

JAERI-M
86-159

構造化プログラミング支援ツールEOS77

1986年11月

徳田 伸二・常松 俊秀・竹田 辰興

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）
あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

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

© Japan Atomic Energy Research Institute, 1986

編集兼発行 日本原子力研究所
印刷 日立高速印刷株式会社

構造化プログラミング支援ツールEOS 77

日本原子力研究所那珂研究所核融合研究部

徳田伸二・常松俊秀・竹田辰興

(1986年10月11日受理)

TRITON計画の一環として構造化プログラミングを支援するプリプロセッサEOS 77を開発した。EOS 77はすでに開発されているEOSの上位バージョンであり、FORTRAN 77に準拠している。さらにEOS 77は構造化プログラムを開発する上で有効ないくつかの拡張機能を備えている。このプリプロセッサ・システムは大規模計算機コードを開発し利用する上で極めて有効である。本報告書はEOS 77プリプロセッサ・システムの利用手引書である。

Structured Programming Supporting Tool EOS77

Shinji TOKUDA, Toshihide TSUNEMATSU and Tatsuoki TAKEDA

Department of Thermonuclear Fusion Research

Naka Fusion Research Establishment

Japan Atomic Energy Research Institute

Naka-machi, Naka-gun, Ibaraki-ken

(Received October 11, 1986)

As a part of the project TRITON a preprocessor (EOS77 preprocessor) which supports structured programming was developed. EOS77 is an upper-compatible version of the preprocessor EOS and is constructed so as to generate FORTRAN77 programs. EOS77 provides other functions to facilitate structured programming. The EOS77 preprocessor system is very useful in development and use of large-scale computer codes. This report is a user's manual of this system.

Keywords: EOS77 Preprocessor, TRITON Project, FORTRAN 77, Structured Programming, User's Manual

目 次

1. はじめに	1
2. EOS 77 の使用例	1
2.1 EOS 77 ソース・プログラムの例	2
2.2 プリプロセスの手順	3
3. EOS 77 文法	4
3.1 用語説明	4
3.2 サイズ定義	5
3.3 *COMDECK 文およびテキスト文	5
3.4 CG 文	6
3.5 *DECK 文	6
3.6 テキスト挿入命令	6
3.7 *DO 文と *END DO 文	8
3.8 PARAMETER 文の処理	8
3.9 行コメント	9
3.10 被挿入テキストの行番号	9
4. EOS 77 の使い方	9
4.1 入出力機番	9
4.2 オプション・パラメータ	10
4.3 オプション・パラメータの入力方法	11
5. まとめ	11
謝 辞	12
文 献	12
付録 1 EOS 77 プリプロセッサのエラーメッセージ	27
付録 2 JCL リスト	32

Contents

1. Introduction	1
2. Examples of use of EOS77	1
2.1 Examples of EOS77 source programmes	2
2.2 Examples of preprocessing	3
3. EOS77 syntax	4
3.1 Glossary	4
3.2 Size definition	5
3.3 *COMDECK statement and TEXT statement	5
3.4 CG statement	6
3.5 *DECK statement	6
3.6 Insertion commands of a text	6
3.7 *DO statement and *END DO statement	8
3.8 Treatment of PARAMETER statement	8
3.9 Line comment	9
3.10 Line numbers of inserted texts	9
4. How to use the EOS77 preprocessor	9
4.1 Logical unit number of data sets used in EOS77	9
4.2 Option parameters	10
4.3 Input of option parameters	11
5. Summary	11
Acknowledgments	12
References	12
Appendix 1 Error messages of the EOS77 preprocessor	27
Appendix 2 JCL procedure	32

1. はじめに

大規模計算機コードの開発を効率的にするためには構造化プログラミングが有力である。¹⁾ 構造化プログラミングでは、

- (1) 計算機コードの機能分割 (モジュール化)
- (2) 処理の流れが top-down であること
- (3) 各処理の局所性がよいこと

を重視する。しかしながら FORTRAN 言語やそのコンパイラは構造化プログラミングを支援するようには必ずしもできていない。FORTRAN77 の PARAMETER 文やブロック IF 文によってその点は改善されたがまだ不十分な点が多い。特に GOTO 文や DO 文に必要な文番号(文末)が上記(2)(3)を著しく損うことはよく知られている。現状では科学技術計算コードの大部分が FORTRAN で書かれているので構造化プログラミングを支援するためのプリプロセス言語が要求される。理論解析研究室でも主として上記(1)を補助するため、COMMON 領域中の配列サイズを自由にできるプリプロセッサ EOS を開発公表した。²⁾ その後、FORTRAN77が導入されたので、当研究室もそれに対応した EOS77 を開発した。本報告書は EOS77 の利用手引書である。

EOS77 は EOS の上位バージョンであり、既存の EOS プログラム (EOS 言語で書かれたプリプロセス前のソース・プログラム) もそのままコンパイル可能な FORTRAN プログラムに変換される。また EOS77 は「*DO」文を持っており、それを FORTRAN77 の「DO」文の代わり用いることによって文番号が全然ないプログラムを書くことができるので各処理単位の局所性のよいプログラミングが可能になる。さらに EOS77 はいくつかの拡張機能を持っている。

本報告書では、EOS を初めて使う利用者を考慮して、

第 2 章で EOS77 で書かれた簡単なソース・プログラムを例示しながら EOS77 の機能とプリプロセスの手順を説明する。

その後、第 3 章で EOS77 の文法を、第 4 章では EOS77 プリプロセッサの使用法を述べる。そして第 5 章では EOS77 に残されている課題について検討する。

2. EOS77 の使用例

本章では EOS77 でコーディングした簡単なプログラムを例にとって、EOS77 の機能とプリプロセッサを使って、FORTRAN プログラムに変換後、コンパイル・リンクまでの手順を具体的に示す。プログラムの内容は周期関数の定積分を台形公式で求めるものである。プログラム全体は示されていないが、EOS77 の主な機能を知るには十分である。また、計算機利用の現状を考慮して TSS 下で EOS77 プリプロセッサを利用する方法を示す。

1. はじめに

大規模計算機コードの開発を効率的にするためには構造化プログラミングが有力である。¹⁾ 構造化プログラミングでは、

- (1) 計算機コードの機能分割 (モジュール化)
- (2) 処理の流れが top-down であること
- (3) 各処理の局所性がよいこと

を重視する。しかしながら FORTRAN 言語やそのコンパイラは構造化プログラミングを支援するようには必ずしもできていない。FORTRAN77 の PARAMETER 文やブロック IF 文によってその点は改善されたがまだ不十分な点が多い。特に GOTO 文や DO 文に必要な文番号(文末)が上記(2)(3)を著しく損うことはよく知られている。現状では科学技術計算コードの大部分が FORTRAN で書かれているので構造化プログラミングを支援するためのプリプロセス言語が要求される。理論解析研究室でも主として上記(1)を補助するため、COMMON 領域中の配列サイズを自由にできるプリプロセッサ EOS を開発公表した。²⁾ その後、FORTRAN77が導入されたので、当研究室もそれに対応した EOS77 を開発した。本報告書は EOS77 の利用手引書である。

EOS77 は EOS の上位バージョンであり、既存の EOS プログラム (EOS 言語で書かれたプリプロセス前のソース・プログラム) もそのままコンパイル可能な FORTRAN プログラムに変換される。また EOS77 は「*DO」文を持っており、それを FORTRAN77 の「DO」文の代わり用いることによって文番号が全然ないプログラムを書くことができるので各処理単位の局所性のよいプログラミングが可能になる。さらに EOS77 はいくつかの拡張機能を持っている。

本報告書では、EOS を初めて使う利用者を考慮して、

第 2 章で EOS77 で書かれた簡単なソース・プログラムを例示しながら EOS77 の機能とプリプロセスの手順を説明する。

その後、第 3 章で EOS77 の文法を、第 4 章では EOS77 プリプロセッサの使用法を述べる。そして第 5 章では EOS77 に残されている課題について検討する。

2. EOS77 の使用例

本章では EOS77 でコーディングした簡単なプログラムを例にとって、EOS77 の機能とプリプロセッサを使って、FORTRAN プログラムに変換後、コンパイル・リンクまでの手順を具体的に示す。プログラムの内容は周期関数の定積分を台形公式で求めるものである。プログラム全体は示されていないが、EOS77 の主な機能を知るには十分である。また、計算機利用の現状を考慮して TSS 下で EOS77 プリプロセッサを利用する方法を示す。

2.1 EOS77ソース・プログラムの例

EOS ソース・プログラムはファイル名が J3331.EOS77EX.FORT77 である PO ファイルに格納されている (図 1)。

図 1 で先頭にあるメンバー名 #COM1 は「サイズ・ブロック」と「テキスト・ブロック」とよぶものからなっている。

残り 8 つのメンバー名には「プログラム・ブロック」とよぶ、FORTRAN で主プログラムおよび副プログラムに相当するプログラム (必ずしも対応は 1対1である必要はない) が格納されている。

