JAERI-Data/Code
2000-039

原子力コードの高速化（移植編）

－平成11年度作業報告書－

2001年1月

川崎　信夫*・根本　俊行*・川井　渉*・小笠原　忍
石附　茂*・久米　悦雄・箭竹　陽一*・足立　将晶

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

本レポートは，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。
入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-1195　茨城県那珂郡東海村）あて，お申し越しください。なお，このほかに財団法人原子力弘済会資料センター（〒319-1195　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。

This report is issued irregularly.
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, 319-1195, Japan.

編集兼発行　　日本原子力研究所

JAERI－Data/Code 2000－039

原子力コードの高速化（移植編）
－ 平成11年度作業報告書 －

日本原子力研究所計算科学技術推進センター
川崎 信夫＊・根本 俊行＊・川井 渉＊・小笠原 忍※
石附 茂＊・久米 悦雄・箭竹 陽一＊＊・足立 将晶※

（2000年10月4日受理）

本報告書は，平成11年度に計算科学技術推進センター情報システム管理課で行った原子力コードの高速化作業のうち，VPP500またはAP3000への移植整備作業について記述したものである．原子力コードの高速化作業は，平成11年度に18件行われた．これらの作業内容は，今後同種の作業を行う上での参考となりうるよう，作業を大別して「ベクトル／並列化編」，「スカラ並列化編」及び「移植編」の3分冊にまとめた．

本報告書の「移植編」では，生体分子の分子動力学パッケージAMBER5，（連続・多群）汎用中性子・光子輸送計算モンテカルロコードMVP/GMVP，MCNPライブラリ自動編集システムautonj，SPECTER/SPECOMPコード，核融合炉事故解析コードMELCOR-FUS及びサブチャンネル解析コードCOBRA-TFのVPP500及びAP3000への整備について記述している．

日本原子力研究所（東海駐在）：〒319-1195 茨城県那珂郡東海村白方白根2-4
※ 外来研究員：富士通株式会社
＊ 富士通株式会社
＊＊ 株式会社日立製作所

JAERI−Data/Code 2000−039

Vectorization, Parallelization and Porting of Nuclear Codes (Porting) - Progress Report Fiscal 1999 -

Nobuo KAWASAKI*, Toshiyuki NEMOTO*, Wataru KAWAI*,
Shinobu OGASAWARA※, Shigeru ISHIZUKI*, Etsuo KUME,
Yo-ichi YATAKE** and Masaaki ADACHI※

Center for Promotion of Computational Science and Engineering
(Tokai Site)
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received October 4 , 2000)

Several computer codes in the nuclear field have been vectorized, parallelized and transported on the FUJITSU VPP500 system, the AP3000 system, the SX-4 system and the Paragon system at Center for Promotion of Computational Science and Engineering in Japan Atomic Energy Research Institute. We dealt with 18 codes in fiscal 1999. These results are reported in 3 parts, i.e., the vectorization and the parallelization part on vector processors, the parallelization port on scalar processors and the porting part. In this report, we describe the porting.

In this porting part, the porting of Assisted Model Building with Energy Refinement code version 5 (AMBER5), general purpose monte carlo codes for neutron and photon transport calculations based on continuous energy and multigroup methods (MVP/GMVP), automatic editing system for MCNP library code (autonj), neutron damage calculations for materials irradiations and neutron damage calculations for compounds code (SPECTER/SPECOMP), severe accident analysis code (MELCOR) and COolant Boiling in Rod Arrays, Two-Fluid code (COBRA-TF) on the VPP500 system and/or the AP3000 system are described.

Keywords : SPECTER/SPECOMP, COBRA-TF, AMBER5, autonj, MVP/GMVP, MELCOR-FUS, VPP500, AP3000, Nuclear Codes

---

※ On leave from FUJITSU, Ltd
* FUJITSU,Ltd
** HITACHI,Ltd

目　　次

Contents

1. はじめに

計算科学技術推進センター情報システム管理課では，原研が保有する各種スーパーコンピュータの効率的な運用とコンピュータ資源の有効利用を促進するため，計算需要の多い原子力コードをユーザに代わってスーパーコンピュータ上に整備し，それぞれのコードに最適な高速化を施す作業を実施している．この作業は，コンピュータの効率的利用を推進するのみならず，ユーザの計算待ち時間の短縮を通じてユーザの仕事の効率化へも貢献するものと思われる．

原子力コードの高速化作業は，平成11年度に18件行われた．これらの作業内容は，今後同種の作業を行う上での参考となりうるよう，作業を大別して，「ベクトル／並列化編」，「スカラ並列化編」及び「移植編」の3分冊にまとめた．

本報告書の「移植編」では，生体分子の分子動力学パッケージAMBER5，（連続・多群）汎用中性子・光子輸送計算モンテカルロコードMVP/GMVP，MCNPライブラリ自動編集システムautonj，SPECTER/SPECOMPコード，核融合炉事故解析コードMELCOR-FUS及びサブチャンネル解析コードCOBRA-TFのVPP500及びAP3000への整備について記述している．

別冊の「ベクトル／並列化編」では，JAMコード及び3次元熱流体解析コードSTREAMを対象に実施したベクトル化作業について，相対論的分子軌道法コードRSCAT，相対論的密度汎関数法コードRDFT及び高速3次元中性子拡散ノード法コードMOSRA-Lightを対象に実施したベクトル並列化作業について記述している．また，別冊の「スカラ並列化編」では，高エネルギー核子・中間子輸送計算コードNMTC，ブラソフプラズマシミュレーションコードDA-VLASOV及び中性子・光子結合モンテカルロ輸送計算コードMCNP4B2を対象に実施したParagon向けのスカラ並列化作業について記述している．なお，平成11年度に実施した高速化作業のうち，ここで取り上げなかったいくつかのコードに関しては，ユーザとの連名により別途JAERI-Data/Codeを執筆する予定であるので，そちらを参照されたい．

2章では，生体分子の分子動力学パッケージAMBER5のVPP500及びAP3000への整備作業について述べる．本作業では，主計算コードを対象に，ベクトル化・並列化済みの部分を有効にするとともに，大規模データ向けへの改良を実施した．本パッケージは，元ソースから各ベンダ向け且つ各コンパイラ向けのソースを抜き出し，これをコンパイル対象にすることができることから，VPP500及びAP3000向けのソースについてそれぞれ整備を行った．

3章では，（連続・多群）汎用中性子・光子輸送計算モンテカルロコードMVP/ GMVPのAP3000への整備作業について述べる．本作業では，MPIライブラリを用いたMVP/GMVP（並列版）をAP3000上に整備した．

4章では，MCNPライブラリ自動編集システムautonjのAP3000への整備作業について述べる．本作業では，autonjコードをAP3000に整備するとともに，多くのユーザが共用出来るようにバッチ起動等の共同利用形態への変更及び代表的な処理条件で作成されたJENDL-3.2の断面積ライブラリ群の登録を実施した．

5章では，SPECTER/SPECOMPコードのAP3000への整備作業について述べる．本作業では，VAX-FORTRANでコーディングされていた当該コードをAP3000上に整備した．

6章では，核融合炉事故解析コードMELCOR-FUSのVPP500でのベクトル処理による高速

化の可能性調査について述べる．本作業では，VPP500上の解析ツールsamplerを使用しコードの動的解析を行い，上位10ルーチンに関して，高速化した場合の各ルーチン毎の計算速度向上率（予測値）を推定した．また，コード全体としての計算速度向上率（予測値）の推定も実施した．

7章では，サブチャンネル解析コードCOBRA-TFのベクトル処理による高速化の可能性調査について述べる．本作業では，先ず当該コードをVPP500にインストールし，効果的なベクトル処理が可能かどうかの調査を行った．その結果，ベクトル処理には不向きと判断し，AP3-000へ整備を実施した．

なお，本報告書の2章の作業は川井が，3章，4章の作業は川崎が，5章の作業は小笠原が，6章の作業は石附が，7章の作業は根本が担当した．

2. AMBER5 パッケージのインストール

本章では，富士通製分散メモリ型ベクトル並列計算機 VPP500/42 （以下 VPP500）及び富士通製共用 UNIX サーバ AP3000/24 （以下 AP3000：分散メモリ型スカラ並列計算機）における，生体分子の分子動力学計算パッケージ AMBER5(Assisted Model Building with Energy Refinement version 5) のインストールについて述べる．

本パッケージは，オリジナル版が，開発元及び配布元である Kollman research group(at the University of California San Francisco.) [1] によって，既にベクトル化と並列化が施されているパッケージである．且つそれらのベクトル化と並列化が各ベンダー向け，且つ各コンパイラ向けに施されている非常に利用性の高いコードである．

しかし，ユーザは AP3000 上でシングル実行を行なっていた．また，ユーザが行なっている DNA 損傷シミュレーションは，中規模データを用いた場合，AP3000 上で 1 ジョブの実行に 2 時間 30 分 を要し，これを 1000 回程繰り返して初めて 1 ケースの問題についてのシミュレーションが終了するという，極端な長時間ジョブであった．

そこで，この実行時間を短縮するため，今回の高速化作業が依頼された．行なった作業は，オリジナルパッケージの高速化部をそのまま利用するインストール作業がメインになった．

作業対象は，パッケージにいくつか存在するコードの内，1000 回繰り返しの対象である sander と gibbs である．しかし，gibbs については，高速化部（並列化部）に障害を持っており，また，ユーザがまだ gibbs を利用していないため，入力データや実行結果が揃っていないことから，今回の作業ではシングル実行のみの実行確認を行なった．その他，link，edit，parm，anal コードについては大規模データ向けの修正を行なった．AMBER5 には，これらの他にもコードが存在するが，作業では触れなかったため，本報告書でも触れない．

2.1 パッケージの概要

本パッケージには，複数のコードが存在する．これらは，主計算を行なうコード群，主計算に対する入力データを作成するコード群，主計算結果を解析するコード群に分けることができる．Fig. 2.1 に示す，四角い箱で囲んでいる個々の名前が，これらコードの名前であり，矢印の方向が AMBER5 パッケージ全体の処理フローである．斜体文字で記述しているものは，前のコードの出力データから次のコードの入力データになるものである．

(1) link コード

AMBER 標準の residue データベース，または prep コードで作成された residue データベースを基にトポロジーを抽出する．

(2) edit コード

PDB (Protein Data Bank) 座標を読み込み，これらを link で作成されたトポロジーに適用

する．PDB 座標で不可能の場合には，link の出力データである lnkbin データ中の内部座標を用い edit 自身が座標を作成する．

(3) parm コード

モデルに存在する原子の共有結合，結合角，二面角を決め，それらに対し，force field file から適切な力場パラメータを設定する．

(4) sander コード

基本的なエネルギー極小化と分子動力学プログラムである．原子間の共有結合については，NMR restraints(distance restraint, angle restraint, torsion restraint) と，NOESY restraint, Chemical shift, Pseudo-Contact shift を取り扱うことが可能である．また，非共有結合については，ファンデルワールス力 (Van Dar Waals) と静電相互作用 (electrostatic interaction) を取り扱うことが可能である．巨視的格子での単位格子の全静電エネルギーを計算する方法としては，PME(Particle Mesh Ewald) 法が用いられる．

(5) gibbs コード

自由エネルギー計算のためのプログラムであり，sander に似ている．

(6) anal コード

エネルギーの分析と構造の分析に用いられる．

(7) AMBER5 のディレクトリ構成

ここで，以降の説明のため AMBER5 パッケージのディレクトリ構成の一部を Fig. 2.2 に示す．

2.2 sander コードのインストール

ここでは，sander コードの VPP500 と AP3000 へのインストール作業について述べる．ユーザは AP3000 を利用して実行していたため，作業も AP3000 から始めた．これは，インストール後のコードの実行結果を確認する上で，同じ計算機で比較を行なった方がメリットが大きいからである．

2.2.1 AMBER5 パッケージのコンパイル&実行方法

既述のように，AMBER5 は各ベンダー向け，且つ各コンパイラ向けにコーディングされているコード群から成る．よって，VPP500 や AP3000 で実行する場合も，両計算機向けにコンパイルと実行を行なう必要がある．

(1) コンパイル方法

コンパイルプロセスでは，各ベンダー向け且つ各コンパイラ向けソースプログラムが混在している元のソースプログラムから，該当ベンダー向け且つ該当コンパイラ向けのソースプログラム

を抜き出し，これを実際のコンパイル対象のソースプログラムにする．ソースプログラムの抜き出しは，元のソースプログラムに記述されている # で始まるシンボル行と，C 言語プリプロセッサである cpp コマンド，cpp コマンドへのオプションにより行なわれる．各コードをコンパイルする場合は，ソースプログラム以外に，次の 4 つのファイルを使用する．

a. シェルスクリプト Makeall

ディレクトリ src 上に存在する．これを実行すると，各コードが格納されているディレクトリを移動しながら，make コマンドを実行する．通常，複数コードを同時にコンパイルする場合に利用する．

b. メイクファイル

各コードが格納されているディレクトリ上に存在する．各コードをコンパイルする場合に，make コマンドの引数になるファイルである．

c. シェルスクリプト Compile

ディレクトリ src 上に存在する．b. のメイクファイル中のコンパイルコマンドの記述部において実行される．コードを構成する各ファイルを実際にコンパイルするファイルである．

d. 環境変数設定ファイル MACHINE

各ベンダー向け，且つ各コンパイラ向けの cpp コマンドや，cpp のオプション，コンパイルコマンド，コンパイルオプション等は，ディレクトリ src/Machine 配下に存在する Machine.* ファイル中において環境変数として利用できるように記述されている．これらのファイルは，MACHINE からシンボリックリンクによって使用される．これらファイル中の環境変数は，Compile の実行初期段階で有効化される．

(2) 実行方法

実行方法は，各コードとも同様の形態であるが，各ロードモジュールに対しオプション（引数）を与える必要がある．これらオプションによって，何のデータか識別される．次に実行例を示す．

```
load_module -O \
-i standardin \
-o standardout \
-p Ex1.topo \
-c Ex1.coord \
-r restart \
-inf information \
-ms state
```

これを，qsub コマンドの引数に指定する実行シェルスクリプトに記述する．

2.2.2 AP3000 での sander の高速化（インストール）

ここでは，AP3000 でのスカラ並列化版 sander のインストール作業において，オリジナルパッケージに対して施した修正等について述べる．

(1) 環境変数設定ファイル MACHINE の準備

AP3000 上で用いる環境変数設定ファイルは，ディレクトリ src/Machine 配下に存在するファイル Machine.fuj_ap_frt と Machine.fuj_ap_f77 である．これらのいずれかに，ディレクトリ src 上に用意するファイル MACHINE からシンボリックリンクを行なって使用する．オリジナルの Machine.fuj_ap_frt を Fig. 2.3 に，Machine.fuj_ap_f77 を Fig. 2.4 に示す．

(2) 実行確認

AP3000 上で，Machine.fuj_ap_frt と Machine.fuj_ap_f77 を用いてコンパイルし，実データを用いてシングル実行と 1～4CPU まで用いた並列実行を行なった．Machine.fuj_ap_frt で作成したロードモジュールの実行は問題無く終了することを確認した．しかし，Machine.fuj_ap_f77 で作成したロードモジュールの並列実行では，次のメッセージが出力されて実行が途中終了した．

```
ccccccccccccccccccccccccccccccccccccccccc

      -DMEM_ALLOC not compatible with
      parallel versions for now, This
      message is intended to produce
      a compiler error.. remove the
      -D*_MP, -DMPI or -DMEM_ALLOC flag
      from your MACHINE file or use a non-
      parallel one & try again.

ccccccccccccccccccccccccccccccccccccccccc
```

元のソースプログラムにおいて，MEM_ALLOC 部を調査した結果，これは動的メモリ確保のための部分で出力されたメッセージであった．よって，上のメッセージより，並列化版 sander ではメモリを動的に確保する仕様ではないということが判明した．ユーザは，Machine.fuj_ap_f77 を用いてシングル実行を行なっていたため，特に配列サイズを指定する必要がない環境で（動的メモリ確保を利用して）実行していたことになる．これより，並列実行では必要な配列サイズを指定しなければ，宣言サイズ不足で実行が途中終了してしまう場合が発生するため，ユーザは配列サイズの複数回のリサイズ操作が必要になり煩わしい．しかし，今回の作業において大規模データ向けの対応をした（2.5 参照）ため，サイズ不足で途中終了する確率は充分低くなっていることから，ここでは，元のソースプログラムから動的確保部を抜き出すオプション -DMEM_ALLOC をファイル MACHINE から単純に省くことにした．これを Fig. 2.5 に示す．尚，このオプションを省いた場合の実行は問題無く終了することを確認した．

2.2.3 VPP500 での sander の高速化（インストール）

ここでは，VPP500 でのベクトル並列化版 sander のインストール作業において，オリジナルパッケージに対して施した修正等について述べる．

(1) 環境変数設定ファイル MACHINE の準備

VPP500 上で用いる環境変数設定ファイルは，ディレクトリ src/Machine 配下に存在するファイル Machine.fuj_vpp である．これに，ディレクトリ src 上に用意するファイル MACHINE からシンボリックリンクを行なって使用する．オリジナルの Machine.fuj_vpp を Fig. 2.6 に示す．しかしこの設定内容は，VPP300 を想定して作成されているため，原研 VPP500 では使用できない．そこで，原研 VPP500 仕様向けの設定に変更した．変更後を Fig. 2.7 に示す．

(2) 実行確認

Machine.fuj_vpp を用いてコンパイルし，AP3000 で使用した実データと同じデータを用いて，ベクトル実行を行なったが，コード中のエラー処理部からのメッセージが出力され，実行が途中で終了（異常終了）した．

・FUJFFT（高速フーリエ変換）部の無効化

異常終了の原因を調査するため，ファイル MACHINE に記述されている cpp へのオプションを一つずつ省いてロードモジュールを作成し，各ロードモジュールの実行を行なった．その結果，元のソースプログラムから FFT（高速フーリエ変換）部を抜き出すための cpp オプションである -DFUJFFT を有効にすると異常終了することが判明した．そこで，元のソースプログラムにおいてシンボル行に FUJFFT を指定している部分についてデバッグを行なった．この結果，FFT（高速フーリエ変換）後の結果が，FUJFFT を有効にした場合と無効にした場合とで異なっていた．更に調査した結果，富士通科学用サブルーチンライブラリ（SSL2）の利用時に異常終了になることが分かった．

次に，ファイル MACHINE において -DFUJFFT を cpp のオプションから除いて，1～4CPU までを用いたベクトル並列実行を行なった．その結果，いずれの結果も AP3000 上での結果と大きく異なっており，3CPU 実行では異常終了した．

・VECTOR（ベクトル計算機用）部の無効化

初めに，3CPU 実行時の異常について調査した．その結果，ある配列のサイズをオーバー（領域破壊）して 0 クリアと値の定義をしている箇所があった．そこで，暫定的に該当配列のサイズを大きく（領域破壊回避）して実行を行なったが，更に別の部分で結果が異常になっていた．

これらについては，元のソースプログラムにおけるシンボル行が Fig. 2.8のようになっている部分に対して VECTOR 部と MPI 部を抜き出すようにコンパイルをした

場合，VECTOR 部が有効になる部分と VECTOR 部より MPI 部が優先的に有効になる部分があるため，同一の変数や配列でも定義&参照範囲が異なっていたことが原因であった．この現象は，異常終了しなかった 2 または 4CPU を用いた並列実行の場合も発生していた．これらの現象を回避するには，ファイル MACHINE での cpp へのオプションである -DVECTOR を取り除く必要があった．

以上，2 つの理由から，VPP500 向けのファイル MACHINE から -DFUJFFT と -DVECTOR のオプションを取り除くことにした．これにより，FUJFFT 部は AMBER5 のオリジナル FFT 部に，VECTOR 部はスカラ計算機を利用する場合に有効化される部分に置き換わることになる．ファイル MACHINE の修正部を Fig. 2.9 に示す．-DFUJFFT と -DVECTOR のオプションを取り除いてコンパイルした場合では，実行が問題無く終了することを確認した．

2.2.4 インストールした sander コードの実行時間

ここでは，インストールした sander コードのベクトル実行時間，並列実行時間について述べる．

2.2.4.1 AP3000 での実行時間

AP3000 上での sander コードのスカラ実行，スカラ並列実行時間を Table 2.1 に示す．これにおいて，ユーザの実行環境での実行時間は，Machine.fuj_ap_f77 でのシングル実行である．これによると，ユーザ利用環境における実行時間に比較して，オリジナルの並列化部を有効にしたコードの実行時間は，4 プロセスで 約 3.1 倍の倍率が出ている．

2.2.4.2 VPP500 での実行時間

オリジナルのコンパイルを行なった場合の，ベクトル実行時間，ベクトル並列実行時間を Table 2.2 に示す．これによると，AP3000 程の性能は出ていない．これは，[FUJFFT,VECTOR] 部の無効化や計算機のハード自体の性能が原因と考えられる．

2.2.5 インストールした sander コードに対する評価

ここでは，sander コードの実行時間の評価，[FUJFFT,VECTOR] 部に対するユーザとの検討結果と，実行結果の評価について述べる．

2.2.5.1 実行時間の評価と [FUJFFT,VECTOR] 部の検討

インストール版 sander コードの 4CPU 並列実行時間は，ユーザ利用環境における実行時間に比較すると，AP3000 で約 3.1 倍，VPP500 で約 2.4 倍の倍率が出ている．VPP500 上で更に性能を上げるには，[FUJFFT,VECTOR] 部を有効にした場合のソースプログラムを修正しなければならない．

そこで，ユーザに次のいずれかを選択して頂くことにした．

a. AP3000 を利用する．
b. [FUJFFT,VECTOR] 部の修正を行なって VPP500 を利用する．

c.　[FUJFFT,VECTOR] 部の修正を行なわないで VPP500 を利用する.

その結果，初めは実行時間やコードについて何も問題の無い AP3000 上での実行を希望された．しかし，AP3000 と VPP500 の計算機運用状況とユーザが今まで AP3000 上で実行してきた経験から，ジョブのスループットは，VPP500 の方が良いと判断された．また，コードも [FUJFFT,VECTOR] 部の修正は難解（ユーザ側で修正して頂くように依頼）で時間がかかるため，[FUJFFT,VECTOR] 部の修正は行なわないことになり，最終的に上の c. を選択して頂いた.

2.2.5.2　実行結果について

VPP500 を使用して頂くことになったため，sander コードの [FUJFFT,VECTOR] 部を無効化してコンパイル＆実行した結果を確認して頂いた．この時点では，4CPU のみの実行結果を確認して頂き，問題無いとの評価であった．また，この時点以降での実際の利用においても実行結果は問題無いとの評価である.

2.3　gibbs コードについて

gibbs コードはユーザがまだ利用したことのないコードであるため，暫定データのみを作成して頂き，これを用いてベクトル実行の実行確認をおこなった．これについては，Fig. 2.7（ファイル Machine.fuj_vpp）の内容で実行が問題無く終了することを確認した．また，実行結果をユーザに確認して頂いたが，問題無いとの評価であった．次に，同様のデータを用いてベクトル並列実行の実行確認を行なったが，次のメッセージが出力され異常終了した.

```
'MPI version of gibbs is currently disabled.'
```

ソースプログラムを確認した結果，このメッセージはメインルーチンの最初の実行文で write されているものであり，並列実行は不可能であることが判明した．本来であれば，MPI version を修正する必要があるが，これについてユーザと検討した結果，次の理由から修正作業は保留になった.

- 現時点で，ユーザがまだ gibbs コードを利用したことがない.
- 実データがない.
- 実データがないため，実行結果もない.
- 修正作業を行なっても，実データや実行結果がないため，評価ができない.

2.4　ジョブ投入の自動化

ユーザが行なう DNA 損傷シミュレーションは，既述のように，長時間ジョブを約 1000 回投入しなければ 1 ケースについての計算が終らない．よって，ジョブ投入操作が単純に多数回（リスタートジョブとして）繰り返される．しかし，リスタートジョブの投入直前に，前のジョブが終了したか否かをその都度確認しなければならない.

そこで，この煩わしさを取り除くことと実行効率の向上を目的にジョブ投入を自動化するためのシェルスクリプトを提供した．提供したシェルスクリプトは [2] のオリジナル版を基にユーザ用にカスタマイズしたものである．このシェルスクリプトを Fig. 2.10 に示す．これにより，平

日に限らず，夜間や休日等も（計算機の運用が停止しない限り）ジョブ投入を行なうことが可能になった．オリジナル版のシェルスクリプトに対して追加した機能は，ジョブがCPU オーバーによって終了した時に自動シェルスクリプトも終了させる機能である．バッチジョブがCPU オーバーで終了した時は，NQS のログファイルに出力されるメッセージに 'SIGXCPU' の文字列が含まれる．これを利用して自動シェルスクリプトを終了させるようにした．

2.5 大規模データに対する修正

今回の作業の時点では，ユーザは中規模（原子数 約20000個）データを用いたシミュレーションを行なっていたが，今後，大規模（原子数 約50000個）データを用いたシミュレーションも行ないたいとの希望により，その際利用する各コードに対して，大規模データ向けの対応を行なった．

対応を行なったコードは，link，edit，parm，sander，anal の各コードであり，それぞれの大規模データを用いて行なった．いずれのコードも配列サイズを宣言しているインクルードファイル `sizes.h` において，各サイズをリサイズするのみで実行が問題無く終了することを確認した．また，それぞれの実行結果をユーザに確認して頂いた．いずれの結果も問題無いとの評価であった．

2.6 まとめ

今回の作業結果，コード単体の性能が良い AP3000 版ではなくVPP500 版のコードをVPP500 で利用して頂くことになった．これは，両計算機におけるジョブのスループットの違いが大きな理由である．また，ユーザが行なう DNA 損傷シミュレーションは，多ケースの問題について分析を行なう場合，数千回ものジョブ投入が必要であることから，ジョブ投入自動化シェルスクリプトを提供した．

これらにより，ユーザには，コード単体の性能向上（インストール）だけでなく，VPP500 を利用することによるジョブのスループット向上，更にはジョブ投入の自動化シェルスクリプトによる実行効率の向上を評価していただいた．

今後，特に本コードのように多数回のジョブを投入するコードに対しては，コード単体の性能のみではなく，ジョブのスループット全体を考慮していかなければ，高速化作業に対する更なるユーザの満足度向上は得られないことを痛感した．

Table 2.1 Calculation time of sander for scalar-single and 1-4 scalar-parallel by using `Machine.fuj_ap_frt` and `Machine.fuj_ap_f77` on AP3000.

※ ユーザ実行環境

	`Machine.fuj_ap_frt`	`Machine.fuj_ap_f77`
	実時間	実時間
シングル実行	7185 sec	7225 sec （※）
1process 並列実行	7759 sec	7844 sec
2process 並列実行	4180 sec	4249 sec
3process 並列実行	2961 sec	3027 sec
4process 並列実行	2334 sec	2389 sec

Table 2.2 Calculation time of sander for scalar-single, vector-single and 1-4 vector-parallel by using `Machine.fuj_vpp` on VPP500.

	`Machine.fuj_vpp`
	実時間
スカラ実行	26095 sec
ベクトル実行	9672 sec
1process ベクトル並列実行	caluculation impossible(CPU over)
2process ベクトル並列実行	5290 sec
3process ベクトル並列実行	3876 sec
4process ベクトル並列実行	3019 sec

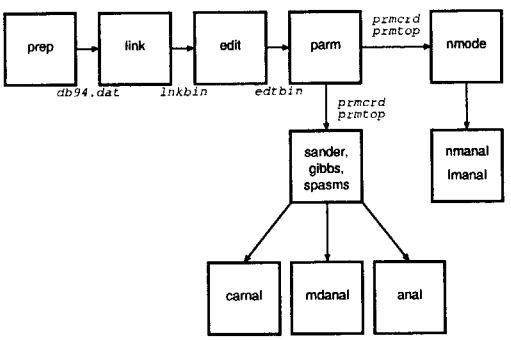

Fig. 2.1 Flow of codes in AMBER5 package.

