	-----	1' 19	-----	1' 17

6.3 リスタート機能の整備

6.3.1 目的

現実のレーザー発振をシミュレートするためには、電子のミラー間往復回数を数百回にまで増やさなくてはいけない（現時点ではまだ、発振のシミュレートは行なわれておらず、必要な回数は正確にはわかっていないが、数時間の計算が必要である）。従って、レーザー発振に至るまでリスタート方式で実行できるようにする必要がある。

6.3.2 方法

ここではリスタート方式を採用する。ファイル入出力の混乱を避けるため初期条件を埋め込んだ初期計算のみ行うコードと、リスタート計算を行うコードの2つを作る。前者は必要な物理定数のみ入力して計算を行い、ある往復回数までの計算結果を出力ファイルに格納するものとする。後者は格納されたデータファイルからデータを読み出して、そこから与えた往復回数までの物理量を計算し、結果をリスタートファイルに格納するものとする。これらの処理の流れをFig. 6.4に示す。いずれのコードもMAINルーチンで計算された電子と光子場に関する物理量をある反復回数の後にリスタートファイルに書き出す様にし、リスタートコードはMAINルーチンの最初でリスタートファイルの物理量を配列に読み込む様にした。

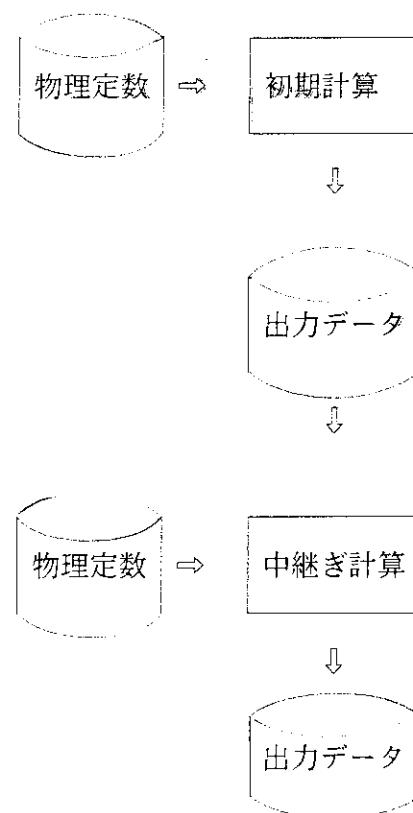


Fig. 6.4 Restart procedure for the long time run

6.4 まとめ

1次元自由電子レーザーコードのベクトル化と整備を行った。ベクトル化については、最も計算時間のかかる部分が電子数についてのループであり、電子どうしの相互作用を無視しているため本質的に各々の電子に対する並列計算が可能である。従って、今回のような方法は有効である。また、基本的なアルゴズムの変更により高速化される可能性もあり今後のコードの改良が望まれる。

7. HEATING7-VPの改良

7.1 概要

3次元熱解析コード HEATING7-VP^(16, 17) の改良を行った。本コードでは構造物を幾つかのリージョン（メッシュ）で区切り、各点での熱の流れをシミュレーションする。旧バージョンは、2次元解析では四角形、及び三角形、3次元解析では直方体、及び三角柱のリージョンに区切ることが可能であった^(18, 19)。しかし、三角形（三角柱）リージョンの境界条件は、面対面境界条件しか取り扱えなかった。そのため、物性値における回りの影響を考慮した面対面（材料の接続）の計算は不可能であった。今回の改良の目的は、三角形、及び三角柱リージョンにも面対面の計算を可能とすることである。本報告書では、この改良の基礎となる事柄、改良したサブルーチン、及びテスト計算の結果について述べる。

7.2 物性領域定義の改良について

7.2.1 リージョン定義の改良

本コードは解析対象物をリージョンと呼ばれる小さな領域で分割して、シミュレーションを行う。解析の対象となる領域は、2次元解析の場合は四角形、3次元解析の場合は直方体メッシュに格子分割される。これらの格子は複数の物性が扱えるようにいくつかにまとめられている。これらの格子をリージョンと呼ぶ。

旧バージョンではFig. 7.1、およびFig. 7.2のように境界部分以外では、2次元解析のとき三角形リージョン、3次元解析のとき三角柱リージョンを定義することはできない。面対面の計算は四角形（直方体）リージョンのみしか定義できない。理由は小格子内でリージョンを2つに区切ることが不可能であったからである。

今回の改良により、2つのリージョンが定義できるようになり、三角形（三角柱）リージョンの面対面の計算を可能にした。

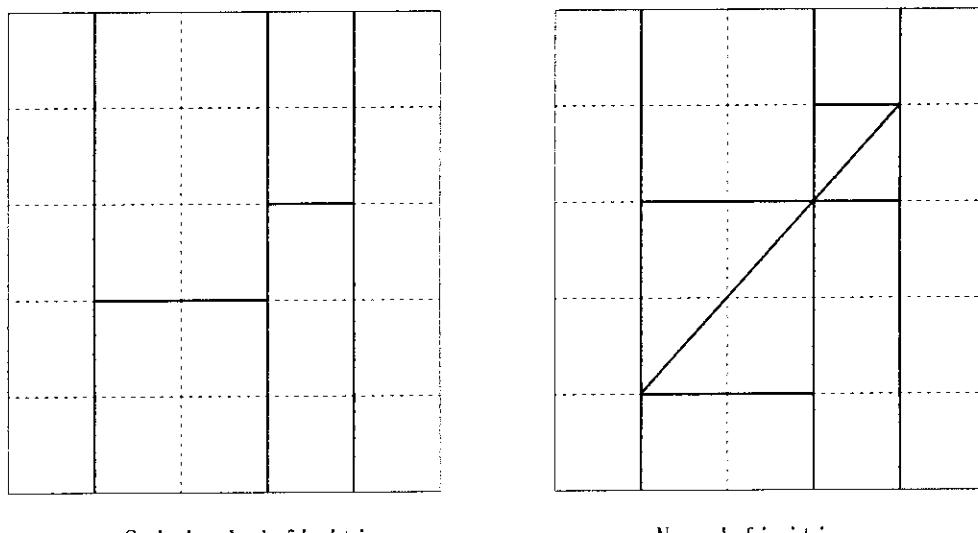
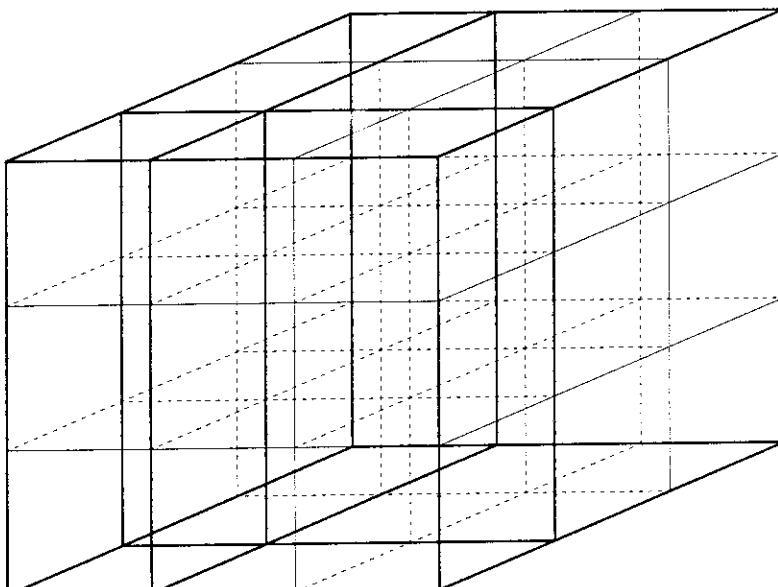
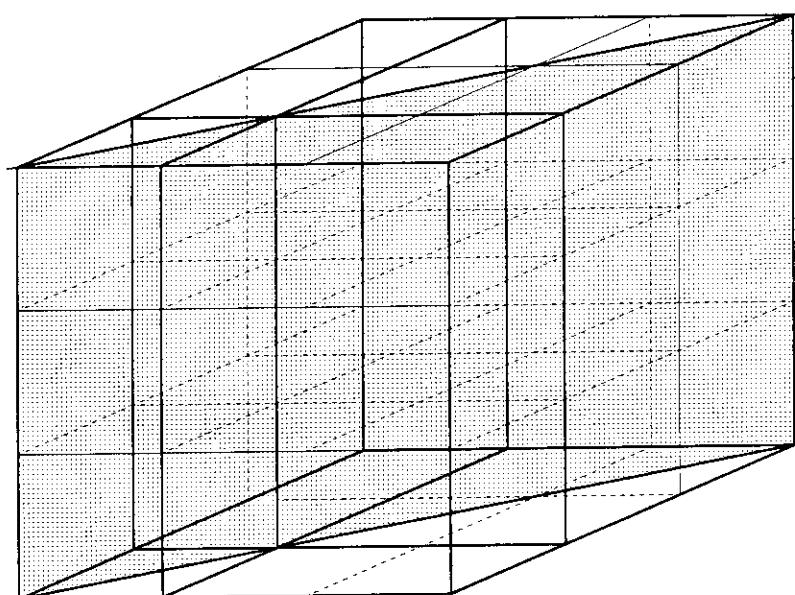


Fig. 7.1 Improvement of the region definition in the two dimensional analysis



Original definition



New definition

Fig. 7.2 Improvement of the region definition in the three dimensional analysis

(1) 旧バージョンのリージョン定義方法

HEATING7-VP コードにおけるリージョンの定義方法は次の 2 通りの方法がある。

i) 1 方向 1 配列表示

計算格子は節点を中心に考えると 8 方向に存在する。これらの格子がどのリージョンに含まれるかを示すために、それぞれの方向についてのリージョンを表す配列があり、節点の番号とリージョンの番号を表すことができる。

例

NILFR (3) = 4

ここで、NILFR は節点を中心にして X 軸の負の方向、Y 軸の負の方向、Z 軸の負の方向にある格子を意味する配列名である。添字の 3 は節点の番号である。4 はリージョン番号を表す。Table 7.1 にこれら配列名の一覧、Fig. 7.3 及び Fig. 7.4 に 2 次元、3 次元における各配列の表す方向を示す。なお、2 次元解析の場合は、Z 軸の概念はない。そのため、Z 軸の負方向、正方向に相当するリージョン番号は、同じ番号が入る。

	X 軸に対しての方向	Y 軸に対しての方向	Z 軸に対しての方向	番号
NILBR NILFR	負 負	負 負	負 正	① ②
NIRBR NIRFR	負 負	正 正	負 正	③ ④
NOLBR NOLFR	正 正	負 負	負 正	⑤ ⑥
NORBR NORFR	正 正	正 正	負 正	⑦ ⑧

Table 7.1 Table of array names that show directions

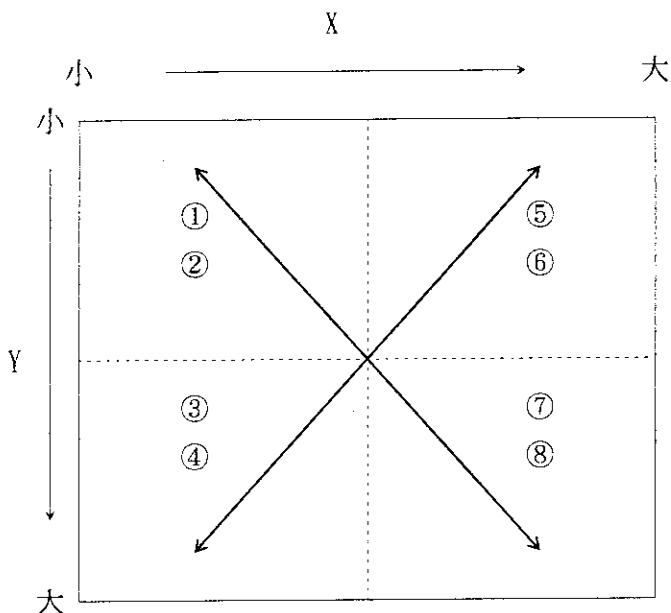


Fig. 7.3 Direction of each array names in the two dimensional analysis
(These numbers correspond to numbers of each arrays in Table 7.1)

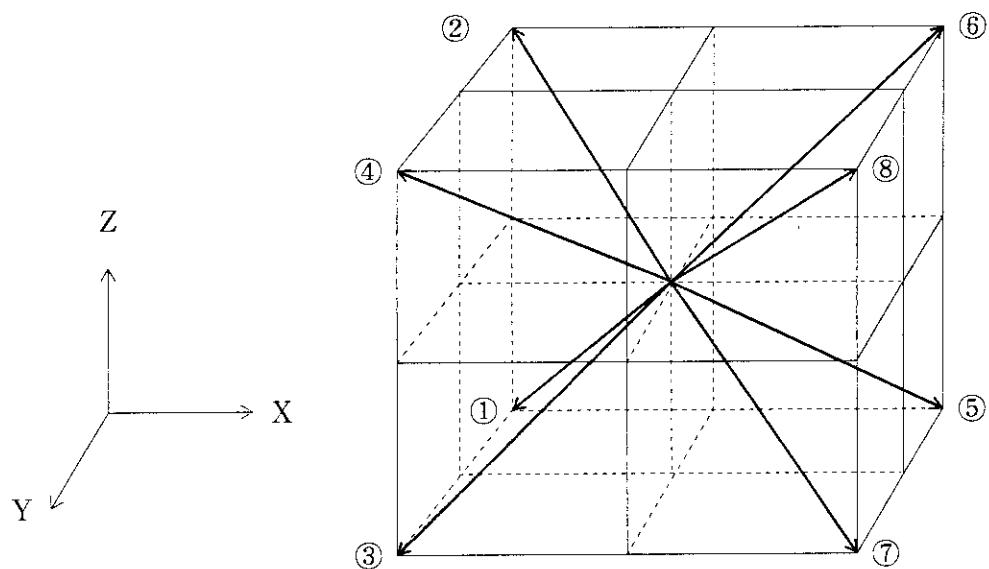


Fig. 7.4 Directions of each array names in the three dimensional analysis
(These numbers correspond to numbers of each arrays in Table 7.1)

Fig. 7.5、及びFig. 7.6に節点番号6に注目した時のこれらの配列の値を示す。この図における数字は節点番号、○内に数字がリージョン番号を表している。

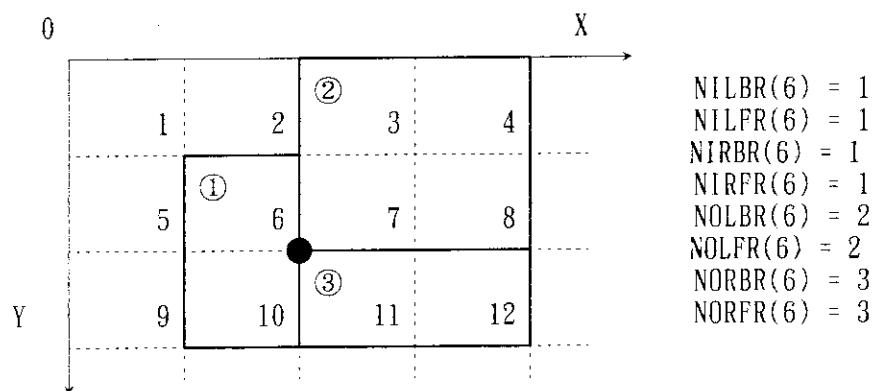
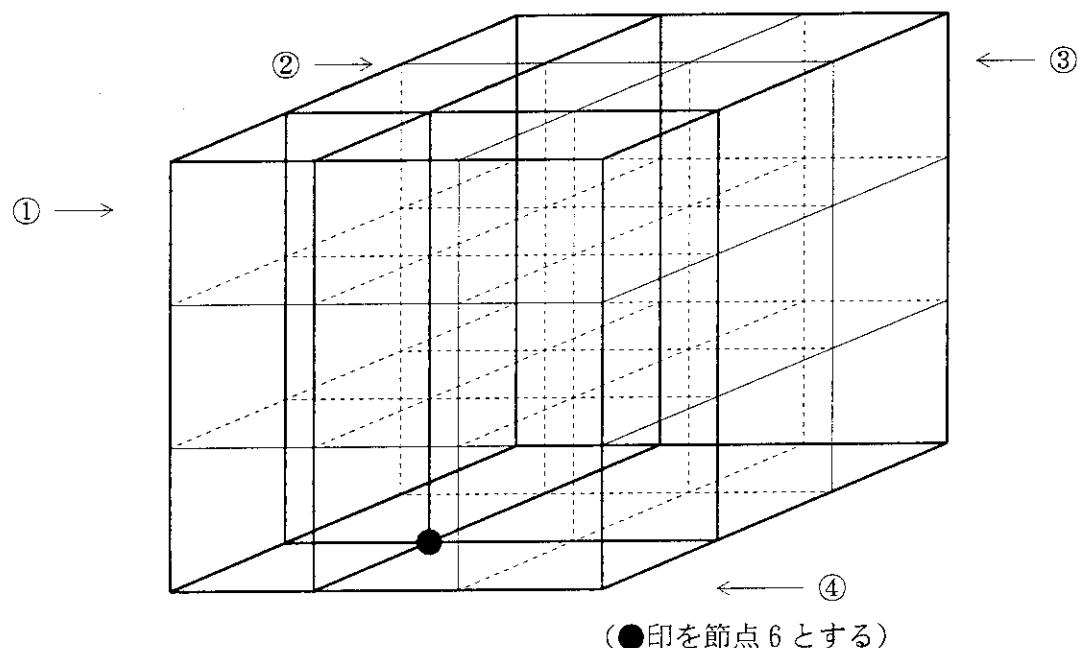


Fig. 7.5 Example of region definition in the two dimensional analysis



$\text{NILBR}(6) = 0$	$\text{NOLBR}(6) = 0$
$\text{NILFR}(6) = 2$	$\text{NOLFR}(6) = 3$
$\text{NIRBR}(6) = 0$	$\text{NORBR}(6) = 0$
$\text{NIRFR}(6) = 1$	$\text{NORFR}(6) = 4$

Fig. 7.6 Example of region definition in the three dimensional analysis

ii) 多方向 1 配列表示

i) では 1 つの方向につき、1 つの配列が決められていた。しかし、プログラムとして計算を行うには、1 つの配列で全ての方向のリージョンが表現できる方が都合がよい。HEATING7-VP では一つの配列 MREGT(サブルーチンによっては MREG) でリージョンを表現する。

MREGT (N, K, J, I) = リージョン番号

N : 節点の番号

K(1または2) : Z 値に対し、K=1 のとき負の方向、K=2 のとき正の方向

J(1または2) : Y 値に対し、J=1 のとき負の方向、J=2 のとき正の方向

I(1または2) : X 値に対し、I=1 のとき負の方向、I=2 のとき正の方向

i) の Fig. 7.5 の例を MREGT を使用して表現すると以下のようになる。

```

MREGT(6, 1, 1, 1) = 1
MREGT(6, 2, 1, 1) = 1
MREGT(6, 1, 2, 1) = 1
MREGT(6, 2, 2, 1) = 1
MREGT(6, 1, 1, 2) = 2
MREGT(6, 2, 1, 2) = 2
MREGT(6, 1, 2, 2) = 3
MREGT(6, 2, 2, 2) = 3

```

同様に、Fig. 7.6 の例を MREGT を使用して表現すると以下のようになる。

```

MREGT(6, 1, 1, 1) = 0
MREGT(6, 2, 1, 1) = 2
MREGT(6, 1, 2, 1) = 0
MREGT(6, 2, 2, 1) = 1
MREGT(6, 1, 1, 2) = 0
MREGT(6, 2, 1, 2) = 3
MREGT(6, 1, 2, 2) = 0
MREGT(6, 2, 2, 2) = 4

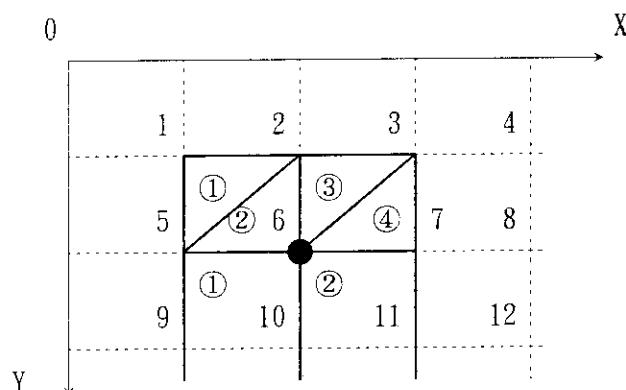
```