図 2 にメンバー名 #COM1 の内容を示す。①がサイズ・ブロック (必ず先頭にある) で以下に現われるテキスト・ブロック中のサイズ変数 (NT および MAXCS) の値を設定している。この時、式による設定も可能である (サイズ式による定義)。また、「セパレーター ▼; ▼」の後にコメントを書くことができる。テキスト・ブロックは、第 1 コラムから書かれる *COMDECK 文 (図 2 の ②) とテキスト文 (図 2 の ③) から成る。プログラム・ブロック中のテキスト挿入命令 ▼ * CALL ▼ 文によってプリプロセッサは配列宣言子 (図 2 では DIMENSION) 中のサイズ変数に、サイズ・ブロックでされる定義に従って値を代入してプログラム・ブロック中に挿入する。*) その例を図 3 に示す。図 3a はメンバー名 INPUT に格納されているプログラム・ブロックで、プリプロセスの実行によって図 3b に示すコンパイル可能な FORTRAN プログラムが作り出される (プログラム・ブロックの作り方について後述)。この時、被挿入テキストの行番号は各 *CALL 文の行番号から始まる公差 1 の等差数列になっている。また *CALL 文の代わりに *INCLUDE 文が使える。

図 2 の ④は ¥付サイズの例であり、E 宣言文以外の FORTRAN 文に値を設定するのに使用できる。④はメンバー名 PRESET にあるプログラム・ブロック中の *INCLUDE 文 (図 4a) で展開され、ここではデフォルト値の設定に使われている (図 4b)。

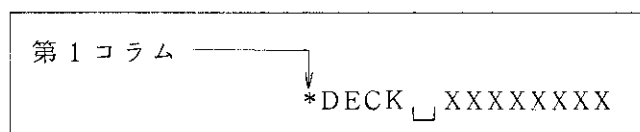
図 2 の ⑤は、メンバー名 DOCUM に格納されているプログラム・ブロック中にある %name 文 (図 5a ★印) によって、テキストの前後に引用符をつけて文字列として展開される (図 5b ★印)。展開される文字列は定積分の被積分関数を表わしている。なお、この DOCUM は、プログラム自身がドキュメントを作成する副プログラムで EOS77 の機能を利用している (ドキュメント機能の利用例については図 10 参照)。

図 2 の ⑥はメンバー名 FUNCCS のプログラム・ブロックで展開され (図 6.a), ⑤に相当する関数が関数副プログラム FT で定義される (図 6.b)。したがって、被積分関数を変更する場合、#COM1 の相当する部分を変更するだけでよく、他のメンバー名に格納されている EOS77 ソース・プログラムを変更する必要はない。また 図 2 の ⑦は *DO 文でプリプロセスによって FORTRAN の DO 文に変換される (図 6.b 参照)。この時、文番号は 99999 から始まって 1 ずつ小さい文番号が使われる。ただしユーザーの用いている文番号は除かれる。各 *DO 文に

*) DIMENSION 文の他に次の 7 つの宣言文に対して有効である: INTEGER, REAL, LOGICAL, COMPLEX, DOUBLEPRECISION, CHARACTER および COMMON 文, 以下これらの 8 つの宣言文を E 宣言文と呼ぶ。

対して *END DO 文が1つ必要である。

図 7. a にプログラム・ブロックの例を示す (メンバー名 DAIKEI)。各プログラム・ブロックの先頭には *DECK 文:



が必要である。(図 7. a の①) ここで 'XXXXXXXX' は DECK 名とよばれる英字で始まる 8 文字以内の英数字列である。

図 7. a の②は「変数名変更リスト」付きの *CALL 文である。この場合、プリプロセッサはテキスト中の変数名 FUNC を変数名 F に変えて展開する (図 7. b の②)。また、③はプログラム・ブロック中に書かれた *DO 文である。

2.2 プリプロセスの手順

プリプロセッサ EOS 77 のロード・モジュールはファイル名

J 2636. EOS 77V2. LOAD

に格納されており、原研計算機ユーザー全員に公開されている。プリプロセッサは PS ファイルに格納されている EOS ソース・プログラムを FORTRAN プログラムへの翻訳の対象にするため、EOS ソース・プログラムが PO ファイルに格納されている場合、前もって PS ファイルに移すことが必要である。この時、サイズ・ブロックが先頭にこなければならぬ (図 1 参照)。

TSS 環境下でプリプロセスする時の private catalogued procedure ETOF 77 と ETOF 77 CL を図 8 に示す。

前者は、PO ファイルの 1 メンバーに格納されているプログラム・ブロックを FORTRAN プログラムに変換した後、コンパイルを行なう。サイズ・ブロックおよびテキスト・ブロックはメンバー名 #COM1 に格納されていると仮定している。後者は PO ファイルに格納されている EOS ソース・プログラムから、コンパイルおよびリンクによって実行不可能なロード・モジュール (ファイル名のデフォルト値は Z. LOAD (TEMPNAME)) を作る。なお、いずれの場合も FORTRAN プログラムはファイル JXXXX. F 77. FORT 77 に作られる。

ここでは、ETOF 77CL の使用例を示す (図 9)。

すでに述べたように、このプログラムにはプログラムに関するドキュメントを作成する機能を持っている。すなわち、実行時に要求されるネームリスト入力に対して何も応答しない場合 (すなわち、すべてデフォルト値が使用された場合)、あるいは、間違った値が入力された場合、プログラムは、その内容および変数の説明書を出力して終了する。副プログラム DOCUM で説明書を出力する。図 10 に EOS 77 の機能を利用した DOCUM のリストを、また図 11 にその使用例を示す。このように EOS 77 を利用すれば、ロード・モジュールの内容を知ることができるので計算機コードの利用が容易になる。なお、EOS 77 プリプロセッサをバッチ・ジョブで使う場合の JCL を付録 2 に示す。

3. EOS77 文法

3.1 用語説明

(a) E 宣言文

次の 8 つの FORTRAN 宣言文を E 宣言文と総称する：

```

INTEGER
REAL
LOGICAL
COMPLEX
DOUBLEPRECISION
CHARACTER
COMMON
DIMENSION

```

(b) サイズ変数

配列のサイズを設定するパラメータ。英字で始まる 6 文字以内の英数字列で、すべて整数型として扱われる。サイズ変数はサイズ・ブロックの中で定義され、以下に述べるサイズ式の中および¥付きサイズ式の中で用いられる。

(c) サイズ式

配列のサイズを設定する算術式で E 宣言文の配列宣言子の括弧の中に書かれる。使用できる算術式は四則演算および $\nabla^{**}\nabla$ (FORTRAN 文法と同じ意味をもつ) である。

(d) ¥付サイズ式

$\nabla^{¥(式)}\nabla$ ($\nabla^{¥(N)}$, $\nabla^{¥(L*M+N)}$, $\nabla^{¥(L**2)}$ など) の形式で書かれるサイズ式でテキスト・ブロック内の任意のテキストに使用できる。

(e) サイズ・ブロック

サイズ変数を定義する。もしあれば必ずプログラムの先頭にある。

(f) テキスト・ブロック

プログラム・ブロックに挿入される一群のテキストを定義する。各テキスト・ブロックは *COMDECK 文とその後にあるテキスト文よりなる。テキスト文に、サイズ変数、サイズ式、¥付サイズ式が使用される。

(g) プログラム・ブロック

テキスト挿入命令や*DO文, その他のEOS 77の命令を含むFORTRAN形式のプログラム部分。一般に, EOF 77ソース・プログラムは, サイズ・ブロック, テキスト・ブロックおよびプログラム・ブロックから構成される。なお, すべてのブロックにおいて, 第1コラムが' C 'の時, その行はコメント行とみなされる (CG文参照)。

3.2 サイズ定義

- (a) サイズの定義には「値による定義」と「式による定義」がある。
- (b) 「式による定義」では, サイズ式を含む算術式が許され, 「値による定義」がすべて登録された後, 「式による定義」が実行される。
- (c) 使用フィールドは第1コラムから第71コラムである。第72コラムに英数字記号があると, それは継続記号とみなされる。継続は最大19行までである。

3.3 *COMDECK文およびテキスト文

- (a) *COMDECK文は第1コラムから書かれ, その後にCOMDECK名を書く。
- (b) COMDECK名は英字で始まる8文字以内の英数字か, その後に'#'と9ケタ以内の数字を付けたもので全体で16文字以内になければならない。