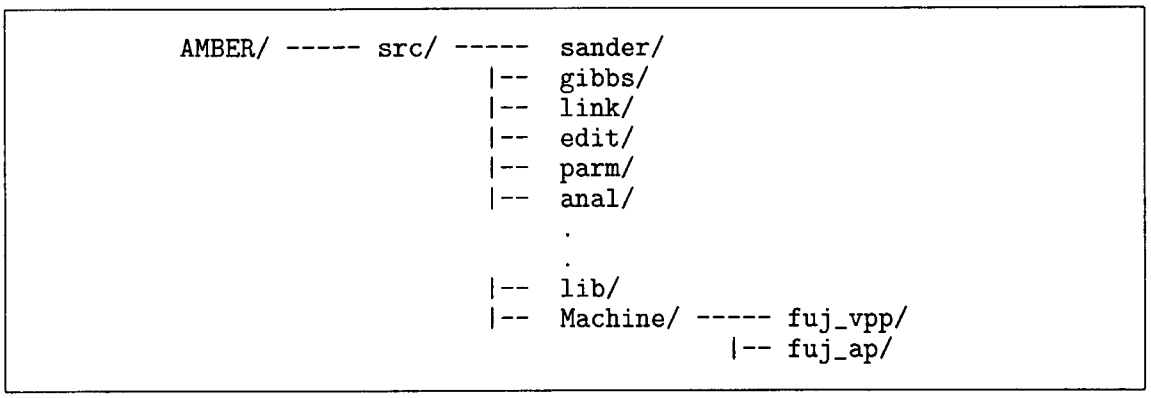

Fig. 2.2 Directory structure of AMBER5 package.

```
#! /bin/csh -v

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DISTAR2  -DEWALD "
setenv LOAD "frt -fw -Kdalign,ULTRA,V8PLUS,eval,preex,preload,fuse,loop"
setenv LOADLIB " -lm "
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
setenv MACHINEFLAGS "-DISTAR2  -DEWALD -DMPI -DMPI_NATIVE -DPROFILE \
-I/opt/FSUNmpiap/include"
setenv LOAD "frt -fw -Kdalign,ULTRA,V8PLUS,eval,preex,preload,fuse, \
loop"
setenv LOADLIB " -lm -L/opt/FSUNmpiap/lib -lmpi -L/opt/FSUNaprun/lib \
-lemi -lmpl"
# -------- END PARALLEL FLAGS --------

setenv MACHINE "Sun SPARC"
setenv MACH SPARC

setenv CCCMD fcc

# CPP is the cpp for this machine
setenv CPP /usr/ccs/lib/cpp

# SYSDIR is the name of the system-specific source directory relative
# to src/*/
setenv SYSDIR Machine/bsd

# little or no optimization:
setenv L0 "frt -c -fw -O1 -Kdalign,ULTRA,V8PLUS"

# modest optimization (local scalar):
setenv L1 "frt -c -fw -O2 -Kdalign,ULTRA,V8PLUS"

# high scalar optimization (but not vectorization):
setenv L2 "frt -c -fw -O3 -Kdalign,ULTRA,V8PLUS,eval,preex,preload, \
fuse,loop"

# high optimization (may be vectorization, not parallelization):
setenv L3 "frt -c -fw -O3 -Kdalign,ULTRA,V8PLUS,eval,preex,preload, \
fuse,loop"

# ranlib, if it exists
setenv RANLIB ranlib

# spasms configuration
#SPASMS MACHINE.sparc_spasms_config
#SPASMS unix
#SPASMS n390
#SPASMS large
```

Fig. 2.3 Machine.fuj_ap_frt.

```
#! /bin/csh -v

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DISTAR2  -DMEM_ALLOC -DEWALD "
setenv LOADLIB " -lm"
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
#setenv LOADLIB " -lm -Bdynamic -L/opt/FSUNmpiap/lib -lmpi  \
#-L/opt/FSUNaprun/lib -lemi -lmpl"
#setenv MACHINEFLAGS "-DISTAR2  -DEWALD -DMPI -DMPI_NATIVE  \
#-DPROFILE -I/opt/FSUNmpiap/include"
# -------- END PARALLEL FLAGS --------

# MACHINE FLAGS
setenv MACHINE "Sun SPARC"
setenv MACH SPARC

# LOADER/LINKER:
setenv LOAD "f77 -Bstatic -fast -xtarget=ultra -xarch=v8plus"

# CPP is the cpp for this machine
setenv CPP /usr/ccs/lib/cpp

# SYSDIR is the name of the system-specific source directory relative
# to src/*/
setenv SYSDIR Machine/bsd

# COMPILER ALIASES:

# little or no optimization:
setenv L0 "f77 -c -N150 -fast -O1 -xtarget=ultra -xarch=v8plus "

# modest optimization (local scalar):
setenv L1 "f77 -c -N150 -fast -O2  -xtarget=ultra -xarch=v8plus"

# high scalar optimization (but not vectorization):
setenv L2 "f77 -c -N150 -fast -O3  -xtarget=ultra -xarch=v8plus"

# high optimization (may be vectorization, not parallelization):
setenv L3 "f77 -c -N150 -fast -O3 -xtarget=ultra -xarch=v8plus"

# ranlib, if it exists
setenv RANLIB ranlib

# spasms configuration
#SPASMS MACHINE.sparc_spasms_config
#SPASMS unix
#SPASMS n390
#SPASMS large
```

Fig. 2.4 Machine.fuj_ap_f77.

```
# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DISTAR2  -DMEM_ALLOC -DEWALD "

                    ↓ -DMEM_ALLOC を削除

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DISTAR2  -DEWALD "
```

Fig. 2.5 Modified statement in `Machine.fuj_ap_f77`.

```
#! /bin/csh

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER "
setenv LOAD "frt -Wv"
setenv LOADLIB " -lm -lssl2vp"
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
#setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER  \
#-DMPI -DMPI_NATIVE -I/usr/lang/mpi/include"
#setenv LOAD "frt -Wv -Wl,-P"
#setenv LOADLIB " -lm -lssl2vp -L/usr/lang/mpi/lib -lmpi -lmp"
# -------- END PARALLEL FLAGS --------

setenv MACHINE "Fujitsu VPP300 series"
setenv MACH "UXP"

setenv CPP "/lib/cpp"

# SYSDIR is the name of the system-specific source directory for  \
# makemake
setenv SYSDIR Machine/fuj_vpp

# COMPILER ALIASES:

# little or no optimization:
setenv L0 "frt -c -Sw -Ab -w -On -Wv,-Of,-te,-ilfunc"

# modest optimization (typical local scalar):
setenv L1 "frt -c -Sw -Ab -w -Ob -Wv,-Of,-te,-ilfunc"

# high scalar optimization (but not vectorization):
setenv L2 "frt -c -Sw -Ab -w -Oe -Wv,-Of,-te,-ilfunc"

# high optimization (may be vectorization, not parallelization):
setenv L3 "frt -c -Sw -Ab -w -Oe -Wv,-Of,-te,-ilfunc"

# ranlib, if it exists
setenv RANLIB "echo no ranlib"

# spasms configuration
#SPASMS MACHINE.fujitsu_spasms_config
#SPASMS unix
#SPASMS n390
#SPASMS large
```

Fig. 2.6 Machine.fuj_vpp.

```
#! /bin/csh

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER "
setenv LOAD "frtpx -Wv"
setenv LOADLIB " -lm -lssl2vp"
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER  \
-DMPI -DMPI_NATIVE -I/usr/lang/mpi/include"
setenv LOAD "frtpx -Wv -Wl,-P"
setenv LOADLIB " -lm -lssl2vp -L/usr/lang/mpi/lib -lmpi -lmp"
# -------- END PARALLEL FLAGS --------

setenv MACHINE "Fujitsu VPP300 series"
setenv MACH "UXP"

setenv CPP "/lib/cpp"

setenv CCCMD fccpx

setenv L0 "frtpx -c -Of -Wv,-te"
setenv L1 "frtpx -c -Of -Wv,-te"
setenv L2 "frtpx -c -Of -Wv,-te"
setenv L3 "frtpx -c -Of -Wv,-te"

# SYSDIR is the name of the system-specific source directory for  \
# makemake
setenv SYSDIR Machine/fuj_vpp

# COMPILER ALIASES:

# little or no optimization:
#setenv L0 "frtpx -c -Sw -Ab -w -On -Wv,-Of,-te,-ilfunc"

# modest optimization (typical local scalar):
#setenv L1 "frtpx -c -Sw -Ab -w -Ob -Wv,-Of,-te,-ilfunc"

# high scalar optimization (but not vectorization):
#setenv L2 "frtpx -c -Sw -Ab -w -Oe -Wv,-Of,-te,-ilfunc"

# high optimization (may be vectorization, not parallelization):
#setenv L3 "frtpx -c -Sw -Ab -w -Oe -Wv,-Of,-te,-ilfunc"

# ranlib, if it exists
setenv RANLIB "echo no ranlib"

# spasms configuration
#SPASMS MACHINE.fujitsu_spasms_config
#SPASMS unix
#SPASMS n390
#SPASMS large
```

Fig. 2.7 Modified `Machine.fuj_vpp` for VPP500.

```
                .
                .
                .
                .

#ifdef MPI
     .......
     .......
#else
#ifdef VECTOR
     .......
     .......
#else
     .......
     .......
#endif
#endif
                .
                .
                .
                .

#ifdef VECTOR
     .......
     .......
#endif

#ifdef MPI
     .......
     .......
#endif
                .
                .
                .
                .
```

Fig. 2.8 Example of symbol statements in original sander code.

```
# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER "
setenv LOAD "frtpx -Wv"
setenv LOADLIB " -lm -lssl2vp"
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DFUJFFT -DVECTOR -DOTHER  \
-DMPI -DMPI_NATIVE -I/usr/lang/mpi/include"
setenv LOAD "frtpx -Wv -Wl,-P"
setenv LOADLIB " -lm -lssl2vp -L/usr/lang/mpi/lib -lmpi -lmp"
# -------- END PARALLEL FLAGS --------

                    ↓ -DFUJFFT -DVECTOR を削除

# ------- SEQUENTIAL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DOTHER"
setenv LOAD "frtpx -Wv"
setenv LOADLIB " -lm -lssl2vp"
# -------- END SEQUENTIAL FLAGS --------
#
# ------- PARALLEL FLAGS ----------
setenv MACHINEFLAGS "-DREGNML -DEWALD -DOTHER -DMPI -DMPI_NATIVE  \
-I/usr/lang/mpi/include"
setenv LOAD "frtpx -Wv -Wl,-P"
setenv LOADLIB " -lm -lssl2vp -L/usr/lang/mpi/lib -lmpi -lmp"
# -------- END PARALLEL FLAGS --------
```

Fig. 2.9 Modified statements in `Machine.fuj_vpp`.

```
#!/bin/csh -f

set joblim=2
set sleept=100
set no=1

qsub -e AAA.err.$no Run.sh.$no

while(1)

if(-f AAA.err.$no) then
  grep SIGXCPU AAA.err.$no
  if($status) then
    @ no++
    qsub -e AAA.err.$no Run.sh.$no
  else
    echo ''----- Execution error\!\! -----''
    exit(-1)
  endif
endif

if($no == $joblim) then
  exit(-1)
endif

sleep $sleept

end
```

Fig. 2.10 Shell script to submit job automatically for restart.

参 考 文 献

[1] http://www.amber.ucsf.edu/amber, AMBER infomation - The program package - AMBER5 Manuals.

[2] http://consult.tokai.jaeri.go.jp/vpp500/vpp500.html, VPP500利用のページ - VPP500システムのバッチジョブの連続投入方法.

3. MVP/GMVP コードの AP3000 への整備

連続エネルギー法及び多群法に基づく汎用中性子・光子輸送計算モンテカルロコード MVP/GMVP (General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations based on Continuous Energy and Multigroup Methods) の整備作業を行った．本コードは事象駆動型アルゴリズムを用いてベクトル計算機に対応しており，もちろんスカラー計算機にも対応している．従って，種々のワークステーションへのインストールが可能である．また，並列計算機にも対応しており，PVM や MPI といった一般的な並列化ライブラリを用いた並列計算も可能である．しかし MPI ライブラリを用いた並列計算は，Intel 社製の分散メモリ型スカラー超並列計算機システム Paragon(以下，Paragon と呼ぶ) では正常に動作しているが，富士通製分散メモリ型並列サーバ AP3000(以下，AP3000 と呼ぶ) 上では正常に動作していなかった．実行時のエラーメッセージが何も出力されず出力途中で終了していた．

本整備作業では，上記問題点の原因調査を含め，MVP/GMVP が AP3000 上で正常に動作するまでの整備作業を行った．以下に，これらの作業内容について報告する．

3.1 コード概要

従来のスカラーモンテカルロコードは一つの粒子の起こす事象を逐次追跡し，そのヒストリーが終わった時，次の新しい粒子の追跡を始めるというヒストリー駆動型アルゴリズムを用いているためベクトル計算機上での高速化が期待できなかった．これに対し，MVP/GMVP コード [1] は事象駆動型アルゴリズムを用いており，ベクトル計算機に対応している．その方法にはいくつかあるが，本コードではスタック駆動領域選択型アルゴリズムを用いている．この方法では，多数の粒子を同時に追跡の対象としている (同時追跡の単位をバッチと呼ぶ)．並列計算では，このバッチ単位に複数の CPU により分割処理されるよう考慮されている．機能概要は以下の通りである．

(1) 対象問題

固有値及び固定源問題を解くことができる．また，時間依存問題も解くことができる．

(2) 扱う粒子

中性子，光子，またそれらを同時に取り扱うことも可能である．また，電子の制動放射による光子も扱うことができる．

(3) 幾何形状

形状表現には組み合わせ形状 (Combinatorial Geometry) を用いている．

(4) 断面積

GMVP コードでは ANISN 型 P_l 定数または二重微分型断面積が使用できる．MVP では専用の核データライブラリーを使う．現在 JENDL-3.1，JENDL-3.2，ENDF/B-VI，JEF-2.2 の核データが利用可能である．なお，非分離共鳴の断面積は確立テーブル法を用いて表している．

(5) 境界条件

真空，完全反射，等方反射条件が使用できる．

(6) 分散低減法

Russian roulette kill と splitting，及びそれらを用いた importance sampling， weight window 法が利用できる．

(7) 評価法

粒子束は track length， collision による評価が可能である．また，固有値は track length, collision， analog estimator 及び中性子バランス法でそれぞれ計算され，最尤法によりこれらを組み合わせて最確値と分散が出力される．

3.2 AP3000 へのインストール

最新のソースプログラムファイル，確認用入力データファイル，及び AP3000 用実行シェルスクリプトファイル等を入手して， MVP/GMVP のコンパイル環境，及び実行環境を作成した．作成にあたっては，インストール等の説明資料 [2–4] を参考にした．

(1) アーカイブファイルの展開

アーカイブファイル mvp2.0beta3.10a.tar.gz を入手してこれを展開した．このファイルはソースプログラムファイルとコンパイル環境を含んでいる．実際の展開はホームディレクトリに MVP.GMVP ディレクトリを作成し，この中で実行した．展開するために使用したコマンドは以下の通りである．これによりディレクトリ mvp-2.0beta が作成され，この中にソースプログラムファイル群を含むディレクトリ src， README ファイル， INSTALL.guide ファイル等が作成された．

```
% gunzip  mvp2.0beta3.10a.tar.gz
% tar xvf mvp2.0beta3.10a.tar
```

(2) 環境変数の設定

MVP/GMVP コードのコンパイル，実行には環境変数 MVP_DIR， MVPHOST， MVPLIB_DIR を指定しておく必要がある． MVP_DIR は MVP/GMVP ファイルが格納されているディレクトリのパス名であり， MVPHOST は機種タイプ， MVPLIB_DIR は断面積ライブラリの検索ファイルが格納されているディレクトリのパス名である． .cshrc ファイルに指定した内容を Fig. 3.1に示す．

(3)make 環境の作成

makemake コマンドを使用して， AP3000 の MPI 版用のコンパイル環境を作成した．ディレクトリ mvp-2.0beta/src/OBJ.AP3K.mpi が作成され，これには，メイクファイル，ソースプログラムファイル及びオブジェクトプログラムファイルを格納するディレクトリ群が格納される．この他に実行可能プログラムファイルが格納されるディレクトリ mvp-2.0beta/bin/AP3K

が作成された．makemake コマンドの実体は，「(1) アーカイブファイルの展開」の段階でディレクトリ mvp-2.0beta/etc に作成されている．

(4) 実行可能プログラムの作成

MVP/GMVP の実行可能プログラムを作成するには，vmake コマンドを使用する．AP3000 の MPI 版を作成するには，ディレクトリ mvp-2.0beta/src/OBJ.AP3K.mpi において，以下のように vmake コマンドを投入する．

```
% vmake .mpi mvp   (MVP の場合)
% vmake .mpi gmvp  (GMVP の場合)
```

vmake の中で AP3000 の MPI 版のソースプログラムが生成され，make が実行されることにより，ディレクトリ mvp-2.0beta/bin/AP3K に実行可能プログラムが作成される．オリジナルのプログラムソースは mvp-2.0beta/src 内の対応ディレクトリに格納されているが，AP3000 の MPI 版ソースプログラムは，mvp-2.0beta/src/OBJ.AP3K.mpi の対応ディレクトリに格納される．以下のような対応関係となる．

```
(a)mvp-2.0beta/src/mvp     →  mvp-2.0beta/src/OBJ.AP3K.mpi/mvp
(b)mvp-2.0beta/src/shared  →  mvp-2.0beta/src/OBJ.AP3K.mpi/shared
(c)mvp-2.0beta/src/utils   →  mvp-2.0beta/src/OBJ.AP3K.mpi/utils
(d)mvp-2.0beta/src/artcore →  mvp-2.0beta/src/OBJ.AP3K.mpi/artcore
(e)mvp-2.0beta/src/gmvp    →  mvp-2.0beta/src/OBJ.AP3K.mpi/gmvp
```

MVP の場合は上記 (a) ～ (d) であり，GMVP の場合は (b)，(c)，(e) である．また，実行可能プログラムは MVP の場合 mvp.mpi が，GMVP の場合は gmvp.mpi が作成される．

(5) 実行環境の作成

ディレクトリ MVP.GMVP に MVP と GMVP のそれぞれの実行用ディレクトリを作成して，入手した入力データ，実行シェルスクリプトを格納した．実行シェルスクリプトはいずれも，入力データのファイル名，キュークラス名，PE 数を会話形式で取得し，MVP と GMVP の実行をバッチ形式で実行させるものである．各実行環境を以下に示す．また実行にあたっては，MVP，GMVP のいずれの場合も入力データ中の変数 NPE の値を使用プロセス数に設定して行った．例えば，3 プロセスによる並列実行の場合には，NPE=3 とした．

	MVP の場合	GMVP の場合
実行ディレクトリ名	Go.mvp	Go.gmvp
入力データ	pwr01.test.inp	gmvp.mpi.test.inp
実行シェルスクリプト	runmvp.aprun.sh	rungmvp.aprun.sh

MVP/GMVP インストール後の主なディレクトリ構成を Fig. 3.2に示す．

3.3 問題点

本整備作業の中で検出した問題点を以下に示す．並列実行という観点では，MVP と GMVP はプログラム的に全く同様の構造をしているため，いずれも同様の問題を持っている．問題点 (1) は本作業を行うきっかけとなったものである．MVP，GMVP からのエラーメッセージは何も出力されず止まってしまった．実際には，入力データをモニタ出力中に CPU タイムアウト (SIGXCPU) が発生していた．この他のものは，これを解決し，1～4 プロセスの並列実行の正常動作を確認するまでに検出した問題点である．

(1) 入力データのモニタ出力中の CPU タイムアウト
(2) ルートプロセスにおけるプリント出力の途中終了
(3) 各プロセスによるプリント出力の混在出力

以下では，MVP の場合を例にして，各問題点の内容を記述する．

3.3.1 入力データのモニタ出力中の CPU タイムアウト

実行シェルスクリプト runmvp.aprun.sh を利用して，プロセス数を 4 として MVP を起動したところ，入力データを標準出力にモニタ出力しているところで，以下のメッセージを出力して異常終了した．標準出力には全 853 行のうち 838 行が出力されており，CPU タイムアウトで終了していた．

```
jwe0017i-u The program was terminated was signal number SIGXCPU.
```

MVP における入力データの読み込み方法は，まず実行シェルスクリプト runmvp.aprun.sh の中で指定した I/O 機番とそのファイル名の対，及びデフォルトの I/O 機番の並びと入力データファイル pwr01.test.inp の内容とを，この順に結合して新たなファイル $$.MVPinp を作成し，そして，このファイルを入力リダイレクションにより標準入力とする方法である．問題発生時の出力状況から判ることは，ランク 0 のプロセスにおいてこの標準入力からデータを読み込み，標準出力にモニタ出力しているということである．MVP における入力データの取り込み方法を Fig. 3.3 に示す．この方式は，GMVP の場合も全く同じである．

まず，どこで CPU タイムアウトが発生しているかを調査するため，デバッグ WRITE 文を挿入して絞り込み作業を行った．その結果，判ったことは，ランク 0 のプロセスにおける標準入力からの読み込み処理で end 指定子の処理 (read(INP,'(A)',end=120)(Fig. 3.3 参照)) が実行されていないことである．ラベル 120 以降が実行されていなかった．入力データの最後尾に "/END" レコードを追加して，これを最終レコードとして陽に意識して入力を終了するようプログラムを変更すると，ラベル 120 以降は実行された．MPI_BCAST 処理からランク 0 ～3 の標準出力へのプリント出力までの正常動作が確認できた．一方，aprun コマンドによる並列プログラム実行環境におけるリダイレクションしたファイルの終了指定子処理を調査したところ，実装上の問題があり，仕様制約となっていることが判った．

3.3.2 ルートプロセスにおけるプリント出力の途中終了

3.3.1のCPUタイムアウト問題の解決後，1プロセスによる実行の動作確認を行ったところ，標準出力へのプリント出力がランダム・ウォークの経過情報('MONTE CARLO RUN'部)までしか出力されなかった．後に続く'RESULT OF EIGENVALUE CALCULATION'部から'EVENT MONITOR'部までは出力されなかった．AP3000の場合，並列プロセス起動コマンド"aprun"の標準出力，及び標準エラー出力への出力が保証されるのは，最初のMPIライブラリ呼出し(MPI_INIT())からMPI環境の終了処理(MPI_FINALIZE())までの間である．実際には，ACTMPPルーチン内のMONTE CARLO処理(ACTIONルーチン)後に，全てのプロセスにおいてMPI_FINALIZE()が呼ばれていた．ランク0以外のプロセスは，ここで処理を終了(STOP文の実行)するので問題ないが，ランク0のプロセスは，この後TALLY DATAの解析と出力処理(CADENZルーチン)が続くため，プリント出力が途中で終了してしまうことになった．

3.3.3 各プロセスによるプリント出力の混在出力

3.3.2の解決後，2プロセスによる実行の動作確認を行った．実行は正常終了したが，標準出力へのプリント出力がランク0のプロセスのものとランク1のプロセスのものとが入り交じって出力されていた．

並列実行を阻害するというほどではないが，各プロセスから出力されるプリント出力が一つのファイルに混在して出力されると見にくくなる．この部分はどのプロセスからの出力かということを判別しながら見なければならないからである．AP3000の場合，標準出力は起動した各プロセス毎に接続され，各プロセスに共通な一つのファイルを形成する．これに対してMVP及びGMVPは，各プロセスがそれぞれ独立に標準出力にプリント出力しているため，その出力結果が混在することになってしまった．

3.4 修正内容

「3.3 問題点」の各項目ごとにその修正内容を以下に示す．

3.4.1 入力データのモニタ出力中のCPUタイムアウト

本問題の原因は，並列プログラムの実行コマンドaprunの環境下では，入力リダイレクションしたファイルの終端子が検出できないというところにある．この問題を回避するには，次の二つの方法がある．

(1) 入力データで対処する．
終了レコード(例えば，/ENDレコード)を設けて，プログラムで終了判定する．
(2) 入力データファイルを入力リダイレクションせず，通常のファイルとする．
open文でオープンして入力処理を行なう．

入力データで対処する方法は，処置としては簡単であるが，利用する上で他計算機との統一性に欠ける．MVP/GMVP は種々の計算機に対応しており，AP3000 を利用する時に特殊なところがあるのは好ましくない．従って，修正方法としては，利用インタフェースに影響しない (2) の方法を選択することにした．修正箇所は，実行シェルスクリプトと入力データファイルの読み込み処理である．実行シェルスクリプトの修正は，最終的な入力データファイル `$$.MVPinp` (または，`$$.GMVPinp`) を入力リダイレクションから MVP(または，GMVP) の実行時オプションへの変更である．入力データファイルの読み込み処理の修正は，実行時オプションで指定された入力データファイルの名前を取り込み，オープンして，処理後クローズする処理の追加である．ここで言う入力データファイルの読み込み処理は，ディレクトリ `shared` の `inplst.f` ファイルの処理であり，MVP と GMVP に共通である．実行シェルスクリプトの修正内容を Fig. 3.4 に示す．また，サブルーチン `INPLST` の修正内容を Fig. 3.5に示す．

3.4.2 ルートプロセスにおけるプリント出力の途中終了

本問題は，ルートプロセスにおけるプリント出力の処理がまだ続くにもかかわらず，本プロセスの MPI 終了処理 (`MPI_FINALIZE`) を行ったために生じた．従って，ルートプロセスの MPI 終了処理を最終のプリント出力処理以降に行うよう修正した．MVP と GMVP の関連箇所の処理構造が似ているため，全く同様の修正を行った．修正箇所は MVP，GMVP それぞれの二つのルーチン (`CENTER`，`ACTMPP`) である．Fig. 3.6 に MPI 終了処理の位置，及び本問題の修正内容を示す．

3.4.3 プリント出力の混在出力

プリント出力が混在するのは，各プロセスがそれぞれ別個に，標準出力に出力しているからである．従って，混在出力を回避するには，標準出力にプリント出力するプロセスを一本に絞るか各プロセス対応に出力先を分けるかである．出力プロセスの一本化は出力箇所が多数あり，修正量，処理効率の面で好ましくない．ここでは，出力先を分離する方式を選択した．

MVP，及び GMVP における標準出力への出力は，FORTRAN の場合は `WRITE` 文により行われており，C 言語の場合には `printf` 関数により行われている．`WRITE` 文の場合，出力先の装置参照番号の指定は大部分が共通ブロック内の機番変数を利用しており，固定数値 (=6) を直接指定しているのは一部である．`printf` 関数の場合はこの関数自体が標準出力への出力関数であり，ここではコンパイル日付・時間を出力している．従って，修正点は以下の三つである．

- 各プロセス対応に共通ブロック内機番変数の初期化，及びプリント出力ファイルのオープンを行う．
- プリント出力ファイルの装置参照番号指定には共通ブロック内機番変数を使用する．
- コンパイル日付・時間をプリント出力ファイルに出力する．

(1) 機番変数の初期化，及びプリント出力ファイルのオープン

プリント出力ファイルの装置参照番号はインクルードファイル_IOUNIT(shared/INC ディレクトリ内) の中で固定的に 6 と定義されている．修正にあたっては，プリント出力ファイルを指定しているこれらの機番変数 IPR， IOG， IOW， IOT， IOF を共通変数として定義し，各プロセス内で使用するプリント出力ファイルの装置参照番号の初期値 IPRxx を定義する．そしてサブルーチン TSKMPI(shared ディレクトリ内) の中で， MPI 初期化後のランクを利用して各プロセス対応のプリント出力ファイルの装置参照番号を決定し，オープンする． TSKMPI の中では，上記の機番変数の他にプリント出力ファイルに利用されている IUPR についても定義した．修正内容を Fig. 3.7 に示す．

(2) 装置参照番号指定の固定数値から機番変数への変更

プリント出力ファイルへの出力時，固定数値 (=6) を直接指定しているのは，以下の 4 ルーチンである．

- mvp　ディレクトリ ： CADENZ　　PRMNTR
- shared ディレクトリ ： USRSRC
- utilis ディレクトリ ： JFOPEN

修正は，大体が以下のような include 文の挿入と write 文の変更である．但し， PRMNTR は HEADER と LABEL に指定する値に 6 を指定しているので機番変数に変更した．

```
include 'INC/_IOUNIT' (または， include '../shared/INC/_IOUNIT')
・・・
write(IPR,・・・)
```

(3) コンパイル日付・時間のプリント出力ファイルへの出力

C の関数 PRDATE の中では，コンパイル日付・時間を得てそれらを printf 関数で出力している．これを修正するため， PRDATE 関数の中ではコンパイル日付・時間を得るだけにして，それらを呼出し元のサブルーチン STAMP に引き継ぎ，そこでプリント出力ファイルに出力するようにした． STAMP はディレクトリ shared に存在するが， PRDATE はディレクトリ mvp と gmvp に存在する． Fig. 3.8 に変更内容を示す．

3.5　動作確認

3.2 (5) 実行環境の作成 で記した実行環境，実行方法に基づき MVP， GMVP を実行した．いずれの場合も，実行プロセス数が 1， 2， 3， 4 の 4 ケースについて実施した．確認については， Paragon の出力結果と比較検討することにより行った．概ね同様の出力結果を得ることができたことで，問題無しと判断した．

3.6 まとめ

本作業の目的は，AP3000 において MVP/GMVP の並列実行を実現することであった．

調査の結果，並列実行を阻害する最大の要因は，並列実行環境におけるリダイレクションしたファイルの終了指定子処理にあることが判り，これを回避するよう対処した．この時，解析に利用したテストケースは MVP，GMVP ともそれぞれ一つだけであったが，プロセス間の通信処理をほぼ網羅していたため，並列実行の実現という基本的な問題を解決するには十分であった．

```
#
# --- MVP/GMVP configuration ---
#
setenv MVP_DIR /dgXX/g0XXX/jXXXX/MVP.GMVP/mvp-2.0beta
setenv MVPHOST AP3K
setenv MVPLIB_DIR /dg02/g0435/j3803/pub/MVPlib/INDEX
       if( $?path ) then
           set path=($MVP_DIR/etc $MVP_DIR/bin/$MVPHOST $path)
       else
           set path=($MVP_DIR/etc $MVP_DIR/bin/$MVPHOST)
       endif


       注:/dgXX/g0XXX/jXXX/ はホームディレクトリの絶対パス名
```

Fig. 3.1 Definition of environment variables MVP_DIR,MVPHOST and MVPLIB_DIR.

```
$HOME
  |
  |
  +-MVP.GMVP-+-Go.mvp    ---+-pwr01.test.in     (f)
  |          |               +-runmvp.aprun.sh  (f)
  |          |
  :          +-Go.gmvp   ---+-gmvp.mpi.test.inp(f)
  |          |               +-rungmvp.aprun.sh (f)
  |          |
  |          +-mvp-2.0beta-+-INSTALL.guide     (f)
  |                        +-README            (f)
  |                        +-bin  ---AP3K-------+--fat      (f)
  |                        |                    +--mvp.mpi  (f)
  |                        |                    +--gmvp.mpi(f)
  |                        +-etc--+--makemake (f)
  |                        |      +--vmake    (f)
  |                        |      :
  |                        |
  |                        +-src--+--OBJ.AP3K.mpi--+--mvp
  |                        |      |                 +--shared
  |                        :      +--mvp            +--utils
  |                               +-shared          +--artcore
  |                               +-utils           +--gmvp
  |                               +-artcore         :
  |                               +-gmvp
  |                               +-tools
  |                               :
  注：$HOME はホームディレクトリ
      (f) はファイルを表示
```

Fig. 3.2 Directory structure in MVP/GMVP.

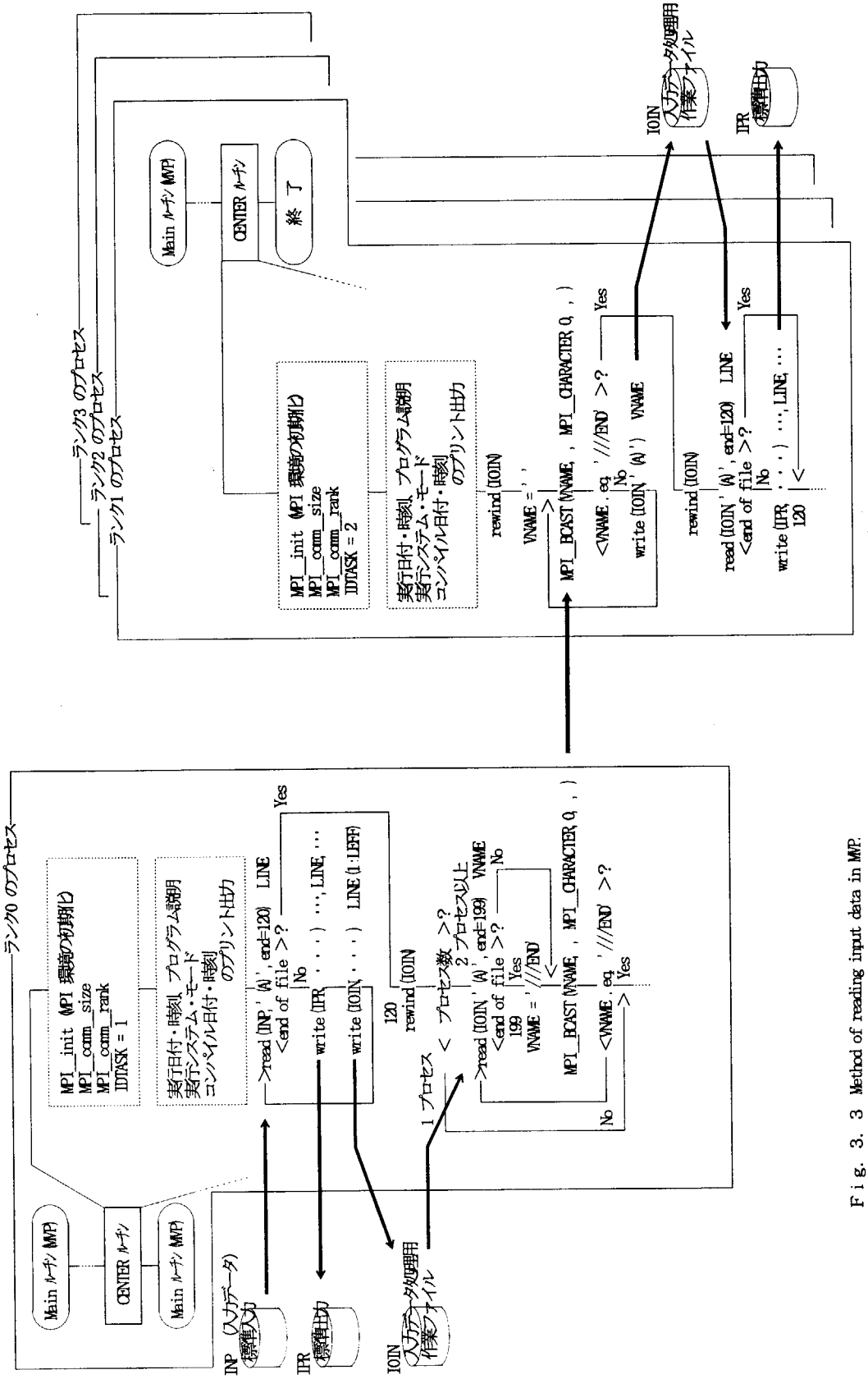

Fig. 3. 3 Method of reading input data in MVP.