(2) リージョン指定方法の改良

前節での方法を用いた場合には、小格子内に 2 つの物性値を定義することはできない。そこで、配列の次元を上げることで対処することにした。次にその改良点を示す。

i) 1方向1配列表示の改良

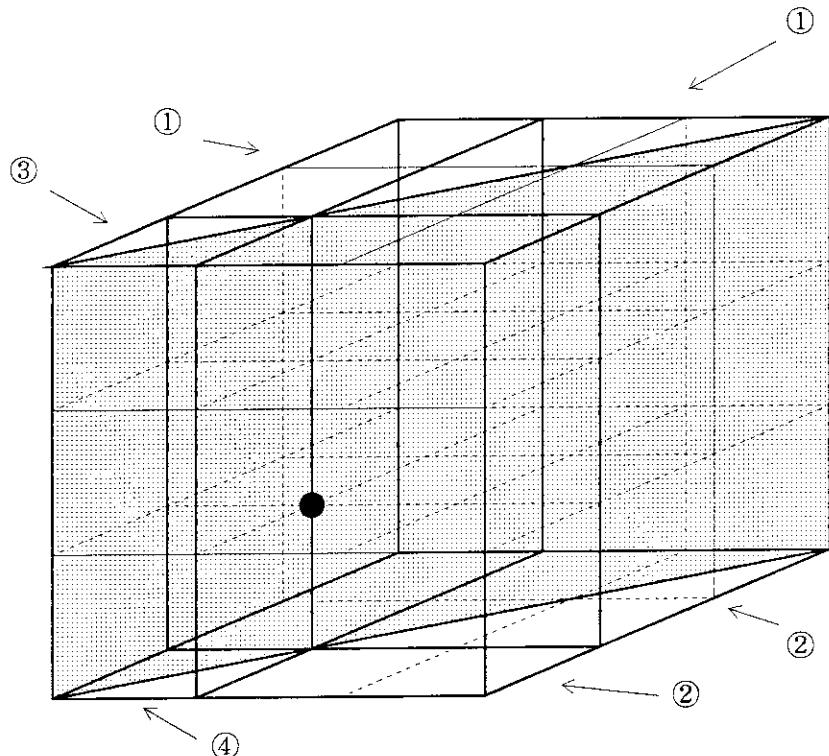
それぞれの方向を示す配列の次元を上げる。第2の添字が同一方向のそれぞれのリージョンを示す。この添字をリージョン添字と呼ぶことにする。Fig. 7.7、及びFig. 7.8にリージョン添字を使用した例を示す。



$\text{NILBR} (6, 1) = -2$	$\text{NOLBR} (6, 1) = -3$
$\text{NILBR} (6, 2) = 0$	$\text{NOLBR} (6, 2) = -4$
$\text{NILFR} (6, 1) = -2$	$\text{NOLFR} (6, 1) = -3$
$\text{NILFR} (6, 2) = 0$	$\text{NOLFR} (6, 2) = -4$
$\text{NIRBR} (6, 1) = 1$	$\text{NORBR} (6, 1) = 2$
$\text{NIRBR} (6, 2) = 0$	$\text{NORBR} (6, 2) = 0$
$\text{NIRFR} (6, 1) = 1$	$\text{NORFR} (6, 1) = 2$
$\text{NIRFR} (6, 2) = 0$	$\text{NORFR} (6, 2) = 0$

(注意) 三角形リージョンを表す場合、リージョン番号を負の値にして指定する

Fig. 7.7 Example of triangle-region in the two dimensional analysis



$\text{NILBR} (6, 1) = 1$	$\text{NOLBR} (6, 1) = -1$
$\text{NILBR} (6, 2) = 0$	$\text{NOLBR} (6, 2) = -2$
$\text{NILFR} (6, 1) = 1$	$\text{NOLFR} (6, 1) = -1$
$\text{NILFR} (6, 2) = 0$	$\text{NOLFR} (6, 2) = -2$
$\text{NIRBR} (6, 1) = -3$	$\text{NORBR} (6, 1) = 2$
$\text{NIRBR} (6, 2) = -4$	$\text{NORBR} (6, 2) = 0$
$\text{NIRFR} (6, 1) = -3$	$\text{NORFR} (6, 1) = 2$
$\text{NIRFR} (6, 2) = -4$	$\text{NORFR} (6, 2) = 0$

Fig7.8 Example of region in the three dimensional analysis

次元を上げることで小格子内で 2 つに分割されたリージョンが表現できる。リージョン添字の 1、2 の違いは、リージョンの入力順である。例えば、Fig. 7.7 の場合、三角リージョン 3 は NOLFR に対し先に入力されたため、リージョン添字 1 に定義する。よって、この順になる。一方、四角形リージョンは、リージョン添字が 1 のところに定義する。よって四角形リージョンのある方向では、必ずリージョン添字の 2 の配列の値は 0 となる。

ii) 多方向 1 配列表示の改良

この場合も 1 方向 1 配列表示の場合と同様に配列の次元を上げることで、三角形リージョン

が定義可能となる。

$$\text{MREGT}(N, L, K, J, I) = \text{リージョン番号}$$

ここでLが新たに追加したリージョン添字である。(2) i) のFig. 7.7の例と同じ計算体系を示す次のようにある。

$\text{MREGT}(6, 1, 1, 1, 1) = -2$	$\text{MREGT}(6, 1, 1, 1, 2) = -3$
$\text{MREGT}(6, 2, 1, 1, 1) = 0$	$\text{MREGT}(6, 2, 1, 1, 2) = -4$
$\text{MREGT}(6, 1, 2, 1, 1) = -2$	$\text{MREGT}(6, 1, 2, 1, 2) = -3$
$\text{MREGT}(6, 2, 2, 1, 1) = 0$	$\text{MREGT}(6, 2, 2, 1, 2) = -4$
$\text{MREGT}(6, 1, 1, 2, 1) = 1$	$\text{MREGT}(6, 1, 1, 2, 2) = 2$
$\text{MREGT}(6, 2, 1, 2, 1) = 0$	$\text{MREGT}(6, 2, 1, 2, 2) = 0$
$\text{MREGT}(6, 1, 2, 2, 1) = 1$	$\text{MREGT}(6, 2, 2, 2, 2) = 2$
$\text{MREGT}(6, 2, 2, 2, 1) = 0$	$\text{MREGT}(6, 1, 2, 2, 2) = 0$

一方Fig. 7.8の計算体系は次のように表せる。

$\text{MREGT}(6, 1, 1, 1, 1) = 1$	$\text{MREGT}(6, 1, 1, 1, 2) = -1$
$\text{MREGT}(6, 2, 1, 1, 1) = 0$	$\text{MREGT}(6, 2, 1, 1, 2) = -2$
$\text{MREGT}(6, 1, 2, 1, 1) = 1$	$\text{MREGT}(6, 1, 2, 1, 2) = -1$
$\text{MREGT}(6, 2, 2, 1, 1) = 0$	$\text{MREGT}(6, 2, 2, 1, 2) = -2$
$\text{MREGT}(6, 1, 1, 2, 1) = -3$	$\text{MREGT}(6, 1, 1, 2, 2) = 2$
$\text{MREGT}(6, 2, 1, 2, 1) = -4$	$\text{MREGT}(6, 2, 1, 2, 2) = 0$
$\text{MREGT}(6, 1, 2, 2, 1) = -3$	$\text{MREGT}(6, 2, 2, 2, 2) = 2$
$\text{MREGT}(6, 2, 2, 2, 1) = -4$	$\text{MREGT}(6, 1, 2, 2, 2) = 0$

(3) リージョン定義に関するサブルーチンと改良点

リージョン定義に関するサブルーチンで、改良を行ったものは次の3つである。

i) HEATING

HEATING7-VP はメモリ節約のためコード内で使用する各配列を、大きな配列からいくつかに区切って使用している。サブルーチンHEATING6 は大きな配列を各配列に分割するため、各配列の先頭アドレスを計算をしているルーチンである。1方向1配列表示のNILFR 等や多方向1配列表示MREGT もここでアドレス計算され、割り当てられる。今回の改良ではこれらの配列が三角形(三角柱)リージョンを表現するため次元上げされているため、それぞれオリジナルの2倍の領域を使用することになる。そこで先頭アドレス計算部分を変更し、2倍の領域を割り当てるようにした。

ii) PNTSTR

PNTSTRは1方向1配列表示のNILFR等にリージョン番号を定義するルーチンである。ここで四角形(直方体)リージョン、及び三角形(三角柱)リージョンをNILFR等に定義できるようにNILFR等の配列の次元上げを行う。但し四角形(直方体)リージョン、及び三角形(三角柱)リージョンを定義するときは、後の計算でリージョンを検索する場合の効率を考慮して、リージョン添字が1から定義するようにする。従ってリージョン添字1に四角形リージョンが定義された場合はリージョン添字2には0が設定される。また、リージョン添字1に三角形リージョンが定義されている場合は、リージョン添字2に三角形リージョンが定義されていることがある(小格子内に2つのリージョンが設定されている場合)。変更例をFig. 7.9に示す。

DO 4200 N = 1, NT
 DO 4100 I = 1, NRGTRT
 リ
 ।
 ジ
 ョ
 ン
 の
 ル
 ।
 プ
 節
 点
 の
 ル
 ।
 プ

```

    IF( ..... ) THEN
      IF( ..... ) THEN
        C.TN      NILBR(N) = -I           ←— オリジナル
        IF(NILBR(N,1), EQ. 0) THEN      (コメント化)
          NILBR(N, 1) = -I
        ELSE
          NILBR(N, 2) = -I
        END IF
      END IF
    END IF
  
```

4100 CONTINUE
 4200 CONTINUE

Fig. 7.9 Example of improvement

iii) POINTS

POINTSは、1方向1配列表示のNILFR等のリージョン定義ルーチンであり、NILFR等の初期化（ゼロクリア）と四角形（直方体）リージョンの定義を行う。初期化はリージョン添字1、及び2に対して行う。ここでは四角形のみのリージョン定義を行うのでリージョン添字1に関してのみ行うようとする。

7.2.2 物性値計算部分の改良

各リージョンにおける面对面の物性値の計算は、サブルーチン THRMPR 以降で行われる。THRMPR以下のプログラムのツリー構造をFig. 7.10 に示す。この中でTHRMT0、THRMT1、THRMT2、THRMT3が各リージョンに関して計算を行っている。今回の改良では、これらのサブルーチンで三角形（三角柱）リージョンの計算が行えるようにプログラムを修正する必要がある。但し、THRMPRはリージョンに関する計算を行っているが、このサブルーチンは四角形（直方体）リージョンのみの計算しか行わないので、三角形（三角柱）リージョンが扱えるような改良は行わない。

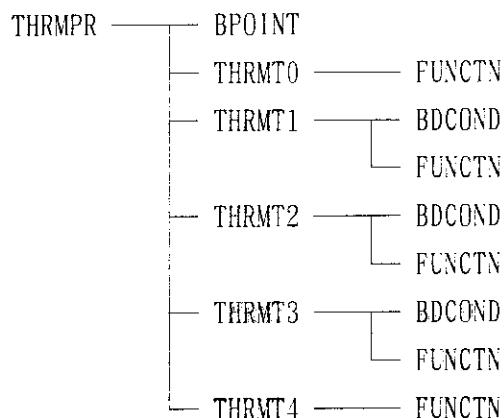


Fig. 7.10 Tree structure of subroutines

(1) 各サブルーチンの機能と改良点

i) THRMPR

物性値計算部分の実質上メインルーチンである。節点に関するDOループがあり各節点の回りのリージョンについて計算を行う。このサブルーチン内では何箇所かで各リージョンを参照するが、四角形（直方体）リージョンしか扱わないので、MREGT のリージョン添字が 1 についてのみ考えればよい。三角形（三角柱）リージョンを含む計算はこのサブルーチンから呼ばれる THRMT0、THRMT1、THRMT2、THRMT3、THRMT4で行われる。

ii) BPOINT

初期状態における境界条件の設定を行うサブルーチンである。1 方向 1 配列表示である NILFR 等を使って多方向 1 配列表示の MREG を定義している。これらを三角形（三角柱）リージョンが定義できるように配列の次元上げを行う。ただし、ここでは四角形（直方体）リージョンのみの計算を扱う。

iii) THRMTO

各節点の回りに存在するリージョンについての計算を行う。これらのリージョンの中に三角形（三角柱）リージョンが存在する場合、それらについても参照するので、リージョン添字の1、及び2のそれぞれを参照する。具体的にはリージョン添字LをDOループで1から2まで回すようにした。Fig. 7.11 に改良の概要を示す。

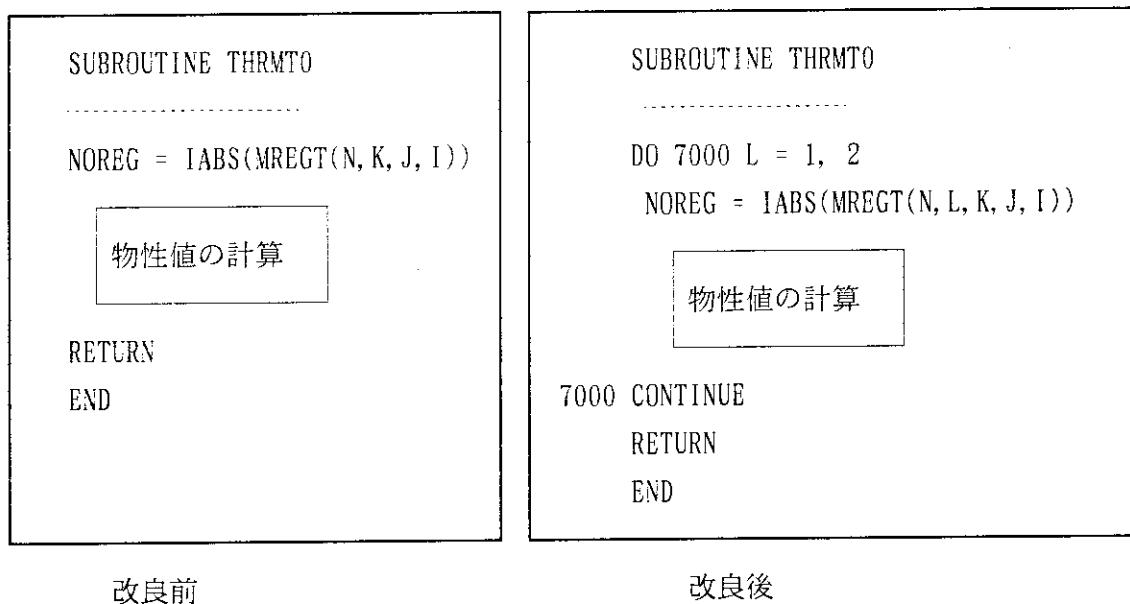


Fig. 7.11 Improvement of THRMTO subroutine

iv) THRMTO1

各節点の回りに存在するリージョンについての計算を行うが、そのリージョンの回りのリージョンからの影響も考慮する（但しY、Z方向のみ）。THRMTOと同様、全てのリージョンに関して計算を行うので、リージョン添字が1、及び2のそれぞれを参照する。修正内容もTHRMTOと同様である。

v) THRMTO2

各節点の回りに存在するリージョンについての計算を行うが、そのリージョンの回りのリージョンからの影響も考慮する（但しX、Z方向のみ）。THRMTOと同様、全てのリージョンに関して計算を行うので、リージョン添字の1、及び2のそれぞれを参照する。修正内容もTHRMTOと同様である。

vi) THRMTO3

各節点の回りに存在するリージョンについての計算を行うが、そのリージョンの回りのリージョンからの影響も考慮する（但しX、Y方向のみ）。THRMTOと同様、全てのリージョンに関

して計算を行うので、リージョン添字の 1、及び 2 のそれぞれを参照する。修正内容も THRMTO と同様である。

vii) THRMT4

各節点の回りに存在するリージョンについての計算を行う。そのリージョンの回りのリージョンの影響は考慮されない。全てのリージョンに関して計算を行うので、リージョン添字の 1、及び 2 のそれぞれを参照するようとする。修正内容は THRMTO と同様である。

viii) PREP

HEATING と温度分布を印刷するサブルーチン TEMPOUT のインターフェイスルーチンである（但し、THRMPR 下にはない）。各地点での温度を決定するために、その場所の材質も要因となる。ここでは材質を判断するために、1 方向 1 配列表示の NIRFR、及び NOLFR によるリージョン番号が参照される。これらの配列を次元上げする。

7.3 テスト計算

7.3.1 入力データ

三角柱リージョンが隣合った3次元入力データを用いて、テスト計算を行った。入力データのリージョンの構造はFig. 7.12 の通りである。また、実行JCL をFig. 7.13 に示す。

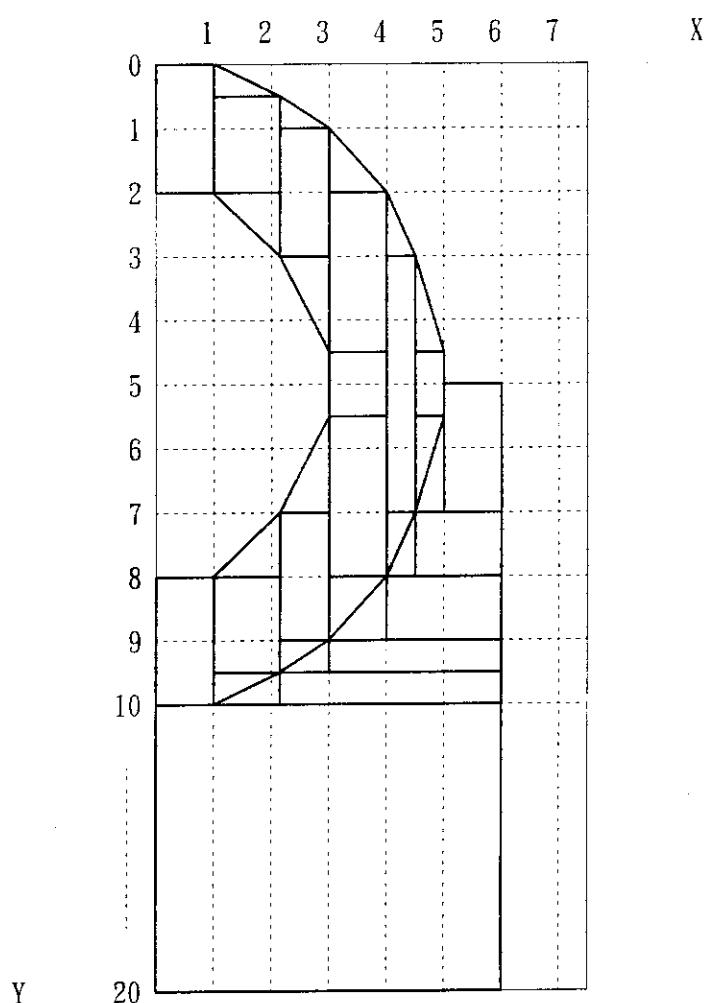


Fig. 7.12 Example of input data
(ALL regions have the same Z-value)

```

TWC(5 4 8 3 5) SRP
//FORT77 EXEC FORT77,SO='J9127.ALLHEAT',A='ELM(*',
//           B='AE',OPT=3,RGN=5000K
// EXEC LKED77
//LINK.SYSIN DD DSN=J3355.HEAT6ASM.OBJ(ASM01),DISP=SHR,LABEL=(,,,IN)
// EXEC GO,OBSIZE=137
//FT05F001 DD DSN=J9127.HEATINGS.DATA(TESTRUN),DISP=SHR,LABEL=(,,,IN)
//FT06F001 DD DSN=J9127.F6S4K@9.DATA,DISP=(NEW,CATLG,DELETE),
//           UNIT=TSSWK,SPACE=(TRK,(300,100),RLSE),
//           DCB=(RECFM=FBA,BLKSIZE=20550,LRECL=137)
//FT01F001 DD DSN=J9127.F1S4K@9.DATA,DISP=(NEW,CATLG,DELETE),
//           UNIT=TSSWK,SPACE=(TRK,(100,100),RLSE),
//           DCB=(RECFM=VBS,LRECL=23472,BLKSIZE=23476)
//FT04F001 DD DSN=J9127.F4S4K@9.DATA,DISP=(NEW,CATLG,DELETE),
//           UNIT=TSSWK,SPACE=(TRK,(100,100),RLSE),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//

```

Fig. 7.13 Example of JCL

7.3.2 計算結果

この入力データを用いて実際に計算を行った。計算結果は検討の結果、妥当の値が得られることが分かった。なお、実行はFujitsu M780 上でコンパイラはFORTRAN77 V10L30 で行った。実行時間は 9秒47であり、使用メモリは基本領域が 928KB、拡張領域が 27.8MB である。また I/O 回数は61回である。

8. SCDAP/RELAP5のAE化

8.1 概要

炉心燃料損傷挙動解析コードSCDAP/RELAP5⁽²⁰⁾、及び軽水炉安全性解析コードRELAP5⁽²¹⁾は米国アイダホ国立研究所(INEL)で開発され、不定期に日本原子力研究所(以下原研)に配布されている。これらのコードは大規模であり、今回作業を行ったSCDAP/RELAP5/MOD2.5コードでは、サブルーチン数784、ライン数は約17万行にも達している。

現在、これらのロードモジュールは、メモリ節約のためオーバレイ構造となっている。しかし、SCDAP/RELAP5コードのロードモジュールをオーバレイ構造にしても、解析対象によってはメモリの基本領域の許容最大値8Mバイト以内で実行できない。RELAP5コードのロードモジュールも、約7Mバイトを必要とする。将来、解析対象を大きくする場合や、新コンパイラ“FORTRAN77EX”を使用してロードモジュールを作成する場合、8Mバイトを越える可能性があり、これでは実行不可能である。

