

汎用非線形構造解析システムFINAS 平成7年度 改良・拡張および整備報告書

(動力炉・核燃料開発事業団 契約業務報告書)

1998年3月

株式会社 CRT総合研究所

本資料の全部または一部を複写・複製・転載する場合は、下記にお問い合わせ
してください。

〒319-1194 茨城県那珂郡東海村村松4番地49
核燃料サイクル開発機構
技術展開部 技術協力課

Inquiries about copyright and reproduction should be addressed to:
Technical Cooperation Section,
Technology Management Division,
Japan Nuclear Cycle Development Institute
4-49 Muramatsu, Tokai-mura, Naka-gun, Ibaraki 319-1194,
Japan

© 核燃料サイクル開発機構 (Japan Nuclear Cycle Development Institute)
1998

汎用非線形構造解析システム FINAS 平成 7 年度 改良・拡張および整備報告書

千葉 隆久^{*}, 上野 睦郎^{*}, 酒井 新吉^{*}
柄谷 和輝^{*}, 長谷川 豊^{*}, 都筑 新^{*}

要 旨

本報告書は、高速炉等の各種構造解析に適用するために開発された汎用非線形構造解析システム FINAS (Finite Element Nonlinear Structural Analysis System) の機能の改良、拡張および整備に関するものであり、平成 7 年 10 月から平成 8 年 3 月までの間に検討された内容およびその成果をまとめたものである。

以下に、改良、拡張および整備項目を示す。

- ・異方性材料の取り扱い
- ・高精度の構成モデルの適用
- ・FASTENING および縮合における計算効率の改善
- ・有限要素の修正
- ・FINAS (V. 13.0) の整備と公開

本報告書は、株式会社 CRC 総合研究所が、動力炉・核燃料開発事業団との契約により実施した業務報告書である。

契約番号 : 08HA (技) 083

事業団担当部署および担当者 : 月森 和之

(大洗工学センター 基盤技術開発部 構造・材料技術開発室)

※ : 株式会社 CRC 総合研究所

Improvement, Extension and Maintenance of Finite Element Nonlinear Structural Analysis System "FINAS" for fiscal 1995

Takahisa Chiba*, Mutsuro Ueno*, Shinkichi Sakai*
Kazuteru Garatani*, Yutaka Hasegawa*, Arata Tsuzuki*

Abstract

A general purpose finite element nonlinear structural analysis system FINAS has been developed since 1976 for fast breeder reactor components. Improvement, extension and maintenance of this system has been continued hitherto. This report describes the results of the work which was carried out during October 1995 to March 1996.

The items of improvement, extension and maintenance are as follows.

- Treatment of anisotropic materials
- Application of advanced constitutive equations
- Improvement of calculation efficiency of the matrix reduction procedure and FASTENING function
- Revision of a finite element
- Arrangement of FINAS (V. 13.0) and preparation for release

Work performed by CRC Research Institute, Inc. under contract with Power Reactor and Nuclear Fuel Development Corporation.

PNC Liaison : Kazuyuki Tsukimori

(Structure and Material Research Section, O-arai Engineering Center)

※ : CRC Research Institute, Inc.

目 次

緒言	1
A. 異方性材料の取り扱い	3
B. 高精度の構成モデルの組込み	15
C. FASTENING および縮合における計算効率の改善	125
D. 有限要素の修正	143
E. F I N A S (V13.0)の整備と公開	149
結言	153

緒 言

有限要素法による汎用非線形構造解析システム F I N A S (Finite Element Nonlinear Structural Analysis System)は、高速増殖炉における構造上の種々の問題を解決するための有効な手段となるように、動力炉・核燃料開発事業団（以下、動燃事業団）において昭和51年よりその開発が始められた。以降、最初の5年間(Phase I)では非弾性解析を中心に基本的な解析機能が備えられ、昭和56年からの5年間(Phase II)では動燃事業団の実用に供しながら、各種解析機能、前/後処理機能、検証例題の充実を図った。この10年間で F I N A S は汎用構造解析コードとしての機能を備え、昭和60年より F I N A S の一般公開が始まった。昭和61年からの5年間(Phase III)は、本来の目的であった高速増殖炉における解析要求に応えるだけでなく、様々な工業・工学分野での利用も考慮され、F I N A S をより高度で汎用性の高い構造解析システムとするため拡張・整備が行われた。

平成7年度は、F I N A S 改良・拡張および整備のための費用は、動燃事業団と弊社との間で契約を締結した「コンピュータ・プログラム『F I N A S』の利用者への使用許諾等に関する業務代行契約書」（平成6年12月1日付け）第9条 第3項に従い、前年度の販売金額の40%を割り当てることになった。

本年度はその主旨に沿い、改良・拡張項目として、異方性材料の取扱い、超弾性材料に対する解析機能、FASTENINGおよび縮合等の計算効率の改善、有限要素の修正等を検討したが、動燃事業団大洗工学センターの要請により、超弾性材料に対する解析機能の代わりに、高精度の構成モデルの適応を優先することになった。また、F I N A S (V13.0)の整備およびバージョンに対応した使用説明書の拡張を行ない、ユーザーへ公開した。

A. 異方性材料の取り扱い

1. 組込み機能の概要

現在の FINAS は異方性を持った材料特性を定義する場合、ユーザーサブプログラム XELMATおよびXALPAにより、応力-ひずみマトリックス及び線膨張マトリックスを直接指定することになっている。このとき、ユーザーサブプログラムにより各応力場に対応したマトリックスを作ることも、応力を定義する座標系と材料特性を定義する座標系が異なる場合には、これらの座標変換をサブプログラムの中で行う必要が生じる。このように現在の FINAS では異方性材料を定義することは可能であるが、全てユーザーの作成するサブプログラムに依存しているため、非常に煩雑な手続きが必要であった。

この点を改良するため、熱弾性領域に限定して直交異方性材料が MATERIAL データを用いて簡単に入力できるようにする。改良作業は2段階とし、第1段階では直交異方性材料におけるヤング係数、ポアソン比、せん断弾性係数、線膨張係数の材料データ（各3個ずつ計12個）が MATERIAL データにより入力可能とし、これらが応力成分と同じの座標系により定義されたとする。第2段階では異方性を定義する座標系として全体座標系およびユーザーが定義する任意の座標系を使用可能とする。

本年度の作業範囲はこの第1段階までとし、FINASにおいて応力成分と同じ座標系による直交異方性弾性材料モデルの定義を可能とする。また、これら異方性を持った材料定数は温度依存性を持たないものとする。

2. 理論

FEMにおいて、等方性材料と異方性材料のちがいは、応力-ひずみマトリックスと線膨張マトリックスの差となって現われる。ここでは、まず3次元応力場について、直交異方性弾性材料に対する応力-ひずみマトリックス $[E^e]$ と線膨張マトリックス $[\alpha]$ を導く。

2.1 3次元応力場における直交異方性弾性材料の応力-ひずみマトリックス $[E^e]$ の導出

弾性材料におけるたわみ性マトリックス $[C^e]$ は、Hookの法則により弾性ひずみ $\{\epsilon^e\}$ と応力 $\{\sigma\}$ 間の関係とし以下のように書ける。

$$\{\epsilon^e\} = [C^e] \{\sigma\} \quad 1)$$

応力-ひずみマトリックス $[E^e]$ は $[C^e]$ の逆行列として以下のように定義される。

$$\{\sigma\} = [E^e] \{\epsilon^e\} \quad 2)$$

$$[E^e] = [C^e]^{-1} \quad 3)$$

ここでは、式の展開を簡単にするために応力と弾性ひずみをベクトル型で表記しているが、本来これらは2階のテンソルであり、各々9つの成分を持つ。従って、これらの間の関係を定義する $[E^e]$ または $[C^e]$ は4階のテンソルとなり、最大91個の成分を持ち得る。ただし、応力及びひずみ成分の対称性とひずみエネルギーが定義されるなら $[E^e]$ または $[C^e]$ は対称性を持つことにより、独立した成分の数は21個となる。つまり、最も一般的な異方性材料における応力-ひずみ関係は21個の成分を持つ。これは、ベクトル表記により、 $[E^e]$ を 6×6 の対称マトリックスで定義した場合の各成分である。直交異方性材料においては独立した成分の数はさらに減り9つとなる。

直交異方性材料のたわみ性マトリックス $[C^e]$ を材料試験により定められる3つの縦弾性係数 (E_1, E_2, E_3)、6つのポアソン比 ($\nu_{12}, \nu_{13}, \nu_{21}, \nu_{23}, \nu_{31}, \nu_{32}$) 3つのせん断弾性係数 (G_{12}, G_{23}, G_{31}) により、以下のように定義する。

$$\begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix} = \begin{pmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{21}}{E_2} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{31}}{E_3} & -\frac{\nu_{32}}{E_3} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{pmatrix} \quad 1)$$

ただし、前述の $[C^e]$ の対称性より以下の式が成り立つから、独立した材料定数は9つとなる。

$$\frac{\nu_{12}}{E_1} = \frac{\nu_{21}}{E_2}, \quad \frac{\nu_{13}}{E_1} = \frac{\nu_{31}}{E_3}, \quad \frac{\nu_{23}}{E_2} = \frac{\nu_{32}}{E_3} \quad 4)$$

ここでは $E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}, G_{23}, G_{31}$ の9つの材料定数を独立とすると、残りの $\nu_{13}, \nu_{21}, \nu_{32}$ は4)式より以下のように与えられる。

$$\nu_{21} = \nu_{12} \frac{E_2}{E_1}, \nu_{13} = \nu_{31} \frac{E_1}{E_3}, \nu_{32} = \nu_{23} \frac{E_3}{E_2} \quad 5)$$

求めるべき、応力-ひずみ関係マトリックスは、1)'の $[C^e]$ の逆行列として次式となる。

$$[E^e] = \begin{pmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 & 0 \\ E_{12} & E_{22} & E_{23} & 0 & 0 & 0 \\ E_{13} & E_{23} & E_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & E_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & E_{66} \end{pmatrix} \quad 6)$$

ただし、6)式の各項は次式で与えられる。

$$\begin{aligned} E_{11} &= \frac{1}{E_2 E_3 \Delta} (1 - \nu_{23} \nu_{32}) \\ E_{22} &= \frac{1}{E_1 E_3 \Delta} (1 - \nu_{13} \nu_{31}) \\ E_{33} &= \frac{1}{E_1 E_2 \Delta} (1 - \nu_{12} \nu_{21}) \\ E_{12} &= \frac{1}{E_2 E_3 \Delta} (\nu_{21} + \nu_{31} \nu_{23}) \\ E_{23} &= \frac{1}{E_1 E_3 \Delta} (\nu_{32} + \nu_{12} \nu_{31}) \\ E_{13} &= \frac{1}{E_1 E_2 \Delta} (\nu_{13} + \nu_{12} \nu_{23}) \\ E_{44} &= G_{12}, E_{55} = G_{23}, E_{66} = G_{31} \\ \Delta &= \frac{1}{E_1 E_2 E_3} (1 - \nu_{12} \nu_{21} - \nu_{23} \nu_{32} - \nu_{31} \nu_{13} - 2\nu_{12} \nu_{23} \nu_{31}) \end{aligned} \quad 7)$$

2.2 3次元応力場における直交異方性材料の線膨張係数マトリックス $[\alpha]$ の導出

熱ひずみ増分 $\{d\epsilon^T\}$ は瞬間線膨張係数を成分とする線膨張係数マトリックス $[\alpha]$ と、温度増分 dT によって以下のように定義される。

$$\{d\epsilon^T\} = [\alpha] dT \quad 8)$$

直交異方性材料においては、各方向の瞬間線膨張係数 $\alpha_1, \alpha_2, \alpha_3$ を用いて熱ひずみが以下のように定義される。

$$\{d\epsilon_x^T d\epsilon_y^T d\epsilon_z^T d\gamma_{xy}^T d\gamma_{yz}^T d\gamma_{zx}^T\}^T = \{\alpha_1 \alpha_2 \alpha_3 0 0 0\}^T dT \quad 9)$$

2.3 各応力場における $[E^e]$ と $[\alpha]$ の導出

以下、FINASで定義されている11の応力場について、応力成分と直交異方性弾性材料についての $[E^e]$ と $[\alpha]$ を示す。ここにKFIELDはFINAS内部で用いられている応力場の識別変数である。

・単軸応力場 (KFIELD=1)

単軸応力場では以下のように E_1 と α_1 を定義すればよいが、これらは等方性の場合に同様であり、特に異方性のマトリックスを作る必要はない。

$$\begin{aligned} \{\sigma\} &= \{\sigma_x\} \\ [E^e] &= [E] \\ \{\alpha\} &= \{\alpha_1\} \end{aligned} \tag{10}$$

・平面応力場 (KFIELD=2)

平面応力場では、 $[E^e]$ マトリックスの成分を先の手順に従って計算し直す必要がある。ここでは詳細を記さないが、結果として以下の式が得られる。また、必要な材料定数は $(E_1, E_2, \nu_{12}, G_{12}, \alpha_1, \alpha_2)$ の6個となる。

$$\begin{aligned} \{\sigma\} &= \{\sigma_x \sigma_y \tau_{xy}\}^T \\ [E^e] &= \begin{pmatrix} E_{11} & E_{12} & 0 \\ E_{12} & E_{22} & 0 \\ 0 & 0 & E_{33} \end{pmatrix} \\ \{\alpha\} &= \{\alpha_1 \alpha_2 0\}^T \end{aligned} \tag{11}$$

ただし、

$$\begin{aligned} E_{11} &= \frac{E_1^2}{E_1 - E_2 \nu_{12}^2} \\ E_{22} &= \frac{E_2^2}{E_1 - E_2 \nu_{12}^2} \\ E_{12} &= \frac{E_1 E_2 \nu_{12}}{E_1 - E_2 \nu_{12}^2} \\ E_{33} &= G_{12} \end{aligned}$$

・ 3次元応力場 (KFIELD=3)

3次元応力場は、先に導いた 6), 7), 9) 式である。

・ 軸対称応力場 (KFIELD=4)

軸対称応力場は3次元応力場において、せん断項を1項のみにした場合と同様になる。従って $[E^e]$ の各項は7) 式で与えられる。必要な材料定数は ($E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}, \alpha_1, \alpha_2, \alpha_3$) の10個である。

$$\{\sigma\} = \{\sigma_R \sigma_Z \sigma_\theta \tau_{RZ}\}^T$$

$$[E^e] = \begin{pmatrix} E_{11} & E_{12} & E_{13} & 0 \\ E_{12} & E_{22} & E_{23} & 0 \\ E_{13} & E_{23} & E_{33} & 0 \\ 0 & 0 & 0 & E_{44} \end{pmatrix} \quad (12)$$

$$\{\alpha\} = \{\alpha_1 \alpha_2 \alpha_3 0\}^T$$

・ 平面ひずみ応力場 (KFIELD=5)

FINASの平面ひずみ応力場はZ方向の応力を陽に計算している。従って、この場合も軸対称応力場と同様、 $[E^e]$ の各項は7) 式で与えられる。必要な材料定数も ($E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}, \alpha_1, \alpha_2, \alpha_3$) の10個である。

$$\{\sigma\} = \{\sigma_X \sigma_Y \sigma_Z \tau_{XY}\}^T$$

$$[E^e] = \begin{pmatrix} E_{11} & E_{12} & E_{13} & 0 \\ E_{12} & E_{22} & E_{23} & 0 \\ E_{13} & E_{23} & E_{33} & 0 \\ 0 & 0 & 0 & E_{44} \end{pmatrix} \quad (13)$$

$$\{\alpha\} = \{\alpha_1 \alpha_2 \alpha_3 0\}^T$$

・梁の応力場 (KFIELD=6)

梁の応力場では、直成分が1つとせん断成分が2つであり、以下のように簡単な式になる。必要な材料定数は $(E_1, G_{12}, G_{31}, \alpha_1)$ の4個である。

$$\begin{aligned} \{\sigma\} &= \{\sigma_x \tau_{xy} \tau_{zx}\}^T \\ [E^s] &= \begin{pmatrix} E_1 & 0 & 0 \\ 0 & G_{12} & 0 \\ 0 & 0 & G_{31} \end{pmatrix} \\ \{\alpha\} &= \{\alpha_1 \ 0 \ 0\}^T \end{aligned} \quad 14)$$

・シェル応力場 (KFIELD=7)

シェル応力場は、平面応力場においてせん断項を3項にした場合に等しい。必要な材料定数は $(E_1, E_2, \nu_{12}, G_{12}, G_{23}, G_{31}, \alpha_1, \alpha_2)$ の8個である。

$$\begin{aligned} \{\sigma\} &= \{\sigma_x \sigma_y \tau_{xy} \tau_{yz} \tau_{zx}\}^T \\ [E^s] &= \begin{pmatrix} E_{11} & E_{12} & 0 & 0 & 0 \\ E_{12} & E_{22} & 0 & 0 & 0 \\ 0 & 0 & E_{33} & 0 & 0 \\ 0 & 0 & 0 & E_{44} & 0 \\ 0 & 0 & 0 & 0 & E_{55} \end{pmatrix} \\ \{\alpha\} &= \{\alpha_1 \ \alpha_1 \ 0 \ 0 \ 0\}^T \end{aligned} \quad 15)$$

ただし

$$\begin{aligned} E_{11} &= \frac{E_1^2}{E_1 - E_2 \nu_{12}^2} \\ E_{22} &= \frac{E_2^2}{E_1 - E_2 \nu_{12}^2} \\ E_{12} &= \frac{E_1 E_2 \nu_{12}}{E_1 - E_2 \nu_{12}^2} \\ E_{33} &= G_{12}, \quad E_{44} = G_{23}, \quad E_{55} = G_{31} \end{aligned}$$

・薄肉軸対称シェル応力場 (KFIELD=8)

薄肉軸対称シェル応力場は、せん断項がなく、直成分については平面応力場と同じである。必要な材料定数は ($E_1, E_2, \nu_{12}, \alpha_1, \alpha_2$) の5個である。

$$\begin{aligned} \{\sigma\} &= \{\sigma_s \sigma_\theta\}^T \\ [E^e] &= \begin{pmatrix} E_{11} & E_{12} \\ E_{12} & E_{22} \end{pmatrix} \\ \{\alpha\} &= \{\alpha_1 \alpha_2\}^T \end{aligned} \quad (16)$$

ただし

$$\begin{aligned} E_{11} &= \frac{E_1^2}{E_1 - E_2 \nu_{12}^2} \\ E_{22} &= \frac{E_2^2}{E_1 - E_2 \nu_{12}^2} \\ E_{12} &= \frac{E_1 E_2 \nu_{12}}{E_1 - E_2 \nu_{12}^2} \end{aligned}$$

・ELBOW3Rの応力場 (KFIELD=9)

特殊な応力場であり異方性を定義することは少ないと思われるが、平面応力と同様の考え方を適用すればよい。必要な材料定数は ($E_1, E_2, \nu_{12}, G_{12}, \alpha_1, \alpha_2$) の6個となる。

$$\begin{aligned} \{\sigma\} &= \{\sigma_1 \sigma_s \tau_{sn}\}^T \\ [E^e] &= \begin{pmatrix} E_{11} & E_{12} & 0 \\ E_{12} & E_{22} & 0 \\ 0 & 0 & E_{33} \end{pmatrix} \\ \{\alpha\} &= \{\alpha_1 \alpha_2 0\}^T \end{aligned} \quad (17)$$

ただし、 $[E^e]$ マトリックスの成分は 11) 式と同じ

・CONT2の応力場 (KFIELD=10)

接触要素CONT2の応力場であり、FINAS 内部で特殊な扱いをしている。この要素については異方性を考慮することは無く、また $[E^e]$ マトリックスを作成する必要もない。

・軸対称シェル応力場 (KFIELD=11)

ここでの扱いは、平面応力場に同様である。必要な材料定数も ($E_1, E_2, \nu_{12}, G_{12}, \alpha_1, \alpha_2$) の6個となる。

$$\{\sigma\} = \{\sigma_s \sigma_\theta \tau_{s\theta}\}^T$$

$$[E^e] = \begin{pmatrix} E_{11} & E_{12} & 0 \\ E_{12} & E_{22} & 0 \\ 0 & 0 & E_{33} \end{pmatrix} \quad 18)$$

$$\{\alpha\} = \{\alpha_1 \alpha_2 0\}^T$$

ただし、

$$E_{11} = \frac{E_1^2}{E_1 - E_2 \nu_{12}^2}$$

$$E_{22} = \frac{E_2^2}{E_1 - E_2 \nu_{12}^2}$$

$$E_{12} = \frac{E_1 E_2 \nu_{12}}{E_1 - E_2 \nu_{12}^2}$$

$$E_{33} = G_{12}$$

以上、FINAS で用いられている各応力場を表1に、直交異方性弾性材料を定義するのに必要な材料定数を表2に示す。また、表に示したように FINAS の平面応力要素は、QMEM4D2等2次元空間で用いられる要素と、QMEM4等3次元空間で用いられる要素があり、前者の応力は全体座標系で定義するが、後者の応力は要素座標系で定義する。

(参考文献)

山田嘉昭 "塑性・粘弾性" 3章 有限要素法の基礎と応用シリーズ6 培風館 1980
井上達雄 "弾性力学の基礎" 5章 日刊工業新聞社 1979

表1 FINAS の応力場

KFIELD	応力場	成分数	応力成分	座標系	備考
1	単軸応力場	1	σ_x	要素	
2	平面応力場 (~D2要素)	3	$\sigma_X, \sigma_Y, \tau_{XY}$	全体	2次元要素
	平面応力場		$\sigma_x, \sigma_y, \tau_{xy}$	要素	3次元要素
3	3次元応力場	6	$\sigma_X, \sigma_Y, \sigma_Z, \tau_{XY}, \tau_{YZ}, \tau_{ZX}$	全体	
4	軸対称応力場	4	$\sigma_R, \sigma_Z, \sigma_\theta, \tau_{RZ}$	全体	
5	平面ひずみ応力場	4	$\sigma_X, \sigma_Y, \sigma_Z, \tau_{XY}$	全体	
6	梁の応力場	3	$\sigma_x, \tau_{xy}, \tau_{zx}$	要素	
7	シェル応力場	5	$\sigma_x, \sigma_y, \tau_{xy}, \tau_{yz}, \tau_{zx}$	要素	
8	薄肉軸対称シェル応力場	2	σ_s, σ_θ	要素	
9	ELBOW3Rの応力場	3	$\sigma_l, \sigma_s, \tau_{sn}$	要素	
10	CONT2の応力場	-	-	-	[E ^o]を作らない
11	軸対称シェル応力場	3	$\sigma_s, \sigma_\theta, \tau_{s\theta}$	要素	

表2 直交異方性弾性材料を定義するのに必要な材料定数

KFIELD	応力場	直交異方性材料の材料定数	線膨張係数
1	単軸応力場	E_1	α_1
2	平面応力場 (~D2要素)	$E_1, E_2, \nu_{12}, G_{12}$	α_1, α_2
	平面応力場	$E_1, E_2, \nu_{12}, G_{12}$	α_1, α_2
3	3次元応力場	$E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}, G_{23}, G_{31}$	$\alpha_1, \alpha_2, \alpha_3$
4	軸対称応力場	$E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}$	$\alpha_1, \alpha_2, \alpha_3$
5	平面ひずみ応力場	$E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}$	$\alpha_1, \alpha_2, \alpha_3$
6	梁の応力場	E_1, G_{12}, G_{31}	α_1
7	シェル応力場	$E_1, E_2, \nu_{12}, G_{12}, G_{23}, G_{31}$	α_1, α_2
8	薄肉軸対称シェル応力場	E_1, E_2, ν_{12}	α_1, α_2
9	ELBOW3Rの応力場	$E_1, E_2, \nu_{12}, G_{12}$	α_1, α_2
10	CONT2の応力場	-	-
11	軸対称シェル応力場	$E_1, E_2, \nu_{12}, G_{12}$	α_1, α_2

3. システム設計

3.1 システム構成

システム全体の流れ、および構成は変更なし。

3.2 入力データ

・異方性材料に対するオプションキーの追加

材料特性を定義するMATERIALカードの中のB₂カードの66～70カラムにオプションキーとして k_{10} を追加する。 k_{10} は異方性材料に対するオプションであり、その値は =0 : 等方性材料、=1 : 直交異方性材料（異方性軸は各要素で定義される応力の方向に一致）を意味し、無記入の場合は0とする。

ただし、将来的に k_{10} の値は、=2 : 直交異方性材料（異方性軸は全体座標に一致）、=3 : 直交異方性材料（異方性軸はM CORDカードで定義、M CORDカードは熱伝導解析で既に定義されているものに同様）を追加する。

・異方性材料に対する材料定数の入力

材料特性を定義するMATERIALカードの中のCカードの21～30カラムで定義される材料特性の種類 *kind* として、異方性材料の材料定数である $E_1, E_2, E_3, \nu_{12}, \nu_{23}, \nu_{31}, G_{12}, G_{23}, G_{31}, \alpha_1, \alpha_2, \alpha_3$ の12種類を表現する $E1, E2, E3, NU12, NU23, NU31, G12, G23, G31, ALPHA1, ALPHA2, ALPHA3$ を追加する。

3.3 出力データ

1) プリント出力

モデルデータの詳細情報プリントの拡張を行う。

2) プロット出力

変更なし。

3) ポストテープ出力

変更なし。

3.4 データブロック

材料に関するテーブルであるMATNにおいて異方性材料に対するオプションキーを1つ追加する。このオプションは識別コード15番 異方性材料のオプションとなり、値は1つで先の k_{10} の値を保持する。

異方性材料に対する最大12個の材料定数は、従来の A1～A10, B1～B6 と同じ領域を用いる。つまり、入力はE1, NU12等使用者が理解しやすい記号を用いるが、FINASの内部ではこれらが表3のような領域に保存される。表3の番号とは、材料特性番号であり、サブルーチンMATEXTにより材料定数を取り出すときに使用する。

表3 直交異方性材料の材料定数

番号	本来の記号	入力記号	意味	番号	本来の記号	入力記号	意味
14	A1	E1	E_1	20	B1	G12	G_{12}
15	A2	E2	E_2	21	B2	G23	G_{23}
16	A3	E3	E_3	22	B3	G31	G_{31}
17	A4	NU12	ν_{12}	23	B4	ALPHA1	α_1
18	A5	NU23	ν_{23}	24	B5	ALPHA2	α_2
19	A6	NU31	ν_{31}	25	B6	ALPHA3	α_3

3.5 グローバルコモン

コモン /MATRAL/ に異方性材料に対するオプションキー KANISO (Key of ANISotropic) を追加する。KANISO はサブルーチン MATINI において、データブロック MATN に追加した15番目のオプションキーの値が設定されるが、これは入力データの k_{10} の値である。

3.6 組込みATOM

変更なし。

4. モジュール設計

4.1 入力関連モジュール

異方性材料に対するオプションキーの追加と材料定数入力部分の拡張により表4のモジュールが拡張される。

表4 入力関連モジュール

モジュール名	内 容	サブルーチン名
IFP1	入力データの読み込みとデータブロックの作成 オプションキー k10 の入力とMATN へのセット E1,NU12等のデータ入力とMATNへのセット	
DMLOUT	入力データ詳細情報のプリント k10 の内容のプリント 異方性が定義された場合の材料定数のプリント	

4.2 [E^e] と [α] マトリックス関連モジュール

異方性が定義された場合、各応力場に応じて適切な [E^e] と [a] の設定を行う。[E^e] と [a] は EMSALL 及び EMSUPD の2つのモジュールで使用するが、今回の拡張では、その差を意識する必要はない。

[E^e] の設定についてはサブルーチンELMATXにおいて、KANISOを参照し、異方性が定義されている場合は、新たに作成するサブルーチンへジャンプさせ、この中で[E^e]を作る。

[a]についてはFINASの場合、直接熱ひずみを計算するため、陽に現われないが、サブルーチンESUBTで設定する。ただし、このとき熱ひずみの計算は2節の8)式を積分した以下の式で計算される。

$$\{\Delta\varepsilon^T\} = \int_T^{T+\Delta T} [\alpha] dT \quad (19)$$

表5 [E^e] と [α] の使用 関連モジュール

モジュール名	内 容	サブルーチン名
EMSALL EMSUPD	各応力場に応じ、適切な [E ^e] と [α] を設定する。	ELMATX ESUBT

B. 高精度の構成モデルの組込み

1. 材料パラメータの温度依存性を考慮した大野-王モデルの組み込み

1.1 機能拡張の概要

近年、弾塑性材料に対して材料挙動の記述性を向上させた各種の高精度な弾塑性構成モデルが提案されている。特に非線形繰返し非硬化領域モデル、マルチレイヤーモデル、2曲面繰返し塑性モデルはFINASへ組み込まれており、利用者は簡単に有限要素解析で使用することができる。しかし、これら3つのモデルを含め、ほとんどの弾塑性構成モデルは、単軸および多軸繰返し負荷でのラチェット変形において、ラチェットひずみの累積を過大に評価するという問題が指摘されている。

大野-王モデル^{1),2),3),4)}は動的回復の臨界状態を考慮した移動硬化モデルの定式化を行うことにより、この問題の改善を図った弾塑性構成モデルである。文献には大野-王モデルとして多直線型の応力-ひずみ関係を持つモデルIと、べき乗型の関数を用いることにより滑らかな応力-ひずみ関係を表現し得るモデルIIが提案されている。後に、このモデルは高速増殖炉の炉容器に発生する液面熱ラチェット問題に適用するために、材料パラメータの温度依存性を考慮した拡張がなされた⁵⁾。

大野-王モデルは従来のモデルに較べると、ラチェット挙動の記述性に優れるという大きな利点を持っている。しかし、このモデルはまだ研究段階にあり、特に材料パラメータの温度依存性に関してかなり簡単なモデル化がなされているため、今後改良が加えられる可能性がある。従って、今回はFINAS V13.0で整備された、任意の構成モデル組み込み機能を用いて大野-王モデルを定義する。

任意の構成モデル組み込み機能はユーザーサブルーチンXMAT1, XMAT2により構成モデルを定義するもので、一度ユーザーサブルーチンを組込んだロードモジュールを作成すれば、以降FINASに用意されている構成モデルと同様にモデルを利用することができる。しかし、従来の仕様では材料定数を全てXMAT1, XMAT2の中で用意しなければならなかった。この制限はユーザーサブルーチン作成者の負担を軽減させるためのものだが、利用者は材料定数を変更する場合、その都度サブプログラムXMAT1, XMAT2を変更し、ロードモジュールを作り直す必要があった。このような煩雑な手続きを改良するため、今回新たにユーザー定義の構成モデル用の材料データを入力するユーザーサブルーチンXREADを加えた。XREADはFINASの実行時に1度だけ呼ばれ、入力データのEND FINASカード以降に書かれた材料データを読み込み、XMAT1, XMAT2に引渡す

ために適当なコモンブロックにセットする。END FINAS カード以降のデータはFINASの実行に影響しないので、XREADによりFINASの入力データとは全く独立にユーザーサブルーチンXMAT1, XMAT2に材料データを与えることができる(1.3.3 参照)。

また、材料パラメータの温度依存性を考慮した大野-王モデル⁵⁾は、10個の背応力成分を持つが、任意の構成モデル組込み機能によるユーザーが定義する内部状態変数はベクトル型、スカラー型、フラッグ型とも各々最大9個に制限されていた。よって、今回この修正を行い、ユーザーが定義するベクトル型内部状態変数を最大9個から19個に拡張した。ただし、10個以上のベクトル型内部状態変数を定義した場合、解析は正常に行われるが、10番目以上の変数はプロット図で指定できない(記述するフィールドが1カラムしかない)ために、プロット図が得られないという問題がある。そこで、使用説明書等の記述はあくまで9個までとし、10以上の値を定義しても警告メッセージ(1.3.1 参照)は出すが、解析は正常に行われるような仕様とした。

今回組込んだ大野-王モデルは基本的に材料パラメータの温度依存性を考慮しており、文献⁵⁾に従ったものであるが(温度依存性を考慮した場合はモデルIの材料定数のみが示されている)、温度依存性を考慮しない場合やモデルIIへの拡張が選択できるようになっている。

これらユーザーサブルーチンにより定義された構成モデルはFINASによる検証例題により機能の検証が行われ、さらに多軸問題ではXMAT1, XMAT2のドライバープログラムであるADMODELによる検証も実施した。

- 1) 大野信忠、王建定：動的回復の臨界状態を考慮した移動硬化則(第1報、定式化および基本的特徴)日本機械学会論文集(A編)58巻547号, 488- (1992-3)
- 2) 大野信忠、王建定：動的回復の臨界状態を考慮した移動硬化則(第2報、実験結果に基づく検討)日本機械学会論文集(A編)58巻547号, 496- (1992-3)
- 3) N. Ohno, J.-D. Wang : Kinematic Hardening Rules with Critical State of Dynamic Recovery, Part I : Formulation and Basic Features for Ratchetting Behavior, Int. J. of Plasticity, Vol.9 pp375-390 (1993)
- 4) N. Ohno, J.-D. Wang : Kinematic Hardening Rules with Critical State of Dynamic Recovery, Part II : Application to Experiments of Ratchetting Behavior, Int. J. of Plasticity, Vol.9 pp391-403 (1993)
- 5) 小林峰雄、大野信忠：軸方向温度分布変動を受ける円筒の熱ラチェット解析(実験との比較)日本機械学会東海支部豊橋地区講演会講演論文集 305 No.953-2 (1995-7/7)

1.2 大野-王モデルの定式化

1.2.1 温度変動がない場合の定式化 (モデルI, モデルII)

大野-王モデルの特徴は背応力にあり、他の部分は従来の古典的な弾塑性モデルに同一である。従って、ここでは背応力の発展式について、その導出過程¹⁾を記す。

大野-王モデルでは背応力成分を M 個の成分に分解する。

$$\alpha = \sum_{i=1}^M \alpha_i \quad 1)$$

動的回復は背応力がある臨界値に達したときに生じると仮定し、この臨界状態を次式によって表現する。

$$f_i = \bar{\alpha}_i^2 - r_i^2 = 0 \quad 2)$$

右辺第1項は背応力成分の大きさを表現し、以下の式で定義する。

$$\bar{\alpha}_i = \sqrt{\frac{3}{2} \alpha_i : \alpha_i} \quad 3)$$

大野-王モデルでは、動的回復の臨界状態を考慮した背応力の発展式が、以下のように表現されると仮定する。

$$\dot{\alpha}_i = \frac{2}{3} h_i \dot{\epsilon}^p - H(f_i) \dot{\lambda}_i \frac{\alpha_i}{r_i} \quad 4)$$