(例)

```
*COMDECK  □ TEXT
*COMDECK  □ TEXT#01
```

- (c) テキスト文には以下のものが書ける。

FORTRAN文

サイズ式を含むE宣言文

¥付きサイズ式を含むE宣言文以外のFORTRAN文

*DO文

CG文

なお, FORTRAN 77のPARAMETER文に対する処理については3.8参照。

(例)

```
*COMDECK  □ EXAMPL#01
COMMON A (M, N)
DIMENSION WA (M*N)
EQUIVALENCE (A (1, 2), WA (¥(M+1)))
KK = ¥((M*N)**2)
```

3.4 CG文

- (a) EOS77 プリプロセッサは第1カラムから書かれた `!CG!` をキーワードとみなして、その行をコメント行と見なさず `!付` サイズ式の評価を行なう。(図12の①参照)
- (b) CG文に対する継続行として `!CG+` を用いる。

<注意>

CG文はEOS77 と他のプリプロセッサとの共存を考慮して用意されていて、キーワード `!CG!` は変更することができる。その方法については4章参照。

3.5 *DECK文

- (a) プログラム・ブロックは*DECK文で始まる。
- (b) *DECK文は第1カラムから書かれ、その後にDECK名を書く。
- (c) DECK名は英字で始まる8文字以内の英数字より成る。
- (d) *DECK文の後には通常のFORTRAN文、テキスト挿入命令および*DO文を任意行書くことができる。

3.6 テキスト挿入命令

テキスト挿入命令には

- (1) *CALL文, INCLUDE文, *INCLUDE文
- (2) *COMMENT文
- (3) %name文

の3種類があり、1つのCOMDECKに対して3つの処理ができる。

[1] *CALL文, INCLUDE文, *INCLUDE文

- (a) これらの挿入命令は同時に混在して使用できる。
- (b) *CALL文は第1カラムから書く。INCLUDE文および*INCLUDE文はFORTRAN77の文法に従う³⁾。したがって、INCLUDE文は第7カラム以後に、*INCLUDE文は第1カラムから書かれる。
- (c) これらの挿入命令は1行内に書かれ、継続は許されない。
- (d) EOS77はINCLUDE文、*INCLUDE文に書かれているメンバー名をCOMDECK名の中から探し、そのテキストを展開する。メンバー名がCOMDECK名の中にない場合、INCLUDE文、*INCLUDE文をそのまま展開し、warningメッセージを出力する。
- (e) これらのテキスト挿入命令には変数名変更リストを書くことができる。

[2] 変数名変更リスト

- (a) *CALL文(INCLUDE文、*INCLUDE文)および後で述べる*COMMENT文には変数名変更リストを書ける。
- (b) この命令によって、テキスト中のE宣言文の変数名・配列名を変更できる。E宣言文以

外の変数名・配列名は変更できない。

(例)

```
*INCLUDE 'A' 'AA' , 'B' 'BB'
```

(c) 変数名変更リストで変更できる変数名・配列名は最大30個までであり、継続は許されず一行以内にならなければならない。

[3] *COMENT文

- (a) *COMENTは第1カラムから書く。
- (b) *COMENTの位置に、COMDECKに含まれるテキストが注釈行として展開される。すなわち、第1カラムを 'C' に変えて展開する。(図12の②参照)
- (c) 変数名リストが使える。

<注意>

よく言われることであるが、注釈行に書いてある内容とソース・プログラムの内容は一致しないことが多い。プログラムの開発途上で、プログラマーがソース・プログラムの変更に対応して、いちいち注釈行まで変更しないのが、不一致の生じる主たる原因と考えられる。

*COMENT文は図12に示した例のように、注釈行の内容とソース・プログラムの内容との不一致をなくすための補助手段として用意されている。

[4] %name文

- (a) この挿入命令が現われたら、EOS 77 プリプロセッサはテキスト文の前後に引用符をつけて文字列を展開する。
- (b) %nameを認識する文は、注釈行以外のFORTRAN文とCG文である。
- (c) 文字列として展開されるテキストは、テキスト文の第1行のみであり、さらに、第7コラム以後である。
- (d) 引用符中、FORTRAN文のH-FORMAT中に現われる%name文は、%name文として認識されない。

(例)

```
*COMDECK TEXT1
      Y = X**2 + 1.0
*DECK SUB
-----
DATA C1/'%TEXT'/
C2 = %TEXT
```



```
DATA C1/'%TEXT'/
C2 = 'Y = X**2 + 1.0'
```

3.7 *DO文と *END DO文

- (a) *DO文はDO文，DO-WHILE文，DO-UNTILE文の端末文を自動生成する。
 (b) *DO文に対して*END DO文が必ず1対1に対応しなければならない。したがって

```
DO 100 i = 1, N
DO 100 j = 1, N
A(i, j) = 0.0
100 CONTINUE
```

に相当することを*DO文ではできない。

正しくない使い方	正しい使い方
*DO i = 1, N	*DO i = 1, N
*DO j = 1, N	*DO j = 1, N
A(i, j) = 0.0	A(i, j) = 0.0
*END DO	*END DO
	*END DO

- (c) *DO文，*END DO文は第7カラム以後に書かれる。
 (d) *DO文に¥付きサイズ式が使える。ただし，その場合，*DO文はテキスト・ブロック中になければならない（図12③参照）。

<注意>

EOS77 プリプロセッサは，テキスト挿入処理を行なった時，*DO文の処理を行なう。したがって，テキスト・ブロック中に¥付きサイズ式を伴う*DO文がある時，プログラム・ブロック中の所定の位置に*DO文が挿入され（その時，¥付きサイズ式の評価が行なわれる），その後，*DO文の処理が行なわれる。

3.8 PARAMETER文の処理

EOS77では配列宣言の添字にサイズ変数以外の変数が現われた場合，warningメッセージを出力して処理を続行する。そして，添字に使われている変数をそのままにしてプログラム・ブロック中に展開される。したがって以下の使い方が可能である。

(例)

```
*COMDECK PARAM
      PARAMETER (N=10, M=5)
*COMDECK TEXT
      REAL*8 X(N), Y(N, M)
```

ただし，プリプロセッサはPARAMETER文がテキスト・ブロック中に設定されているかどうか，PARAMETER文がFORTRAN77の文法上正しいかどうかのチェックは一切行なわない。

3.9 行コメント

(a) EOS 77 では $\nabla;\nabla$ (セミコロンの) および $\nabla:\nabla$ (コロンの) を区切り記号としてその後に注釈を書くことができ、これを行コメントと呼ぶ。

(例)

<pre>MAXA = 100 ; MAX LENGTH : OF THE VECTOR A</pre>

(b) 行コメントは、サイズ・ブロック、テキスト・ブロック、プログラム・ブロックの任意の箇所で書くことができる。

(c) 行コメントは空文の中にも書くことができる。

(d) 引用符中にある $\nabla;\nabla$ および $\nabla:\nabla$, 括弧内にある $\nabla:\nabla$ は区切り記号とみなされない。

3.10 被挿入テキストの行番号

FORTRAN 77 コンパイラーのエラー・メッセージと EOS ソース・プログラムの行番号の対応を保つため、挿入および生成される行の行番号は、プログラム・ブロックの行番号の順序を維持するようにつけられる。すなわち、

(a) 被挿入テキストの行番号は、挿入命令の行番号から始まる公差 1 の等差数列になる。

(b) *DO 文および *END DO 文で生成される DO 文および DO 文末文の行番号には、それぞれ、対応する *DO 文および *END DO 文の行番号が使われる。

4. EOS 77 の使い方

4.1 入出力機番

EOS 77 プリプロセッサが使用する入出力機番を以下に示す。

機番	入力/出力	内 容
5	入力 *)	サイズ変数の入力
6	出力	プリプロセッサの処理情報 **)
9	入力	EOS ソース・プログラム ***)
10	出力	FORTRAN ソース・プログラム ***)
11	入力 +)	オプション・パラメータ ***)

<注意>

*) サイズ変数をカード入力で行なう場合、オプションパラメータの設定において CA-RDSW=1 と必ずしなければならない。

3.9 行コメント

(a) EOS 77 では $\nabla;\nabla$ (セミコロン) および $\nabla:\nabla$ (コロン) を区切り記号としてその後に注釈を書くことができ、これを行コメントと呼ぶ。

(例)

<pre>MAXA = 100 ; MAX LENGTH : OF THE VECTOR A</pre>

(b) 行コメントは、サイズ・ブロック、テキスト・ブロック、プログラム・ブロックの任意の箇所で書くことができる。

(c) 行コメントは空文の中にも書くことができる。

(d) 引用符中にある $\nabla;\nabla$ および $\nabla:\nabla$, 括弧内にある $\nabla:\nabla$ は区切り記号とみなされない。

3.10 被挿入テキストの行番号

FORTRAN 77 コンパイラーのエラー・メッセージと EOS ソース・プログラムの行番号の対応を保つため、挿入および生成される行の行番号は、プログラム・ブロックの行番号の順序を維持するようにつけられる。すなわち、