本作業では、ロードモジュールのオーバレイ構造をやめ、AE化版ロードモジュールを作成した。AE化を行うと大きな配列変数やコモン領域がメモリの拡張領域に展開されるため、配列変数を大きくとることができ、解析対象をかなり大きくすることができる。AE化版ロードモジュールを作成するために変更を加えたソースはSCDAP/RELAP5/MOD2.5である。RELAP5コードも同様の方法でAE化することができる。

この章ではAE化のための変更内容、実際のメモリ使用状況などについて述べることにする。

8.2 AE化のための調査

ここでは、AE化版ソースプログラムを作成するために、SCDAP/RELAP5/MOD2.5のソースプログラムについて調査を行った結果について述べる。

8.2.1 オーバレイ構造とAE化

(1) オーバレイ構造

通常の単純構造のロードモジュールは、実行されるとき、一度に全てのロードモジュールをローディングする。それに対し、オーバレイ構造のプログラムは、いくつかのセグメントからなるロードモジュールであり、このロードモジュールが実行されるとき、セグメントを単位としてローディングが行われる。したがって、実行に必要な主記憶容量は、最大のセグメントの容量となる。このため、実行時に使用するメモリ領域は、単純構造のロードモジュールよりオーバレイ構造のロードモジュールのほうが少なくて済む。

SCDAP/RELAP5及びRELAP5コードを単純構造のロードモジュールにすると、大きくなりすぎてしまい、大部分のコンピュータでは実行不可能である。そのため、オーバレイ構造のロード

モジュールを採用している。

(2) AE化

AE化とは、16M バイト以下の基本領域（原研では8Mバイト）だけで実行していたプログラムを、16M バイトを越えた拡張領域も使用して実行できるようにすることである。具体的には、アドレス拡張機能を利用することにより、

- i) ロードモジュールの拡張領域での実行
- ii) 巨大データの拡張領域への割り付け

が可能となる。

i) は、システム制御プログラムがアドレス拡張機能を備えていれば可能であり、原研のシステムにおいても可能である。ただし、オーバレイ構造のプログラムは拡張領域で実行することはできない。

ii) は、コンパイラオプションとしてAEを指定して翻訳することにより、16M バイトを越えるような特定の巨大データ（配列）を使用できるようにする。拡張領域に割付けされるデータは、COMMONブロックに属する変数および配列（ただし、そのCOMMONブロックに属するすべての変数、及び配列に対して初期値が設定されてないこと）とCOMMONブロックに属さない初期値の設定されていない配列である。

8.2.2 ソースプログラムの構成

SCDAP/RELAP5/MOD2.5 コードは、1組のアセンブリソースと3組のFORTRAN ソースから成っている。これらについてTable. 8.1 に示す。

Table. 8.1 Organization of the SCDAP/RELAP5/MOD2.5 code

ソース名	ファイル名(ID:J0001)	内容	言語
SELP3FS	SR. SELP3FS. FORT77 SR. SELP3FS. INCLUDE	SCDAP 本体	FORTRAN
ENVRLS	SR. ENVRLS. FORT77 SR. ENVRLS. INCLUDE	Environmental library	FORTRAN
MATPROS	SR. MATPROS. FORT77 SR. MATPROS. INCLUDE	Material property library	FORTRAN
なし（原研 で作成）	LT. ENVR19S. ASM	残りCPU 時間、ロケーション バイトアドレスを得る	アセンブリ

8.2.3 コードのAE化時の障害と解決方法

(1) 配列変数の初期化

SELP3FS、ENVRLS、MATPROS はFORTRAN で書かれているので通常のFORTRAN コンパイルは可能であるが、AEオプションを指定しても拡張領域を使用することができない。理由は SELP3FS にはブロックデータが存在し、またENVRLSにも同じくブロックデータが存在するためである。ブロックデータは、COMMONブロックに含まれる変数、配列に DATA 文によって初期値を与えるものであり、AE化の制約を満足しない。また、あるサブルーチンでは、COMMON変数にDATA文により初期値を与えていている。これもAE化の障害となっている。

(2) アセンブリ

アセンブリソースは拡張領域を用いて実行することを前提としていない。通常のアセンブリソースをアセンブルすると、基本領域で実行するような (RMODE [常駐モード] = 24 、 AMODE [動作モード] = 24) オブジェクトが作成される。このモードでは、拡張領域での実行ができない。そのため RMODE=ANY、 AMODE=31 で作成する必要がある。

(3) オーバレイ構造

SCDAP/RELAP5シリーズ(RELAP5 も含む) のロードモジュールの特色は、オーバレイ構造になっていることである。これは以前、SCDAP/RELAP5の開発が、主記憶メモリが小さい CDC 計算機で行われていたためである。拡張領域を使用するロードモジュールは、8.2.1 で述べた理由により、オーバレイ構造であってはいけない。

(4) グラフ出力機能

SCDAP/RELAP5シリーズには、グラフィックライブラリDISSPLA を用いたグラフ出力機能が備えている。そのため、ロードモジュール作成時(LINK時)には、DISSPLA を結合する必要がある。DISSPLA ライブラリは、RMODE=24、 AMODE=24 で作成されているため、拡張領域での実行は不可能である。

従ってSCDAP/RELAP5シリーズは、解析計算とグラフ出力は別々のプログラムを採用することにし、解析計算にはAE化版を、グラフ出力にはオーバレイ版を使用することにした。

8.3 AE化のための変更作業

8.3.1 使用しないサブルーチン、ファンクションの削除

SELP3FS、ENVRLS、MATPROS に含まれるサブルーチン、ファンクションの中には親ルーチンが存在しないものがある。これらの副プログラムの中には、COMMONデータに対し、 DATA 文によって初期値を与えていているものがあるため、 AE 化の障害となっている。なるべくソース自体を小さくする必要もあり、削除することにした。

8.3.2 ブロックデータの変更

(1) ブロックデータによる初期値設定の廃止

SELP3FS には “BLKDTA” 、 ENVRLS には “ENVBLK” というブロックデータがある。BLKDTAに對しては、 DATA文による初期値設定をやめ、代入文による定義に置き換えた(但し、 COMMONブロックMASKBKに関しては、小さいのでそのままにしておく)。またブロックデータの形式からサブルーチンの形式に変更し、メインルーチンRELAP5より呼び出されるようにした。変更後の BLKDTA の一部を Fig. 8.1 に示す。ENVBLKについては、サブルーチン化した BLKDTA の中で代入文によって設定するように変更し、ブロックデータ ENVBLK を削除した。

(2) ブロックデータのサブルーチン化に伴う変更

BLKDTAのサブルーチン化に伴い、他のサブルーチンについても修正が必要になった。BLKDTAの中で宣言されていたCOMMONブロックのうち、 COMMON ブロック RCOMPA の初期値設定はインクルードファイル (インクルード名: CMPALF) の内部で、 DATA文による初期値設定をしていた。BLKDTAをサブルーチン化するために、このインクルードファイルを無効 (コメント化) にし、サブルーチン BLKDTA の中で、代入文によって初期値を設定した。

このインクルードファイルを参照しているサブルーチンとファンクションの中で、COMMON宣言がなく通常のローカルの配列宣言があるものがあった。インクルードファイル CMPALF の DATA 文が有効であれば、その値が参照されるため問題はない。しかし、BLKDTAをサブルーチン化する時に、インクルード CMPALF を無効にしたため、これらローカル配列は未定義になってしまった。調査の結果、COMMONブロック RCOMPA に含まれる配列は、初期値設定した後は再定義されることがないことが分かったので、 CMPALF をインクルードしていたサブルーチンとファンクションの中で RCOMPA の COMMON 宣言を設けることにした (Fig. 8.2 参照) 。 RCOMPA の COMMON 宣言を加えた副プログラムは IEDIT、ISSI、MAJOUT、RCKPLT である。

```

CAE      DATA ((CSOL(FE ,J),J=1,3) /3* 0.0/
              CSOL(FE ,1) = 0.000
              CSOL(FE ,2) = 0.000
              CSOL(FE ,3) = 0.000
CAE      DATA ((CSOL(RU ,J),J=1,3) /3* 0.0/
              CSOL(RU ,1) = 0.000
              CSOL(RU ,2) = 0.000
              CSOL(RU ,3) = 0.000
CAE      DATA ((CSOL(BA ,J),J=1,3) /3* 0.0/
              CSOL(BA ,1) = 0.000
              CSOL(BA ,2) = 0.000
              CSOL(BA ,3) = 0.000
CAE      DATA ((CSOL(SB ,J),J=1,3) /3* 0.0/
              CSOL(SB ,1) = 0.000
              CSOL(SB ,2) = 0.000
              CSOL(SB ,3) = 0.000
CAE      DATA ((CSOL(CS ,J),J=1,3) /3* 0.0/
              CSOL(CS ,1) = 0.000
              CSOL(CS ,2) = 0.000
              CSOL(CS ,3) = 0.000
CAE      DATA ((CSOL(I ,J),J=1,3) /3* 0.0/
              CSOL(I ,1) = 0.000
              CSOL(I ,2) = 0.000
              CSOL(I ,3) = 0.000
CAE      DATA ((CLIQ(I2 ,J),J=1,4) /-5.21E+03, 22.9, 0.0, 0.0/
              CLIQ(I2 ,1) = -5.21D+03
              CLIQ(I2 ,2) = 22.9D0
              CLIQ(I2 ,3) = 0.0D0
              CLIQ(I2 ,4) = 0.0D0
CAE      DATA ((CLIQ(CSI ,J),J=1,4) /-1.55E+04, 20.4, 0.0, 0.0/
              CLIQ(CSI ,1) = -1.55D+04
              CLIQ(CSI ,2) = 20.4D0
              CLIQ(CSI ,3) = 0.0D0
              CLIQ(CSI ,4) = 0.0D0
CAE      DATA ((CLIQ(CSOH,J),J=1,4) /-1.54E+04, 22.8, 0.0, 0.0/
              CLIQ(CSOH,1) = -1.54D+04
              CLIQ(CSOH,2) = 22.8D0
              CLIQ(CSOH,3) = 0.0D0
              CLIQ(CSOH,4) = 0.0D0
CAE      DATA ((CLIQ(TE ,J),J=1,4) /-1.84E+04, 56.9, -9.95, 0.0/
              CLIQ(TE ,1) = -1.84D+04
              CLIQ(TE ,2) = 56.9D0
              CLIQ(TE ,3) = -9.95D0
              CLIQ(TE ,4) = 0.0D0
CAE      DATA ((CLIQ(HI ,J),J=1,4) /-0.94E+03, 8.53, 0.236, 0.0/
              CLIQ(HI ,1) = -0.94D+03
              CLIQ(HI ,2) = 8.53D0
              CLIQ(HI ,3) = 0.236D0
              CLIQ(HI ,4) = 0.0D0
CAE      DATA ((CLIQ(H2TE,J),J=1,4) /-0.35E+03,-13.21, 8.04, 0.0/
              CLIQ(H2TE,1) = -0.35D+03
              CLIQ(H2TE,2) = -13.21D0
              CLIQ(H2TE,3) = 8.04D0
              CLIQ(H2TE,4) = 0.0D0

```

Fig. 8.1 A part of subroutine BLKDTA

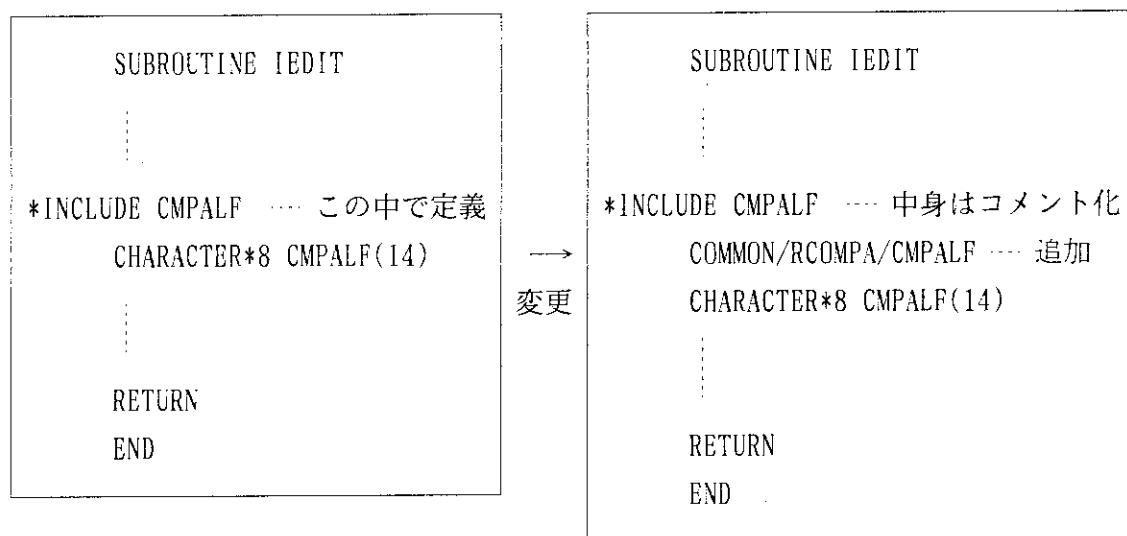


Fig. 8.2 Example of improvement (subroutine IEDIT)

(3) サブルーチンでのDATA文による初期値設定の廃止

サブルーチンPMINVDでは、COMMONブロックPMNVCC、及びPMINVCに含まれる配列変数にDATA文による初期値設定をしている。調査の結果、初期値設定のみで再定義はされない。そこで、サブルーチンBLKDTAにCOMMON宣言を設け、そこで代入文による初期値設定をした。

8.3.3 アセンブリ

オーバレイ版で使用しているアセンブリソースをそのままアセンブルすると、 RMODE=24、AMODE=24の基本領域動作モードになってしまふ。拡張領域を使用するために、 RMODE=ANY、AMODE=31の拡張領域動作モードにする必要があるので、アセンブリソースにRMODE 命令、およびAMODE 命令を挿入した。

また、ロケーションバイトアドレスを得る“LOCB”は24ビットアドレッシングモード用に作成されている。これも、AMODE=31、すなわち31ビットアドレッシングモード用に変更する必要がある。Fig. 8.3に、変更点を図示したアセンブリソースを示す。

8.3.4 リンケージ制御文

SCDAP/RELAP5/MOD2.5 コードは、結合編集 (LINK) 時に 3 種類のリンケージ制御文を採用していた。それは、オーバレイ構造のロードモジュールを作成するためのOVERLAY 制御文、及びINSERT制御文と、 COMMON セクションを順序付けするORDER 制御文である。

オーバレイ構造を廃止するため OVERLAY制御文、およびINSERT制御文をなくした。 ORDER制御文によるCOMMONセクションの順序付けは必要である（すなわちSCDAP/RELAP5コードは、いく

つかのCOMMONセクションを順序づけることで、あたかもひとつの大きな COMMON領域として扱っていることと、COMMONブロックのBUNTIMとFTBFETの間にSCDAP 計算部分で使用するCOMMONブロックが存在しなくてはいけないようになっているためである）。したがって、ORDER 制御文はそのままにした。この制御文をFig. 8.4に示す。

```

*
*      *** RCPU ***
*
*      THIS FUNCTION RETURNS THE REMAINING CPU TIME
*          FROM CURRENT JOB STEP TIME
*          (AE VERSION)
*
*      X = RCPU ( < XE, XL > )
*
*          WHERE RCPU : REMAINING TIME ( SEC )    REAL*8
*                XE   : ELAPSED TIME   ( SEC )    REAL*8 ( OPTION )
*                XL   : LIMITED TIME   ( SEC )    REAL*8 ( OPTION )
*
*          ** CAUTION **    IF RESULT IS NEGATIVE, ERROR IS DETECTED.
*
RCPU      CSECT
RCPU      AMODE    31  ←
RCPU      RMODE    ANY ←
B         12(15)
DC        X'7'
DC        CL7'RCPU ,
STM       14,12,12(13)
LR        12,15
USING    RCPU,12
ST        13,SAVE+4
LA        13,SAVE
*
*      GET ELAPSED TIME
*
L         4,X'214'
LM        2,3,160(4)
SRDL     2,12
STM       2,3,TIME
OI        TIME,X4E
LD        2,TIME
LTDR     2,2
BNP       ERROR
DD        2,CONVO1
LTR       1,1
BZ        E01
L         5,0(1)
STD       2,0(5)
EO1      EQU    *
*
*      GET LIMITED TIME
*
LM        2,3,224(4)
SRDL     2,12
STM       2,3,TIME
OI        TIME,X4E
LD        0,TIME
LTDR     0,0
BNP       ERROR
DD        0,CONVO1
LTR       1,1
BZ        L01

```

Fig. 8.3 Assembler source for AE version (continued)

```

L      5,4(1)
STD   0,0(5)
L01  EQU   *
*
*      REMAINING = LIMITED - ELAPSED
*
SDR   0,2
*
*
RETURN EQU   *
L      13,SAVE+4
LM    0,12,20(13)
XR    15,15
MVI   12(13),X'FF'
BR    14
*
*
ERROR  EQU   *
LD    0,NEGA
B     RETURN
*
*
SAVE   DS    18F
NEGA   DC    D'-1.0'
CONV01 DC    D'1000000.0'
TIME   DS    D
X4E    EQU   X'4E'
END
*
*      *** LOCB ***
*
*      THIS FUNCTION RETURNS THE LOCATION OF VARIABLE.
* (AE VERSION)
*      I = LOCB (< X >)
*
*      WHERE LOCB : LOCATION (BY BYTE)
*      X    : VARIABLE
*
LOCB  CSECT
LOCB  AMODE  31 ←
LOCB  RMODE  ANY ←
L     0,0(1)
SLL   0,1
SRL   0,1
*     SRL   0,4
*     < SRL   0,1 > *** BY BYTE
*     < SRL   0,4 > *** BY 8 BYTE WORD
BSM   0,14
END

```

Fig. 8.3 Assembler source for AE version

```
*      SCDAP/RELAPS/MOD2.5      AE VERSION
*
*      MARCH.28 1991
*
*===== SEG100 =====
*
* OVERLAY NODE000
    ORDER  FAST,GENRL,COMCTL,CTRL,STATEC
    ORDER  STATEC,VIRTUL,RMADAC,BUNTIM
***** SCDAP COMMONS START POINTER BUNTIM
    ORDER  BUNTIM,SCDOUT,THPLOT,INTCOM,PLNDAT
    ORDER  PLNDAT,IOCOM,SOLCOM,CONS,SCDCOM,NHTARA
    ORDER  NHTARA,FGRCOM,DEBOUT
    ORDER  DEBOUT,RGACCT,EFFPRP,TRNOT1,TRNOT2,CMATP
    ORDER  CMATP,NDXAARA,EHTO,MISCON
    ORDER  MISCON,TBLSP,FGINV,CMDAT,SLBCOM,GRSCGR
    ORDER  GRSCGR,PARG,GRSPRG,VHEADG,CDFINC,BALFAR
    ORDER  BALFAR,RADATA,CPRDAT,PRDAT,MADATC
    ORDER  MADATC,EHTCO,HFUSON,FARAYS,BLN2WK
    ORDER  BLN2WK,NBTIM,CMPTIM,BCONDS
    ORDER  BCONDS,PTS COM,CDFINV,BLOONA,BLOONB,BLOOND
    ORDER  BLOOND,DXCOM1,DXCOM2,DXCOM3
    ORDER  OXCOM3,FPNONV,FPMASS,BWGEOM
    ORDER  BWGEOM,BWPROP,BWCONS,BWTRNS,SLUMPV,HARDPN
    ORDER  HARDPN,FECOM,NRCOM,UOXCOM,ALCM,COUPL
    ORDER  COUPL,CPROPS,IPARM,PARM,RUPT,DEBCOM
    ORDER  DEBCOM,SCDDAT,UPHDPN
    ORDER  UPHDPN,FTBFET
ENTRY RELAPS
NAME SELAP
```

Fig. 8.4 Linkage control statement for AE version

8.4 AE化のための変更作業後のロードモジュールについて

8.3で変更を施したAE化版を用いて、ロードモジュールを作成してみた。SCDAP/RELAP5コードは、解析する対象によって作業領域を大きくしたり小さくしたりできる。これは、LFSIZというパラメータで、いくつかのCOMMONブロックの配列の大きさを決定しているためである。LFSIZを何通りか変えてロードモジュールを作成し、リンクエージMAPよりその大きさを調査してみた。なお、今回のAE化版ロードモジュールは、解析計算専用として作成したので、DISSPLAなどのグラフィックライブラリは結合しなかった。LINK時に、オプションLETをつければ、未定義な外部参照があってもその外部参照を実行時に参照しなければ正しく動作する。