上式左辺の第1項は硬化を表し、 h_i は加工硬化係数と同様な材料定数である。第2項は動的回復項であり、Heaviside 関数により²⁾式が成立する臨界状態でのみ作用する。未定係数は臨界状態が保持されるための適応条件である下式を用いて求める。

$$\dot{f}_i = 0 \quad (\alpha_i : \dot{\alpha}_i = 0) \quad 5)$$

よって、未定係数は以下のように与えられる。

$$\dot{\lambda}_i = \langle \dot{\epsilon}^p : k_i \rangle h_i \quad 6)$$

ただし、

$$k_i = \frac{\alpha_i}{\bar{\alpha}_i} \quad 7)$$

6), 7) を 4) に代入すれば以下の式となり、

$$\dot{\alpha}_i = h_i \left[\frac{2}{3} \dot{\epsilon}^p - H(f_i) \langle \dot{\epsilon}^p : k_i \rangle \frac{\alpha_i}{r_i} \right] \quad 8)$$

さらに $h_i = \zeta_i$ とおくと、モデル I における背応力の発展式が得られる。

$$\dot{\alpha}_i = \zeta_i \left[\frac{2}{3} r_i \dot{\epsilon}^p - H(f_i) \langle \dot{\epsilon}^p : k_i \rangle \alpha_i \right] \quad 9)$$

ここまでのモデル化 (モデル I) は、2) 式で定義する曲面の内部では背応力項の動的回復が全く生じないと仮定した。しかし、これは背応力が限界値に近づくとつれて非線形的に活発になると考える方が自然である。この着想により 9) 式を以下のように拡張したのがモデル II である。モデル II では材料定数 m_i を用い、Heaviside 関数をべき乗型の関数で表現する。また、 m_i が無限大のとき、モデル I とモデル II は同じになる。以下にモデル II における背応力の発展式を示す。

$$\dot{\alpha}_i = \zeta_i \left[\frac{2}{3} r_i \dot{\epsilon}^p - \left(\frac{\alpha_i}{r_i} \right)^{m_i} \langle \dot{\epsilon}^p : k_i \rangle \alpha_i \right] \quad 10)$$

(参考) 式中で使われる記号は以下の意味を持つ

:	2 階のテンソル間の内積	$a:b = \text{tr}(ab)$
$\langle \rangle$	Macauley の括弧	$x \geq 0$ なら $\langle x \rangle = x$, $x < 0$ なら $\langle x \rangle = 0$
H	Heaviside 関数	$x \geq 0$ なら $H(x) = 1$, $x < 0$ なら $H(x) = 0$
$ $	テンソルの大きさ	$ x ^2 = x : x$

1.2.2 材料パラメータの温度依存性を考慮した定式化

大野らは、文献⁵⁾において大野-王モデルを材料パラメータの温度依存性を考慮して拡張した。この拡張ではモデルを簡単にするため、背応力と抗応力の温度履歴非依存性を仮定している。このモデルでは、背応力と抗応力が温度依存性を表す同じ関数 $\phi(T)$ を分離した以下の式で、各々表現できるとする。

$$\alpha^*(T) = \Phi(T) \alpha(T_0) \quad (11)$$

$$\sigma_y(T) = \Phi(T) \sigma_y(T_0) \quad (12)$$

ここに11) 式右辺の $\alpha(T_0)$ 動は基準温度: T_0 における背応力であり、温度非依存の場合の1) 式の左辺に対応する。また12) 式右辺の $\sigma_y(T_0)$ も同様に基準温度における抗応力である。ただし、下に示すように $\phi(T)$ は基準温度において1になるような関数とする。

$$\Phi(T_0) = 1 \quad (13)$$

11), 12) で表される背/抗応力と偏差応力 s により、降伏関数は以下のように書ける。

$$f = \sqrt{\frac{3}{2} (s - \alpha^*) : (s - \alpha^*)} - \sigma_y \quad (14)$$

よって、背応力が温度変動と塑性ひずみ（変形）により発展すると考えるなら、その全微分形は以下のように分離された形になる。

$$d\alpha^* = \alpha(T_0) \frac{d\Phi(T)}{dT} dT + \Phi(T) \frac{\partial \alpha(T_0)}{\partial \bar{\epsilon}^p} d\bar{\epsilon}^p \quad (15)$$

一方、抗応力は変形により進展しないと仮定しているため、これも温度依存項が簡単に分離される。

$$d\sigma_y = \sigma_y(T_0) \frac{d\Phi(T)}{dT} dT \quad (16)$$

以上の定式化により、基準温度における背/抗応力さえ計算しておけば、これに $\phi(T)$ を乗することにより簡単に温度依存性を考慮した解析に拡張することができる。

1.2.3 構成方程式の導出

ここでは、大野-王モデルの構成方程式を導く前に、一般的な背/抗応力型の弾塑性モデルについて、その構成方程式を導出する。なお、本項ではFINASの任意の構成モデル組込み機能で指定されている形の構成方程式を導くことを目的としている。また、式展開については各種の応力場に対応させるため、従来のベクトル型の定式化を採用している。

1.2.3.1 材料パラメータの温度依存性を考慮した背/抗応力型構成モデルの構成式

温度変動下において以下のような背/抗応力型の構成モデルを定義する。ただし、材料は等方性とし、ミーゼス型の降伏条件に従うとする。

・ひずみの分解

熱弾塑性領域において、全ひずみ増分 $\{d\epsilon\}$ は弾性ひずみ増分 $\{d\epsilon^e\}$ と塑性ひずみ増分 $\{d\epsilon^p\}$ 、熱ひずみ増分 $\{d\epsilon^T\}$ の和として以下のように定義される。

$$\{d\epsilon\} = \{d\epsilon^e\} + \{d\epsilon^p\} + \{d\epsilon^T\} \quad (17)$$

・Hookの法則

弾性ひずみ $\{\epsilon^e\}$ と応力 $\{\sigma\}$ の間には以下のようによく知られたHookの法則が成立つ。ここに、 $[D^e]$ は弾性係数マトリックスであり、等方性材料ではヤング率とポアソン比の関数となる。

$$\{\sigma\} = [D^e] \{\epsilon^e\} \quad (18)$$

18) 式は増分形ではなく、温度 T の関数である $[D^e]$ の各状態で成立する。従って、18) 式の増分形は 19) 式となる。

$$\{d\sigma\} = [D^e] \{d\epsilon^e\} + \frac{\partial [D^e]}{\partial T} \{\epsilon^e\} dT \quad (19)$$

・塑性ひずみ

塑性ひずみは流れ側 (flow rule) により、塑性ポテンシャル f (ここでは降伏関数と同じとする) の接線方向を向く。ここに λ^p は塑性状態が継続するなら正のスカラー量となり、除荷の判定に用いられる。

$$\{d\epsilon^p\} = \lambda^p \left(\frac{\partial f}{\partial \sigma} \right) \quad (20)$$

・熱ひずみ

熱ひずみ増分 $\{d\epsilon^T\}$ は瞬間線膨張係数 α^T を成分に持つ線膨張マトリックス $\{\alpha^T\}$ と温度増分 dT により、以下のように書ける。

$$\{d\epsilon^T\} = \{\alpha^T\} dT \quad (21)$$

・降伏関数

ここでは降伏関数 f として背応力 $\{\alpha\}$ 、抗応力 κ とした、背/抗応力型の関数を用いる。この降伏関数は先に述べたように塑性ポテンシャルとしても使用している。

$$f = \bar{\sigma} - \kappa \quad (22)$$

$\bar{\sigma}$ は相当応力であり、ここでは $\{\sigma - \alpha\}$ についてのミーゼス型関数とする。

$$\bar{\sigma} = \bar{\sigma}(\{\sigma - \alpha\}) \quad (23)$$

・背応力の進展則

背応力の進展則を、以下のような形に仮定する。ここに H_c は背応力のみが進展するときの加工硬化係数であり、その増分の方角を表わす $\{\eta\}$ は9) 式で定義する。

$$\{d\alpha\} = \sqrt{\frac{2}{3}} H_c \{\eta\} d\bar{e}^p + \frac{\partial \{\alpha\}}{\partial T} dT \quad (24)$$

$$\{\eta\} = \frac{\{d\alpha\}}{|\{d\alpha\}|} \quad (25)$$

ここに $|\{d\alpha\}|$ は $\{d\alpha\}$ の大きさを表わすが、この計算は $\{d\alpha\}$ をテンソルとして扱い、 $\{\eta\}$ もベクトル表記で各項が正しく求まるような変換が必要である。

$$|\{d\alpha\}| = \sqrt{d\alpha_{ij} d\alpha_{ij}} \quad (26)$$

また $\bar{d}e^p$ は相当塑性ひずみであり、以下のように定義する。

$$\bar{\sigma} d\bar{e}^p = \{\sigma - \alpha\}^T \{de^p\} \quad (27)$$

・抗応力の進展則

抗応力の進展則も、24) 式に同様に定義する。

$$d\kappa = H_i \bar{d}e^p + \frac{\partial \kappa}{\partial T} dT \quad (28)$$

以上の式より構成方程式を導く。ただし構成方程式は、任意の構成モデルへの組込みを容易するために以下の形に変形する。

$$\{d\sigma\} = [D] \{d\epsilon\} + \{dg\} \quad (29)$$

・弾性域

22) 式で定義される f 関数が負であれば、弾性域にあり塑性ひずみ増分は発生しない。このときの構成方程式は 19) 式と 21) 式より容易に求まり、以下の式となる。

$$\{d\sigma\} = [D^e] \{d\epsilon\} + \left(\frac{\partial [D^e]}{\partial T} \{\epsilon^e\} - [D^e] \{\alpha^T\} \right) dT \quad (30)$$

・塑性域

22) 式で定義される f 関数が 0 であれば塑性域にあり、さらに塑性状態が続くなら増分後も 22) 式が成立する。よって近似的に全微分の式 31) が成り立つ。31) 式は適応条件式と呼ばれ、負荷後も応力が降伏曲面上を動くための条件である。

$$\begin{aligned} f &= 0 \quad \text{より} \\ df &= 0 \end{aligned} \quad (31)$$

31) 式は以下のように展開される。

$$\begin{aligned} df &= \left(\frac{\partial f}{\partial \sigma} \right)^T \{d\sigma\} + \left(\frac{\partial f}{\partial \alpha} \right)^T \{d\alpha\} + \frac{\partial f}{\partial \kappa} d\kappa \\ &= \left(\frac{\partial f}{\partial \sigma} \right)^T \{d\sigma\} - \left(\frac{\partial f}{\partial \sigma} \right)^T \{d\alpha\} - d\kappa \end{aligned} \quad (31)'$$

あとは、31) 式の各増分項に 17), 19), 20), 21), 24), 27), 28), 29), 30) を代入し、 $\overline{d\epsilon^p}$ を求め構成方程式を導く。ただし、参考までに今回のようなミーゼス型の降伏関数を仮定するなら以下の式が成り立つ。

$$\overline{\sigma} = \{\sigma - \alpha\}^T \left(\frac{\partial f}{\partial \sigma} \right)$$

$$\lambda^p = \overline{d\epsilon^p}$$

以上の式より、構成方程式は以下のように導かれる。

$$\begin{aligned}
 \{d\sigma\} = & \left\{ [D^e] - \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e]}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} \{d\varepsilon\} \\
 & - \left\{ [D^e] - \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e]}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} \{\alpha^T\} dT \\
 & + \left\{ \frac{\partial [D^e]}{\partial T} \{\varepsilon^e\} - \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T \frac{\partial [D^e]}{\partial T} \{\varepsilon^e\}}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} dT \\
 & + \left\{ \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T \left(\frac{\partial \alpha}{\partial T} \right) + [D^e] \left(\frac{\partial f}{\partial \sigma} \right) \frac{\partial \kappa}{\partial T}}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} dT \quad (32)
 \end{aligned}$$

または、29) 式より

$$\{d\sigma\} = [D^{ep}] \{d\varepsilon\} + \{dg\} \quad (29)'$$

$$[D^{ep}] = \left\{ [D^e] - \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e]}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} \quad (33)$$

$$\{dg\} = - [D^{ep}] \{\alpha^T\} dT$$

$$\begin{aligned}
 & + \left\{ \frac{\partial [D^e]}{\partial T} \{\varepsilon^e\} - \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T \frac{\partial [D^e]}{\partial T} \{\varepsilon^e\}}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} dT \\
 & + \left\{ \frac{[D^e] \left(\frac{\partial f}{\partial \sigma} \right) \left(\frac{\partial f}{\partial \sigma} \right)^T \left(\frac{\partial \alpha}{\partial T} \right) + [D^e] \left(\frac{\partial f}{\partial \sigma} \right) \frac{\partial \kappa}{\partial T}}{H_i + \sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} + \left(\frac{\partial f}{\partial \sigma} \right)^T [D^e] \left(\frac{\partial f}{\partial \sigma} \right)} \right\} dT \quad (34)
 \end{aligned}$$

1.2.3.2 大野-王モデルの構成式

大野-王モデルも抗/背応力型の弾塑性構成モデルでから、その構成式は前節に同様である。ただし、抗/背応力の発展式については前節の式に対比させるために、多少の式展開が必要になる。ここでは詳細な式展開までは触れないが、大野-王モデルにおけるこれらの項を示す。

・降伏関数

降伏関数 22) 式は 14) 式となる。ここに抗応力について σ_y と κ の物理的な意味は同じであり、背応力の α^* は温度依存を考慮したもの、 α は基準温度での値 (11) 式参照) である。

$$f = \sqrt{\frac{3}{2} (s - \alpha^*) : (s - \alpha^*)} - \sigma_y \quad (14)$$

・背応力の発展則

背応力の発展則 24) 式は 15) 式で表される。

$$d\alpha^* = \alpha \frac{d\Phi}{dT} dT + \Phi \frac{\partial \alpha}{\partial \bar{\epsilon}^p} d\bar{\epsilon}^p \quad (15)$$

ここに、右辺第 2 項は温度非依存の式 9), 10) より導かれ、以下の式となる。

モデル I

$$\frac{\partial \alpha_i}{\partial \bar{\epsilon}^p} = \sqrt{\frac{2}{3}} \left[\zeta_i r_i \left(\frac{\frac{\partial f}{\partial s}}{\left| \frac{\partial f}{\partial s} \right|} - H(f) \langle \Lambda \rangle \frac{\alpha_i}{|\alpha_i|} \right) \right] \quad (35)$$

モデル II

$$\frac{\partial \alpha_i}{\partial \bar{\epsilon}^p} = \sqrt{\frac{2}{3}} \left[\zeta_i r_i \left(\frac{\frac{\partial f}{\partial s}}{\left| \frac{\partial f}{\partial s} \right|} - \left(\frac{\alpha_i}{r_i} \right)^{m_i+1} \langle \Lambda \rangle \frac{\alpha_i}{|\alpha_i|} \right) \right] \quad (36)$$

ただし、

$$\Lambda = \frac{\frac{\partial f}{\partial s}}{\left| \frac{\partial f}{\partial s} \right|} \frac{\alpha_i}{|\alpha_i|} \quad (37)$$

また、参考まで、32), 34) 式に現われる硬化項は以下のようになる。

モデル I

$$\sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} = \Phi \sum_{i=1}^M \zeta_i r_i (1 - H(f_i) \langle \Lambda \rangle \langle \Lambda \rangle) \quad (38)$$

モデル II

$$\sqrt{\frac{2}{3}} H_c \left(\frac{\partial f}{\partial \sigma} \right)^T \{\eta\} = \Phi \sum_{i=1}^M \zeta_i r_i \left(1 - \left(\frac{\alpha_i}{r_i} \right)^{m_i+1} \langle \Lambda \rangle \langle \Lambda \rangle \right) \quad (39)$$

右辺第 2 項で Λ が 2 乗になっているのが、本モデルの特徴である。

・抗応力の進展則

抗応力の進展則 28) 式は、このモデルでは第 1 項がないために 16) で表される。

$$d\kappa = \sigma_y(T_0) \frac{d\Phi}{dT} dT \quad (16)$$

$$\dot{H}_i = 0 \quad (40)$$

以上、材料パラメータの温度依存性を考慮した大野-王モデルの構成式を示した。

1.3 任意の構成モデル組込み機能を用いた大野-王モデル

今回の大野-王モデルは任意の構成モデル組込み機能を用いている。従って、このモデルを使用する場合は、予め大野-王モデルを定義したユーザーサブルーチンをリンクして、FINAS のロードモジュールを作成する必要がある。ただし、一旦これらのサブルーチンが組込まれてしまえば、後は従来のFINAS を利用するのと同様、新たに作成したロードモジュールによる実行が可能となる（1.3.4.2 参照）。

また、今回作成したユーザーサブルーチンはFINAS で使用している全ての応力場（表.1 参照、ただし $kfield=10$ は除く）に対応するように作られている。よって、任意の構成モデル組込み機能を使用することのできる全ての要素において大野-王モデルが利用可能である。ただし、一部のシェル要素（QFLA4R, QFLA4S, QFLA4RT 等）は、任意の構成モデル組込み機能が整備されていないために、このモデルを使用できない。

ここでは、任意の構成モデル組込み機能を用いた大野-王モデルについて、内部状態変数の定義、ユーザーサブルーチン XMAT1, XMAT2、ユーザーサブルーチン XREAD による材料パラメータの入力、使用方法について述べる。

表 1 FINAS の応力場

$kfield$	応力場	成分数	応力成分	座標系	備考
1	単軸応力場	1	σ_x	要素	
2	平面応力場 (~D2要素)	3	$\sigma_x, \sigma_y, \tau_{xy}$	全体	2次元要素
	平面応力場		$\sigma_x, \sigma_y, \tau_{xy}$	要素	3次元要素
3	3次元応力場	6	$\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$	全体	
4	軸対称応力場	4	$\sigma_r, \sigma_z, \sigma_\theta, \tau_{rz}$	全体	
5	平面ひずみ応力場	4	$\sigma_x, \sigma_y, \sigma_z, \tau_{xy}$	全体	
6	梁の応力場	3	$\sigma_x, \tau_{xy}, \tau_{zx}$	要素	
7	シェル応力場	5	$\sigma_x, \sigma_y, \tau_{xy}, \tau_{yz}, \tau_{zx}$	要素	
8	薄肉軸対称シェル応力場	2	σ_s, σ_θ	要素	
9	ELBOW3Rの応力場	3	$\sigma_l, \sigma_s, \tau_{sn}$	要素	
10	CONT2の応力場	-	-	-	[E ^e]を作らない
11	軸対称シェル応力場	3	$\sigma_s, \sigma_\theta, \tau_{s\theta}$	要素	

1.3.1. 内部状態変数の定義

任意の構成モデル組込み機能では、各構成モデルに特有な内部状態変数（塑性ひずみ、背応力、抗応力等）について、必要な大きさを宣言して用いることになっているが、この数はベクトル型、スカラー型、フラッグ型、各々9つまでに制限されている。これに対して、文献⁵⁾に示されている材料パラメータ温度依存性を考慮した大野-王モデルは背応力だけで10成分を持つため、今回の組込みにおいてその拡張が必要になった。しかし、10個以上のベクトル型内部状態変数を定義しても、成分を記述するフィールドが5カラムに制限されているので、プロット図において10番目以上の変数を指定することができないという問題が生じた。これを回避するには成分名を変更する以外ないが、このような変更は使用者の混乱を招くことになりかねない。とはいえ、一般の解析では10番目以降の内部状態変数がプロットできなくても、大勢に影響はないと考えられるため、今回はプロットの部分の拡張を見送り、ベクトル型の内部状態変数の最大定義数を9から19に拡張した。これによりベクトル型の内部状態変数の項数として10以上の値を定義しても解析は正常に行われるが、10以上の項についてのプロット出力ができない。またこの制限により、使用説明書等の記述はあくまで最大9個までとし、10以上の値を定義した場合、警告メッセージ（`*** USER WARNING MESSAGE 2871, USER VECTOR TYPE STATUS VARIABLES COUNT EXCEED 9`）が出力される。

今回のモデルは多くの背応力項を持つ。従って内部状態変数の領域を節約するために、熱ひずみ、弾性ひずみ等、他の状態変数から計算可能なものは、できるだけ内部状態変数からはずし、必要最小限のみの定義とした。また、背応力については温度依存性のない、基準温度での背応力 α_i を内部状態変数に選び、各温度での背応力 α^* が必要な時はその都度、温度依存性を表す関数 ϕ を乗じている。

大野-王モデルの組込みのために用いた内部状態変数を表.2.1~2.3に示す。背応力の項数は可変となっているが、その最大数は $nb = 10$ に制限されている。

表 2.1 ベクトル型内部状態変数 ($nvct = nb + 1$)

番号	定式化の記号	FINASの変数名	サブルーチンの変数名	内 容
1	ϵ^p	USTV(1,1)	EPSP (1)	塑性ひずみ
2	α_1	USTV(1,2)	ALI (1,1)	背応力第一成分
3	α_2	USTV(1,3)	ALI (1,2)	背応力第二成分
:	:	:	:	:
nb	α_{nb-1}	USTV(1, nb)	ALI (1, $nb-1$)	背応力第 $nb-1$ 成分
$nb+1$	α_{nb}	USTV(1, $nb+1$)	ALI (1, nb)	背応力第 nb 成分

表 2.2 スカラー型内部状態変数 ($nscr = 2$)

番号	定式化の記号	FINASの変数名	サブルーチンの変数名	内 容
1	ϵ^p	USTS(1)	EPB	相当塑性ひずみ
2	ϕ	USTS(2)	PHAI	温度依存の関数 ϕ

表 2.3 フラッグ型内部状態変数 ($nflg = 1$)

番号	定式化の記号	FINASの変数名	サブルーチンの変数名	内 容
1		IFLG(1)	IELPL	弾塑性状態フラッグ (弾性 : =0, 塑性 : =1)

※ $nvct$, $nscr$, $nflg$ は MATERIAL データの USER MODEL カードで指定する。 nb は背応力の項数で END FINAS 以降の USER MATERIAL カードのオプションキーで指定する (デフォルト:10, 最大:10)。

1.3.2. XMAT1, XMAT2 について

大野-王モデルの組込みに用いたユーザーサブルーチン XMAT1, XMAT2 のフローチャートを図.1, 2 に示す。XMAT1, XMAT2 および次節で説明する XREAD は多くのサブルーチンを呼び出すが、一覧を表.3 に示す。これらのサブルーチンの名前は FINAS 側ルーチンとの重複を避けるために、全て Y で始まる。また YF で始まるものは他のモデルでも共通に使えるもの、YM で始まるものはマトリックス演算ルーチン、YOW で始まるものは大野-王モデルに特有なルーチン、YX で始まるものが XREAD から呼ばれるルーチンとしている。さらにサブルーチン PEXIT, PAGE は FINAS 側のルーチンであり、各々データ異常時の計算中断とプリント出力における改ページ制御を行う。

今回の組込みにおいて、大野-王モデルは動的回復の臨界状態においては 2) 式をかなり精度よく満足しないと、解が発散するという傾向が見られた。これは 4) 式の未定定数を定めるために 5) 式のような、2) 式の臨界状態を満足すると仮定して適応条件を導いているためと考えられる。このためステップの途中で臨界状態が発生するときは、サブステップ法によるステップ分割を行い、各々の領域を別々に評価している。この手順はステップ分割が 1 回とは限らないために、再帰的な呼び出しが必要となる。ただし FORTRAN-77 の規格ではこれが許されず、今回はサブルーチン YOWBP1 ~ YOWBP8 を使った疑似的な再帰呼び出しアルゴリズムを採用している。

サブルーチンXMAT1のフローチャート

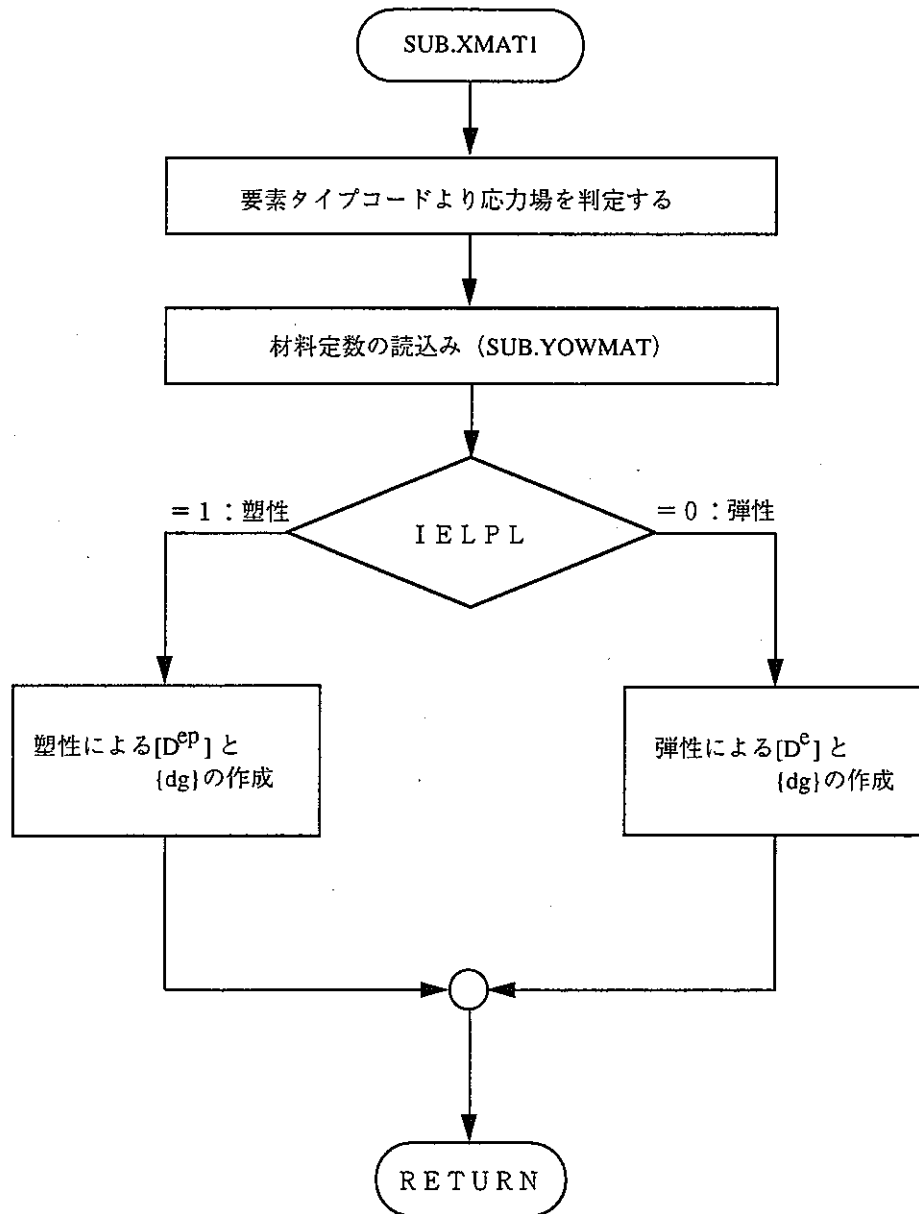


図.1 サブルーチン XMAT1 のフローチャート

サブルーチンXMAT2のフローチャート

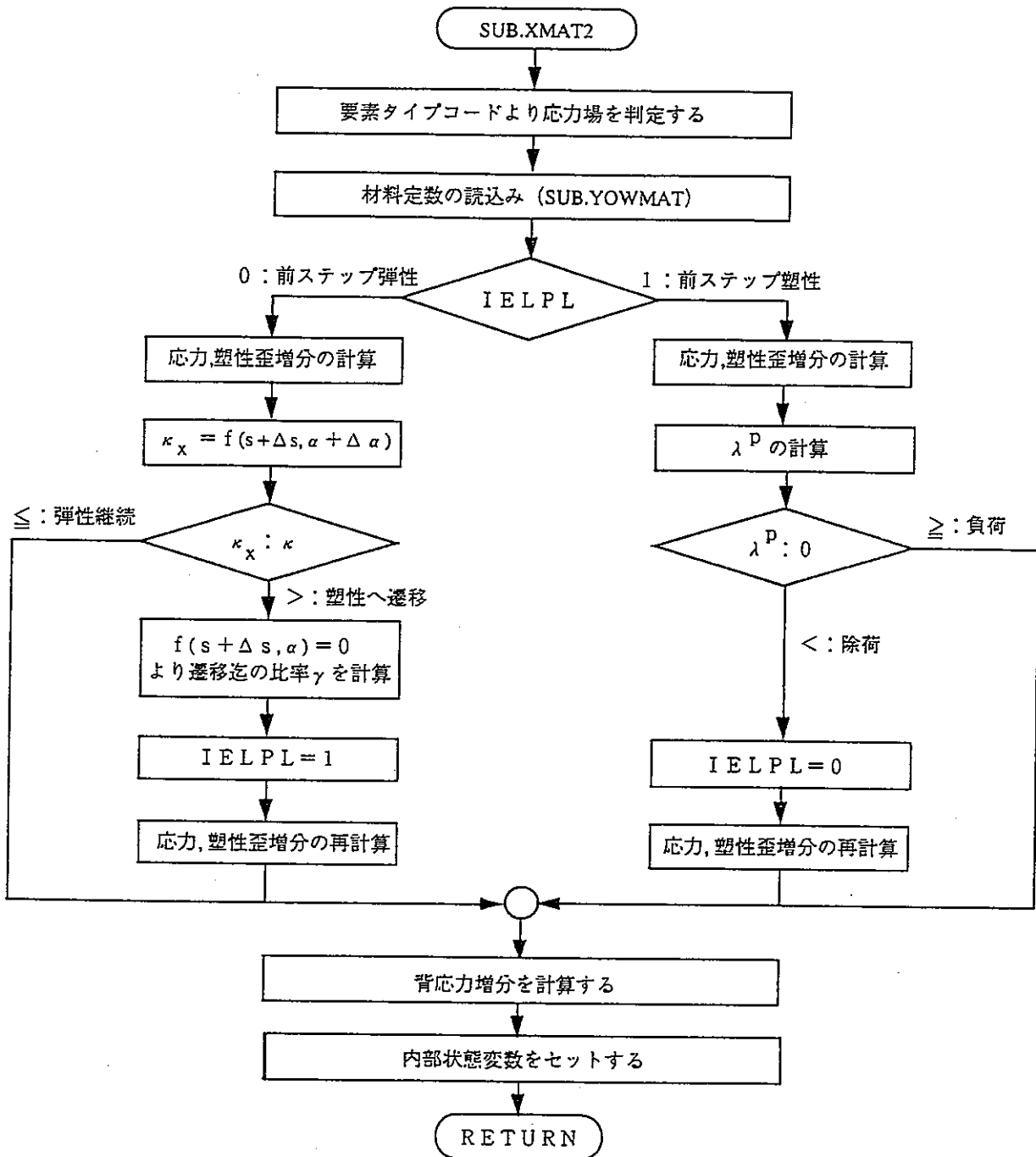


図.2 サブルーチン XMAT2 のフローチャート

表. 3 大野一王モデルを定義するサブルーチン

ルーチン名	機 能
XMAT1	[D], {dg}を定義する。
XMAT2	状態変数の増分を定義する。
XREAD	USER MATERIAL データの読み込み。
YFDE	[D ^e]マトリクスの計算。
YFDEI	[D ^e] ⁻¹ マトリクスの計算。
YFDEP	[D ^{ep}]マトリクスの計算。
YFDKDT	抗応力の温度微分項の計算。
YFDSE	弾性の場合の応力増分の計算。
YFDSEP	塑性の場合の応力増分の計算。
YFEPSE	応力より弾性ひずみを計算。
YFEPSP	塑性ひずみを計算。
YFGAM	弾性から塑性へ遷移するときの比率を計算。
YFGE	弾性の場合の{dg}をセット。
YFGEP	塑性の場合の{dg}をセット。
YFIELD	要素タイプコードより応力場を識別。
YFLOW2	{dF/dσ}を計算。
YFNRMV	ベクトル表記をテンソル表記に変換し、正規化する。
YFPBAR	相当塑性ひずみの計算。
YFSB2	相当応力の計算。
YFVEXP	多直線型データを補間。
YFXLAM	λ ^p の計算。
YMAB	マトリクスとベクトルの積。
YMBB	ベクトルとベクトルの内積。

表. 3 大野一王モデルを定義するサブルーチン(つづき)

ルーチン名	機 能
YOWBPi	ステップが臨界状態をまたぐときの状態変数の計算 ($i=1\sim 8$)。
YOWBPV	ステップが臨界状態をまたぐかを判定。
YOWBPX	ステップが臨界状態をまたぐかを判定 (YOWBPV の上位ルーチン)。
YOWDA	背応力増分の計算。
YOWDAT	背応力の温度に対する変化率の計算。
YOWEPV	内部状態変数の計算。
YOWFI	動的回復の臨界状態を表す f_i 関数。
YOWHD	塑性硬化係数の計算。
YOWMAT	材料パラメータの取りだし。
YXINT5	5 カラムの整数型入力データを識別。
YXREAD	USER MATERIAL データの読み込みルーチン。
YXRHED	USER MATERIAL データのエコーバックのタイトル表示。
DBMATN	ベクトル型内部状態変数の最大定義数の拡張。

なお、表3に示したサブルーチンのソースリストを章末に添付する。

1.3.3. XREAD による材料パラメータの入力

1.3.3.1 サブプログラム XREAD の仕様

XREAD は任意の構成モデル組込み機能等において、使用者が定義した材料パラメータを読み込むためのサブプログラムである。このサブルーチンは FINAS の実行時に 1 度だけ呼ばれ、入力データの END FINAS カード以降に書かれたデータを読み込み、適当なコモンブロックにセットするように組まれる。これにより、任意の構成モデルを定義する XMAT1, XMAT2 では、コモンブロックからこれらの材料パラメータを受け取ることができる。END FINAS カード以降の入力データは FINAS の実行に影響しないため、XREAD を用いることにより FINAS のデータとは全く独立にユーザーサブルーチン XMAT1, XMAT2 に材料パラメータを与えることができる。

XREAD による材料パラメータの入力は任意の構成モデル組込み機能に限定されたものではないが、他のユーザーサブルーチンと比較して使用方法が特殊なため、当面一般の利用者には非公開とする。

(1) サブプログラム XREAD のパラメータ

```

SUBROUTINE XREAD (LU5)
IMPLICIT REAL*8 (A-H, O-Z)
      ↓ 使用者がコーディング
RETURN
END

```

- ・ 入力パラメータ

LU5 : 材料パラメータを読み込むための外部装置識別子
FINAS の入力データを指し、XREAD が呼ばれる時には END FINAS カードの次のカードにファイルが位置づけられている。