MVPの場合

```
#!/bin/sh
#
 ・・・
if [ $# -gt 0 ] ; then
 ・・・
else
   echo " MVP input file -> "
   read INP
   echo " Q-class  | PE |  CPU/elapse  | memory     "
   echo " pd       |  4 | 10min/20min  | 2000MB/1PE "
   echo " p        |  4 |    6h/12h    | 2000MB/1PE "
   echo " input queue class -> "
   read Q
   echo " number of PEs -> "
   read NP
fi
 ・・・

cat > $$.mpish << END_SH
#!/bin/sh
#
MVP_DIR=/dg02/g0435/j5028/mvp2.0b3.10a ; export MVP_DIR
MVPLIB_DIR=/dg02/g0435/j3803/pub/MVPlib/INDEX ; export MVPLIB_DIR
MVPHOST=AP3K ; export MVPHOST

cd ¥$QSUB_WORKDIR

INP=$INP
EXEC=¥$MVP_DIR/bin/¥$MVPHOST/mvp.mpi
OUT=`basename ¥$INP .inp`.¥$MVPHOST.MPI.PE$NP.`date +%b%d.%H.%M.%S`

cat - ¥$INP << END_PRE > $$.MVPinp
@PREINP
/24f ¥$MVPLIB_DIR/neutron.art.index
 ・・・
/10 /20 /30 /55f /15f /16 /75f
@END
END_PRE                                  修正前
aprun -nproc $NP -ult ¥$EXEC < $$.MVPinp > ¥$OUT

END_SH

qsub -q $Q -lP $NP -C MVP -r MVP.PE$NP -eo $$.mpish
```

修正後

```
aprun -nproc $NP -ult ¥$EXEC -- $$.MVPinp > ¥$OUT
```

GMVPの場合

```
#!/bin/sh
#
 ・・・
if [ $# -gt 0 ] ; then
 ・・・
else
   echo " GMVP input file -> "
   read INP
   echo " Q-class  | PE |  CPU/elapse  | memory     "
   echo " pd       |  4 | 10min/20min  | 2000MB/1PE "
   echo " p        |  4 |    6h/12h    | 2000MB/1PE "
   echo " input queue class -> "
   read Q
   echo " number of PEs -> "
   read NP
fi
 ・・・

cat > $$.mpish << END_SH
#!/bin/sh
#
MVP_DIR=/dg02/g0435/j5028/mvp2.0b3.10a ; export MVP_DIR
MVPLIB_DIR=/dg02/g0435/j3803/pub/MVPlib/INDEX ; export MVPLIB_DIR
MVPHOST=AP3K ; export MVPHOST

cd ¥$QSUB_WORKDIR

INP=$INP
EXEC=¥$MVP_DIR/bin/¥$MVPHOST/gmvp.mpi
OUT=`basename ¥$INP .inp`.¥$MVPHOST.MPI.PE$NP.`date +%b%d.%H.%M.%S`

cat - ¥$INP << END_PRE > $$.GMVPinp
@PREINP
/9  /40 /41 /96 /97 /98 /99f
/10 /20 /30 /55f /15f /16 /75f
@END
END_PRE
                                         修正前
aprun -nproc $NP -ult ¥$EXEC < $$.GMVPinp > ¥$OUT

END_SH

qsub -q $Q -lP $NP -C GMVP -r GMVP.PE$NP -eo $$.mpis
```

修正後

```
aprun -nproc $NP -ult ¥$EXEC -- $$.GMVPinp > ¥$OUT
```

Fig. 3. 4　Modification of shell script for execution

修正前

```
  subroutine INPLST( IPR,   INP,   IOIN,  NTASKO,MLIMIT )

      .   .   .

  100 read(INP,'(A)',end =120) LINE

      .   .   .

C
      return
      end
```

修正後

```
      .   .   .
C
      character*16 filnam
      data filnam/'                '/
C
      .   .   .

      if ( INP.eq.5 ) then
         INP4 = 4
         call getarg(1,filnam)
         open( INP4,file=filnam,status='OLD', form = 'FORMATTED' )
      else
         INP4 = INP
      endif
   100 read(INP4,'(A)',end =120) LINE

      .   .   .

      if ( INP.eq.5 ) close( INP4 )
C
      return
      end
```

Fig. 3.5 Modification of subroutine "INPLST".

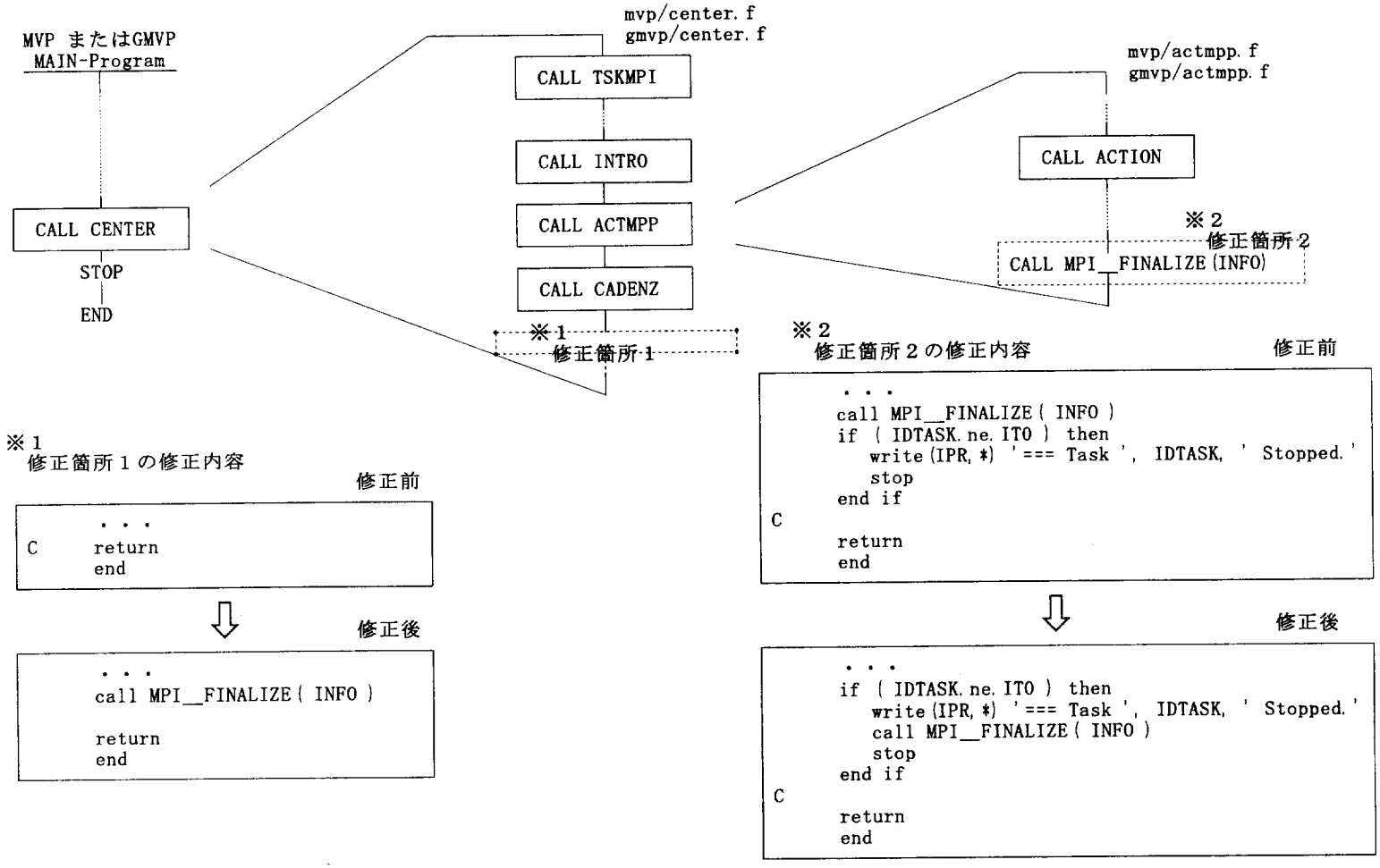

Fig. 3.6 Modification of printing process.

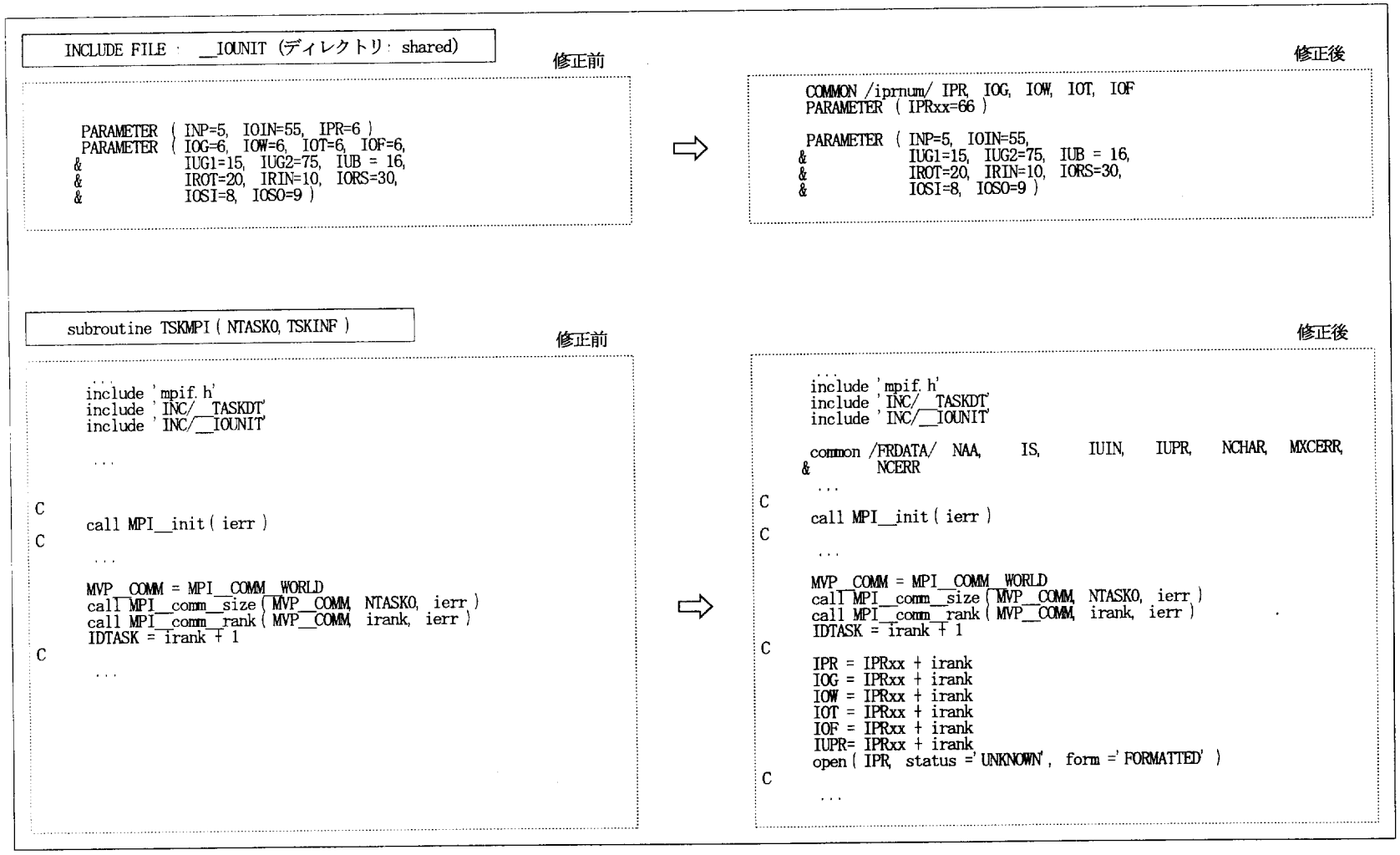

Fig. 3. 7 Additinal process of opening printout files.

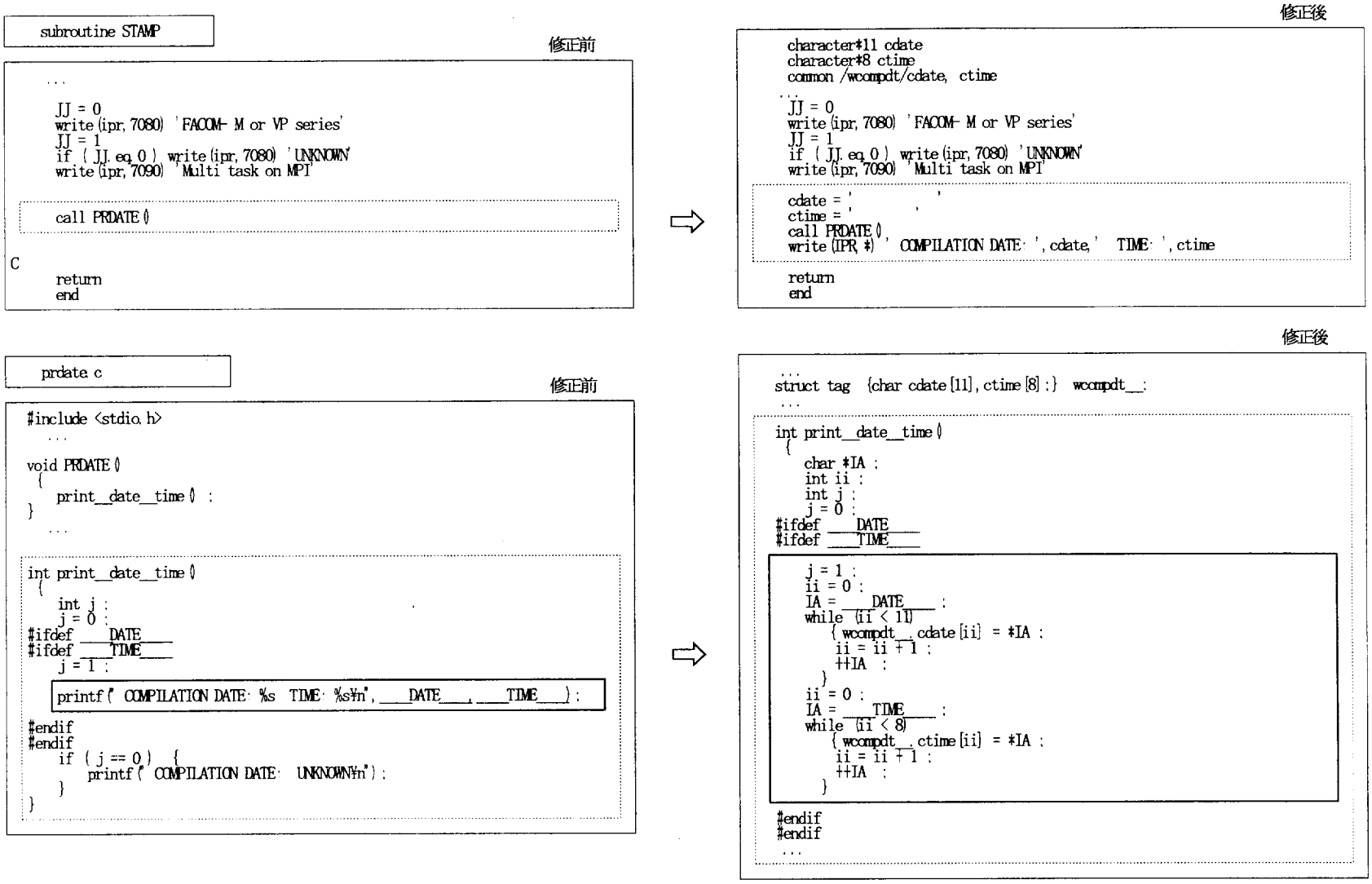

Fig. 3. 8 Modification of processing to print compilation date and time.

参 考 文 献

[1] 森　貴正・中川正幸;MVP/GMVP 連続エネルギー法及び多群法に基づく汎用中性子・光子輸送計算モンテカルロコード, JAERI-Data/Code 94-007.

[2] 長家康展;AP3000 における MVP/GMVP の並列化, 1999 年 1 月 18 日.

[3] README file (for version 2.0 preliminary 1, Oct 1998).

[4] INSTALL.guide file (to MVP/GMVP 2.0 (UNIX version), Nov 1998).

[5] 「MPI/AP V1.0 使用手引書」, 富士通 (株), 1998 年 8 月.

4. autonj システムの AP3000 への整備

MCNP ライブラリ自動編集システム autonj(Automatic Editing System for MCNP Library) [1, 2] の整備作業を行った．本システムは MCNP [3] で使用する断面積ライブラリを作成・編集するシステムであり，原子力コード評価専門部会 MCNP 高温ライブラリ作成ワーキンググループの活動の一環として作成されたものである．開発にあたっては，UNIX ワークステーションの環境で実施されることを前提に作成されており，現在対象としているワークステーションは Hewlett-Packard 社製の HP-9000/735 と Sun microsystems 社製の SUN-4 である．この二つの計算機環境において実行確認が成されている．本システムは，MCNP が利用されている計算機上で動作することにより，MCNP 利用の利便性を向上させることができる．計算科学技術推進センター (以下，計算センターと呼ぶ) で MCNP が利用されている共同利用計算機は富士通製分散メモリ型並列サーバ AP3000(以下，AP3000 と呼ぶ) であり，そのため autonj システムの AP3000 への整備作業が検討された．

本整備作業では，AP3000 向けの修正，その修正版の動作確認，及び代表的な処理条件で作成された断面積ライブラリの計算センターへの登録を行った．以下に，これらの作業内容について報告する．

4.1 システム概要

autonj は，汎用評価済核データファイルから指定された処理条件に基づき，自動的に MCNP コード用の中性子連続エネルギー断面積ライブラリを作成・編集するシステムである．このシステムは，核データ処理コード NJOY を中核とするものであり，NJOY を実施するための前処理と実施後の後処理部分から構成され，以下の五つの実行モジュールから構成される．ここで利用されている NJOY は NJOY97 であり，バージョンは NJOY97.45 である．実際には，この他にいくつかのローカルパッチが施されている．

(1) 核データファイル中の核種識別モジュール (MATLIST)

指定された核データファイル中の核種の MAT 番号とファイル名のリストテーブルを作成し，核種識別ファイルを出力する．

(2)NJOY97 入力データ作成モジュール (NJ-INP)

指定した処理条件で ACE 形式断面積ファイルを作成するための NJOY97 用入力データを核種毎に作成する．

(3)NJOY97 モジュール (NJOY97)

核種毎に作成された検索ファイル `xsdir` と ACE 形式断面積ファイル (type 1 形式) を作成する．

(4)type 1 形式編集モジュール (MKXSDIR)

NJYOY97により核種毎に作成された検索ファイルxsdirとACE形式断面積ファイル(type 1形式)から，それらを統合したtype 1形式のMCNP用連続エネルギー断面積ライブラリを作成する．

(5)MAKXSFモジュール(MAKXSF)

type 1形式のMCNP用連続エネルギー断面積ライブラリからtype 2形式のものを変換して作成する．このモジュールは，作成ライブラリの形式としてtype 2が指定された場合に実行される．

4.2 実行形態の変更

配付された圧縮ファイルを解凍することにより，以下のディレクトリとファイルが得られた．

```
njoy        : NJOY97関連のソースプログラムファイル，他の格納ディレクトリ
Utility     : NJOY97以外のソースプログラムファイル，他の格納ディレクトリ
autonj-test: 検証用入力データ，出力データの格納ディレクトリ
autonj.sh   : 実行用シェルスクリプトファイル
```

実行用シェルスクリプトファイルautonj.shは，ディレクトリnjoy，Utilityの両方を含む利用者ディレクトリの中で，実行されることを前提とした作りとなっている．また，実行の形態は，処理条件等の必要データの入力から各モジュールのコンパイル，実行までを，会話形式で行うようになっている．一方，AP3000は，それぞれが1台のSUNワークステーションに相当するノード群を結合した全体システムとして共同利用されるサーバシステムである．そのため，本システムをAP3000に移植するにあたっては，実行形態を変更した．本システムを構成するモジュール群のロードモジュールを計算センター側で作成し，共用のディレクトリに格納した．また，実行のためのシェルスクリプトファイルも共用のディレクトリに格納した．利用者は実行用シェルスクリプトファイルを利用者側の実行ディレクトリに複写して，そこでこれを実行する形態にした．

4.3 ロードモジュールの作成

各ロードモジュールを作成するシェルスクリプトファイルanjcomp.shは，autonj.shからコンパイルとリンクの部分を抽出して作成した．ロードモジュールの作成は，共用ディレクトリsrcの中で計算センター側で行ない，ここで作成した六つのロードモジュールを共用ディレクトリautonjLMに格納した．これらディレクトリ，及び格納したロードモジュールを以下に示す．

・コンパイル・リンク用共用ディレクトリ：/dg06/center/codelib/autonj/src
・ロードモジュール格納用
　　共用ディレクトリ：/dg06/center/codelib/autonj/autonjLM
・格納したロードモジュール ：

matlist_ld nj-inp_ld mk1xsd_ld mkxsdir_ld makxsf_ld xnjoy

共用ディレクトリ src の構成は，ソースプログラムを格納したディレクトリ njoy，Utility とコンパイル・リンク用シェルスクリプトファイル anjcomp.sh からなり，ここで anjcomp.sh を実行することによりロードモジュールが作成される．これらロードモジュールを共用ディレクトリ autonjLM に移送し，これを利用者側で利用する．このように実行形態を変更したことにより，ソースプログラム mk1xsd，mkxsdir に変更が必要になった．内容はいずれも同じであり，ディレクトリ Utility 中の awr-list を利用するためにオープンする処理を，以下のように変更した．

```
open(iu4,file='Utility/awr-list', ・・・ )
                        ↓
open(iu4,file='/dg06/center/codelib/autonj/src/Utility/awr-list', ・・・ )
```

autonj.sh とこれを元に作成した anjcomp.sh の内容を付録 A，付録 Bに示す．

4.4 実行用シェルスクリプトの修正

オリジナルの実行用シェルスクリプトファイル autonj.sh は HP9000/735 または Sun4 のいずれかだけを意識した作りになっており，会話形式で実行する形態になっている．従って，実行形態を「4.3 ロードモジュールの作成」で述べた形態に変更するため，autonj.sh を変更した．

まず，実行の形態は会話処理のまま，AP3000 用に修正した．次いで，共同利用の観点から，これをバッチ処理用に修正した．修正したこの二つの実行用シェルスクリプトファイルは，以下の共用ディレクトリに格納した．

・格納用共用ディレクトリ ： /dg06/center/codelib/autonj/autonjSH

4.4.1 AP3000 用会話処理シェルスクリプトへの修正

autonj.sh から AP3000 用会話処理シェルスクリプトファイル autonjC.sh への修正点は以下の 3 点である．

(1) コンパイルとリンク関係箇所の削除

コンパイルとリンク関係の箇所は anjcomp.sh に吸収済であり，その部分を削除した．

(2) 共用ディレクトリに格納されたロードモジュールの起動

共用ディレクトリに格納された各ロードモジュールファイルに対応したシェル変数を設定することにより存在チェックを行い，実行文の記述を修正した． Fig. 4.1に修正例を示す．

(3)'HP'，'SUN' 判定箇所における'HP' に対応する処理の削除

AP3000はSUNと同等であるため，HP9000/735に関する箇所を削除した．Fig. 4.2に修正例を示す．

autonjC.shの内容を付録Cに示す．

4.4.2 バッチ処理用シェルスクリプトへの修正

autonj.sh，及びautonjC.shの実体は，大きく分けて，入力データ取込み部と断面積ライブラリ作成部の二つから成る．前者は処理時間がわずかであるため，実行形態を変えず会話処理のままとした．バッチ処理の形態に変更したのは，後者の断面積ライブラリ作成部である．

autonjC.shからAP3000用バッチ処理シェルスクリプトファイルautonjB.shへの修正点は以下の2点である．

(1) キュークラス名の取込み処理の追加

バッチジョブを実行するキュークラスを把握するため，キュークラス名を会話処理で取り込む処理を追加した．入力データ取込み処理後，断面積ライブラリ作成処理に移行する前に挿入した．キュークラス名取込み処理の内容をFig. 4.3に示す．

(2) 断面積ライブラリ作成部のバッチ処理化

断面積ライブラリ作成部をバッチ処理化するため，この作成部全体をバッチジョブ用シェルスクリプトファイルgo.autonj.sh内に，catコマンドを利用して組み込んだ．この時，入力データ取込み処理部で設定されたロードモジュール対応のシェル変数は，バッチジョブに引き継がれないため，新たにgo.autonj.shの中でリンクが張られるように対処した．実際には，lnコマンドを使用した．その修正内容をFig. 4.4に示す．

また，go.autonj.shへの組込み部の中で設定されたシェル変数の置き換え変数($xxx)については，実際の値に置き換えられずにgo.autonj.shに組み込まれるように'$'機能を打ち消すためメタキャラクタ'\'を直前に挿入した．例えば，以下のような対応を行った．

```
set tt=ACE-FILE                    set tt=ACE-FILE
if ( -e $tt ) then        →        if ( -e \$tt ) then
  . . .                              . . .
```

autonjB.shの内容を付録Dに示す．

4.5 動作確認

動作確認としては，以下の二つのことを実施した．

(1)AP3000用autonjシステムの動作確認

(2)AP3000用autonjシステムにより作成した断面積ライブラリの妥当性確認

4.5.1 AP3000用autonjシステムの動作確認

AP3000用に修正した実行シェルスクリプトファイルautonjC.sh，及びautonjB.shを利用した実行確認を行った．それぞれの出力結果を比較するため，以下の入力条件をパラメータとしてautonj.sh，autonjC.sh，autonjB.shを実行した．実行テストケースをTable 4.1に示す．

［確認用実行パラメータ］

(1) 入力モード (interactive/input file)

(2) 使用ディスクサイズのオプション (minimum/medium/maximum)

(3) 作成断面積ライブラリのタイプ (type 1/type 2/type 1 and type 2)

(4) 変換対象ファイル数 (1個／複数個 (=2個))

比較対象の出力結果は，検索ファイル，type 1とtype 2の断面積ライブラリである．オリジナルのautonj.shの結果を比較原本として，これとautonjC.sh，autonjB.shの出力結果を比較した．type 2の断面積ライブラリはバイナリーファイルであるため，サイズのみを比較して，他の二つについてはdiffコマンドを利用して内容比較まで行った．確認結果はデータ内容，ファイルサイズともどのテストケースにおいても一致した．

4.5.2 AP3000用autonjシステムにより作成した断面積ライブラリの妥当性確認

AP3000に整備したautonjシステムを利用して作成した断面積ライブラリについて，MCNPによる検証計算を実行した．計算結果を利用者側に確認依頼したところ妥当である旨の回答を得た．検証計算の条件は以下の通りである．

(1) 検証計算用テストデータ (MCNPの入力データ)

提供された以下の6個のデータを使用した．

(a)fp18.mcn (TCAウラン炉心)

(b)Fe_1.mcn (FNS鉄体系クリーンベンチマーク実験)

(c)C_1.mcn (FNS黒鉛体系漏洩中性子スペクトル測定実験)

(d)Cu_1.mcn (OKTAVIAN銅球からの漏洩中性子スペクトル測定実験)

(e)Sin1.mcn (OKTAVIANシリコン球からの漏洩γ線スペクトル測定実験)

(f)Sig1.mcn ((e)におけるターゲットγ線の計算)

(2) 断面積ライブラリの作成

提供された14個のJENDL-3.2の核データファイル (圧縮ファイル) から，autonjB.shを利用して各々について検索ファイルとtype 2の断面積ライブラリを作成した．この時，断面積処理温度として300 K，pointwise断面積計算精度として0.2%を選択した．上記の検証計算1～6を行うにはこれだけでは不十分であり，これらの他に情報センターに登録してある以下の四つのtype 2の断面積ライブリを追加した．

EL-2 FSXDOS-2 MCPLIB-2 TMCCS-2

MCNP を利用するには検索ファイルを一つに纏める必要がある．そのため，以上 18 個の断面積ライブラリに対応する検索ファイル群から，vi コマンドを利用して一つの検索ファイルを作成した．検索ファイルの構成は共通部と各断面積ライブラリへの索引部であるディレクトリ部から成っており，検索ファイル群から各々のディレクトリ部を集めて編集した．この結果，1 個の検索ファイルと 14 個の type 2 の断面積ライブラリという構成にした．

(3) 妥当性確認に利用した MCNP の版数

MCNP-4B2 を利用した．またこの時，(1) の (b) のケースは処理時間が長いため並列版を使用した．他のケースについては，シングル版を使用した．

4.6 断面積ライブラリの計算科学技術推進センターへの登録

JENDL-3.2 から作成された type 1 の断面積ライブラリとその検索ファイルが 6 種類提供され，これらを MAKXSF モジュールを利用して type 2 のものに変換した．それにより作成された type 2 の断面積ライブラリとその検索ファイルを計算センターに登録した．各断面積ライブラリの作成温度点は 293，600，900，1200，1500，及び 2000 (単位:Kelvin) である．また，いずれも pointwise 断面積計算精度は 0.2% である．

登録先のディレクトリ，登録した断面積ライブラリと検索ファイルを以下に示す．

(1) 登録先のディレクトリ

/dg06/center/codelib/autonj/J32lib

(2) 登録断面積ライブラリ及び検索ファイル

- 293 K : FSXJ32A2 xsdir.FSXJ32A2
- 600 K : FSXJ32B2 xsdir.FSXJ32B2
- 900 K : FSXJ32C2 xsdir.FSXJ32C2
- 1200 K : FSXJ32D2 xsdir.FSXJ32D2
- 1500 K : FSXJ32E2 xsdir.FSXJ32E2
- 2000 K : FSXJ32F2 xsdir.FSXJ32F2

4.7 まとめ

本システムを AP3000 に整備するにあたって，以下のことを基本方針として実施した．

- いくつかの代表的な条件下の断面積ライブラリは共同利用できるよう事前に作成し，計算センターに登録する．
- それら以外の条件のものを利用者が必要とする場合には，利用者が個別に自分の作業域で作成できる．

この方針のもと，共同利用の観点からロードモジュール群を計算センターで一括管理することにした．そのため，実行シェルスクリプトからコンパイル・リンクの処理部を削除し，利用時の処理時間，格納空間の節約を図った．また，利用者の作業域において実際の実行をバッチ処理できるよう整備した．これらにより，容易に利用できる環境が提供できたものと考えている．

検証については，実施したテストケースの範囲内では問題ないことが確認された．対象とした断面積ライブラリの作成条件は1ケースだけであったが，AP3000のSunワークステーションとの同等性，及びAP3000上での本システムの動作確認結果を考慮すれば，Sunワークステーション上の品質と同程度のものがAP3000上でも得られていると言える．

Table 4.1 Test cases for running test of autonj system.

テストパラメータ				実行シェルスクリプト		
Input *mode*[*1]	Disk *size*[*2]	*Type*[*3]	*Files*[*4]	autonj.sh (オリジナル)	autonjC.sh (会話処理)	autonjB.sh (バッチ処理)
interactive	min	I, II	1	○	○	○
	省略	I		——	——	○
	med	II		——	——	○
	max	省略		——	——	○
input from file	max	I, II		——	——	○
interactive	省略	省略	2[*5]	——	○	○
interactive	省略	省略	2[*6]	——	○	○

備考 (1) 各パラメータ間には相関がないため，少なくとも最低１回は選択するようにテストケースを作成した．また，各パラメータの入力処理はいずれの実行シェルスクリプトも同じため，バッチ処理を重点的に実施した．

(2) 略号　○　実行ケース

—— 非実行ケース

*1 入力モード

*2 ディスクサイズ

min : minimum

med : medium

max : maximum

省略: minimum

*3 作成断面積ライブラリの形式（タイプ）

I : type 1

II : type 2

省略: type 2

*4 変換対象ファイル数

*5 JENDL ファイル名を直接指定

*6 JENDL ファイル名を格納したディレクトリ名を指定

修正前

```
        .   .   .

################
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
################

        .   .   .

#---------
matlist_ld
#---------
```

修正後

```
set matlist_ld=/dg06/center/codelib/autonj/autonjLM/matlist_ld

        .   .   .

if ( -e $matlist_ld && -x $matlist_ld ) then
else
echo ' Load module(matlist_ld) does not exist.'
exit
endif

        .   .   .

################
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
################
#
#---------
$matlist_ld
#---------
```

Fig. 4.1 Example of command description modification.