LFSIZ=15000 の時

基本領域	=	4.3MB
拡張領域	=	4.5MB

LFSIZ=20000 の時

基本領域	=	4.3MB
拡張領域	=	5.3MB

LFSIZ=27000 の時

基本領域	=	4.3MB
拡張領域	=	6.4MB

このように、作業領域をかなり大きくしても基本領域は8Mバイト以下なので、原研のシステムで実行可能である。また、拡張領域についても原研では、M780で30Mバイト（昼間）、VP-2600で200Mバイト（同）まで使用可能なので作業領域をさらに大きくすることが可能となった。

9. イオンビーム計算コードと磁場計算コードの結合

9.1 概要

本章では、イオンビーム軌道計算コードNBI3DGFNの改良作業について述べる。今回の改良作業は以下の3項目について行った。

① 3次元磁場解析コードMAGICとの結合

既存のNBI3DGFNは3次元磁場解析コードGFUNを包含しており、GFUNで計算された磁場分布データを基にイオンビームの軌道計算をしている。しかし、GFUNは計算ケースにより計算結果が正しく収束しない場合があり、計算に制限があった。そこで、このGFUNの部分を別の3次元磁場解析コードELF/MAGICと置き換え・結合し、ELF/MAGICで計算された磁場分布データを基に軌道計算が出来るようにした。

② 図化出力の改良：等高線プロットの分割

NBI3DGFNの図化出力のうち、壁面上の熱負荷分布出力はダクト内面の熱負荷分布を1つの平面に展開して描いていたが、これを4面に分けて出力した。

③ 図化出力の改良：1次元図化出力の追加

②で出力された等高線図を縦、または横に切った1次元図形が出力できるように機能を追加した。

今回の改良作業におけるプログラム修正情報はTable9.2～Table9.7にまとめた。

9.2 3次元磁場解析コードELF/MAGICとの結合

9.2.1 結合方法

ELF/MAGIC（以下MAGIC）は株エルフにて開発・販売されている3次元・非線形静磁場解析ソフトウェアであり、原研には現在、ロードモジュールの形で導入されている。MAGICでは改良された積分方程式を用いており、その特徴は次の通りである。

- ① 有限要素法と異なり空間メッシュを切る必要が無く、また磁性材要素用のメッシュも連続である必要がないため、細かい構造と大きな構造の入り組んだ3次元構造の場合等でも容易に入力データの作成ができる。
- ② 計算時間が短い。

- ③ 積分方程式法に改良が施されているため、粗いメッシュでも計算精度が高い。また、正しく入力データが作成されている場合はほぼ確実に計算が収束する。

G F U Nで磁場計算を行った場合のN B I 3 D G F Nの処理流れの概略はFig. 9.1の通りである。今回の改良ではG F U NとM A G I Cのどちらでも磁場計算が出来るようにし、その選択はN B I 3 D G F Nの入力データ中で指定出来るようにした。

N B I 3 D G F Nで使用する磁場分布データをM A G I Cで計算したものと置き換えるには、あらかじめM A G I Cを用いて計算し、ファイルに保存しておいた磁場分布データを読み込むといった手順が必要となる。M A G I Cでの磁場計算結果はユニット番号10のファイル（これをMAPファイルと呼ぶ）に出力される。N B I 3 D G F NとM A G I Cの結合は、このMAPファイルを介して行う。MAPファイルでは、磁場データはV G カードと呼ばれるレコードのフォーマットで記述されている。Fig. 9.2にM A G I C磁場データを使用する場合のデータの流れを、Fig. 9.3にM A G I C磁場データを使用する場合のN B I 3 D G F Nの処理流れの概略を示す。また、Fig. 9.4にV G カードのフォーマットを示す。

MAPファイル中の磁場ベクトルデータは、M A G I Cでの磁場計算の際に与えた空間点での値しか存在しないので、任意の座標における磁場ベクトルは、その座標を囲む8つの空間点の磁場ベクトルデータから線形内挿により求める。Fig. 9.5における任意の座標Pの磁場ベクトルは、式(9-1)で表される。

$$\vec{B} = \{ \vec{dX1*dY1*dZ1*Bm(IX, IY, IZ)} + \\ \vec{dX0*dY1*dZ1*Bm(IX+1, IY, IZ)} + \\ \vec{dX1*dY0*dZ1*Bm(IX, IY+1, IZ)} + \\ \vec{dX1*dY1*dZ0*Bm(IX, IY, IZ+1)} + \\ \vec{dX1*dY0*dZ0*Bm(IX, IY+1, IZ+1)} + \\ \vec{dX0*dY1*dZ0*Bm(IX+1, IY, IZ+1)} + \\ \vec{dX0*dY0*dZ1*Bm(IX+1, IY+1, IZ)} + \\ \vec{dX0*dY0*dZ0*Bm(IX+1, IY+1, IZ+1)} \} / dX*dY*dZ \quad (9-1)$$

9.2.2 M A G I C磁場データの使用方法

N B I 3 D G F Nでは、入力データ中のGFUNコマンドによってG F U Nの計算制御を行っている。今回の結合作業では、入力データ中のパラメタで使用する磁場計算コードを選択出来るようになり、GFUNコマンド行でM A G I C磁場データに関する補助入力データを記述出来るように改良した。入力データにおけるGFUNコマンドの記述方法はFig. 9.6の通りである。

今回の結合作業において、M A G I Cで計算した磁場データを使用する手順を次のように定めた。

- (1) Fig. 9.6のIBGFUNに0を入力する

- (2) Fig. 9.6のGFUNコマンド行にFig. 9.7のMAGIC補助入力データを記述する。
- (3) 'END' を挿入してGFUNコマンド行の終わりを示す。

さて、Fig. 9.8のMAPファイルの例を元に、NBI3DGFNの入力データの中でMAGIC補助入力データをどのように記述すれば良いかを説明する。仮に、このMAPファイルのデータセット名を'J****.MAP.DATA'とする。Fig. 9.8のMAPファイルの内容から、磁場データの存在する領域の範囲と刻み幅、空間点の数を各座標毎に求めるとTable 9.1のようになる。この表からFig. 9.7の書式に従ってNBI3DGFNの入力データを作成するとFig. 9.9のようになる。また、MAGIC補助データに関する情報は結果出力リストにFig. 9.10の形式で出力される。

9.3 図化出力の改良：等高線プロット図の分割

9.3.1 概要

NBI3DGFNで計算された壁面上の熱負荷分布の図形出力は、Fig. 9.11のようにダクト内面の熱負荷分布を1つの平面に展開して描いており理解しにくい。今回の作業では、XC軸またはYC軸に平行な4面(Fig. 9.12)に分割して出力するように改良する。このとき、出力図のスケールは全て同一とし、縦横比が崩れないようにする。また、長さの単位系をmmからmへ変更する。

9.3.2 分割方法

Fig. 9.13のダクト断面図において、ダクトの内周に沿った座標系TCの原点をTCLEN(1)、ダクト断面の4隅の円弧の中点をTCLEN(2)～TCLEN(5)としたとき、これら5つの点によって区切られる領域をarea1～area5とする。ダクト展開図におけるこれらの領域はFig. 9.14のようになる。今回の改良では、area1とarea5を1つの領域としたarea1～area4の4つの領域について熱負荷分布等高線図を描くようとする。このとき、等高線図におけるTCメッシュ座標(ダクトの周長をNDHT個に分割した各々の点の座標、NDHTは入力データにて指定)は、各々の領域と平行な座標軸(XC座標及びYC座標)上の値に変換する。円弧上のTCメッシュ座標については、円弧を平面に展開したときのXCまたはYC座標の値とする。従って、TCメッシュ座標はダクトの周の領域によって、次の座標系に変換される。

[ダクト内周上の領域]		[変換後の座標系]
area1 : TCLEN(1) \leq TC < TCLEN(2)	\Rightarrow	Y P
area2 : TCLEN(2) \leq TC < TCLEN(3)	\Rightarrow	X P
area3 : TCLEN(3) \leq TC < TCLEN(4)	\Rightarrow	Y P
area4 : TCLEN(4) \leq TC < TCLEN(5)	\Rightarrow	X P
area5 : TCLEN(5) \leq TC	\Rightarrow	Y P

NBI3DGFNでは熱負荷分布等高線図のTCメッシュ座標値、熱負荷エネルギー値をそれぞれ配列T Cと配列F Z Tに持っている。分割された等高線図を描くためのデータはこれらのデータから作成し、新たに配列T C 2と配列F Z T 2に格納する。このとき、area1とarea5が同一の領域となるように、T C 2とF Z T 2のデータの格納順序をFig. 9.15, Fig. 9.16の通り変更する。その後、配列T C 2, F Z T 2を4つの領域に分割して図化出力ルーチンに渡す。

9.4 図化出力の改良：1次元図化出力の追加

9.4.1 1次元図化出力の方法

4つに分割出力した熱負荷分布等高線図の1つを縦に切ったときに出来る1次元図(Fig. 9.17)を出力する機能を付加する。1次元図の出力枚数、等高線図を切る座標、方向、枚数は任意であり、入力データで与えるものとする。

9.4.2 1次元図化出力に関するデータの与え方

N B I 3 D G F N 入力データの熱負荷分布計算データの最後に次のレコードを追加する。1次元図化出力はこれらのデータによって制御される。

(1)	NG1D
(2)	IXF IYF IZF BPOINT EPOINT NCUT MABIKI
	: (NG1Dレコード)

NG1D (I 5) : 出力グラフの枚数 (NG1D=0のとき1次元図化出力を行わない)
(2)のレコード数に等しい

IXF (I 5) : 1次元図化出力の対象となる熱負荷分布等高線図を決めるフラグ
(Fig. 9.18参照)

IYF (I 5) : 同上

IZF (I 5) : 等高線図を切る方向を決めるフラグ
IZF=0: XC, YC 座標軸に垂直に切る, IZF=1: ZC座標軸に垂直に切る

BPOINT (F 1 0. 0) : 切る位置の開始座標 (Fig. 9.17参照)

EPOINT (F 1 0. 0) : 切る位置の終了座標 (Fig. 9.17参照)

NCUT (I 5) : 切る本数 (Fig. 9.17参照)

MABIKI (I 5) : 1次元図上のセンターシンボルを
プロットするときの間引き数

9.4.3 1次元図における高さの求め方

熱負荷分布等高線図の高さ(熱負荷エネルギー)のデータはメッシュ間隔にしか存在しないので、任意座標の高さは隣接するメッシュの高さから線形内挿により求める。求めた高さは新たな配列F Z T 3に格納する。

XC=T(N)あるいはYC=T(N)で等高線図をZC軸に平行に切る場合の例をFig. 9.19に示す。この

図において、座標 $(T(N), ZC(IZ))$ における高さ $FZT3(IZ, N)$ は、隣接する 2 つの高さ $FZT2(IZ, IT)$ と $FZT2(IZ, IT+1)$ から式(9-2)で求められる。

$$FZT3(IZ, N) = \{ dT1 * FZT2(IZ, IT) + dT0 * FZT2(IZ, IT+1) \} / dT \quad (9-2)$$

$ZC = Z(N)$ で等高線図を ZC 軸に垂直に切る場合の例を Fig. 9. 20 に示す。この図において、座標 $(Z(N), T02(IT))$ における高さ $FZT3(IT, N)$ は、隣接する 2 つの高さ $FZT2(IZ, IT)$ と $FZT2(IZ+1, IT)$ から式(9-3)で求められる。

$$FZT3(IT, N) = \{ dZ1 * FZT2(IZ, IT) + dZ0 * FZT2(IZ+1, IT) \} / dZ \quad (9-3)$$

その後、配列 $FZT3$ は XC 座標、 YC 座標、 ZC 座標の何れか（1 次元化される等高線図によって決まる）と共に新たに作成した 1 次元図化出力ルーチンに渡される。

9.4.4 1 次元図化出力の形式

更に、1 次元図化出力の規則を次のように定めた。

- 1 次元図化出力に関する入力データ (9.4.2 の(2)) の 1 レコードに対して 1 枚の 1 次元図を出力する。
- 1 枚の 1 次元図で切る座標が複数個指定された場合、グラフの線種とセンターシンボルをそれぞれ異なったものにする。そのときのグラフの線種とセンタシンボルの使用する順序は次の通りである。

線種 : 以下の線種が繰り返し使用される

実線	(—)
太線	(—)
点線	(...)
破線	(--)
1 点鎖線	(---)

センタシンボル : ARGUS のセンタシンボルコードの
コード番号 101 ~ 120 の 20 個が順番に使用される

- 1 次元図化出力の右端に、切った位置の座標値がセンターシンボルと共に出力されるようとする。
- 1 枚の 1 次元図で切る本数は最大 20 とする。
- センターシンボルの間引き間隔は入力データで指定する。
- 切る位置が指定領域を越えた場合は以下のメッセージを出力し、切った面の高さを 0 にす

る。

- [メッセージ] CUTTING POINT IS OVER THE AREA : POINT = tc or zc
- ・ フラグIXF, IYF の入力方法に誤りがあった場合は以下のメッセージを出力し、1次元図化出力を行わないようにする。
- [メッセージ] ERROR IN GRAPH1D DATA : IXF IYF = ixf iyf

9.5 実行と評価

N B I 3 D G F Nでの熱負荷分布計算の結果をG F U N磁場データを使用した場合とM A G I C磁場データを使用した場合で比較する。比較は、N B I 3 D G F Nから出力されるグラフをもとに行う。

G F U N磁場データを使用する場合のN B I 3 D G F Nの入力データの例をFig. 9.21 に、この入力データに対するグラフをFig. 9.23 ~Fig. 9.25 に示す。Fig. 9.23は、ダクト展開図における熱負荷分布等高線図である。Fig. 9.24, Fig. 9.25は今回の作業で追加された熱負荷分布等高線図を分割出力したものと、1次元図である。また、M A G I C磁場データを使用する場合のN B I 3 D G F Nの入力データをFig. 9.22 に、この入力データに対するグラフをFig. 9.26 ~Fig. 9.28 に示す。

今回使用したM A G I C磁場データの磁場の向きがG F U N磁場データと逆になっていたので、イオンビームの曲がる向きが逆になり、ダクト上の熱負荷分布はFig. 9.23 とFig. 9.26 のように上下対象となって現れた。両者の磁場の様子は、Fig. 9.29 とFig. 9.30 で確認できる。

両者の図化出力を比較した結果、熱負荷分布図に僅かな違いがあるものの、許容の範囲に収まっていることが分かった。熱負荷分布の差は磁場計算の手法の違いによるものと思われる。

Table 9.1 Region calculated by MAGIC and the number of calculation steps

	最小値 WSTART	最大値 WEND	刻み幅 WD	ステップ数 IWSTEP
X	1.0000D+01	1.3500D+01	1.0000D-01	3 6
Y	-8.0000D-01	8.0000D-01	1.0000D-01	1 7
Z	-3.0000D-01	3.0000D-01	1.0000D-01	7

(単位m)

$$\text{※ } \text{IWSTEP} = (\text{WEND} - \text{WSTART}) / \text{WD} + 1$$

Table 9.2 Modified subroutines

SUBROUTINE	機能	修正内容
B G C A L C	GFUN磁場データの制御	<ul style="list-style-type: none"> IFMAGCの値が1のときに、サブルーチンNAISO を呼び出す処理を追加
B V	磁場ベクトル計算の制御	<ul style="list-style-type: none"> MAGIC データを使用した場合のサブルーチンPVCTR を呼び出す処理を追加
M A I N G F	GFUNのメインルーチン	<ul style="list-style-type: none"> 入力データのGFUN磁場データで' MAGC' が指定された場合の処理を追加 サブルーチンRDMAGCの呼び出しを追加
T R A C	イオン軌道の計算	<ul style="list-style-type: none"> IFMAGCの値が1のときに、サブルーチンNAISO を呼び出す処理を追加
B I	熱負荷分布計算を制御する	<ul style="list-style-type: none"> 熱負荷分布図分割出力の機能を追加 熱負荷分布1次元図化出力の機能を追加 熱負荷分布1次元図化出力のデータ読み込み機能を追加
F L O A D	熱負荷分布の図形出力用データの作成を制御する	<ul style="list-style-type: none"> 作業配列Aの大きさを拡張
P W C T R	熱負荷分布等高線図の出力	<ul style="list-style-type: none"> 長さの単位の変更 [MM] \Rightarrow [M]

Table 9.3 Created subroutines

SUBROUTINE	機能
N A I S O	任意の座標の磁場をMAGIC 磁場データから線形内挿より求める
R D M A G C	MAGIC 磁場データの読み込み
C O N T R 4	熱負荷分布等高線図の分割出力を制御する
P W C T R 2	熱負荷分布等高線図の分割出力
D I V T C	配列T Cを分離出力する領域に従って分割する
T Z 2 X Y Z	熱負荷分布等高線図の分割出力のためのデータ作成
C U T T C	熱負荷分布等高線図をT C軸に垂直に切った場合の1次元図のデータを作成する
C U T Z C	熱負荷分布等高線図をZ C軸に垂直に切った場合の1次元図のデータを作成する
G R A P 1 D	1次元図化出力を制御する
P W C T R 3	1次元図化出力

Table 9.4 Added common variables

COMMON変数	用途	使用モジュール
/LMAGIC/XSTART, YSTART, ZSTART, DX, DY, DZ, IXSTEP, IYSTEP, IZSTEP, XEND, YEND, ZEND	MAGIC 磁場データに関する情報	N A I S O B G C A L C R D M A G C
/MAGICF/IFMAGC	MAGIC 磁場データを使用するかどうかのフラグ	M G C A L C, B V M A I N G F, T R A C
/BMAGIC/BXM(IBMAXP), BYM(IBMAXP), BZM(IBMAXP)	MAGIC 磁場ベクトルデータ	N A I S O R D M A G C B G C A L C

Table 9.5 Added include parameters

イクル名	内 容	用 途	使用モジュール
PARAM	PARAMETER (IBMAXP=50*20*10)	MAGIC 磁場ベクトルデータの最大数 COMMON変数BXM, BYM, BZM の大きさ	B G C A L C R D M A G C T R A C
	PARAMETER (NCUTMX=20)	1次元図化出力時の最大カット数	B I
	PARAMETER (MAX1DG=20)	1次元図化出力図の最大出力枚数	F L O A D

Table 9.6 Extension of work array

アドレス	大きさ	用 途
ID(23)～ID(24)-1	N D H T	配列T C 2 : T C メッシュ座標をX C 座標またはY C 座標に変換した値
ID(24)～ID(25)-1	N D H T * N D H Z	配列F Z T 2 : 分割した熱負荷分布等高線図の高さ (熱負荷エネルギー)
ID(25)～ID(26)-1	N D H Z * N C U T M X あるいは N D H T * N C U T M X	配列F Z T 3 : 1次元図化出力データ (熱負荷エネルギー)

Table 9.7 Modified input data

ファイル	UNIT	データセット名	内 容
NBI3DGFN入力データ	FT05	任 意	<ul style="list-style-type: none"> • G F U N コマンド行にMAGIC 補助入力データを記述可能にした • 1次元図化出力データ行の追加
MAGIC 磁場データ	FT51	NBI3DGFN入力データの中で指定する	<ul style="list-style-type: none"> • M A G I C 磁場データファイル (M A G I C 磁場データを指定した場合のみ使用する) • MAGIC のM A P 出力ファイル