- ・ 出力パラメータ

仮引数としては存在しないが、適当な共通ブロック名を定義することができる。
共通ブロック名の最初の文字は Y であること。

(2) サブプログラム XREAD の使用例

XMAT1, XMAT2で使用する材料パラメータとしてヤング率 (E=21000) とポアソン比 (POI=0.3) を END FINAS 以降の入力データとして読み込む例である。サブプログラム XREAD は次のように作成する。

```

SUBROUTINE XREAD (LU5)
  IMPLICIT REAL*8 (A-H, O-Z)
  COMMON /YMAT/ E, POI
  READ(LU5, '(2F10.0)') E, POI
  RETURN
END

```

FINAS の入力データには END FINAS 以降に次のような材料パラメータを記述しておく。

```

FINAS
  ↓ FINAS の入力データ
END FINAS
21000.0  0.3

```

読み込んだ材料パラメータは共通ブロック /YMAT/ を XMAT1, XMAT2 の中で参照する。例えば、

```

SUBROUTINE XMAT1 ( IEXN, IECOD, .....
  IMPLICIT REAL*8 (A-H, O-Z)
  DIMENSION GEOM (*), .....
  COMMON /YMAT/ E, POI
  ↓ 使用者がコーディング
RETURN
END

```

1.3.3.2 大野-王モデルにおける XREAD

前節では簡単な例により XREAD の概念と使用例を示したが、今回の大野-王モデルは、使用者の使いやすさの点からより複雑な構成となっている。なお、今回考慮した点は以下の通りである。

- ・入力データにおいて、使用者の定義する材料データの先頭を示すヘッダとして USER MATERIAL を持つこと。
- ・使用者の定義する材料データはできるだけ MATERIAL カードと同様の書式を持つこと。
- ・使用者の定義する材料データが入力データとしてプリント出力されること。

従って、使用者の定義する材料データは END FINAS 以降の USER MATERIAL カード以降に定義される。これらのデータは材料データ識別番号とオプションキーを持ち、定数と温度に関するデータテーブル書式について、MATERIAL データと同様の書式を持つ。また、USER MATERIAL カード以降のデータも他の FINAS 入力データと同様にプリント出力（エコー出力）がなされる。これら、USER MATERIAL データの詳細な入力書式は次節に示すが、ここでは XREAD, XMAT1, XMAT2 で使われている共通ブロック /YBXMAT/ について、パラメータ文で定義されている定数を表 .4.1 に、変数を表 .4.2 に、材料種類の番号と材料定数の対応を表.4.3 に示す。

これらの表に示したように大野-王モデルの最大27個のパラメータは各々温度の関数として共通ブロック /YBMAT/ に記憶されている。この方法は材料の種類番号 J によって一般化されているため、他の材料についても拡張可能であるが、材料データのチェック、出力等に関して細かな制御が必要となる。また、USER MATERIAL カードで使用する材料データ識別番号の数の最大や温度に関するデータテーブルの最大値を変更するには XREAD, XMAT1, XMAT2 におけるパラメータ文を変更すればよい。

表.4.1 /YBXMAT/ の定数

変数名	現在の値	意 味
NTBLPA	21	温度に関するデータテーブルの最大値
NMATPA	27	材料特性の種類最大値
NMIDPA	4	USER MATERIAL データで定義する材料データ識別番号の最大
NKEYPA	10	オプションキーの数

表.4.2 /YBXMAT/ の変数

変数名	意 味
MID (K)	K 番目に入力された材料データ識別番号
KEY (N,K)	K 番目の材料データにおけるN 番目のオプションキーの値
NTBL(J,K)	K 番目の材料データにおけるJ 番目材料のテーブルの数 (=1:定数)
TTBL(I,J,K)	K 番目の材料データにおけるJ 番目材料の I 番目の温度値
VTBL(I,J,K)	K 番目の材料データにおけるJ 番目材料の I 番目の材料パラメータ

表.4.3 /YBXMAT/ における J と材料特性の対応

J	材料特性名	意 味	J	材料特性名	意 味
1	E	ヤング率 E	15	ZETA08	モデル定数 ζ_8
2	NU	ポアソン比 ν	16	ZETA09	モデル定数 ζ_9
3	ALPHA	瞬間線膨張係数 α	17	ZETA10	モデル定数 ζ_{10}
4	SIG0	基準温度での降伏応力 $\sigma_y(T_0)$	18	R01	モデル定数 r_1
5	M	モデル定数 $m (=m_i)$	19	R02	モデル定数 r_2
6	PHAI A	ϕ 関数の定数 1 ※	20	R03	モデル定数 r_3
7	PHAI B	ϕ 関数の定数 2 ※	21	R04	モデル定数 r_4
8	ZETA01	モデル定数 ζ_1	22	R05	モデル定数 r_5
9	ZETA02	モデル定数 ζ_2	23	R06	モデル定数 r_6
10	ZETA03	モデル定数 ζ_3	24	R07	モデル定数 r_7
11	ZETA04	モデル定数 ζ_4	25	R08	モデル定数 r_8
12	ZETA05	モデル定数 ζ_5	26	R09	モデル定数 r_9
13	ZETA06	モデル定数 ζ_6	27	R10	モデル定数 r_{10}
14	ZETA07	モデル定数 ζ_7			

※ $\phi(T) = PHAI A * \exp (PHAI B (T+273))$

1.3.4. 使用方法

1.3.4.1 入力データ

ユーザーサブルーチンによる任意の構成モデルにより組込んだ大野-王モデルを解析に使用するためには、入力データの MATERIAL カードで使用宣言と内部状態変数を定義し、さらに材料パラメータをEND FINAS 以降に書かれた USER MATERIAL データにセットする必要がある。

・ MATERIAL データ

	10	20	30	40	50	60	70	80
A	MATERIAL							
B3	imat	USER MODEL	nvct	nscr	nflg			

任意の構成モデル組込み機能で定義したモデルを使う方法は使用説明書 4.32 に示した通りである。FINAS のモデルデータのAカードはMATERIAL データの始まりを指示し、B3 カードで組込んだ大野-王モデルの使用を宣言する。ここに *nvct*, *nscr*, *nflg* は各々モデルで使用するベクトル型、スカラー型、フラッグ型状態変数の数を指定する。大野-王モデルでは *nscr* は 2、*nflg* が 1 に固定されているが、*nvct* のみは可変で $nb+1$ (は背応力の数であり USER MATERIAL データの *k1* で指定) を与える。また、必要に応じて複数のモデル定数を用いる場合はさらにB3 カードを続ける。ただし、USER MATERIAL データには対応する材料定数識別番号を持つデータが存在しなければならない。

カードの種類	カラム	タイプ	変数	内 容	無記入の場合
A	1~8	英 字	—	MATERIALと記入	*
B3	1~5	整 数	<i>imat</i>	材料データ識別番号	*
	11~20	英 字	—	USER MODEL と記入	*
	21~25	正整数	<i>nvct</i>	ベクトル型内部状態変数の数 ($nvc = nb+1$)	*
	26~30	正整数	<i>nscr</i>	スカラー型内部状態変数の数 ($nscr = 2$)	*
	31~35	正整数	<i>nflg</i>	フラッグ型内部状態変数の数 ($nflg = 1$)	*

・ USER MATERIAL データ

	10	20	30	40	50	60	70	80						
A	USER MATERIAL													
B	imat		k1	k2	k3	k4	k5	k6	k7	k8	k9	k10		

USER MATERIAL データは、XREAD で定義した大野-王モデルの材料データを入力する。このデータはFINAS 本体の入力データに影響を与えないためにEND FINAS カードの後に置かれる。USER MATERIAL データは、まず A カードにより使用者の定義する材料データの宣言を行う。この後に材料データ識別番号とオプションキーを定義する B カードが続く。このカードで指定する材料データ識別番号は MATERIAL データの B₃ カードで指定した識別番号と同じでなければならない。また、オプションキーについては使用説明書 MATERIAL データの B₂ カードとは異なり、以下に示すように組込んだ構成モデルに特有な使い方になることに注意が必要である。k₁ は背応力の項数を表し、MATERIAL データの B₃ カードで指定した *nvct-1* であること。また、k₂ はステップが背応力の臨界状態をまたぐときサブステップ法を採用してステップ分割を行うか否かの選択であるが、一般にはよほど細かなステップを採用しない限り、安定した解析結果を得るためにサブステップ法を選択すべきである。k₃ は大野-王モデルにおける材料パラメータ温度依存を考慮するか否かの選択であり、k₃=1 とすると常に温度依存関数 $\phi(T)=1$ を仮定する。ただし、この場合にもヤング率、ポアソン比、線膨張係数の温度依存性は考慮される。k₄ はモデルの選択で、k₄=0 にするとモデル-I（背応力の発展式に Heaviside 関数を用いる場合：式 9）, k₄=1 ではモデル-II（背応力の発展式にべき乗項を用いる場合：式 10）である。モデル II を選択した場合には材料パラメータとして *m*, PHAIA, PHAIB が必要となる。また k₅ 以降については未使用であり、解析に影響を与えないため空白でよい。これらオプションキーのデフォルト値は、オプションを何も記入しないときに材料パラメータの温度依存性を考慮した大野-王モデル⁵⁾になるように設定されている。

実際の材料データの入力については、FINAS の MATERIAL データの書式 1（定数）と書式 5（温度に関するデータテーブル）が使用できる。従って一般的な使用者が定義する材料データの構成は以下のようになる。

また \$ で始まる行は FINAS の入力データに同様コメント行として使用することができる。

A, B, C,, G₁, G₂,
B, G₁, G₂,

カラムの種類	カラム	タイプ	変数	内 容	無記入の場合
A.	1~13	英 字	—	USER MATERIALと記入	*
B	1~5	整 数	<i>imat</i>	材料データ識別番号 (MATERIALデータに対応)	*
	21~25	正整数	<i>k₁</i>	背応力の項数 (<i>nb</i> または <i>M</i> , $1 \leq k_1 \leq 10$)	10
	26~30	正整数	<i>k₂</i>	サブステップ法の選択 (<i>k₂</i> = 0: サブステップ法を使う, <i>k₂</i> = 1: 使わない)	0
	31~35	正整数	<i>k₃</i>	温度依存性の選択 (<i>k₃</i> = 0: 温度依存性を考慮する, <i>k₃</i> = 1: 考慮しない)	0
	36~40	正整数	<i>k₄</i>	モデルの選択 (<i>k₄</i> = 0: モデル-I, <i>k₄</i> = 1: モデル-II)	0
	41~45	正整数	<i>k₅</i>	未使用	0
	46~50	正整数	<i>k₆</i>	未使用	0
	51~55	正整数	<i>k₇</i>	未使用	0
	56~60	正整数	<i>k₈</i>	未使用	0
	61~65	正整数	<i>k₉</i>	未使用	0
	66~70	正整数	<i>k₁₀</i>	未使用	0

書式1 (定数)

	10	20	30	40	50	60	70	80
C		kind	v					

書式1は材料特性が温度、塑性ひずみ等に依存しない定数の場合に用いる。このカードではヤング率、ポアソン比、線膨張係数と大野-王モデルの全ての材料パラメータを入力することができる。またモデル定数 ζ_i, r_i は必要な項数までを入力すればよく、モデル-1では m を、温度依存性を考慮しない場合には $\alpha, PHAIA, PHAIB$ を省略できる。

カードの種類	カラム	タイプ	変数	内 容	無記入の場合
C	11~20	英数字	kind	材料特性の種類 (表4.3 参照) E : ヤング率 E NU : ポアソン比 v ALPHA : 瞬間線膨張係数 α SIG0 : 基準温度での降伏応力 $\sigma_y(T_0)$ M : モデル定数 $m (=m_i)$ PHAIA : ϕ 関数の定数1 PHAIB : ϕ 関数の定数1 $\phi(T) = PHAIA * \exp (PHAIB / (T+273))$ ZETA01 : モデル定数 ζ_1 ZETA02 : モデル定数 ζ_2 ZETA03 : モデル定数 ζ_3 ZETA04 : モデル定数 ζ_4 ZETA05 : モデル定数 ζ_5 ZETA06 : モデル定数 ζ_6 ZETA07 : モデル定数 ζ_7 ZETA08 : モデル定数 ζ_8 ZETA09 : モデル定数 ζ_9 ZETA10 : モデル定数 ζ_{10} R01 : モデル定数 r_1 R02 : モデル定数 r_2 R03 : モデル定数 r_3 R04 : モデル定数 r_4 R05 : モデル定数 r_5 R06 : モデル定数 r_6 R07 : モデル定数 r_7 R08 : モデル定数 r_8 R09 : モデル定数 r_9 R10 : モデル定数 r_{10}	*
	21~30	実数	v	材料特性の値	*

書式 5 (温度に関するデータテーブル)

	10	20	30	40	50	60	70	80
G1		<i>kind</i>	F5	<i>m</i>				
G2		T_1	ν_1	T_2	ν_2	T_3	ν_3	
		T_4	ν_4→	T_m	ν_m	

書式 5 は材料特性を温度に関するデータテーブルで与える場合に使用し、各温度とその温度での材料特性値を温度の小さい順に組で入力する。計算で使われる材料特性はこのデータテーブルから線形に内挿または外挿される。この書式ではヤング率、ポアソン比、線膨張係数を与えることができる。しかし、大野-王モデルにおける他の材料パラメータの温度依存性は ϕ 関数により定義されているため、これらの入力に書式 5 を使うことはできない。

カードの種類	カラム	タイプ	変数	内 容	無記入の場合
G1	11~20	英数字	<i>kind</i>	材料特性の種類 (表4.3 参照) E : ヤング率 E NU : ポアソン比 ν ALPHA : 瞬間線膨張係数 α	*
	21~25	英 字	—	F5 と記入	*
	26~30	正整数	<i>m</i>	温度と材料特性の組数 ($1 \leq m \leq 21$)	*
G2	11~20	実 数	T_1	最初のデータの組の温度	0.0
	21~30	実 数	ν_1	最初のデータの組の材料特性値	0.0

	51~60	実 数	T_3	3番目のデータの組の温度	0.0
61~70	実 数	ν_3	3番目のデータの組の材料特性値 m が 4 以上のときは次のカードの11カラム以降に同様の書式で記入する。	0.0	

1.3.4.2 UNIX ワークステーションによる使用方法

ここでは、現在の計算機環境として代表的な UNIX ワークステーションにおける使用方法を示す。ただし、大野-王モデルを定義するユーザーサブルーチンとして OWMDL.f と FINAS の入力データとして DATA.fns の名前を持つファイルが準備されているとする。

・初回の計算

初めての計算では、大野-王モデルを定義するユーザーサブルーチンを組込む必要がある。以下にサブルーチンを組込みと計算の実行を行う例を示す。なお、この実行によりロードモジュールとして @DATA.exe が作られる。

```
finas -dat DATA.fns -usub OWMDL.f
```

・以降の計算

既に構成モデルが組み込まれたロードモジュールが存在するときは、これを利用することができる。以下にロードモジュールを指定した実行方法を示す。

```
finas -dat DATA.fns -load @DATA.exe
```

1.4 機能の検証

任意の構成モデルにより FINAS に組込んだ大野-王モデルの機能を検証するため、大野らと同じ解析を行い同様の解が得られることを確認した。この検証はFINAS に大野-王モデルを定義する XMAT1, XMAT2 を直接組込んだ場合と、これらのユーザーサブルーチンのドライバープログラムである ADMODEL に組込んだ場合の両者について行った。検証では前者が単軸問題を扱い、後者は、その使いやすさから多軸問題を対象とした。ただし、単軸問題では両者とも同様の解が得られることを確認している。

1.4.1 FINASによる検証例題

FINAS による大野-王モデルの検証は、文献²⁾に基づく材料パラメータの温度依存性を考慮しないモデルIおよびモデルII (VP-209) と、文献⁵⁾による材料パラメータ温度依存性を考慮したモデルI (VP-210) の各々について単軸引張り問題の解析を行った。ここでは以下の2つの例題についてその検証結果を示す。

VP-209 大野-王モデルを用いた単軸引張り問題

VP-210 材料パラメータ温度依存性を考慮した大野-王モデルによる単軸引張り問題

VP-209 大野-王モデルを用いた単軸引張り問題

(1) 目的

任意の構成モデル組込み機能によりユーザーサブルーチンでFINAS に組込まれた大野-王モデル（モデルI とモデルII）を用いて単軸引張り問題の解析を行う。この結果を大野らによって得られた解と比較することにより同モデル解析機能の検証とする。

なお本問題における大野-王モデルは文献¹⁾の定式化に、材料定数と解析例は文献²⁾に従った。

(2) 問題の設定

a) 解析モデル

・構造モデル

断面積1.0mm²、長さ10mmの棒

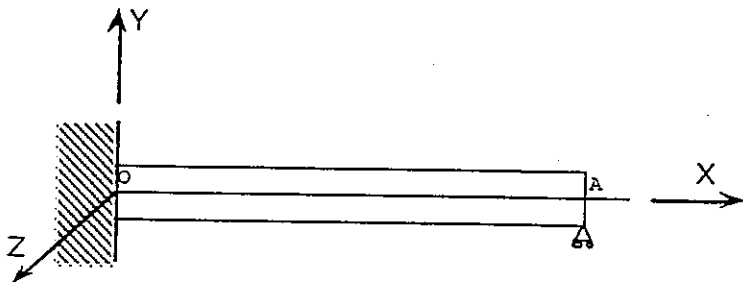


図 209-1 構造モデル

・使用要素

ROD2 : 3次元棒要素

・有限要素モデル

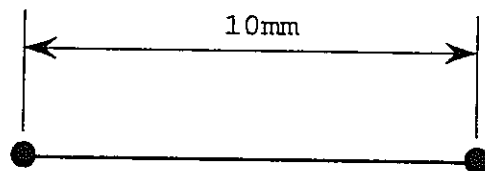


図 209-2 有限要素モデル

b)材料定数

・材料は改良9Cr-1Mo鋼(550℃)とし、文献²⁾より背応力の項数を $M=8$ と $M=4$ 、各々についてモデルIとモデルIIさら4種類のひずみ速度により合計16種類の解析を行う。また、初期降伏応力は文献²⁾の11)式において、塑性ひずみを0とした以下の式より得られ、 $A=5.43 \times 10^{-25}$, $b=8.8$ より以下のように定まる。解析に用いた材料定数を表209-1に示す。

$$\sigma_y = \left(\frac{\dot{\epsilon}}{A} \right)^{\frac{1}{b}}$$

- $\sigma_y = 260.916$ [MPa] ($\dot{\epsilon} = 0.1$ [%/sec])
- $\sigma_y = 241.153$ [MPa] ($\dot{\epsilon} = 0.05$ [%/sec])
- $\sigma_y = 185.633$ [MPa] ($\dot{\epsilon} = 0.005$ [%/sec])
- $\sigma_y = 142.896$ [MPa] ($\dot{\epsilon} = 0.0005$ [%/sec])

表 209-1 解析に用いた材料定数

M = 4		M = 8	
E	1.65×10^5 [MPa]	E	1.65×10^5 [MPa]
ν	0.33	ν	0.33
m_i	7 (モデルIIのみ)	m_i	7 (モデルIIのみ)
ζ_1	1111	ζ_1	2500
ζ_2	370	ζ_2	1111
ζ_3	200	ζ_3	556
ζ_4	125	ζ_4	370
r_1	115.0 [MPa]	ζ_5	263
r_2	57.4 [MPa]	ζ_6	199
r_3	24.3 [MPa]	ζ_7	159
r_4	13.3 [MPa]	ζ_8	125
		r_1	40.0 [MPa]
		r_2	50.0 [MPa]
		r_3	44.0 [MPa]
		r_4	27.0 [MPa]
		r_5	19.0 [MPa]
		r_6	13.5 [MPa]
		r_7	7.0 [MPa]
		r_8	9.4 [MPa]

c)境界条件

点Oにおいて $u_x = u_y = u_z = 0$

点Aにおいて $u_y = u_z = 0$

X方向に全ひずみが2%になるような規定変位を200ステップで与える。

(3) 解析結果

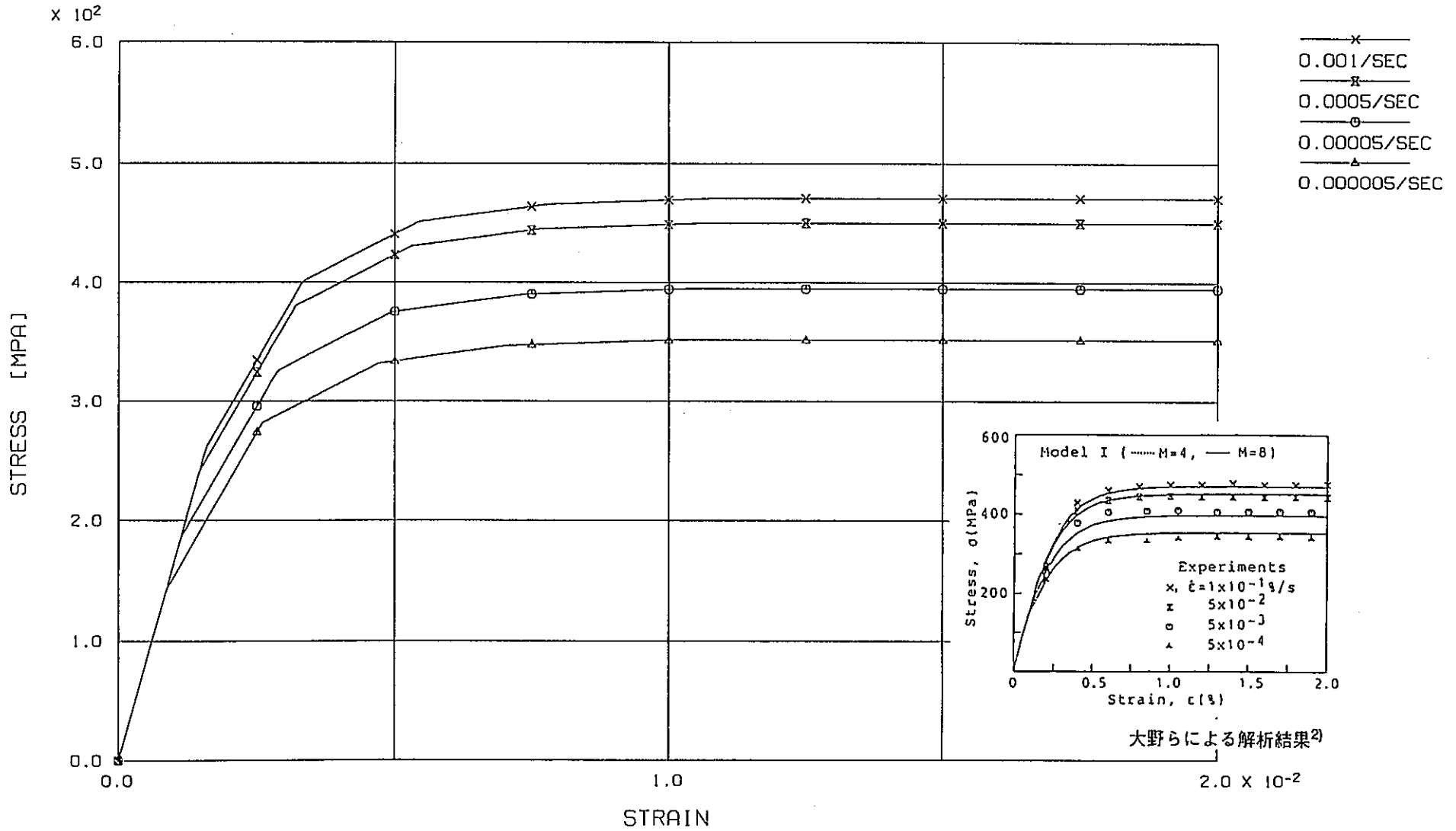
モデルI, モデルII 各々について $M=8$ と $M=4$ の場合の各ひずみ速度による解析結果を、大野らの結果と共に図 209-3,4,5,6 に示す。

(4) 結論

FINASと大野らによる解析結果はよく一致しており、FINASに組込んだ大野-王モデルが正常に動作することを確認した。

(参考文献)

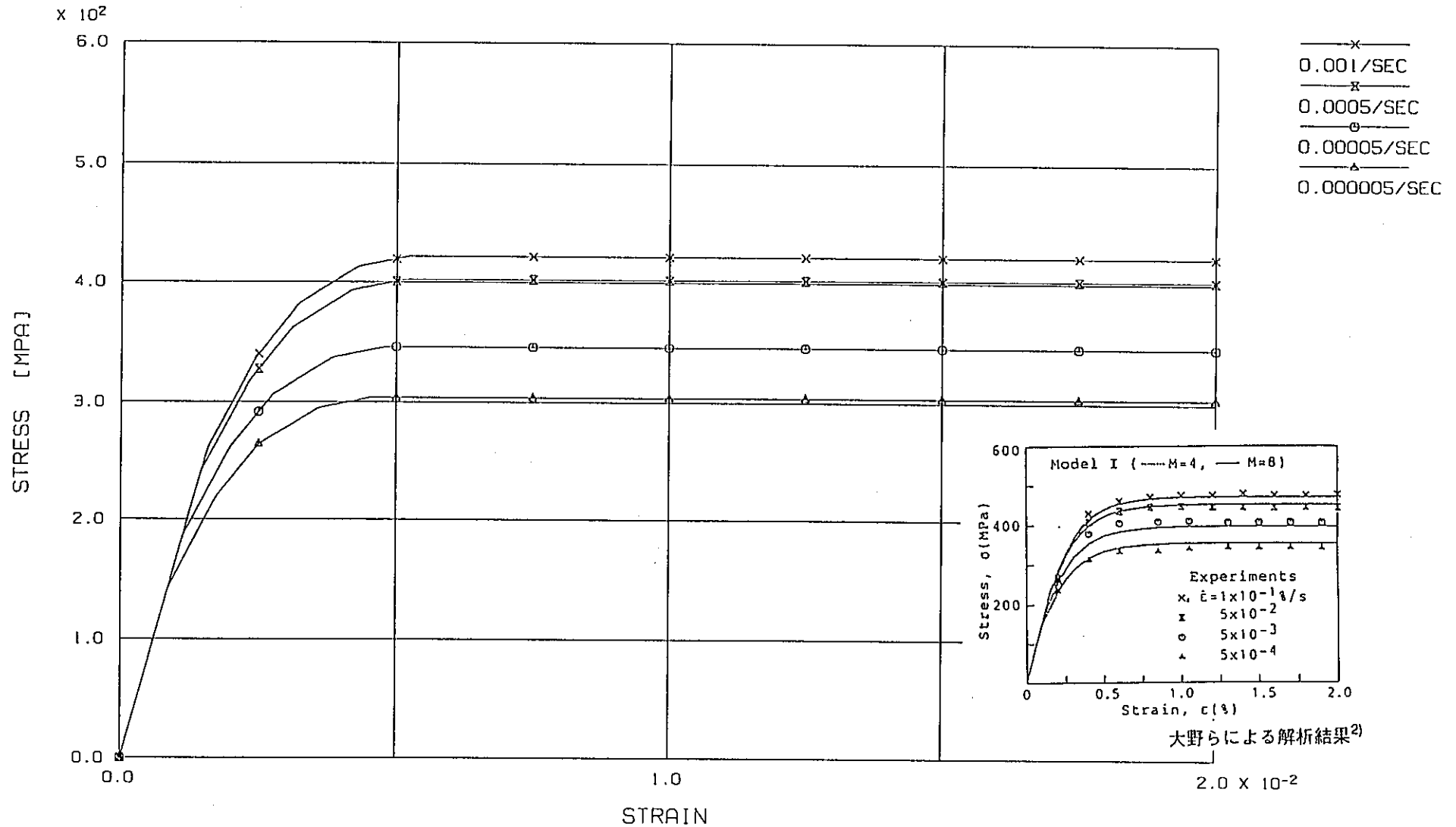
- 1) 大野信忠、王建定：動的回復の臨界状態を考慮した移動硬化則（第1報、定式化および基本的特徴）日本機械学会論文集（A編）58巻547号,488-（1992-3）
- 2) 大野信忠、王建定：動的回復の臨界状態を考慮した移動硬化則（第2報、実験結果に基づく検討）日本機械学会論文集（A編）58巻547号,496-（1992-3）



STRESS-STRAIN RELATION OF MOD.9CR-1M0 FIG-1(A) BY FINAS

MODEL-1 M=4

図 209-3 FINAS による解析結果 (モデルI, M=4)

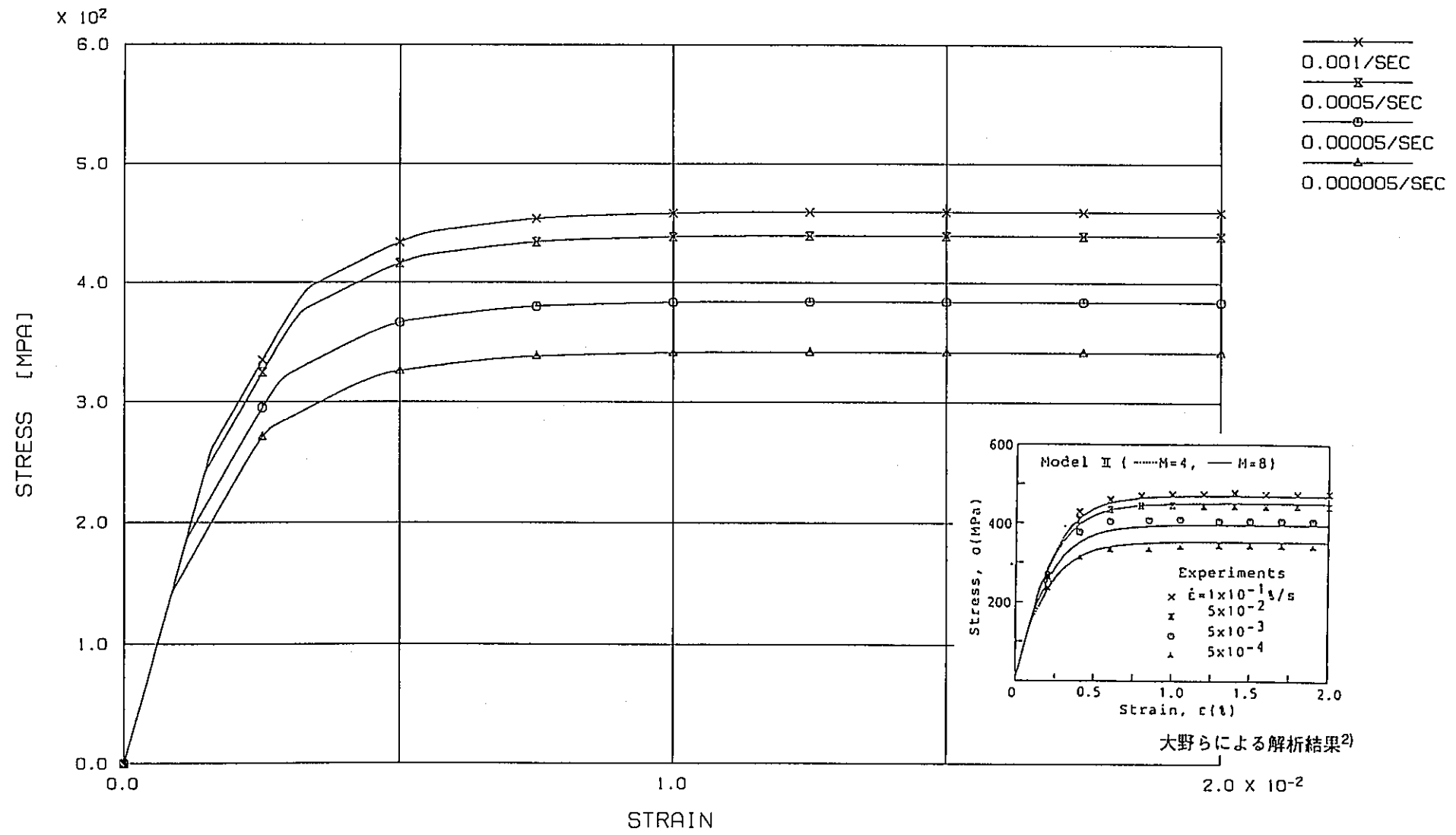


STRESS-STRAIN RELATION OF MOD.9CR-1M0 FIG-1(A) BY FINAS

MODEL-1 M=8

図 209-4 FINAS による解析結果 (モデルI, M=8)

FINAS STATIC ANALYSIS

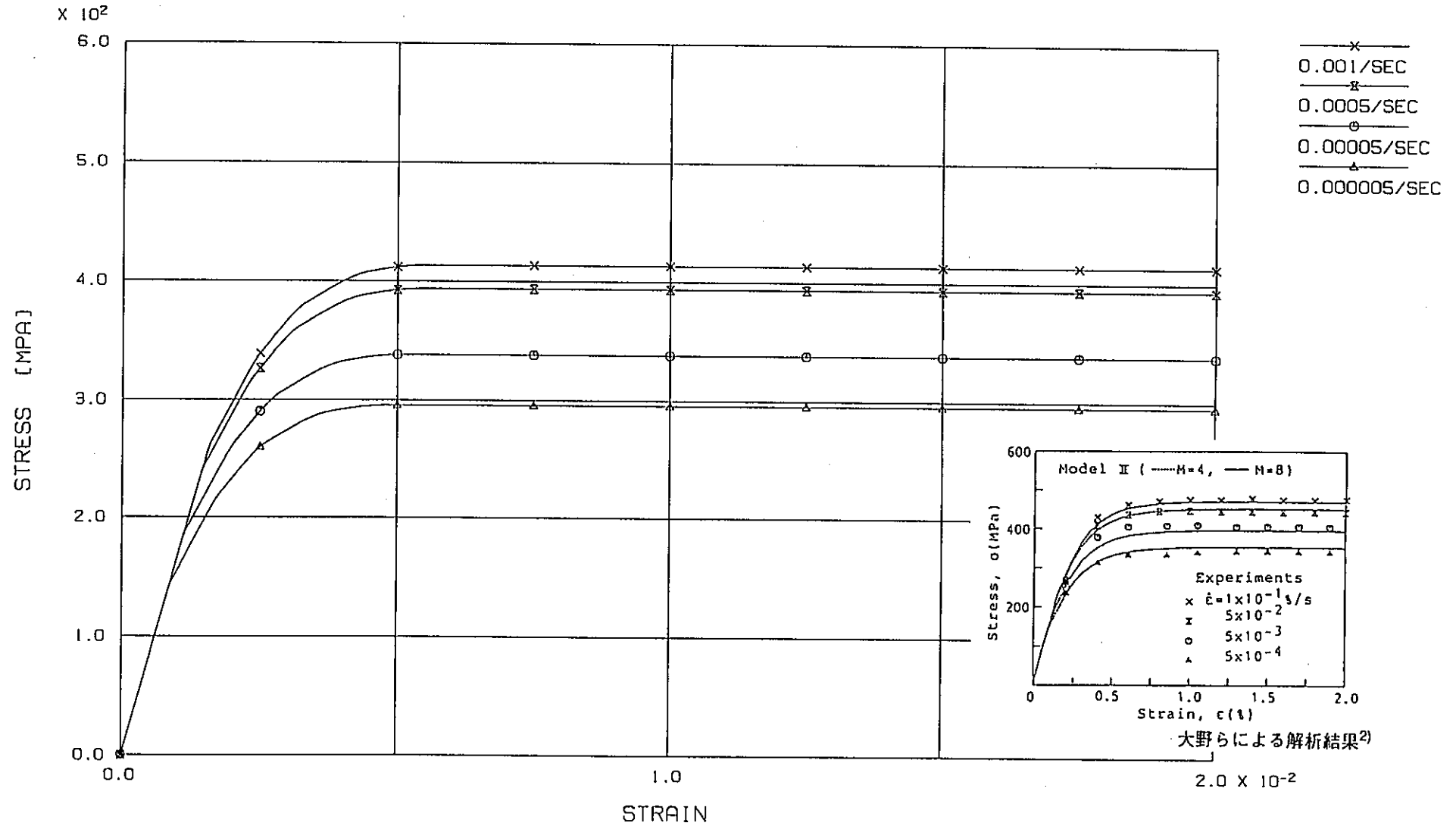


STRESS-STRAIN RELATION OF MOD.9CR-1M0 FIG-1(B) BY FINAS

MODEL-2 M=4

図 209-5 FINAS による解析結果 (モデルII, M=4)