(a) 被挿入テキストの行番号は、挿入命令の行番号から始まる公差 1 の等差数列になる。

(b) *DO 文および *END DO 文で生成される DO 文および DO 文末文の行番号には、それぞれ、対応する *DO 文および *END DO 文の行番号が使われる。

4. EOS 77 の使い方

4.1 入出力機番

EOS 77 プリプロセッサが使用する入出力機番を以下に示す。

機番	入力/出力	内容
5	入力 *)	サイズ変数の入力
6	出力	プリプロセッサの処理情報 **)
9	入力	EOS ソース・プログラム ***)
10	出力	FORTRAN ソース・プログラム ***)
11	入力 +)	オプション・パラメータ ***)

<注意>

*) サイズ変数をカード入力で行なう場合、オプションパラメータの設定において CA-RDSW=1 と必ずしなければならない。

**） 処理情報には

EOS ソースリスト
 エラーメッセージ
 オプション・パラメーター・テーブルのダンプ
 サイズ変数テーブルのダンプ

が含まれる（図9参照）。

***） ファイルは1レコード80バイトのPSファイル。

+) オプション・パラメータを外部ファイルから入力する場合に使う。実行時パラメータの文字列として入力する場合にはALLOCATEしてはならない（外部ファイルからの入力が優先されるため。4.3参照）。

4.2 オプション・パラメータ

プリプロセスの実行値のオプションとして、次の各パラメータの値を指定できる。

パラメータ	デフォルト値	意味
OPKEYW	CG	キーワードの定義（10文字以内）
CMSUP	1	注釈行の消去
KEYSW	1	キーワードの使用
SLIST	0	EOSソース・プログラムのリスト出力
NAMSW	1	%name文の使用
DOSW	1	*DO文の使用
CARDSW	0	サイズ定義の入力カード使用

OPKEYW : キーワード 'CG' を変更する時に使う。

CMSUP : CMSUP=1の時、注釈行がすべて消去されたFORTRANソース・プログラムが出力される（*COMMENT文、CG文で挿入された行は消去されない）。
 CMSUP=0の時、注釈行はそのまま出力される。

KEYSW : KEYSW=1の時、キーワード（デフォルト値 'CG'）についてEOS77処理が有効になる。
 KEYSW=0の時、キーワードは無効になる。

SLIST : SLIST=1の時、EOSソース・プログラムが出力される。
 SLIST=0の時、EOSソース・プログラムは出力されない。

NAMSW : NAMSW=0の時、%name文はプリプロセスの処理対象にならない。*)

DOSW : DOSW=0の時、*DO文はプリプロセスの処理対象にならない。*)

CARDSW : サイズ定義をカード入力で行なう時、CARDSW=1とする（機番は5である）。

<注意>

*) EOSソース・プログラムで%name文や*DO文がない場合に使用すると、プリプロセスの処理が早くなる。

4.3 オプション・パラメータの入力方法

オプション・パラメータの入力方法として、実行時パラメータの文字列として入力する方法（図8参照）と、外部ファイル（機番11）から入力する方法2種類があり、外部ファイル入力が優先される。

[1] 実行時パラメータを使う方法

バッチ・ジョブの場合（付録2参照）

```

//EOS77 EXEC PGM=EOS77
//          PARM= 'CARDSW=1, DOSW=0' ←
//STEPLIB DO DSN=J0566.EOS77V2.LOAD, DISP=SHR

```

TSSの場合

```
CALL 'J0566.EOS77V2.LOAD(EOS77)' 'CARDSW=1, DOSW=0'
```

[2] 外部ファイルから入力する場合

外部ファイルは1レコード80バイトのPSファイルで各1レコードに1つのパラメータだけが指定できる。この時、文字列にスペースがあってもよい。

(例)

CARDSW	=	1
DOSW	=	0
NAMSW	=	0

5. ま と め

TRITON計画の一環として構造化プログラミング支援シールEOS77の開発を行なった。このプリプロセッサはFORTRAN77に準拠したEOSの上位バージョンであるとともに構造化プログラミングを支援するいくつかの拡張機能をそなえている。

FORTRAN77に準拠した逆EOSプリプロセッサも開発されており、その使用手引を近く公表する予定である。

EOS77の利用は構造化プログラミングに役立つと期待されるが、まだ、いくつかの改良点が考えられる。特にサイズ式に書ける演算を論理式まで含めて拡張することが望まれる。また、プログラムが十分に構造化されている場合、COMMON群にその構造を反映した、一種のデータ構造があるはずで、それを定義してコード開発の効率を上げることができよう。

4.3 オプション・パラメータの入力方法

オプション・パラメータの入力方法として、実行時パラメータの文字列として入力する方法（図8参照）と、外部ファイル（機番11）から入力する方法2種類があり、外部ファイル入力が優先される。

(1) 実行時パラメータを使う方法

バッチ・ジョブの場合（付録2参照）

```

//EOS77 EXEC PGM=EOS77
//          PARM= 'CARDSW=1, DOSW=0' ←
//STEPLIB DO DSN=J0566.EOS77V2.LOAD, DISP=SHR

```

TSSの場合

```
CALL 'J0566.EOS77V2.LOAD(EOS77)' 'CARDSW=1, DOSW=0'
```

(2) 外部ファイルから入力する場合

外部ファイルは1レコード80バイトのPSファイルで各1レコードに1つのパラメータだけが指定できる。この時、文字列にスペースがあってもよい。

(例)

CARDSW	=	1
DOSW	=	0
NAMSW	=	0

5. ま と め

TRITON計画の一環として構造化プログラミング支援シールEOS77の開発を行なった。このプリプロセッサはFORTRAN77に準拠したEOSの上位バージョンであるとともに構造化プログラミングを支援するいくつかの拡張機能をそなえている。

FORTRAN77に準拠した逆EOSプリプロセッサも開発されており、その使用手引を近く公表する予定である。

EOS77の利用は構造化プログラミングに役立つと期待されるが、まだ、いくつかの改良点が考えられる。特にサイズ式に書ける演算を論理式まで含めて拡張することが望まれる。また、プログラムが十分に構造化されている場合、COMMON群にその構造を反映した、一種のデータ構造があるはずで、それを定義してコード開発の効率を上げることができよう。

謝 辞

EOS 77 システムの開発に際しての松下樹氏 (数理技研) の御協力に感謝します。また, TRITON 計画全体を通じての, 核融合研究部田中正俊部長の御激励に深く感謝します。

文 献

- 1) B.W.Kernighan and P.J.Plauger; "The Elements of Programing Style", McGraw-Hill, New York, (1974). 日本語版, 木村泉訳: "プログラム書法", 共立出版, 東京 (1976)
- 2) 竹田辰興, 常松俊秀, 栗田源一: "可変配列サイズプログラムのためのプリプロセッサ・システム EOS", JAERI-M 82-097 (1982).
- 3) FACOM OS N/F4 MSP FORTRA77 使用手引書 V10 用, P46.

謝 辞

EOS 77 システムの開発に際しての松下樹氏 (数理技研) の御協力に感謝します。また, TRITON 計画全体を通じての, 核融合研究部田中正俊部長の御激励に深く感謝します。

文 献

- 1) B.W.Kernighan and P.J.Plauger; "The Elements of Programming Style", McGraw-Hill, New York, (1974). 日本語版, 木村泉訳: "プログラム書法", 共立出版, 東京 (1976)
- 2) 竹田辰興, 常松俊秀, 栗田源一: "可変配列サイズプログラムのためのプリプロセッサ・システム EOS", JAERI-M 82-097 (1982).
- 3) FACOM OS M/F4 MSP FORTRA77 使用手引書 V10 用, P46.