MAIN

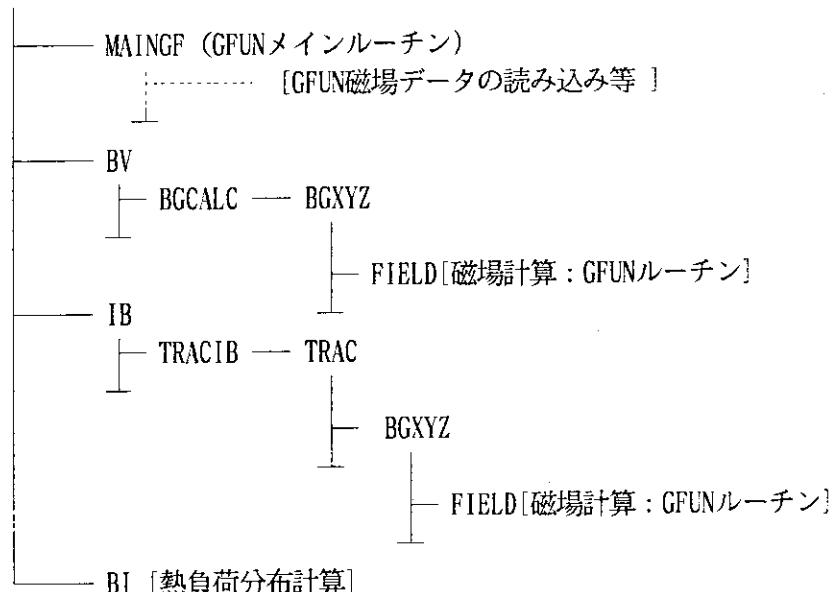


Fig. 9.1 Calculation flow of NBI3DGFN

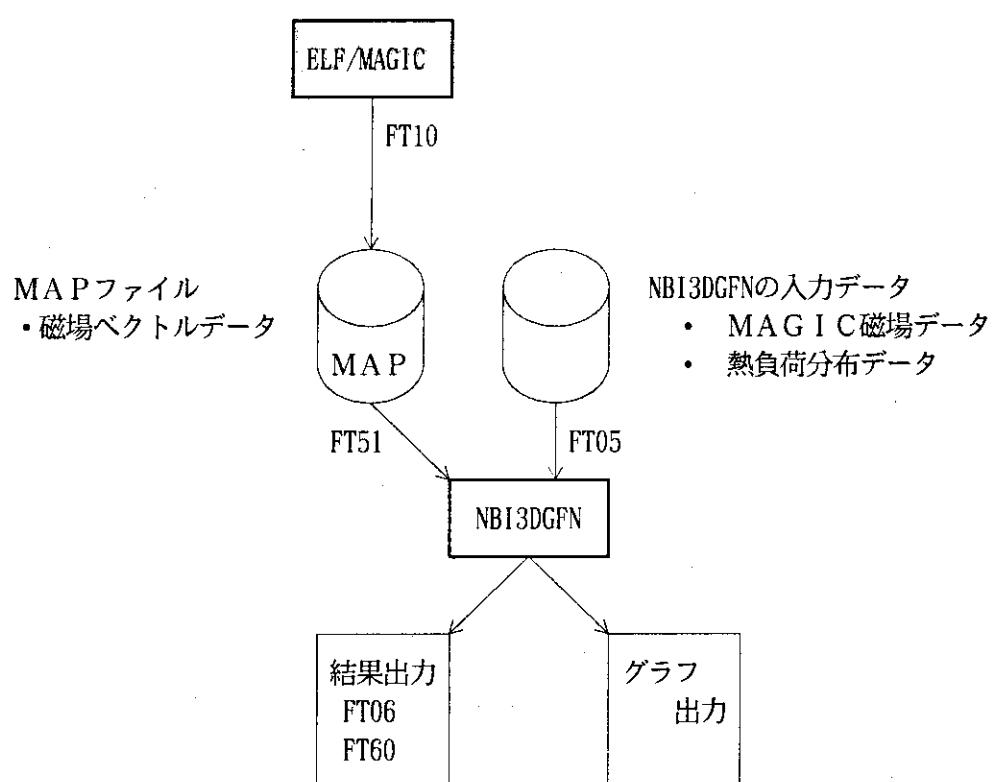


Fig. 9.2 Data flow of NBI3DGFN for the MAP data

MAIN

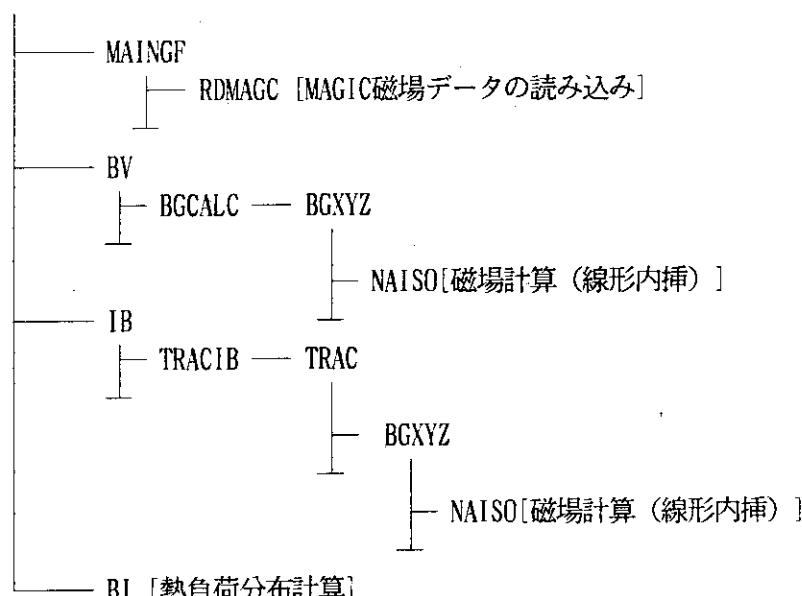
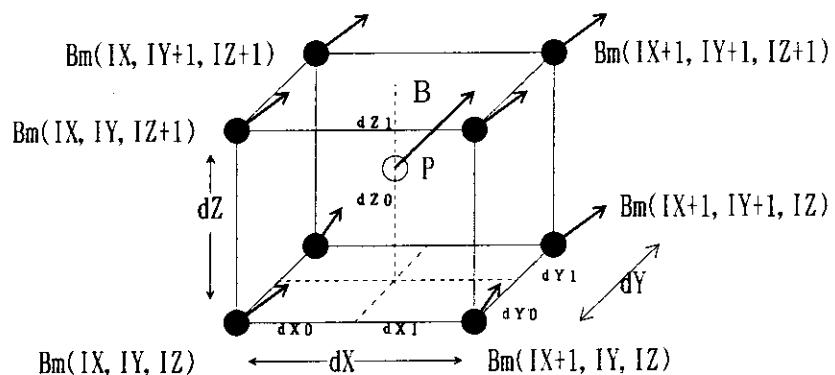


Fig. 9.3 Calculation flow of NBI3DGFN for the MAP data

VG	v n	v x	v y	v z	v 1	v 2	v 3
----	-----	-----	-----	-----	-----	-----	-----

- VG : 空間点の磁場ベクトルを表す 'VG'
- v n : 空間点ベクトルの番号、または空間点の番号
- v x : 空間点ベクトルのX座標 (単位m)
- v y : 空間点ベクトルのY座標 (単位m)
- v z : 空間点ベクトルのZ座標 (単位m)
- v 1 : 空間点ベクトルのX成分 (単位T)
- v 2 : 空間点ベクトルのY成分 (単位T)
- v 3 : 空間点ベクトルのZ成分 (単位T)

Fig. 9.4 Data format of the VG card



Z
 Y
 X
 トーラス座標系

- : 任意の座標 P
- : VGカードにおける空間点
- B : Pにおける磁場ベクトル
- Bm : 空間点磁場ベクトル
- dX, dY, dZ : 空間点のきざみ幅
- dX0, dY0, dZ0, dX1, dY1, dZ1 : Pと隣接する空間点との距離の成分

Fig. 9.5 Magnetic field vector at $P(x, y, z)$ and magnetic field vectors around P

(1) IBGFUN (I 5)	1 2 3 4 5	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td></tr></table>									
(2) GFUNコマンド	1 2 3 4 5 6 7 8	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>									
(3) 'END' (A 4)	1 2 3 4	<table border="1"><tr><td></td><td></td><td></td><td></td></tr></table>									

IBGFUN: = 0 GFUN磁場を用いない
 > 0 GFUN磁場を用い、DRAWコマンドで図化出力
 < 0 GFUN磁場を用い、図化出力しない
 ($\neq 0$ のとき、入力データ中のトロイダル及び、ポロイダル磁場データは
 DUMMYとなる。)
 GFUNコマンド: GFUNコマンドを羅列する (IBGFUN $\neq 0$ のとき有効)
 'END' : GFUNコマンドの終了を示す (IBGFUN=0のときも記述すること)

Fig. 9.6 Data format of GFUN command in input data

	v n	v x	v y	v z	v 1	v 2	v 3
VG	2	1.0000D+01	-8.0000D-01	-3.0000D-01	-8.8838E-04	2.6534E-03	-1.1588E-04
VG	3	1.0100D+01	-8.0000D-01	-3.0000D-01	-8.2209E-04	3.0063E-03	-1.1210E-04
VG	\$	\$	-8.0000D-01	-3.0000D-01	\$	\$	\$
VG	37	1.3500D+01	-8.0000D-01	-3.0000D-01	2.9994E-02	2.0861E-02	7.4990E-03
VG	38	1.0000D+01	-7.0000D-01	-3.0000D-01	-5.9399E-04	2.5488E-03	-7.7477E-05
VG	\$	\$	-7.0000D-01	-3.0000D-01	\$	\$	\$
VG	\$	\$	\$	-3.0000D-01	\$	\$	\$
VG	613	1.3500D+01	8.0000D-01	-3.0000D-01	2.9994E-02	2.0861E-02	7.4990E-03
VG	614	1.0000D+01	-8.0000D-01	-2.0000D-01	\$	\$	\$
VG	\$	\$	\$	\$	\$	\$	\$
VG	4285	1.3500D+01	8.0000D-01	3.0000D-01	2.9994E-02	2.0861E-02	7.4990E-03

Fig. 9.8 Example of the MAP data

0	IBGFUN =0
MAGC	MAGIC補助入力データ
J****.MAP.DATA	
X 10000.0 100.0 36	
Y -800.0 100.0 17	
Z -300.0 100.0 7	
END	'END' コマンド

Fig. 9.9 Example of GFUN command lines in order to use MAP data

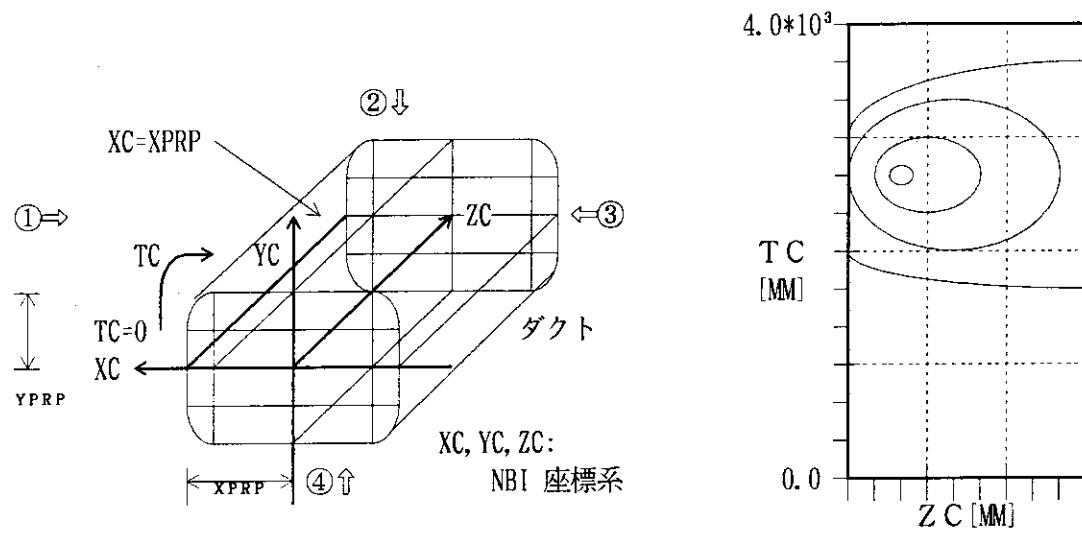
```
#####
### E L F / M A G I C M A G N E T I C U S E ###
#####
IBGFUN = 0 GFUN IS NOT ACTIVE.
```

MAP FILE NAME IS J****.MAP.DATA

XYZ START(MM)	D(MM)	ISTEP
X 10000.	100.	36
Y -800.	100.	17
X -300.	100.	7

READ 4284 VG CARDS FROM J****.MAP.DATA

Fig. 9.10 Example of outlist in the case of using the MAP data



①～④ : XC軸またはYC軸に平行な4面

Fig. 9.11 Structure of the duct and contour of energy on the duct

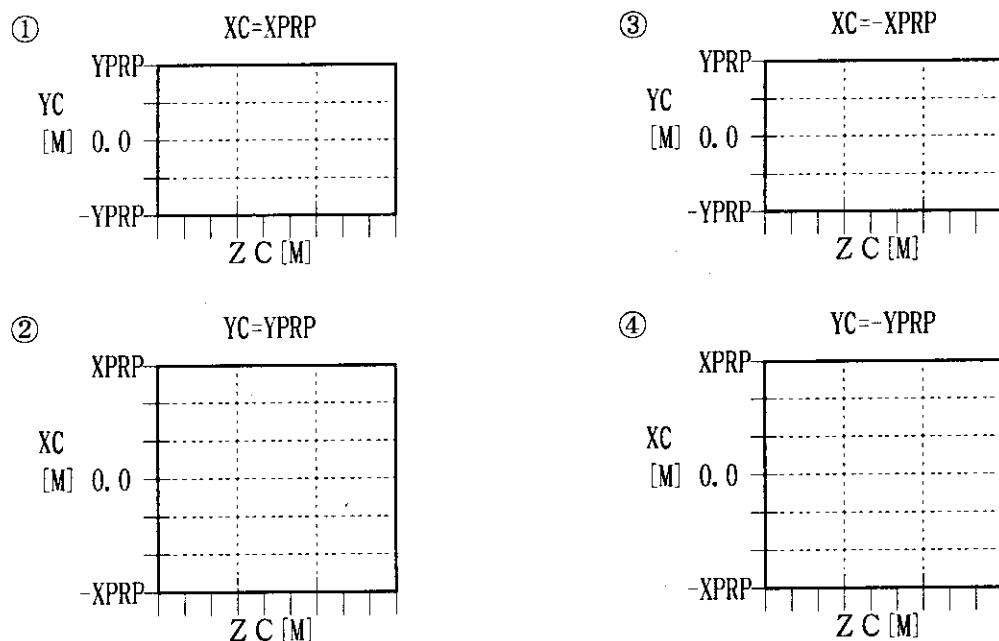


Fig. 9.12 Four separated contours

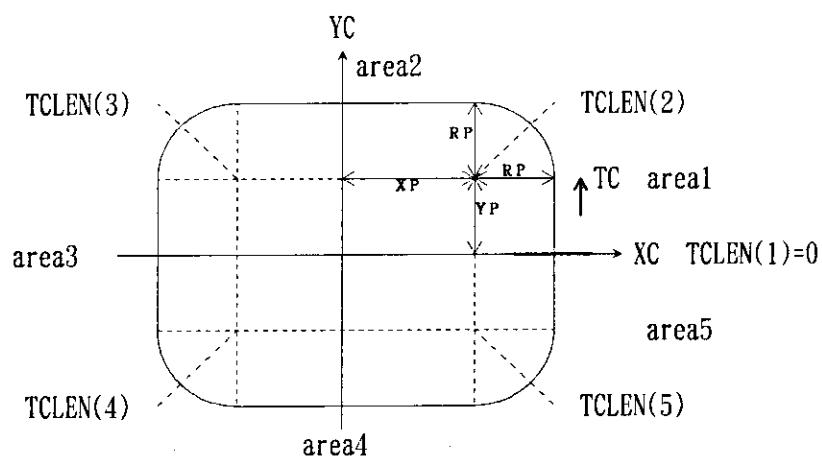


Fig. 9.13 Cross section of the duct and five divided areas

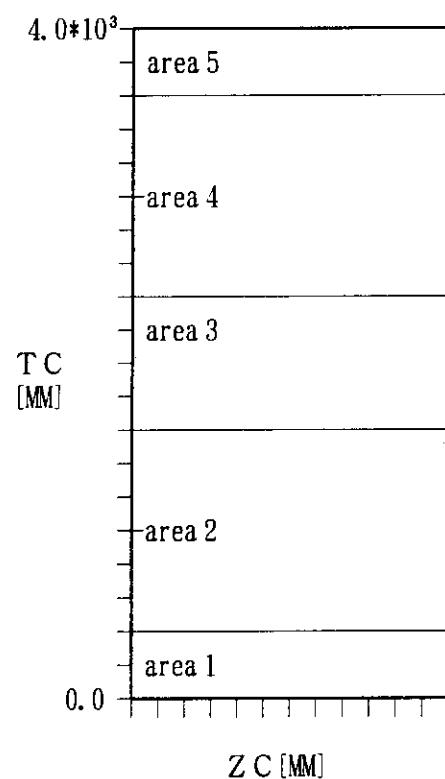
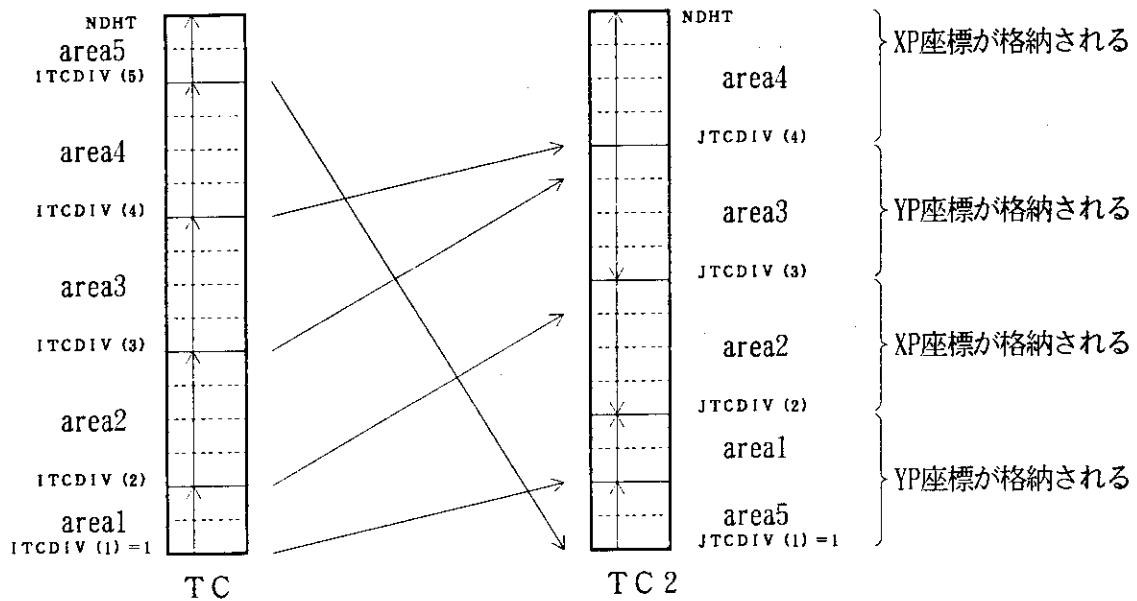


Fig. 9.14 Five areas corresponding to the inner wall of the duct



※ ITCDIVとJTCDIVは配列TCとTC2 のアドレス

Fig. 9.15 Relation between TC and TC2

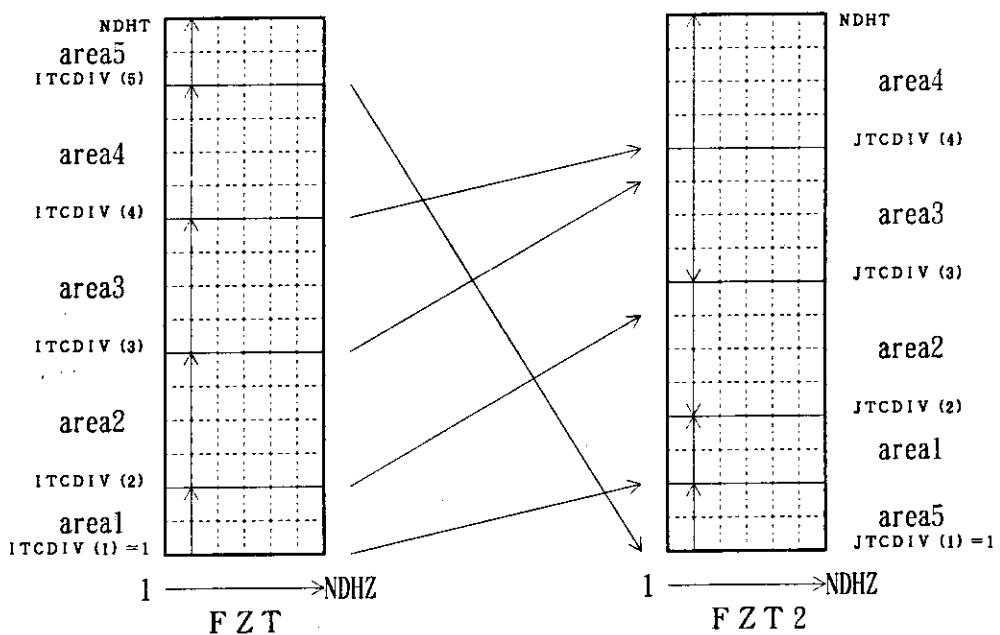
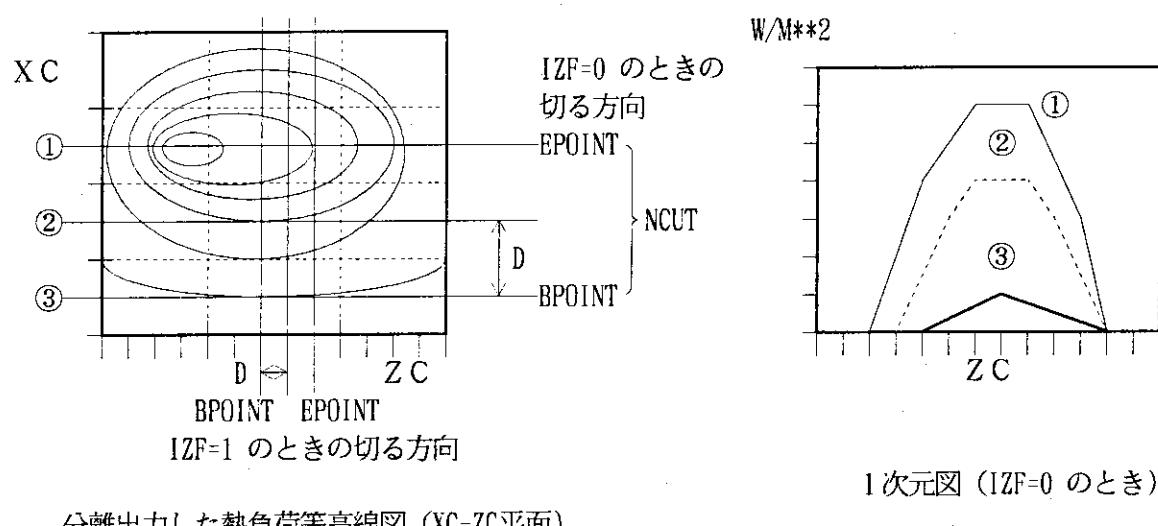