FINAS STATIC ANALYSIS



STRESS-STRAIN RELATION OF MOD.9CR-1M0 FIG-1(B) BY FINAS

MODEL-2 M=8

図 209-6 FINAS による解析結果 (モデルII, M=8)

VP-210 材料パラメータ温度依存性を考慮した
大野-王モデルによる単軸引張り問題

(1) 目的

任意の構成モデル組込み機能によりユーザーサブルーチンでFINAS に組込まれた材料パラメータ温度依存性を考慮した大野-王モデルを用いて各温度における単軸引張り問題の解析を行う。この結果を大野らによって得られた解と比較することにより同モデル解析機能の検証とする。なお本問題における大野-王モデルは文献¹⁾の定式化と解析例を用いている。

(2) 問題の設定

a) 解析モデル

・構造モデル

断面積1.0mm,長さ10mmの棒

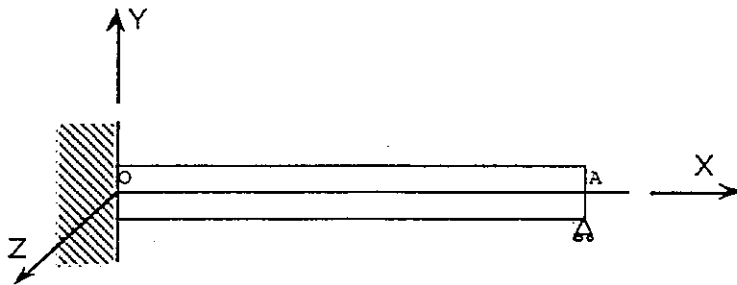


図 210-1 構造モデル

・使用要素

ROD2 : 3次元棒要素

・有限要素モデル

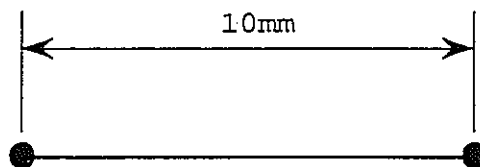


図 210-2 有限要素モデル

b)材料定数

・材料定数は文献¹⁾より316FR鋼を想定し、背応力の項数を $M=10$ としたモデルIである。問題は文献¹⁾に従い、室温 (20℃)、300℃、650℃の各温度における一定温度の単軸引張りとした。弾性の材料定数ヤング率 (E)、ポアソン比 (ν)、線膨張係数 (α) は FINAS 使用説明書表 4.5-5よりSUS316鋼の値を用いた。また、材料パラメータの温度依存性については、基準温度を 300℃ ($T_0=300^\circ\text{C}$) とし、以下のような ϕ 関数を仮定している。

$$\phi(T) = 0.583 \exp\left(\frac{307}{T+273}\right) \quad T [^\circ\text{C}]$$

その他の大野-王モデルにおける材料定数の値を以下に示す。

$\sigma_y(T_0) =$	115 [MPa]		
$r_1 =$	5.50 [MPa]	$\zeta_1 =$	2.00×10^4
$r_2 =$	1.81 [MPa]	$\zeta_2 =$	1.00×10^4
$r_3 =$	2.40 [MPa]	$\zeta_3 =$	5.00×10^3
$r_4 =$	3.44 [MPa]	$\zeta_4 =$	2.50×10^3
$r_5 =$	5.00 [MPa]	$\zeta_5 =$	1.11×10^3
$r_6 =$	6.39 [MPa]	$\zeta_6 =$	5.26×10^2
$r_7 =$	9.61 [MPa]	$\zeta_7 =$	2.56×10^2
$r_8 =$	15.9 [MPa]	$\zeta_8 =$	1.00×10^2
$r_9 =$	20.2 [MPa]	$\zeta_9 =$	4.00×10
$r_{10} =$	68.8 [MPa]	$\zeta_{10} =$	2.00×10

c)境界条件

点Oにおいて $u_x = u_y = u_z = 0$

点Aにおいて $u_y = u_z = 0$

X方向に全ひずみが1.5%になるような規定変位を150ステップで与える。

また、初期温度を室温 (20℃)、300℃、650℃に設定した3つの解析を行った。

(3) 解析結果

FINASによる解析結果を大野らの結果と共に図210-3に示す。また参考までにFINASの入力データを次ページに示す。この例では温度の異なる3つの解析を1回で解いている。

(4) 結論

FINASと大野らによる解析結果はよく一致しており、FINASに組込んだ大野-王モデルが正常に動作することを確認した。

(参考文献)

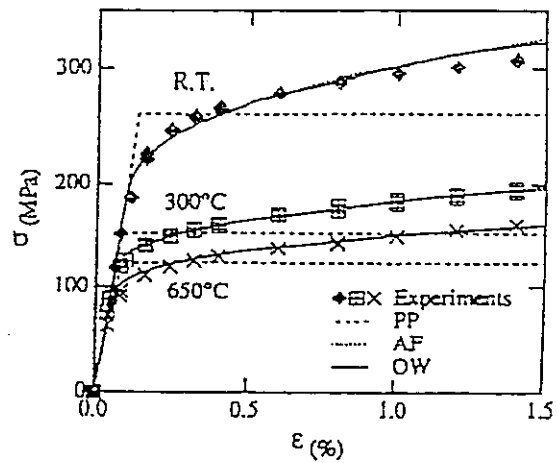
- 1) 小林峰雄、大野信忠：軸方向温度分布変動を受ける円筒の熱ラチェット解析（実験との比較）日本機械学会東海支部豊橋地区講演会講演論文集 305 No.953-2 (1995-7/7)

VP-210の入力データ

```

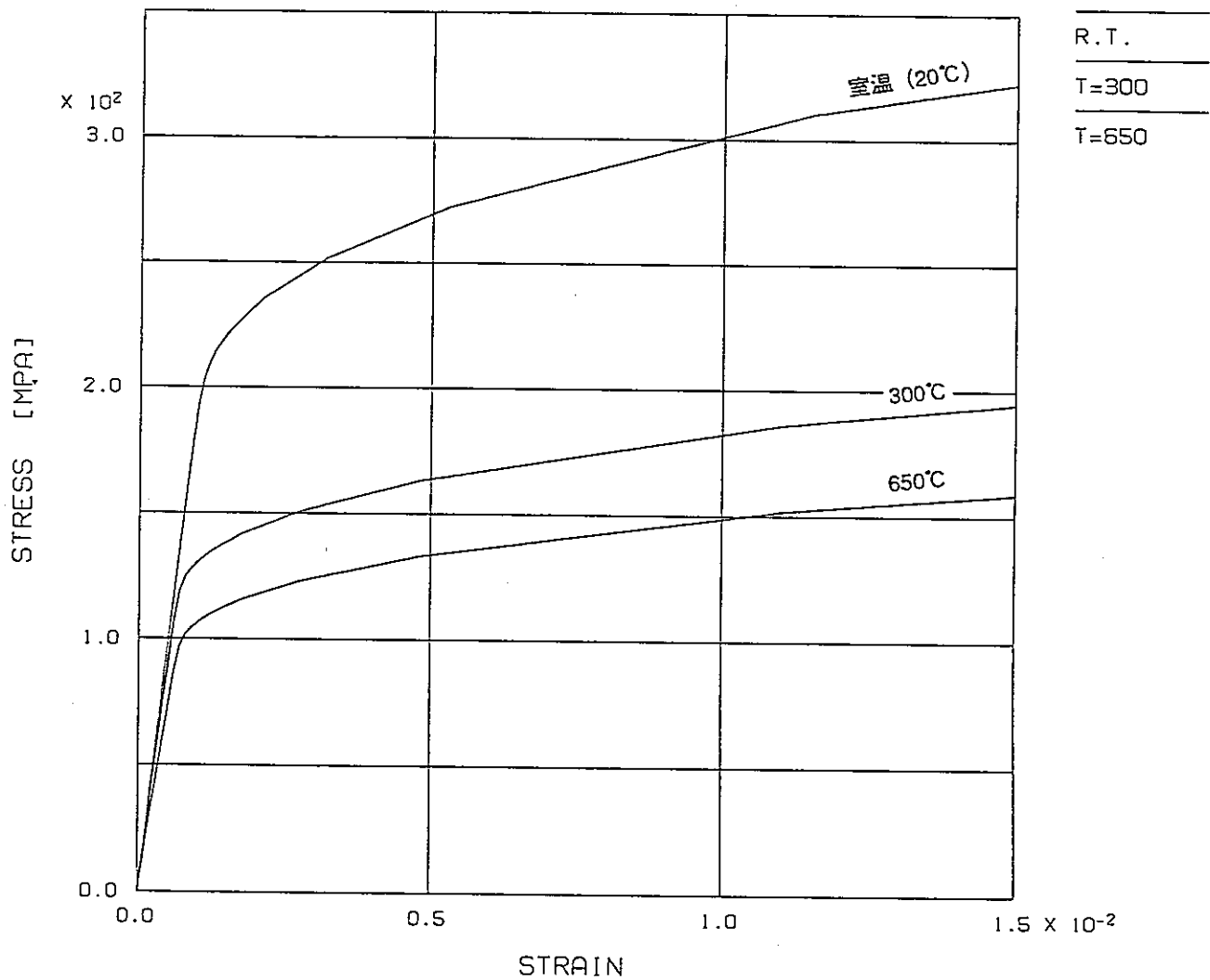
FINAS
TITLE
SUBTITLE
NODPRINT DES
STATIC TE
MODEL
MODE
$
11 0.0 0.0 0.0
12 10.0 0.0 0.0
21 0.0 0.0 0.0
22 10.0 0.0 0.0
31 0.0 0.0 0.0
32 10.0 0.0 0.0
ELEMENT TYPE
1 ROD2
CONNECTION
20 1 1 1 11 12
300 1 1 1 21 22
650 1 1 1 31 32
GEOMETRY
1 1.0
MATERIAL
1 USER MODEL 11 2 1
REFERENCE TEMP
11 12 1 20.0
21 22 1 300.0
31 32 1 650.0
BOUNDARY
1
11 12 1 2 3
21 22 1 2 3
31 32 1 2 3
11 1 1
21 1 1
31 1 1
12 1 10.0
22 1 10.0
32 1 10.0
HISTORY
INTERVAL 1
INTERVAL 150
INTERVAL 600K 1 0.0
INTERVAL 600K 1 0.015
END MODEL
OUTPUT
PRINT SELECT
ELM ALL
DISP NONE
REAC NONE
XYPLOT
OMIT 0
FRAME 210.0 210.0
PTITLE STRESS VERSUS STRAIN RELATION OF 316FR (FIG.1 BY FINAS)
XTITLE STRAIN
YTITLE STRESS [MPA]
YRANGE 0.0 350.0
YDIVISION 7 1
XRANGE 0.0 0.015
XDIVISION 3 0
CURVE 1 1 0 R.T.
CURVE 2 1 0 T=300
CURVE 3 1 0 T=650
PLOT3 EPS1 20 1STRS1 20 1EPS1 300 1STRS1 300 1
EPS1 650 1STRS1 650 1
END OUTPUT
END FINAS
$
.. USER MATERIAL DATA FOR OHNO-WANG MODEL
$
USER MATERIAL
$
.. MATERIAL DATA FOR OHNO-WANG MODEL (MODEL-1) SUS316FR (TEMP. DEP.)
$
$ .. E, NU, ALPHA BY SUS316 (FINAS USER'S MANUAL 4-82)
$
1
E FS 14
20.0 1.9404E5 50.0 1.9208E5 100.0 1.9012E5
150.0 1.8718E5 200.0 1.8424E5 250.0 1.8032E5
300.0 1.7640E5 350.0 1.7248E5 400.0 1.6856E5
450.0 1.6366E5 500.0 1.5876E5 550.0 1.5386E5
600.0 1.4896E5 650.0 1.4406E5
NU FS 14
20.0 0.266 50.0 0.268 100.0 0.272
150.0 0.275 200.0 0.279 250.0 0.283
300.0 0.287 350.0 0.291 400.0 0.295
450.0 0.298 500.0 0.302 550.0 0.306
600.0 0.310 650.0 0.314
ALPHA FS 14
20.0 15.15E-6 50.0 15.65E-6 100.0 16.48E-6
150.0 17.22E-6 200.0 17.85E-6 250.0 18.36E-6
300.0 18.79E-6 350.0 19.19E-6 400.0 19.57E-6
450.0 19.93E-6 500.0 20.28E-6 550.0 20.60E-6
600.0 20.87E-6 650.0 21.09E-6
S160 115.0
PHA1A 0.583
PHA1B 307.0
ZETA01 2.00E4
ZETA02 1.00E4
ZETA03 5.00E3
ZETA04 2.50E3
ZETA05 1.11E3
ZETA06 5.26E2
ZETA07 2.56E2
ZETA08 1.00E2
ZETA09 4.00E1
ZETA10 2.00E1
R01 5.50
R02 1.81
R03 2.40
R04 3.44
R05 5.00
R06 6.39
R07 9.61
R08 15.9
R09 20.2
R10 68.8

```



大野らによる解析結果¹⁾

FINAS STATIC ANALYSIS



STRESS VERSUS STRAIN RELATION OF 316FR (FIG.1 BY FINAS)

図 210-3 FINAS による解析結果

1.4.2 ADMODELによる検証

大野-王モデルはラチェット変形のシミュレーション精度に優れることに大きな特徴がある。従って、ここではラチェット問題について解析を行い、FINAS に組込まれた大野-王モデルの機能検証とする。ただし、多軸ラチェット問題にFINAS を用いることは煩雑なため、非弾性挙動ベンチマーク解析プログラム ADMODEL を使用した。ADMODEL は FINAS の任意の構成モデル組込み機能に用いた XMAT1, XMAT2 のドライバープログラムとして、これらのサブルーチンにより構成モデルを定義し、ベンチマーク解析を行うことができる。また、ADMODEL はひずみまたは応力を直接制御するため、簡単に多軸問題の解析を行うことができる。ここでは、ADMODEL を用いて行った多軸ラチェット変形と、非比例負荷問題として十字形ひずみ経路に対する応力応答の解析結果を示す。

・多軸ラチェット変形

文献²⁾に従って一定引張り応力に繰返しねじり変形を重畳した多軸ラチェット変形の解析を行った。これらはモデルI ($M=8$) とモデルII ($M=4, M=8$) の3種類のモデルについて、引張り応力と繰返しひずみ範囲を表5のように4種類に変化させ、計12ケースの解析を40サイクルまで行った。なお本計算は解析の途中で境界条件を変化させる必要がある。材料は改良9Cr-1Mo鋼(550℃)とし、初期降伏応力は文献²⁾の11)式において、塑性ひずみを0とした以下の式より得られ、 $A=5.43 \times 10^{-25}$, $b=8.8$ より以下のように定めた。解析に用いた材料定数を表6に示す。

$$\sigma_y = \left(\frac{\dot{\epsilon}}{A} \right)^{\frac{1}{b}}$$

$$\sigma_y = 241.153 \text{ [MPa]} \quad (\dot{\epsilon} = 0.05 \text{ [%/sec]})$$

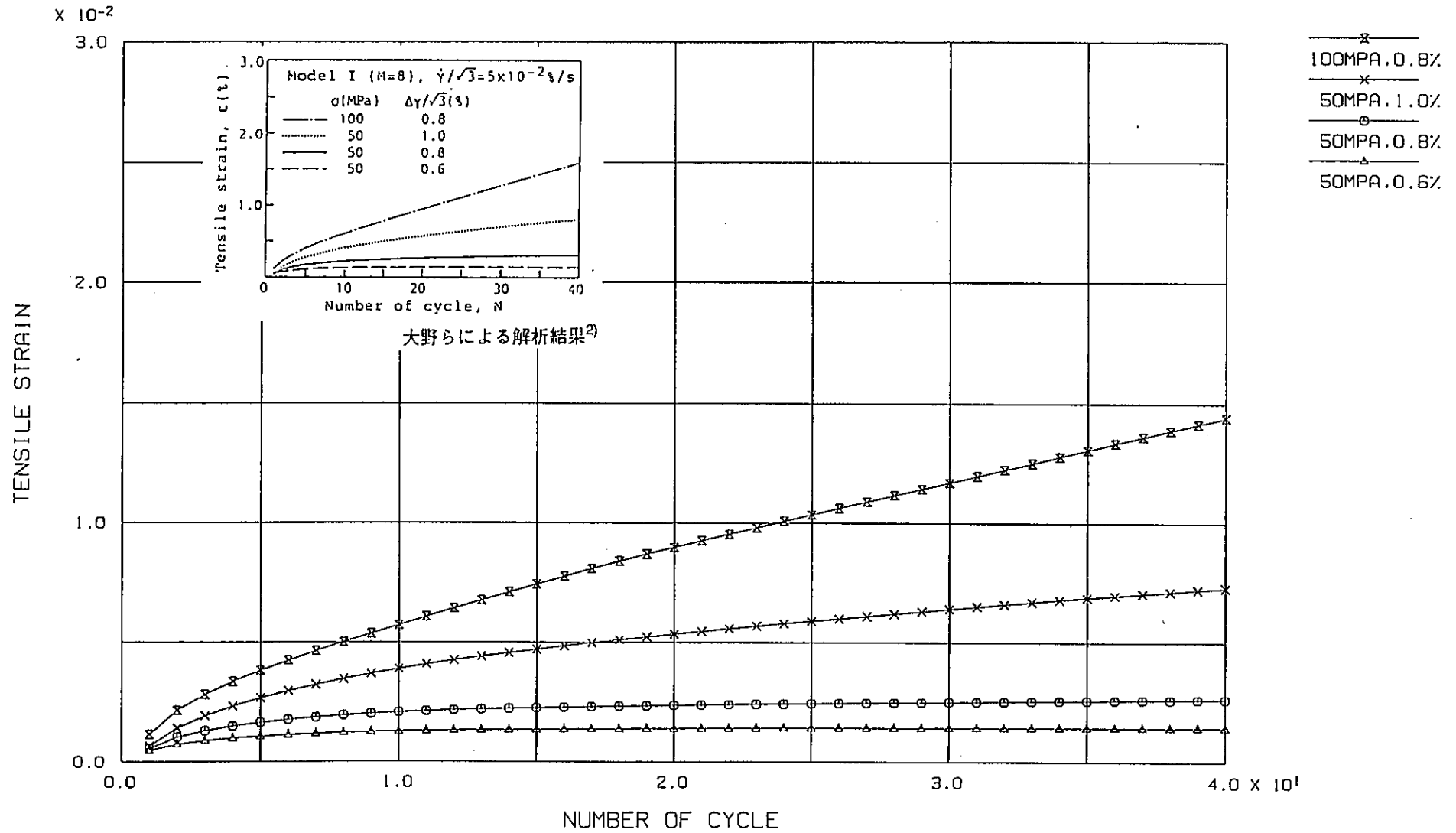
ADMODELにより得られたラチェットひずみを図3,4,5に示す。またモデルII ($M=8$)についてせん断ひずみと引張りひずみの履歴図を図6に示す。図中には大野らによる解析結果も表示しているが、よく一致するしており、多軸ラチェット問題においても本モデルが正常に動作することが確かめられた。

表 5 負荷条件

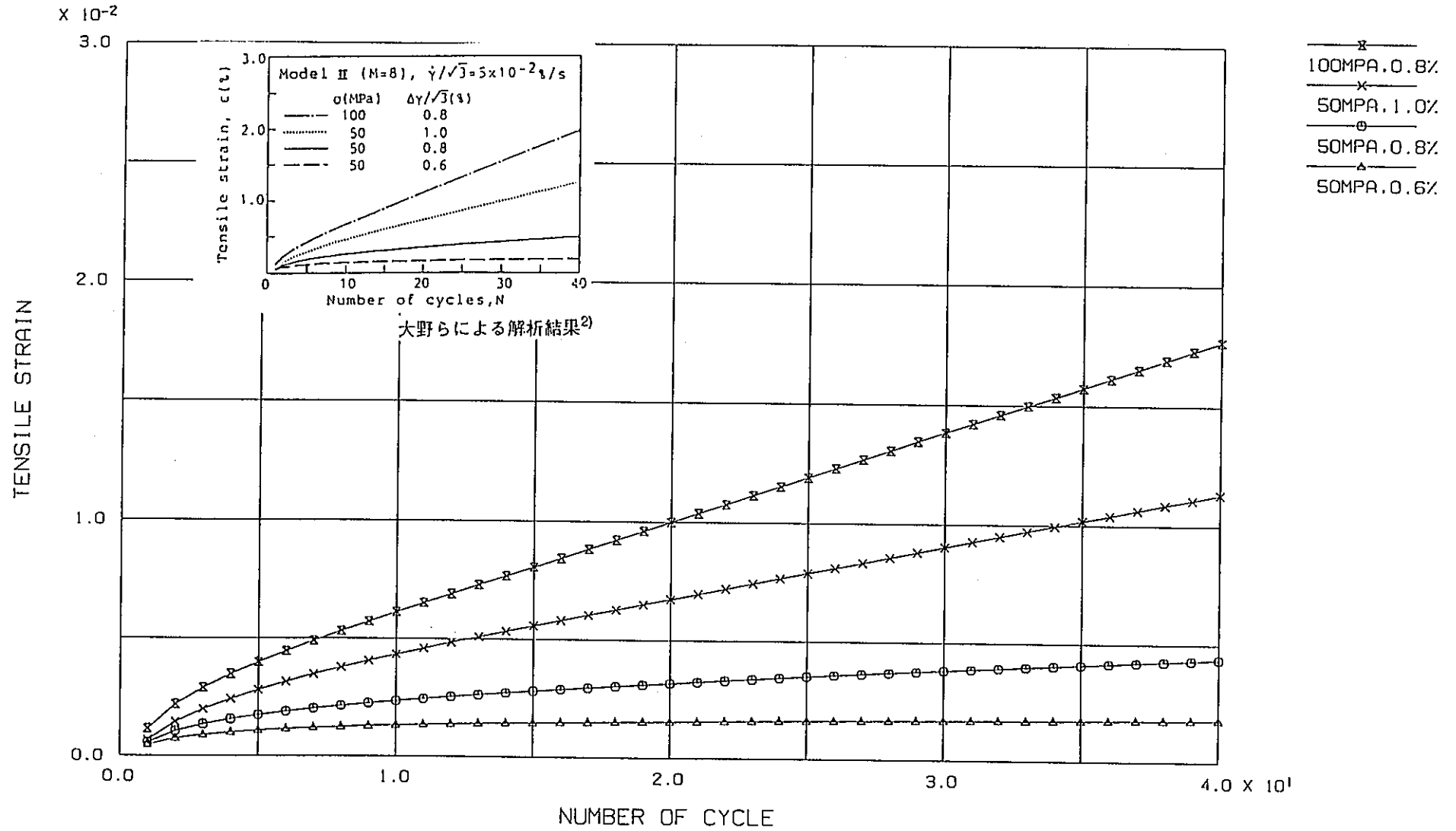
引張り応力 σ	ねじりひずみ範囲 $\Delta\gamma/\sqrt{3}$
100 [MPa]	0.8 [%]
50 [MPa]	1.0 [%]
50 [MPa]	0.8 [%]
50 [MPa]	0.6 [%]

表 6 解析に用いた材料定数

$M = 4$		$M = 8$	
E	1.65×10^5 [MPa]	E	1.65×10^5 [MPa]
ν	0.33	ν	0.33
		m_i	7 (モデルIIのみ)
ζ_1	1111	ζ_1	2500
ζ_2	370	ζ_2	1111
ζ_3	200	ζ_3	556
ζ_4	125	ζ_4	370
r_1	115.0 [MPa]	ζ_5	263
r_2	57.4 [MPa]	ζ_6	199
r_3	24.3 [MPa]	ζ_7	159
r_4	13.3 [MPa]	ζ_8	125
		r_1	40.0 [MPa]
		r_2	50.0 [MPa]
		r_3	44.0 [MPa]
		r_4	27.0 [MPa]
		r_5	19.0 [MPa]
		r_6	13.5 [MPa]
		r_7	7.0 [MPa]
		r_8	9.4 [MPa]

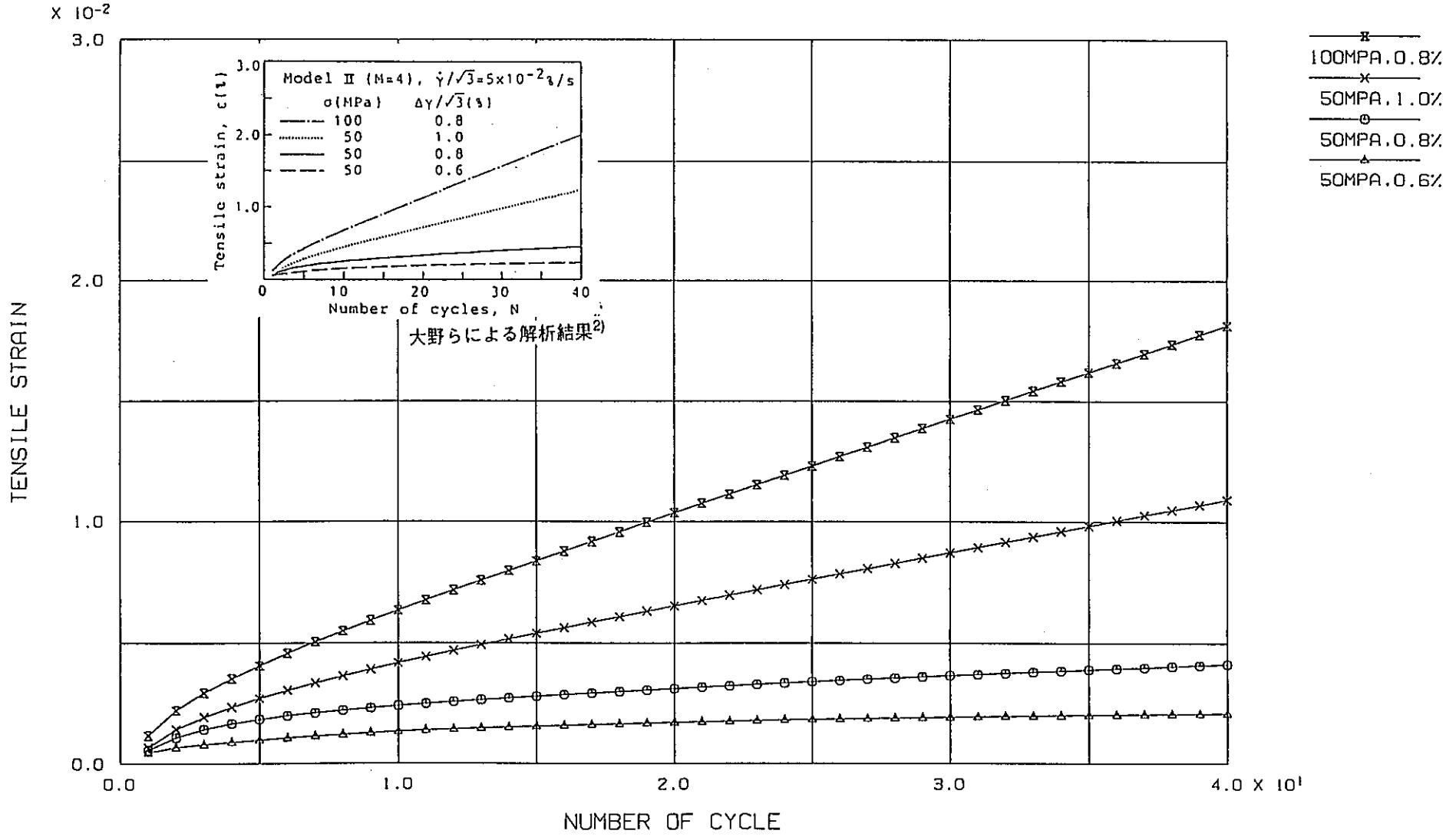


MULTI-AXIAL RATCHET DEFORMATION OF MOD.9CR-1MO (FIG.3-B) BY A MODEL
 MODEL-1 M=8
 図 3 ADMODEL による多軸ラチェット問題の解析結果 (モデルI, M=8)



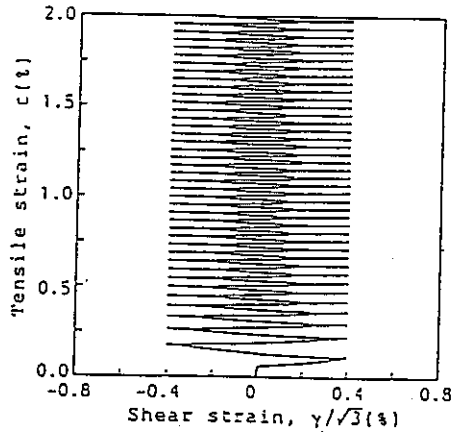
MULTI-AXIAL RATCHET DEFORMATION OF MOD.9CR-1MO (FIG.3-C) BY AMODEL MODEL-2 M=8

図 4 ADMODEL による多軸ラチェット問題の解析結果 (モデルII, M=8)



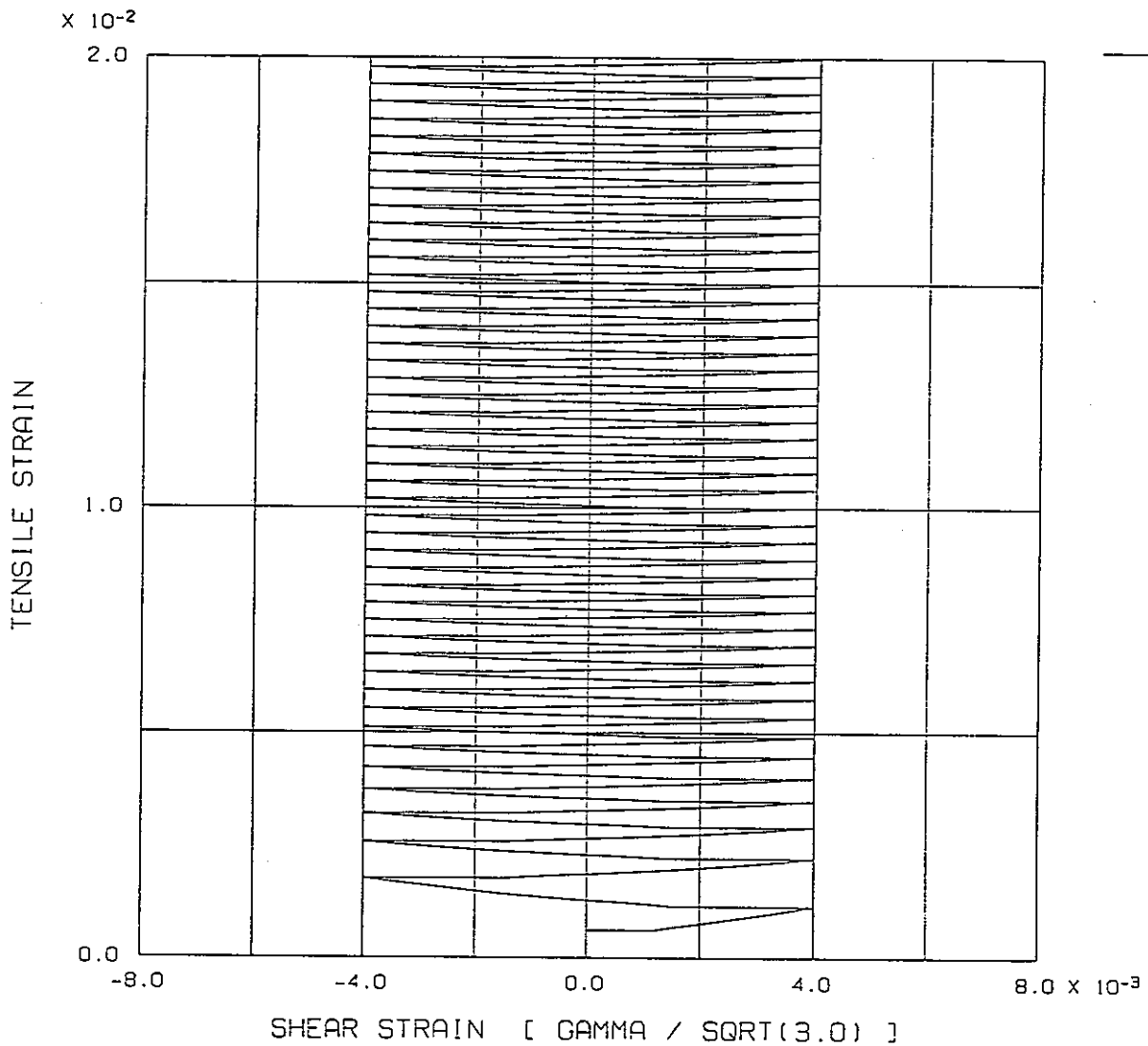
MULTI-AXIAL RATCHET DEFORMATION OF MOD.9CR-1MO (FIG.3-D) BY A MODEL MODEL-2 M=4

図 5 ADMODEL による多軸ラチェット問題の解析結果 (モデルII, M=4)



大野らによる解析結果²⁾

FINAS STATIC ANALYSIS



MULTI-AXIAL RATCHET DEFORMATION OF MOD.9CR-1MO (FIG.4-B) BY AMODEL MODEL-2 M=8

図 6 ADMODEL による多軸ラチェット問題の解析結果 (モデルII, M=8)