修正前

```
stp2:
   if ( $mach == 'HP' ) then
      uncompressdir -f $name1
   else if ( $mach == 'SUN' ) then
      uncompress -f $ {name1} /*.Z >&! autonj_dummy
   endif
```

修正後

```
stp2:

    uncompress -f $ {name1} /*.Z >&! autonj_dummy
```

Fig. 4.2 Example of 'HP' part deletion.

```
####
#### Get a name of queue class for Batch Job
####
   echo " End of input data for autonj "
   echo ' '
   echo " Q-class | CPU/elapse    | memory     "
   echo ' '
   echo " ss      |    1h/ 2h    |  200MB "
   echo " sm      |    6h/12h    |  200MB "
   echo " sl      |   48h/96h    |  200MB "
   echo ' '
   echo " ms      |    1h/ 2h    |  500MB "
   echo " mm      |    6h/12h    |  500MB "
   echo " ml      |   48h/96h    |  500MB "
   echo ' '
   echo " ls      |    1h/ 2h    | 2000MB "
   echo " lm      |    6h/12h    | 2000MB "
   echo " ll      |   48h/96h    | 2000MB "
   echo ' '
   echo ' Input a name of queue class =====> '  | tr -d '\012'
   set Q=$<

echo "class : $Q"
echo ' OK ?  (y/n) ==========> '  | tr -d '\012'
set YN=$<
if ( $YN != 'y' ) then
   if ( $YN != 'Y' ) then
      exit
   endif
endif
```

Fig. 4.3 Process of getting a name of queue class.

```
cat > go.autonj.sh << END_SH
#!/bin/csh
#

set anjLM=/dg06/center/codelib/autonj/autonjLM
ln -s \$anjLM/xnjoy       \$QSUB_WORKDIR/xnjoy
ln -s \$anjLM/matlist_ld \$QSUB_WORKDIR/matlist_ld
ln -s \$anjLM/nj-inp_ld  \$QSUB_WORKDIR/nj_inp_ld
ln -s \$anjLM/mk1xsd_ld  \$QSUB_WORKDIR/mk1xsd_ld
ln -s \$anjLM/mkxsdir_ld \$QSUB_WORKDIR/mkxsdir_ld
ln -s \$anjLM/makxsf_ld  \$QSUB_WORKDIR/makxsf_ld
#
cd \$QSUB_WORKDIR
#

   .   .   .


END_SH
```

Fig. 4.4 Link part of load modules in autonj system by ln command.

参 考 文 献

[1] 「MCNP ライブラリー自動編集システムの作成　作業報告書」，住友原子力工業（株），平成 11 年 1 月.

[2] Maekawa F.,Sakurai K.,Kosako K.,Kume E.,Kawasaki N.,Nomura Y.and Naito Y.: "Development of Automatic Editing System for MCNP Library "autonj"", JAERI-Data/Code 99-048(1999).

[3] Judith F.Briesmeister,Editor:MCNP － A General Monte Carlo N － Particle Transport Code, LA-12625-M,Version 4B Manual, March 1997.

5. SPECTER/SPECOMP コードのインストール

本作業では，SPECTER 及び SPECOMP コード [1] の富士通（株）製共用 UNIX サーバ AP3000/24 [2]（以下 AP3000）へのインストールを行った．SPECTER は，原子炉内で照射した材料の損傷計算を行うコードである．このコードには，簡易版と完全版があり，本作業では，完全版のインストールを行った．SPECOMP は，一つのプログラムであるが，SPECTER の一部のルーチンにもなるものである．両コードは，VAX-FORTRAN でコーディングされており，これを FUJITSU Fortran90 [3]（以下 Fortran90）の仕様にした．以下に，その作業について報告する．

5.1 インストール作業

ここでは，SPECTER 及び SPECOMP を AP3000 へインストールした際の作業について述べる．

5.1.1 ファイル名の割り当て

RIST (高度情報科学技術研究機構) 経由で入手した，SPECTER-ANL パッケージには，次に示すファイルが含まれていた．

NEAREAD.ME
RESTRICT.TXT
NEAINS.BAT
PKUNZIP.EXE
PSR0263.ZIP

NEAREAD.ME ファイルに ZIP ファイルの解凍方法が記述されていた．これより，NEAINS.BAT ファイルを使用し，PSR0263.ZIP ファイルの解凍を行った．その結果，PSR0263_.001 ～ PSR0263_.016 という名前のファイルが生成された．PSR0263_.001 ファイルを調査したところ，各ファイルの説明が記述されていた．各ファイルの説明を Fig. 5.1 に示す．そして，その情報をもとにファイル名の変更を行った．ここで，Fig. 5.1 より，FILE 1 ～ 3 は，次のようにファイル名の変更を行った．

FILE 1 : INFORMATION
FILE 2 : COMMAND.specter
FILE 3 : COMMAND.specomp

また，SPECTER.FOR と SPECOMP.FOR ファイルは，Fortran 原始プログラムであることがわかる．Fortran90 では，Fortran 原始プログラムのファイルのサフィックスは，”.f”

でなくてはならない．よって，それぞれ次に示すようにファイル名の変更を行った．

SPECTER.FOR : specter.f
SPECOMP.FOR : specomp.f

5.1.2 コンパイル

各ファイル (specter.f, specomp.f) のコンパイルを行い，ロードモジュールを作成した．ロードモジュールの作成は，`frt` コマンドを使用して行った．specter.f のロードモジュール名を specter，同様に，specomp.f では，specomp とした．specter.f での `frt` コマンドの使用例を Fig. 5.2 に示す．COMPOUT.specter ファイルには，コンパイル時のエラーメッセージ等の情報が出力される．

各ソースのコンパイルを行った結果，エラー及び WARNING が発生した．その箇所と回避方法を次に示す．

(1) specter.f

specter.f の 229 行めの処理において，エラーが発生した．エラーメッセージを Fig. 5.3 に示す．これは，Fortran90 で許されていない文法である `TYPE` 文を使用しているためである．ソースを解析した結果，配列 `ILEM` を出力しているようである．よって，`TYPE` 文を `WRITE` 文にし，エラーを回避した．装置識別子は，"66" を指定した．修正前と後のその箇所を Fig. 5.4 に示す．

また，129，249 行めの処理において，WARNING が発生した．WARNING メッセージを Fig. 5.5 に示す．これは，旧仕様の記述を含んでいるためである．このことに関し，調査し，テストした結果，括弧の括り方を変えることで WARNING を回避することができた．修正前と後のその箇所を Fig. 5.6 に示す．

(2) specomp.f

specomp.f の 17，23 行めの処理において，WARNING が発生した．WARNING メッセージを Fig. 5.7 に示す．これは，specter.f で発生した WARNING と同じである．よって，specter.f と同様な修正をし，WARNING を回避した．修正前と後のその箇所を Fig. 5.8 に示す．

5.2 プログラムの実行

ここでは，SPECTER 及び SPECOMP の AP3000 上での実行方法について述べる．

5.2.1 SPECTER の実行方法

通常，AP3000 では，プログラムの実行を qsub コマンドを用い，バッチジョブとして行う．この場合，バッチジョブ用にシェルスクリプトを用意しなくてはならない．今回の作業で作成した，シェルスクリプト exe-specter.sh を例にバッチジョブの実行方法を Fig. 5.9 に示す．

シェルスクリプト exe-specter.sh を Fig. 5.10 に示す．入出力データのファイル機番の指定は，COMMAND.specter ファイルを参考にした．また，機番 "66" に指定した WRITE 文で出力されるデータを，"TYPE.out" というファイル名にした．COMMAND.specter ファイルを Fig. 5.11 に示す．シェルスクリプトの内容の説明を Table 5.1 に示す．

5.2.2 SPECOMP の実行方法

SPECTER と同様，シェルスクリプトを作成し，ジョブの実行を行う．今回作成した，シェルスクリプト exe-specomp.sh を Fig. 5.12 に示す．入出力データのファイル機番の指定は，COMMAND.specomp ファイルを参考にした．COMMAND.specomp ファイルを Fig. 5.13 に示す．標準出力ファイル名を，SPECOMP-msg とした．SPECOMP では，標準出力を指定している処理がないため，SPECOMP-msg ファイルには，エラーメッセージが出力される．

5.3 実行結果の確認

AP3000 での実行で生成されたファイル HF32D.OUT，PKAP1.OUT と，SPECTER-ANL パッケージに含まれていた，SPECTER 及び SPECOMP の出力サンプルデータの比較を行った．その結果，実行結果に誤差が生じていた．その誤差は，0.1 % 以下であることから，本コードでは，問題ないことが確認されている．

5.4 まとめ

本作業では，SPECTER 及び SPECOMP の AP3000 へのインストールを行った．文法の違いが多少あったものの，特に問題も無く，スムーズにインストールを行うことができた．実行の確認として，テストデータでの確認しか行うことができず充分とは言えない状態である．今後，ユーザに使用して頂き，品質向上を図って行きたい．

Table 5.1 Shell-script option.

#@$-eo	標準エラー出力を標準出力ファイルへ出力することを指定.
#@$-q	投入するキュークラスの指定. 今回は, ss クラスを指定. キュークラスについては, 資料 1 を参照.
#@$-lt	CPU 時間の制限値を指定.
#@$-C	コメント.
cd $QSUB_WORKDIR	ジョブ投入時のカレントディレクトリに移動することを指定. （ジョブ開始時のカレントディレクトリは, ホームディレクトリになっている.）
setenv	入出力ファイルの指定（標準入出力以外）. 環境変数 fuXX （XX は装置参照番号）でファイル機番を指定.
specter	SPECTER の実行. "<"で標準入力, ">"で標準出力ファイルを指定. 例： specter < [標準入力ファイル] >& [標準出力ファイル]

```
 FILE   CONTENTS
 ----   --------
   1    This information file

**************** COMMAND PROCEDURES ****************
   2    Command procedure to run SPECTER
   3    Command procedure to run SPECOMP ( Type in '@SPECOMP VCT', refer
                                           Page 39 of the manual)

**************** SOURCE PROGRAM ********************
        VAX file name       CONTENTS
        -------------       --------
   4    SPECTER.FOR         SPECTER Fortran-77 source
   5    SPECOMP.FOR         SPECOMP Fortran-77 source


**************** INPUT DATA FILES *****************
                             File assignment
        VAX file name       SPECTER    SPECOMP    CONTENTS
        -------------       -------    -------    --------
   6    COMPOUND.DAT        FOR019                Input data
   7    GAS157.DAT          FOR018                Input data (gas production)
   8    HF32D.DAT           FOR005                Sample input data
   9    MACKLIB.DAT         FOR017                Input data
  10    NAMES.DAT                      FOR017     Input data
  11    NBGIN.DAT           FOR023                Input data
                                                  (neutron-gamma, beta deacy)
  12    PKAMIN.DAT          FOR010     FOR010     Input data
                                                  (Primary knock-on-atom data)
  13    SIGD.DAT            FOR015     FOR012     Input data
                                                  (displacements-per-atom data)
  14    SPECVCT.DAT                    FOR005     Sample input data


***************** SAMPLE OUTPUTS *****************
        CONTENTS
        --------
  15    SPECTER sample output
  16    SPECOMP sample output
```

Fig. 5.1 Information file.

```
% frt specter.f -o specter >& COMPOUT.specter
```

Fig. 5.2 Example of using 'frt' command.

```
jwd1110i-s  "specter.f", line 229: An expected name was not found.
```

Fig. 5.3 Compilation error message.

```
(修正前)
  229      31 TYPE 88,ILEM
(修正後)
           31 write(66,88)ILEM
```

Fig. 5.4 Modification of TYPE sentence.

```
jwd1040i-w  "specter.f", line 129: The old specification is used.
jwd1040i-w  "specter.f", line 249: The old specification is used.
```

Fig. 5.5 Compilation warning message.

```
(修正前)
  129        WRITE (6,78)(((WKER(J,I),I=1,5),CKER(J),CKRD(J)),J=1,52)
  249        WRITE(6,4) ((EF(IA),(DISPCS(IA,JA),JA=1,8)),IA=1,NEX)
(修正後)
  129        WRITE (6,78)((WKER(J,I),I=1,5),CKER(J),CKRD(J),J=1,52)
  249        WRITE(6,4) (EF(IA),(DISPCS(IA,JA),JA=1,8),IA=1,NEX)
```

Fig. 5.6 Modification of WRITE sentence.

```
jwd1040i-w  "specomp.f", line 17: The old specification is used.
jwd1040i-w  "specomp.f", line 23: The old specification is used.
```

Fig. 5.7 Compilation warning message.

(修正前)

```
17        READ(5,*)((Z(I),A(I),ED(I),TGAM(I),FR(I)),I=1,NELEM)

23        WRITE(8,402)((Z(I),A(I),ED(I),TGAM(I),FR(I)),I=1,NELEM)
```

(修正後)

```
17        READ(5,*)(Z(I),A(I),ED(I),TGAM(I),FR(I),I=1,NELEM)

23        WRITE(8,402)(Z(I),A(I),ED(I),TGAM(I),FR(I),I=1,NELEM)
```

Fig. 5.8 Modification of READ and WRITE sentence.

```
% qsub exe-specter.sh
```

Fig. 5.9 Example of using 'qsub' command.

```
#!/bin/csh -f
#@$-eo
#@$-q ss
#@$-lt 0:30:00
#@$-C SPECTER
cd $QSUB_WORKDIR

# input
setenv fu19 COMPOUND.DAT
setenv fu18 GAS157.DAT
setenv fu17 MACKLIB.DAT
setenv fu23 NBGIN.DAT
setenv fu10 PKAMIN.DAT
setenv fu15 SIGD.DAT

# output
setenv fu66 TYPE.out

timex specter < HF32D.DAT >& HF32D.OUT
```

Fig. 5.10 Shell-script for execution of SPECTER (exe-specter.sh).

```
$assign   hf32d.out     for006
$assign   compound.dat  for019
$assign   gas157.dat    for018
$assign   hf32d.dat     for005
$assign   macklib.dat   for017
$assign   nbgin.dat     for023
$assign   pkamin.dat    for010
$assign   sigd.dat      for015
$run      specter
```

Fig. 5.11 "COMMAND.specter" file.

```
#!/bin/csh -f
#@$-eo
#@$-q ss
#@$-lt 0:30:00
#@$-C SPECOMP
cd $QSUB_WORKDIR

# input
setenv fu17 NAMES.DAT
setenv fu10 PKAMIN.DAT
setenv fu12 SIGD.DAT

# output
setenv fu08 SPECP1.OUT
setenv fu15 COMPOUND.OUT
setenv fu13 PKAP1.OUT

timex specomp < SPECVCT.DAT >& SPECOMP-msg
```

Fig. 5.12 Shell-script for execution of SPECOMP (exe-specomp.sh).

```
$assign   spec'p1'.dat for005
$assign   spec'p1'.out for008
$assign   pkamin.dat   for010
$assign   sigd.dat     for012
$assign   pka'p1'.out  for013
$assign   compound.out for015
$assign   names.dat    for017
$run      specomp
```

Fig. 5.13 "COMMAND.specomp" file.

参 考 文 献

[1] L.R.Greenwood and R.K.Smither, 「SPECTER: Neutron Damage Calculations for Materials Irradiations」, ANL/FPP/TM-197 (Jun.1985).

[2] 「AP3000 システム 利用手引 第4版」, 日本原子力研究所 計算科学技術推進センター 情報システム管理室, 1999年 9月.

[3] 「FUJITSU Fortran90 使用手引書 V2用」, 富士通 (株), 1994年 10月.

6. MELCOR-FUS コードのインストールと高速化調査

6.1 はじめに

本作業では，核融合炉事故解析コードＭＥＬＣＯＲ－ＦＵＳ [1] のインストール及び高速化調査を行なった．ＭＥＬＣＯＲ－ＦＵＳコードは，核融合炉の事故解析に用いられているコードである．本コードは，軽水炉炉心損傷事故解析用に開発されたＭＥＬＣＯＲコードをベースに核融合炉事故解析に必要な機能を追加したコードである．ＭＥＬＣＯＲコードに以下のような修正が施され，ＭＥＬＣＯＲ－ＦＵＳコードが作成されている．

（１）水の物性値を３重点以下まで拡張
（２）極低温域でのＨｅ及び空気の状態式を追加
（３）エアロゾルモデルの改良
（４）輻射伝熱の改良

ベースとなるＭＥＬＣＯＲコードは，シビアアクシデント時の冷却系内及び格納容器内の熱水力，燃料棒の温度上昇／損壊，熔融炉心／コンクリート反応，可燃性ガスの生成／移行，放射性核種（エアロゾル，気体）の放出／移行／沈着，工学的安全施設による放射性核種の除去等の現象を解析することが出来る．

インストール及び高速化調査の対象計算機は，分散メモリ型ベクトル並列計算機ＶＰＰ５００ [2,3] である．

6.2 インストール作業

ＭＥＬＣＯＲ－ＦＵＳコードはＭＥＬＣＯＲコード本体に核融合サブルーチンをリンクし作成されるものである．よって，先ずＭＥＬＣＯＲコードをインストールする必要がある．ＭＥＬＣＯＲコードのインストールには，附属の専用インストール支援プログラムを使用しなければならない．よって，ＭＥＬＣＯＲコードをインストールする前に支援プログラムのインストールも必要となる．そのため，以下に示すような３段階の作業を行ないＭＥＬＣＯＲ－ＦＵＳコードをインストールする手順となる．

・ＭＥＬＣＯＲコードインストール用支援プログラムのインストール
・ＭＥＬＣＯＲコード本体のインストール
・核融合サブルーチンのリンク（ＭＥＬＣＯＲ－ＦＵＳコード）

6.2.1 支援プログラムのインストール

支援プログラムのインストールにはＭＥＬＣＯＲコード附属のインストーラを使用する．インストーラ名は，sinstall.unx である．sinstall.unx をコマンド入力することにより，以後は会話

形式でインストールが進められる．会話形式による入力完了後，コンパイルが開始される．コンパイル・リンクが完了すると，Table 6.1 に示す１７個の実行形式ファイルと１個のライブラリが作成される．これらのファイル群がＭＥＬＣＯＲコード本体のインストール時に必要となる．

6.2.2 ＭＥＬＣＯＲコード本体のインストール

ＭＥＬＣＯＲコード本体のインストールには附属のインストーラを使用する．インストーラ名は，muinstall.unx であり，コマンド入力後会話形式でインストールが進められる．会話形式による入力完了後，コンパイルが開始され，実行形式ファイルが２個作成される．作成ファイル名は，ＭＥＬＧＥＮとＭＥＬＣＯＲの２種類である．また，会話形式の入力項目により，デバックモードの実行形式ファイルを作成することも可能である．本作業では，最適化された実行形式ファイルのみを作成した．

6.2.3 ＭＥＬＣＯＲ－ＦＵＳコードのインストール

ＭＥＬＣＯＲ－ＦＵＳコードには附属のインストーラは無く，ＭＥＬＣＯＲコード附属のシェルスクリプトを使用しＭＥＬＣＯＲコード本体に必要ファイルをリンクする方式となっている．リンク用シェルスクリプト名は，mulinkmel.sun であり会話形式で処理が進められるように作られている．会話形式の入力項目により，コンパイル及びリンク処理を制御することが出来る．本作業では，ＭＥＬＧＥＮとＭＥＬＣＯＲに対し以下に示す３個のファイルを結合した．

```
mods.f    mods.dba.f    mods2.dba.f
```

また，実行形式ファイルの接頭語としてＦＵＳを指定しており，ＭＥＬＣＯＲ－ＦＵＳコードの実行形式ファイル名は，ＦＵＳＭＥＬＧＥＮとＦＵＳＭＥＬＣＯＲとして作成されている．

6.3 高速化調査

ＭＥＬＣＯＲ－ＦＵＳコードの高速化の可能性について検討した．現状では１回のシミュレーション当たり数十時間を要しており，高速化の必要がある．ＶＰＰ５００における高速化とは，ベクトル処理による演算時間短縮のことである．ベクトル化を施すことが出来ないアルゴリズムの場合は，ベクトル処理向きに処理を変更する等の修正が必要になる場合もある．調査方針として以下の２項目に従い，調査を行なった．

・コード全体のコスト分布を計測する．
・コスト上位ルーチンについてベクトル処理の可否を判断する．
（アルゴリズムの変更が必要か否かも含める）

6.3.1 コスト分布

ＶＰＰ５００上のＳＡＭＰＬＥＲ [4] を使用し，ＭＥＬＣＯＲ－ＦＵＳコードの動的解析を行なった．入力データとして，us2.3b.inp2r[tend 200] を使用した場合のスカラ実行時の測定結果

をFig. 6.1 に，ベクトル実行時の測定結果をFig. 6.2 に示す．この結果を見ると，特定のルーチンにコストが集中することなく，均一に分散していることが分かる．

6.3.2 高速化の可能性

コード全体として見た場合，コストが均一であることからベクトル化対象ルーチン数が多くなり，作業期間も長期におよぶことになる．例えば，各ルーチンがスカラ比５倍の性能を出したと仮定した場合，コード全体で２倍の性能を出すためには，全コストの７０％に相当するルーチンをチューニングしなければならない．これだけの作業を行なったとしても２倍程度の性能アップにすぎない．一部，ベクトル処理により性能が２０倍程度向上するルーチンも含まれているが，全体に占める割合が小さいため，コード全体の実行時間に与える影響は微小である．よって，長期間のチューニング作業を実施したとしても４～５倍の性能向上が上限であると予測する．

各ルーチン単位で見た場合，コスト上位のルーチンは十分にベクトル処理されておらず，ベクトル化を行なう必要がある．ベクトル化を行なうにあたってアルゴリズムの変更などは必要無く，特に困難な部分は存在しない．しかし，ベクトル長が短いループが多く，ベクトル処理をしても十分な性能は発揮されないと予測する．例として，以下にコスト上位５ルーチンについて解説する．

・ rn1rn4 ルーチン

プログラムとしては，ベクトル処理コーディングへの変更は可能である．多重ループについては，ループの入れ換えやループの１重化等によりベクトル長を拡長することも可能である．一部のループでは，ループ長が４６～２８５のものがあり，ベクトル化による効果が期待できる．しかし，ほとんどのループにおいて，ループ長が１～２３程度と短くなっている．特に，isum 及び isumb 変数を使用したループについては，ループ長が１～４でありベクトル処理には不向きである．このようなループをベクトル処理させた場合，スカラ実行と比較して性能が劣化することとなる．よって，ベクトル化は可能であるが，演算性能の向上はスカラ比２倍程度であると予測する．rn1rn4 ルーチン内の各ＤＯループの情報をTable 6.2 に示す．

・ rn2con ルーチン

プログラムとしては，ベクトル処理コーディングへの変更は可能である．多重ループについては，ループの入れ換えによりベクトル長を拡張することができる．しかし，ループ長が１あるいは１６のループが大半であり，これらのループはベクトル処理には不向きである．一部のループではループ長が４２あるいは４６であり，ベクトル処理の効果が微小ではあるが現れる．ベクトル長４６を確保出来たとしてもスカラ比３倍程度と予測する．よって，ベクトル化は可能であるが，演算性能の向上はスカラ比３倍以下であると予測する．rn2con ルーチン内の各ＤＯループの情報をTable 6.3 に示す．

・ hsrun2 ルーチン

プログラムとしては，ベクトル処理コーディングへの変更は可能である．しかし，ループ長が１～９のループが大半であり，これらのループはベクトル処理には不向きである．一部のループではループ長が４６あるいは７２であり，ベクトル処理の効果が微小ではあるが現れる．ベクトル長７２を確保出来たとしてもスカラ比４倍程度と予測する．ループ長の短いループが多いため，ベクトル化は可能であるが，演算性能の向上はスカラ比３倍以下であると予測する．

hsrun2 ルーチン内の各ＤＯループの情報を Table 6.4 に示す.

・ utlund ルーチン

ベクトル処理可能なコーディングとなっている．ループ長も 61549 と長くベクトル処理に向いている．実行文を多少変更することにより，さらに性能を向上させることができると判断する．演算性能の向上はスカラ比２０倍程度と予測する． utlund ルーチン内の各ＤＯループの情報を Table 6.5 に示す.

・ rn2mox ルーチン

プログラムとしては，ベクトル処理コーディングへの変更は可能である．しかし，ループ長が１～１６とベクトル処理には不向きなループが半数を占めている．残るループは４２以上であり，ベクトル処理の効果が微小であるが現れる．また，１つのループに関しては，ループ長が１７６でありベクトル処理の効果が期待できる．よって，ベクトル化可能であり，演算性能の向上はスカラ比８倍程度と予測する． rn2mox ルーチン内の各ＤＯループの情報を Table 6.6 に示す.

6.3.3 Ｉ／Ｏについて

ＭＥＬＣＯＲ－ＦＵＳコードを実行すると７個のファイルが作成される．このファイルはシミュレーション時間（サイクル数）と比例関係にあり，長時間の場合は作成ファイルも大きくなる．本作業において使用した入力データ（us2.3b.inp2r [tend 173800.0]）では，リスタートファイルと可視化用ファイルと結果出力ファイルの３個のファイル容量が特に大きかった．これら３個のファイル容量を合計すると，約７６ＭＢとなった．また，他の４個のファイルも合わせた総合計量は，約８０ＭＢであった．ＶＰＰ５００におけるディスク環境の場合，８０ＭＢの書き出しに要する時間は，およそ 20 分である．（Ｉ／Ｏ時間はマシンの使用状況により，大きく変動する傾向がある）Ｉ／Ｏ処理は，高速化チューニング不可能であり，高速ファイルシステムを導入するしかない.

6.3.4 他機種による性能測定

ＡＰ３０００にもＭＥＬＣＯＲ－ＦＵＳコードをインストールし，実行時間を測定した．ＶＰＰ５００とＡＰ３０００による実行時間を Table 6.7 に示す．この測定結果を見ると，ＶＰＰ５００ではＡＰ３０００と比較して４倍程度遅いことが分かる．ＶＰＰ５００は，ベルトル型スーパーコンピュータでありベクトル処理を実現することで高い性能を発揮することができる．ベクトル処理が行なわれず，スカラ処理の場合はワークステーションより性能が劣る傾向がある．また，例えベクトル処理を実現したとしてもベクトル長が短い場合は十分な性能は発揮されない．このことから，十分なベクトル長を確保できないコードの場合は，ベクトル型スーパーコンピュータではなくスカラ型のマシンを使用した方が良い.

6.4 まとめ

ＶＰＰ５００においては，スカラ比４倍程度の性能向上が望めると予測する．しかしながら，ＶＰＰ５００上にて４倍の性能向上を実現したとしても，ＡＰ３０００と同程度の性能となってしまう．よって，ベクトル処理向きコーディングへの変更は可能であるが，作業期間が長期にわ

たり，性能向上率も低いことから，ＭＥＬＣＯＲ－ＦＵＳコードはスカラ型スーパーコンピュータ上で使用することが望ましいと判断する．

Table 6.1 Subprogram for instration of MELCOR.

ファイル名	内容
beep	ビープ音を鳴らす
CMP	コードメンテナンスプログラム
cmpcopy	ファイルをコピーする
cmpdif	ファイルを分割する
hisplt	メタファイルを作成する
hispltm	メタファイルを作成する
mupack	ファイルの圧縮・解凍
poppst	メタファイルからプストスクリプトファイルへ変換する
popx11	X11 上に解析結果をグラフィック表示する
popx12	X11 上に解析結果をグラフィック表示する
poptk4	Tektronix 4014 フォーマットでのメタファイル作成
popt15	Tektronix 4115/4125 フォーマットでのメタファイル作成
popt05	Tektronix 4105 フォーマットでのメタファイル作成
popalp	Tektronix 4014 フォーマットでのメタファイルから プストスクリプトファイルへ変換する
polp	Polygon OverLay Processor
popsun	SUNVIEW 上に表示する
t2p	テキストをポストスクリプトへ変換する
librscors.a	RSCOR のライブラリ

Table 6.2 Do-loop list in rn1rn4 routine. (1/3)

DOループ	回転数	備考
DO 1 I=1,nnumhs	72	入力データを元にルーチン内で決定
DO 2 J=1,4	4	プログラム内に直接記述
DO 11 I=1,nsec	11	入力データで定義
DO 11 J=1,ncls	16	入力データで定義
DO 11 K=1,nvol	46	入力データを元にルーチン内で決定
DO 12 I=1,nvol	46	入力データを元にルーチン内で決定
DO 13 I=1,nsec	11	入力データで定義
DO 13 K=1,nvol	46	入力データを元にルーチン内で決定
DO 3401 I=1,ncof,5	57	入力データを元にルーチン内で決定
DO 3402 I=1,ncof,5	57	入力データを元にルーチン内で決定
DO 3403 I=1,ncof,5	57	入力データを元にルーチン内で決定
DO 3404 I=1,ncof,5	57	入力データを元にルーチン内で決定
DO 400 N=1,nvol	46	入力データを元にルーチン内で決定
DO 301 I=1,ncls	16	入力データ内で定義
DO 141 I=1,nsrca	4	入力データ内で定義
DO 142 I=1,nsec	11	入力データ内で定義
DO 143 I=1,nsec	11	入力データ内で定義
DO 390 I=4,nmat	5	入力データを元にルーチン内で決定
DO 395 I=4,nmat	5	入力データを元にルーチン内で決定
DO 398 I=1,nsec	11	入力データ内で定義
DO 401 I=1,ncof	285	入力データを元にルーチン内で決定

Table 6.2 Do-loop list in rn1rn4 routine. (2/3)

DOループ	回転数	備考
DO 402 J=1,ncmp	1	入力データ内で定義
DO 145 I=1,nsec	11	入力データ内で定義
DO 150 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 140 I=1,nsrca	4	入力データ内で定義
DO 134 J=1,nsec	11	入力データ内で定義
DO 135 J=1,nsec	11	入力データ内で定義
DO 155 I=1,nsec	11	入力データ内で定義
DO 160 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 165 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 450 I=1,nsec	11	入力データ内で定義
DO 450 J=1,ncmp	1	入力データ内で定義
DO 446 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 8 L=1,nsec	11	入力データ内で定義
DO 500 J=1,2*nsec+1	23	入力データ内で定義
DO 495 I=1,ncmp	1	入力データ内で定義
DO 600 J=1,2*nsec+1	23	入力データ内で定義
DO 595 I=1,ncmp	1	入力データ内で定義
DO 996 I=1,nsec	11	入力データ内で定義
DO 996 J=1,ncmp	1	入力データ内で定義

Table 6.2 Do-loop list in rn1rn4 routine. (3/3)

DOループ	回転数	備考
DO 997 J=1,ncmp	1	入力データ内で定義
DO 1452 J=1,ncmp	1	入力データ内で定義
DO 470 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 460 I=1,nsec	11	入力データ内で定義
DO 481 I=1,nsec	11	入力データ内で定義
DO 1581 J=1,nclsb	15	入力データを元にルーチン内で決定
DO 396 I=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 445 I=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 9 I=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 479 I=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 480 J=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 1481 K=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 1482 K=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 490 J=isum,isumb	1～4	入力データを元にルーチン内で決定
DO 1445 I=isum,isumb	1～4	入力データを元にルーチン内で決定

Table 6.3 Do-loop list in rn2con routine.

DOループ	回転数	備考
DO 100 NVOL=1,numvol	46	入力データを元にルーチン内で決定
DO 110 I=1,ncls	16	入力データ内で定義
DO 111 J=1,nsec1	1	nsec1 は rn2mox より渡される引数 rn2mox 内の実引数は nsexv であり プログラム内で 1 と定義している
DO 113 J=1,nflt DO 113 K=1,ncls DO 113 I=1,nsec3	1 16 1	nflt は rn2pbd 内で 1 と定義されている 入力データ内で定義 nsec3 は rn2mox より渡される引数 rn2mox 内の実引数は nsexz であり プログラム内で 1 と定義している
DO 200 NFL=1,numfl	42	入力データを元にルーチン内で決定
DO 202 I=1,nsec1	1	nsec1 は rn2mox より渡される引数 rn2mox 内の実引数は nsexv であり プログラム内で 1 と定義している
DO 202 J=1,ncls	16	入力データ内で定義
DO 300 NVOL=1,numvol	46	入力データを元にルーチン内で決定
DO 310 I=1,nsec1	1	nsec1 は rn2mox より渡される引数 rn2mox 内の実引数は nsexv であり プログラム内で 1 と定義している
DO 310 J=1,ncls	16	入力データ内で定義
DO 215 J=1,ncls	16	入力データ内で定義
DO 220 J=1,ncls	16	入力データ内で定義
DO 410 K=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 501 I=1,nsec3	1	nsec3 は rn2mox より渡される引数 rn2mox 内の実引数は nsexz であり プログラム内で 1 と定義している

Table 6.4 Do-loop list in hsrun2 routine.