Fig. 9.16 Relation between FZT and FZT2



BPOINT : 切る位置の開始座標
 EPOINT : 切る位置の終了座標
 D : 切る間隔
 IZF : 切る方向
 NCUT : 切る枚数

Fig. 9.17 Relation between contour and one-dimensional graph

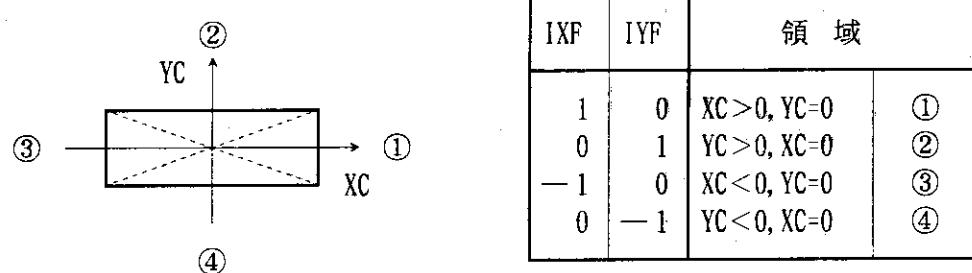


Fig. 9.18 Relation between printed one-dimensional graph and parameter IXF, IYF

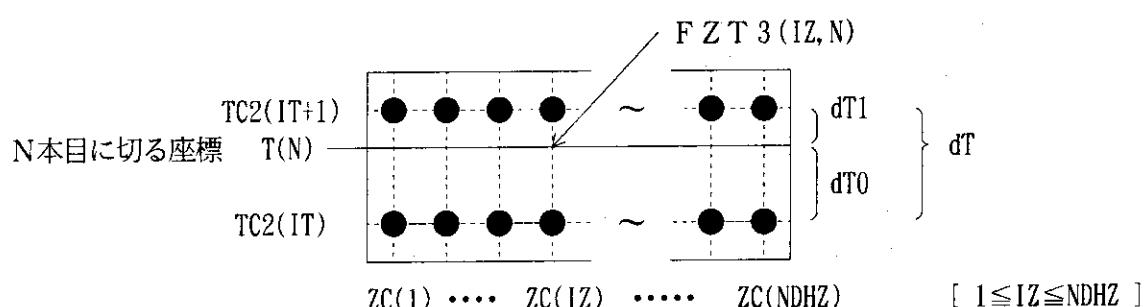


Fig. 9.19 Interpolation in the IZ direction

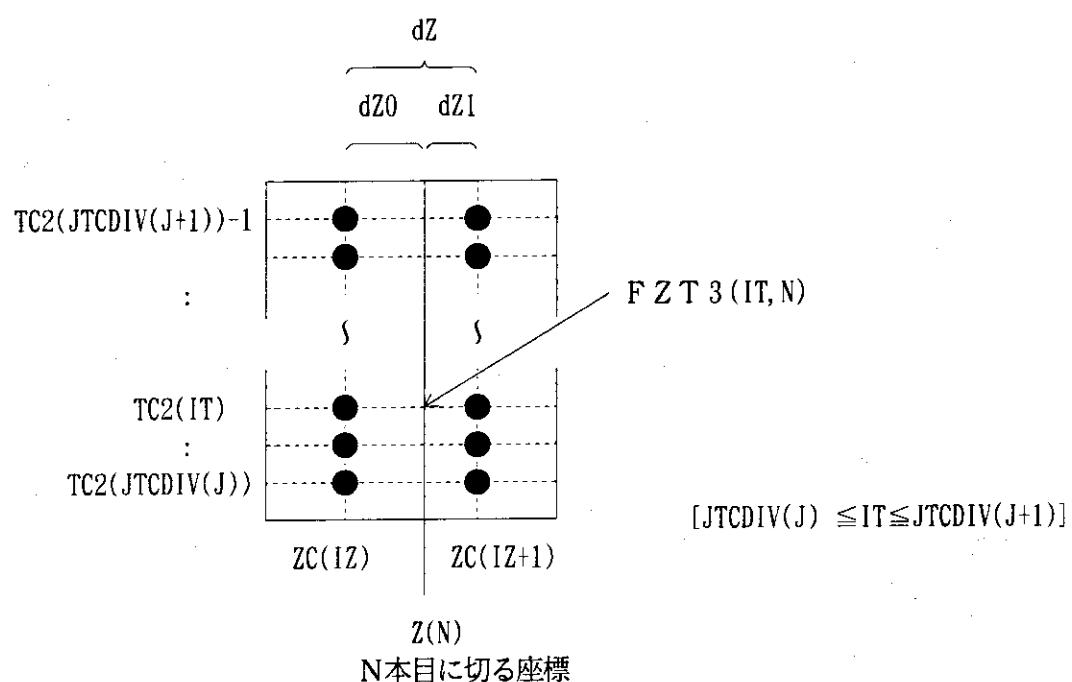


Fig. 9.20 Interpolation in the IT direction

JT-60U N-BBI BENDING COIL : CIRCULAR COILS (100KAT) : 3MRAD : FL 23, 66M (1)メインタイトル
 JT-60U TOROIDAL FIELD } (2)例 磁場データ
 18 2.375 0.405 0.29800 0.0 }
 3.320
 JT-60U POLOIDAL FIELD } (3)例 磁場データ
 1 3.4000 0.0000 1.0 0.0 }
 1
 CLEAR
 DRAW MATB=0, SHAP=0, BND=1, SYMM=1, J= 0, OB=0, X1=339.5, Y1=-0.5
 A=1, B=1, 0, H1=0, 0, R=339.5
 ++
 DRAW MATB=0, SHAP=0, BND=1, SYMM=1, J=1000.0, X1=95.0, Y1=95.0
 A=10, 0, B=10, 0, H1=0, 0, R=95.0, XCEN=1230.0, ZCEN=261.3
 ++
 DRAW MATB=0, SHAP=0, BND=1, SYMM=1, J=1000.0, X1=95.0, Y1=-105.0
 A=10, 0, B=10, 0, H1=0, 0, R=95.0, XCEN=1230.0, ZCEN=261.3
 ++
 RBCD PLANE=ZX, XMIN=0.0, XMAX=2000.0, ZMIN=0.0, ZMAX=2000.0
 GETB X=1230, Y=0, Z=261.3, DY=5, NY=31, ERASE=YES, COMP=IMOD
 BND
 27660.0 0.00 2613.00 4000.0 0.0 2613.0 (5)座標系定義
 2
 0 0 1 20 20 0
 1000.0 -1000.0 1000.0 -1000.0 15360.0 15360.0 } (6)ベクトル図出力
 0 0 1 16 0 24
 1000.0 -1000.0 0.0 0.0 17360.0 14360.0
 0
 300 750.0 250.0 0.0 14360.0 17360.0 } (7)再電離イオン軌道
 (熱負荷分布軌道)
 1
 40 20 0 300 18.75 12.5 50.0 14361.0 50.0 } 热負荷分布用
 1 1
 2 1 0.0 0.0 23660.0 500000.0 1250.0 } 軌道メッシュ等 } (8)热
 負荷
 分布
 計算
 データ
 2500.0 1.0 0.0 0.0 2.5B-05 1.9B-05 30 80 } 热負荷メッシュ等
 14910.0 16410.0 25 6 }
 8
 1.0B+04 9.2B-17
 2.0B+04 1.3B-16
 3.0B+04 1.5B-16
 5.0B+04 1.5B-16
 7.0B+04 1.3B-16
 1.0B+05 1.1B-16
 2.0B+05 7.0B-17
 1.2B+06 1.0B-18
 3 0.0 100.0 2000.0 0.0 100.0 2000.0 } 等高線
 2
 0 1 0 400.0 500.0 3 10 } 1次元図化
 0 1 1 15400.0 15900.0 4 5 } 出力データ
 JT-60U BBM PROFILE (HENKOU)
 1 0 2.0
 1135.0 0.0 0.0 0.0 24000.0
 1.0 23660.0 23660.0 0.172 0.172
 10 5 5 5 5 5 5 5 5 } (9)热負荷分布
 40.0 45.0 } 計算データ (II)
 0 450.0 0.0 14360.0 350.0 250.0 0.0
 10 5 0 0
 2 0 0 0
 1 0.0 0.0 14360.0 750.0 250.0 0.0
 25 12 0 0
 1 0.0 0.0 17360.0 750.0 250.0 0.0
 25 12 0 0

Fig. 9.21 Input data for GFUN

JT-60U N-NBI BENDING COIL : CIRCULAR COILS (100KAT) : 3MRAD : FL 23.66M (1)メインタイトル
 JT-60U TOROIDAL FIELD (2)トロイド 磁場データ
 18 2.375 0.405 0.29800 0.0
 3.320 }
 JT-60U POLOIDAL FIELD (3)ポリオイド 磁場データ
 1 3.4000 0.0000 1.0 0.0
 0 }
 MAGC (4)MAGC補助
 J9050.MAP,DATA 入力データ
 X 10000.0 100.0 36
 Y -800.0 100.0 17
 Z 2313.0 100.0 7
 END (5)座標系定義
 27660.0 0.00 2613.00 4000.0 0.0 2613.0
 2 }
 0 0 1 20 20 0 1000.0 -1000.0 1000.0 -1000.0 15360.0 15360.0 (6)ベクトル図出力
 0 0 1 16 0 24 1001.0 -1000.0 0.0 0.0 17360.0 14360.0 }
 0 }
 300 750.0 250.0 0.0 14360.0 17360.0 (7)再電離イオン軌道
 (熱負荷分布軌道)
 1 40 20 0 300 18.75 12.5 50.0 14361.0 50.0 } 热負荷分布用
 1 1 軌道メッシュ 等
 2 1 0.0 0.0 23660.0 500000.0 1250.0 } (8) 热
 2500.0 1.0 0.0 0.0 2.5E-05 1.9E-05 30 80 负荷分布
 14910.0 16410.0 25 6 等
 8 }
 1.0E+04 9.2E-17
 2.0E+04 1.3E-16
 3.0E+04 1.5E-16
 5.0E+04 1.5E-16
 7.0E+04 1.3E-16
 1.0E+05 1.1E-16
 2.0E+05 7.0E-17
 1.2E+06 1.0E-18 } 断面積
 3 0.0 100.0 2000.0 0.0 100.0 2000.0 等高線
 2 }
 0 1 0 400.0 500.0 3 10 0 1 1 15400.0 15900.0 4 5 } 1次元化
 JT-60U BEAM PROFILE(HENKOU) 出力データ
 1 0 2.0
 1135.0 0.0 0.0 0.0 24000.0
 1.0 23660.0 23660.0 0.172 0.172
 10 5 5 5 5 5 5 5 5 5 5 5 40.0 45.0 } (9)熱負荷分布
 0 450.0 0.0 14360.0 350.0 250.0 0.0 計算データ (II)
 10 5 0 0
 2 0 0 0
 1 0.0 0.0 14360.0 750.0 250.0 0.0
 25 12 0 0 0
 1 0.0 0.0 17360.0 750.0 250.0 0.0
 25 12 0 0 0 }

Fig. 9.22 Input data for MAGIC MAP data

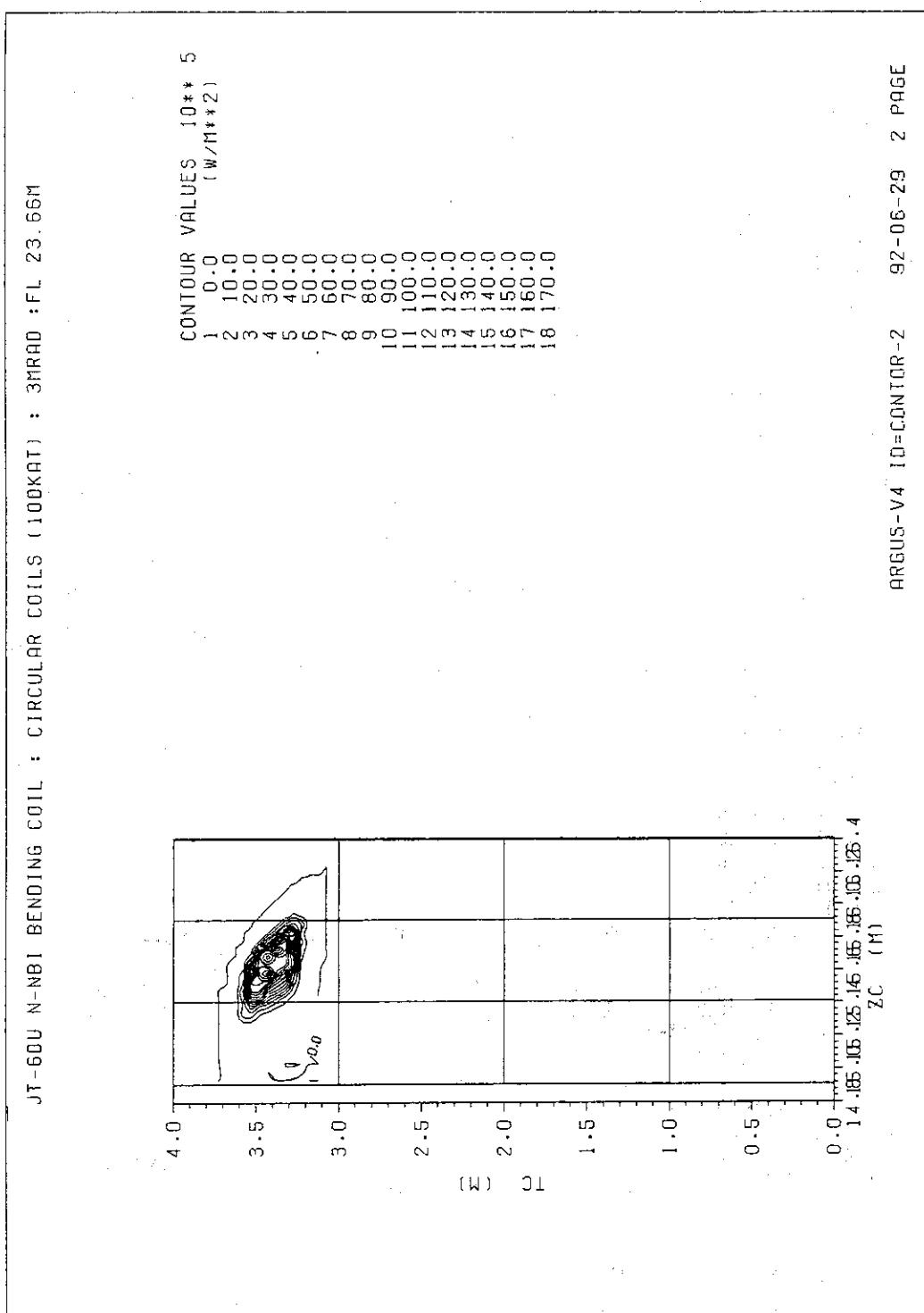


Fig. 9.23 Energy on the inner wall of the duct

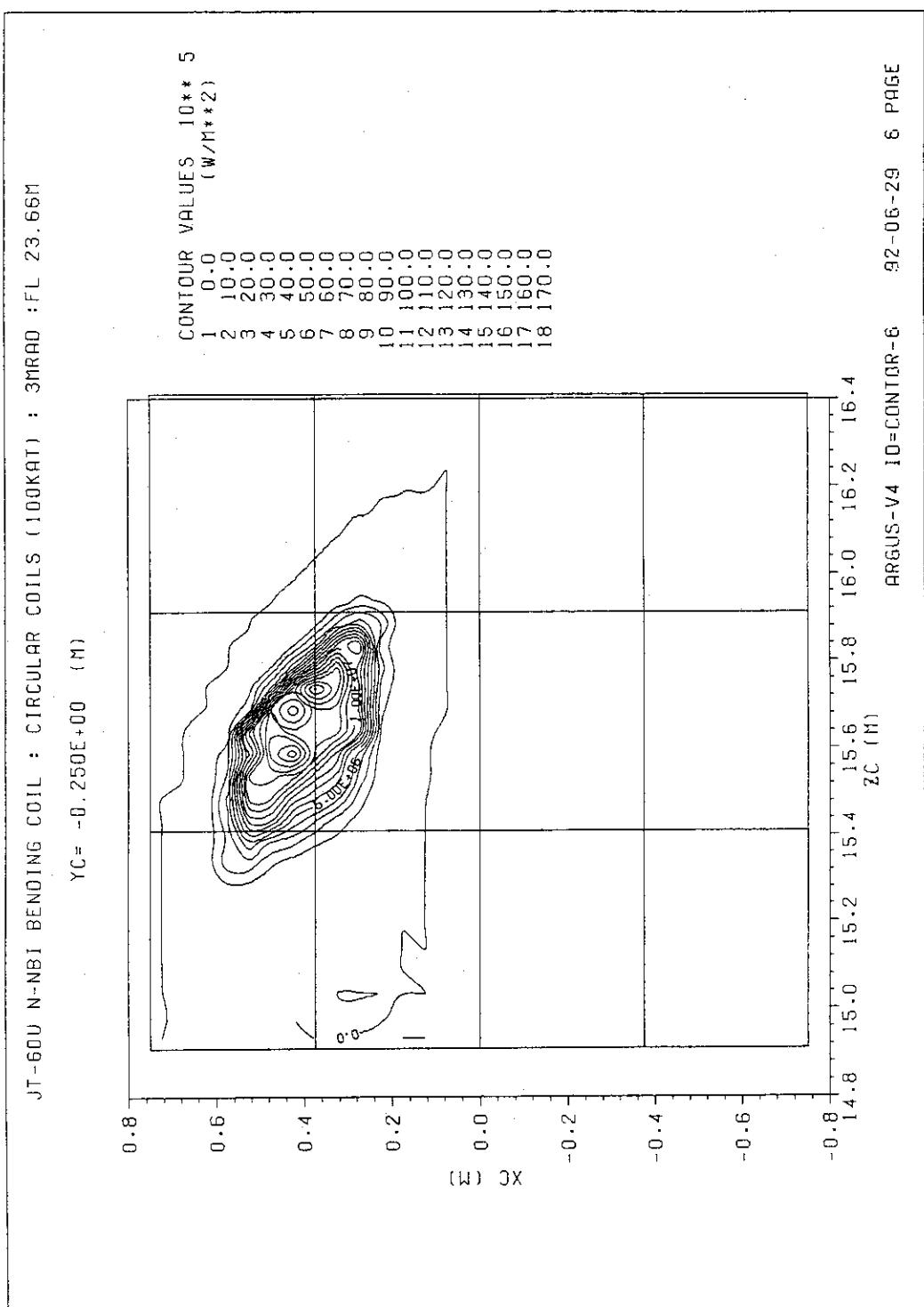
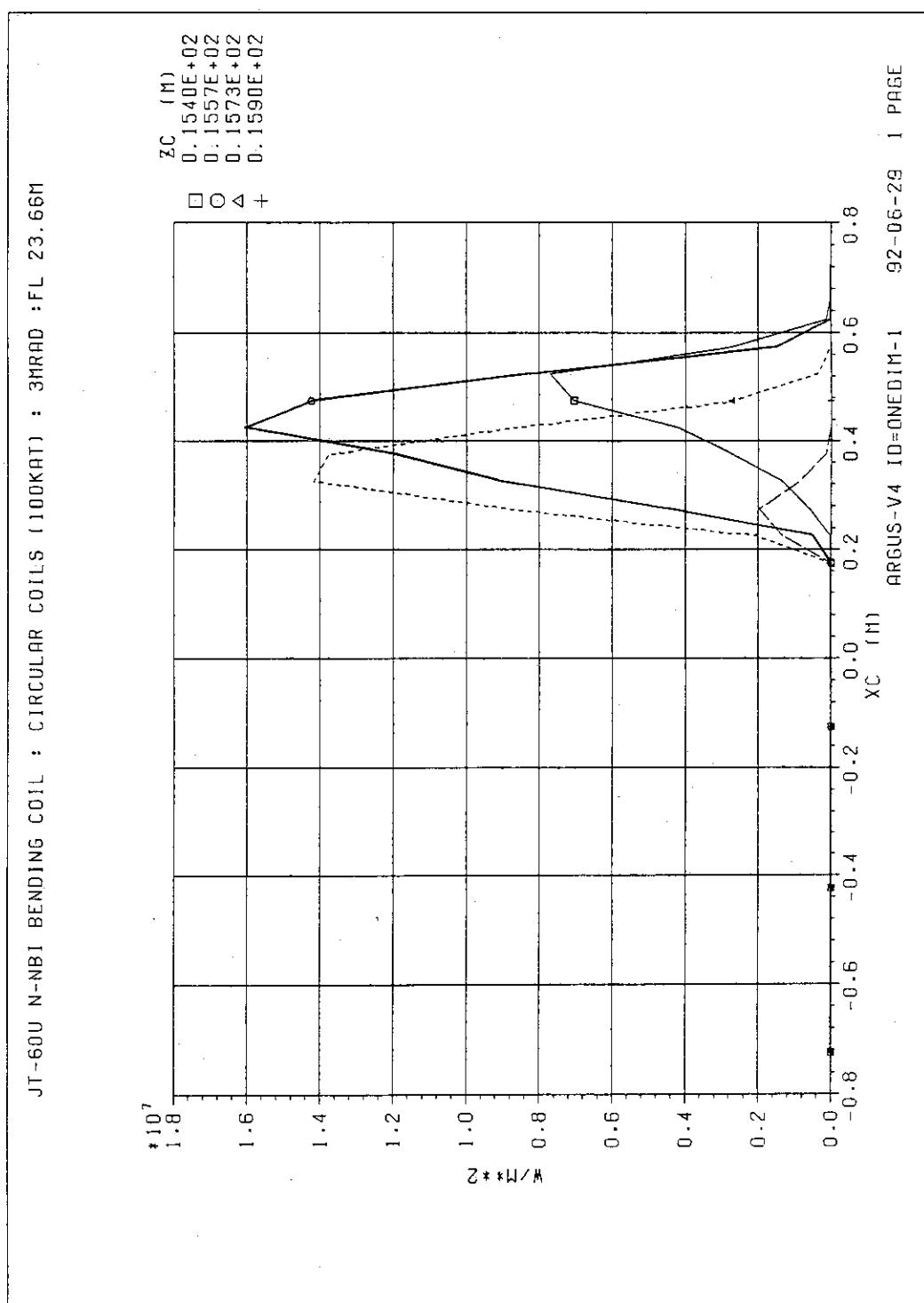