	***** MEMBER NAME *****	***** PAGE NO. *****	***** NO. OF CARDS *****
(NO.=001)	#COM1	0001	49
(NO.=002)	AAMAIN	0001	13
(NO.=003)	DAIKEI	0002	19
(NO.=004)	DOCUM	0002	102
(NO.=005)	FUNCCS	0004	12
(NO.=006)	INPUT	0005	13
(NO.=007)	OUTPUT	0005	29
(NO.=008)	PRESET	0005	19
(NO.=009)	VALFUC	0006	16

			272 CARDS

☒ 1 PO ファイル J3331. EOS77EX. FORT77 のメンバー名リスト。

```

① {
      NTH = 50
      NT  = 2*NTH+1 ; MAXMUM NUMBER OF POINTS FOR FUNCTION
      MAXCS = 5 ; MAXMUM NUMBER OF SIN/COS TERMS
C
② *COMDECK BLNK1
      COMMON
③ {
      ¥ THETA, FUNC, DT, AINTGF, NT
      DIMENSION
      ¥ THETA(NT), FUNC(NT)
C
      *COMDECK CMCNTL
      COMMON /CMCNTL/ LDOCUM, LOUT, NTLIM, NCSLIM
      LOGICAL LDOCUM, LOUT
C
      *COMDECK CMFUNC
      COMMON /FUNCFM/ AC, AS, MAXC, MAXS
      DIMENSION AC(0:MAXCS), AS(MAXCS)
C
      *COMDECK CMFMT
      COMMON /CMFMT/ CFMT(20)
      CHARACTER CFMT*60
C
      *COMDECK NAMELT
      NAMELIST/ NEWDT/
      ¥ NT, LDOCUM, LOUT, MAXC, MAXS,
      ¥ AC, AS
C
④ {
      *COMDECK LOCSET
      NTLIM = ¥(NT)
      NCSLIM = ¥(MAXCS)
      NT = 0
      MAXC = ¥(MAXCS)
      MAXS = ¥(MAXCS)
      AC(0) = 1.DO
C
⑤ {
      *COMDECK CFORM1
      FUNC(T) = SIN(FC(T))
      *COMDECK CFORM2
      FC(T) = AC(0) + SUM(N=1,MAXC)AC(N)*COS(N*T)
C
⑥ {
      *COMDECK LOCFUN
      ZFC = AC(0)
      IF(MAXC .GE. 1) THEN
        *DO NC = 1,MAXC
          ZFC = ZFC + AC(NC)*COS(NC*T)
        *END DO
      ENDIF
C
      ZFT = COS(ZFC)
      ⑦

```

図2 メンバー名#COM1の内容。

(a)

```

*DECK INPUT                                00010000
SUBROUTINE INPUT                            00020000
  IMPLICIT REAL*8 (A-H,O-Z)                 00030000
*CALL BLNK1                                 00040000
*CALL CMCNTL                                00050000
*CALL CMFUNC                                 00060000
*CALL CMFMT                                  00070000
*CALL NAMELT                                 00080000
C                                             00090000
  READ(5, NEWDT)                             00120002
  WRITE(6,NEWDT)                             00130002
  RETURN                                     00370000
  END                                         00380000

```

(b)

```

SUBROUTINE INPUT                            00020000
IMPLICIT REAL*8 (A-H,O-Z)                 00030000
COMMON                                      00040001
¥ THETA, FUNC, DT, AINTGF, NT             00040002
DIMENSION                                  00040003
¥ THETA(101), FUNC(101)                   00040004
COMMON /CMCNTL/ LDOCUM, LOUT, NTLIM, NCSLIM 00050001
LOGICAL LDOCUM, LOUT                      00050002
COMMON /FUNCFM/ AC, AS, MAXC, MAXS        00060001
DIMENSION AC(0:5), AS(5)                  00060002
COMMON /CMFMT/ CFMT(20)                   00070001
CHARACTER CFMT*60                         00070002
NAMELIST/ NEWDT/                           00080001
¥ NT, LDOCUM, LOUT, MAXC, MAXS,           00080002
¥ AC, AS                                   00080003
READ(5, NEWDT)                             00120002
WRITE(6,NEWDT)                             00130002
RETURN                                     00370000
END                                         00380000

```

図3 メンバー名 INPUT に格納されているプログラム・ブロック (図 3a) および EOS 77 で前処理を行って作成された FORTRAN プログラム (図 3b)。

(a)

```

*DECK PRESET
SUBROUTINE PRESET
  IMPLICIT REAL*8 (A-H,O-Z)
*CALL BLNK1
*CALL CMCNTL
*CALL CMFUNC
C -----
*INCLUDE LOCSET
C
  *DO J=1,NCSLIM
  AC(J) = 0.DO
  AS(J) = 0.DO
  *END DO
C
  LDOCUM = .FALSE.
  LOUT   = .FALSE.
C
  RETURN
  END
    
```

00010000
00030000
00040000
00050000
00060000
00070000
00080000
00090000
00100000
00110001
00120000
00130000
00140000
00150000
00151001
00152001
00153001
00160000
00170000

(b)

```

SUBROUTINE PRESET
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON
  ¥ THETA, FUNC, DT, AINTGF, NT
  DIMENSION
  ¥ THETA(101), FUNC(101)
  COMMON /CMCNTL/ LDOCUM, LOUT, NTLIM, NCSLIM
  LOGICAL LDOCUM, LOUT
  COMMON /FUNCFM/ AC, AS, MAXC, MAXS
  DIMENSION AC(0:5), AS(5)
  NTLIM = 101
  NCSLIM = 5
  NT = 0
  MAXC = 5
  MAXS = 5
  AC(0) = 1.DO
  DO 99997 J=1,NCSLIM
  AC(J) = 0.DO
  AS(J) = 0.DO
99997 CONTINUE
  LDOCUM = .FALSE.
  LOUT = .FALSE.
  RETURN
  END
    
```

00030000
00040000
00050001
00050002
00050003
00050004
00060001
00060002
00070001
00070002
00090001
00090002
00090003
00090004
00090005
00090006
00110001
00120000
00130000
00140000
00151001
00152001
00160000
00170000

図4 罫付きサイズ式の使用例。図2の④は図4aのテキスト挿入命令 *INCLUDEで図4bのように展開される。

(a)

```

*DECK DOCUM                                00010000
  SUBROUTINE DOCUM                          00020002
  IMPLICIT REAL*8 (A-H,O-Z)                00030000
*CALL BLNK1                                00040000
*CALL CMCNTL                                00050000
*CALL CMFUNC                                00060000
*CALL CMFMT                                  00070000

  ---  ---  ---  ---  ---  ---
  ---  ---  ---  ---  ---  ---

C                                            00270000
  CFMT(2) =                                00280000
  ¥ 'DEFINITE INTEGRAL OF FUNCTION FUNC, WHERE' 00290000
C                                            00300000
  CFMT(3) = %CFORM1                          00310000
  CFMT(4) = %CFORM2                          00320000
C                                            00330000

```

(b)

```

SUBROUTINE DOCUM                            00020002
IMPLICIT REAL*8 (A-H,O-Z)                  00030000
COMMON                                      00040001
  ¥ THETA, FUNC, DT, AINTGF, NT            00040002
DIMENSION                                   00040003
  ¥ THETA(101), FUNC(101)                  00040004
COMMON /CMCNTL/ LDOCUM, LOUT, NTLIM, NCSLIM 00050001
LOGICAL LDOCUM, LOUT                       00050002
COMMON /FUNCFM/ AC, AS, MAXC, MAXS          00060001
DIMENSION AC(0:5), AS(5)                   00060002
COMMON /CMFMT/ CFMT(20)                    00070001

  ---  ---  ---  ---  ---  ---
  ---  ---  ---  ---  ---  ---

CFMT(2) =                                00280000
¥ 'DEFINITE INTEGRAL OF FUNCTION FUNC, WHERE' 00290000
CFMT(3) = 'FUNC(T) = SIN(FC(T))'           00310000
CFMT(4) = 'FC(T) = AC(0) + SUM(N=1,MAXC)AC(N)*COS(N*T)' 00320000

```

図5 % name 文の使用例。図2の⑤は図5aに示すテキスト挿入命令 % name により、図5bに示すような引用符のついた文字列として展開される。

(a)

```

*DECK FUNCCS                                00010000
REAL FUNCTION FT*8 (T)                       00020000
IMPLICIT REAL*8 (A-H,O-Z)                   00030000
*CALL CMFUNC                                  00040000
ZFT = 0.DO                                    00050001
C                                              00060001
*CALL LOCFUN                                  00070001
C                                              00080001
FT = ZFT                                      00090001
C                                              00100001
RETURN                                        00110001
END                                            00120001

```

(b)

```

REAL FUNCTION FT*8 (T)                       00020000
IMPLICIT REAL*8 (A-H,O-Z)                   00030000
COMMON /FUNCFM/ AC, AS, MAXC, MAXS          00040001
DIMENSION AC(0:5), AS(5)                    00040002
ZFT = 0.DO                                    00050001
ZFC = AC(0)                                  00070002
IF(MAXC .GE. 1) THEN                          00070003
DO 99998 NC = 1,MAXC                          00070004
ZFC = ZFC + AC(NC)*COS(NC*T)                  00070005
99998 CONTINUE                                00070006
ENDIF                                          00070007
ZFT = COS(ZFC)                                00070008
FT = ZFT                                      00090001
RETURN                                        00110001
END                                            00120001

```

図6 EOS77の機能を用いた関数副プログラム(図6a)および前処理後のFORTRANプログラム(図6b)。

(a)