・非比例変形（十字形ひずみ経路）

文献²⁾に従いADMODELにより図7に示すような十字形ひずみ経路による非比例負荷での応力応答を計算した。これらはモデルI ($M=8$) とモデルII ($M=8$) の2種類のモデルについて行い、材料定数は多軸ラチェット変形と同様とした。

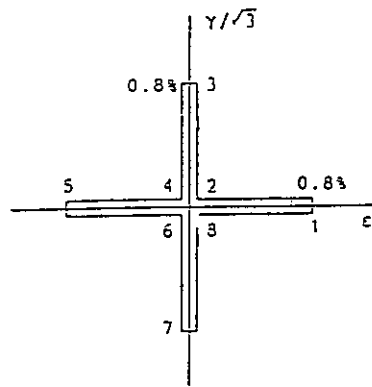
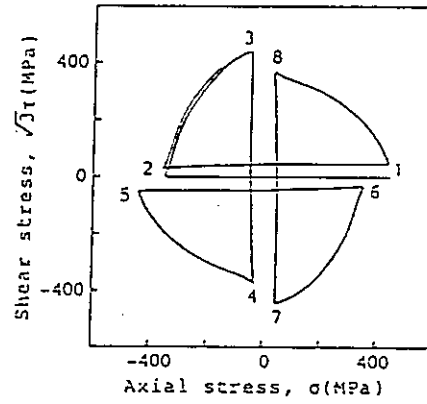


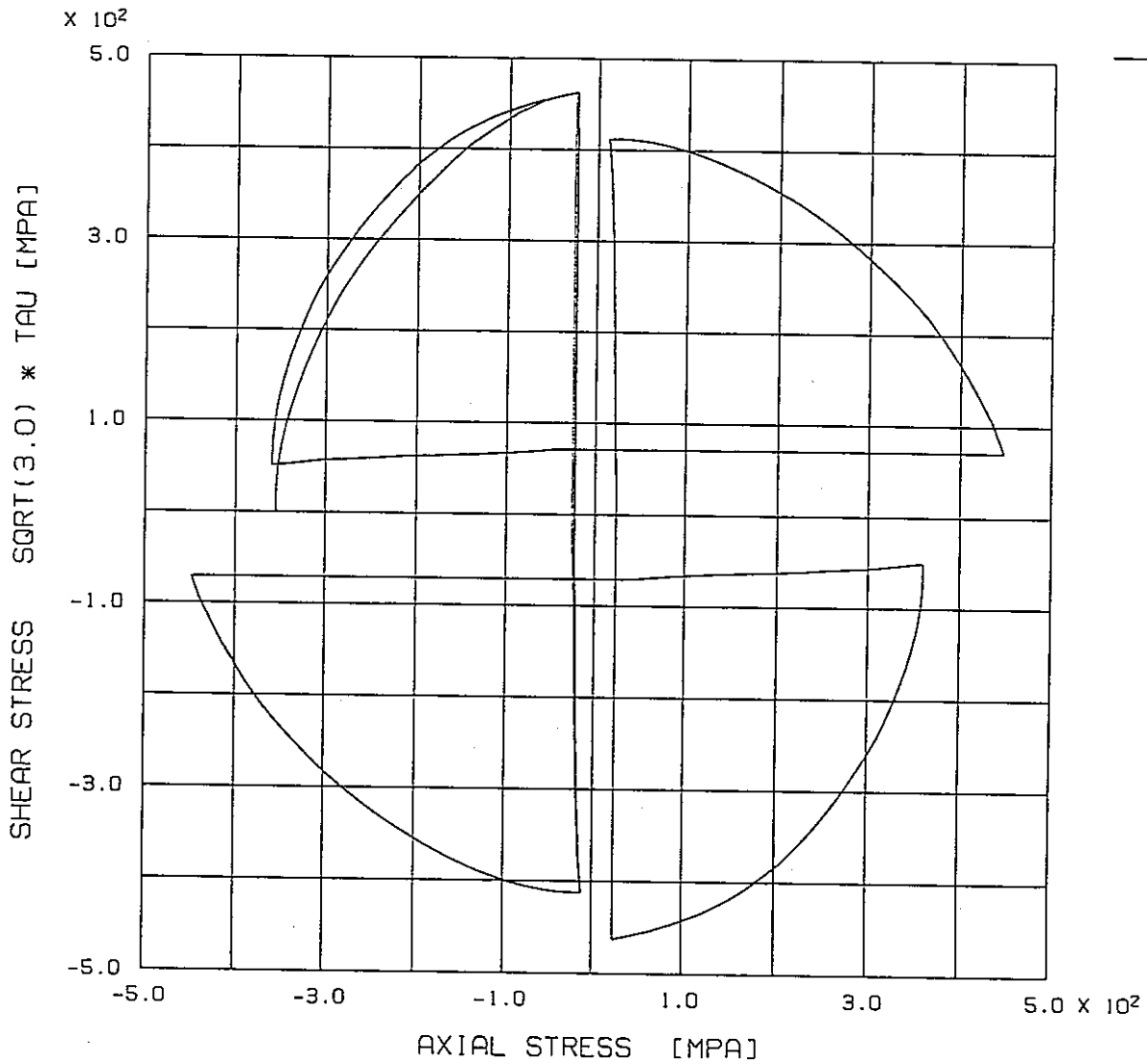
図7 十字形ひずみ経路

ADMODELにより得られた軸応力とせん断応力の履歴を図8,9に示す。図中には大野らによる解析結果も表示しているが、この結果とADMODELによる結果はよく一致しており、非比例負荷問題においても大野-王モデルが正常に動作することが確かめられた。



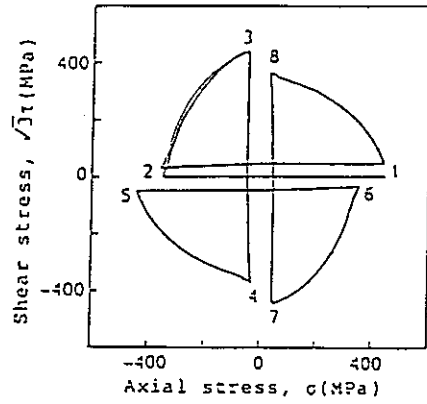
FINAS STATIC ANALYSIS

大野らによる解析結果²⁾



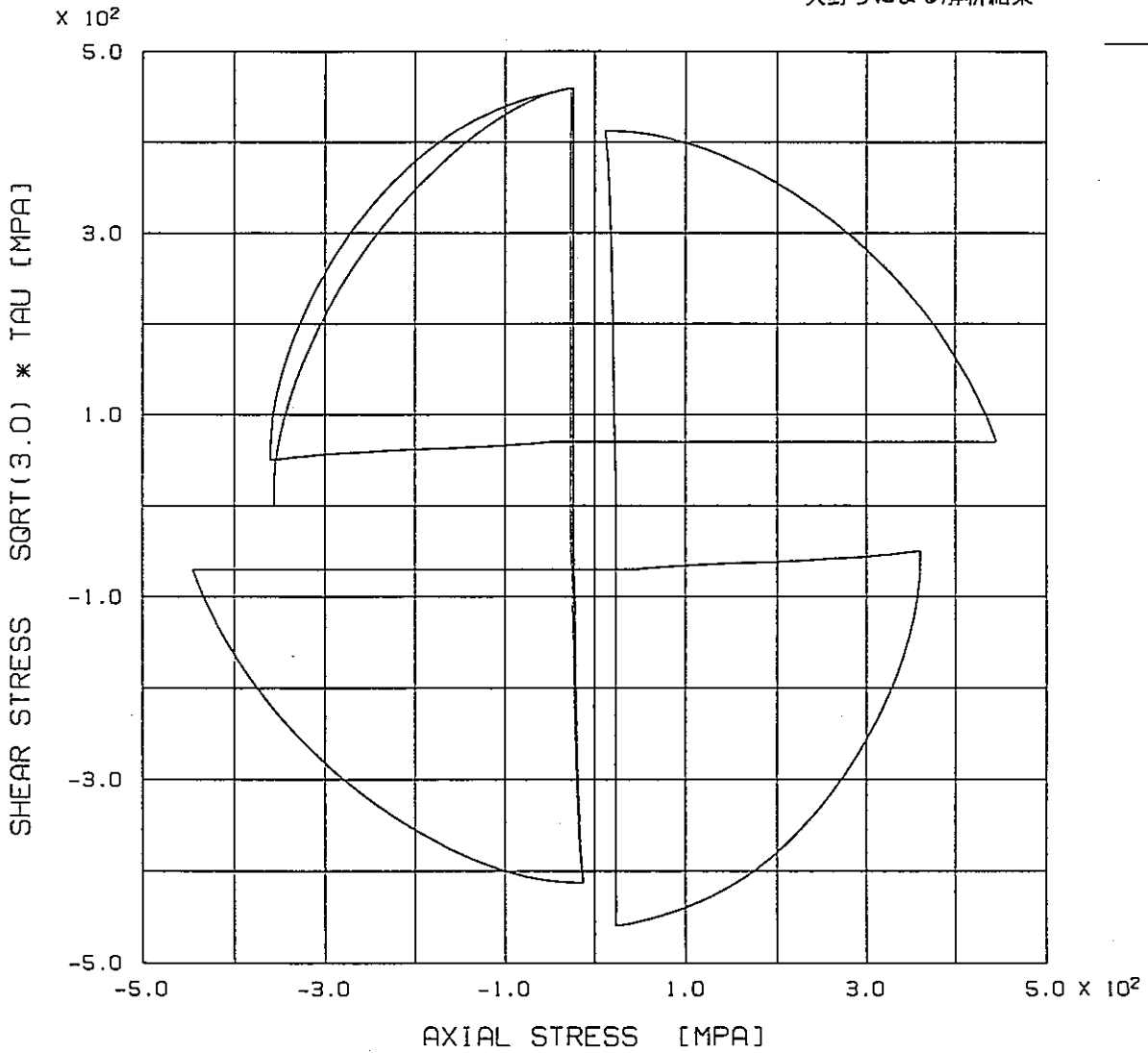
CROSS BAR STRAIN PASS OF MOD.9CR-1MO (FIG.7-A) BY AMODEL MODEL-1 M=8

図 8 ADMODEL による十字形ひずみ経路問題の解析結果 (モデル1, M=8)



FINAS STATIC ANALYSIS

大野らによる解析結果²⁾



CROSS BAR STRAIN PASS OF MOD.9CR-1M0 (FIG.7-B) BY AMODEL
MODEL-2 M=8

図 9 AMODEL による十字形ひずみ経路問題の解析結果 (モデルII, M=8)

付録

大野一王モデルを定義するサブルーチン ソースリスト

SUBROUTINE XMAT1 (IEXN, IECOD, NCOMP, IDMAT, IP, ISTEP,
 TIME, DTIME, TEMP, DTEMP, GEOM, CORD,
 DISP, EPS, SIG, USTV, USTS,
 EMAT, GG, IFLG)

C * FUNCTION OF THIS SUBROUTINE
 C TO CALCULATE CONSTITUTIVE EQUATION AND UPDATE FLAGS
 C FOR FINAS USER DEFINED CONSTITUTIVE MODEL.

$$(DSIG) = (EMAT)(DEPS) + (GG)$$

C PROGRAMED BY K. GARATANI (CRC) ON DEC. 1989 (VER 1.0)
 C ARG. NNOR REMOVED ON JUL. 1993 (VER 1.1)

C * INPUT VARIABLES

C IEXN D. ARG (INT) : ELEMENT EXTERNAL ID.
 C IECOD D. ARG (INT) : ELEMENT TYPE CODE. (REF. TABLE C. 1-1)
 C NCOMP D. ARG (INT) : COMPONENTS NO. OF STRESS OR STRAIN
 C IDMAT D. ARG (INT) : MATERIAL ID.
 C IP D. ARG (INT) : ELEMENT INTEGRATION POINT ID.
 C ISTEP D. ARG (INT) : STEP COUNT
 C TIME D. ARG (INT) : TIME
 C DTIME D. ARG (REAL*8) : TIME INCREMENT
 C TEMP D. ARG (REAL*8) : TEMPERATURE
 C DTEMP D. ARG (REAL*8) : TEMPERATURE INCREMENT
 C GEOM(I) D. ARG (REAL*8) : GEOMETRY DATA
 C CORD(I) D. ARG (REAL*8) : NODAL COORDINATES OF THE ELEMENT
 C DISP(I) D. ARG (REAL*8) : NODAL DISPLACEMENT OF THE ELEMENT
 C EPS (I) D. ARG (REAL*8) : TOTAL STRAIN COMPONENTS
 C SIG (I) D. ARG (REAL*8) : STRESS COMPONENTS
 C USTV(I, J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES (VEC. FORM)
 C USTS(J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES (SCR. FORM)
 C *IFLG(J) D. ARG (INT) : USER DEFINED FLAGS

C * OUTPUT VARIABLES

C EMAT(I, J) D. ARG (REAL*8) : INCREMENTAL STRESS-STRAIN MATRIX
 C GG (I) D. ARG (REAL*8) : PSEUDO STRESS INCREMENT VECTOR
 C *IFLG(J) D. ARG (INT) : USER DEFINED FLAGS

C-----< USER COMMENT BELOW >-----

C=====<< FOR OHNO-WANG MODEL >>=====

C PROGRAMED BY K. GARATANI ON DEC. 1995
 C
 C OHNO-WANG MODEL IMPLEMENTED BY FOLLOWING PAPER (IN JAPANESE)
 C

C .. IMPLEMENTATION FOR MODEL-1 , AND MODEL-2 ..
 C N. OHNO, J. WANG : A KINEMATIC HARDENING RULE WITH CRITICAL
 C STATE OF DYNAMIC RECOVERY (1ST REPORT, FORMULATION AND BASIC
 C FEATURES) JSME-A VOL. 58-547 (1992-3)

C .. IMPLEMENTATION FOR TEMPERATURE DEPENDENCY ..
 C M. KOBAYASHI, N. OHNO : ANALYSIS OF THERMAL RATCHETTING OF A CYLINDER
 C SUBJECTED TO AXIALLY MOVING TEMPERATURE FRONT
 C (COMPARISON WITH EXPERIMENT) JSME 95.7/7 NO. 953-2

C * MODEL STATUS VARIABLES

C USTV(I, J) .. VECTOR VARIABLES

C J= 1: EPSP(6) ; PLASTIC STRAIN
 C J= 2: ALI (6, 1) ; 1-ST BACK STRESS TERM AT T
 C J= 3: ALI (6, 2) ; 2-ND BACK STRESS TERM AT T
 C J= 4: ALI (6, 3) ; 3-TH BACK STRESS TERM AT T
 C J= 5: ALI (6, 4) ; 4-TH BACK STRESS TERM AT T
 C J= 6: ALI (6, 5) ; 5-TH BACK STRESS TERM AT T
 C J= 7: ALI (6, 6) ; 6-TH BACK STRESS TERM AT T
 C J= 8: ALI (6, 7) ; 7-TH BACK STRESS TERM AT T
 C J= 9: ALI (6, 8) ; 8-TH BACK STRESS TERM AT T
 C J=10: ALI (6, 9) ; 9-TH BACK STRESS TERM AT T
 C J=11: ALI (6, 10) ; 10-TH BACK STRESS TERM AT T

C USTS(J) .. SCALAR VARIABLES

C J=1: EPB ; ACCUMULATED EQUIVALENT PLASTIC STRAIN
 C J=2: PHAI ; VALUE OF TEMPERATURE FUNCTION PHAI(T)
 C IFLG(J) .. FLAGS
 C J=1: IELPL ; EL.-PL. FLAG (0:ELASTIC, 1:PLASTIC)

C * LOCAL VARIABLES

C NBPA PARM. (INT) : MAX. COUNT FOR BACK STRESS TERM
 C AL (6) LOCAL (REAL*8) : SUM OF BACK STRESS AT T
 C ALS (6) LOCAL (REAL*8) : SUM OF BACK STRESS AT T
 C EPSE(6) LOCAL (REAL*8) : ELASTIC STRAIN
 C DFDS(6) LOCAL (REAL*8) : FLOW VECTOR (DF/DS)
 C DALSDT(6) LOCAL (REAL*8) : TEMP. DIFF. OF BACK STRESS AT T

IMPLICIT REAL*8 (A-H, O-Z)
 DIMENSION GEOM(*), CORD(*), DISP(*), EPS(*), SIG(*),
 USTV(6,*), USTS(*),
 EMAT(NCOMP, NCOMP), GG(*) , IFLG(*)

C (USER PROGRAM BELOW)
 C
 C PARAMETER (NBPA=10)


```

      DIMENSION ALI (6,NBPA), AL (6), ALS (6),
      EPSE(6), DFDS(6), DALSDT(6),
      ZETA( NBPA), R (NBPA)
C
C ***** SET STRESS FIELD KEY BY ELEMENT CODE
C
      CALL YFIELD( IEXN , IECOD , KFIELD, NCOMPX , NNOR )
C
      IF (NCOMP.NE.NCOMPX) GOTO 9010
C
C ***** SET MATERIAL CONSTATS FOR OHNO-WANG MODEL
C
      CALL YOWMAT( IEXN , IDMAT , TEMP , DTEMP ,
      NB , KS , KT , KM ,
      E1 , PO11 , SIG01 , PHA11 ,
      E2 , PO12 , SIG02 , PHA12 ,
      ALP , AM , ZETA , R )
C
      IF (NB.GT.NBPA) GOTO 9020
C
C ***** SET STATUS VARIABLES AND REAL BACK STRESS AL*
C
      IELPL = IFLG(1)
C
      DO 100 I=1,NCOMP
      AL ( I ) = 0.0
100 CONTINUE
C
      DO 120 IB=1,NB
      DO 110 I=1,NCOMP
      ALI ( I, IB) = USTV( I, 1+IB)
      AL ( I ) = AL ( I ) + ALI ( I, IB)
110 CONTINUE
120 CONTINUE
C
      DO 130 I=1,NCOMP
      ALS( I ) = AL ( I ) * PHA11
130 CONTINUE
C
C ***** GET ELASTIC STRAIN
C
      CALL YFEPSE ( NCOMP , NNOR , E1 , PO11 , SIG ,
      EPSE )
C
C ***** SET TEMPERATURE DIFFERENCE OF STATUS VARIABLES
C

```

```

      CALL YOWDAT ( NCOMP , PHA11 , PHA12 , DTEMP , AL ,
      DALSDT )
C
      IF (DTEMP.EQ.0.0) THEN
      DKAPDT = 0.0
      ELSE
      DKAPDT = (SIG02-SIG01) / DTEMP
      ENDIF
C
C ***** SET 'EMAT' AND 'GG' DEPENDS ON STATUS
C
      IF (IELPL.EQ.0) THEN
C
C .. ELASTIC
C
      CALL YFDE ( KFIELD , NCOMP , E2 , PO12 , EMAT )
C
      CALL YFGE ( KFIELD , NCOMP , NNOR , E1 , PO11 ,
      E2 , PO12 , ALP , DTEMP , EPSE ,
      GG )
      ELSE
C
C .. PLASTIC
C
      CALL YFLOW2( NCOMP , NNOR , SIG , ALS , DFDS )
C
      CALL YOWHD ( NCOMP , NNOR , NB , KM , AM ,
      ZETA , R , DFDS , ALI , HD )
C
      HD = HD * PHA12
C
      CALL YFDEP ( KFIELD , NCOMP , NCOMP , E2 , PO12 ,
      HD , DFDS , EMAT )
C
      AKAP2 = SIG02
C
      CALL YFDKDT( NCOMP , NNOR , SIG , ALS , AKAP2 ,
      DTEMP , DKAPDT )
C
      CALL YFGEP ( KFIELD , NCOMP , NNOR , E1 , PO11 ,
      E2 , PO12 , ALP , HD , DTEMP ,
      DKAPDT , DALSDT, DFDS , EPSE , GG )
      ENDIF
C
      RETURN
C
C ***** ERROR
C

```

```

9010 WRITE(6,*) ' ***** ERROR ON SUB. XMAT1 : ILLEGAL NCOMP ',
      NCOMP, IEXN
      CALL PEXIT
      STOP
9020 WRITE(6,*) ' ***** ERROR ON SUB. XMAT1 : B. STRESS COUNT EXCEED ',
      NBPA,NB
      CALL PEXIT
      STOP
      END
SUBROUTINE XMAT2 ( IEXN, IECOD, NCOMP, IDMAT, IP, ISTEP,
      TIME, DTIME, TEMP, DTEMP, GEOM, CORD,
      DISP, EPS, SIG, USTV, USTS,
      DDISP, DEPS, DSIG, DUSTV, DUSTS, IFLG )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   CALCULATE INCREMENT OF STATUS VARIABLES AND UPDATE FLAGS
C   FOR FINAS USER DEFINED CONSTITUTIVE MODEL.
C
C   PROGRAMED BY K. GARATANI (CRC) ON DEC. 1989 (VER 1.0)
C   ARG. NNOR REMOVED ON JUL. 1993 (VER 1.1)
C-----
C * INPUT VARIABLES
C   IEXN   D. ARG (INT)   : ELEMENT EXTERNAL ID.
C   IECOD  D. ARG (INT)   : ELEMENT TYPE CODE. (REF. TABLE C.1-1)
C   NCOMP  D. ARG (INT)   : COMPONENTS NO. OF STRESS OR STRAIN
C   IDMAT  D. ARG (INT)   : MATERIAL ID.
C   IP     D. ARG (INT)   : ELEMENT INTEGRATION POINT ID.
C   ISTEP  D. ARG (INT)   : STEP COUNT
C   TIME   D. ARG (INT)   : TIME
C   DTIME  D. ARG (REAL*8) : TIME INCREMENT
C   TEMP   D. ARG (REAL*8) : TEMPERATURE
C   DTEMP  D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   GEOM(1) D. ARG (REAL*8) : GEOMETROY DATA
C   CORD(1) D. ARG (REAL*8) : NODAL COORDINATES OF THE ELEMENT
C   DISP(1) D. ARG (REAL*8) : NODAL DISPLACEMENT OF THE ELEMENT
C   DDISP(1) D. ARG (REAL*8) : NODAL DISPLACEMENT INCREMENT
C   EPS (1) D. ARG (REAL*8) : TOTAL STRAIN COMPONENTS
C   DEPS (1) D. ARG (REAL*8) : TOTAL STRAIN COMPONENTS INCREMENT
C   SIG (1) D. ARG (REAL*8) : STRESS COMPONENTS
C   USTV(1,J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES (VEC. FORM)
C   USTS(J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES (SCR. FORM)
C   *IFLG(J) D. ARG (INT) : USER DEFINED FLAGS
C-----
C * OUTPUT VARIABLES
C   DSIG (1) D. ARG (REAL*8) : STRESS COMPONENTS INCREMENT
C   DUSTV(1,J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES INCREMENT

```

```

C   DUSTS(J) D. ARG (REAL*8) : USER DEFINED ST. VARIABLES INCREMENT
C   *IFLG(J) D. ARG (INT) : USER DEFINED FLAGS
C
C-----< USER COMMENT BELLOW >-----
C
C=====<< FOR OHNO-WANG MODEL >>=====
C
C   PROGRAMED BY K. GARATANI ON DEC. 1995
C
C   OHNO-WANG MODEL IMPLEMENTED BY FOLLOWING PAPER (IN JAPANESE)
C
C .. IMPLEMENTATION FOR MODEL-1 , AND MODEL-2 ..
C   N. OHNO, J. WANG : A KINEMATIC HARDENING RULE WITH CRITICAL
C   STATE OF DYNAMIC RECOVERY (1ST REPORT, FORMULATION AND BASIC
C   FEATURES) JSME-A VOL.58-547 (1992-3)
C
C .. IMPLEMENTATION FOR TEMPERATURE DEPENDENCY ..
C   M. KOBAYASHI, N. OHNO : ANALYSIS OF THERMAL RATCHETING OF A CYLINDER
C   SUBJECTED TO AXIALLY MOVING TEMPERATURE FRONT
C   (COMPARION WITH EXPERIMENT) JSME 95.7/7 NO.953-2
C
C * MODEL STATUS VARIABLES
C   USTV(1,J) .. VECTOR VARIABLES
C   J= 1: EPSP(6) : PLASTIC STRAIN
C   J= 2: AL1 (6, 1) : 1-ST BACK STRESS TERM AT TO
C   J= 3: AL1 (6, 2) : 2-ND BACK STRESS TERM AT TO
C   J= 4: AL1 (6, 3) : 3-TH BACK STRESS TERM AT TO
C   J= 5: AL1 (6, 4) : 4-TH BACK STRESS TERM AT TO
C   J= 6: AL1 (6, 5) : 5-TH BACK STRESS TERM AT TO
C   J= 7: AL1 (6, 6) : 6-TH BACK STRESS TERM AT TO
C   J= 8: AL1 (6, 7) : 7-TH BACK STRESS TERM AT TO
C   J= 9: AL1 (6, 8) : 8-TH BACK STRESS TERM AT TO
C   J=10: AL1 (6, 9) : 9-TH BACK STRESS TERM AT TO
C   J=11: AL1 (6,10) : 10-TH BACK STRESS TERM AT TO
C   USTS(J) .. SCALAR VARIABLES
C   J=1: EPB : ACCUMULATED EQUIVALENT PLASTIC STRAIN
C   J=2: PHA1 : VALUE OF TEMPERATURE FUNCTION PHA1(T)
C   IFLG(J) .. FLAGS
C   J=1: IELPL : EL.-PL. FLAG (0:ELASTIC, 1:PLASTIC)
C-----
C * LOCAL VARIABLES
C   NBPA PARM. (INT) : MAX. COUNT FOR BACK STRESS TERM
C   AL (6) LOCAL (REAL*8) : SUM OF BACK STRESS AT TO
C   ALS (6) LOCAL (REAL*8) : SUM OF BACK STRESS AT T
C   EPSE(6) LOCAL (REAL*8) : ELASTIC STRAIN
C   DFDS(6) LOCAL (REAL*8) : FLOW VECTOR (DF/DS)
C   DALSDT(6) LOCAL (REAL*8) : TEMP. DIFF. OF BACK STRESS AT T

```

```

C-----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION GEOM(*), CORD(*), IFLG(*),
             DISP(*), EPS(*), SIG(*), USTV(6,*), USTS(*),
             DDISP(*), DEPS(*), DSIG(*), DUSTV(6,*), DUSTS(*)
C
C..... ( USER PROGRAM BELOW ) .....
C
  PARAMETER (NBPA=10)
  DIMENSION ZETA(NBPA), R(NBPA)
  DIMENSION EPSE (6) , AL (6) , ALS(6) , ALI (6,NBPA) ,
             DAL1(6,NBPA) ,
             SIG2(6) , ALS2(6) ,
             EPSEX(6) , SIGX(6) ,
             DSIGX(6) ,
             DSIGY(6)
  DIMENSION DEPSP (6) , DFDS (6) , DALSDT (6) ,
             DFDSX(6) , DALSDY(6)
  DIMENSION DEPSY (6)
C
C//////// PHASE 1: PRE PROCESS
C
C***** SET STRESS FIELD KEY BY ELEMENT CODE
C
  CALL YFIELD( IEXN , IECOD , KFIELD, NCOMPX , NNOR )
C
  IF (NCOMP.NE.NCOMPX) GOTO 9010
C
C***** SET MATERIAL CONSTATS FOR OHNO-WANG MODEL
C
  CALL YOWMAT( IEXN , IDMAT , TEMP , DTEMP ,
             NB , KS , KT , KM ,
             E1 , PO11 , SIG01 , PHA11 ,
             E2 , PO12 , SIG02 , PHA12 ,
             ALP , AM , ZETA , R )
C
  IF (NB.GT.NBPA) GOTO 9020
C
C***** SET STATUS VARIABLES AND REAL BACK STRESS AL*
C
  IELPL = IFLG(1)
C
  DO 100 I=1,NCOMP
    AL ( I ) = 0.0
  100 CONTINUE

```

```

C
  DO 120 IB=1,NB
    DO 110 I=1,NCOMP
      ALI(I,IB) = USTV(I,1+IB)
      AL ( I ) = AL ( I ) + ALI(I,IB)
    110 CONTINUE
  120 CONTINUE
C
  DO 130 I=1,NCOMP
    ALS(I) = AL ( I ) * PHA11
  130 CONTINUE
C
C***** GET ELASTIC STRAIN
C
  CALL YFEPSE ( NCOMP , NNOR , E1 , PO11 , SIG ,
             EPSE )
C
C***** SET TEMPERATURE DIFFERENCE OF STATUS VARIABLES
C
  CALL YOWDAT ( NCOMP , PHA11 , PHA12 , DTEMP , AL ,
             DALSDT )
C
  IF (DTEMP.EQ.0.0) THEN
    DKAPDT = 0.0
  ELSE
    DKAPDT = (SIG02-SIG01) / DTEMP
  ENDIF
C
C//////// PHASE 2: GET STATUS VARIABLES INCREMENT
C
  IF (IELPL.EQ.0) THEN
C
C..... PREV. STEP WAS ELASTIC
C..... GET STRESS INC.
C
  CALL YF0SE ( KFIELD , NCOMP , NNOR , E1 , PO11 ,
             E2 , PO12 , ALP , DTEMP , EPSE ,
             DEPS , DSIG )
C
C..... GET PLASTIC STRAIN INC.
C
  DO 200 I=1,NCOMP
    DEPSP(I) = 0.0
  200 CONTINUE
  DEPB = 0.0
C
C..... GET STATUS VARIABLES INC.
C
  DO 220 IB=1,NB
    DO 210 I=1,NCOMP
      DALI(I,IB) = 0.0

```

```

210 CONTINUE
220 CONTINUE
C
DO 230 I=1, NCOMP
  SIG2(I) = SIG(I) + DSIG(I)
  ALS2(I) = AL (I) * PHA12
230 CONTINUE
C
AKAP2 = SIG02
CALL YFSB2 (NCOMP, NNOR, SIG2, ALS2, AKAPX)
C
IF (AKAPX*1.001.GT.AKAP2) THEN
C
C      ... ELASTIC TO PLASTIC
C      (ASSUME BACK STRESS CHANGE)
C
CALL YFGAM ( NCOMP , NNOR , SIG , DSIG , ALS2 ,
            AKAP2 , GAM )
C      ... ELASTIC PART -X
DO 240 I=1, NCOMP
  DSIGX(I) = DSIG(I) * GAM
  SIGX(I) = SIG (I) + DSIGX(I)
240 CONTINUE
C
EX = E1 + (E2 -E1 )*GAM
POIX = PO11 + (PO12 -PO11 )*GAM
PHAIX = PHA12
C
CALL YFEPSE( NCOMP , NNOR , EX , POIX , SIGX ,
            EPSEX )
C
CALL YFLOW2( NCOMP , NNOR , SIGX , ALS2 , DFDSX )
C
C      ... PLASTIC PART -Y
C
DO 250 I=1, NCOMP
  DEPSY(I) = DEPS(I)*(1.0-GAM)
250 CONTINUE
  DTY = DTEMP *(1.0-GAM)
C      (B. STRESS CHANGE AT ELASTIC)
DO 260 I=1, NCOMP
  DALSDY(I) = 0.0
260 CONTINUE
C      .. GET STRESS INC.
CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,

```

```

EX , POIX , E2 , PO12 , PHA12 ,
ALP , AM , ZETA , R , DTY ,
DKAPDT , DALSDY , DFDSX , ALI , EPSEX ,
DEPSY , DSIGY , DALI , DEPS , DEPB )
C
IF (KS.NE.0) THEN
C
C      .. CHECK BREAK POINT
C
CALL YOWBPX ( NCOMP , NNOR , NB , R , ALI ,
            DALI , KBP , BP )
C
IF (KBP.NE.0) THEN
C
C      .. SUBSTEP ON BREAK POINT
C
CALL YOWBP1( KFIELD , NCOMP , NNOR , NB , KM ,
            EX , POIX , PHAIX , E2 , PO12 ,
            PHA12 , ALP , AM , ZETA , R ,
            DTY , DKAPDT , DALSDY , BP , SIGX ,
            ALI , EPSEX , DEPSY , DSIGY , DALI ,
            DEPS , DEPB )
C
ENDIF
C
ENDIF
C
C      ... SUM UP STATUS VARIABLES
DO 300 I=1, NCOMP
  DSIG(I) = DSIGX(I) + DSIGY(I)
300 CONTINUE
C      ... SET FLAG TO PLASTIC
  IELPL = 1
C
ENDIF
C
ELSE
C
C      .... PREV. STEP WAS PLASTIC
C
AKAP2 = SIG02
CALL YFDKDT( NCOMP , NNOR , SIG , ALS , AKAP2 ,
            DTEMP , DKAPDT )
C      ... CHECK UNLOADING
CALL YFLOW2( NCOMP , NNOR , SIG , ALS , DFDS )
C
CALL YFXLAM( KFIELD , NCOMP , NNOR , E1 , PO11 ,
            E2 , PO12 , ALP , DTEMP , DKAPDT ,

```

```

      DALSDT , DFDS , EPSE , DEPS , XLAM )
C
IF (XLAM.GE.0.0) THEN
C
      ... PLASTIC STATUS CONTINUE
      .. GET EACH S. V. INC.
C
      CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,
      E1 , PO11 , E2 , PO12 , PHA12 ,
      ALP , AM , ZETA , R , DTEMP ,
      DKAPDT , DALSDT , DFDS , ALI , EPSE ,
      DEPS , DSIG , DALI , DEPS , DEPSP )
C
IF (KS.NE.0) THEN
C
      .. CHECK BREAK POINT
C
      CALL YOWBPX ( NCOMP , NNOR , NB , R , ALI ,
      DALI , KBP , BP )
C
IF (KBP.NE.0) THEN
C
      .. SUBSTEP ON BREAK POINT
C
      CALL YOWBP1( KFIELD , NCOMP , NNOR , NB , KM ,
      E1 , PO11 , PHA11 , E2 , PO12 ,
      PHA12 , ALP , AM , ZETA , R ,
      DTEMP , DKAPDT , DALSDT , BP , SIG ,
      ALI , EPSE , DEPS , DSIG , DALI ,
      DEPS , DEPSP )
C
      ENDIF
C
      ENDIF
C
ELSE
C
      ... PLASTIC TO ELASTIC (UNLOAD)
      .. GET STRESS INC.
C
      CALL YFDSE ( KFIELD , NCOMP , NNOR , E1 , PO11 ,
      E2 , PO12 , ALP , DTEMP , EPSE ,
      DEPS , DSIG )
C
      .. GET PLASTIC STRAIN INC.
C
DO 450 I=1,NCOMP
      DEPSP(I) = 0.0
450 CONTINUE
      DEPSP = 0.0
C
      .. GET STATUS VARIABLES INC.
C
DO 470 IB=1,NB
      DO 460 I=1,NCOMP
          DALI(I,IB) = 0.0