Fig. 9.24 Energy on the inner wall of the duct (area 4)

Fig. 9.25 Energy on the inner wall of the duct ($ZC = 0.1540E + 02 \sim 0.1590E + 02$ (M))

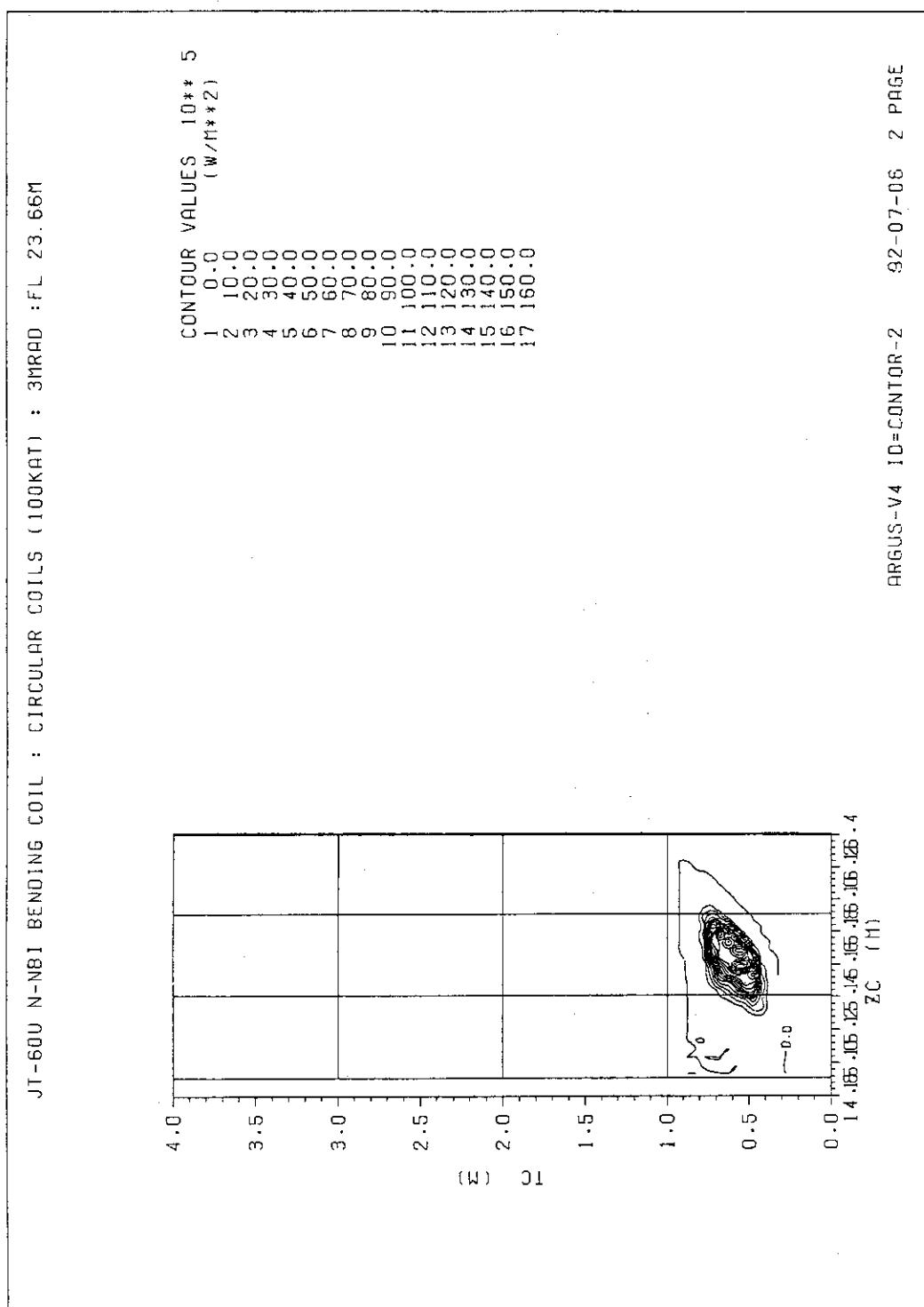


Fig. 9.26 Energy on the inner wall of the duct

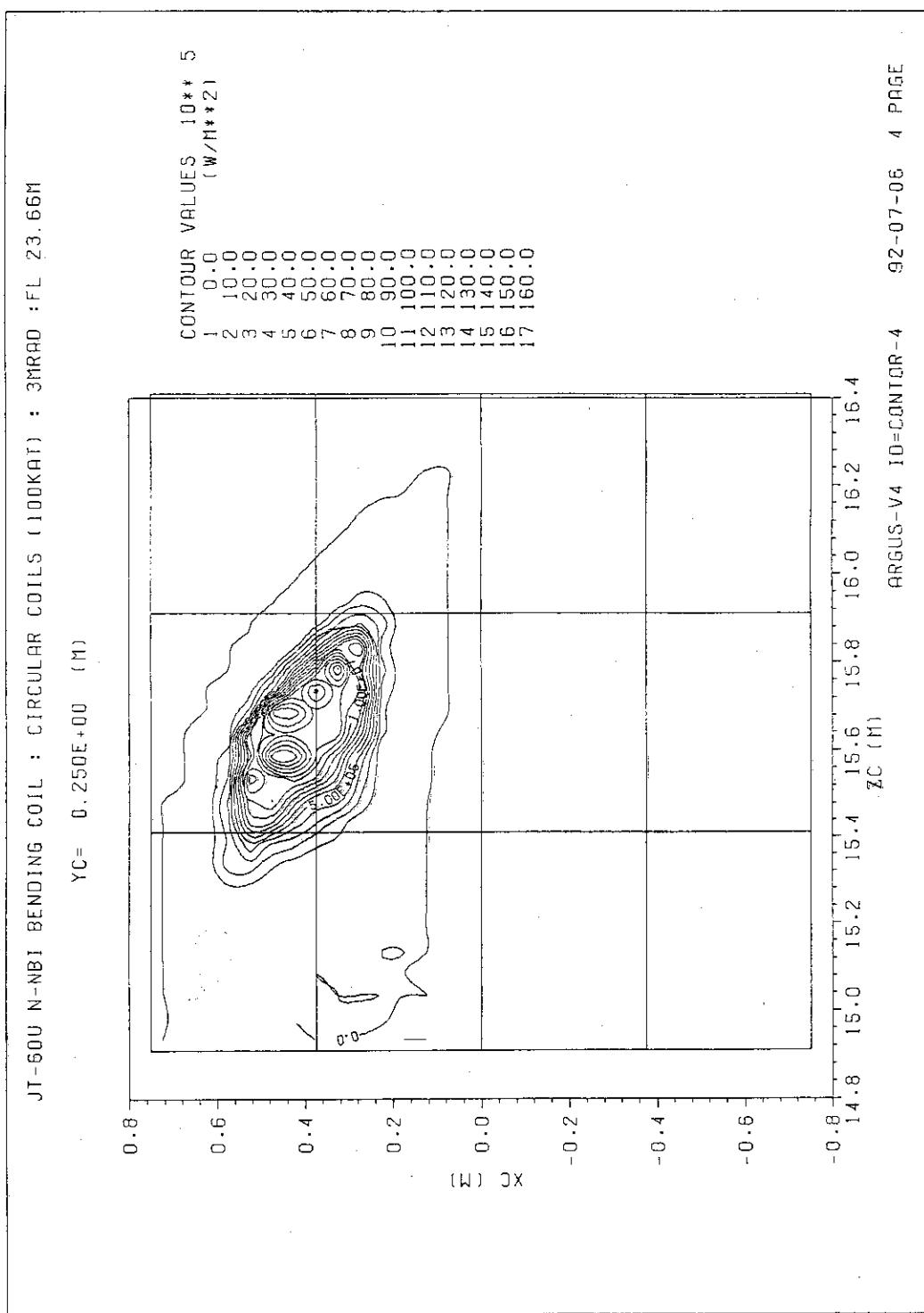


Fig. 9.27 Energy on the inner wall of the duct (area 1)

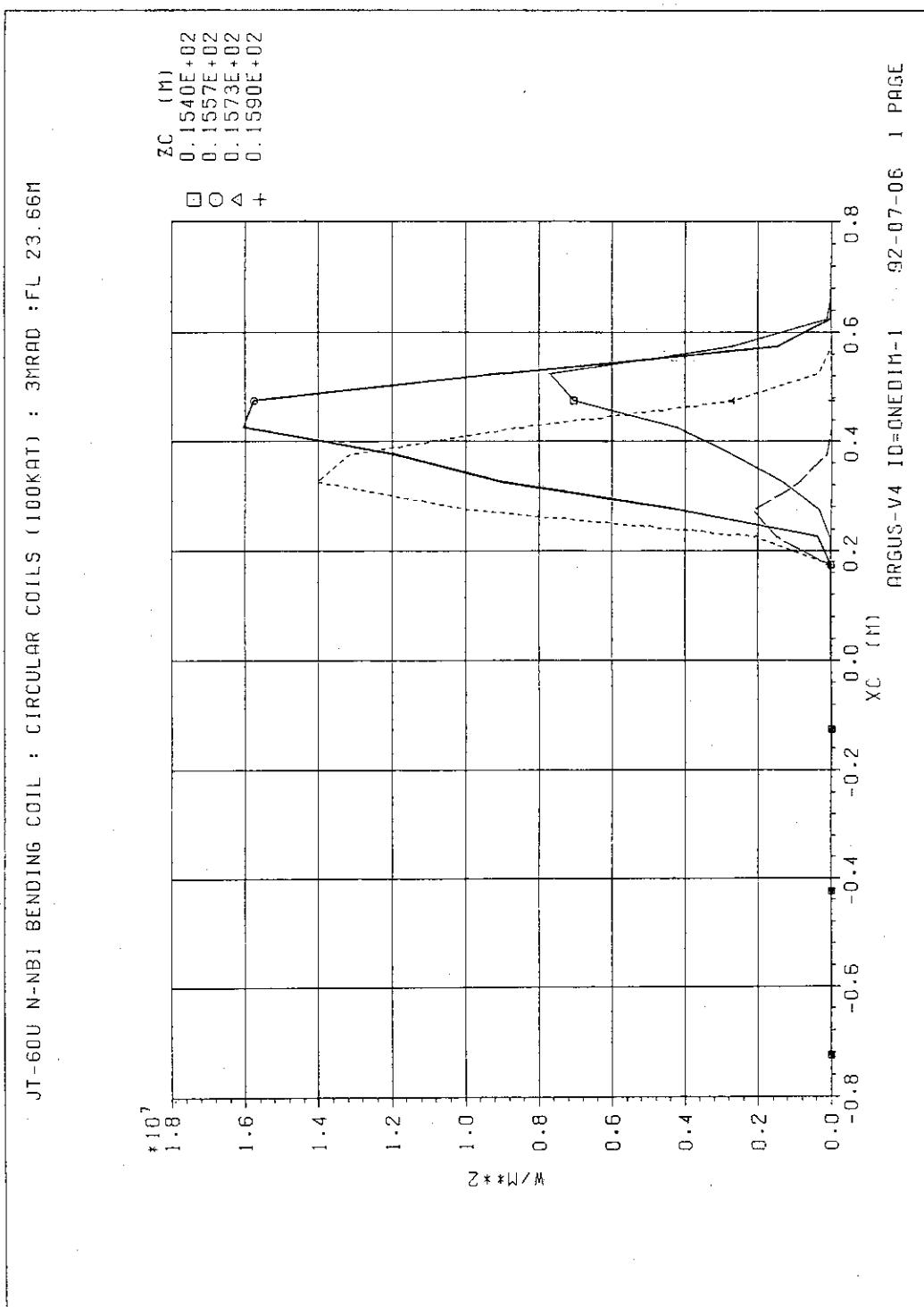


FIG. 9.28 Energy on the inner wall of the duct ($ZC = 0.154OE + 0.02 \sim 0.159OE + 0.02$ (M))

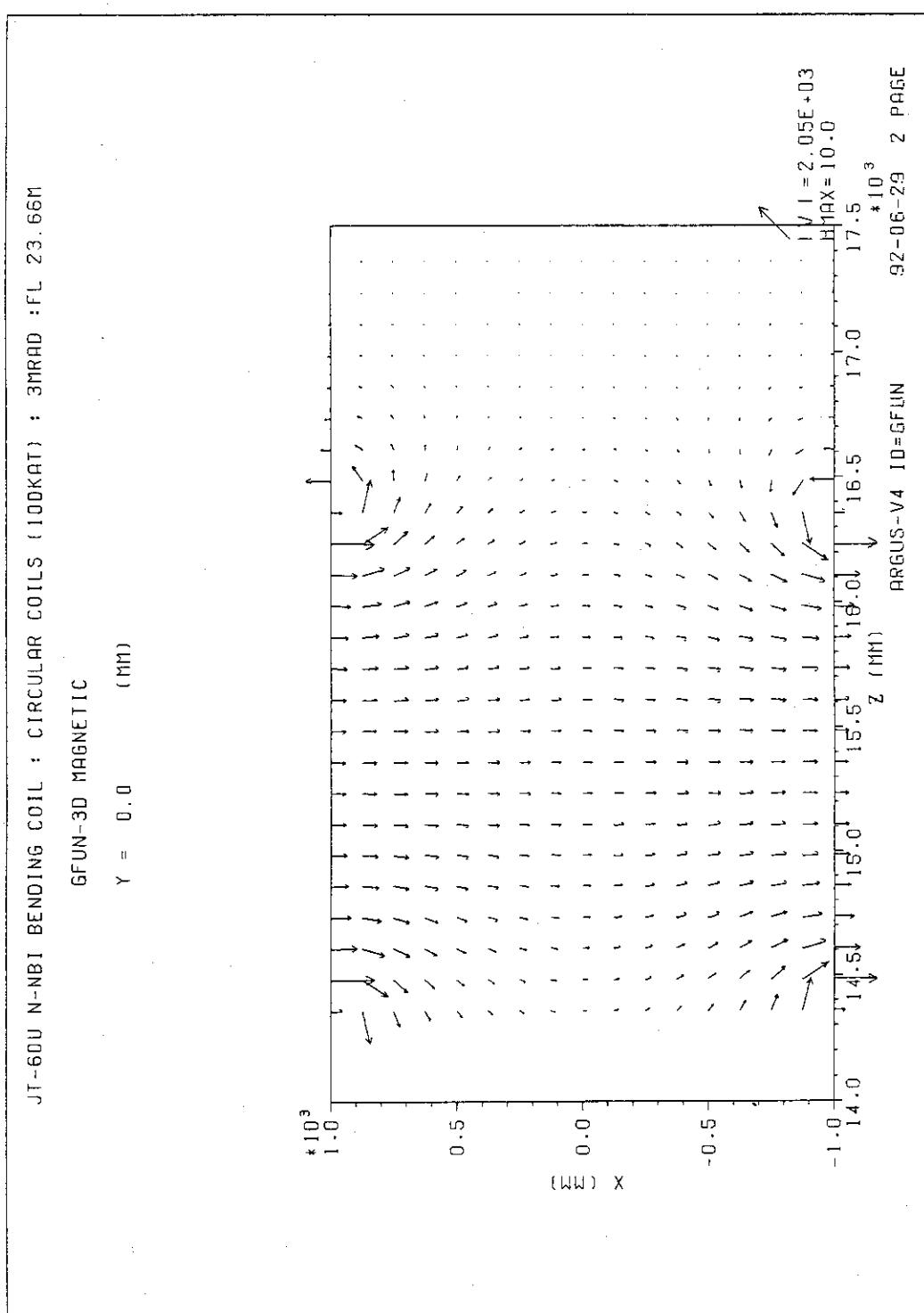


Fig. 9.29 Magnetic field calculated by GFUN

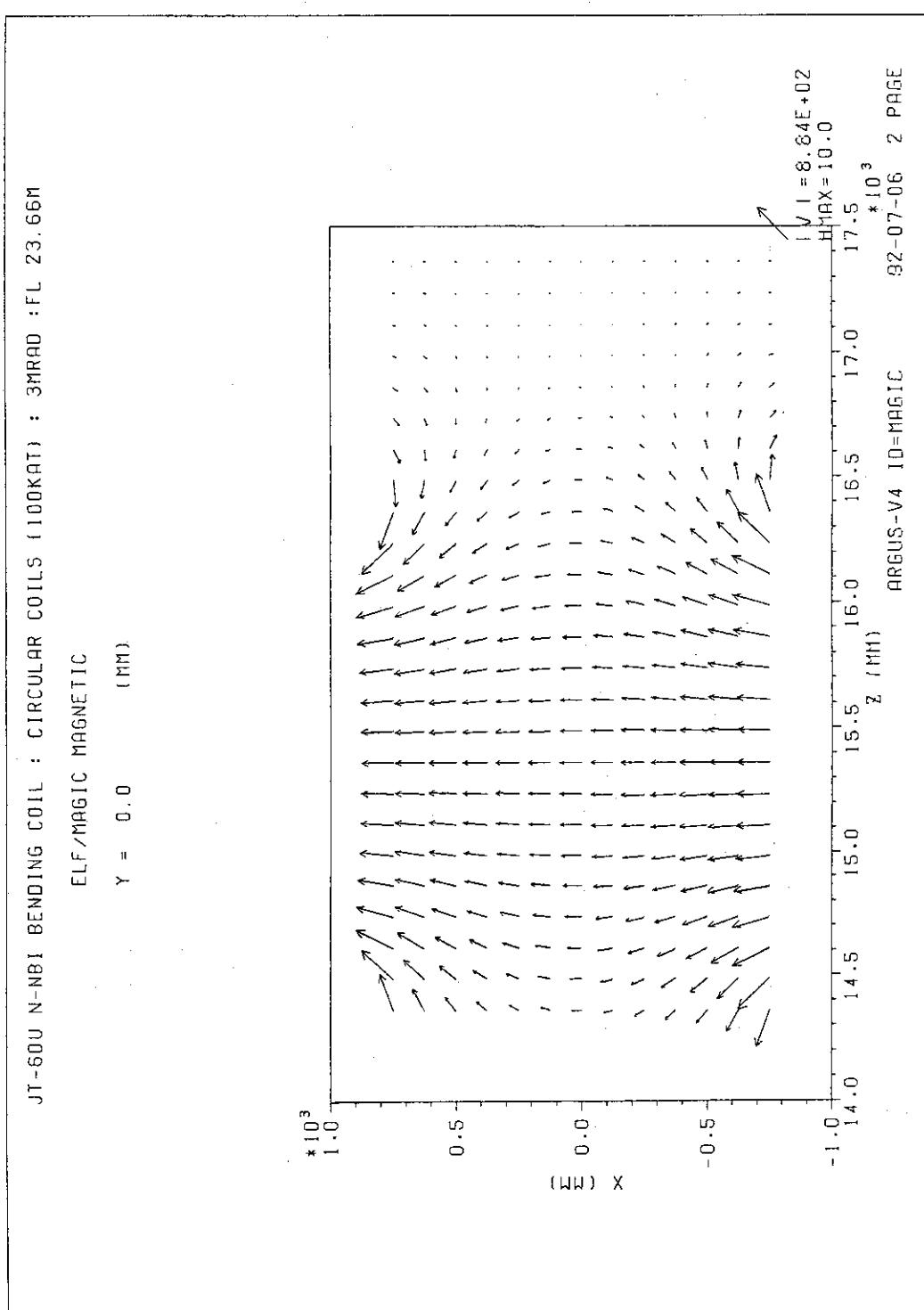


Fig. 9.30 Magnetic field calculated by MAGIC

10. おわりに

情報システムセンター・外来研究員はこの1年間の間にMEUDAS4 コード、 MEUDAS4 FFT 版コード、 FORCE コード、 STREAMコード、自由電子レーザーコードのベクトル化、 HEATING7-VP コード、 SCDAP/RELAP5コード、 NBI3DGFNコードの改良を行った。これらの改良されたコードは現在、ユーザによってよく利用されている。