<pre> *DECK DAIKEI SUBROUTINE DAIKEI IMPLICIT REAL*8 (A-H,O-Z) *CALL BLNK1 'FUNC'F' C IF(NT .EQ. 2) THEN AINTGF = (F(1) + F(NT))*DT/2.DO ELSE ZSUM = (F(1)+F(NT))/2.DO C *DO I=2,NT-1 ZSUM = ZSUM + F(I) *END DO C AINTGF = ZSUM*DT ENDIF C RETURN END </pre>	<p>①</p> <p>←</p> <p>←</p> <p>②</p> <p>□</p> <p>③</p>	<pre> 00010000 00030000 00040000 00050000 00060000 00070000 00080000 00090000 00100000 00110000 00120000 00130000 00140000 00150000 00160000 00170000 00180000 00190000 00200000 </pre>
---	---	---

(b)

<pre> SUBROUTINE DAIKEI IMPLICIT REAL*8 (A-H,O-Z) COMMON ¥ THETA,F, DT, AINTGF, NT DIMENSION ¥ THETA(101), F(101) IF(NT .EQ. 2) THEN AINTGF = (F(1) + F(NT))*DT/2.DO ELSE ZSUM = (F(1)+F(NT))/2.DO DO 99999 I=2,NT-1 ZSUM = ZSUM + F(I) 99999 CONTINUE AINTGF = ZSUM*DT ENDIF RETURN END </pre>	<p>←</p> <p>②</p>	<pre> 00030000 00040000 00050001 00050002 00050003 00050004 00070000 00080000 00090000 00100000 00120000 00130000 00140000 00160000 00170000 00190000 00200000 </pre>
---	-------------------	---

図7 メンバー名 DAIKEI に格納されているプログラム・ブロック (図7a) および前処理後の FORTRAN プログラム (図7b)。

```

list tssmac.clist(etof77)
  J3331.TSSMAC.CLIST(ETOF77)
00010 PROC 1 ELM SF(EOS77EX)
00011 FREE AT(ZZZ)
00012 DELETE F77.FORT77
00020 ATTR ZZZ LRECL(80) BL(16000) REC(F B)
00030 ALLOC DA(&SF..FORT77(#COM1) &SF..FORT77(&ELM)) F(FT09F001) SHR
00040 ALLOC DA(F77.FORT77) F(FT10F001) NEW SP(20 5) T UNIT(TSSWK) USING(ZZZ)
00050 ALLOC DA(*) F(SYSIN) REU
00060 WRITE ===== EOS ----->FORT77=====
00070 CALL 'J2636.EOS77V2.LOAD(EOS77)' 'KEYSW=1,NAMSW=1,DOSW=1'
00140 FREE F(FT09F001 FT10F001)
00150 FREE AT(ZZZ)
00170 WRITE ===== FORTRAN 77 =====
00180 FORT77 F77
  END OF DATA
READY

```

- 図 8 private catalogued procedure ETOF77 (図 8a) および ETOF77CL (図 8b)。ETOF77 は PO ファイルの 1 メンバーに格納されるプログラム・ブロックを FORTRAN プログラムに変換した後コンパイルを行なう。ETOF77CL は PO ファイルに格納されている EOS ソース・プログラムから実行可能なロード・モジュールを作る。

```

list tssmac.clis(etof77cl)
  J3331.TSSMAC.CLIS(ETOF77CL)
00010 PROC 1 SF OPT(3) ADD(FLAG(I)) LM(Z.LOAD(TEMPNAME))
00020 FREE AT(ZZZ)
00030 DELETE F77.FORT77
00040 ATTR ZZZ LRECL(80) BL(16000) REC(F B)
00050 WRITE ===== PO -----> PS =====
00060 ALLOC DA(&SF..FORT77) F(PO) SHR
00070 ALLOC DA(@PS) F(PS) NEW SP(20 5) T UNIT(WK10) USING(ZZZ)
00080 ALLOC DA(*) F(SYSIN) REU
00090 CALL 'J2636.TOOL#MA.LOAD(POTOPS)'
00100 FREE F(PO PS)
00110 WRITE ===== EOS ----->FORT77=====
00120 ALLOC DA(@PS) F(FT09F001) OLD REU
00130 ALLOC DA(F77.FORT77) F(FT10F001) NEW SP(10 5) T UNIT(WK10) USING(ZZZ)
00140 CALL 'J2636.EOS77V2.LOAD(EOS77)' 'KEYSW=0,NAMSW=1,DOSW=1'
00150 FREE F(FT09F001 FT10F001)
00160 DELETE @PS
00170 FREE AT(ZZZ)
00180 WRITE =====COMPILE AND LINK =====
00190 FREE AT(DCB)
00200 DELETE @@.OBJ
00210 ATTR DCB LRECL(80) BL(3200) REC(F B) DSORG(PS)

00220 ALLOC DA(@@.OBJ) NEW SP(30 10) T RELEASE USING(DCB)
00230 CONTROL LIST PROMPT MSG
00240 FORT77 F77 OPT(&OPT) &ADD OBJ(@@.OBJ)
00250 SET &LC = &LASTCC
00260 WRITE <<< COMPLICATION CODE = &LC >>>
00270 IF &LC GT 4 THEN EXIT
00280 LINK @@.OBJ LIB('SYS2.FORTLIB' 'SYS9.SSL2.LOAD' 'J0566.OLYMPIA.LOAD') +
00281 LOAD(&LM)
00290 WRITE <<< LINKAGE CODE = &LASTCC >>>
00300 EXIT
  END OF DATA
READY

```

☒ 8 b

```
.etof77cl eos77ex
FILE ZZZ NOT FREED, IS NOT ALLOCATED
ENTRY (A) J3331.F77.FORT77 DELETED
===== PO -----> PS =====
===== EOS ----->FORT77=====
```

===== E O S 7 7 OPTION PARAMETER TABLE =====

PREP	#	
SIZEP	\$	
NAMEP	%	
DOPREP	#	
SEPR1	:	
SEPR2	:	
OPKEYW	CG	
LINECM		1
CMSUP		1
KEYSW		0
SLIST		0
NAWSW		1
CARDSW		0
DOSW		1

PASS-1 -- DECK & COMDECK

PASS-2 -- #DO & #ENDDO

===== E O S 7 7 SIZE TABLE DUMP =====

NAME	VAL
NTH	50
MAXCS	5
NT	101

===== E O S 7 7 TOTAL ACCOUNT TABLE =====

SOURCE	=	272
ERROR	=	0
WARNING	=	0

FTMAIN CPU TIME = 367 (MS)

ENTRY (A) J3331.@PS DELETED
=====COMPILE AND LINK =====

FILE DCB NOT FREED, IS NOT ALLOCATED
ENTRY (A) J3331.@@.OBJ DELETED
FORT77 F77 OPT(3) FLAG(1) OBJ(@@.OBJ)
FORTRAN 77 COMPILER ENTERED
END OF COMPILATION

<<< COMPLICATION CODE = 0 >>>
LINK @@.OBJ LIB('SYS2.FORTLIB' 'SYS9.SSL2.LOAD' 'J0566.OLYMPIA.LOAD') LOAD(Z.LO
AD(TEMPNAME))
** MEMBER NAME ** TEMPNAME NOW REPLACED IN LIBRARY.
<<< LINKAGE CODE = 0 >>>
READY

☒ 9 private catalogued procedure ETOF77CL の実行例


```

*DECK DOCUM                                00010000
SUBROUTINE DOCUM                            00020002
IMPLICIT REAL*8 (A-H,O-Z)                 00030000
*CALL BLNK1                                 00040000
*CALL CMCNTL                                00050000
*CALL CMFUNC                                 00060000
*CALL CMFMT                                  00070000
*CALL NAMELT                                 00080000
C                                             00090000
CFMT(1) = '(1H1) '                          00100000
WRITE(6,FMT=CFMT(1))                       00110000
C                                             00120000
ICON = 0                                     00130000
IF(NT .LE. 1 .OR. NT .GT. NTLIM)           00140000
IF(MAXC .LT. 0 .OR. MAXC .GT. NCSLIM)      00150000
IF(MAXS .LT. 0 .OR. MAXS .GT. NCSLIM)      00160000
C                                             00170000
IF(ICON .NE. 0) THEN                        00180000
CFMT(1)='(1H0,5X,A,5X,I4,A) '             00190000
CFMT(2)='(1H ,5X,A) '                      00200000
WRITE(6,FMT=CFMT(1))                       00210000
¥ '***** INVALID INPUT DATA ; ICON =',ICON, ' *****' 00220000
WRITE(6,FMT=CFMT(2))                       00230000
¥ '***** SEE THE FOLLOWIG TABLE *****' 00240000
LDOCUM = .TRUE.                             00250000
ENDIF                                        00260000
C                                             00270000
CFMT(2) =                                    00280000
¥ 'DEFINITE INTEGRAL OF FUNCTION FUNC, WHERE' 00290000
C                                             00300000
CFMT(3) = %CFORM1                           00310000
CFMT(4) = %CFORM2                           00320000
C                                             00330000
CFMT(11) = '(1H0,////) '                   00340000
CFMT(12) = '(1H ,5X,A) '                   00350000
CFMT(13) = '(1H ,2(5X,A,I3)) '             00360000
CFMT(14) = '(1H ,5X,A/5X,(1P10D12.4)) '    00370000
C                                             00380000
WRITE(6,FMT=CFMT(11))                      00390000
WRITE(6,FMT=CFMT(12)) CFMT(2)              00400000
WRITE(6,FMT=CFMT(12)) CFMT(3)              00410000
WRITE(6,FMT=CFMT(12)) CFMT(4)              00420000
C                                             00430000
IF(ICON .EQ. 0) THEN                        00440000
C                                             00450000
CFMT(10) = '(1H ,5X,A,I4) '                00460000
WRITE(6,FMT=CFMT(10)) 'NUMBER OF POINTS (NT) =', NT 00470000
WRITE(6,FMT=CFMT(13)) 'MAXC =', MAXC, 'MAXS =', MAXS 00480000
WRITE(6,FMT=CFMT(14)) 'AC =', (AC(J), J=0, MAXC) 00490000
WRITE(6,FMT=CFMT(14)) 'AS =', (AS(J), J=1, MAXS) 00500004
C                                             00510000
ENDIF                                        00520000
C                                             00530000
IF(.NOT. LDOCUM) RETURN                     00540001
C *****                                  00550001
CALL PRESET                                 00560001
C *****                                  00570005
CFMT(1)='NT : NUMBER OF POINTS ; 2 <= NT <= ' 00580001
CFMT(2)='DEFAULT VALUE OF NT ='            00581001
CFMT(11)='(1H0,5X,A,I4) '                  00581101
CFMT(12)='(1H0,5X,A/5X,(1P10D12.4)) '     00581201
CFMT(13)='(1H0,5X,A) '                     00582001
WRITE(6,FMT=CFMT(11)) CFMT(1), NTLIM