```

```

460 CONTINUE
470 CONTINUE
C
      .. SET FLAGS
      IELPL = 0
C
      ENDIF
C
      ENDIF
C
C ////////////// PHASE 3: POST PROCESS (STORE ST. VARIABLES)
C
      DO 500 I=1,NCOMP
          DUSTV(I,1) = DEPS(I)
500 CONTINUE
      DO 520 IB=1,NB
          DO 510 I=1,NCOMP
              DUSTV(I,1+IB) = DALI(I,IB)
510 CONTINUE
520 CONTINUE
C
      DUSTS(1) = DEPSP
      DUSTS(2) = PHA12 - USTS(2)
C
      IFLG(1) = IELPL
C
      RETURN
C
C ***** ERROR
C
9010 WRITE(6,*) ' ***** ERROR ON SUB. XMAT2 : ILLEGAL NCOMP ',NCOMP
      CALL PEXIT
      STOP
9020 WRITE(6,*) ' ***** ERROR ON SUB. XMAT2 : B. STRESS COUNT EXCEED ',
      NBPA,NB
      CALL PEXIT
      STOP
      END
      SUBROUTINE XREAD(LU5)
C-----
C * FUNCTION OF THIS SUBROUTINE
C READ INPUT DATA AFTER 'END FINAS' (FINAS USER SUBROUTINE)
C PROGRAMED BY K. GARATANI (CRC) ON OCT. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C LU5 D.ARG (INT) : FILE CODE TO READ
C-----

```

```

C * OUTPUT VARIABLES
C   NONE
C
C-----< USER COMMENT BELOW >-----
C
C-----<< FOR OHNO-WANG MODEL >>-----
C   BY K. GARATANI (CRC) ON DEC. 1995 (VER 1.0)
C * USER MATERIAL DATA INPUT ROUTINE
C   FORMAT IS SAME AS MATERIAL DATA SECTION.
C   CONSTANT OR TEMPERATURE TABLE (F5) TYPE ALLOWED
C-----
C * DATA BLOCK /YBXMAT/ FOR MATERIAL DATA DEFINITION
C   NTBLPA   PARM. (INT) : MAX. COUNT OF MATERIAL TABLE
C                                     NTBLPA MUST BE MULTIPLE OF 3
C   NMATPA   PARM. (INT) : MAX. COUNT OF MATERIAL KINDS
C   NMIDPA   PARM. (INT) : MAX. COUNT OF MATERIAL ID.
C   NKEYPA   PARM. (INT) : MAX. COUNT OF OPTION KEY
C
C   MID ( K ) /YBXMAT/(INT) : K-TH MATERIAL ID.
C   KEY ( IO,K ) /YBXMAT/(INT) : K-TH OPTION KEY DATA
C   NTBL ( J,K ) /YBXMAT/(INT) : DATA LENGTH OF K-TH MAT. ID.
C                                     AND J-TH KIND (=0:NONE DATA)
C   TTBL ( I, J, K ) /YBXMAT/(REAL*8) : I-TH TEMPERATURE TABLE OF
C                                     K-TH MAT. ID. AND J-TH KIND
C   VTBL ( I, J, K ) /YBXMAT/(REAL*8) : I-TH MATERIAL VALUE TABLE OF
C                                     K-TH MAT. ID. AND J-TH KIND
C-----
C * MATERIAL KIND KEY AS BELOW
C   KIND (J)  MNEMONIC  PHYSICAL MEANING
C   1         E         YOUNG'S MODULAS
C   2         NU        POISSON'S RATIO
C   3         ALPHA     INSTANTANEOUS THERMAL EXPANTION RATIO
C   4         SIGO      INITIAL YIELD STRESS
C   5         M         CONSTATNT FOR MODEL-2
C   6         PHA1A     TEMPERATURE FUNC. CONSTANTS-1
C   7         PHA1B     TEMPERATURE FUNC. CONSTANTS-2
C   8         ZETA01    BACK STRESS CONSTANTS FOR TERM-01
C   9         ZETA02    BACK STRESS CONSTANTS FOR TERM-02
C   10        ZETA03    BACK STRESS CONSTANTS FOR TERM-03
C   11        ZETA04    BACK STRESS CONSTANTS FOR TERM-04
C   12        ZETA05    BACK STRESS CONSTANTS FOR TERM-05
C   13        ZETA06    BACK STRESS CONSTANTS FOR TERM-06
C   14        ZETA07    BACK STRESS CONSTANTS FOR TERM-07
C   15        ZETA08    BACK STRESS CONSTANTS FOR TERM-08
C   16        ZETA09    BACK STRESS CONSTANTS FOR TERM-09
C   17        ZETA10    BACK STRESS CONSTANTS FOR TERM-10

```

```

C   18        R01      BACK STRESS CONSTANTS FOR TERM-01
C   19        R02      BACK STRESS CONSTANTS FOR TERM-02
C   20        R03      BACK STRESS CONSTANTS FOR TERM-03
C   21        R04      BACK STRESS CONSTANTS FOR TERM-04
C   22        R05      BACK STRESS CONSTANTS FOR TERM-05
C   23        R06      BACK STRESS CONSTANTS FOR TERM-06
C   24        R07      BACK STRESS CONSTANTS FOR TERM-07
C   25        R08      BACK STRESS CONSTANTS FOR TERM-08
C   26        R09      BACK STRESS CONSTANTS FOR TERM-09
C   27        R10      BACK STRESS CONSTANTS FOR TERM-10
C-----
C * OPTION KEYS
C   KEY (1,K) /YBXMAT/(INT) : BACK STRESS COUNT (DEF.=10)
C   KEY (2,K) /YBXMAT/(INT) : SUBSTEP METHOD (0:YES,1:NO)
C   KEY (3,K) /YBXMAT/(INT) : TEMP. DEPENDENCY (0:YES,1:NO)
C   KEY (4,K) /YBXMAT/(INT) : =0:MODEL-1,=1:MODEL-2 (M)
C-----
C * CALLED SUBROUTINES
C   YXRHED   : PRINT INPUT DATA ECHO HEADER
C   YXREAD   : READ/WRITE INPUT DATA
C   YXINT5   : READ 5 COLUMNED INTEGER DATA
C-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   CHARACTER LINE*80 , BLK10*10
C
C   PARAMETER ( NTBLPA = 21 , NMATPA = 27 , NMIDPA = 4 ,
C               NKEYPA = 10 )
C
C   COMMON /YBXMAT/
C     MID ( NMIDPA ) , KEY ( NKEYPA , NMIDPA ) ,
C     NTBL ( NMATPA , NMIDPA ) ,
C     TTBL ( NTBLPA , NMATPA , NMIDPA ) ,
C     VTBL ( NTBLPA , NMATPA , NMIDPA )
C
C     .. MAT. MNEMONIC
C
C   DIMENSION MKINDL ( NMATPA )
C   CHARACTER*10 MKIND ( NMATPA )
C
C     // UPPER CASE //
C
C   DATA MKIND
C   / 'E' , 'NU' , 'ALPHA' ,
C   'SIGO' , 'M' , 'PHA1A' ,
C   'PHA1B' , 'ZETA01' , 'ZETA02' ,
C   'ZETA03' , 'ZETA04' , 'ZETA05' ,
C   'ZETA06' , 'ZETA07' , 'ZETA08' ,
C   'ZETA09' , 'ZETA10' , 'R01'

```

```

      'R02      'R03      'R04      '
      'R05      'R06      'R07      '
      'R08      'R09      'R10      /
DATA MKINDL
/      2      . 2      . 5
      4      . 1      . 5
      5      . 6      . 6
      6      . 6      . 6
      6      . 6      . 6
      6      . 6      . 3
      3      . 3      . 3
      3      . 3      . 3
      3      . 3      . 3      /
C
DATA LU6 /      6      /
DATA BLK10 /      /
C ***** CLEAR TABLE
C
DO 120 K=1,NMIDPA
DO 100 J=1,NMATPA
NTBL(J,K) = 0
100 CONTINUE
DO 110 J=1,NKEYPA
KEY(J,K) = 0
110 CONTINUE
MID(K) = 0
120 CONTINUE
C
C ***** PRINT HEADER
C
CALL YXRHED
C
C ***** FIND 'USER MATERIAL' CARD
C
LC = 0
C
200 CONTINUE
C
CALL YXREAD ( LU5, LC, KEND, LINE )
IF ( KEND .EQ. 1 ) GOTO 9010
C
// UPPER CASE //
C
IF ( INDEX(LINE(1:13), 'USER MATERIAL') .EQ. 0 ) GOTO 200
C

```

```

C ***** MATERIAL ID LOOP
C
K MID = 0
C
C *** READ MATERIAL ID. AND OPTION KEY
C
CALL YXREAD ( LU5, LC, KEND, LINE )
C
1000 CONTINUE
C
IF ( KEND .EQ. 1 ) GOTO 5000
C
K MID = K MID + 1
IF ( K MID .GT. NMIDPA ) GOTO 9015
C
CALL YXINT5 ( LINE(1:5), MID(KMID) )
IF ( MID(KMID) .LE. 0 ) GOTO 9030
C
DO 1100 IM = 1, KMID-1
IF ( MID(KMID) .EQ. MID(IM) ) GOTO 9020
1100 CONTINUE
C
C .. OPTION KEY
C
K POS = 21
DO 1200 IK = 1, NKEYPA
CALL YXINT5 ( LINE(KPOS:KPOS+4), KEY(IK, KMID) )
K POS = K POS + 5
1200 CONTINUE
C
READ (LINE, 501) ( KEY(I, KMID), I=1, 10 )
C
C *** MATERIAL CONSTANT LOOP
C
2000 CONTINUE
C
CALL YXREAD ( LU5, LC, KEND, LINE )
C
IF ( KEND .EQ. 1 ) GOTO 5000
IF ( LINE(1:10) .NE. BLK10 ) GOTO 1000
C
DO 4000 JMAT=1,NMATPA
C
C .. SEARCH MAT. MNEMONIC
IFLG = INDEX( LINE(11:20), MKIND(JMAT) (1:MKINDL(JMAT)) )
C
IF ( IFLG .NE. 0 ) THEN
C

```

```

C      IF ( NTBL (JMAT,KMID) .NE. 0 ) GOTO 9035
C      // UPPER CASE //
C      IF ( INDEX( LINE(21:25), 'F5' ) .EQ. 0 ) THEN
C          . CONSTANT FORM
C          READ (LINE(21:30),502) VTBL(1,JMAT,KMID)
C              NTBL( JMAT,KMID) = 1
C      ELSE
C          . F5 FORM
C          CALL YXINT5 ( LINE(26:30), NTBL(JMAT,KMID) )
C          IF ( MOD(NTBLPA,3) .NE. 0 ) GOTO 9040
C          IF ( NTBL(JMAT,KMID) .GT. NTBLPA ) GOTO 9050
C          IF ( NTBL(JMAT,KMID) .LE. 0 ) GOTO 9060
C
C          DO 3000 IDAT = 3, NTBL(JMAT,KMID)+2, 3
C
C              CALL YXREAD ( LU5, LC, KEND, LINE )
C              IF ( KEND .EQ. 1 ) GOTO 9010
C              READ (LINE,503)
C                  ( TTBL(1,JMAT,KMID), VTBL(1,JMAT,KMID)
C                      , I=IDAT-2, IDAT)
C
C          3000 CONTINUE
C
C          ENDIF
C          . NEXT MAT.
C          GOTO 2000
C
C          ENDIF
C
C          4000 CONTINUE
C
C          GOTO 9070
C
C          5000 CONTINUE
C
C          ----- FOR OHNO-WANG MODEL -----
C
C          DO 7000 KMID=1, NMIDPA
C
C              MATID = MID(KMID)
C
C              IF (MATID.EQ.0) GOTO 8000
C
C          ***** SET DEFAULT VALUES
C
C              .. BACK STRESS COUNT (DEF.=10)

```

```

C      IF ( KEY(1,KMID) .EQ. 0 ) KEY(1,KMID) = 10
C      IF ( KEY(1,KMID) .GT. 10 ) GOTO 9610
C
C      ***** CHECK MATERIAL DATA
C          .. E, NU, PLPHA (CONST. OR F5)
C
C      IF ( NTBL( 1,KMID).EQ.0 ) GOTO 9620
C      IF ( NTBL( 2,KMID).EQ.0 ) GOTO 9620
C      IF ( (NTBL( 3,KMID).EQ.0) .AND. (KEY(3,KMID).EQ.0) ) GOTO 9620
C          .. SIGO (BLOW ONLY CONST.)
C
C      IF ( NTBL( 4,KMID).NE.1 ) GOTO 9620
C          .. M
C
C      IF ( (NTBL( 5,KMID).NE.1) .AND. (KEY(4,KMID).EQ.1) ) GOTO 9620
C          .. PHA1A, PHA1B
C
C      IF ( (NTBL( 6,KMID).NE.1) .AND. (KEY(3,KMID).EQ.0) ) GOTO 9620
C      IF ( (NTBL( 7,KMID).NE.1) .AND. (KEY(3,KMID).EQ.0) ) GOTO 9620
C          .. R, ZETA
C
C      DO 6100 JMAT=8, 7+KEY(1,KMID)
C          IF ( NTBL(JMAT ,KMID).NE.1 ) GOTO 9620
C          IF ( NTBL(JMAT+10,KMID).NE.1 ) GOTO 9620
C
C      6100 CONTINUE
C
C      ***** WRITE OPTION INFORMATION
C
C      CALL PAGE(5, IDUMMY)
C
C      IF ( KEY(4,KMID).EQ.1 ) THEN
C          WRITE(6,6001) MATID
C      ELSE
C          WRITE(6,6002) MATID
C      ENDIF
C      WRITE(6,6003) KEY(1,KMID)
C      IF ( KEY(2,KMID).EQ.0 ) THEN
C          WRITE(6,6004)
C      ELSE
C          WRITE(6,6005)
C      ENDIF
C      IF ( KEY(3,KMID).EQ.0 ) THEN
C          WRITE(6,6006)
C      ELSE
C          WRITE(6,6007)
C      ENDIF
C
C      7000 CONTINUE
C
C      8000 CONTINUE

```



```

C
  RETURN
C
C ***** FORMAT
C
  501 FORMAT(20X, 10I5)
  502 FORMAT(F10.0)
  503 FORMAT(10X, 6F10.0)
C
  6001 FORMAT(1H0, 7X, '* OHNO-WANG MODEL (MODEL-2) MAT. ID = ', 15)
  6002 FORMAT(1H0, 7X, '* OHNO-WANG MODEL (MODEL-1) MAT. ID = ', 15)
  6003 FORMAT(1H , 17X, 'BACK STRESS COUNT      : ', 13)
  6004 FORMAT(1H , 17X, 'SUBSTEP INTEGRATION   : YES ')
  6005 FORMAT(1H , 17X, 'SUBSTEP INTEGRATION   : NO  ')
  6006 FORMAT(1H , 17X, 'TEMPERATURE DEPENDENCY : YES ')
  6007 FORMAT(1H , 17X, 'TEMPERATURE DEPENDENCY : NO  ')
C
C ***** ERROR
C
  9010 CONTINUE
    WRITE (LU6, 9011)
    CALL PEXIT
  9011 FORMAT(' ***** ERROR ON SUB.XREAD : UNEXPECTED END OF ',
    ' FILE')
C
  9015 CONTINUE
    WRITE (LU6, 9016) NMIDPA
    CALL PEXIT
  9016 FORMAT(' ***** ERROR ON SUB.XREAD : MAT. ID EXCEED ', 15)
C
  9020 CONTINUE
    WRITE (LU6, 9021)
    CALL PEXIT
  9021 FORMAT(' ***** ERROR ON SUB.XREAD : MAT. ID DUPLICATED ')
C
  9030 CONTINUE
    WRITE (LU6, 9031) MID(KMID)
    CALL PEXIT
  9031 FORMAT(' ***** ERROR ON SUB.XREAD : ILLEGAL MATERIAL ',
    ' ID ', 15)
C
  9035 CONTINUE
    WRITE (LU6, 9036)
    CALL PEXIT
  9036 FORMAT(' ***** ERROR ON SUB.XREAD : MAT. DATA DUPLICATED ')
C

```

```

9040 CONTINUE
  WRITE (LU6, 9041) NTBLPA
  CALL PEXIT
9041 FORMAT(' ***** ERROR ON SUB.XREAD : ILLEGAL NTBLPA ', 15,
  ' MUST BE MULTIPLE OF 3 ')
C
9050 CONTINUE
  WRITE (LU6, 9051) NTBLPA
  CALL PEXIT
9051 FORMAT(' ***** ERROR ON SUB.XREAD : TABLE DATA EXCEED ', 15)
C
9060 CONTINUE
  WRITE (LU6, 9061) NTBL(JMAT, KMID)
  CALL PEXIT
9061 FORMAT(' ***** ERROR ON SUB.XREAD : ILLEGAL TABLE COUNT ', 15)
C
9070 CONTINUE
  WRITE (LU6, 9071) LINE(11:20)
  CALL PEXIT
9071 FORMAT(' ***** ERROR ON SUB.XREAD : ILLEGAL MATERIAL ', A10)
C .....
9610 CONTINUE
  WRITE (LU6, 9611) KEY(1, KMID)
  CALL PEXIT
9611 FORMAT(' ***** ERROR ON SUB.XREAD : B. STRESS COUNT EXCEED ', 15)
C
9620 CONTINUE
  WRITE (LU6, 9621) MATID
  CALL PEXIT
9621 FORMAT(' ***** ERROR ON SUB.XREAD : ILLEGAL MATERIAL DATA ', 15)
C
  END
  SUBROUTINE YFDE (KFIELD, LDE, E, POI, DE)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET ISOTROPIC ELASTIC STRESS-STRAIN MATRIX 'DE'
C   PROGRAMED BY K.GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C   LDE     D. ARG (INT)   : FIRST ARRAY LENGTH OF 'DE'
C   E      D. ARG (REAL*8) : ELASTIC (YOUNG'S) MODULAS
C   POI    D. ARG (REAL*8) : POISSON'S RATIO
C-----
C * OUTPUT VARIABLES
C   DE(1, J) D. ARG (REAL*8) : ELASTIC STRESS-STRAIN MATRIX

```

```

C-----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION DE(LDE,*)
C
C ***** BRANCH BY STRESS FIELD
C
  GOTO ( 100 , 200 , 300 , 400 , 500 ,
        600 , 700 , 800 , 900 , 1000 , 1100 ) , KFIELD
C
C          .. UNDEFINED STRESS FIELD ..
C
  WRITE(6,*) ' ***** ERROR ON SUB.YFDE: ILLEGAL STRESS FIELD '
              , KFIELD
  CALL PEXIT
  STOP
C          .. UNIAXIAL ..
100 CONTINUE
  DE(1,1) = E
  GOTO 5000
C          .. PLANE STRESS ..
200 CONTINUE
  A1 = E/(1.0-POI**2)
  A2 = A1*POI
  A3 = E/(2.0*(1.0+POI))
  DE(1,1) = A1
  DE(2,1) = A2
  DE(3,1) = 0.0
  DE(1,2) = A2
  DE(2,2) = A1
  DE(3,2) = 0.0
  DE(1,3) = 0.0
  DE(2,3) = 0.0
  DE(3,3) = A3
  GOTO 5000
C          .. 3-DIMENSIONAL ..
300 CONTINUE
  A1 = E*(1.0-POI)/(1.0+POI)/(1.0-2.0*POI)
  A2 = E* POI / (1.0+POI)/(1.0-2.0*POI)
  A3 = E/(2.0*(1.0+POI))
  DE(1,1) = A1
  DE(2,1) = A2
  DE(3,1) = A2
  DE(4,1) = 0.0
  DE(5,1) = 0.0
  DE(6,1) = 0.0

```

```

  DE(1,2) = A2
  DE(2,2) = A1
  DE(3,2) = A2
  DE(4,2) = 0.0
  DE(5,2) = 0.0
  DE(6,2) = 0.0
  DE(1,3) = A2
  DE(2,3) = A2
  DE(3,3) = A1
  DE(4,3) = 0.0
  DE(5,3) = 0.0
  DE(6,3) = 0.0
  DE(1,4) = 0.0
  DE(2,4) = 0.0
  DE(3,4) = 0.0
  DE(4,4) = A3
  DE(5,4) = 0.0
  DE(6,4) = 0.0
  DE(1,5) = 0.0
  DE(2,5) = 0.0
  DE(3,5) = 0.0
  DE(4,5) = 0.0
  DE(5,5) = A3
  DE(6,5) = 0.0
  DE(1,6) = 0.0
  DE(2,6) = 0.0
  DE(3,6) = 0.0
  DE(4,6) = 0.0
  DE(5,6) = 0.0
  DE(6,6) = A3
  GOTO 5000
C          .. AXIS SYMMETRICAL ..
400 CONTINUE
  A1 = E*(1.0-POI)/(1.0+POI)/(1.0-2.0*POI)
  A2 = E* POI / (1.0+POI)/(1.0-2.0*POI)
  A3 = E/(2.0*(1.0+POI))
  DE(1,1) = A1
  DE(2,1) = A2
  DE(3,1) = A2
  DE(4,1) = 0.0
  DE(1,2) = A2
  DE(2,2) = A1
  DE(3,2) = A2
  DE(4,2) = 0.0
  DE(1,3) = A2
  DE(2,3) = A2

```

DE(3, 3) = A1
 DE(4, 3) = 0.0
 DE(1, 4) = 0.0
 DE(2, 4) = 0.0
 DE(3, 4) = 0.0
 DE(4, 4) = A3
 GOTO 5000

C .. PLANE STRAIN ..
 500 CONTINUE
 GOTO 400

C .. BEAM ..

600 CONTINUE
 A3 = E/(2.0*(1.0+PO1))
 DE(1, 1) = E
 DE(2, 1) = 0.0
 DE(3, 1) = 0.0
 DE(1, 2) = 0.0
 DE(2, 2) = A3
 DE(3, 2) = 0.0
 DE(1, 3) = 0.0
 DE(2, 3) = 0.0
 DE(3, 3) = A3
 GOTO 5000

C .. SHELL ..

700 CONTINUE
 A1 = E/(1.0-PO1**2)
 A2 = A1*PO1
 A3 = E/(2.0*(1.0+PO1))
 A4 = A3*5.000/6.000
 DE(1, 1) = A1
 DE(2, 1) = A2
 DE(3, 1) = 0.0
 DE(4, 1) = 0.0
 DE(5, 1) = 0.0
 DE(1, 2) = A2
 DE(2, 2) = A1
 DE(3, 2) = 0.0
 DE(4, 2) = 0.0
 DE(5, 2) = 0.0
 DE(1, 3) = 0.0
 DE(2, 3) = 0.0
 DE(3, 3) = A3
 DE(4, 3) = 0.0
 DE(5, 3) = 0.0
 DE(1, 4) = 0.0
 DE(2, 4) = 0.0

DE(3, 4) = 0.0
 DE(4, 4) = A4
 DE(5, 4) = 0.0
 DE(1, 5) = 0.0
 DE(2, 5) = 0.0
 DE(3, 5) = 0.0
 DE(4, 5) = 0.0
 DE(5, 5) = A4
 GOTO 5000

C .. THIN AXIS SYM. SHELL ..

800 CONTINUE
 A1 = E/(1.0-PO1**2)
 A2 = A1*PO1
 DE(1, 1) = A1
 DE(2, 1) = A2
 DE(1, 2) = A2
 DE(2, 2) = A1
 GOTO 5000

C .. ELBOW ..

900 CONTINUE
 GOTO 200

C .. CONTACT (NOT SUPPORTED) ..

1000 CONTINUE
 GOTO 9000

C .. AXIS SYMMETRICAL SHELL ..

1100 CONTINUE
 A1 = E/(1.0-PO1**2)
 A2 = A1*PO1
 A4 = E/(2.0*(1.0+PO1))*5.000/6.000
 DE(1, 1) = A1
 DE(2, 1) = A2
 DE(3, 1) = 0.0
 DE(1, 2) = A2
 DE(2, 2) = A1
 DE(3, 2) = 0.0
 DE(1, 3) = 0.0
 DE(2, 3) = 0.0
 DE(3, 3) = A4
 GOTO 5000

C BRANCH END

5000 CONTINUE

C RETURN

C .. ERROR ..

9000 CONTINUE
 WRITE(6, *) ' ***** ERROR ON SUB.YFDE : ILLEGAL KFIELD '

```

                                KFIELD
      CALL PEXIT
      STOP
      END
      SUBROUTINE YFDEI (NCOMP, NNOR, LDEI, E, POI, DEI)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET ISOTROPIC ELASTIC STRESS-STRAIN INVERSE MATRIX 'DE'-INV
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP  D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR   D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   LDEI   D. ARG (INT)   : FIRST ARRAY LENGTH OF 'DE'-INV
C   E      D. ARG (REAL*8) : ELASTIC (YOUNG'S) MODULAS
C   POI    D. ARG (REAL*8) : POISSON'S RATIO
C-----
C * OUTPUT VARIABLES
C   DEI(I, J) D. ARG (REAL*8) : INV. OF ELASTIC STRESS-STRAIN MATRIX
C-----
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION DEI(LDEI,*)
C
C ***** 0 DIV. TRAP
C
      IF ( E.EQ.0.0 ) GOTO 9010
C
C ***** CLEAR MATRIX
C
      DO 11 J=1, NCOMP
        DO 10 I=1, NCOMP
          DEI(I, J) = 0.0
        10 CONTINUE
      11 CONTINUE
C
C ***** NORMAL COMPONENTS
C
      GOTO ( 100 , 200 , 300 ) NNOR
C
      WRITE(6,*) ' ***** ERROR ON SUB. YFDEI : ILLEGAL NNOR ', NNOR
      CALL PEXIT
      STOP
C
      .. 1-NORMAL
100 CONTINUE
      DEI(1,1) = 1.0 / E

```

```

      GOTO 500
C
      .. 2-NORMAL
200 CONTINUE
      DEI(1,1) = 1.0 / E
      DEI(2,2) = DEI(1,1)
      DEI(1,2) = -POI*DEI(1,1)
      DEI(2,1) = DEI(1,2)
      GOTO 500
C
      .. 3-NORMAL
300 CONTINUE
      DEI(1,1) = 1.0 / E
      DEI(2,2) = DEI(1,1)
      DEI(3,3) = DEI(1,1)
      DEI(1,2) = -POI*DEI(1,1)
      DEI(1,3) = DEI(1,2)
      DEI(2,1) = DEI(1,2)
      DEI(2,3) = DEI(1,2)
      DEI(3,1) = DEI(1,2)
      DEI(3,2) = DEI(1,2)
      GOTO 500
C
      500 CONTINUE
C
C ***** SHEAR COMPONENTS
C
      DO 600 I=NNOR+1, NCOMP
        DEI(I,1) = 2.0*(1.0+POI)/E
      600 CONTINUE
C
      RETURN
C
      .. ERROR ..
9010 CONTINUE
      WRITE(6,*) ' ***** ERROR ON SUB. YFDEI : 0 ELASTIC MODU. ', E
      CALL PEXIT
      STOP
      END
      SUBROUTINE YFDEP ( KFIELD , NCOMP , LDEP , E , POI ,
      HD , DFDS , DEP )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET ISOTROPIC ELASTOPLASTIC STRESS-STRAIN MATRIX 'DEP'
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT

```

```

C   LDE      D. ARG (INT)   : FIRST ARRAY LENGTH OF 'DEP'
C   E        D. ARG (REAL*8) : ELASTIC (YOUNG'S) MODULAS
C   POI      D. ARG (REAL*8) : POISSON'S RATIO
C   HD       D. ARG (REAL*8) : PLASTIC HARDENING COEFFICIENT
C   DFDS(I)  D. ARG (REAL*8) : FLOW VECTOR (DF/DS)
C-----
C * OUTPUT VARIABLES
C   DEP (I, J) D. ARG (REAL*8) : ELASTOPLASTIC STRESS-STRAIN MATRIX
C-----
C * CALLED SUBROUTINES
C   YFDE      : GET ELASTIC STRESS-STRAIN MATRIX 'DE'
C   YMAB      : MATRIX MULTIPLY ROUTINE ( A * B )
C   YMBB      : VECOR MULTIPLY ROUTINE ( (B)T * B )
C-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION DFDS(*), DEP(LDEP,*)
C   DIMENSION DE(6,6), DF(6)
C
C ***** GET 'DE'
C
C   CALL YFDE (KFIELD , 6 , E , POI , DE )
C
C ***** 'DF' = 'DE' * 'DFDS'
C
C   CALL YMAB (NCOMP, NCOMP, 6, DE, DFDS, DF)
C
C ***** CAL. DENOMINATOR
C
C   CALL YMBB (NCOMP, DF, DFDS, FDF)
C
C   DENOM = HD + FDF
C
C ***** CAL. ELASTOPLASTIC D 'DEP'
C
C   DO 110 I=1, NCOMP
C     DO 100 J=1, NCOMP
C       DEP (I, J) = DE (I, J) - DF (I)*DF (J)/DENOM
C000 CONTINUE
C110 CONTINUE
C
C   RETURN
C   END
C   SUBROUTINE YFDKDT (NCOMP, NNOR, SIG, AL, AKAP, DT, DKAPDT)
C-----
C * FUNCTION OF THIS SUBROUTINE

```

```

C   CALCULATE TEMPERETURE DIFFERENTIAL OF YIELD SURFACE
C   (INCLUDE OVER YIELD SURFACE CORRECTION)
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP     D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR      D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   SIG(I)    D. ARG (REAL*8) : STRESS VECTOR
C   AL (I)    D. ARG (REAL*8) : BACK STRESS VECTOR
C   AKAP      D. ARG (REAL*8) : YIELD STRESS AT NEXT STEP
C   DT        D. ARG (REAL*8) : TEMPERATURE INCREMENT
C-----
C * OUTPUT VARIABLES
C   DKAPDT    D. ARG (REAL*8) : TEMPERATURE DIFF. OF YIELD STRESS
C-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION SIG(*), AL (*)
C
C ***** NO TEMPERATURE INC. TRAP
C
C   IF (DT.EQ.0.0) THEN
C     DKAPDT = 0.0
C     RETURN
C   ENDIF
C
C ***** GET YIELD STRESS AT PREVIOUS STEP (INCLUDE OVER VALUE)
C
C   CALL YFSB2 (NCOMP, NNOR, SIG, AL, AKAP1)
C
C ***** DKAP/DT INCLUDE MODIFICATION OF OVER YIELD SURFACE
C
C   DKAPDT = ( AKAP - AKAP1 ) / DT
C
C   RETURN
C   END
C   SUBROUTINE YFDSE ( KFIELD, NCOMP , NNOR , E1 , POI1 ,
C                     E2 , POI2 , ALP , DT , EPSE ,
C                     DEPS , DSIG )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   GET STRESS INCREMENT ON ELASTIC
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD    D. ARG (INT)   : FINAS STRESS FIELD CODE

```

```

C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   E1      D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   PO11    D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   PO12    D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   ALP     D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   DT      D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   EPSE(I) D. ARG (REAL*8) : ELASTIC STRAIN
C   DEPS(I) D. ARG (REAL*8) : STRAIN INCREMENT
-----
C * OUTPUT VARIABLES
C   DSIG(I) D. ARG (REAL*8) : STRESS INCREMENT
-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION EPSE(*), DEPS(*), DSIG(*)
C   DIMENSION DE(6,6), DGE(6)
C
C ***** GET ELASTIC STRESS-STRAIN MATRIX 'DE'
C
C   CALL YFDE ( KFIELD, 6 , E2 , PO12 , DE )
C
C ***** SET ELASTIC PSEUDO STRESS INCREMENT VECTOR 'DGE'
C
C   CALL YFGE ( KFIELD, NCOMP , NNOR , E1 , PO11 ,
C             E2 , PO12 , ALP , DT , EPSE ,
C             DGE )
C
C ***** CONSTITUTIVE EQUATION ( DSIG = DE*DEPS + DGE )
C
C   DO 110 I=1,NCOMP
C     DSIG(I) = 0.0
C     DO 100 J=1,NCOMP
C       DSIG(I) = DSIG(I) + DE(I,J)*DEPS(J)
C   100 CONTINUE
C     DSIG(I) = DSIG(I) + DGE(I)
C   110 CONTINUE
C
C   RETURN
C   END
C   SUBROUTINE YFDSEP( KFIELD , NCOMP , NNOR , E1 , PO11 ,
C                   E2 , PO12 , ALP , HD , DT ,
C                   DKAPDT , DALDT , DFDS , EPSE , DEPS ,
C                   DSIG )
-----

```

```

C * FUNCTION OF THIS SUBROUTINE
C   GET STRESS INCREMENT ON PLASTIC
C                                     PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
-----
C * INPUT VARIABLES
C   KFIELD D. ARG (INT)   : FINAS STRESS FIELD CODE
C   NCOMP  D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR   D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   E1     D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   PO11   D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   E2     D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   PO12   D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   ALP    D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   HD     D. ARG (REAL*8) : PLASTIC HARDENING COEFF.
C   DT     D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C   DALDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL)/DT
C   DFDS(I) D. ARG (REAL*8) : FLOW VECTOR (DF/DS)
C   EPSE(I) D. ARG (REAL*8) : ELASTIC STRAIN
C   DEPS(I) D. ARG (REAL*8) : STRAIN INCREMENT
-----
C * OUTPUT VARIABLES
C   DSIG(I) D. ARG (REAL*8) : STRESS INCREMENT
-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION DALDT(*), DFDS(*), EPSE(*), DEPS(*), DSIG(*)
C   DIMENSION DEP(6,6), DGEP(6)
C
C ***** GET ELASTOPLASTIC STRESS-STRAIN MATRIX 'DEP'
C
C   CALL YFDEP ( KFIELD , NCOMP , 6 , E2 , PO12 ,
C             HD , DFDS , DEP )
C
C ***** SET ELASTOPLASTIC PSEUDO STRESS INCREMENT VECTOR 'DGEP'
C
C   CALL YFGEP ( KFIELD , NCOMP , 6 , E1 , PO11 ,
C             E2 , PO12 , ALP , HD , DT ,
C             DKAPDT , DALDT , DFDS , EPSE , DGEP )
C
C ***** CONSTITUTIVE EQUATION ( DSIG = DE*DEPS + DGE )
C
C   DO 110 I=1,NCOMP
C     DSIG(I) = 0.0
C     DO 100 J=1,NCOMP
C       DSIG(I) = DSIG(I) + DEP(I,J)*DEPS(J)