ベクトル化はFORTRAN77 コンパイラ(V10)による最後のベクトル化作業となった。これからは新コンパイラFORTRAN77EX(V12)によるベクトル化となる。FORTRAN77EX(V12)はFORTRAN77(V10)より機能強化されたコンパイラなので、ベクトル化を通してVP-2600 の有効利用が一層進むものと期待している。

謝辞

HEATING7-VP コードの改良の際に、コード及び改良内容について貴重な助言を頂きました(株)カナザワコンピュータサービスの室伏昭氏に感謝いたします。

SCDAP/RELAP5コードの改良の際に、アセンブラーのAE化に貴重な助言を頂きました富士通(株)の相沢広氏、及び(株)カナザワコンピュータサービスの酒井治行氏に感謝いたします。

富士通(株)の山田圭子さんには各種事務処理を担当してもらい、業務の円滑な遂行が達成できました。感謝いたします。

本レポートを書く機会を与えて下さいました、情報システムセンター室長秋元正幸氏、及び富士通(株)スーパーコンピュータシステム部第2システム課課長橋本道夫氏、及び富士通長野システムエンジニアリング(株)構造解析研究室主管研究員南多善氏に感謝いたします。

10. おわりに

情報システムセンター・外来研究員はこの1年間の間にMEUDAS4 コード、MEUDAS4 FFT 版コード、FORCE コード、STREAMコード、自由電子レーザーコードのベクトル化、HEATING7-VP コード、SCDAP/RELAP5コード、NBI3DGFNコードの改良を行った。これらの改良されたコードは現在、ユーザによってよく利用されている。

ベクトル化はFORTRAN77 コンパイラ(V10)による最後のベクトル化作業となった。これからは新コンパイラFORTRAN77EX(V12)によるベクトル化となる。FORTRAN77EX(V12)はFORTRAN77(V10)より機能強化されたコンパイラなので、ベクトル化を通してVP-2600 の有効利用が一層進むものと期待している。

謝辞

HEATING7-VP コードの改良の際に、コード及び改良内容について貴重な助言を頂きました(株)カナザワコンピュータサービスの室伏昭氏に感謝いたします。

SCDAP/RELAP5コードの改良の際に、アセンブラーのAB化に貴重な助言を頂きました富士通(株)の相沢広氏、及び(株)カナザワコンピュータサービスの酒井治行氏に感謝いたします。

富士通(株)の山田圭子さんには各種事務処理を担当してもらい、業務の円滑な遂行が達成できました。感謝いたします。

本レポートを書く機会を与えて下さいました、情報システムセンター室長秋元正幸氏、及び富士通(株)スーパーコンピュータシステム部第2システム課課長橋本道夫氏、及び富士通長野システムエンジニアリング(株)構造解析研究室主管研究員南多善氏に感謝いたします。

参考文献

- (1) FACOM FORTRAN SSL II 使用手引書, 富士通(株), 1980年12月
- (2) FACOM O S IV FORTUNE V10, 富士通(株), 1986年 3月
- (3) 大金康夫, 吉田清; 磁界計算コード "F O R C E", JAERI-memo 60-334、1985 年11月
- (4) 島本進; 超伝導電磁石に関する研究, 電子技術総合研究所研究報告 第 722号, 総合電子技術研究所, 1971年12月
- (5) FACOM FORTRAN SSL 使用手引書, 富士通(株), 1976年 9月
- (6) 大野豊, 磯田和夫; 数値計算ハンドブック, オーム社, 1990年 9月
- (7) 功刀資彰, 横川三津夫, 千葉猛美; 三次元熱流体解析コード' STREAM' のベクトル化と高機能化, JAERI-M 88-119, 1988年 7月
- (8) 例えば, 日野幹雄著, 流体力学, 朝倉書店, 1989年
- (9) FACOM アナライザ使用手引書 (FORTRAN, VP用) V10L20用, 富士通(株)
- (10) 根本俊行, 鈴木孝一郎, 他; 新FORTRAN コイパイラの導入とベクトル計算機の効果的利用法, JAERI-M 92-105, 1992年 7月
- (11) FACOM O S IV/F 4 MSP FORTRAN77使用手引書, 富士通(株), 1985年 6月
- (12) 佐々木健; 自由電子レーザー発振過程一次元計算プログラム作成報告書, 1990年
- (13) 佐々木健; 高エネルギー研究所, 研究会資料, 1991年
- (14) Charles A. Brau; 'Free Electron Lasers' second edition
(Academic Press, 1991)
- (15) G. Dattoli, A. Renieri ; 'Experimental and Theoretical aspects of Free Electr
Laser' in 'Laser handbook' edited by M. L. Stich and M. Bass
(Elsevier, 1985)

- (16) D.C. Elrod, G.E. Giles, W.D. Turner ; 'HEATING6: A MULTIDIMENSIONAL HEAT CONDUCTION ANALYSIS WITH THE FINITE-DIFFERENCE FORMULATION'
NUREG/CR-0200, Volume 2. Section F10, ORNL/NUREG/CSD-2/V2 (Oct. 1981)
- (17) 「HEATING-6 , HSCXの改良」作業報告書, (財)原子力データセンター, 1986年 3月
- (18) 「流路解析コードの改造」作業報告書, (財)原子力データセンター, 1983年11月
- (19) 「流路解析コードの改良」作業報告書, (財)原子力データセンター, 1984年 9月
- (20) Chris M. allison, et al. ; SCDAP/RELAP5/MOD1 Code Manual, Vol. 1, Vol. 2,
PIN NO. A6360, INEL(Feb. 1988)
- (21) V.H. Ransom, et al. ; RELAP5/MOD2 Code Manual, Vol. 1, Vol. 2,
BGG-SAAM-6337, INEL(Apr. 1984)

付 錄 VBIFD3, VFSINT 使用手引

VBIFD3

[1] 登録申請年月日

平成3年7月

[2] 登録者

情報システムセンター 鈴木 孝一郎 (6448)

[3] 表題

B-spline 2次元補間式による補間、数値微分、数値積分

(富士通 SSL II サブルーチン DBIFD3 のベクトル化版)

[4] 機能

$x-y$ 平面上の格子点 (x_i, y_j) ($x_1 < x_2 < \dots < x_{n_x}, y_1 < y_2 < \dots < y_{n_y}$) において関数値 $f_{ij} = f(x_i, y_j)$ が与えられたとき、 l_{\max} 個の点 $P_l (VX_l, VY_l)$ ($l=1, \dots, l_{\max}$) における補間値あるいは偏微分値あるいは領域 $[x_1 \leq x \leq VX_1, y_1 \leq y \leq VY_1]$ 上の二重積分値を求める (Fig. VBIFD3.1 参照)。

但し、本サブルーチンを利用する前に、SSL II サブルーチン DBIC3 により、 x 方向の接点列 $\{\xi_i\}$ 及び y 方向の接点列 $\{\eta_j\}$ 、更に B-spline 2次元補間式、

$$S(x, y) = \sum_{\beta=-m+1}^{n_y-m} \sum_{\alpha=-m+1}^{n_x-m} c_{\alpha, \beta} N_{\alpha, m+1}(x) N_{\beta, m+1}(y) \quad (1)$$

における補間係数 $c_{\alpha, \beta}$ が計算されているとする。ここで m は奇数で B-spline $N_{\alpha, m+1}(x)$ 及び $N_{\beta, m+1}(y)$ の次数である。 $x_1 \leq VX_1 \leq x_{n_x}, y_1 \leq VY_1 \leq y_{n_y}, m \geq 3$ 及び $n_x \geq m+2, n_y \geq m+2$ であること。

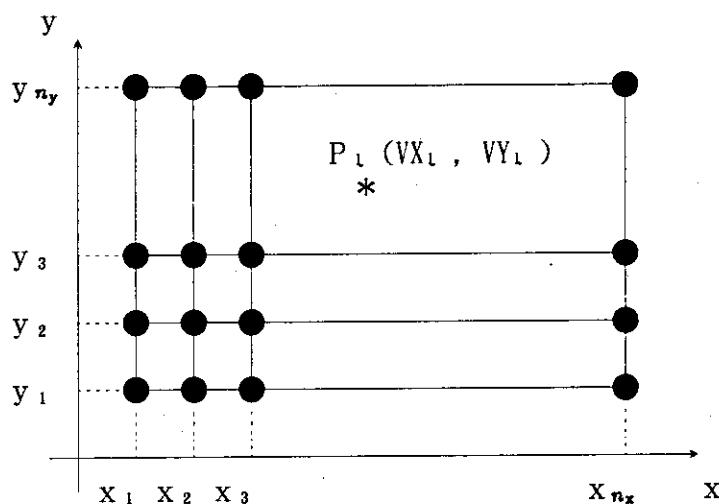


Fig. VBIFD3.1 領域 $R = \{(x, y) \mid x_1 \leq x \leq x_{n_x}, y_1 \leq y \leq y_{n_y}\}$ 中の P

〔5〕呼び出し方

```
CALL VBIFD3 (X, NX, Y, NY, M, C, K, XT,
              ISWX, VX, IX, ISWY, VY, IY,
              F, VW, IWK, LMAX, ICON)
```

X ・・・ 入力： x 方向の離散点 x_i 、大きさ n_x の 1 次元配列
 NX ・・・ 入力： x_i の個数 n_x
 Y ・・・ 入力： y 方向の離散点 y_i 、大きさ n_y の 1 次元配列
 NY ・・・ 入力： y_i の個数 n_y
 M ・・・ 入力： B-spline の次数 m
 C ・・・ 入力： 補間係数 $c_{\alpha,\beta}$ (DBICD3 の出力)、C (K, NY) なる 2 次元配列
 K ・・・ 入力： 2 次元配列 C の整合寸法 k
 VX ・・・ 入力： 点 P_t (VX_t , VY_t) の x 座標、
 大きさ l_{max} の 1 次元配列
 IX ・・・ 入力： $x_i \leq VX_t \leq x_{i+1}$ をみたす i、 $VX_t = x_n$ のときは $IX = n_x - 1$
 出力： $x_i \leq VX_t \leq x_{i+1}$ をみたす i
 大きさ l_{max} の 1 次元配列
 VY ・・・ 入力： 点 P_t (VX_t , VY_t) の y 座標、
 大きさ l_{max} の 1 次元配列
 IY ・・・ 入力： $y_i \leq VY_t \leq y_{i+1}$ をみたす i、 $VY_t = y_n$ のときは $IY = n_y - 1$
 出力： $y_i \leq VY_t \leq y_{i+1}$ をみたす i
 大きさ l_{max} の 1 次元配列
 F ・・・ 出力： 補間値あるいは偏微分値あるいは積分値、
 大きさ l_{max} の 1 次元配列
 VW ・・・ 出力： 作業領域、
 大きさ l_{max} , $4(m+1)+\max(n_x, n_y)+\text{INT}((m+1)/2)*4+4$ の 2 次元配列
 IWK ・・・ 出力： 作業領域、大きさ l_{max} , 6 の 2 次元配列
 LMAX ・・・ 入力： 整合寸法、点 P の数 l_{max}
 ICON ・・・ 出力： コンディションコード、大きさ l_{max} の 1 次元配列

〔6〕使用上の注意

S S L II サブルーチンである DBIFD3 を繰り返し呼び出して使用する場合に本サブルーチンを使うと効果的である（ベクトル長 l_{max} でベクトル化される）。

Fig. VBIFD3.2 に DBIFD3 と VBIFD3 を使用した場合を比較する。2 つのプログラムは互いに等価である。

[DBIFD3を使用した場合]

```

DO 100 N=1,L
    CALL DBIFD3(X, NX, Y, NY, M, C, K, XT, ISWX, VX(N), IX(N),
*           ISWY, VY(N), IY(N), F(N), VW(1,1), ICON(N) )
100   CONTINUE

```

[VBIFD3を使用した場合]

```

CALL VBIFD3(X, NX, Y, NY, M, C, K, XT, ISWX, VX, IX,
*           ISWY, VY, IY, F, VW, IWK, LMAX, ICON )

```

Fig. VBIFD3.2 VBIFD3の使用法

[7] 解法および参考文献

B-spline 2次元補間式

参考文献：FACOM マニュアル：“FACOM FORTRAN S S L II 使用手引書”

[8] 記憶容量

116 kbyte

[9] 計算時間

点Pの個数 l_{max} が50のとき、S S L II サブルーチンDBIFD3の約4.9倍。

(Fig. VBIFD3.2 の使用法での比較)

[10] 精度

S S L II サブルーチンDBIFD3に等しい。

[11] 内蔵するエラーメッセージ

参考文献のB I F D 3 の項を参照のこと。

(但し、V B I F D 3 では l_{max} 個のエラーメッセージを配列に格納する)

[12] 言語

F O R T R A N

[13] 使用エントリ名

付属ルーチン VECCAD, VECBAS, VCBAS2

S S L II ルーチン MGSSL

[14] 公開の程度

原研内のみ

VFSINT

〔1〕 登録申請年月日

平成3年7月

〔2〕 登録者

情報システムセンター 鈴木 孝一郎 (6448)

〔3〕 表題

離散型 sine 変換(台形公式、2基底FFT)

(富士通 SSL II サブルーチン DFSINT のベクトル化版)

〔4〕 機能

周期 2π である m 個の奇関数 $x_i(t)$ があり、その各々について半周期を n 等分した n 個の標本があるとする。これらの $n * m$ 個の標本からなる集合 $\{x_{ij}\}$

$$x_{ij} = x_i(\pi j / n) \quad (i=1, 2, \dots, m : \\ j=0, 1, 2, \dots, n-1) \quad (1)$$

が与えられたとき、 m 個の奇関数について台形公式による離散型 sine 変換、またはその逆変換を高速変換手法 (FFT) により行う。ただし、 $n = 2^l$ (l : 正整数) であること。

① sine 変換

 $\{x_{ij}\}$ を入力し、式(2) で定義する変換を行い、フーリエ係数 $\{nb_{ik}/2\}$ を求める。

$$\left\{ \begin{array}{l} \frac{nb_{1k}}{2} = \sum_{j=0}^{n-1} x_{1j} \sin(\pi k j / n), \quad k=0, 1, \dots, n-1 \\ \vdots \\ \frac{nb_{mk}}{2} = \sum_{j=0}^{n-1} x_{mj} \sin(\pi k j / n), \quad k=0, 1, \dots, n-1 \end{array} \right. \quad (2)$$

但し、 $x_{i0} = 0 \quad (i=1, 2, \dots, m)$

② sine 逆変換

$\{b_{ik}\}$ を入力し、式(3) で定義する変換を行い、フーリエ級数 $\{x_{ij}\}$ を求め る。

$$\left\{ \begin{array}{l} x_{1j} = \sum_{k=0}^{n-1} b_{1k} \sin(\pi k j / n), \quad j=0, 1, \dots, n-1 \\ \vdots \\ x_{mj} = \sum_{k=0}^{n-1} b_{mk} \sin(\pi k j / n), \quad j=0, 1, \dots, n-1 \end{array} \right. \quad (3)$$

但し、 $b_{i0} = 0 \quad (i=1, 2, \dots, m)$

〔5〕 呼び出し方

```

CALL VFSINT (A, NR, N, TAB, NRS, NRE, ICON)
      A    ... 入力: {xij} または {bik}
            出力: {n * bik / 2} または {xij}
                  大きさm, nの2次元配列
      N    ... 入力: 奇関数1つあたりの標本数 n
      M    ... 入力: 奇関数の数 m
      TAB   ... 入力: 変換で使用された三角関数表が格納される
                  大きさn/2 - 1の1次元配列
      NRS, NRE  ... 入力: FFTを行う奇関数の範囲
      ※ NRS, NRE の指定で、NRS 番目の奇関数からNRE 番目の奇関数まで
          FFTを行う。
      ICON  ... 出力: コンディションコード

```

〔6〕 使用上の注意

S S L II サブルーチンであるDFSINTを繰り返し呼び出して使用する場合に本サブルーチンを使うと効果的である（ベクトル長：(NRE-NRS) でベクトル化される）。

Fig. VFSINT.1 にDFSINTを使用した場合とVFSINTを使用した場合を比較する。

配列B と配列A は標本値を格納する順序が異なることに注意しなければならない。

〔DFSINTを使用した場合〕

```

DO 100 I = 1, M
    CALL DFSINT(B(1, I), N, TAB, ICON)
100 CONTINUE

```

〔VFSINTを使用した場合〕

```

DO 10 I = 1, M
    A(I, J) = B(J, I)
10 CONTINUE
    CALL VFSINT(A, M, N, TAB, 1, M, ICON)

```

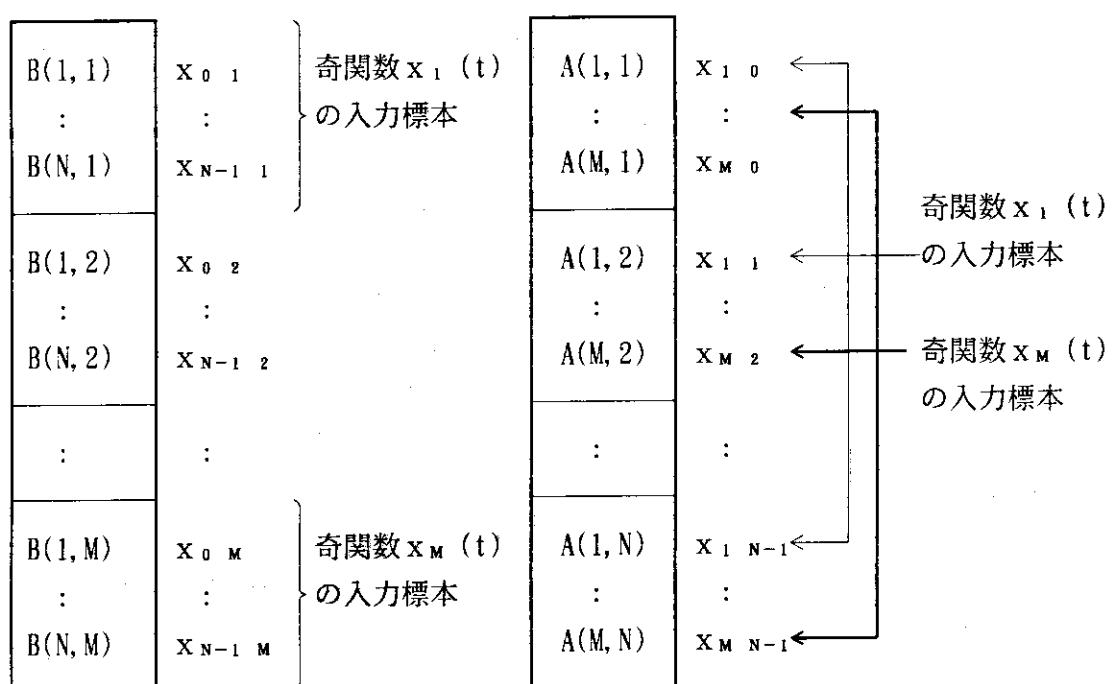


Fig. VFSINT. I VFSINTの使用法

〔7〕 解法および参考文献

台形公式, 2 基底FFT

参考文献: FACOM マニュアル: "FACOM FORTRAN S S L II 使用手引書"

〔8〕 記憶容量

76 kbyte

[9] 計算時間

奇関数の個数が 511 の場合で、SSL II サブルーチン DFSINT の約 30 倍。
(Fig. VFSINT.1 に示される使い方をした場合の比較)

[10] 精度

SSL II サブルーチン DFSINT に等しい。

[11] 内蔵するエラーメッセージ

参考文献の F S I N T の項を参照のこと。

[12] 言語

F O R T R A N

[13] 使用エントリ名

付属ルーチン ····· VUPNR2, VUSINM

SSL II ルーチン ····· DUTABT, MGSSL

[14] 公開の程度

原研内のみ