```

図 10 EOS 77 の機能を利用したプログラム・ドキュメント作成副プログラム
 DOCUM のリスト。

	WRITE(6,FMT=CFMT(11)) CFMT(2), NT	00583001
C	CFMT(3)=	00590001
	¥'MAXC : MAXMUM NUMBER OF COS TERMS ; 0 <= MAXC <= '	00600001
	CFMT(4)='DEFAULT VALUE OF MAXC ='	00610001
	CFMT(5)=	00620001
	¥'AC(0), , AC(MAXC) : COEFFICIENTS OF COS TERMS'	00630001
	CFMT(6)=	00640001
	¥'DEFAULT VALUES OF AC ='	00650001
	WRITE(6,FMT=CFMT(11)) CFMT(3), NCSLIM	00660003
	WRITE(6,FMT=CFMT(11)) CFMT(4), MAXC	00661001
	WRITE(6,FMT=CFMT(13)) CFMT(5)	00662005
	WRITE(6,FMT=CFMT(12)) CFMT(6),(AC(J), J=0,MAXC)	00663001
		00664005
		00670001
	CFMT(7)=	00680001
	¥'MAXS : MAXMUM NUMBER OF SIN TERMS ; 0 <= MAXS <= '	00690001
	CFMT(8)='DEFAULT VALUE OF MAXS ='	00700001
	CFMT(9)=	00710001
	¥'AS(1), , AS(MAXS) : COEFFICIENTS OF SIN TERMS'	00720001
	CFMT(10)=	00730001
	¥'DEFAULT VALUES OF AS ='	00740003
	WRITE(6,FMT=CFMT(11)) CFMT(7), NCSLIM	00741001
	WRITE(6,FMT=CFMT(11)) CFMT(8), MAXS	00742001
	WRITE(6,FMT=CFMT(13)) CFMT(9)	00743001
	WRITE(6,FMT=CFMT(12)) CFMT(10),(AS(J), J=1,MAXS)	00744005
		00745001
	CFMT(1)='LOUT : OUTPUTING PARAMETER'	00746001
	CFMT(2)='IF LOUT = TRUE THEN'	00747001
	CFMT(3)='OUTPUTS THE *NT* VALUES OF FUNCTION.'	00748004
	CFMT(4)='DEFAULT VALUE OF LOUT ='	00749001
	CFMT(14)='(1H0,5X,A,L4)'	00749101
	WRITE(6,FMT=CFMT(13)) CFMT(1)	00749201
	WRITE(6,FMT=CFMT(13)) CFMT(2)	00749301
	WRITE(6,FMT=CFMT(13)) CFMT(3)	00749401
	WRITE(6,FMT=CFMT(14)) CFMT(4) , LOUT	00749504
		00749601
	CFMT(5)='THIS JOB STOPS BECAUSE LDOCUM = TRUE.'	00749701
	WRITE(6,FMT=CFMT(13)) CFMT(5)	00749801
	STOP	00749901
	END	00750001

図10 (つづき)

```
call z
TEMPNAME ASSUMED AS MEMBERNAME
00120002 &NEWDT ?
&newdt &end
&NEWDT
NT=0,LDOCUM=F,LOUT=F,MAXC=5,MAXS=5,AC=1.0000000000000000,0.0,0.0,0.0,0.0,0.0,A
S=0.0,0.0,0.0,0.0,0.0,
&END
```

```
***** INVALID INPUT DATA ; ICON =      100 *****
***** SEE THE FOLLOWIG TABLE *****
```

```
DEFINITE INTEGRAL OF FUNCTION FUNC, WHERE
FUNC(T) = SIN(FC(T))
FC(T)   = AC(0) + SUM(N=1,MAXC) AC(N)*COS(N*T)
NT      : NUMBER OF POINTS ;    2 <= NT <=          101
DEFAULT VALUE OF NT =                                0
MAXC    : MAXMUM NUMBER OF COS TERMS ; 0 <= MAXC <=    5
DEFAULT VALUE OF MAXC =                              5
AC(0), . . . . , AC(MAXC) : COEFFICIENTS OF COS TERMS
```

```
-DEFAULT VALUES OF AC =
 1.0000D+00  0.0      0.0      0.0      0.0      0.0
MAXS : MAXMUM NUMBER OF SIN TERMS ; 0 <= MAXS <=    5
DEFAULT VALUE OF MAXS =                              5
AS(1), . . . . , AS(MAXS) : COEFFICIENTS OF SIN TERMS
DEFAULT VALUES OF AS =
 0.0      0.0      0.0      0.0      0.0
LOUT : OUTPUTING PARAMETER
IF LOUT = TRUE THEN
OUTPUTS THE *NT* VALUES OF FUNCTION.
DEFAULT VALUE OF LOUT =                               F
THIS JOB STOPS BECAUSE LDOCUM = TRUE.
```

READY

図 11 プログラム・ドキュメントの出力内容。

(a)

```

C
      NTH   = 50
      NT    = 2*NTH+1      ; MAXMUM NUMBER OF POINTS FOR FUNCTION
      MAXCS = 5           ; MAXMUM NUMBER OF SIN/COS TERMS
C
*COMDECK CGEX
CG   COMMON
CG   ¥     THETA(¥(NT)),  FUNC(¥(NT))
      DIMENSION A(MAXCS), AA(NT)
*COMDECK COMENT
      MAX LENGTH OF VECTOR A (MAXA) = ¥(MAXCS)
*COMDECK ASET
      *DO J=1,¥(MAXCS)    ← ③
      A(J) = 0.DO
      *END DO
      AA(¥(MAXCS)) = A(¥(MAXCS))
C

```

```

*DECK ACOMENT
      SUBROUTINE SUBA
C
*INCLUDE CGEX
*COMENT COMENT
C
*CALL ASET
C
      A(1) = 1.DO
C
      RETURN
      END

```

(b)

```

      SUBROUTINE SUBA
CG   COMMON
CG   ¥     THETA(101),  FUNC(101)
      DIMENSION A(5), AA(101)
C     MAX LENGTH OF VECTOR A (MAXA) = 5
      DO 99999 J=1,5
      A(J) = 0.DO
99999 CONTINUE
      AA(5) = A(5)
      A(1) = 1.DO
      RETURN
      END

```

図 12 CG文, *COMENT文, ¥付きサイズ式を伴う *DO文の使用例。

付録1 EOS77 プリプロセッサのエラー・メッセージ

プリプロセス時のエラーのレベルにはEOS 77 文法に関するエラー (ERROR レベル)とユーザーに注意を促すWARNING レベルのエラーおよびエラー発生時点で処理をただちに停止するSYSTEM ERROR レベルの3種類がある (ERROR レベル, WARNING レベルのエラーが発生してもプリプロセス処理は続行される)。

ERROR レベルおよびWARNING レベルのエラー・メッセージは, エラー発生時が挿入処理時 (PASS-1) か *DO 文の処理時 (PASS-2) かによって以下に示す仕様に従って出力される。