```

```

100 CONTINUE
      DSIG(I) = DSIG(I) + D6EP(I)
110 CONTINUE
C
      RETURN
      END
      SUBROUTINE YFEPSE (NCOMP, NNOR, E, POI, SIG, EPSE)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET ELASTIC STRAIN
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C   E D. ARG (REAL*8) : ELASTIC (YOUNG'S) MODULAS
C   POI D. ARG (REAL*8) : POISSON'S RATIO
C   SIG (I) D. ARG (REAL*8) : STRESS COMPONENTS
C-----
C * OUTPUT VARIABLES
C   EPSE(I) D. ARG (REAL*8) : ELASTIC STRAIN BY SIG(I)
C-----
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION SIG(*), EPSE(*)
      DIMENSION DEINV(6, 6)
C
C ***** GET 'DE' - INV MATRIX
C
      CALL YFDEI (NCOMP, NNOR, 6, E, POI, DEINV)
C
C **** 'EPSE' = 'DE' - INV * 'SIG'
C
      CALL YMAB (NCOMP, NCOMP, 6, DEINV, SIG, EPSE)
C
      RETURN
      END
      SUBROUTINE YFEPSP (NCOMP, NNOR, E, POI, ALP, DT, DEPS, DSIG, DEPSP)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   CALCULATE PLASTIC STRAIN-INCREMENT BY STRAIN DEFINITION
C   D'EPS' = D'EPS-E' + D'EPS-T' + D'EPS-P'
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT

```

```

C   NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C   E D. ARG (REAL*8) : ELASTIC (YOUNG'S) MODULAS
C   POI D. ARG (REAL*8) : POISSON'S RATIO
C   ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   DEPS (I) D. ARG (REAL*8) : (TOTAL) STRAIN INCREMENT
C   DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C-----
C * OUTPUT VARIABLES
C   DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C-----
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION DSIG(*), DEPS(*), DEPSP(*)
      DIMENSION DEPST(6), DEPSE(6)
C
C ***** GET THERMAL STRAIN
C
      DO 100 I=1, NCOMP
         IF (I.LE. NNOR) THEN
            DEPST(I) = ALP*DT
         ELSE
            DEPST(I) = 0.0
         ENDIF
      100 CONTINUE
C
C ***** GET ELASTIC STRAIN
C
      CALL YFEPSE (NCOMP, NNOR, E, POI, DSIG, DEPSE)
C
C ***** CAL. PLASTIC STRAIN INCREMENT
C
      DO 200 I=1, NCOMP
         DEPSP(I) = DEPS(I) - DEPST(I) - DEPSE(I)
      200 CONTINUE
C
      RETURN
      END
      SUBROUTINE YFGAM (NCOMP, NNOR, SIG, DSIG, AL, SIGY, GAM)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   CALCULATE BREAK POINT RATIO 'GAM' AT ELASTIC TO PLASTIC
C   (FOR MISES TYPE YIELD FUNCTION)
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES

```

```

C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   SIG (1) D. ARG (REAL*8) : STRESS VECTOR
C   DSIG (1) D. ARG (REAL*8) : STRESS INCREMENT VECTOR
C   AL (1)  D. ARG (REAL*8) : BACK STRESS VECTOR
C   SIGY    D. ARG (REAL*8) : YIELD STRESS

```

```
-----
```

```

C * OUTPUT VARIABLES
C   GAM      D. ARG (REAL*8) : RATIO OF BREAK POINT

```

```
-----
```

```

C * LOCAL VARIABLES
C   SS3 (1)  D. ARG (REAL*8) : DEVIATORIC STRESS AT 3-D SPACE
C   DSS3 (1) D. ARG (REAL*8) : DEVIATORIC STRESS INC. AT 3-D SPACE
C   AL3 (1)  D. ARG (REAL*8) : BACK STRESS AT 3-D SPACE

```

```
-----
```

```

C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION SIG(*),DSIG(*),AL (*)
  DIMENSION SS3(6),DSS3(6),AL3(6)

```

```

C
C ***** EXTEND EACH STRESS TO 3D-FIELD

```

```

C .. GET MEAN STRESS

```

```

  SSM = 0.0
  DSSM = 0.0
  DO 100 I=1,NNOR
    SSM = SSM + SIG(I)
    DSSM = DSSM + DSIG(I)
  100 CONTINUE
  SSM = SSM / 3.000
  DSSM = DSSM / 3.000

```

```

C .. NORMAL COMPONENTS

```

```

  DO 200 I=1,3
    IF (I.LE.NNOR) THEN
      SS3(I) = SIG(I) - SSM
      DSS3(I) = DSIG(I) - DSSM
      AL3(I) = AL(I)
    ELSE
      SS3(I) = - SSM
      DSS3(I) = - DSSM
    ENDIF
  200 CONTINUE

```

```

C
  IF (NNOR.EQ.1) THEN
    AL3(2) = -AL(1)/2.0
    AL3(3) = -AL(1)/2.0
  ENDIF

```

```

C
  IF (NNOR.EQ.2) THEN
    AL3(3) = -AL(1)-AL(2)
  ENDIF

```

```

C .. SHEAR COMPONENTS

```

```

  J = NNOR+1
  DO 300 I=4,6
    IF (J.LE.NCOMP) THEN
      SS3(I) = SIG(J)
      DSS3(I) = DSIG(J)
      AL3(I) = AL(J)
    ELSE
      SS3(I) = 0.0
      DSS3(I) = 0.0
      AL3(I) = 0.0
    ENDIF
    J = J + 1
  300 CONTINUE

```

```

C
C ***** SET CONSTANTS BY TENSOR MULTIPLY

```

```

  AA = 0.0
  B1 = 0.0
  B2 = 0.0
  C1 = 0.0
  C2 = 0.0
  C3 = 0.0

```

```

C
  DO 400 I=1,3
    AA = AA + DSS3(I) * DSS3(I)
    B1 = B1 + SS3(I) * DSS3(I)
    B2 = B2 + DSS3(I) * AL3(I)
    C1 = C1 + SS3(I) * SS3(I)
    C2 = C2 + SS3(I) * AL3(I)
    C3 = C3 + AL3(I) * AL3(I)
  400 CONTINUE

```

```

C
  DO 410 I=4,6
    AA = AA + ( DSS3(I) * DSS3(I) ) * 2.0
    B1 = B1 + ( SS3(I) * DSS3(I) ) * 2.0
    B2 = B2 + ( DSS3(I) * AL3(I) ) * 2.0
    C1 = C1 + ( SS3(I) * SS3(I) ) * 2.0
    C2 = C2 + ( SS3(I) * AL3(I) ) * 2.0
    C3 = C3 + ( AL3(I) * AL3(I) ) * 2.0
  410 CONTINUE

```

```

C

```



```

C ***** GET 'GAM' PROCEDURE BY OGASAWARA
C
C   COF = 1.5D0
C
C   BB = B1 - B2
C   CC = C1 - 2.0*C2 + C3 - 1.0/COF*SIGY**2
C
C   IF (AA.EQ.0)   GO TO 10
C   DD  = BB**2 - AA*CC
C
C   ADD = ABS(DD)
C   IF (ADD.LT.1.0D-50) DD = 0.0
C   IF (DD.LT.0.0)   GO TO 30
C   DD  = SQRT(DD)
C   GAM1 = (-BB+DD)/AA
C   GAM2 = (-BB-DD)/AA
C
C   IF ((GAM1.GT.1.0).AND.(GAM1.LT.1.0001D0)) GAM1 = 1.0
C   IF ((GAM1.GT.1.0).OR.(GAM1.LT.0.0)) GO TO 20
C   GAM = GAM1
C   GO TO 999
C
C 20 CONTINUE
C   IF ((GAM2.GT.1.0).AND.(GAM1.LT.1.0001D0)) GAM1 = 1.0
C   IF ((GAM2.GT.1.0).OR.(GAM2.LT.0.0)) GO TO 30
C   GAM = GAM2
C   GO TO 999
C
C 10 CONTINUE
C   IF (BB.EQ.0.0) GO TO 30
C   GAM1 = -CC/(2.0*BB)
C   IF ((GAM1.GT.1.0).OR.(GAM1.LT.0.0)) GO TO 30
C   GAM = GAM1
C   GO TO 999
C
C 30 CONTINUE
C   GAM = 0.0
C 999 CONTINUE
C   RETURN
C   END
C   SUBROUTINE YFGE ( KFIELD, NCOMP, NNOR, E1, PO11,
C                   E2, PO12, ALP, DT, EPSE,
C                   DGE )
C -----
C * FUNCTION OF THIS SUBROUTINE
C   SET PSEUDO STRESS INCREMENT VECTOR 'DGE' ON ELASTIC

```

```

C                                     PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C -----
C * INPUT VARIABLES
C   KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C   NCOMP  D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR   D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C   E1     D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   PO11   D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   E2     D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   PO12   D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   ALP    D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   DT     D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   EPSE(I) D. ARG (REAL*8) : ELASTIC STRAIN
C -----
C * OUTPUT VARIABLES
C   DGE (I) D. ARG (REAL*8) : PSEUDO STRESS INCREMENT VECTOR
C -----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION EPSE(*), DGE(*)
C   DIMENSION DEDT(6,6), DE2(6,6), ALPV(6)
C
C   CALL YFDE (KFIELD, 6, E2, PO12, DE2)
C
C ***** TEMPERATURE DIFFERENE OF 'DE' MATRIX 'DE/DT'
C
C   IF ( ( DT.EQ.0.0 ) .OR.
C        ( (E1.EQ.E2).AND.(PO11.EQ.PO12) ) ) THEN
C
C       DO 110 J=1,NCOMP
C         DO 100 I=1,NCOMP
C           DEDT(I,J) = 0.0
C 100    CONTINUE
C 110    CONTINUE
C
C   ELSE
C
C       CALL YFDE (KFIELD, 6, E1, PO11, DEDT)
C
C       DO 210 J=1,NCOMP
C         DO 200 I=1,NCOMP
C           DEDT(I,J) = ( DE2(I,J)-DEDT(I,J) ) / DT
C 200    CONTINUE
C 210    CONTINUE
C
C   ENDIF

```

```

C
C ***** THERMAL EXPANSION VECTOR
C
  DO 300 I=1, NCOMP
    IF ( I.LE. NNOR ) THEN
      ALPV(I) = ALP
    ELSE
      ALPV(I) = 0.0
    ENDIF
  300 CONTINUE
C
C ***** PSEUDO STRESS INCREMENT
C
  DO 410 I=1, NCOMP
    DGE(I) = 0.0
    DO 400 J=1, NCOMP
      DGE(I) = DGE(I) + DEDT(I, J)*EPSE(J) - DE2(I, J)*ALPV(J)
    400 CONTINUE
    DGE(I) = DGE(I)*DT
  410 CONTINUE
C
  RETURN
  END
  SUBROUTINE YFGEP ( KFIELD, NCOMP, NNOR, E1, PO11,
    E2, PO12, ALP, HD, DT,
    DKAPDT, DALDT, DFDS, EPSE, DGEP )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET PSEUDO STRESS INCREMENT VECTOR 'DGE' ON PLASTIC
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   E1      D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   PO11    D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   PO12    D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   ALP     D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   HD      D. ARG (REAL*8) : PLASTIC HARDENING COEFF.
C   DT      D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   DKAPDT  D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C   DALDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL)/DT
C   DFDS (I) D. ARG (REAL*8) : FLOW VECTOR (DF/DS)

```

```

C   EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C-----
C * OUTPUT VARIABLES
C   DGEP (I) D. ARG (REAL*8) : PSEUDO STRESS INCREMENT VECTOR
C-----
C * CALLED SUBROUTINES
C   YFDE      : GET ELASTIC STRESS-STRAIN MATRIX 'DE'
C   YMAB      : MATRIX MULTIPLY ROUTINE ( A * B )
C   YMBB      : VECOR MULTIPLY ROUTINE ( (B)T * B )
C-----
C
  IMPLICIT REAL*8 (A-H, O-Z)
  DIMENSION DALDT(*), DFDS(*), EPSE(*), DGEP(*)
  DIMENSION DEDT(6, 6), DE2(6, 6), ALPV(6), DF(6), DEPX(6, 6), DEPE(6),
    TERM1(6), TERM2(6), TERM3(6)
C
  CALL YFDE (KFIELD, 6, E2, PO12, DE2 )
C
C ***** TEMPERATURE DIFFERENE OF 'DE' MATRIX 'DE/DT'
C
  IF ( ( DT.EQ.0.0 ) .OR.
    ( E1.EQ.E2 ) .AND. ( PO11.EQ.PO12 ) ) THEN
C
  DO 110 J=1, NCOMP
    DO 100 I=1, NCOMP
      DEDT(I, J) = 0.0
    100 CONTINUE
  110 CONTINUE
C
  ELSE
C
  CALL YFDE (KFIELD, 6, E1, PO11, DEDT )
C
  DO 210 I=1, NCOMP
    DO 200 J=1, NCOMP
      DEDT(I, J) = ( DE2(I, J)-DEDT(I, J) ) / DT
    200 CONTINUE
  210 CONTINUE
C
  ENDIF
C
C ***** 'DF' = 'DE' * 'DFDS'
C
  CALL YMAB (NCOMP, NCOMP, 6, DE2, DFDS, DF)
C
C ***** CAL. DENOMINATOR

```

```

C
C CALL YMBB(NCOMP, DF, DFDS, FDF)
C
C DENOM = HD + FDF
C
C ***** CAL. ELASTOPLASTIC D 'DEP'
C
C DO 1100 I=1, NCOMP
C DO 1000 J=1, NCOMP
C DEPX(I, J) = DE2(I, J) - DF(I)*DF(J)/DENOM
1000 CONTINUE
1100 CONTINUE
C
C ***** TERM-1 .. THERMAL EXPANSION
C
C *** THERMAL EXPANSION VECTOR
C
C DO 1200 I=1, NCOMP
C IF ( I.LE.NNOR ) THEN
C ALPV(I) = ALP
C ELSE
C ALPV(I) = 0.0
C ENDIF
1200 CONTINUE
C
C *** GET TERM-1
C
C CALL YMAB(NCOMP, NCOMP, 6, DEPX, ALPV, TERM1)
C
C ***** TERM-2 .. CHANGE OF ELASTIC MODU.
C
C *** 'DEPE' = D(DE)/DT * EPSE
C
C CALL YMAB(NCOMP, NCOMP, 6, DEDT, EPSE, DEPE)
C
C *** FDE = (DF/DS)T * D(DE)/DT * EPSE
C
C CALL YMBB(NCOMP, DFDS, DEPE, FDE)
C
C *** GET TERM-2
C
C DO 2000 I=1, NCOMP
C TERM2(I) = DEPE(I) - DF(I)*FDE/DENOM
2000 CONTINUE
C
C ***** TERM-3 .. CHANGE OF ST. VARI

```

```

C
C *** FADT = (DF/DS)T * D(AL)/DT
C
C CALL YMBB(NCOMP, DFDS, DALDT, FADT)
C
C *** GET TERM-3
C
C DO 3000 I=1, NCOMP
C TERM3(I) = ( DF(I)*FADT + DF(I)*DKAPDT ) / DENOM
3000 CONTINUE
C
C ***** PSEUDO STRESS INCREMENT
C
C DO 5000 I=1, NCOMP
C DGEPI(I) = ( TERM3(I) + TERM2(I) - TERM1(I) ) * DT
5000 CONTINUE
C
C RETURN
C END
C SUBROUTINE YFIELD(IECN, IECOD, KFIELD, NCOMP, NNOR)

```

C * FUNCTION OF THIS SUBROUTINE
C SET STRESS FIELD KEY 'KFIELD' (FINAS INTERNAL CODE)
C AND STRESS COMPONENTS COUNT 'NCOMP', NORMAL COUNT 'NNOR'
C PROGRAMED BY K.GARATANI ON NOV. 1995 (VER 1.0)

C * INPUT VARIABLES
C IECN D. ARG (INT) : ELEMENT EXTERNAL ID. NO.
C IECOD D. ARG (INT) : ELEMENT IDENTIFICATION CODE BY FINAS

C * OUTPUT VARIABLES
C KFIELD D. ARG (INT) : STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT

C * COMMENT
C KFIELD CODE AS BELOW
C

KFIELD	NCOMP	NNOR	- STRESS FIELD -
1	1	1	UNIAXIAL STRESS
2	3	2	PLANE STRESS
3	6	3	3-DIMENSIONAL
4	4	3	AXIS SYMMETRICAL
5	4	3	PLANE STRAIN
6	3	1	BEAM
7	5	2	SHELL

```

C      8      2      2      THIN AXIS SYMMETRICAL SHELL
C      9      3      2      ELBOW OR NAXSHEL
C     10      8      0      CONTACT
C     11      3      2      AXIS SYMMETRICAL SHELL
C-----
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C ***** BRANCH BY ELEMENT CODE (V13.0 ED01)
C
C      GOTO ( 40 , 10 , 60 , 40 , 20 , 20 , 30 , 20 , 70 , 20 ,
.          70 , 90 , 80 , 60 , 60 , 30 , 70 , 20 , 40 , 20 ,
.          20 , 40 , 80 , 50 , 50 , 60 , 40 , 20 , 100 , 30 ,
.          30 , 30 , 90 , 30 , 30 , 60 , 70 , 9 , 20 , 70 ,
.          70 , 20 , 20 , 20 , 9 , 9 , 9 , 9 , 9 , 90 ,
.          9 , 90 , 9 , 60 , 20 , 20 , 50 ) IECOD
C
C      .. UNDEFINED CODE ..
C
C 9 CONTINUE
C      WRITE(6,*) ' ***** ERROR ON SUB.YFIELD : ILLEGAL ELEMENT CODE '
C      , IECOD , IEXN
C
C      CALL PEXIT
C      STOP
C
C      .. UNIAXIAL STRESS ..
C 10 CONTINUE
C      KFIELD = 1
C      NCOMP = 1
C      NNOR = 1
C      GOTO 1000
C
C      .. PLANE STRESS ..
C 20 CONTINUE
C      KFIELD = 2
C      NCOMP = 3
C      NNOR = 2
C      GOTO 1000
C
C      .. 3-DIMENSIONAL ..
C 30 CONTINUE
C      KFIELD = 3
C      NCOMP = 6
C      NNOR = 3
C      GOTO 1000
C
C      .. AXIS SYMMETRICAL ..
C 40 CONTINUE
C      KFIELD = 4
C      NCOMP = 4
C      NNOR = 3

```

```

C      GOTO 1000
C
C 50 CONTINUE
C      KFIELD = 5
C      NCOMP = 4
C      NNOR = 3
C      GOTO 1000
C
C 60 CONTINUE
C      KFIELD = 6
C      NCOMP = 3
C      NNOR = 1
C      GOTO 1000
C
C 70 CONTINUE
C      KFIELD = 7
C      NCOMP = 5
C      NNOR = 2
C      GOTO 1000
C
C 80 CONTINUE
C      KFIELD = 8
C      NCOMP = 2
C      NNOR = 2
C      GOTO 1000
C
C 90 CONTINUE
C      KFIELD = 9
C      NCOMP = 3
C      NNOR = 2
C      GOTO 1000
C
C 100 CONTINUE
C      KFIELD = 10
C      NCOMP = 8
C      NNOR = 0
C      GOTO 1000
C
C 110 CONTINUE
C      KFIELD = 11
C      NCOMP = 3
C      NNOR = 2
C      GOTO 1000
C
C      ..... BRANCH END .....
C 1000 CONTINUE
C

```

```

.. PLANE STRAIN ..
.. BEAM ..
.. SHELL ..
.. THIN AXIS SYM. SHELL ..
.. ELBOW ..
.. CONTACT ..
.. AXIS SYMMETRICAL SHELL ..

```

```

RETURN
C
END
SUBROUTINE YFLOW2(NCOMP, NNOR, SIG, AL, FLOW)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET FLOW VECTOR (MISES TYPE WITH BACK STRESS)
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   SIG (I) D. ARG (REAL*8) : STRESS VECTOR
C   AL (I)  D. ARG (REAL*8) : BACK STRESS VECTOR
C-----
C * OUTPUT VARIABLES
C   FLOW(I) D. ARG (REAL*8) : FLOW VECTOR (DF/DS, VECTOR TYPE)
C-----
C * LOCAL VARIABLES
C   S (I)   D. ARG (REAL*8) : DEVIATORIC STRESS AT 3-D SPACE
C   AL3 (I) D. ARG (REAL*8) : BACK STRESS AT 3-D SPACE
C   FLW3(I) D. ARG (REAL*8) : FLOW VECTOR AT 3-D SPACE
C   SBAR    D. ARG (REAL*8) : EQUIVALENT STRESS (MISES TYPE)
C-----
C * FLOW VECTOR WHEN F=SIGB (MISES TYPE)
C   3/2 / SIGB * S(I) : NORMAL COMPONENTS
C   3/2 / SIGB * S(I) * 2 : SHEAR COMPONENTS
C-----
C
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION SIG(*), AL (*), FLOW(*)
DIMENSION S (6), AL3(6), FLW3(6)
C
C ***** GET EQUIVALENT STRESS BY EXTEND 3-D STRESS FIELD
C
DO 100 I3D=1, 6
S (I3D) = 0.0
AL3(I3D) = 0.0
100 CONTINUE
C .. NORMAL COMPONENTS
DO 200 IFLD=1, NNOR
S (IFLD) = SIG(IFLD)
AL3(IFLD) = AL (IFLD)
200 CONTINUE
C .. GET DEVIATORIC STRESS
SM = ( S(1)+S(2)+S(3) )/3.000

```

```

C
DO 300 I3D=1, 3
S (I3D) = S(I3D) - SM
300 CONTINUE
C .. GET OTHER BACK STRESS
IF (NNOR.EQ.1) THEN
AL3( 2) = -AL3(1)/2.000
AL3( 3) = -AL3(1)/2.000
ENDIF
C
IF (NNOR.EQ.2) THEN
AL3( 3) = -(AL3(1)+AL3(2))
ENDIF
C .. SHEAR COMPONENTS
I3D = 4
DO 400 IFLD=NNOR+1, NCOMP
S (I3D) = SIG(IFLD)
AL3(I3D) = AL (IFLD)
I3D = I3D + 1
400 CONTINUE
C .. BACK STRESS EFFECT.
DO 500 I3D=1, 6
S (I3D) = S(I3D) - AL3(I3D)
500 CONTINUE
C .. MISES STRESS
SBAR = SQRT( 3.000/2.000 * ( S(1)**2+S(2)**2+S(3)**2 +
2.000*(S(4)**2+S(5)**2+S(6)**2) ) )
C
C ***** 0 STRESS TRAP
C
IF (SBAR.EQ.0.0) THEN
C
DO 600 IFLD=1, NCOMP
FLOW(IFLD) = 0.0
600 CONTINUE
C
RETURN
C
ENDIF
C .. GET FLOW VECTOR (VECTOR TYPE)
COF = 3.000/2.000 / SBAR
C .. NORMAL COMPONENTS
DO 700 I3D=1, 3
FLW3(I3D) = COF * S (I3D)
700 CONTINUE

```

```

C          . SHEAR COMPONENTS
      DO 710 I3D=4,6
          FLW3(I3D) = COF * S (I3D) * 2.000
      710 CONTINUE
C
C ***** SET FLOW VECTOR FOR EACH STRESS FIELD
C
      DO 800 IFLD=1, NNOR
          FLOW(IFLD) = FLW3(IFLD)
      800 CONTINUE
C
      I3D = 4
      DO 810 IFLD=NNOR+1, NCOMP
          FLOW(IFLD) = FLW3(I3D)
          I3D = I3D + 1
      810 CONTINUE
C
      RETURN
      END
      SUBROUTINE YFNRMV ( KEY , NCOMP , NNOR , VECT , TSRN )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   NORMALIZE VECTOR AND EXPAND 3-D SPACE TENSOR
C   (STRESS FIELD VECTOR INPUT AND 3-D TENSOR OUTPUT)
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KEY      D. ARG (INT)   : DOUBLED SHEAR TERM KEY
C                       =0:NO (STRESS) , =1:YES (STRAIN)
C   NCOMP    D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR     D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   VECT (*) D. ARG (REAL*8) : INPUT STRESS OR STRAIN VECTOR
C                       (INCOMPRESSIBILITY ASSUMED)
C-----
C * OUTPUT VARIABLES
C   TSRN(6) D. ARG (REAL*8) : NORMALIZED TENSOR (3-D TYPE)
C-----
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION VECT(*), TSRN(*)
C
C ***** CLEAR TENSOR
C
      DO 100 I=1,6
          TSRN(I) = 0.0
      100 CONTINUE

```

```

C
C ***** EXTEND TO 3-D STRESS SPACE
C
      IF (NNOR.EQ.1) THEN
          TSRN(1) = VECT(1)
          TSRN(2) = VECT(1) * ( -0.500 )
          TSRN(3) = TSRN(2)
      ELSE IF (NNOR.EQ.2) THEN
          TSRN(1) = VECT(1)
          TSRN(2) = VECT(2)
          TSRN(3) = -( VECT(1)+VECT(2) )
      ELSE IF (NNOR.EQ.3) THEN
          TSRN(1) = VECT(1)
          TSRN(2) = VECT(2)
          TSRN(3) = VECT(3)
      ENDIF
C          .. SHEAR COMPONENTS (DEP. KEY)
      I3D = 4
      DO 200 I=NNOR+1, NCOMP
          IF (KEY.EQ.0) THEN
              TSRN(I3D) = VECT(I)
          ELSE
              TSRN(I3D) = VECT(I) * ( 0.500 )
          ENDIF
          I3D = I3D + 1
      200 CONTINUE
C
C ***** GET TENSOR LENGTH
C
      TLEN = SQRT( TSRN(1)**2 + TSRN(2)**2 + TSRN(3)**2 +
                  ( TSRN(4)**2 + TSRN(5)**2 + TSRN(6)**2 ) * 200 )
C
C ***** NORMALIZE
C
      IF (TLEN.EQ.0.0) THEN
          TSRN(1) = 0.0
          TSRN(2) = 0.0
          TSRN(3) = 0.0
          TSRN(4) = 0.0
          TSRN(5) = 0.0
          TSRN(6) = 0.0
      ELSE
          TSRN(1) = TSRN(1)/TLEN
          TSRN(2) = TSRN(2)/TLEN
          TSRN(3) = TSRN(3)/TLEN
          TSRN(4) = TSRN(4)/TLEN

```

```

      TSRN(5) = TSRN(5)/TLEN
      TSRN(6) = TSRN(6)/TLEN
    ENDIF
  C
  RETURN
  END
  SUBROUTINE YFPBAR(KFIELD, DEPS, DEPBAR)
  C-----
  C * FUNCTION OF THIS SUBROUTINE
  C   SET EQUIVALENT PLASTIC STRAIN (MISES TYPE)
  C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
  C-----
  C * INPUT VARIABLES
  C   KFIELD   D. ARG (INT)   : STRESS FIELD CODE
  C   DEPS(*)  D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT VECTOR
  C-----
  C * OUTPUT VARIABLES
  C   DEPBAR   D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN
  C-----
  C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION DEPS (*)
  C ***** BRANCH BY STRESS FIELD
  C
  GOTO ( 100 , 200 , 300 , 400 , 500 ,
        600 , 700 , 800 , 900 , 1000 , 1100 ) , KFIELD
  C
  C .. UNDEFINED STRESS FIELD ..
  C
  WRITE(6,*) ' ***** ERROR ON SUB. YFPBAR : ILLEGAL STRESS FIELD '
  , KFIELD
  CALL PEXIT
  STOP
  C .. UNIAXIAL ..
  100 CONTINUE
  DEPBAR = ABS(DEPS(1))
  GOTO 5000
  C .. PLANE STRESS ..
  200 CONTINUE
  DEPS3 = -(DEPS(1)+DEPS(2))
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS(2)**2+DEPS3 **2
  . + (DEPS(3)**2
  . )
  . )/2D0 )
  GOTO 5000

```

```

  C .. 3-DIMENSIONAL ..
  300 CONTINUE
  C
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS(2)**2+DEPS(3)**2
  . + (DEPS(4)**2+DEPS(5)**2+DEPS(6)**2)/2D0 )
  . )
  GOTO 5000
  C .. AXIS SYMMETRICAL ..
  400 CONTINUE
  C
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS(2)**2+DEPS(3)**2
  . + (DEPS(4)**2
  . )/2D0 )
  . )
  GOTO 5000
  C .. PLANE STRAIN ..
  500 CONTINUE
  GOTO 400
  C .. BEAM ..
  600 CONTINUE
  DEPS2 = -DEPS(1)/2.0
  DEPS3 = DEPS2
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS2 **2+DEPS3**2
  . + (DEPS(2)**2+DEPS(3)**2
  . )/2D0 )
  . )
  GOTO 5000
  C .. SHELL ..
  700 CONTINUE
  DEPS3 = -(DEPS(1)+DEPS(2))
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS(2)**2+DEPS3 **2
  . + (DEPS(3)**2+DEPS(4)**2+DEPS(5)**2)/2D0 )
  . )
  GOTO 5000
  C .. THIN AXIS SYM. SHELL ..
  800 CONTINUE
  DEPS3 = -(DEPS(1)+DEPS(2))
  DEPBAR = SQRT( 2D0/3D0 *
  . ( DEPS(1)**2+DEPS(2)**2+DEPS3 **2 )
  . )
  GOTO 5000
  C .. ELBOW ..
  900 CONTINUE

```

```

      GOTO 200
C          .. CONTACT (NOT SUPPORTED) ..
1000 CONTINUE
      GOTO 9000
C          .. AXIS SYMMETRICAL SHELL ..
1100 CONTINUE
      GOTO 200
C          ..... BRANCH END .....
5000 CONTINUE
C
      RETURN
C
9000 CONTINUE
      WRITE(6,*) ' ***** ERROR ON SUB. YFPBAR : ILLEGAL KFIELD ',
                KFIELD
      CALL PEXIT
      STOP
C
      END
      SUBROUTINE YFSB2 (NCOMP, NNOR, SIG, AL, SBAR)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET EQUIVALENT STRESS (MISES TYPE WITH BACK STRESS)
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP  D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR   D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   SIG(I) D. ARG (REAL*8) : STRESS VECTOR
C   AL (I) D. ARG (REAL*8) : BACK STRESS VECTOR
C-----
C * OUTPUT VARIABLES
C   SBAR   D. ARG (REAL*8) : EQUIVALENT STRESS (MISES TYPE)
C-----
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION SIG(*), AL (*)
      DIMENSION S (6), AL3(6)
C
C ***** EXTEND 3-D STRESS FIELD
C
      DO 100 I3D=1, 6
          S (I3D) = 0.0
          AL3(I3D) = 0.0
100 CONTINUE
C          .. NORMAL COMPONENTS

```

```

      DO 200 IFLD=1, NNOR
          S (IFLD) = SIG(IFLD)
          AL3(IFLD) = AL (IFLD)
200 CONTINUE
C          .. GET DEVIATORIC STRESS
          SM = ( S(1)+S(2)+S(3) )/3.0D0
C
          DO 300 I3D=1, 3
              S (I3D) = S(I3D) - SM
300 CONTINUE
C          .. GET OTHER BACK STRESS
          IF (NNOR.EQ.1) THEN
              AL3( 2) = -AL3(1)/2.0D0
              AL3( 3) = -AL3(1)/2.0D0
          ENDIF
C
          IF (NNOR.EQ.2) THEN
              AL3( 3) = -(AL3(1)+AL3(2))
          ENDIF
C          .. SHEAR COMPONENTS
          I3D = 4
          DO 400 IFLD=NNOR+1, NCOMP
              S (I3D) = SIG(IFLD)
              AL3(I3D) = AL (IFLD)
              I3D = I3D + 1
400 CONTINUE
C          .. BACK STRESS EFFECT.
          DO 500 I3D=1, 6
              S (I3D) = S(I3D) - AL3(I3D)
500 CONTINUE
C
C ***** GET EQUIBARENT STRESS (MISES TYPE)
C
          SBAR = SQRT( 3.0D0/2.0D0 *(
                S(1)**2+S(2)**2+S(3)**2 +
                2.0D0*(S(4)**2+S(5)**2+S(6)**2) ) )
C
          RETURN
          END
          SUBROUTINE YFVEXP(X, XTBL, YTBL, NL, Y)
C-----
C * FUNCTION OF THIS SUBROUTINE
C   INTERPOLATE OR INTERPOLATE THE Y-VALUE,
C   GIVEN BY X-VALUE TABLE FORMAT
C   PROGRAMED BY K. GARATANI ON MAR. 1991 (VER 1.0)
C-----
C * INPUT VARIABLES

```



```

C      X      D. ARG (REAL*8) : X-VALUE TO GET THE Y-VALUE
C      XTBL(1) D. ARG (REAL*8) : X-VALUE TABLE
C      YTBL(1) D. ARG (REAL*8) : Y-VALUE TABLE
C      NL      D. ARG (INT)   : LENGTH OF X AND Y-VALUE
-----
C * OUTPUT VARIABLES
C      Y      D. ARG (REAL*8) : GETTED Y-VALUE AT X
-----
C * COMMENT
C      X-VALUE TABLE XTBL(1) MUST BE INCREMENTAL OR DECREMENTAL ORDER
-----
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION XTBL(*),YTBL(*)
C
C ***** NON-TABLE FORM TRAP
C
C      IF ( NL.LT.2 ) THEN
C        WRITE(6,*) '***** ERROR ON SUB.YFVEXP : ILLEGAL NL ',NL
C        CALL PEXIT
C        STOP
C      ENDIF
C
C ***** BRANCH BY INC. OR DEC. TYPE
C
C      IF ( XTBL(1).LE.XTBL(NL) ) THEN
C        .... INCREMENT TYPE
C      *** EXTERPOLATION
C        .. X < XTBL(1)
C        IF (X.LE.XTBL( 1)) THEN
C          Y = YTBL( 1) + (YTBL( 2)-YTBL( 1)) * ( X -XTBL( 1))
C            / (XTBL( 2)-XTBL( 1))
C        GOTO 900
C      ENDIF
C
C        .. X > XTBL(NLEN)
C      IF (X.GE.XTBL(NL)) THEN
C        Y = YTBL(NL) + (YTBL(NL)-YTBL(NL-1)) * ( X -XTBL(NL) )
C            / (XTBL(NL)-XTBL(NL-1))
C        GOTO 900
C      ENDIF
C
C      *** INTERPOLATION
C        .. TT(1)<T<TT(NP)
C      DO 100 I=2,NL
C        IF (X.LE.XTBL(I)) THEN
C          Y = YTBL(I-1) + (YTBL(I)-YTBL(I-1)) * ( X -XTBL(I-1))