DOループ	回転数	備考
DO 3004 N=1,numvol	46	入力データを元にルーチン内で決定
DO 10 J=1,numvol	46	入力データを元にルーチン内で決定
DO 12 K=1,nummat	9	入力データを元にルーチン内で決定
DO 11 K=1,3	3	プログラム内に直接記述
DO 116 J=1,numvol	46	入力データを元にルーチン内で決定
DO 115 J=1,numhs	72	入力データを元にルーチン内で決定
DO 5010 J=1,numhs	72	入力データを元にルーチン内で決定
DO 120 K=1,3	3	プログラム内に直接記述
DO 130 N=mesh1,mesh2	1～6	ルーチン内で計算し決定される
DO 489 K=1,3	3	プログラム内に直接記述
DO 490 N=node1,node2	7	入力データを元にルーチン内で決定
DO 501 N=1,nodes	7	入力データを元にルーチン内で決定
DO 3100 N=mesh1,mesh2	1～6	入力データを元にルーチン内で決定
DO 3200 N=2,nmesh	1～5	入力データを元にルーチン内で決定
DO 3350 NODE=nodes-1,1,-1	6	入力データを元にルーチン内で決定
DO 3300 N=1,nodes	7	入力データを元にルーチン内で決定
DO 3310 N=node1,node2	7	入力データを元にルーチン内で決定
DO 3400 N=1,nodes	7	入力データを元にルーチン内で決定
DO 3405 N=1,nodes	7	入力データを元にルーチン内で決定

Table 6.5 Do-loop list in utlund routine.

DOループ	回転数	備考
DO 100 I=n1,n2	61549	入力データを元にルーチン内で決定

Table 6.6 Do-loop list in rn2mox routine. (1/3)

DOループ	回転数	備考
DO 1 J=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 1 I=1,ncls	16	入力データ内で定義
DO 35 I=1,numfl	42	入力データを元にルーチン内で決定
DO 97 I=1,nsec	11	入力データ内で定義
DO 10 I=1,nconva	176	入力データ内で定義
DO 30 I=1,numfl	42	入力データを元にルーチン内で決定
DO 95 I=1,nnfl	42	入力データを元にルーチン内で決定
DO 95 J=1,nsec	11	入力データ内で定義
DO 5 I=1,numfl	42	入力データを元にルーチン内で決定
DO 96 I=1,nnfl	42	入力データを元にルーチン内で決定
DO 1200 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 1200 J=1,ncls	16	入力データ内で定義
DO 1201 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1201 I=1,ncls	16	入力データ内で定義
DO 200 I=1,ncls	16	入力データ内で定義
DO 200 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1202 K=1,nnvol	16	入力データを元にルーチン内で決定
DO 1202 I=1,ncls	16	入力データ内で定義
DO 210 I=1,ncls	16	入力データ内で定義
DO 210 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1220 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 1220 J=1,ncls	16	入力データ内で定義

Table 6.6 Do-loop list in rn2mox routine. (2/3)

DOループ	回転数	備考
DO 220 I=1,ncls	16	入力データ内で定義
DO 220 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 220 J=1,nsecx	11	入力データ内で定義
DO 222 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 222 J=1,ncls	16	入力データ内で定義
DO 230 I=1,ncls	16	入力データ内で定義
DO 230 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 230 J=1,nsecx	11	入力データ内で定義
DO 232 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 232 J=1,ncls	16	入力データ内で定義
DO 20 J=1,nconvv	16	入力データ内で定義
DO 1240 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 1240 J=1,ncls	16	入力データ内で定義
DO 1203 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1203 I=1,ncls	16	入力データ内で定義
DO 240 I=1,ncls	16	入力データ内で定義
DO 240 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1204 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 1204 J=1,ncls	16	入力データ内で定義
DO 250 I=1,ncls	16	入力データ内で定義
DO 250 K=1,nnvol	46	入力データを元にルーチン内で決定

Table 6.6 Do-loop list in rn2mox routine. (3/3)

DOループ	回転数	備考
DO 1260 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 1260 J=1,ncls	16	入力データ内で定義
DO 260 I=1,ncls	16	入力データ内で定義
DO 260 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 262 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 262 J=1,ncls	16	入力データ内で定義
DO 270 I=1,ncls	16	入力データ内で定義
DO 270 K=1,nnvol	46	入力データを元にルーチン内で決定
DO 272 I=1,nflt	1	nflt は rn2pbd 内で 1 と定義されている
DO 272 J=1,ncls	16	入力データ内で定義
DO 11 K=1,nsec	11	入力データ内で定義
DO 11 I=1,ncls	16	入力データ内で定義
DO 11 J=1,numfl	42	入力データを元にルーチン内で決定
DO 12 I=1,ncls	16	入力データ内で定義
DO 12 J=1,numfl	42	入力データを元にルーチン内で決定
DO 21 I=1,ncls	16	入力データ内で定義
DO 21 I=1,numfl	42	入力データを元にルーチン内で決定

Table 6.7 Execution time.

マシン	ＣＰＵ時間	ＥＬＡＰＳ時間
ＡＰ３０００	1分1秒	1分30秒
ＶＰＰ５００（ベクトル）	4分00秒	5分51秒
ＶＰＰ５００（スカラ）	4分52秒	9分47秒

使用データ： us2.3b.inp2r [tend　200]

```
real          9:47.83
user          4:52.52
sys           2:25.38
vu-user          0.50
vu-sys           0.00
vuw-user         0.00
vuw-sys          0.00

 Status                          : Serial
 Number of Processors            : 1

 Type                            : cpu
 Interval (msec)                 : 10

 Synthesis Information
     Count|    Percent|     VL| Name
      3950|       13.5|      -|  rn1rn4_
      3281|       11.2|      -|  rn2con_
      1580|        5.4|      -|  hsrun2_
      1433|        4.9|      -|  utlund_
      1207|        4.1|      -|  rn2mox_
      1068|        3.7|      -|  pfuns4_
       515|        1.8|      -|  rn1fun_
       509|        1.7|      -|  mpevan_
       503|        1.7|      -|  cvhmom_
       490|        1.7|      -|  rn1rn1_
       487|        1.7|      -|  h2o1ph_
       476|        1.6|      -|  daxpy_
       465|        1.6|      -|  rn1mds_
       412|        1.4|      -|  hstran_
       385|        1.3|      -|  tfbfnd_
       367|        1.3|      -|  rn2rn2_
       346|        1.2|      -|  hshuse_
       341|        1.2|      -|  cvtsve_
       336|        1.1|      -|  tfvdbg_
       333|        1.1|      -|  rn1rn6_
       321|        1.1|      -|  h2o2ph_
       316|        1.1|      -|  dabmix_
       306|        1.0|      -|  h2osau_
       303|        1.0|      -|  rn2rn0_
       301|        1.0|      -|  ncgpro_
       280|        1.0|      -|  mpintp_
       269|        0.9|      -|  mpprpm_
       263|        0.9|      -|  cvhrn3_
       251|        0.9|      -|  rn1rn2_
       242|        0.8|      -|  mpmix_
       197|        0.7|      -|  dgeco_
       193|        0.7|      -|  cvthrm_
       182|        0.6|      -|  mpprpl_
       169|        0.6|      -|  hscvpl_
       168|        0.6|      -|  cvtwge_
       168|        0.6|      -|  cvhccn_
```

Fig. 6.1 Dynamic behavior of scalar version. (1/5)

```
166|      0.6|     -|  h2oest_
163|      0.6|     -|  satnc_
153|      0.5|     -|  fthrm_
151|      0.5|     -|  mpdfvl_
147|      0.5|     -|  tfvdbf_
147|      0.5|     -|  hscvat_
146|      0.5|     -|  mpdif_
142|      0.5|     -|  fdiff_
140|      0.5|     -|  mprho1_
138|      0.5|     -|  dasum_
136|      0.5|     -|  rn1dbd_
134|      0.5|     -|  cvhmtx_
132|      0.5|     -|  rn1dif_
130|      0.4|     -|  cvhmdt_
125|      0.4|     -|  fgrav_
122|      0.4|     -|  idamax_
121|      0.4|     -|  cvhfrf_
117|      0.4|     -|  cfrun_
110|      0.4|     -|  hsbndy_
102|      0.3|     -|  rn1bta_
102|      0.3|     -|  dscal_
102|      0.3|     -|  h2oeqn_
100|      0.3|     -|  mpeval_
 98|      0.3|     -|  dgefa_
 95|      0.3|     -|  hsflpr_
 88|      0.3|     -|  ddot_
 87|      0.3|     -|  cvtsat_
 86|      0.3|     -|  oxidize_
 85|      0.3|     -|  etime_
 82|      0.3|     -|  mpdifa_
 80|      0.3|     -|  mpevam_
 79|      0.3|     -|  hsprop_
 78|      0.3|     -|  hsrun1_
 77|      0.3|     -|  rn1rks_
 77|      0.3|     -|  hsvvol_
 70|      0.2|     -|  hsqflx_
 68|      0.2|     -|  mprhoa_
 67|      0.2|     -|  cfdbv_
 64|      0.2|     -|  cvtneq_
 62|      0.2|     -|  rn1dsc_
 58|      0.2|     -|  hsfrac_
 58|      0.2|     -|  mpvis_
 55|      0.2|     -|  hsdmtc_
 53|      0.2|     -|  flcok_
 53|      0.2|     -|  cvhvel_
 51|      0.2|     -|  tfvalu_
 51|      0.2|     -|  cvhdon_
 51|      0.2|     -|  rn2dcf_
 49|      0.2|     -|  tfvdbe_
 48|      0.2|     -|  cvhcar_
 47|      0.2|     -|  inplnb_
 47|      0.2|     -|  cvhbl_
```

Fig. 6.1 Dynamic behavior of scalar version. (2/5)

43	0.1	-	cfrdbc_
40	0.1	-	rn2dbd_
39	0.1	-	condnc_
38	0.1	-	cvhpad_
37	0.1	-	hsrun3_
35	0.1	-	mxxcfa_
35	0.1	-	cvhtkt_
35	0.1	-	cvhmds_
34	0.1	-	inprcw_
34	0.1	-	hsecom_
34	0.1	-	rn2rn1_
34	0.1	-	hsboil2_
33	0.1	-	mpvis1_
32	0.1	-	cvhpov_
32	0.1	-	cvhhmc_
32	0.1	-	cvhbvt_
31	0.1	-	inpupc_
29	0.1	-	hsenst_
29	0.1	-	fun1_
28	0.1	-	mpcpsa_
26	0.1	-	mxx2pt_
25	0.1	-	hsedt_
25	0.1	-	h2o2pl_
25	0.1	-	hsmtrn_
25	0.1	-	hsmcom_
25	0.1	-	cvhvve_
24	0.1	-	cvhdtc_
23	0.1	-	inpnbk_
23	0.1	-	cvhmxm_
23	0.1	-	dchloc_
22	0.1	-	rn2bol2_
22	0.1	-	hsrun4_
22	0.1	-	hsradm_
22	0.1	-	cvhdl_
21	0.1	-	cfrdbb_
21	0.1	-	rn1rnh_
21	0.1	-	cvhbv1_
19	0.1	-	cvhalp_
19	0.1	-	dchdhq_
19	0.1	-	mpthc1_
18	0.1	-	hsedto_
18	0.1	-	cvhjun_
18	0.1	-	cvhrn4_
18	0.1	-	rn1cfc_
17	0.1	-	rn1ed5_
17	0.1	-	hsdbd_
17	0.1	-	mpthca_
16	0.1	-	cvhedt_
16	0.1	-	cvtcar_
15	0.1	-	dchdhr_
15	0.1	-	dchccs_
15	0.1	-	dchexp_

Fig. 6.1 Dynamic behavior of scalar version. (3/5)

```
 15|      0.1|      -|  mexrun_
 14|      0.0|      -|  encrad_
 14|      0.0|      -|  mexlop_
 13|      0.0|      -|  rn1ed1_
 13|      0.0|      -|  h2osat_
 12|      0.0|      -|  oxirate_
 12|      0.0|      -|  cvhspc_
 11|      0.0|      -|  hscvms_
 11|      0.0|      -|  rn1rn3_
 11|      0.0|      -|  ncgenr_
 11|      0.0|      -|  mxxcrt_
 11|      0.0|      -|  cvhdbd_
 10|      0.0|      -|  mxxrs_
 10|      0.0|      -|  cfedt_
 10|      0.0|      -|  cvhrn1_
 10|      0.0|      -|  cvhvmt_
 10|      0.0|      -|  h2o2pd_
 10|      0.0|      -|  cvhbv2_
 10|      0.0|      -|  cfrval_
  9|      0.0|      -|  cvhfog_
  9|      0.0|      -|  dgesl_
  9|      0.0|      -|  cvhmeb_
  8|      0.0|      -|  rn1rn5_
  8|      0.0|      -|  cvhbv3_
  8|      0.0|      -|  rn1cfb_
  8|      0.0|      -|  rhonc_
  8|      0.0|      -|  h2oept_
  7|      0.0|      -|  rn1ed3_
  7|      0.0|      -|  hscnds_
  7|      0.0|      -|  cvhzpl_
  6|      0.0|      -|  fledg_
  6|      0.0|      -|  flpum_
  6|      0.0|      -|  rn1dbn_
  6|      0.0|      -|  dchccn_
  6|      0.0|      -|  cvhbfx_
  5|      0.0|      -|  fledt_
  5|      0.0|      -|  tfedt_
  5|      0.0|      -|  rn1gs2_
  5|      0.0|      -|  rn1bcl_
  5|      0.0|      -|  rn1rkf_
  5|      0.0|      -|  dchdht_
  5|      0.0|      -|  cvhmt2_
  5|      0.0|      -|  mexdtc_
  4|      0.0|      -|  cvhzcp_
  4|      0.0|      -|  rn2boil_
  4|      0.0|      -|  mexstq_
  4|      0.0|      -|  hsdbs_
  4|      0.0|      -|  cvhrn2_
  4|      0.0|      -|  cvhdbn_
  4|      0.0|      -|  rn2dbc_
  3|      0.0|      -|  ncgrio_
  3|      0.0|      -|  mxxpzk_
```

Fig. 6.1 Dynamic behavior of scalar version. (4/5)

```
     3|       0.0|      -|  sleq1_
     3|       0.0|      -|  rn2mov_
     3|       0.0|      -|  dchdhu_
     3|       0.0|      -|  mexrn1_
     3|       0.0|      -|  flvot_
     3|       0.0|      -|  mxxcpy_
     3|       0.0|      -|  rn1hsr_
     3|       0.0|      -|  rn2mow_
     2|       0.0|      -|  mexoot_
     2|       0.0|      -|  rn1rio_
     2|       0.0|      -|  mpedt_
     2|       0.0|      -|  mxxplu_
     2|       0.0|      -|  mxxsrt_
     2|       0.0|      -|  rn1gbt_
     2|       0.0|      -|  rn1fhl_
     2|       0.0|      -|  cvhdbm_
     2|       0.0|      -|  dgefs_
     2|       0.0|      -|  mxxcpu_
     2|       0.0|      -|  mpdfsa_
     2|       0.0|      -|  rn1dbc_
     2|       0.0|      -|  mxxplw_
     2|       0.0|      -|  mexums_
     2|       0.0|      -|  rn1dbr_
     1|       0.0|      -|  mxxveb_
     1|       0.0|      -|  inpscc_
     1|       0.0|      -|  crakra_
     1|       0.0|      -|  crakin_
     1|       0.0|      -|  crackr_
     1|       0.0|      -|  cvhrio_
     1|       0.0|      -|  mexuin_
     1|       0.0|      -|  dchedt_
     1|       0.0|      -|  inpndb_
     1|       0.0|      -|  inploc_
     1|       0.0|      -|  mexedt_
     1|       0.0|      -|  mexsd0_
     1|       0.0|      -|  cvhdbc_
     1|       0.0|      -|  mexedj_
     1|       0.0|      -|  mex3dt_
     1|       0.0|      -|  hsdbm_
     1|       0.0|      -|  cfdbd_
     1|       0.0|      -|  hsdbn_
     1|       0.0|      -|  dchcls_
     1|       0.0|      -|  mextms_
     1|       0.0|      -|  mpmdf2_
     1|       0.0|      -|  rn2spr_
     1|       0.0|      -|  mexdtr_
     1|       0.0|      -|  dchmdb_
     1|       0.0|      -|  cfdbm_
     1|       0.0|      -|  rn1prs_

 29229|          |      -|  TOTAL
```

Fig. 6.1 Dynamic behavior of scalar version. (5/5)

```
real         5:51.58
user         4:00.26
sys            11.93
vu-user        45.62
vu-sys          0.00
vuw-user        0.00
vuw-sys         0.00

 Status                        : Serial
 Number of Processors          : 1

 Type                          : cpu
 Interval (msec)               : 10

 Synthesis Information
     Count|   Percent|     VL| Name
      2067|       8.6|    287|  rn1rn4_
      1733|       7.2|     11|  rn2con_
      1502|       6.3|      4|  hsrun2_
      1306|       5.4|      -|  pfuns4_
       673|       2.8|      5|  daxpy_
       595|       2.5|    517|  mpevan_
       567|       2.4|      4|  mpmix_
       562|       2.3|     89|  cvhmom_
       537|       2.2|     27|  ncgpro_
       505|       2.1|      -|  hshuse_
       489|       2.0|      -|  h2o1ph_
       468|       1.9|      2|  tfvdbg_
       457|       1.9|      -|  mpprpm_
       424|       1.8|      -|  hstran_
       383|       1.6|      1|  tfbfnd_
       364|       1.5|      5|  dabmix_
       339|       1.4|      6|  cvtsve_
       332|       1.4|      -|  h2o2ph_
       311|       1.3|    675|  rn1rn1_
       306|       1.3|     19|  rn1rn6_
       289|       1.2|      -|  mpprpl_
       272|       1.1|      3|  rn1fun_
       248|       1.0|      -|  mpintp_
       234|       1.0|     11|  rn2rn0_
       231|       1.0|      -|  tfvdbf_
       222|       0.9|    220|  rn2mox_
       209|       0.9|      6|  cvtwge_
       190|       0.8|    360|  h2oest_
       189|       0.8|    332|  rn1rn2_
       183|       0.8|    212|  cvhrn3_
       180|       0.7|      -|  hsbndy_
       178|       0.7|      -|  hscvpl_
       170|       0.7|    843|  cvthrm_
       170|       0.7|      -|  mprho1_
       164|       0.7|    374|  h2osau_
       159|       0.7|      6|  hscvat_
```

Fig. 6.2 Dynamic behavior of vector version. (1/5)

159	0.7	-	fthrm_
158	0.7	-	mpdif_
156	0.6	-	fgrav_
155	0.6	9	cvhmdt_
151	0.6	-	mpdfvl_
151	0.6	2	cfrun_
147	0.6	2034	utlund_
141	0.6	-	tfvdbe_
140	0.6	5	dscal_
138	0.6	17	dgeco_
132	0.5	-	satnc_
130	0.5	-	cvhfrf_
128	0.5	5	dasum_
127	0.5	39	cvhmtx_
125	0.5	23	hsrun1_
119	0.5	-	fdiff_
118	0.5	-	mpeval_
117	0.5	-	mpevam_
117	0.5	-	dgefa_
117	0.5	8	rn1dif_
115	0.5	11	rn2rn2_
114	0.5	-	tfvalu_
112	0.5	-	oxidize_
110	0.5	9	cvhccn_
108	0.4	-	h2oeqn_
105	0.4	416	rn2dcf_
104	0.4	-	mpdifa_
103	0.4	-	rn1bta_
99	0.4	-	etime_
90	0.4	3	ddot_
87	0.4	-	mprhoa_
86	0.4	2	hsvvol_
84	0.3	1	rn1dsc_
80	0.3	-	hsflpr_
80	0.3	-	hsqflx_
80	0.3	94	rn1rnh_
77	0.3	9	cvhvel_
74	0.3	374	cvtsat_
64	0.3	-	hsdmtc_
62	0.3	22	cvhhmc_
61	0.3	-	cfdbv_
59	0.2	-	flcok_
57	0.2	-	hsfrac_
52	0.2	12	idamax_
52	0.2	8	cvhmds_
52	0.2	1	hsprop_
51	0.2	-	mpvis_
49	0.2	-	condnc_
47	0.2	6	cvtneq_
44	0.2	4	hsmcom_
43	0.2	-	cfrdbc_
43	0.2	6	cvhpov_

Fig. 6.2 Dynamic behavior of vector version. (2/5)

43	0.2	118	cvhdl_
41	0.2	-	hsrun3_
39	0.2	-	inprcw_
39	0.2	10	cvhpad_
39	0.2	-	cvhalp_
37	0.2	8	cvhbl_
37	0.2	7	cvhdon_
36	0.1	16	hsenst_
35	0.1	-	inplnb_
34	0.1	-	cvhbvt_
33	0.1	-	inpnbk_
33	0.1	-	hsecom_
31	0.1	-	mxx2pt_
31	0.1	-	fun1_
30	0.1	586	cvhvve_
29	0.1	-	inpupc_
28	0.1	-	hsmtrn_
27	0.1	-	dgesl_
26	0.1	-	hsrun4_
26	0.1	-	cvhmxm_
25	0.1	-	hsedt_
25	0.1	31	cvhtkt_
24	0.1	26	rn2bol2_
24	0.1	683	cvhzcp_
24	0.1	-	mpcpsa_
23	0.1	-	mxxcfa_
23	0.1	517	cvhrn4_
23	0.1	-	cfrdbb_
22	0.1	-	rn1dbd_
21	0.1	-	rn1cfc_
20	0.1	-	hsdbd_
20	0.1	6	cvhdtc_
20	0.1	-	rn2dbd_
20	0.1	-	dchdhr_
19	0.1	-	hsedto_
19	0.1	9	cvhbv3_
18	0.1	-	cvhedt_
17	0.1	-	cfrval_
17	0.1	-	hsboil2_
15	0.1	-	hsradm_
15	0.1	-	dchexp_
15	0.1	-	rn1rn3_
14	0.1	-	mxxrs_
14	0.1	-	dchloc_
14	0.1	-	h2oept_
13	0.1	-	mpthc1_
13	0.1	-	mpthca_
12	0.0	26	cvhbv1_
12	0.0	1023	rn1mds_
12	0.0	37	cvhjun_
11	0.0	99	rn1ed5_
11	0.0	-	mexlop_

Fig. 6.2 Dynamic behavior of vector version. (3/5)

11	0.0	-	cvhfog_
11	0.0	1	cvhzpl_
11	0.0	6	cvhspc_
10	0.0	-	cfedt_
10	0.0	46	cvhvmt_
10	0.0	-	h2osat_
10	0.0	-	rn1cfb_
9	0.0	-	rn1gs2_
9	0.0	-	dchccs_
9	0.0	-	mpvis1_
9	0.0	-	encrad_
9	0.0	-	ncgenr_
8	0.0	11	rn1ed1_
8	0.0	-	dchdhq_
8	0.0	-	hscvms_
8	0.0	-	cvhrn1_
8	0.0	-	mexstq_
8	0.0	-	hscnds_
8	0.0	-	mexrun_
7	0.0	-	rn1ed3_
7	0.0	8	cvhbv2_
7	0.0	-	rn2rn1_
7	0.0	375	h2o2pl_
7	0.0	-	oxirate_
6	0.0	-	cvhbfx_
6	0.0	-	cvhdbd_
5	0.0	-	fledt_
5	0.0	-	fledg_
5	0.0	-	tfedt_
5	0.0	-	mxxplw_
5	0.0	-	rn1bcl_
5	0.0	-	dgefs_
5	0.0	-	mxxcpu_
5	0.0	2	sleq1_
5	0.0	-	rhonc_
4	0.0	-	dchcls_
4	0.0	-	rn2mow_
4	0.0	-	flvot_
4	0.0	-	cfdbd_
4	0.0	-	h2o2pd_
4	0.0	-	cvhrn2_
4	0.0	-	mexrn1_
3	0.0	-	mxxsrt_
3	0.0	72	rn1rn5_
3	0.0	-	mpdfsa_
3	0.0	-	mextms_
3	0.0	-	cvhmt2_
3	0.0	-	rn1vpr_
3	0.0	-	rn1dbn_
3	0.0	-	cvhmeb_
3	0.0	-	cvhcar_
3	0.0	-	mexedt_

Fig. 6.2 Dynamic behavior of vector version. (4/5)

```
         2|        0.0|      -|  inpndb_
         2|        0.0|      -|  mpedt_
         2|        0.0|      -|  dchdhu_
         2|        0.0|      -|  mexsd0_
         2|        0.0|      -|  rn1dbc_
         2|        0.0|      -|  dchccn_
         2|        0.0|    512|  rn1hsr_
         2|        0.0|      -|  rn2bol1_
         2|        0.0|      -|  cfdbn_
         2|        0.0|      -|  dchddt_
         2|        0.0|      -|  rn2mov_
         2|        0.0|      -|  rn1dbq_
         2|        0.0|      9|  cvtcar_
         2|        0.0|     72|  hsdbs_
         2|        0.0|      -|  cvhbvc_
         2|        0.0|      -|  cvhbvv_
         2|        0.0|      -|  mxxcrt_
         1|        0.0|      -|  mexoot_
         1|        0.0|      -|  crakra_
         1|        0.0|      -|  crackr_
         1|        0.0|      -|  mexedi_
         1|        0.0|      -|  hsrio3_
         1|        0.0|      -|  mxxplt_
         1|        0.0|      -|  mxxpzk_
         1|        0.0|      -|  mxxkdm_
         1|        0.0|      -|  rn1dpt_
         1|        0.0|      -|  mexdtc_
         1|        0.0|      -|  dchmdb_
         1|        0.0|      -|  rn2boil_
         1|        0.0|      -|  mpmdf2_
         1|        0.0|      -|  rn1cfa_
         1|        0.0|      -|  rn2dbo_
         1|        0.0|      -|  cfdbm_
         1|        0.0|      -|  hsdmfx_
         1|        0.0|      -|  dchdht_
         1|        0.0|      -|  rn1dbr_
         1|        0.0|      -|  mexums_
         1|        0.0|      -|  mesdmc_
         1|        0.0|      -|  hsdbc_
         1|        0.0|      -|  rn2spr_
         1|        0.0|      -|  mexsd4_
         1|        0.0|      -|  cfdbo_
         1|        0.0|      -|  cvhrio_
         1|        0.0|      -|  rn1rio_

     24012|           |    227|  TOTAL
```

Fig. 6.2 Dynamic behavior of vector version. (5/5)

参 考 文 献

[1] 「放射性物質移行挙動解析コードのワークステーションモデル化」，東芝アドバンストシステム株式会社，1997 年 5 月.

[2] UXP/M VPP FORTRAN77 EX/VP 使用手引書 V12 用，富士通 (株)，1994 年 9 月.

[3] UXP/M VPP FORTRAN77 EX/VPP 使用手引書 V12 用，富士通 (株)，1994 年 1 月.

[4] 「UXP/M VPP アナライザ使用手引書 V10 用」，富士通（株），1994 年 1 月.

7. COBRA-TF コードのインストール

7.1 はじめに

サブチャンネル解析コード COBRA-TF(Coolant Boiling in Rod Arrays, Two-Fluid) [1,2] の AP3000 へのインストールを行なった．これは，当初 VPP500 （以下 VPP）上で，ベクトル処理による高速実行を目指したものであったが，高速化調査をしたところ，ベクトル化効果，工数，平成 12 年度の計算機交換等を考慮すると，高速化チューニングせずに AP3000 上で実行すべきと結論づけたためである．この報告書では， VPP へのインストール，ベクトル化のための調査内容， AP3000 へのインストール， CPU 時間測定結果等について述べる．

7.2 コード概要

このコードは，３次元２流体（液体，蒸気）３領域（液膜，液滴，蒸気）モデルを用いて気液２相流の熱水力状況を計算するコードである．主として水冷却炉の燃料棒の冷却限界を解析するサブチャンネル解析に用いられている．モデルの特徴から，冷却水が少なくなって発熱体表面が薄い液膜で冷却されている状況を解析するのに適しており， BWR の解析等に有効である．数値解法としては，流体の質量，運動量及びエネルギー保存則を semi-implicit の有限差分式に表現した連立方程式を， Guss-Seidel 法で解いている．コードの規模は，約 23000 行，サブルーチン数は 82 個である．

このコードは，米国 PNL(Pacific Northwest Laboratory) で， CDC(Control Data Corporation) の計算機を使って開発された．この CDC 版コードを原研に導入し，富士通Mシリーズ計算機用にコンバートされた．その後， SUN の EWS にも導入されている [3]．さらに， VPP にも導入が試みられたが，ベクトル実行ができなかったという経緯を持つ．よって，今回は VPP にインストールし，高速化調査を行なった．

7.3 VPP500 へのインストール

COBRA-TF コードは，現在 SUN 上で動作しており，コンパイラも SUN コンパイラを使用している． VPP にインストールするためには， VPP コンパイラの仕様に変更することが必要である．ここでは変更した仕様内容について述べる．

7.3.1 仕様の変更

SUN コンパイラとの仕様で大きく異なる点は CPU 時間の取得等のサービスルーチンと配列宣言方法，初期値の有無である．これを VPP 仕様に変更した．

(1) 時間に関する変更

サブルーチンSECONDでは，SUNのサービスルーチンETIME [4]を使用してCPU時間を取得していた．これに替えて，コメント化されていた富士通（VPP）コンパイラ用のCPU時間取得サービスルーチンCLOCKM [5]を使用するように変更した．修正したルーチンをFig. 7.1に示す．

また，いくつかのサブルーチンから呼ばれているサービスルーチンTIMEの機能は，現在の時刻をCHARACTER*8でhh:mm:ssの形態で取得する．VPPコンパイラにも同名のサービスルーチンがあるが，仕様が異なるためこのままでは使えない．そこで，各ルーチンでのTIMEの呼び出しをサブルーチンJTIMEの呼び出しに変更し，同機能をもつサブルーチンJTIMEを作成した．このJTIMEをFig. 7.2に示す．

(2) 日付に関する変更

サブルーチンZDATEでは（実行時の）日付を取得するサービスルーチンとしてDATEを使用している．VPPにも同名のサービスルーチンがあるが，仕様が異なるため，このままでは使用できない．これもコメント化されているVPPコンパイラの仕様に合致するDATEの呼び出しを生かすことにより対応した．サブルーチンZDATEをFig. 7.3に示す．

(3) 共有された配列の宣言の変更

インクルードファイルMVYDT3ではFig. 7.4に示すように，COMMONブロックに属する配列とEQUIVALENCEによる領域共有をしている大きさ1の配列IGRFX，およびGDUMがある．実際のコードの内部では，これら配列の添字は1以上のものが使われている．VPPでは，このような配列宣言より大きな添字を使用することはできない（仮引数の配列宣言は除く）．このため，共有されいる配列（この場合それぞれGRFX，IGDUM）の大きさと同じにする変更を行なった．変更を行なったMVYDT3をFig. 7.5に示す．

(4) 未定義変数の変更

COBRA-TFコード内には，何も定義もされずに参照される変数・配列が存在した．SUNコンパイラでは，未定義の変数は0（ゼロ）が定義されるが，VPPの場合，未定義変数は不定の値が入っている場合があり，この参照により間違った計算結果を生ずる．このため，初期値の設定を行なった．

・サブルーチンSEDIT

サブルーチンSEDITでは，変数AVITが（タイムステップの値で，一番最初（ステップ数が0）の時）未定義であった．ステップ毎にタイムステップの値を書き出すため，最初のステップのみ未定義の値が書かれる．処置としてDATA文で初期値0.0を定義した．修正部分の図示は省略する．

・サブルーチンVELOC

サブルーチンVELOCでは，変数VMAXTVが未定義であった．これは速度の最大値を格納する変数と思われる．この変数は親ルーチンXSCHOMから引数で渡されている．よって，XSCHOMでDATA文による初期値0.0を定義した．修正部分の図示は省略する．