PASS-1 に関するエラー・メッセージ仕様

```
*** erlevel eno :: ルーチン名 lno
システム・ルーチン名 --- メッセージ
```

PASS-2 に関するエラー・メッセージ仕様

```
*** erlevel eno :: ルーチン名
システム・ルーチン名 --- メッセージ
```

erlevel : ERROR あるいは WARNING
 eno : エラー・コード
 ルーチン名 : エラーの発生したユーザー・ルーチン名。
 lno : エラーの発生した行番号。

システム・ルーチン名: エラーを出力したEOS 77 システム・プログラムのルーチン名。

SYSTEM ERROR レベルのエラーが発生すると, EOS 77 プリプロセッサはエラーメッセージ

```
*** STOP --- メッセージ
```

を出力して停止する。このシステム・エラーは

(1) 入出力のハード・エラー

あるいは

(2) プリプロセス処理に必要な制御テーブルのオーバー・フロー

によって起こる。(2)が発生するのは主としてEOS ソース・プログラムが大きすぎる場合であり, 制御テーブルの大きくして, EOS 77 のロード・モジュールを作り直す必要がある。したがって制御テーブルのオーバー・フローに関係したシステム・エラーが発生した場合は本報告書の著者に連絡して下さい。

EOS 77 プリプロセッサ・エラー・メッセージ (ERROR level) 一覧表

6	メッセージ: ONLY SIZE BLOCK
	サイズ・ブロックしかない。
7	メッセージ: SIZE SYMBOL CANNOT DEFINE ALL
	サイズ・ブロックの処理がすべて終わっても, サイズ変数がすべて定義できない。(未定義サイズ変数がある)
8	メッセージ: SIZE DEFFINITION ERROR
	サイズ変数の定義の文法エラー
21	メッセージ: UNDEFINED SIZE SYMBOL (XXXX)
	サイズ変数XXXXが未定義である。
22	メッセージ: UNDEFINED COMDECK NAME (XXXX)
	COMDECK名XXXXが未定義である。
23	メッセージ: ILLIGAL BLOCK NAME (XXXX)
	COMDECK名又はDECK名のエラー
24	メッセージ: ILLIGAL SYMBOL (XXXX)
	変数名のエラー
25	メッセージ: DUPLICATE COMDECK NAME (XXXX)
	同一COMDECK名がすでに登録されている。
26	メッセージ: RENAME LIST SYNTAX ERROR
	変数名変更リストに文法エラーがある。
27	メッセージ: DUPLICATE NAME IN RENAME LIST
	変数名変更リストの中に, 同一の旧変数名が現われた。
28	メッセージ: TOO MANY RENAME LIST
	変数名変更リストが30組以上ある。
30	メッセージ: ILLIGAL OPTION
	未定義のオプション・パラメータが入力された。
31	メッセージ: MORE THAN 20 LINES
	継続行が20行を越えた。
32	メッセージ: INVALID EXPRESSION
	式の文法エラー, 又は式中に許されない記号が現われた。

33	メッセージ： INVALID OPERATOR 演算のエラー
34	メッセージ： TOO LONG NAME 変数名が長すぎてバッファに入らない。 名前の終わり（英数字，空白以外の文字）が見つからない。
35	メッセージ： UNBALANCED PARENTHESES 右カッコ ')' が足りない。
36	メッセージ： FORTRAN SYNTAX ERROR 文法エラー
37	メッセージ： UNBALANCED QUOTE 引用符が閉じていない。
38	メッセージ： IT IS NOT VALUE 数以外の文字列を数値に変換しようとした。
39	メッセージ： TOO LARGE NUMBER 整数の最大値を越えた数がある。（-214783647～214783647）
40	メッセージ： ¥-expression syntax error ¥付サイズ式の文法エラー
41	メッセージ： SYNTAX ERROR OF H-FORMAT FORMAT文のH形編集の文法エラー
42	メッセージ： THIS STATEMENT IS NOT ALLOWED IN COMDECK *CALL文等，COMDECKブロックに許されていない文があった。
90	メッセージ： NEST ERROR DO文のネスト・エラー
91	メッセージ： NEST ERROR DO文のネスト・エラー，*ENDDOに対して，*DO文が見つからない。
92	メッセージ： NOT FOUND TERMINAL STATEMENT DO文の端末がない。
93	メッセージ： NOT FOUND *ENDDO-STATEMENT *ENDDO文がない。
95	メッセージ： LABEL TABLE OVER FLOW ラベル登録テーブルがオーバー・フローした。
96	メッセージ： DO-STACK OVER FLOW DO文処理用のスタックがオーバー・フローした。

EOS77 プリプロセッサ・エラー・メッセージ (WARNING level) 一覧表

102	メッセージ: SIZE POOL OVER FLOW -- PUT TO FILE
	サイズの式による定義のためのバッファがオーバー・フローした。以後、ファイルに出力する。
103	メッセージ: TEXT TABLE OVER FLOW -- PUT TO FILE
	テキスト・テーブルがオーバー・フローした。以後、ファイルに出力する。
105	メッセージ: TOO MANY LABELS, START FROM FIRST LABEL
	ラベルが多いので、再び最初から始めて新ラベルを作る。
111	メッセージ: UNDEFIMED SIZE SYMBOL (XXXX)
	XXXXが未定義である。 PARAMETER文で定義されているかどうかチェック
112	メッセージ: MORE THAN 20 LINES, AND PACKING
	編集後、継続行が20行を越えた。空白をつめてパッキングして出力する。
113	メッセージ: STATEMENT START CONTINUE -LINE
	継続行から文が始まっている。
114	メッセージ: UNDEFINED COMDECK NAME (XXXX)
	FORTTRAN77のINCLUDE文に現われるメンバー名XXXXが、COMDECK名として登録されていない。
115	メッセージ: SIZE SYMBOL CANNOT DEFINE ALL
	EOSソースのサイズ・ブロック内で、サイズ変数がすべて定義できなかった。 機番5から未定義サイズ変数を定義しなければならない。

EOS77 プリプロセッサ・エラー・メッセージ (SYSTEM ERROR level) 一覧表

<p>メッセージ： ***STOP --- READ ERROR, DEV = dev 機番 dev から入力時, エラーが発生した。</p>
<p>メッセージ： ***STOP --- WRITE ERROR, DEV = dev 機番 dev へ出力時, エラーが発生した。</p>
<p>メッセージ： ***STOP --- SIZE TABLE OVER FLOW サイズ変数のテーブルがオーバー・フローした。 著者に連絡。</p>
<p>メッセージ： ***STOP --- SIZE BUFFER OVER FLOW サイズ変数の ' 式による定義 ' のための制御テーブルがオーバー・フローした。 著者に連絡。</p>
<p>メッセージ： ***STOP --- TEXT TABLE OVER FLOW COMDECK テキストの制御テーブルがオーバー・フローした。 著者に連絡。</p>
<p>メッセージ： ***STOP --- LABEL TABLE OVER FLOW ラベル・テーブルがオーバー・フローした。 著者に連絡。</p>

付録2 JCLリスト

EOS77プリプロセッサをバッチ・ジョブで起動する場合の JCL の例を以下に示す。この例では、POファイル J3331.EOS77EX.FORT77 に格納されている EOS ソース・プログラムから実行可能なロード・モジュールを J3331.EOS77EX.LOAD(TEMPNAME) に作っている。ここに示した例と第4章を参考にすれば各自の目的に合った JCL を作れる。

```
//JCLG JOB
// EXEC JCLG
//SYSIN DD DATA,DLM='##'
// JUSER 12543331,SH.TOKUDA,0021.01
    T.O C.O W.1 P.O I.2
    OPTP PASSWORD=██████████,NOTIFY=J3331,MSGLEVEL=(1,1,1)
// EXEC VIVAPO,DSN='J3331.EOS77EX.FORT77'
//FT02F001 DD DSN=&&PSFILE,UNIT=TSSWK,SPACE=(TRK,(20,10)),
//          DISP=(NEW,PASS),DCB=(RECFM=FB,BLKSIZE=19040,LRECL=80)
POTOPS
//EOS77 EXEC PGM=EOS77,PARM='KEYSW=1,NAMSW=1,DOSW=1'
//STEPLIB DD DSN=J2636.EOS77V2.LOAD,DISP=SHR
//FT06F001 DD SYSOUT=*
//FT09F001 DD DSN=&&PSFILE,DISP=(OLD,DELETE,DELETE)
//FT10F001 DD DSN=&&F77,DISP=(NEW,PASS,DELETE),
//          UNIT=TSSWK,SPACE=(TRK,(30,10)),
//          DCB=(RECFM=FB,BLKSIZE=19040,LRECL=80)
// EXEC FORT77,SO=&&F77,OPT=3,A='SOURCE',LCT=60,Q=
// EXEC LKEDCT77,LM='J3331.EOS77EX',UNIT=TSSWK
##
//
```