・サブルーチンINTFR

サブルーチン INTFR では，変数 AINTFC が未定義であった．最初に積和として求められているが，その式以降は再度，新たな定義がされている（Fig. 7.6参照）．積和の結果を使用しているところがないため，コード作成時の修正忘れか，これから変更を加える途中のものか不明である．AINTFC を DATA 文でゼロクリアしておけば問題ないので，この処置を行なった．

7.4 高速化調査

VPP 版ソースファイルを使用して高速化調査を行なった．詳細を以下に示す．

7.4.1 動的挙動調査

VPP 版ソースファイルを使用して動的挙動解析ツール sampler [6]，及び analyzer [7] を用いて動的解析を行なった．これにはテスト問題 as01 を使用した．sampler の結果を Fig. 7.7，analyzer の total list を Table 7.1に示す．これによると，サブルーチン TGAS，XSCHEM，INTFR 等のコストが高く，これらを中心に高速化する必要がある．なお，参考までに全体のツリー図を Fig. 7.8に示す．

7.4.2 高速化の可能性

COBRA-TF コードは，CDC 計算機を用い開発されているが，ベクトル化コーディングはまったく意識されていない．また，コストが大きいサブルーチンは，大規模なものが多く，ベクトル化のための変更作業は長期にわたるものが多い．上位３つのルーチンでコストが約 70 %であり，これらをベクトル化しただけではせいぜい２倍程度である，これ以上の倍率を出すためには，上位の 10 ルーチン程度（コストの約 90 %）はベクトル化する必要がある．

FORTRAN プログラムにおいてベクトル化の対象となるのは，DO ループである．そして効率のよいベクトル化効果を発揮するには，ループの回転数であるベクトル長を長くする必要がある．COBRA-TF の場合，ベクトル化の対象なる DO ループには，形状を意味するセクション数ループ，ノード数ループ，（流れの）軸方向ループがある．しかしながら，これらのループは，どのテスト問題でも十分な長さを持っていない．VPP では，ベクトル長が 2048 の時に最大のベクトル化効果が発揮できるようになっている．ベクトル長が短い場合，特に 10 以下の場合であると，スカラ計算より遅い場合が多々ある．10 以上の場合でも，ある程度ベクトル長が長くないと，ベクトル化効果があまり発揮されない．

これらをまとめると，COBRA-TF コードは工数の割に，ベクトル化効果は発揮されにくいコードと思われる．

7.4.3 各ルーチンについて

ここでは，コストが高いルーチンに対して，高速化調査を行なった結果について示す．

(1) サブルーチン TGAS

蒸気の温度を計算するルーチンであり，コストは全体の約 36 %である．このルーチンにはベ

クトル化対象となるDOループが存在しない．ルーチン自体も大きくなく，コストが高いのは，実行回数が多いためである．このルーチンはサブルーチンPOST3D，PROP，XSCHEM等で呼ばれている．これらルーチン内では，大規模（約1000ステップ）な3重ループの中で呼び出されている．外側から順にセクション数ループ，ノード数ループ，（流れの）軸方向ループで構成されており，それぞれループ長はテスト問題as01で1，25，2，テスト問題as02で1，31，1，テスト問題as03で，1，49，1である．これらのループをまとめて，TGASに引き込み，ベクトル処理することは可能である（一部ベクトル化非対象部分がある）が，親ルーチンを大きく変更する必要がある．しかも，ループをまとめてもas01でベクトル長50，as02で31，as03で49である．このため，ベクトル化効果はあまり期待できない．

(2) サブルーチンINTFR

相間，及び壁と各相間の輸送量を求めるルーチンであり，コストは約13％である．この親ルーチンはXSCHEMであり，セクション数ループ，ノード数ループの2重ループ内から呼ばれている．INTFR内では，軸方向ループ（これも大規模）から成り立っている．さらに，このループ内には，いくつかのDOループが存在しているが，いずれもループ長は短い．ゆえに，ベクトル化効果はあまり期待できない．

(3) サブルーチンXSCHEM

時間発展計算における，新しい時間の解を求めるルーチンで，実質的な過渡計算のメインルーチンである．このルーチンは，TGASの項目で述べたように，セクション数ループ，ノード数ループ，軸方向ループの3重ループの構造になっている．ベクトル化するには，これらのループをまとめること，ループ内に存在するベクトル化非対象部分（独立して計算できない部分等）をベクトル化対象ループから外すために，いくつかにループを分割する必要がある．しかしながら，ループが巨大なために工数が多くかかるのと，十分なベクトル長が望めない．

(4) サブルーチンSAT

水と蒸気の飽和エンタルピ等を求めるルーチンで，コストは約4％である．構造的にはTGASと同じで，このルーチン内にDOループは存在せず，親ルーチンのXSCHEMやPOST3Dから3重ループを引き込んでベクトル化することになる．

7.5 AP3000での実行

高速化調査の結果より，ベクトル化を保留し，AP3000で実行することにした．AP3000はスカラ計算機ながらCPUの性能は600Mflopsあり，ベクトル化されていない，もしくはベクトル化効果が少ないコードは，VPPより高速に実行することができるためである．

7.5.1 AP3000へのインストール

VPP用に変換したコードを，AP3000にインストールする際，富士通コンパイラを使用する限り，大きな非互換項目はない．COBRA-TFコードに関しても修正を行なったのはサービス

ルーチンひとつである．

(1) サービスルーチンの変更

VPP では，現時刻を取得するサービスルーチンとして，TIME を使用したが，AP3000 ではこれが無い．このため，同機能を持つサービスルーチン ITIME を使用することにした．機能は同じであるが使い方の仕様が異なるため，VPP 用に作成したサブルーチン JTIME を Fig. 7.9に示すように変更した．

7.5.2 ロードモジュール作成・実行

ここでは，AP3000 上における COBRA-TF コードのロードモジュール作成方法，および実行方法について述べる．

(1) ロードモジュールの作成

ロードモジュール作成には，makefile にて作成するようにした．また，高速に処理できるように最適化のコンパイルオプションをいくつか指定した．指定したコンパイルオプションを説明する．

-O3： 多重ループの構成変更，ループアンローリングの強化，ソフトウァアパイプライニング，最適化機能の繰り返し最適化を行なう

ULTRA,V8PLUS： UltraSPARC 向けの最適化を行なう

eval： 演算評価方法の最適化を行なう

preex： 不変式の先行評価を行なう

ただし，サブルーチン GRAF，SETUP，XSCHEM に関しては，不変式の先行評価の副作用によりエラーが発生するため，preex を外した．makefile の一部を Fig. 7.10に示す．

(2) 実行

AP3000 上で実行するには，実行シェルスクリプトを作成し，NQS によるバッチ処理を行なう．実行シェルスクリプトは提供されたものに，実行クラス・CPU 時間等の計算資源を指定しただけである．実行シェルスクリプトを Fig. 7.11に示す．

7.6 CPU 時間測定

VPP，AP3000 上で実際にコードを実行し，CPU 時間の測定を行なった．この結果を Table 7.2に示す．VPP での測定には，スカラ実行にし，最適化レベルは -Oe を指定した．なお，ベクトル実行の場合，オリジナル版コードは，ほぼベクトル化されていないので，同程度の時間である．AP3000 上では２つのモードで測定を行なった．ひとつはユーザ殿から提供していただいたソースとコンパイル・リンクシェルスクリプトによるもので，Sun コンパイラ (f77) を使用している．もう１つは VPP 用にコンバートしたソース，及び富士通 FORTRAN コンパイ

ラ (frt) を使用したものである．Sun コンパイラの方は，コンパイルオプションに最適化オプションが指定されておらず，富士通コンパイラには最適化オプション (-03 等) を使用しているため，高速な処理になっている．

7.7 まとめ

ベクトル化を行なうには，長期にわたってコードを変更する必要がある．また，現在の VPP では，解析対象にもよるが，効率のよいベクトル化効果はあまり望めない．ベクトル化しても，スカラ実行時の 4 倍程度と予測する．一方，AP3000 上での実行では，VPP500 の実行と比較して，3.5 倍程度高速に実行することができる．

また，平成 12 年度の冬には，東海研究所の VPP，AP3000 のリプレースが予定されており，より高速な計算機が導入されることになっている．導入される計算機については，まだ決定していないが，高速なスカラ計算機での実行が適していると思われる．また，工数の問題を除き，短いベクトル長でも高速に処理することが可能なベクトル計算機であれば，その計算機の仕様に合ったチューニングを施すことで，高速処理が期待できる．よって，現時点では，AP3000 上で実行し，新計算機が導入された時点で，利用形態を決定したほうがよい．

原子力コードは，採用されている解法アルゴリズムや構造によって，効率の出やすい計算機・出にくい計算機が存在する．今回扱った COBRA-TF コードは，ベクトル計算機には向いていなかった．しかしながら，現在各メーカーで開発されている計算機には，短いベクトル長でも効果があるものや，スカラ計算機ながら性能が 1 GFLOPS を越えるものも開発されている．予定されている計算機のリプレースによって，これらの特色を持つ計算機が使用できることになれば，COBRA-TF コードもチューニングを行なうことで高速な処理が可能となる．また，COBRA-TF 以外でも，これまでの計算機では高速計算ができなかったコードも，高速計算の対象になることが，十分に考えられる．

Table 7.1 Dynamic behavior of original COBRA-TF code(ANALYZER).

name	count	v-cost	%	s-cost	%	v-leng	v-rate	v-effect
TGAS	4724859	.9403E10	30.9	.9403E10	28.4	2	0.0	1.0- 1.0
XSCHEM	20451	.5002E10	16.4	.5700E10	17.2	6	21.0	1.0- 1.2
INTFR	531726	.4524E10	14.9	.4511E10	13.6	1	0.5	1.0- 1.0
VOLLIQ	2126957	.2308E10	7.6	.2329E10	7.0	3	0.9	1.0- 1.0
XTRA1	1022550	.2100E10	6.9	.2113E10	6.4	3	1.0	1.0- 1.0
POST3D	20451	.1209E10	4.0	.1207E10	3.6	2	0.8	1.0- 1.0
REDUCE	1022550	.1188E10	3.9	.2353E10	7.1	5	92.9	1.2- 2.6
SAT	3191398	.1098E10	3.6	.1098E10	3.3	1	0.0	1.0- 1.0
BACOUT	1022550	.7544E09	2.5	.1374E10	4.2	5	83.8	1.2- 2.3
VELOC	531726	.6570E09	2.2	.4818E09	1.5	1	27.8	0.6- 0.9
PROP	1575404	.4590E09	1.5	.4590E09	1.4	1	0.0	1.0- 1.0
VDRIFT	531726	.4190E09	1.4	.4190E09	1.3	1	0.0	1.0- 1.0
GSSOLV	20451	.3948E09	1.3	.4592E09	1.4	3	64.8	1.0- 1.4
FILLRO	1022550	.3028E09	1.0	.6376E09	1.9	5	88.0	1.2- 2.8
PREP3D	20451	.2280E09	0.7	.1839E09	0.6	2	77.2	0.6- 1.1
TRANSP	1575404	.1796E09	0.6	.1796E09	0.5	1	0.0	1.0- 1.0
GASP	2126905	.1042E09	0.3	.1042E09	0.3	1	0.0	1.0- 1.0

Table 7.2 Comparison of CPU time.

テスト問題	as01	as02	as03	as04
AP3000(f77 コンパイラ)	7m55s49	5m35s53	9m01s00	4m43s21
AP3000(frt コンパイラ)	4m16s85	3m32s17	5m26s66	2m41s28
VPP500(スカラ)	14m24s73	11m43s42	18m25s50	8m49s32

```
C     ====================================================================
C
C      S E C O N D
C
C     == COBRATF(ADD) ================================================
      SUBROUTINE SECOND( SEC )
      IMPLICIT REAL*8(A-H,O-Z)
cieaj
cvpp  REAL*4 ELAP(2)
cieaj
C
C----
cieaj CALL CLOCKM( MSEC )
cieaj SEC = MSEC/1000.0D0
cieaj
cvpp  CALL ETIME(ELAP)              ← コメント化した部分
      CALL CLOCKM( MSEC )           ← コメントを外した部分
      SEC = MSEC/1000.0D0           ←      同上
cieaj
cieaj   Sun FORTRAN LIBRARY ROUTINES : ETIME
cieaj
cieaj     ELAP(1)          = USER TIME
cieaj     ELAP(2)          = SYSTEM TIME
cieaj     ELAP(1)+ELAP(2)  = ELAPSED TIME
cieaj
cvpp  SEC=ELAP(1)+ELAP(2)           ← コメント化した部分
ieaj
      RETURN
      END
```

Fig. 7.1 Replacement of CPU time measurement routine.

```
cvpp
c     TIME GET routine
cvpp
      SUBROUTINE JTIME(ATIME)
      CHARACTER*8 ATIME
      CALL TIME(ITM)
      ITM = ITM/1000
      IH  = ITM/3600
      IM  = MOD(ITM,3600)/60
      IS  = MOD(ITM,60)
      WRITE(ATIME(1:2),'(I2)') IH
      WRITE(ATIME(4:5),'(I2)') IM
      WRITE(ATIME(7:8),'(I2)') IS
      IF(ATIME(1:1).EQ.' ') ATIME(1:1)='0'
      IF(ATIME(4:4).EQ.' ') ATIME(4:4)='0'
      IF(ATIME(7:7).EQ.' ') ATIME(7:7)='0'
      ATIME(3:3) = ':'
      ATIME(6:6) = ':'
C
       RETURN
       END
```

Fig. 7.2 Time measurement routine JTIME.

```
     SUBROUTINE ZDATE( CDAT )
cieaj
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*8 CDAT
cvpp  CHARACTER*9 DATSUN
cvpp  CHARACTER*2 CYEAR
cvpp  CHARACTER*2 CMONTH
cvpp  CHARACTER*2 CDAY
cieaj
cvpp  CALL DATE(DATSUN)
      CALL DATE(CDAT)
cvpp  CDAY =DATSUN(1:2)
cvpp  CYEAR=DATSUN(8:9)
cvpp  CALL IDATE(IMM,IDD,IYY)
cvpp  IF (IMM.EQ. 1) CMONTH='01'
cvpp  IF (IMM.EQ. 2) CMONTH='02'
cvpp  IF (IMM.EQ. 3) CMONTH='03'
cvpp  IF (IMM.EQ. 4) CMONTH='04'
cvpp  IF (IMM.EQ. 5) CMONTH='05'
cvpp  IF (IMM.EQ. 6) CMONTH='06'
cvpp  IF (IMM.EQ. 7) CMONTH='07'
cvpp  IF (IMM.EQ. 8) CMONTH='08'
cvpp  IF (IMM.EQ. 9) CMONTH='09'
cvpp  IF (IMM.EQ.10) CMONTH='10'
cvpp  IF (IMM.EQ.11) CMONTH='11'
cvpp  IF (IMM.EQ.12) CMONTH='12'
cvpp  CDAT=CYEAR//'-'//CMONTH//'-'//CDAY
cieaj
      RETURN
      END
```

Fig. 7.3 Date acquisition routine ZDATE.

```
     REAL*4 GRFX,TGDMP,GRFN,GDUM
cieaj
      CHARACTER*4 CGRFX  (20000)
      CHARACTER*4 CIGDUM (35)
      CHARACTER*4 CINDCMP(MIDIM)
      CHARACTER*4 CTGDMP (MIDIM)
cieaj
CAKI  COMMON /MVYDT3/ GRFX  (12000),TGDMP (MIDIM),GRFN  (M5DIM),
      COMMON /MVYDT3/ GRFX  (20000),TGDMP (MIDIM),GRFN  (M5DIM),
     +                INDCMP(MIDIM),IGRFIT(40   ),IGDUM (35   ),
     +                IGRF  (M5DIM,2 )
      DIMENSION       IGRFX(1),GDUM(1)      ← 配列の大きさ 1
      EQUIVALENCE    (IGRFX(1),GRFX(1)),(IGDUM(1),GDUM(1))  ← 共有
      cieaj
      EQUIVALENCE    (GRFX(1)   ,CGRFX(1)  )
      EQUIVALENCE    (IGDUM(1)  ,CIGDUM(1) )
      EQUIVALENCE    (INDCMP(1) ,CINDCMP(1))
      EQUIVALENCE    (TGDMP(1)  ,CTGDMP(1) )
cieaj
```

Fig. 7.4 Declaration of array.

```
      REAL*4 GRFX,TGDMP,GRFN,GDUM
cieaj
      CHARACTER*4 CGRFX  (20000)
      CHARACTER*4 CIGDUM (35)
      CHARACTER*4 CINDCMP(MIDIM)
      CHARACTER*4 CTGDMP (MIDIM)
cieaj
CAKI  COMMON /MVYDT3/ GRFX  (12000),TGDMP (MIDIM),GRFN  (M5DIM),
      COMMON /MVYDT3/ GRFX  (20000),TGDMP (MIDIM),GRFN  (M5DIM),
     +                INDCMP(MIDIM),IGRFIT(40   ),IGDUM (35   ),
     +                IGRF  (M5DIM,2 )
cvpp  DIMENSION       IGRFX(1),GDUM(1)
      DIMENSION       IGRFX(20000),GDUM(35)
      EQUIVALENCE    (IGRFX(1),GRFX(1)),(IGDUM(1),GDUM(1))
cieaj
      EQUIVALENCE    (GRFX(1)   ,CGRFX(1)  )
      EQUIVALENCE    (IGDUM(1)  ,CIGDUM(1) )
      EQUIVALENCE    (INDCMP(1) ,CINDCMP(1))
      EQUIVALENCE    (TGDMP(1)  ,CTGDMP(1) )
cieaj
```

Fig. 7.5 Modification of array declaration.

```
      SUBROUTINE INTFR( IDUMY, J, JP1, JM1, ICHOSE, MSK )
                     .
                     .
      DO 1500 IC = 1, ICHN
          I          = NCSEC(IC,ISECT)
                     .
                     .
         AINTFC    = AINTFC*DMIN1(ALL,AL(I,JP1))/ALP      ← 未定義参照
                     .
                     .
C---- INTERFACIAL HEAT TRANSFER COEFFICIENT AND DRAG
        AINTFC = 3.0D0*ALP*APPP*DX/RBUB                   ← 再定義
        XKIC   = .125D0*CDB*RLP*URVL*AINTFC*DXI
                     .
                     .
01500 CONTINUE
```

Fig. 7.6 Variable AINTFC in subroutine INTFR.

```
Status                          : Serial
 Number of Processors           : 1

 Type                           : cpu
 Interval (msec)                : 1

 Synthesis Information
     Count|   Percent|    VL| Name
     32312|      36.3|     -|  tgas_
     19377|      21.8|     1|  intfr_
     11171|      12.6|     6|  xschem_
      3399|       3.8|     -|  sat_
      3398|       3.8|     2|  post3d_
      2652|       3.0|     5|  volliq_
      2442|       2.7|     -|  prop_
      2164|       2.4|     1|  veloc_
      2020|       2.3|    10|  xtra1_
      1707|       1.9|     5|  reduce_
      1417|       1.6|     4|  bacout_
      1276|       1.4|    74|  gssolv_
      1236|       1.4|     -|  transp_
      1179|       1.3|     -|  gasp_
       895|       1.0|     -|  vdrift_
       778|       0.9|     1|  prep3d_
       473|       0.5|     6|  fillro_
       220|       0.2|     -|  second_
       146|       0.2|     -|  curve_
       117|       0.1|     -|  volvap_
       116|       0.1|     -|  dvdpv_
       109|       0.1|     -|  dvdhl_
       108|       0.1|     -|  trans_
        66|       0.1|     -|  dvdhv_
        44|       0.0|     -|  result_
        41|       0.0|     -|  graf_
        22|       0.0|     -|  newdlt_
        20|       0.0|     -|  afromh_
        13|       0.0|     -|  curve1_
        13|       0.0|     -|  splitt_
        11|       0.0|     -|  timchk_
        10|       0.0|     -|  input_
         5|       0.0|     -|  timstp_
         2|       0.0|     -|  outer_
         1|       0.0|     -|  setin_
         1|       0.0|     -|  setout_
         1|       0.0|     -|  zdate_
         1|       0.0|     -|  sswtch_

     88963|          |     9|  TOTAL
```

Fig. 7.7 Dynamic behavior of original COBRA-TF code(SAMPLER).

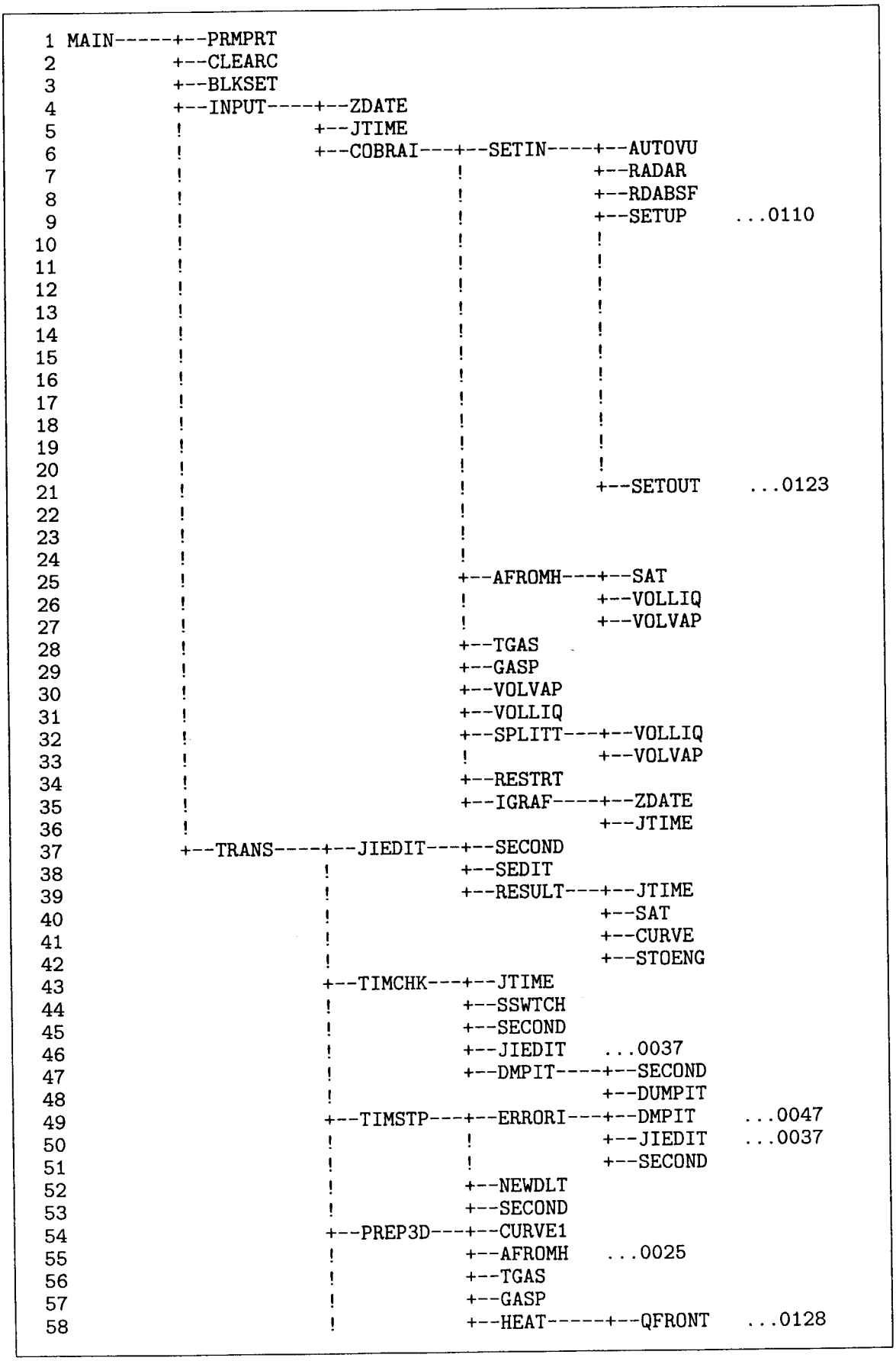

Fig. 7.8 Tree structure of COBRA-TF code (1/3).

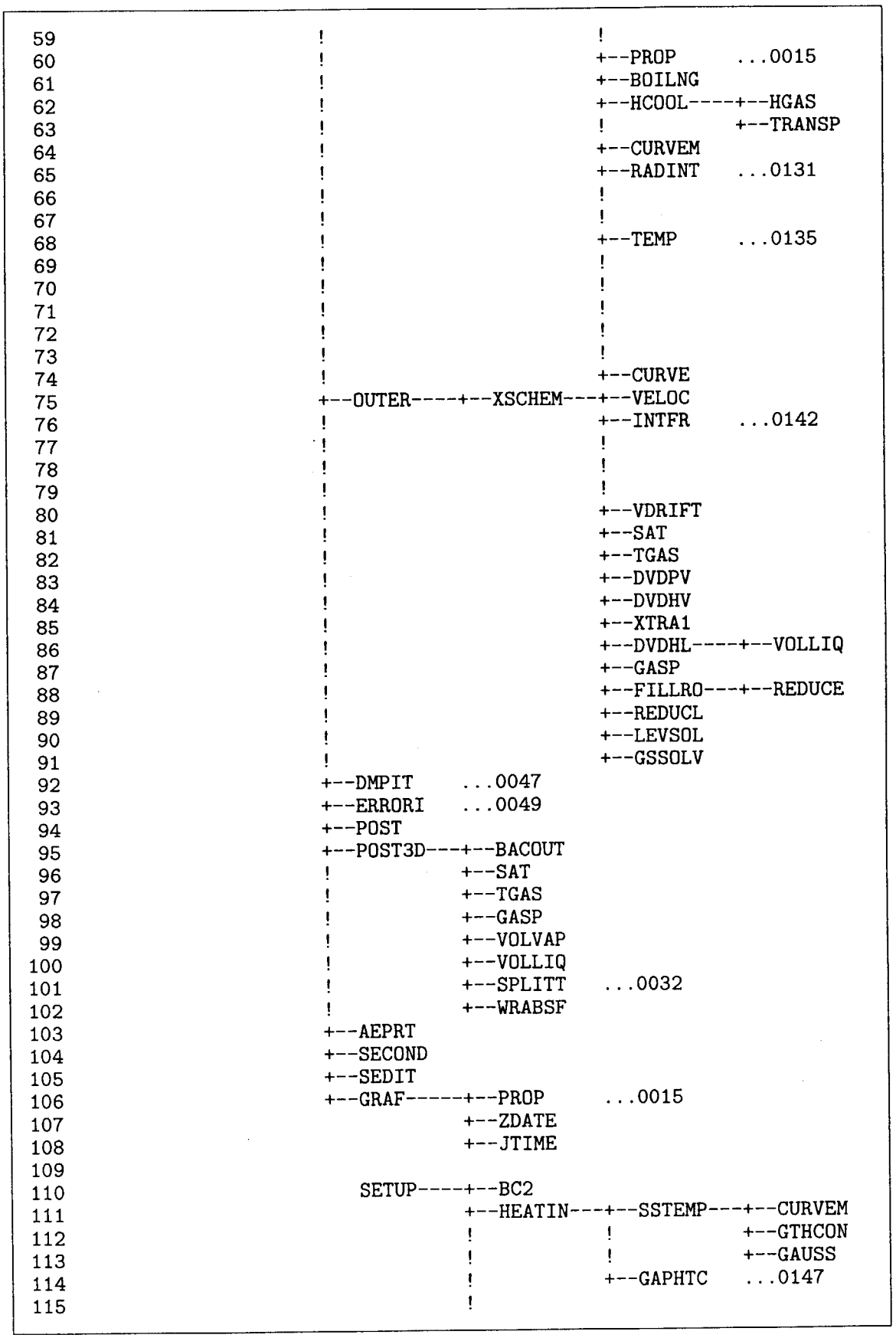

Fig. 7.8 Tree structure of COBRA-TF code (2/3).

```
116                                    +--PROP-----+--SAT
117                                    !           +--CURVE
118                                    !           +--TGAS
119                                    !           +--TRANSP
120                                    !           +--HGAS
121                                    +--CURVE
122
123                            SETOUT---+--ZDATE
124                                     +--JTIME
125                                     +--SAT
126                                     +--PROP     ...0015
127
128                            QFRONT---+--MOVE-----+--CURVEM
129                                                 +--CURVE
130
131                            RADINT---+--RADAR1
132                                     +--RADQ-----+--ABDROP
133                                                 +--LIN2
134
135                            TEMP-----+--CURVE
136                                     +--GTHCON
137                                     +--CURVEM
138                                     +--GAPHTC   ...0013
139                                     +--QOXIDE
140                                     +--GAUSS
141
142                           INTFR----+--PROP     ...0015
143                                    +--GRID-----+--CURVEM
144                                    +--TGAS
145                                    +--TRANSP
146
147                           GAPHTC---+--DEFORM---+--CURVE
148                                    +--GTHCON
```

Fig. 7.8 Tree structure of COBRA-TF code (3/3).

```
      SUBROUTINE JTIME(ATIME)
      CHARACTER*8 ATIME
      DIMENSION IW(3)
      CALL ITIME(IW)
      WRITE(ATIME(1:2),'(I2)') IW(1)
      WRITE(ATIME(4:5),'(I2)') IW(2)
      WRITE(ATIME(7:8),'(I2)') IW(3)
      IF(ATIME(1:1).EQ.' ') ATIME(1:1)='0'
      IF(ATIME(4:4).EQ.' ') ATIME(4:4)='0'
      IF(ATIME(7:7).EQ.' ') ATIME(7:7)='0'
      ATIME(3:3) = ':'
      ATIME(6:6) = ':'
C
       RETURN
       END
```

Fig. 7.9 Time measurement routine JTIME(AP3000 version).

```
F77     =       frt
FOPT    =       -O3 -KULTRA,V8PLUS,eval,preex
INCDIR  =       ../inc
FFLAGS  =       -I$(INCDIR)
LD      =       frt
LDFLAGS =
LIBS    =
TARGET  =       cobra

FSRCS   =       \
abdrop.f  aeprt.f   afromh.f  autovu.f  bacout.f  bc2.f     blkset.f \
blockd.f  boilng.f  clearc.f  cobrai.f  curve.f   curve1.f  curvem.f \
deform.f  dmpit.f   dumpit.f  dvdhl.f   dvdhv.f   dvdpv.f   errori.f \
fillro.f  gaphtc.f  gasp.f    gauss.f   graf.f    grid.f    gssolv.f \
gthcon.f  hcool.f   heat.f    heatin.f  hgas.f    igraf.f   input.f  \
intfr.f   jiedit.f  levsol.f  lin2.f    main.f    move.f    newdlt.f \
outer.f   post.f    post3d.f  prep3d.f  prmprt.f  prop.f    qfront.f \
qoxide.f  radar.f   radint.f  radq.f    rdabsf.f  reduce.f  reducl.f \
restrt.f  result.f  sat.f     second.f  sedit.f   setin.f   setout.f \
setup.f   splitt.f  sstemp.f  sswtch.f  stoeng.f  temp.f    tgas.f   \
timchk.f  timstp.f  trans.f   transp.f  vdrift.f  veloc.f   volliq.f \
volvap.f  wrabsf.f  xschem.f  xtra1.f   zdate.f

FOBJS   =       $(FSRCS:.f=.o)

.f.o            :
        $(F77) $(FOPT) $(FFLAGS) -c $<


$(TARGET)    :  $(LOCAL_OBJS) $(FOBJS)
        $(LD) $(LDFLAGS) -o $@ $(LOCAL_OBJS) $(FOBJS) $(LIBS)

graf.o  :       graf.f
        $(F77) -O3 -KULTRA,V8PLUS,eval $(FFLAGS) -c graf.f
setup.o :       setup.f
        $(F77) -O3 -KULTRA,V8PLUS,eval $(FFLAGS) -c setup.f
xschem.o        :       xschem.f
        $(F77) -O3 -KULTRA,V8PLUS,eval $(FFLAGS) -c xschem.f

#abdrop.o :
aeprt.o :  \
        $(INCDIR)/PJAERI  $(INCDIR)/TWOPHS   $(INCDIR)/UNITSI
```

Fig. 7.10 Modification of makefile

```
#! /bin/csh -f
#@$-eo
#@$-r COBRA
#@$-q ss
#@$-C  COBRA
cd $QSUB_WORKDIR
# rm core ft??
#
#  assign input data to ft01
#

ln -s ../input/as04.inp ft01
#ll
#
#  execute cobratf
#
timex ../vsrc/cobra >& err01.lst
#
# rename output files
#
mv ft05 as04.in2
mv ft06 as04.lst
mv ft07 as04.msg
mv ft08 as04.dmp
mv ft09 as04.dm2
mv ft11 as04.plt
mv ft59 as04.tty
#
#  delete work files
\rm core ft??
#
```

Fig. 7.11 Shell script for execution

参 考 文 献

[1] Alexandre EZZIDI, Tsutomu OKUBO, and Yoshio MURAO "IMPROVEMENT OF COBRA-TF CODE MODELS FOR LIQUID ENTRAINMENTS IN FILM-MIST FLOW", JAERI-M 93-133, July 1993.

[2] 最適評価コードの整備 II (36) (-COBRA-TF コードの相関式の調査 -) 作業報告書, 株式会社 アイ・イー・エー・ジャパン, 平成 8 年 12 月.

[3] 最適評価コードの整備 II (31) (-COBRA-TF コードのワークステーションへの導入 -), 株式会社 アイ・イー・エー・ジャパン, 1996 年 3 月.

[4] FORTRAN ライブラリ・リファレンスマニュアル, Sun Microsystems Computer Corporation, 1997 年 4 月.

[5] UXP/M VPP VPP FORTRAN77 EX/VP 使用手引書 V12 用, 富士通株式会社, 平成 6 年 9 月.

[6] UXP/M VPP アナライザ使用手引書 V10 用, 富士通株式会社, 1994 年 1 月.

[7] UXP/M アナライザ使用手引書 (FORTRAN,VP 用) V10L20 用, 富士通株式会社, 1992 年 2 月.

8. おわりに

計算科学技術推進センター情報システム管理課で実施している原子力コードの高速化作業は，平成11年度に18件の作業を完了した．平成12年度前期にも13件の作業が計画されている．これらの作業は，計算時間の大きい原子力コードを原研が保有する各種スーパーコンピュータ向けに最適なベクトル化，並列化を施す高速化チューニングを行うものであり，コード実行時間の大幅な短縮に寄与している．さらに，単一プロセッサ上ではメモリ不足から実行できないようなジョブを並列化により実行可能にするなど，計算機のスループット向上，ターンアラウンドタイム短縮，それによるユーザの仕事の効率化，計算可能なジョブの範囲の拡大など，計算機の効率的な運用と計算機資源の有効利用に大いに貢献するものと考えている．

本報告書では，生体分子の分子動力学パッケージAMBER5，（連続・多群）汎用中性子・光子輸送計算モンテカルロコードMVP/GMVP，MCNPライブラリ自動編集システムautonj，SPECTOR/SPECOMPコード，核融合炉事故解析コードMELCOR-FUS及びサブチャンネル解析コードCOBRA-TFのVPP500及びAP3000への整備について記述した．

原研では，平成12年度末に東海研，那珂研及び関西研の各地区のスーパーコンピュータの更新を予定しており，今回移植対象計算機であったVPP500及びAP3000も新機種へと移行する．これに伴い，新たな移植作業の発生が予想されるが，これまでの作業で培われてきたノウハウがさらに生かされて行くものと思われる．

本報告書が原子力コードの高速化に携わる人々に多少なりとも参考になれば幸いである．

謝　辞

本作業を行う上で，作業を依頼された数値実験技術開発グループ ミロスラフピナック氏（2章），炉特性研究室 長家康展氏（3章），MCNP高温ライブラリ作成WG（4章），材料試験炉部計画課 島川聡司氏（5章），ITER開発安全評価グループ 荒木隆夫氏（6章）及び将来型炉研究室 大久保務氏（7章）には，コード内容の把握に際し御協力頂きました．また，本報告書の作成に当り数値実験グループの渡辺正氏には御指導と御助言をいただきました．さらに，本作業を円滑に遂行するための各種事務処理については山田圭子氏に御協力を頂きました．ここにこれらの方々に感謝の意を表します．最後に，本報告書を執筆する機会を与えて下さいました計算科学技術推進センター長秋元正幸氏，情報システム管理課長藤井実氏，(株)富士通R&Dシステム部長平沢健一氏に感謝致します．

付録 A 実行シェルスクリプト (autonj.sh : オリジナル版)

```
#! /bin/csh
#
# ##########################################################################
#  autonj.sh  :  shell script for producing automatically the continuous
#                energy cross section library for the mcnp code from
#                jendl by the njoy97 code.
# ##########################################################################
#    Japan Atomic Energy Research Institute (JAERI)
#    coordinator: Dr. F. Maekawa (FNS) and Dr. K. Sakurai (TCA)
# ##########################################################################
#    produced by K. Kosako (SAEI) on Jan. 29, 1999.
# ##########################################################################
#
# ---- initialize ----
set blank=' '
set name=(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
#
echo ' '
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo '  autonj ..... automatic production system of a mcnp cross section'
echo '               library from JENDL by the NJOY97 code.'
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo ' '
stp00:
echo ' Please select your computer environment.'
echo '  HP  = HP9000-series, HP-C-series'
echo '  SUN = Sun SparcStation4/5'
echo ' ==========> ' | tr -d '\012'
set mach=$<
if ( $mach == 'hp' ) set mach=HP
if ( $mach == 'sun' ) set mach=SUN
if ( $mach == 'HP' || $mach == 'SUN' ) goto stp01
echo '..... Unknown computer name is detected.  Please reenter.'
goto stp00
stp01:
echo ' '
echo ' Please select an input mode.'
echo '  0 = input from a file.'
echo '  1 = interactive input on this terminal.'
echo ' ==========> ' | tr -d '\012'
set mode=$<
stp03:
echo ' '
echo ' Please select the option of using disk space (or producing files).'
echo '  0 = minimum disk size (mcnp library)'
echo '  1 = medium  disk size (mcnp library and output list)'
echo '  2 = maximum disk size (library, list and ace file)'
echo ' (maximum size is about 1.4 GB for JENDL-3.2.)'
echo ' (If blank, default size is minimum.)'
echo ' ==========> ' | tr -d '\012'
set disksp=$<
if ( $disksp == $blank ) set disksp=0
if ( $disksp == '0' ) goto stp04
if ( $disksp == '1' ) goto stp04
if ( $disksp == '2' ) goto stp04
echo '..... Unknown size option is detected.  Please reenter.'
goto stp03
stp04:
####
#### start of input data for autonj ####
```

```
####
if ( $mode == '0' ) then
   echo ' '
   echo ' Please enter a file name stored the input data.'
   echo ' (If blank, default name "inp-autonj" is used.)'
   echo ' ==========> ' | tr -d '\012'
   set infile=$<
   if ( $infile == $blank ) then
      set infile=inp-autonj
   else
      cp $infile inp-autonj
   endif
   goto stp10
endif
rm -f inp-autonj
set tt=JENDL-eval
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
if ( $mode == '1' ) then
   stp0:
   echo ' '
   echo ' Please enter the path name where JENDL file are located.'
   echo ' (When you finish entering all file names, please enter a blank.)'
   echo ' (If a directory name is given, all files in the directory are'
   echo '  processed. Only one directory name is allowed. The last'
   echo '  character must not be a slash (/).)'
   set ncount=0
   stp1:
      echo ' ==========> ' | tr -d '\012'
      set name1=$<
      if ( $name1 == $blank ) then
         if ( $ncount < 1 ) then
            echo '..... JENDL-3.2 file is not entered.  Please reenter.'
            goto stp0
         endif
         goto stp3
      endif
      if ( -d $name1 ) goto stp2
      if ( -e $name1 ) then
         set name2=$name1:e
         if ( $name2 == 'Z' | | $name2 == 'gz' ) then
            cp $name1 JENDL-eval
            set name3=$name1:t
            if ( $name2 == 'Z' ) then
               uncompress -f JENDL-eval/$name3
            else if ( $name2 == 'gz' ) then
               gunzip JENDL-eval/$name3
            endif
            set name1=JENDL-eval/$name3:r
         endif
         echo $name1 >>! inp-autonj
         @ ncount = $ncount + 1
         set name[$ncount]=$name1
      else
         echo '             **********************************************'
         echo '             Above file name is not found.  Please reenter.'
      endif
      goto stp1
   stp2:
```

```
   if ( $mach == 'HP' ) then
      uncompressdir -f $name1
   else if ( $mach == 'SUN' ) then
      uncompress -f ${name1}/*.Z  >&! autonj_dummy
   endif
   echo 'directory path:' $name1 >>! inp-autonj
   echo '          list of files in the directory above'
   foreach name2 ($name1/*)
      set name3=$name2:e
      if ( $name3 == 'Z' || $name3 == 'gz' ) then
         cp $name2 JENDL-eval
         set name4=$name2:t
         if ( $name3 == 'Z' ) then
            uncompress -f JENDL-eval/$name4
         else if ( $name3 == 'gz' ) then
            gunzip JENDL-eval/$name4
         endif
         set name5=JENDL-eval/$name4:r
      else
         set name5=$name2
      endif
      echo $name5 >>! inp-autonj
      @ ncount = $ncount + 1
      set name[$ncount]=$name5
      echo '             ' $name5
   end
stp3:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the MAT numbers of nuclides that you want to process.'
echo ' (If blank, all nuclides contained in the specified files are processed.)'
stp4:
   echo ' ==========> ' | tr -d '\012'
   set mat1=$<
   if ( $mat1 == $blank ) goto stp5
   echo $mat1 >>! inp-autonj
   goto stp4
stp5:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the processing temperature with NJOY97.'
echo ' (If blank, default temperature is 293 K.)'
echo ' (for example:  300       <--- 300 kelvin)'
echo ' (               27C      <--- 27 celsius = 300.15 K)'
echo ' ==========> ' | tr -d '\012'
set tempe=$<
if ( $tempe == $blank ) set tempe=293.
echo $tempe >>! inp-autonj
echo ' '
echo ' Please enter a library identification number in zaid for MCNP.'
echo ' (It corresponds to the "suff" parameter in input data of NJOY/ACER.)'
echo ' (It must be integer with two degits.)'
echo ' (If blank, default identification number is 38. (If Fe, 26000.38c))'
echo ' ==========> ' | tr -d '\012'
set suff=$<
if ( $suff == $blank ) set suff=38
echo $suff >>! inp-autonj
echo ' '
echo ' Please enter a tolerance of pointwise cross section data.'
echo ' (The unit of tolerance is [%].)'
echo ' (If blank, default tolerance is 0.2 %.)'
echo ' ==========> ' | tr -d '\012'
set toler=$<
if ( $toler == $blank ) set toler=0.2
echo $toler >>! inp-autonj
stp6:
```

```
   echo ' '
   echo ' Please enter the maximum number of discrete gamma-ray energies.'
   echo ' (It must be less than 1000 for MCNP-4B or 200 for the previous version.)'
   echo ' (If blank, default number is 1000.)'
   echo ' ==========> ' | tr -d '\012'
   set dsgam=$<
   if ( $dsgam == $blank ) set dsgam=1000
   if ( $dsgam > 1000 ) then
      echo '..... The number is greater than 1000.  Please reenter.'
      goto stp6
   endif
   echo $dsgam >>! inp-autonj
   echo ' '
   echo ' Please enter the general name of nuclear data file.'
   echo ' (It will be a part of the "hk" in input data of NJOY/ACER.)'
   echo ' (Maximum 16 characters.)'
   echo ' (If blank, the name will be prepaired from the nuclear data file.)'
   echo ' (for example: jendl-3.2 )'
   stp7:
   echo '            (....+....1....+.)'
   echo ' ==========> ' | tr -d '\012'
   set titl=$<
   if ( $titl == $blank ) then
      echo ' ' >>! inp-autonj
   else
      if ( $mach == 'HP' ) then
         set n= `expr length $titl`
      else if ( $mach == 'SUN' ) then
         set n= `expr $titl : '.*'`
      endif
      if ( $n > 16 ) then
         echo '..... The number of characters in name exceeded 16.  Please reenter.'
         goto stp7
      else
         echo $titl >>! inp-autonj
      endif
   endif
else
   echo ' *** fatal error *** illegal mode was detedted; mode=' $mode
   exit
endif
stp10:
echo ' '
echo ' Please enter types of MCNP cross section library.'
echo ' (If blank, default type of library is Type-2.)'
echo '  1 = Type-1 (text format)'
echo '  2 = Type-2 (binary format)'
echo ' 12 = Type-1 and Type-2'
echo ' ==========> ' | tr -d '\012'
set libf=$<
if ( $libf == $blank ) then
   set libf=2
   goto stp11
else if ( $libf == '1' || $libf == '2' || $libf == '12' ) then
   goto stp11
endif
goto stp10
stp11:
echo ' '
echo ' Please enter a file name of the MCNP cross section library.'
echo ' (Maximum 7 characters.)'
echo ' (When the given name is "fsxlib2", the complete name is "fsxlib21",'
echo '  if it is Type-1.)'
echo ' (If blank, default library name is "fsxlb32".)'
stp12:
echo ' ==========> ' | tr -d '\012'
```

```
set libnm=$<
if ( $libnm == $blank ) set libnm=fsxlb32
if ( $mach == 'HP' ) then
   set n=`expr length $libnm`
else if ( $mach == 'SUN' ) then
   set n=`expr $libnm : '.*'`
endif
if ( $n > 7 ) then
   echo '..... The number of character of library name exceeded 7.  Please reenter.'
   goto stp12
endif
echo ' '
if ( $libf == '1' ) then
   echo ' Please enter a file name of the directory file for the library.'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
else if ( $libf == '2' ) then
   echo ' Please enter a file name of the directory file for the library.'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 2
else if ( $libf == '12' ) then
   echo ' Please enter a file name of the directory files for the library.'
   echo ' --- type-1 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
   echo ' --- type-2 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm2=$<
   if ( $dirnm2 == $blank ) set dirnm2=xsdir.$ {libnm} 2
endif
####
#### end of input data for autonj ####
####
echo ' '
echo ' ::::                                                          ::::'
echo ' :::: Start the processing stage of nuclides by NJOY97 ::::'
echo ' ::::                                                          ::::'
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
#
if ( -e autonj-stop ) rm -f autonj-stop
##############
####  xnjoy : fortran program for processing the nuclear data file
####          (njoy97.45/autonj version)
##############
set xnjoy=njoy/xnjoy
if ( -e $xnjoy && -x $xnjoy ) then
else
   cd njoy
   if ( $mach == 'HP' ) then
      cp Make-hp Makefile
   else if ( $mach == 'SUN' ) then
      cp Make-sun Makefile
   endif
   make  >&! autonj_dummy
   rm -f Makefile
   cd ..
endif
```

```
################
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
################
if ( $mach == 'HP' ) then
   f77 +U77 -o matlist_ld Utility/matlist.f >&! autonj_dummy
else if ( $mach == 'SUN' ) then
   f77 -o matlist_ld Utility/matlist.f >&! autonj_dummy
endif
rm -f matlist.o
#---------
matlist__ld
#---------
if ( -e autonj-stop ) goto err1
#
################
####  nj-inp  : fortran program for making the input data of njoy97
################
if ( $mach == 'HP' ) then
   f77 +U77 -o nj-inp_ld Utility/nj-inp.f >&! autonj_dummy
else if ( $mach == 'SUN' ) then
   f77 -o nj-inp_ld Utility/nj-inp.f >&! autonj_dummy
endif
rm -f nj-inp.o
if ( $disksp == '2' ) goto stp13
################
####  mk1xsd  : fortran program for making the new single mcnp library
####            with type-1 from single dir and ace file
################
if ( $mach == 'HP' ) then
   f77 +U77 -o mk1xsd_ld Utility/mk1xsd.f >&! autonj_dummy
else if ( $mach == 'SUN' ) then
   f77 -o mk1xsd_ld Utility/mk1xsd.f >&! autonj_dummy
endif
rm -f mk1xsd.o
stp13:
set tt=ACE-FILE
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
set tt=OUT-FILE
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
rm -f autonj-count
set count=0
echo $count > autonj-count
rm -f autonj-xsdir acef1
#
####
#### loop of processing of all nuclides requested
#### execute the njoy97 code
####
stp20:
```

```
@ count++
if ( -e autonj-count ) then
#  ---------
   nj-inp_ld
#  ---------
   if ( -e autonj-stop ) goto err1
else
   goto stp30
endif
echo ' ' $count '..... processing nuclide ..... ' ` cat autonj-now `
#-------------
$xnjoy < input  >&! autonj_dummy
#-------------
if ( -e tape91 ) then
   if ( -z tape91 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape91 ACE-FILE/ace.$count
   else
      mv tape91 ACE-FILE/ace.1
   endif
else
   goto stp21
endif
if ( -e tape92 ) then
   if ( -z tape92 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape92 ACE-FILE/dir.$count
   else
      mv tape92 ACE-FILE/dir.1
   endif
else
   goto stp21
endif
if ( $disksp == '0' || $disksp == '1' ) then
#  ---------
   mk1xsd_ld  >&! autonj_dummy
#  ---------
   if ( -e autonj-stop ) goto err1
   rm -f ACE-FILE/*
endif
if ( $disksp == '0' ) then
   rm output
else
   mv output OUT-FILE/out.$count
   compress OUT-FILE/out.$count
endif
rm -f tape* autonj_dummy
goto stp20
stp21:
echo ' *** fatal error *** ace file or directory file was not found or zero-size.'
echo '                     njoy process was not completed.'
if ( -e tape91 || -e tape92 ) ls -l tape9?
exit
stp30:
@ count = $count - 1
echo ' total number of processed nuclides ..... ' $count
if ( $disksp == '0' || $disksp == '1' ) goto stp31
###############
####  mkxsdir : fortran program for making the new single mcnp library
####            with type-1
###############
if ( $mach == 'HP' ) then
   f77 +U77 -o mkxsdir_ld Utility/mkxsdir.f >&! autonj_dummy
else if ( $mach == 'SUN' ) then
   f77 -o mkxsdir_ld Utility/mkxsdir.f >&! autonj_dummy
endif
rm -f mkxsdir.o
```

```
#---------
mkxsdir_ld  >&! autonj_dummy
#---------
if ( -e autonj-stop ) goto err1
stp31:
if ( $libf == '1' ) goto stp32
###############
####  makxsf  : fortran program for converting to the requested mcnp
####            library format (makxsf took from mcnp-4a)
###############
f77 -o makxsf_ld Utility/makxsf.f >&! autonj_dummy
rm -f makxsf.o
if ( $libf == '2' ) then
   set dirnm=$dirnm1
else if ( $libf == '12' ) then
   set dirnm=$dirnm2
endif
rm -f tprint specs $ {libnm} 2 $dirnm
# convert from type-1 (text) to type-2 (binary) format
echo 'autonj-xsdir ' $dirnm              >> specs
echo 'acef1 ' $ {libnm} 2  2  2048  512   >> specs
echo ' '                                 >> specs
#--------
makxsf_ld  >&! autonj_dummy
#--------
stp32:
if ( $libf == '1' | | $libf == '12' ) then
   sed "s/acef1/$ {libnm} 1/" < autonj-xsdir > $dirnm1
   mv acef1 $ {libnm} 1
endif
echo ' '
echo ' file names of produced MCNP library:'
echo ' ------------------------------------'
if ( $libf == '1' | | $libf == '12' ) then
   set ns= ` ls -l $ {libnm} 1 `
   echo ' type-1 (text) form'
   echo '      cross section library .....' $ {libnm} 1 '    ' $ns[5] '[bytes]'
   echo '      directory .................' $dirnm1
endif
if ( $libf == '2' | | $libf == '12' ) then
   set ns= ` ls -l $ {libnm} 2 `
   echo ' type-2 (binary) form'
   echo '      cross section library .....' $ {libnm} 2 '    ' $ns[5] '[bytes]'
   echo '      directory .................' $dirnm
endif
echo ' '
if ( $disksp == '1' | | $disksp == '2' ) then
   echo ' Output-list file of njoy97 for each nuclide was stored into OUT-FILE/out.*.Z.
endif
if ( $disksp == '2' ) then
   echo ' ACE file of njoy97 for each nuclide was stored into ACE-FILE/ace.*.'
endif
####
#### ending process (including remove files)
####
if ( -e autonj-xsdir ) rm -f autonj-xsdir
if ( -e acef1 ) rm -f acef1
rm -f autonj-matlist autonj-now autonj_dummy
rm -f *_ld input tprint specs
rm -fR JENDL-eval
if ( $disksp == '0' | | $disksp == '1' ) then
   rm -fR ACE-FILE
endif
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
```

```
echo ' The autonj process is successfully terminated.'
echo ' Congratulations and good-bye.'
echo ' ..........................'
exit
#
####
#### error process
####
err1:
echo ' '
echo ' * * * fatal error * * * fatal error * * * fatal error * * *'
echo ' The error condition below was detected in an execution program.'
cat autonj-stop
echo ' autonj was stop... .. .'
exit
```

付録 B AP3000用コンパイル・リンク シェルスクリプト

```
#! /bin/csh
#
# ##########################################################################
#  autonj.sh  :  shell script for producing automatically the continuous
#                energy cross section library for the mcnp code from
#                jendl by the njoy97 code.
# ##########################################################################
#    Japan Atomic Energy Research Institute (JAERI)
#    coordinator: Dr. F. Maekawa (FNS) and Dr. K. Sakurai (TCA)
# ##########################################################################
#    produced by K. Kosako (SAEI) on Jan. 29, 1999.
# ##########################################################################
#    UPDATE:
# 15 Apr 1999 : (N.Kawasaki)
#        Change autonj.sh to anjcomp.sh(for compiling and linking only).
# ##########################################################################
#
##############
####  xnjoy : fortran program for processing the nuclear data file
####          (njoy97.45/autonj version)
##############

if ( -e xnjoy && -x xnjoy ) then
else
   cd Njoy
   cp Make-sun Makefile
   make  >&! autonj_dummy
   rm -f Makefile
   rm -f *.o
   mv xnjoy ..
   cd ..
endif

################
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
################

if ( -e matlist_ld ) then
else
  f77 -o matlist_ld Utility/matlist.f >&! autonj_dummy
  rm -f matlist.o
endif

################
####  nj-inp  : fortran program for making the input data of njoy97
################

if ( -e nj-inp_ld ) then
else
  f77 -o nj-inp_ld Utility/nj-inp.f >&! autonj_dummy
  rm -f nj-inp.o
endif

################
####  mk1xsd  : fortran program for making the new single mcnp library
####            with type-1 from single dir and ace file
################

if ( -e mk1xsd_ld ) then
else
  f77 -o mk1xsd_ld Utility/mk1xsd.f >&! autonj_dummy
```

```
  rm -f mk1xsd.o
endif

###############
####  mkxsdir : fortran program for making the new single mcnp library
####            with type-1
###############

if ( -e mkxsdir_ld ) then
else
  f77 -o mkxsdir_ld Utility/mkxsdir.f >&! autonj_dummy
  rm -f mkxsdir.o
endif

###############
####  makxsf  : fortran program for converting to the requested mcnp
####            library format (makxsf took from mcnp-4a)
###############

if ( -e makxsf_ld ) then
else
  f77 -o makxsf_ld Utility/makxsf.f >&! autonj_dummy
  rm -f makxsf.o
endif

####
#### ending process (including remove files)
####
mv Njoy/autonj_dummy ./nout.njoy
mv autonj_dummy ./nout.etc
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
echo ' The compiling and linking of autonj codes is terminated.'
echo ' Congratulations and good-bye.'
echo ' ..........................'
exit
#
```

付録 C 実行シェルスクリプト (autonjC.sh : 会話処理版)

```
#! /bin/csh
#
# ##########################################################################
#  autonj.sh  :  shell script for producing automatically the continuous
#                energy cross section library for the mcnp code from
#                jendl by the njoy97 code.
# ##########################################################################
#    Japan Atomic Energy Research Institute (JAERI)
#    coordinator: Dr. F. Maekawa (FNS) and Dr. K. Sakurai (TCA)
# ##########################################################################
#    produced by K. Kosako (SAEI) on Jan. 29, 1999.
# ##########################################################################
#    UPDATE:
# 15 Apr 1999 : (N.Kawasaki)
#        Convert original version into AP3000 version.
# ##########################################################################
#
# ---- Existence check for load modules of autonj system ----
# ##########################################################################
#    xnjoy, matlist_ld, nj-inp_ld, mk1xsd_ld, mkxsdir_ld, makxsf_ld
# ##########################################################################

set xnjoy=/dg06/center/codelib/autonj/autonjLM/xnjoy
set matlist_ld=/dg06/center/codelib/autonj/autonjLM/matlist_ld
set nj_inp_ld=/dg06/center/codelib/autonj/autonjLM/nj-inp_ld
set mk1xsd_ld=/dg06/center/codelib/autonj/autonjLM/mk1xsd_ld
set mkxsdir_ld=/dg06/center/codelib/autonj/autonjLM/mkxsdir_ld
set makxsf_ld=/dg06/center/codelib/autonj/autonjLM/makxsf_ld
set awr_list=/dg06/center/codelib/autonj/src/Utility/awr-list

if ( -e $xnjoy && -x $xnjoy ) then
else
echo ' Load module(xnjoy) does not exist.'
exit
endif

if ( -e $matlist_ld && -x $matlist_ld ) then
else
echo ' Load module(matlist_ld) does not exist.'
exit
endif

if ( -e $nj_inp_ld && -x $nj_inp_ld ) then
else
echo ' Load module(nj-inp_ld) does not exist.'
exit
endif

if ( -e $mk1xsd_ld && -x $mk1xsd_ld ) then
else
echo ' Load module(mk1xsd_ld) does not exist.'
exit
endif
if ( -e $mkxsdir_ld && -x $mkxsdir_ld ) then
else
echo ' Load module(mkxsdir_ld) does not exist.'
exit
endif

if ( -e $makxsf_ld && -x $makxsf_ld ) then
else
```

```
echo ' Load module(makxsf_ld) does not exist.'
exit
endif

if ( -e $awr_list ) then
else
echo ' File(awr_list) does not exist.'
exit
endif

# ---- initialize ----
set blank=' '
set name=(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
#
echo ' '
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo '  autonj ..... automatic production system of a mcnp cross section'
echo '               library from JENDL by the NJOY97 code.'
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo ' '

stp00:

stp01:
echo ' '
echo ' Please select an input mode.'
echo '  0 = input from a file.'
echo '  1 = interactive input on this terminal.'
echo ' ==========> ' | tr -d '\012'
set mode=$<
stp03:
echo ' '
echo ' Please select the option of using disk space (or producing files).'
echo '  0 = minimum disk size (mcnp library)'
echo '  1 = medium  disk size (mcnp library and output list)'
echo '  2 = maximum disk size (library, list and ace file)'
echo ' (maximum size is about 1.4 GB for JENDL-3.2.)'
echo ' (If blank, default size is minimum.)'
echo ' ==========> ' | tr -d '\012'
set disksp=$<
if ( $disksp == $blank ) set disksp=0
if ( $disksp == '0' ) goto stp04
if ( $disksp == '1' ) goto stp04
if ( $disksp == '2' ) goto stp04
echo '..... Unknown size option is detected.  Please reenter.'
goto stp03
stp04:
#
####
#### start of input data for autonj ####
####
if ( $mode == '0' ) then
   echo ' '
   echo ' Please enter a file name stored the input data.'
   echo ' (If blank, default name "inp-autonj" is used.)'
   echo ' ==========> ' | tr -d '\012'
   set infile=$<
   if ( $infile == $blank ) then
      set infile=inp-autonj
   else
      cp $infile inp-autonj
   endif
   goto stp10
endif
#
rm -f inp-autonj
```

```
set tt=JENDL-eval
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
if ( $mode == '1' ) then
   stp0:
   echo ' '
   echo ' Please enter the path name where JENDL file are located.'
   echo ' (When you finish entering all file names, please enter a blank.)'
   echo ' (If a directory name is given, all files in the directory are'
   echo '  processed. Only one directory name is allowed. The last'
   echo '  character must not be a slash (/).)'
   set ncount=0
   stp1:
      echo ' ==========> ' | tr -d '\012'
      set name1=$<
      if ( $name1 == $blank ) then
         if ( $ncount < 1 ) then
            echo '..... JENDL-3.2 file is not entered.  Please reenter.'
            goto stp0
         endif
         goto stp3
      endif
      if ( -d $name1 ) goto stp2
      if ( -e $name1 ) then
         set name2=$name1:e
         if ( $name2 == 'Z' || $name2 == 'gz' ) then
            cp $name1 JENDL-eval
            set name3=$name1:t
            if ( $name2 == 'Z' ) then
               uncompress -f JENDL-eval/$name3
            else if ( $name2 == 'gz' ) then
               gunzip JENDL-eval/$name3
            endif
            set name1=JENDL-eval/$name3:r
         endif
         echo $name1 >>! inp-autonj
         @ ncount = $ncount + 1
         set name[$ncount]=$name1
      else
         echo '             **********************************************'
         echo '             Above file name is not found.  Please reenter.'
      endif
      goto stp1
   stp2:

      uncompress -f $ {name1} /*.Z  >&! autonj_dummy

      echo 'directory path:' $name1 >>! inp-autonj
      echo '          list of files in the directory above'
      foreach name2 ($name1/*)
         set name3=$name2:e
         if ( $name3 == 'Z' || $name3 == 'gz' ) then
            cp $name2 JENDL-eval
            set name4=$name2:t
            if ( $name3 == 'Z' ) then
               uncompress -f JENDL-eval/$name4
            else if ( $name3 == 'gz' ) then
               gunzip JENDL-eval/$name4
```

```
         endif
         set name5=JENDL-eval/$name4:r
      else
         set name5=$name2
      endif
      echo $name5 >>! inp-autonj
      @ ncount = $ncount + 1
      set name[$ncount]=$name5
      echo '             ' $name5
   end
stp3:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the MAT numbers of nuclides that you want to process.'
echo ' (If blank, all nuclides contained in the specified files are processed.)'
stp4:
   echo ' ==========> ' | tr -d '\012'
   set mat1=$<
   if ( $mat1 == $blank ) goto stp5
   echo $mat1 >>! inp-autonj
   goto stp4
stp5:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the processing temperature with NJOY97.'
echo ' (If blank, default temperature is 293 K.)'
echo ' (for example:  300       <--- 300 kelvin)'
echo ' (               27C      <--- 27 celsius = 300.15 K)'
echo ' ==========> ' | tr -d '\012'
set tempe=$<
if ( $tempe == $blank ) set tempe=293.
echo $tempe >>! inp-autonj
echo ' '
echo ' Please enter a library identification number in zaid for MCNP.'
echo ' (It corresponds to the "suff" parameter in input data of NJOY/ACER.)'
echo ' (It must be integer with two degits.)'
echo ' (If blank, default identification number is 38. (If Fe, 26000.38c))'
echo ' ==========> ' | tr -d '\012'
set suff=$<
if ( $suff == $blank ) set suff=38
echo $suff >>! inp-autonj
echo ' '
echo ' Please enter a tolerance of pointwise cross section data.'
echo ' (The unit of tolerance is [%].)'
echo ' (If blank, default tolerance is 0.2 %.)'
echo ' ==========> ' | tr -d '\012'
set toler=$<
if ( $toler == $blank ) set toler=0.2
echo $toler >>! inp-autonj
stp6:
echo ' '
echo ' Please enter the maximum number of discrete gamma-ray energies.'
echo ' (It must be less than 1000 for MCNP-4B or 200 for the previous version.)'
echo ' (If blank, default number is 1000.)'
echo ' ==========> ' | tr -d '\012'
set dsgam=$<
if ( $dsgam == $blank ) set dsgam=1000
if ( $dsgam > 1000 ) then
   echo '..... The number is greater than 1000.  Please reenter.'
   goto stp6
endif
echo $dsgam >>! inp-autonj
echo ' '
echo ' Please enter the general name of nuclear data file.'
echo ' (It will be a part of the "hk" in input data of NJOY/ACER.)'
echo ' (Maximum 16 characters.)'
```

```
   echo ' (If blank, the name will be prepaired from the nuclear data file.)'
   echo ' (for example: jendl-3.2 )'
   stp7:
   echo '            (....+....1....+.)'
   echo ' ==========> ' | tr -d '\012'
   set titl=$<
   if ( $titl == $blank ) then
      echo ' ' >>! inp-autonj
   else
      set n= ` expr $titl : '.*' `

      if ( $n > 16 ) then
         echo '..... The number of characters in name exceeded 16.  Please reenter.'
         goto stp7
      else
         echo $titl >>! inp-autonj
      endif
   endif
else
   echo ' *** fatal error *** illegal mode was detedted; mode=' $mode
   exit
endif
stp10:
#
echo ' '
echo ' Please enter types of MCNP cross section library.'
echo ' (If blank, default type of library is Type-2.)'
echo '  1 = Type-1 (text format)'
echo '  2 = Type-2 (binary format)'
echo ' 12 = Type-1 and Type-2'
echo ' ==========> ' | tr -d '\012'
set libf=$<
if ( $libf == $blank ) then
   set libf=2
   goto stp11
else if ( $libf == '1' || $libf == '2' || $libf == '12' ) then
   goto stp11
endif
goto stp10
stp11:
echo ' '
echo ' Please enter a file name of the MCNP cross section library.'
echo ' (Maximum 7 characters.)'
echo ' (When the given name is "fsxlib2", the complete name is "fsxlib21",'
echo '  if it is Type-1.)'
echo ' (If blank, default library name is "fsxlb32".)'
stp12:
echo ' ==========> ' | tr -d '\012'
set libnm=$<
if ( $libnm == $blank ) set libnm=fsxlb32

   set n= ` expr $libnm : '.*' `

if ( $n > 7 ) then
   echo '..... The number of character of library name exceeded 7.  Please reenter.'
   goto stp12
endif
echo ' '
if ( $libf == '1' ) then
   echo ' Please enter a file name of the directory file for the library.'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
else if ( $libf == '2' ) then
   echo ' Please enter a file name of the directory file for the library.'
```

```
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 2
else if ( $libf == '12' ) then
   echo ' Please enter a file name of the directory files for the library.'
   echo ' --- type-1 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
   echo ' --- type-2 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm2=$<
   if ( $dirnm2 == $blank ) set dirnm2=xsdir.$ {libnm} 2
endif
####
#### end of input data for autonj ####
####
echo ' '
echo ' ::::                                                    ::::'
echo ' :::: Start the processing stage of nuclides by NJOY97 ::::'
echo ' ::::                                                    ::::'
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
#
if ( -e autonj-stop ) rm -f autonj-stop
#
##############
####  xnjoy : fortran program for processing the nuclear data file
####          (njoy97.45/autonj version)
##############
#

#
###############
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
###############
#
#---------
$matlist_ld
#---------
if ( -e autonj-stop ) goto err1
#
###############
####  nj-inp  : fortran program for making the input data of njoy97
###############
#

#
###############
####  mk1xsd  : fortran program for making the new single mcnp library
####            with type-1 from single dir and ace file
###############
#

set tt=ACE-FILE
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
```

```
else
   mkdir $tt
endif
set tt=OUT-FILE
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
rm -f autonj-count
set count=0
echo $count > autonj-count
rm -f autonj-xsdir acef1
#
####
#### loop of processing of all nuclides requested
#### execute the njoy97 code
####
stp20:
@ count++
if ( -e autonj-count ) then
#  ---------
   $nj_inp_ld
#  ---------
   if ( -e autonj-stop ) goto err1
else
   goto stp30
endif
echo ' ' $count '..... processing nuclide ..... '  ` cat autonj-now `
#-------------
$xnjoy < input  >&! autonj_dummy
#-------------
if ( -e tape91 ) then
   if ( -z tape91 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape91 ACE-FILE/ace.$count
   else
      mv tape91 ACE-FILE/ace.1
   endif
else
   goto stp21
endif
if ( -e tape92 ) then
   if ( -z tape92 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape92 ACE-FILE/dir.$count
   else
      mv tape92 ACE-FILE/dir.1
   endif
else
   goto stp21
endif
if ( $disksp == '0' || $disksp == '1' ) then
#  ---------
   $mk1xsd_ld  >&! autonj_dummy
#  ---------
   if ( -e autonj-stop ) goto err1
   rm -f ACE-FILE/*
endif
if ( $disksp == '0' ) then
   rm output
else
```

```
   mv output OUT-FILE/out.$count
   compress OUT-FILE/out.$count
endif
rm -f tape* autonj_dummy
goto stp20
stp21:
echo ' *** fatal error *** ace file or directory file was not found or zero-size.'
echo '                     njoy process was not completed.'
if ( -e tape91 | | -e tape92 ) ls -l tape9?
exit
#
stp30:
@ count = $count - 1
echo ' total number of processed nuclides ..... ' $count
#
if ( $disksp == '0' | | $disksp == '1' ) goto stp31
#
###############
####  mkxsdir : fortran program for making the new single mcnp library
####            with type-1
###############
#
#---------
$mkxsdir_ld  >&! autonj_dummy
#---------
if ( -e autonj-stop ) goto err1
#
stp31:
if ( $libf == '1' ) goto stp32
#
###############
####  makxsf  : fortran program for converting to the requested mcnp
####            library format (makxsf took from mcnp-4a)
###############
#

if ( $libf == '2' ) then
   set dirnm=$dirnm1
else if ( $libf == '12' ) then
   set dirnm=$dirnm2
endif
rm -f tprint specs $ {libnm} 2 $dirnm
# convert from type-1 (text) to type-2 (binary) format
echo 'autonj-xsdir ' $dirnm              >> specs
echo 'acef1 ' $ {libnm} 2  2  2048  512   >> specs
echo ' '                                 >> specs
#--------
$makxsf_ld  >&! autonj_dummy
#--------
stp32:
if ( $libf == '1' | | $libf == '12' ) then
   sed "s/acef1/$ {libnm} 1/" < autonj-xsdir > $dirnm1
   mv acef1 $ {libnm} 1
endif
echo ' '
echo ' file names of produced MCNP library:'
echo ' ------------------------------------'
if ( $libf == '1' | | $libf == '12' ) then
   set ns= ` ls -l $ {libnm} 1 `
   echo ' type-1 (text) form'
   echo '      cross section library .....' $ {libnm} 1 '     ' $ns[5] '[bytes]'
   echo '      directory .................' $dirnm1
endif
if ( $libf == '2' | | $libf == '12' ) then
   set ns= ` ls -l $ {libnm} 2 `
   echo ' type-2 (binary) form'
```

```
   echo '      cross section library .....' ${libnm} 2 '    ' $ns[5] '[bytes]'
   echo '      directory ................' $dirnm
endif
echo ' '
if ( $disksp == '1' || $disksp == '2' ) then
   echo ' Output-list file of njoy97 for each nuclide was stored into OUT-FILE/out.*.Z.
endif
if ( $disksp == '2' ) then
   echo ' ACE file of njoy97 for each nuclide was stored into ACE-FILE/ace.*.'
endif

####
#### ending process (including remove files)
####
if ( -e autonj-xsdir ) rm -f autonj-xsdir
if ( -e acef1 ) rm -f acef1
rm -f autonj-matlist autonj-now autonj_dummy
rm -f input tprint specs
rm -fR JENDL-eval
if ( $disksp == '0' || $disksp == '1' ) then
   rm -fR ACE-FILE
endif
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
echo ' The autonj process is successfully terminated.'
echo ' Congratulations and good-bye.'
echo ' ..........................'
exit
#
####
#### error process
####
err1:
echo ' '
echo ' * * * fatal error * * * fatal error * * * fatal error * * *'
echo ' The error condition below was detected in an execution program.'
cat autonj-stop
echo ' autonj was stop... .. .'
exit
```

付録 D 実行シェルスクリプト (autonjB.sh : バッチ処理版)

```
#! /bin/csh
#
# ##########################################################################
#  autonj.sh  :  shell script for producing automatically the continuous
#                energy cross section library for the mcnp code from
#                jendl by the njoy97 code.
# ##########################################################################
#    Japan Atomic Energy Research Institute (JAERI)
#    coordinator: Dr. F. Maekawa (FNS) and Dr. K. Sakurai (TCA)
# ##########################################################################
#    produced by K. Kosako (SAEI) on Jan. 29, 1999.
# ##########################################################################
#    UPDATE:
# 15 Apr 1999 : (N.Kawasaki)
#        Convert original version into AP3000 version.
# 15 Apr 1999 : (N.Kawasaki)
#        Change the running mode from 'Interactive' to 'Batch'.
# ##########################################################################
#
# ---- Existence check for load modules of autonj system ----
# ##########################################################################
#    xnjoy, matlist_ld, nj-inp_ld, mk1xsd_ld, mkxsdir_ld, makxsf_ld
# ##########################################################################
set xnjoy=/dg06/center/codelib/autonj/autonjLM/xnjoy
set matlist_ld=/dg06/center/codelib/autonj/autonjLM/matlist_ld
set nj_inp_ld=/dg06/center/codelib/autonj/autonjLM/nj-inp_ld
set mk1xsd_ld=/dg06/center/codelib/autonj/autonjLM/mk1xsd_ld
set mkxsdir_ld=/dg06/center/codelib/autonj/autonjLM/mkxsdir_ld
set makxsf_ld=/dg06/center/codelib/autonj/autonjLM/makxsf_ld
set awr_list=/dg06/center/codelib/autonj/src/Utility/awr-list
if ( -e $xnjoy && -x $xnjoy ) then
else
echo ' Load module(xnjoy) does not exist.'
exit
endif

if ( -e $matlist_ld && -x $matlist_ld ) then
else
echo ' Load module(matlist_ld) does not exist.'
exit
endif

if ( -e $nj_inp_ld && -x $nj_inp_ld ) then
else
echo ' Load module(nj-inp_ld) does not exist.'
exit
endif

if ( -e $mk1xsd_ld && -x $mk1xsd_ld ) then
else
echo ' Load module(mk1xsd_ld) does not exist.'
exit
endif
if ( -e $mkxsdir_ld && -x $mkxsdir_ld ) then
else
echo ' Load module(mkxsdir_ld) does not exist.'
exit
endif

if ( -e $makxsf_ld && -x $makxsf_ld ) then
else
```

```
echo ' Load module(makxsf_ld) does not exist.'
exit
endif

if ( -e $awr_list ) then
else
echo ' File(awr_list) does not exist.'
exit
endif

# ---- initialize ----
set blank=' '
set name=(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
#
echo ' '
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo '  autonj ..... automatic production system of a mcnp cross section'
echo '               library from JENDL by the NJOY97 code.'
echo ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::'
echo ' '

stp00:

stp01:
echo ' '
echo ' Please select an input mode.'
echo '  0 = input from a file.'
echo '  1 = interactive input on this terminal.'
echo ' ==========> ' | tr -d '\012'
set mode=$<
stp03:
echo ' '
echo ' Please select the option of using disk space (or producing files).'
echo '  0 = minimum disk size (mcnp library)'
echo '  1 = medium  disk size (mcnp library and output list)'
echo '  2 = maximum disk size (library, list and ace file)'
echo ' (maximum size is about 1.4 GB for JENDL-3.2.)'
echo ' (If blank, default size is minimum.)'
echo ' ==========> ' | tr -d '\012'
set disksp=$<
if ( $disksp == $blank ) set disksp=0
if ( $disksp == '0' ) goto stp04
if ( $disksp == '1' ) goto stp04
if ( $disksp == '2' ) goto stp04
echo '..... Unknown size option is detected.  Please reenter.'
goto stp03
stp04:
#
####
#### start of input data for autonj ####
####
if ( $mode == '0' ) then
   echo ' '
   echo ' Please enter a file name stored the input data.'
   echo ' (If blank, default name "inp-autonj" is used.)'
   echo ' ==========> ' | tr -d '\012'
   set infile=$<
   if ( $infile == $blank ) then
      set infile=inp-autonj
   else
      cp $infile inp-autonj
   endif
   goto stp10
endif
#
rm -f inp-autonj
```

```
set tt=JENDL-eval
if ( -e $tt ) then
   if ( -d $tt ) then
      rm -f $tt/* >&! autonj_dummy
   else
      rm -f $tt
      mkdir $tt
   endif
else
   mkdir $tt
endif
if ( $mode == '1' ) then
   stp0:
   echo ' '
   echo ' Please enter the path name where JENDL file are located.'
   echo ' (When you finish entering all file names, please enter a blank.)'
   echo ' (If a directory name is given, all files in the directory are'
   echo '  processed. Only one directory name is allowed. The last'
   echo '  character must not be a slash (/).)'
   set ncount=0
   stp1:
      echo ' ==========> ' | tr -d '\012'
      set name1=$<
      if ( $name1 == $blank ) then
         if ( $ncount < 1 ) then
            echo '..... JENDL-3.2 file is not entered.  Please reenter.'
            goto stp0
         endif
         goto stp3
      endif
      if ( -d $name1 ) goto stp2
      if ( -e $name1 ) then
         set name2=$name1:e
         if ( $name2 == 'Z' | | $name2 == 'gz' ) then
            cp $name1 JENDL-eval
            set name3=$name1:t
            if ( $name2 == 'Z' ) then
               uncompress -f JENDL-eval/$name3
            else if ( $name2 == 'gz' ) then
               gunzip JENDL-eval/$name3
            endif
            set name1=JENDL-eval/$name3:r
         endif
         echo $name1 >>! inp-autonj
         @ ncount = $ncount + 1
         set name[$ncount]=$name1
      else
         echo '             ********************************************'
         echo '             Above file name is not found.  Please reenter.'
      endif
      goto stp1
   stp2:

      uncompress -f $ {name1} /*.Z  >&! autonj_dummy

      echo 'directory path:' $name1 >>! inp-autonj
      echo '          list of files in the directory above'
      foreach name2 ($name1/*)
         set name3=$name2:e
         if ( $name3 == 'Z' | | $name3 == 'gz' ) then
            cp $name2 JENDL-eval
            set name4=$name2:t
            if ( $name3 == 'Z' ) then
               uncompress -f JENDL-eval/$name4
            else if ( $name3 == 'gz' ) then
               gunzip JENDL-eval/$name4
```

```
         endif
         set name5=JENDL-eval/$name4:r
      else
         set name5=$name2
      endif
      echo $name5 >>! inp-autonj
      @ ncount = $ncount + 1
      set name[$ncount]=$name5
      echo '             ' $name5
   end
stp3:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the MAT numbers of nuclides that you want to process.'
echo ' (If blank, all nuclides contained in the specified files are processed.)'
stp4:
   echo ' ==========> ' | tr -d '\012'
   set mat1=$<
   if ( $mat1 == $blank ) goto stp5
   echo $mat1 >>! inp-autonj
   goto stp4

stp5:
echo ' ' >>! inp-autonj
echo ' '
echo ' Please enter the processing temperature with NJOY97.'
echo ' (If blank, default temperature is 293 K.)'
echo ' (for example:  300       <--- 300 kelvin)'
echo ' (               27C      <--- 27 celsius = 300.15 K)'
echo ' ==========> ' | tr -d '\012'
set tempe=$<
if ( $tempe == $blank ) set tempe=293.
echo $tempe >>! inp-autonj
echo ' '
echo ' Please enter a library identification number in zaid for MCNP.'
echo ' (It corresponds to the "suff" parameter in input data of NJOY/ACER.)'
echo ' (It must be integer with two degits.)'
echo ' (If blank, default identification number is 38. (If Fe, 26000.38c))'
echo ' ==========> ' | tr -d '\012'
set suff=$<
if ( $suff == $blank ) set suff=38
echo $suff >>! inp-autonj
echo ' '
echo ' Please enter a tolerance of pointwise cross section data.'
echo ' (The unit of tolerance is [%].)'
echo ' (If blank, default tolerance is 0.2 %.)'
echo ' ==========> ' | tr -d '\012'
set toler=$<
if ( $toler == $blank ) set toler=0.2
echo $toler >>! inp-autonj
stp6:
echo ' '
echo ' Please enter the maximum number of discrete gamma-ray energies.'
echo ' (It must be less than 1000 for MCNP-4B or 200 for the previous version.)'
echo ' (If blank, default number is 1000.)'
echo ' ==========> ' | tr -d '\012'
set dsgam=$<
if ( $dsgam == $blank ) set dsgam=1000
if ( $dsgam > 1000 ) then
   echo '..... The number is greater than 1000.  Please reenter.'
   goto stp6
endif
echo $dsgam >>! inp-autonj
echo ' '
echo ' Please enter the general name of nuclear data file.'
echo ' (It will be a part of the "hk" in input data of NJOY/ACER.)'
```

```
   echo ' (Maximum 16 characters.)'
   echo ' (If blank, the name will be prepaired from the nuclear data file.)'
   echo ' (for example: jendl-3.2 )'
   stp7:
   echo '           (....+....1....+.)'
   echo ' ==========> ' | tr -d '\012'
   set titl=$<
   if ( $titl == $blank ) then
      echo ' ' >>! inp-autonj
   else
      set n= ` expr $titl : '.*' `
      if ( $n > 16 ) then
         echo '..... The number of characters in name exceeded 16.  Please reenter.'
         goto stp7
      else
         echo $titl >>! inp-autonj
      endif
   endif
else
   echo ' *** fatal error *** illegal mode was detedted; mode=' $mode
   exit
endif
stp10:
#
echo ' '
echo ' Please enter types of MCNP cross section library.'
echo ' (If blank, default type of library is Type-2.)'
echo '  1 = Type-1 (text format)'
echo '  2 = Type-2 (binary format)'
echo ' 12 = Type-1 and Type-2'
echo ' ==========> ' | tr -d '\012'
set libf=$<
if ( $libf == $blank ) then
   set libf=2
   goto stp11
else if ( $libf == '1' || $libf == '2' || $libf == '12' ) then
   goto stp11
endif
goto stp10
#
stp11:
echo ' '
echo ' Please enter a file name of the MCNP cross section library.'
echo ' (Maximum 7 characters.)'
echo ' (When the given name is "fsxlib2", the complete name is "fsxlib21",'
echo '  if it is Type-1.)'
echo ' (If blank, default library name is "fsxlb32".)'
stp12:
echo ' ==========> ' | tr -d '\012'
set libnm=$<
if ( $libnm == $blank ) set libnm=fsxlb32

   set n= ` expr $libnm : '.*' `

if ( $n > 7 ) then
   echo '..... The number of character of library name exceeded 7.  Please reenter.'
   goto stp12
endif
echo ' '
set dirnm2=$blank
if ( $libf == '1' ) then
   echo ' Please enter a file name of the directory file for the library.'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
```

```
else if ( $libf == '2' ) then
   echo ' Please enter a file name of the directory file for the library.'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 2
else if ( $libf == '12' ) then
   echo ' Please enter a file name of the directory files for the library.'
   echo ' --- type-1 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '1".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm1=$<
   if ( $dirnm1 == $blank ) set dirnm1=xsdir.$ {libnm} 1
   echo ' --- type-2 library ---'
   echo ' (If blank, default library name is "xsdir.'$ {libnm} '2".)'
   echo ' ==========> ' | tr -d '\012'
   set dirnm2=$<
   if ( $dirnm2 == $blank ) set dirnm2=xsdir.$ {libnm} 2
endif
####
#### end of input data for autonj ####
####

####
#### Get a name of queue class for Batch Job
####
   echo " End of input data for autonj "
   echo ' '
   echo " Q-class | CPU/elapse   | memory     "
   echo ' '
   echo " ss      |    1h/ 2h   |  200MB "
   echo " sm      |    6h/12h   |  200MB "
   echo " sl      |   48h/96h   |  200MB "
   echo ' '
   echo " ms      |    1h/ 2h   |  500MB "
   echo " mm      |    6h/12h   |  500MB "
   echo " ml      |   48h/96h   |  500MB "
   echo ' '
   echo " ls      |    1h/ 2h   | 2000MB "
   echo " lm      |    6h/12h   | 2000MB "
   echo " ll      |   48h/96h   | 2000MB "
   echo ' '
   echo ' Input a name of queue class =====> ' | tr -d '\012'
   set Q=$<

echo "class : $Q"
echo ' OK ?  (y/n) ==========> ' | tr -d '\012'
set YN=$<
if ( $YN != 'y' ) then
   if ( $YN != 'Y' ) then
      exit
   endif
endif

cat > go.autonj.sh << END_SH
#!/bin/csh
#

set anjLM=/dg06/center/codelib/autonj/autonjLM
ln -s \$anjLM/xnjoy      \$QSUB_WORKDIR/xnjoy
ln -s \$anjLM/matlist_ld \$QSUB_WORKDIR/matlist_ld
ln -s \$anjLM/nj-inp_ld  \$QSUB_WORKDIR/nj_inp_ld
ln -s \$anjLM/mk1xsd_ld  \$QSUB_WORKDIR/mk1xsd_ld
ln -s \$anjLM/mkxsdir_ld \$QSUB_WORKDIR/mkxsdir_ld
ln -s \$anjLM/makxsf_ld  \$QSUB_WORKDIR/makxsf_ld
```

```
#
cd \$QSUB_WORKDIR
#

echo ' '
echo ' ::::                                                          ::::'
echo ' :::: Start the processing stage of nuclides by NJOY97         ::::'
echo ' ::::                                                          ::::'
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
#
if ( -e autonj-stop ) rm -f autonj-stop
#
##############
####  xnjoy : fortran program for processing the nuclear data file
####          (njoy97.45/autonj version)
##############
#

#
################
####  matlist : fortran program for making the list of mat number of
####            processing nuclides
################
#
#---------
matlist_ld
#---------
if ( -e autonj-stop ) goto err1
#
################
####  nj-inp  : fortran program for making the input data of njoy97
################
#

#
################
####  mk1xsd  : fortran program for making the new single mcnp library
####            with type-1 from single dir and ace file
################
#

set tt=ACE-FILE
if ( -e \$tt ) then
   if ( -d \$tt ) then
      rm -f \$tt/* >&! autonj_dummy
   else
      rm -f \$tt
      mkdir \$tt
   endif
else
   mkdir \$tt
endif
set tt=OUT-FILE
if ( -e \$tt ) then
   if ( -d \$tt ) then
      rm -f \$tt/* >&! autonj_dummy
   else
      rm -f \$tt
      mkdir \$tt
   endif
else
   mkdir \$tt
endif
rm -f autonj-count
set count=0
echo \$count > autonj-count
```

```
rm -f autonj-xsdir acef1
#
####
#### loop of processing of all nuclides requested
#### execute the njoy97 code
####
stp20:
@ count++
if ( -e autonj-count ) then
#  ---------
   nj_inp_ld
#  ---------
   if ( -e autonj-stop ) goto err1
else
   goto stp30
endif
echo ' ' \$count '..... processing nuclide ..... ' \ ` cat autonj-now\ `
#------------
xnjoy < input  >&! autonj_dummy
#------------
if ( -e tape91 ) then
   if ( -z tape91 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape91 ACE-FILE/ace.\$count
   else
      mv tape91 ACE-FILE/ace.1
   endif
else
   goto stp21
endif
if ( -e tape92 ) then
   if ( -z tape92 ) goto stp21
   if ( $disksp == '2' ) then
      mv tape92 ACE-FILE/dir.\$count
   else
      mv tape92 ACE-FILE/dir.1
   endif
else
   goto stp21
endif
if ( $disksp == '0' || $disksp == '1' ) then
#  ---------
   mk1xsd_ld  >&! autonj_dummy
#  ---------
   if ( -e autonj-stop ) goto err1
   rm -f ACE-FILE/*
endif
if ( $disksp == '0' ) then
   rm output
else
   mv output OUT-FILE/out.\$count
   compress OUT-FILE/out.\$count
endif
rm -f tape* autonj_dummy
goto stp20

stp21:
echo ' *** fatal error *** ace file or directory file was not found or zero-size.'
echo '                     njoy process was not completed.'
if ( -e tape91 || -e tape92 ) ls -l tape9?
goto stp99

#
stp30:
@ count = \$count - 1
echo ' total number of processed nuclides ..... ' \$count
#
```

```
if ( $disksp == '0' | | $disksp == '1' ) goto stp31
#
###############
####  mkxsdir : fortran program for making the new single mcnp library
####            with type-1
###############
#

#---------
mkxsdir_ld  >&! autonj_dummy
#---------
if ( -e autonj-stop ) goto err1
#

stp31:
if ( $libf == '1' ) goto stp32
#
###############
####  makxsf  : fortran program for converting to the requested mcnp
####            library format (makxsf took from mcnp-4a)
###############
#
if ( $libf == '2' ) then
   set dirnm=$dirnm1
else if ( $libf == '12' ) then
   set dirnm=$dirnm2
endif
rm -f tprint specs $ {libnm} 2 \$dirnm
# convert from type-1 (text) to type-2 (binary) format
echo 'autonj-xsdir ' \$dirnm              >> specs
echo 'acef1 ' $ {libnm} 2  2  2048  512   >> specs
echo ' '                                  >> specs
#--------
makxsf_ld  >&! autonj_dummy
#--------

stp32:
if ( $libf == '1' | | $libf == '12' ) then
   sed "s/acef1/$ {libnm} 1/" < autonj-xsdir > $dirnm1
   mv acef1 $ {libnm} 1
endif
echo ' '
echo ' file names of produced MCNP library:'
echo ' ------------------------------------'
if ( $libf == '1' | | $libf == '12' ) then
   set ns=\ ` ls -l $ {libnm} 1\ `
   echo ' type-1 (text) form'
   echo '      cross section library .....' $ {libnm} 1 '    ' \$ns[5] '[bytes]'
   echo '      directory .................' $dirnm1
endif
if ( $libf == '2' | | $libf == '12' ) then
   set ns=\ ` ls -l $ {libnm} 2\ `
   echo ' type-2 (binary) form'
   echo '      cross section library .....' $ {libnm} 2 '    ' \$ns[5] '[bytes]'
   echo '      directory .................' \$dirnm
endif
echo ' '
if ( $disksp == '1' | | $disksp == '2' ) then
   echo ' Output-list file of njoy97 for each nuclide was stored into OUT-FILE/out.*.Z.
endif
if ( $disksp == '2' ) then
   echo ' ACE file of njoy97 for each nuclide was stored into ACE-FILE/ace.*.'
endif
####
#### ending process (including remove files)
####
```

```
if ( -e autonj-xsdir ) rm -f autonj-xsdir
if ( -e acef1 ) rm -f acef1
rm -f autonj-matlist autonj-now autonj_dummy

#rm -f *_ld input tprint specs
rm -f input tprint specs
rm -fR JENDL-eval

if ( $disksp == '0' || $disksp == '1' ) then
   rm -fR ACE-FILE
endif
echo ' '
date '+ DATE: %m/%d/%y%n TIME: %H:%M:%S'
echo ' '
echo ' The autonj process is successfully terminated.'
echo ' Congratulations and good-bye.'
echo ' ..........................'
goto stp99
#
####
#### error process
####
err1:
echo ' '
echo ' * * * fatal error * * * fatal error * * * fatal error * * *'
echo ' The error condition below was detected in an execution program.'
cat autonj-stop
echo ' autonj was stop... .. .'
#
####
#### Remove load modules in current directory
####
stp99:
rm xnjoy *_ld

END_SH

qsub -q $Q -C autonj -eo go.autonj.sh
```

国際単位系（SI）と換算表

表1　SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表2　SIと併用される単位

名称	記号
分，時，日	min，h，d
度，分，秒	°，′，″
リットル	l，L
トン	t
電子ボルト	eV
原子質量単位	u

1 eV=1.60218×10^{-19}J

1 u=1.66054×10^{-27}kg

表3　固有の名称をもつSI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s^{-1}
力	ニュートン	N	$m\cdot kg/s^2$
圧力，応力	パスカル	Pa	N/m^2
エネルギー,仕事,熱量	ジュール	J	N·m
工率，放射束	ワット	W	J/s
電気量，電荷	クーロン	C	A·s
電位，電圧，起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンス	S	A/V
磁束	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m^2
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	℃	
光束	ルーメン	lm	cd·sr
照度	ルクス	lx	lm/m^2
放射能	ベクレル	Bq	s^{-1}
吸収線量	グレイ	Gy	J/kg
線量等量	シーベルト	Sv	J/kg

表4　SIと共に暫定的に維持される単位

名称	記号
オングストローム	Å
バーン	b
バール	bar
ガル	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

1 Å=0.1nm=10^{-10}m

1 b=100fm^2=10^{-28}m^2

1 bar=0.1MPa=10^{5}Pa

1 Gal=1cm/s^2=10^{-2}m/s^2

1 Ci=3.7×10^{10}Bq

1 R=2.58×10^{-4}C/kg

1 rad=1cGy=10^{-2}Gy

1 rem=1cSv=10^{-2}Sv

表5　SI接頭語

倍数	接頭語	記号
10^{18}	エクサ	E
10^{15}	ペタ	P
10^{12}	テラ	T
10^{9}	ギガ	G
10^{6}	メガ	M
10^{3}	キロ	k
10^{2}	ヘクト	h
10^{1}	デカ	da
10^{-1}	デシ	d
10^{-2}	センチ	c
10^{-3}	ミリ	m
10^{-6}	マイクロ	μ
10^{-9}	ナノ	n
10^{-12}	ピコ	p
10^{-15}	フェムト	f
10^{-18}	アト	a

(注)

1．表1－5は「国際単位系」第5版，国際度量衡局1985年刊行による。ただし，1 eVおよび1 uの値はCODATAの1986年推奨値によった。

2．表4には海里，ノット，アール，ヘクタールも含まれているが日常の単位なのでここでは省略した。

3．barは，JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。

4．EC閣僚理事会指令ではbar，barnおよび「血圧の単位」mmHgを表2のカテゴリーに入れている。

換算表

力	N(=10^{5}dyn)	kgf	lbf
	1	0.101972	0.224809
	9.80665	1	2.20462
	4.44822	0.453592	1

粘度　1 Pa·s(N·s/m^2)=10 P(ポアズ)(g/(cm·s))

動粘度　1m^2/s=10^{4}St(ストークス)(cm^2/s)

圧力	MPa(=10bar)	kgf/cm^2	atm	mmHg(Torr)	lbf/in^2(psi)
	1	10.1972	9.86923	7.50062×10^3	145.038
	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322×10^{-4}	1.35951×10^{-3}	1.31579×10^{-3}	1	1.93368×10^{-2}
	6.89476×10^{-3}	7.03070×10^{-2}	6.80460×10^{-2}	51.7149	1

エネルギー・仕事・熱量	J(=10^7 erg)	kgf·m	kW·h	cal(計量法)	Btu	ft·lbf	eV
	1	0.101972	2.77778×10^{-7}	0.238889	9.47813×10^{-4}	0.737562	6.24150×10^{18}
	9.80665	1	2.72407×10^{-6}	2.34270	9.29487×10^{-3}	7.23301	6.12082×10^{19}
	3.6×10^6	3.67098×10^5	1	8.59999×10^5	3412.13	2.65522×10^6	2.24694×10^{25}
	4.18605	0.426858	1.16279×10^{-6}	1	3.96759×10^{-3}	3.08747	2.61272×10^{19}
	1055.06	107.586	2.93072×10^{-4}	252.042	1	778.172	6.58515×10^{21}
	1.35582	0.138255	3.76616×10^{-7}	0.323890	1.28506×10^{-3}	1	8.46233×10^{18}
	1.60218×10^{-19}	1.63377×10^{-20}	4.45050×10^{-26}	3.82743×10^{-20}	1.51857×10^{-22}	1.18171×10^{-19}	1

1 cal= 4.18605J（計量法）

= 4.184J（熱化学）

= 4.1855J（15℃）

= 4.1868J（国際蒸気表）

仕事率　1 PS(仏馬力)

= 75 kgf·m/s

= 735.499W

放射能	Bq	Ci
	1	2.70270×10^{-11}
	3.7×10^{10}	1

吸収線量	Gy	rad
	1	100
	0.01	1

照射線量	C/kg	R
	1	3876
	2.58×10^{-4}	1

線量当量	Sv	rem
	1	100
	0.01	1

(86年12月26日現在)

原子力コードの高速化（移植編）
—平成11年度作業報告書—