```

```

C            / (XTBL(I)-XTBL(I-1))
C          GOTO 900
C        ENDIF
100 CONTINUE
C
C      WRITE(6,*) '***** ERROR ON SUB.YFVEXP : ILLEGAL X-VALUE ',X
C      CALL PEXIT
C      STOP
C
C      ELSE
C        .... DECREMENT TYPE
C      *** EXTERPOLATION
C        .. X > XTBL(1)
C        IF (X.GE.XTBL( 1)) THEN
C          Y = YTBL( 1) + (YTBL( 2)-YTBL( 1)) * ( X -XTBL( 1))
C            / (XTBL( 2)-XTBL( 1))
C        GOTO 900
C      ENDIF
C
C        .. X < XTBL(NLEN)
C      IF (X.LE.XTBL(NL)) THEN
C        Y = YTBL(NL) + (YTBL(NL)-YTBL(NL-1)) * ( X -XTBL(NL) )
C            / (XTBL(NL)-XTBL(NL-1))
C        GOTO 900
C      ENDIF
C
C      *** INTERPOLATION
C        .. TT(1)>T>TT(NP)
C      DO 200 I=2,NL
C        IF (X.GE.XTBL(I)) THEN
C          Y = YTBL(I-1) + (YTBL(I)-YTBL(I-1)) * ( X -XTBL(I-1))
C            / (XTBL(I)-XTBL(I-1))
C        GOTO 900
C      ENDIF
200 CONTINUE
C
C      WRITE(6,*) '***** ERROR ON SUB.YFVEXP : ILLEGAL X-VALUE ',X
C      CALL PEXIT
C      STOP
C
C      ENDIF
C
C      900 CONTINUE
C      RETURN
C      END
SUBROUTINE YFXLAM( KFIELD, NCOMP, NNOR, E1, PO11,
E2, PO12, ALP, DT, OKAPDT,

```

```

          DALDT , DFDS , EPSE , DEPS , XLAM )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   CALCULATE NUMERATOR OF 'LAMDA' (TO JUDGE UNLOADING)
C   PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   E1      D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   PO11    D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   PO12    D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   ALP     D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C   DT      D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   DKAPDT  D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C   DALDT(1) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL)/DT
C   DFDS (1) D. ARG (REAL*8) : FLOW VECTOR (DF/DS)
C   EPSE (1) D. ARG (REAL*8) : ELASTIC STRAIN
C   DEPS (1) D. ARG (REAL*8) : STRAIN INCRMENT
C-----
C * OUTPUT VARIABLES
C   XLAM    D. ARG (REAL*8) : NUMERATOR OF 'LAMDA'
C-----
C * CALLED SUBROUTINES
C   YFDE    : GET ELASTIC STRESS-STRAIN MATRIX 'DE'
C   YMAB    : MATRIX MULTIPLY ROUTINE ( A * B )
C   YMBB    : VECOR MULTIPLY ROUTINE ( (B)T * B )
C-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION DALDT(*), DFDS(*), EPSE(*), DEPS(*)
C   DIMENSION DEDT(6,6), DE2(6,6), ALPV(6), DF(6),
C           DEDTEE(6), DAL(6)
C
C   CALL YFDE (KFIELD , 6 , E2 , PO12 , DE2 )
C
C ***** TEMPERATURE DIFFERENE OF 'DE' MATRIX 'DE/DT'
C
C   IF ( ( DT.EQ.0.0 ) .OR.
C       ( (E1.EQ.E2).AND.(PO11.EQ.PO12) ) ) THEN
C
C       DO 110 J=1,NCOMP
C         DO 100 I=1,NCOMP
C           DEDT(I,J) = 0.0

```

```

100 CONTINUE
110 CONTINUE
C
C   ELSE
C
C       CALL YFDE (KFIELD , 6 , E1 , PO11 , DEDT )
C
C       DO 210 I=1,NCOMP
C         DO 200 J=1,NCOMP
C           DEDT(I,J) = ( DE2(I,J)-DEDT(I,J) ) / DT
C200 CONTINUE
C210 CONTINUE
C
C   ENDIF
C
C ***** GET TERM-1
C           .. 'DF' = 'DE' * 'DFDS'
C
C   CALL YMAB(NCOMP,NCOMP,6,DE2,DFDS,DF)
C
C           .. TERM1 = 'DFDS' * 'DE' * 'DEPS'
C
C   CALL YMBB(NCOMP,DF,DEPS,TERM1)
C
C ***** GET TERM-2
C           .. 'DEDTEE' = 'DEDT' * 'EPSE'
C
C   CALL YMAB(NCOMP,NCOMP,6,DEDT,EPSE,DEDTEE)
C
C           .. THERMAL EXPANSION 'ALPV'
C
C
C   DO 300 I=1,NCOMP
C     IF ( I.LE.NNOR ) THEN
C       ALPV(I) = ALP
C     ELSE
C       ALPV(I) = 0.0
C     ENDIF
C300 CONTINUE
C           .. 'DAL' = 'DE' * 'ALP'
C
C   CALL YMAB(NCOMP,NCOMP,6,DE2,ALPV,DAL)
C
C           .. GET TERM-2
C
C   TERM2 = 0.0
C

```



```

C      E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C      PO12     D. ARG (REAL*8) : POISSON' S RATIO AFTER INCREMENT
C      PHA12    D. ARG (REAL*8) : TEMP. DEP. FUG PHA1(T) AFTER INC.
C      ALP      D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C      AM       D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C      ZETA(1)  D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(1)
C      R (1)    D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (1)
C      DT       D. ARG (REAL*8) : TEMPERATURE INCREMENT
C      DKAPDT   D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C      DALSDT(1) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C      BP       D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C      SIG (1)  D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C      ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C      EPSE (1) D. ARG (REAL*8) : ELASTIC STRAIN
C      DEPS (1) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C      DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C      DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C      DEPSP(1) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C      DEPB     D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C      X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C      IMPLICIT REAL*8 (A-H, O-Z)
C      DIMENSION ZETA (*), R (*),
C      SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C      DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP (3)
C      PARAMETER (NBPA=10)
C      DIMENSION SIGX (6), ALIX (6, NBPA), EPSEX (6),
C      DSIGX (6), DALIX (6, NBPA), DEPSX (6), DEPSPX (6),
C      DFDSX (6), ALX (6), ALSX (6)
C
C      IF (NB. GT. NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C      BP(1) = 0.0
C      BP(2) = BPX
C      BP(3) = 1.0
C
C ***** INITIALIZE OUTPUT VALUES
C
C      DO 110 I=1, NCOMP
C
C      DSIG (I) = 0.0
C      DEPSP(I) = 0.0
C      DO 100 IB=1, NB
C          DALI (I, IB) = 0.0
C      100 CONTINUE
C      110 CONTINUE
C
C      DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
C      DO 210 I=1, NCOMP
C          SIGX (I) = SIG (I)
C          EPSEX(I) = EPSE(I)
C      DO 200 IB=1, NB
C          ALIX(I, IB) = ALI (I, IB)
C      200 CONTINUE
C      210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C      DO 1000 IS=2, 3
C          .. SET INCREMENT
C          DBP = BP(IS) - BP(IS-1)
C
C          DTX = DT * DBP
C
C      DO 300 I=1, NCOMP
C          DEPSX(I) = DEPS(I) * DBP
C      300 CONTINUE
C          .. SET INTER VALUES
C          EX1 = E1 + (E2 - E1) * BP(IS-1)
C          EX2 = E1 + (E2 - E1) * BP(IS)
C          PO1X1 = PO11 + (PO12 - PO11) * BP(IS-1)
C          PO1X2 = PO11 + (PO12 - PO11) * BP(IS)
C          PHX1 = PHA11 + (PHA12 - PHA11) * BP(IS-1)
C          .. SUM BACK STRESS TO GET DF/DS
C
C      DO 400 I=1, NCOMP
C          ALX(I) = 0.0
C      400 CONTINUE
C
C      DO 420 IB=1, NB
C          DO 410 I=1, NCOMP
C              ALX(I) = ALX(I) + ALIX(I, IB)
C          410 CONTINUE
C          420 CONTINUE

```

```

C
DO 430 I=1, NCOMP
  ALSX(I) = PHX1 * ALX(I)
430 CONTINUE
C
      .. INTERNAL VARIABLES INC.
C
CALL YFLOW2( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,
             EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
             ALP , AM , ZETA , R , DTX ,
             DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
             DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
      .. CHECK NEXT BREAK POINT
C
IF (IS.EQ.3) THEN
C
  CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
              DALIX , KBP , BP2 )
C
      .. TO NEXT BREAK POINT CAL.
C
IF (KBP.NE.0) THEN
C
  CALL YOWBP2( KFIELD , NCOMP , NNOR , NB , KM ,
              EX1 , POIX1 , PHX1 , E2 , POI2 ,
              PHA12 , ALP , AM , ZETA , R ,
              DTX , DKAPDT , DALSDT , BP2 , SIGX ,
              ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
              DEPSPX , DEPBX )
C
ENDIF
C
ENDIF
C
      .. SUM UP VARIABLES
C
DO 510 I=1, NCOMP
  DSIG(I) = DSIG(I) + DSIGX(I)
  DEPS(I) = DEPS(I) + DEPSPX(I)
DO 500 IB=1, NB
  DALI(I, IB) = DALI(I, IB) + DALIX(I, IB)
500 CONTINUE
510 CONTINUE
C
DEPB = DEPB + DEPBX
C
      .. SET NEXT VALUE
C
IF (IS.EQ.2) THEN

```

```

C
DO 610 I=1, NCOMP
  SIGX(I) = SIGX(I) + DSIGX(I)
DO 600 IB=1, NB
  ALIX(I, IB) = ALIX(I, IB) + DALIX(I, IB)
600 CONTINUE
610 CONTINUE
C
  CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
              EPSEX )
C
ENDIF
C
1000 CONTINUE
C
RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
              NBPA, NB
  CALL PEXIT
  STOP
  END
SUBROUTINE YOWBP2( KFIELD , NCOMP , NNOR , NB , KM ,
                  E1 , POI1 , PHA11 , E2 , POI2 ,
                  PHA12 , ALP , AM , ZETA , R ,
                  DT , DKAPDT , DALSDT , BPX , SIG ,
                  ALI , EPSE , DEPS , DSIG , DALI ,
                  DEPSX , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C   WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   NB      D. ARG (INT)   : TOTAL BACK STRESS COUNT
C   KM      D. ARG (INT)   : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C                                     (MODEL-2 USES M-VALUE)
C   E1      D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C   POI1    D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C   PHA11   D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C   E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C   POI2    D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C   PHA12   D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.

```

```

C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER ZETA(1)
C R (1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER R (1)
C DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C DALSDT(1) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C BP D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C SIG (1) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C EPSE (1) D. ARG (REAL*8) : ELASTIC STRAIN
C DEPS (1) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C DSIG (1) D. ARG (REAL*8) : STRESS INCREMENT
C DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C DEPSP(1) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C DEPB D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C
C IMPLICIT REAL*8 (A-H, O-Z)
C DIMENSION ZETA (*), R (*),
C SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP (3)
C PARAMETER (NBPA=10)
C DIMENSION SIGX(6), ALIX(6, NBPA), EPSEX(6),
C DSIGX(6), DALIX(6, NBPA), DEPSX(6), DEPSPX(6),
C DFDSX(6), ALX(6), ALSX(6)
C
C IF (NB.GT.NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C BP(1) = 0D0
C BP(2) = BPX
C BP(3) = 1D0
C
C ***** INITIALIZE OUTPUT VALUES
C
C DO 110 I=1, NCOMP
C DSIG (I) = 0.0
C DEPSP(I) = 0.0
C DO 100 IB=1, NB

```

```

C DALI(1, IB) = 0.0
100 CONTINUE
110 CONTINUE
C
C DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
C DO 210 I=1, NCOMP
C SIGX (I) = SIG (I)
C EPSEX(I) = EPSE(I)
C DO 200 IB=1, NB
C ALIX(1, IB) = ALI (1, IB)
200 CONTINUE
210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C DO 1000 IS=2, 3
C .. SET INCREMENT
C
C DBP = BP(IS) - BP(IS-1)
C
C DTX = DT * DBP
C
C DO 300 I=1, NCOMP
C DEPSX(I) = DEPS(I) * DBP
300 CONTINUE
C
C .. SET INTER VALUES
C
C EX1 = E1 + (E2 - E1) * BP(IS-1)
C EX2 = E1 + (E2 - E1) * BP(IS)
C PO1X1 = PO11 + (PO12 - PO11) * BP(IS-1)
C PO1X2 = PO11 + (PO12 - PO11) * BP(IS)
C PHX1 = PHA11 + (PHA12 - PHA11) * BP(IS-1)
C .. SUM BACK STRESS TO GET DF/DS
C
C DO 400 I=1, NCOMP
C ALX(I) = 0D0
400 CONTINUE
C
C DO 420 IB=1, NB
C DO 410 I=1, NCOMP
C ALX(I) = ALX(I) + ALIX(I, IB)
410 CONTINUE
420 CONTINUE
C
C DO 430 I=1, NCOMP
C ALSX(I) = PHX1 * ALX(I)

```

```

430 CONTINUE
C
C      .. INTERNAL VARIABLES INC.
C
CALL YFLOW2( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,
            EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
            ALP , AM , ZETA , R , DTX ,
            DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
            DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
C      .. CHECK NEXT BREAK POINT
C
IF (IS.EQ.3) THEN
C
CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
            DALIX , KBP , BP2 )
C
C      .. TO NEXT BREAK POINT CAL.
C
IF (KBP.NE.0) THEN
C
CALL YOWBP3( KFIELD , NCOMP , NNOR , NB , KM ,
            EX1 , POIX1 , PHX1 , E2 , POI2 ,
            PHA12 , ALP , AM , ZETA , R ,
            DTX , DKAPDT , DALSDT , BP2 , SIGX ,
            ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
            DEPSPX , DEPBX )
C
C
C      ENDIF
C
C      ENDIF
C
C      .. SUM UP VARIABLES
C
DO 510 I=1,NCOMP
    DSIG (I) = DSIG (I) + DSIGX (I)
    DEPSX(I) = DEPSX(I) + DEPSPX(I)
DO 500 IB=1,NB
    DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500 CONTINUE
510 CONTINUE
C
DEPB = DEPB + DEPBX
C
C      .. SET NEXT VALUE
C
IF (IS.EQ.2) THEN
C
DO 610 I=1,NCOMP
    SIGX(I) = SIGX(I) + DSIGX (I)

```

```

DO 600 IB=1,NB
    ALIX(I,IB) = ALIX(I,IB) + DALIX (I,IB)
600 CONTINUE
610 CONTINUE
C
CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
            EPSEX )
C
C      ENDIF
C
1000 CONTINUE
C
RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
            NBPA,NB
CALL PEXIT
STOP
END
SUBROUTINE YOWBP3( KFIELD , NCOMP , NNOR , NB , KM ,
                E1 , POI1 , PHA11 , E2 , POI2 ,
                PHA12 , ALP , AM , ZETA , R ,
                DT , DKAPDT , DALSDT , BPX , SIG ,
                ALI , EPSE , DEPS , DSIG , DALI ,
                DEPSX , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE
C GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C POI1 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C PHA11 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C POI2 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C PHA12 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(I)

```

```

C   R (I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (I)
C   DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C   DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C   DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C   BP D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C   SIG (I) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C   ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C   EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C   DEPS (I) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C   DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C   DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C   DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C   DEPB D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C   X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C
C   IMPLICIT REAL*8 (A-H, O-Z)
C   DIMENSION ZETA (*), R (*),
C           SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C           DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP (3)
C   PARAMETER (NBPA=10)
C   DIMENSION SIGX (6), ALIX (6, NBPA), EPSEX (6),
C           DSIGX (6), DALIX (6, NBPA), DEPSX (6), DEPSPX (6),
C           DFDSX (6), ALX (6), ALSX (6)
C
C   IF (NB.GT.NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C   BP(1) = 0D0
C   BP(2) = BPX
C   BP(3) = 1D0
C
C ***** INITIALIZE OUTPUT VALUES
C
C   DO 110 I=1, NCOMP
C     DSIG (I) = 0.0
C     DEPSP(I) = 0.0
C     DO 100 IB=1, NB
C       DALI(I, IB) = 0.0
C   100 CONTINUE
C   110 CONTINUE

```

```

C
C   DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
C   DO 210 I=1, NCOMP
C     SIGX (I) = SIG (I)
C     EPSEX(I) = EPSE(I)
C     DO 200 IB=1, NB
C       ALIX(I, IB) = ALI (I, IB)
C   200 CONTINUE
C   210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C   DO 1000 IS=2, 3
C     .. SET INCREMENT
C     DBP = BP(IS) - BP(IS-1)
C
C     DTX = DT * DBP
C
C     DO 300 I=1, NCOMP
C       DEPSX(I) = DEPS(I) * DBP
C   300 CONTINUE
C     .. SET INTER VALUES
C     EX1 = E1 + (E2 -E1 ) *BP(IS-1)
C     EX2 = E1 + (E2 -E1 ) *BP(IS)
C     POIX1 = POI1 + (POI2 -POI1 ) *BP(IS-1)
C     POIX2 = POI1 + (POI2 -POI1 ) *BP(IS)
C     PHX1 = PHA11 + (PHA12-PHA11) *BP(IS-1)
C     .. SUM BACK STRESS TO GET DF/DS
C     DO 400 I=1, NCOMP
C       ALX(I) = 0D0
C   400 CONTINUE
C
C     DO 420 IB=1, NB
C       DO 410 I=1, NCOMP
C         ALX(I) = ALX(I) + ALIX(I, IB)
C   410 CONTINUE
C   420 CONTINUE
C
C     DO 430 I=1, NCOMP
C       ALSX(I) = PHX1 * ALX(I)
C   430 CONTINUE
C     .. INTERNAL VARIABLES INC.
C
C

```



```

CALL YFLOW2( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,
            EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
            ALP , AM , ZETA , R , DTX ,
            DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
            DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
C          .. CHECK NEXT BREAK POINT
IF (IS.EQ.3) THEN
C
CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
            DALIX , KBP , BP2 )
C          .. TO NEXT BREAK POINT CAL.
IF (KBP.NE.0) THEN
C
CALL YOWBP4( KFIELD , NCOMP , NNOR , NB , KM ,
            EX1 , POIX1 , PHX1 , E2 , POI2 ,
            PHA12 , ALP , AM , ZETA , R ,
            DTX , DKAPDT , DALSDT , BP2 , SIGX ,
            ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
            DEPSPX , DEPBX , )
C
ENDIF
C
ENDIF
C
C          .. SUM UP VARIABLES
DO 510 I=1,NCOMP
    DSIG (I) = DSIG (I) + DSIGX (I)
    DEPSP(I) = DEPSP(I) + DEPSPX(I)
    DO 500 IB=1,NB
        DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500 CONTINUE
510 CONTINUE
C
DEPB = DEPB + DEPBX
C          .. SET NEXT VALUE
IF (IS.EQ.2) THEN
C
DO 610 I=1,NCOMP
    SIGX(I) = SIGX(I) + DSIGX (I)
    DO 600 IB=1,NB
        ALIX(I,IB) = ALIX(I,IB) + DALIX (I,IB)
600 CONTINUE

```

```

610 CONTINUE
C
CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
            EPSEX )
C
ENDIF
C
1000 CONTINUE
C
RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
            NBPA,NB
CALL PEXIT
STOP
END
SUBROUTINE YOWBP4( KFIELD , NCOMP , NNOR , NB , KM ,
                E1 , POI1 , PHA11 , E2 , POI2 ,
                PHA12 , ALP , AM , ZETA , R ,
                DT , DKAPDT , DALSDT , BPX , SIG ,
                ALI , EPSE , DEPS , DSIG , DALI ,
                DEPSP , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE
C GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C POI1 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C PHA11 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C POI2 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C PHA12 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(I)
C R (I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (I)
C DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT

```

```

C   DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C   BP       D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C   SIG  (I) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C   ALI  (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C   EPSE  (I) D. ARG (REAL*8) : ELASTIC STRAIN
C   DEPS  (I) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C   DSIG  (I) D. ARG (REAL*8) : STRESS INCREMENT
C   DALI  (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C   DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C   DEPB  (I) D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C   X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C
C   IMPLICIT REAL*8 (A-H, O-Z)
C   DIMENSION ZETA (*), R  (*),
C             SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C             DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP(3)
C   PARAMETER (NBPA=10)
C   DIMENSION SIGX(6), ALIX(6, NBPA), EPSEX(6),
C             DSIGX(6), DALIX(6, NBPA), DEPSX (6), DEPSX(6),
C             DFDSX(6), ALX (6) , ALSX (6)
C
C   IF (NB. GT. NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C   BP(1) = ODO
C   BP(2) = BPX
C   BP(3) = 1DO
C
C ***** INITIALIZE OUTPUT VALUES
C
C   DO 110 I=1, NCOMP
C     DSIG (I) = 0.0
C     DEPSP(I) = 0.0
C     DO 100 IB=1, NB
C       DALI (I, IB) = 0.0
C   100 CONTINUE
C   110 CONTINUE
C
C   DEPB = 0.0
C

```

```

C ***** SET INITIAL VALUES
C
C   DO 210 I=1, NCOMP
C     SIGX (I) = SIG (I)
C     EPSEX(I) = EPSE(I)
C     DO 200 IB=1, NB
C       ALIX(I, IB) = ALI (I, IB)
C   200 CONTINUE
C   210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C   DO 1000 IS=2, 3
C     .. SET INCREMENT
C     DBP = BP(IS) - BP(IS-1)
C
C     DTX = DT * DBP
C
C     DO 300 I=1, NCOMP
C       DEPSX(I) = DEPS(I) * DBP
C   300 CONTINUE
C     .. SET INTER VALUES
C     EX1 = E1 + (E2 -E1 )*BP(IS-1)
C     EX2 = E1 + (E2 -E1 )*BP(IS)
C     POIX1 = POI1 + (POI2 -POI1 )*BP(IS-1)
C     POIX2 = POI1 + (POI2 -POI1 )*BP(IS)
C     PHX1 = PHA11 + (PHA12-PHA11)*BP(IS-1)
C     .. SUM BACK STRESS TO GET DF/DS
C
C     DO 400 I=1, NCOMP
C       ALX(I) = ODO
C   400 CONTINUE
C
C     DO 420 IB=1, NB
C       DO 410 I=1, NCOMP
C         ALX(I) = ALX(I) + ALIX(I, IB)
C   410 CONTINUE
C   420 CONTINUE
C
C     DO 430 I=1, NCOMP
C       ALSX(I) = PHX1 * ALX(I)
C   430 CONTINUE
C     .. INTERNAL VARIABLES INC.
C
C     CALL YFLOW2( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
C     CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,

```

```

      EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
      ALP , AM , ZETA , R , DTX ,
      DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
      DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
C      .. CHECK NEXT BREAK POINT
IF (IS.EQ.3) THEN
C
C      CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
      DALIX , KBP , BP2 )
C      .. TO NEXT BREAK POINT CAL.
IF (KBP.NE.0) THEN
C
C      CALL YOWBP5( KFIELD , NCOMP , NNOR , NB , KM ,
      EX1 , POIX1 , PHX1 , E2 , POI2 ,
      PHA12 , ALP , AM , ZETA , R ,
      DTX , DKAPDT , DALSDT , BP2 , SIGX ,
      ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
      DEPSPX , DEPBX , )
C
C      ENDIF
C
C      ENDIF
C
C      .. SUM UP VARIABLES
DO 510 I=1,NCOMP
      DSIG (I) = DSIG (I) + DSIGX (I)
      DEPSP(I) = DEPSP(I) + DEPSPX(I)
DO 500 IB=1,NB
      DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500 CONTINUE
510 CONTINUE
C
C      DEPB = DEPB + DEPBX
C      .. SET NEXT VALUE
IF (IS.EQ.2) THEN
C
C      DO 610 I=1,NCOMP
      SIGX(I) = SIGX(I) + DSIGX (I)
      DO 600 IB=1,NB
      ALIX(I,IB) = ALIX(I,IB) + DALIX (I,IB)
600 CONTINUE
610 CONTINUE
C
C      CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,

```

```

      EPSEX )
C
C      ENDIF
C
C      1000 CONTINUE
C
C      RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B.STRESS COUNT EXCEED ',
      NBPA,NB
      CALL PEXIT
      STOP
      END
      SUBROUTINE YOWBP5( KFIELD , NCOMP , NNOR , NB , KM ,
      E1 , POI1 , PHA11 , E2 , POI2 ,
      PHA12 , ALP , AM , ZETA , R ,
      DT , DKAPDT , DALSDT , BPX , SIG ,
      ALI , EPSE , DEPS , DSIG , DALI ,
      DEPSP , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE
C GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C POI1 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C PHA11 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C POI2 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C PHA12 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER ZETA(1)
C R (1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER R (1)
C DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C DALSDT(1) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C BP D. ARG (REAL*8) : BREAK POINT VALUES (OKBP<1)
C SIG (1) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP

```

```

C     ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C     EPSE (1) D. ARG (REAL*8) : ELASTIC STRAIN
C     DEPS (1) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C     DSIG (1) D. ARG (REAL*8) : STRESS INCREMENT
C     DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C     DEPSP(1) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C     DEPBB   D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C     X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C
C     IMPLICIT REAL*8 (A-H, O-Z)
C     DIMENSION ZETA (*), R (*),
C             SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C             DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP (3)
C     PARAMETER (NBPA=10)
C     DIMENSION SIGX (6), ALIX (6, NBPA), EPSEX (6),
C             DSIGX (6), DALIX (6, NBPA), DEPSX (6), DEPSX (6),
C             DFDSX (6), ALX (6), ALSX (6)
C
C     IF (NB. GT. NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C     BP (1) = 0D0
C     BP (2) = BPX
C     BP (3) = 1D0
C
C ***** INITIALIZE OUTPUT VALUES
C
C     DO 110 I=1, NCOMP
C         DSIG (I) = 0.0
C         DEPSP (I) = 0.0
C         DO 100 IB=1, NB
C             DALI (I, IB) = 0.0
C     100 CONTINUE
C     110 CONTINUE
C
C     DEPBB = 0.0
C
C ***** SET INITIAL VALUES
C
C     DO 210 I=1, NCOMP
C         SIGX (I) = SIG (I)
C         EPSEX (I) = EPSE (I)
C         DO 200 IB=1, NB
C             ALIX (I, IB) = ALI (I, IB)
C     200 CONTINUE
C     210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C     DO 1000 IS=2, 3
C         .. SET INCREMENT
C         DBP = BP (IS) - BP (IS-1)
C
C         DTX = DT * DBP
C
C         DO 300 I=1, NCOMP
C             DEPSX (I) = DEPS (I) * DBP
C     300 CONTINUE
C         .. SET INTER VALUES
C         EX1 = E1 + (E2 - E1) * BP (IS-1)
C         EX2 = E1 + (E2 - E1) * BP (IS)
C         POIX1 = POI1 + (POI2 - POI1) * BP (IS-1)
C         POIX2 = POI1 + (POI2 - POI1) * BP (IS)
C         PHX1 = PHA11 + (PHA12 - PHA11) * BP (IS-1)
C         .. SUM BACK STRESS TO GET DF/DS
C         DO 400 I=1, NCOMP
C             ALX (I) = 0D0
C     400 CONTINUE
C         DO 420 IB=1, NB
C             DO 410 I=1, NCOMP
C                 ALX (I) = ALX (I) + ALIX (I, IB)
C     410 CONTINUE
C     420 CONTINUE
C         DO 430 I=1, NCOMP
C             ALSX (I) = PHX1 * ALX (I)
C     430 CONTINUE
C         .. INTERNAL VARIABLES INC.
C
C     CALL YFLOW2 ( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
C     CALL YOWEPV ( KFIELD , NCOMP , NNOR , NB , KM ,
C             EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
C             ALP , AM , ZETA , R , DTX ,
C             DKAPDT , DALSDT , DFDSX , ALIX , EPSEX )

```

```

DEPSX , DSIGX , DALIX , DEPSPX, DEPBX )
C
C          .. CHECK NEXT BREAK POINT
IF (IS.EQ.3) THEN
C
CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
             DALIX , KBP , BP2 )
C          .. TO NEXT BREAK POINT CAL.
IF (KBP.NE.0) THEN
C
CALL YOWBP6( KFIELD , NCOMP , NNOR , NB , KM ,
             EX1 , POIX1 , PHX1 , E2 , POI2 ,
             PHA12 , ALP , AM , ZETA , R ,
             DTX , DKAPDT , DALSDT , BP2 , SIGX ,
             ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
             DEPSPX , DEPBX )
C
ENDIF
C
ENDIF
C
C          .. SUM UP VARIABLES
DO 510 I=1,NCOMP
    DSIG (I) = DSIG (I) + DSIGX (I)
    DEPS (I) = DEPS (I) + DEPSPX (I)
DO 500 IB=1,NB
    DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500 CONTINUE
510 CONTINUE
C
DEPB = DEPB + DEPBX
C          .. SET NEXT VALUE
IF (IS.EQ.2) THEN
C
DO 610 I=1,NCOMP
    SIGX (I) = SIGX (I) + DSIGX (I)
DO 600 IB=1,NB
    ALIX (I,IB) = ALIX (I,IB) + DALIX (I,IB)
600 CONTINUE
610 CONTINUE
C
CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
            EPSEX )
C
ENDIF

```

```

C
1000 CONTINUE
C
RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
             NBPA,NB
CALL PEXIT
STOP
END
SUBROUTINE YOWBP6( KFIELD , NCOMP , NNOR , NB , KM ,
                 E1 , POI1 , PHA11 , E2 , POI2 ,
                 PHA12 , ALP , AM , ZETA , R ,
                 DT , DKAPDT , DALSDT , BPX , SIG ,
                 ALI , EPSE , DEPS , DSIG , DALI ,
                 DEPS , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE
C GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C POI1 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C PHA11 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C POI2 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C PHA12 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(I)
C R (I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (I)
C DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C BP D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C SIG (I) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C ALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C DEPS (I) D. ARG (REAL*8) : STRAI INCREMENT

```

```

-----
C * OUTPUT VARIABLES
C   DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C   DALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C   DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C   DEPB   D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
-----
C * LOCAL VARIABLES
C   X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
-----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION ZETA (*), R (*),
             SIG (*), ALI (6, *), EPSE (*), DALSDT(*),
             DSIG (*), DALI (6, *), DEPS (*), DEPSP (*), BP(3)
  PARAMETER (NBPA=10)
  DIMENSION SIGX(6), ALIX(6,NBPA), EPSEX(6),
             DSIGX(6), DALIX(6,NBPA), DEPSX(6), DEPSPX(6),
             DFDSX(6), ALX(6), ALSX(6)
C
  IF (NB.GT.NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
  BP(1) = ODO
  BP(2) = BPX
  BP(3) = 1DO
C
C ***** INITIALIZE OUTPUT VALUES
C
  DO 110 I=1,NCOMP
    DSIG(I) = 0.0
    DEPSP(I) = 0.0
    DO 100 IB=1,NB
      DALI(I,IB) = 0.0
  100 CONTINUE
  110 CONTINUE
C
  DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
  DO 210 I=1,NCOMP
    SIGX(I) = SIG(I)
    EPSEX(I) = EPSE(I)
  DO 200 IB=1,NB

```

```

    ALIX(I,IB) = ALI(I,IB)
  200 CONTINUE
  210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
  DO 1000 IS=2,3
    .. SET INCREMENT
    DBP = BP(IS) - BP(IS-1)
C
    DTX = DT * DBP
C
  DO 300 I=1,NCOMP
    DEPSX(I) = DEPS(I) * DBP
  300 CONTINUE
C
    .. SET INTER VALUES
    EX1 = E1 + (E2 -E1 )*BP(IS-1)
    EX2 = E1 + (E2 -E1 )*BP(IS)
    POIX1 = POI1 + (POI2 -POI1 )*BP(IS-1)
    POIX2 = POI1 + (POI2 -POI1 )*BP(IS)
    PHX1 = PHA11 + (PHA12-PHA11)*BP(IS-1)
C
    .. SUM BACK STRESS TO GET DF/DS
  DO 400 I=1,NCOMP
    ALX(I) = ODO
  400 CONTINUE
C
  DO 420 IB=1,NB
    DO 410 I=1,NCOMP
      ALX(I) = ALX(I) + ALIX(I,IB)
  410 CONTINUE
  420 CONTINUE
C
  DO 430 I=1,NCOMP
    ALSX(I) = PHX1 * ALX(I)
  430 CONTINUE
C
    .. INTERNAL VARIABLES INC.
C
  CALL YFLOW2( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
  CALL YOWEPV( KFIELD , NCOMP , NNOR , NB , KM ,
             EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
             ALP , AM , ZETA , R , DTX ,
             DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
             DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
    .. CHECK NEXT BREAK POINT

```

```

IF (IS.EQ.3) THEN
C
CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
DALIX , KBP , BP2 )
C
.. TO NEXT BREAK POINT CAL.
IF (KBP.NE.0) THEN
C
CALL YOWBP7( KFIELD , NCOMP , NNOR , NB , KM ,
EX1 , POIX1 , PHX1 , E2 , POI2 ,
PHAI2 , ALP , AM , ZETA , R ,
DTX , DKAPDT , DALSDT , BP2 , SIGX ,
ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
DEPSPX , DEPBX , )
C
ENDIF
C
ENDIF
C
.. SUM UP VARIABLES
C
DO 510 I=1,NCOMP
DSIG (I) = DSIG (I) + DSIGX (I)
DEPSP(I) = DEPSP(I) + DEPSPX(I)
DO 500 IB=1,NB
DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500 CONTINUE
510 CONTINUE
C
DEPB = DEPB + DEPBX
C
.. SET NEXT VALUE
IF (IS.EQ.2) THEN
C
DO 610 I=1,NCOMP
SIGX(I) = SIGX(I) + DSIGX (I)
DO 600 IB=1,NB
ALIX(I,IB) = ALIX(I,IB) + DALIX (I,IB)
600 CONTINUE
610 CONTINUE
C
CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
EPSEX )
C
ENDIF
C
1000 CONTINUE
C

```

```

RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
NBPA,NB
CALL PEXIT
STOP
END
SUBROUTINE YOWBP7( KFIELD , NCOMP , NNOR , NB , KM ,
E1 , POI1 , PHAI1 , E2 , POI2 ,
PHAI2 , ALP , AM , ZETA , R ,
DT , DKAPDT , DALSDT , BPX , SIG ,
ALI , EPSE , DEPS , DSIG , DALI ,
DEPSP , DEPB )
C
-----
C * FUNCTION OF THIS SUBROUTINE
C GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C
-----
C * INPUT VARIABLES
C KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C POI1 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C PHAI1 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C POI2 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C PHAI2 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA(1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER ZETA(1)
C R (I) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER R (I)
C DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C BP D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C SIG (I) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C DEPS (I) D. ARG (REAL*8) : STRAI INCREMENT
C
-----
C * OUTPUT VARIABLES
C DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT

```

```

C   DALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C   DEPS(1) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C   DEPB D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C * LOCAL VARIABLES
C   X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
C-----
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION ZETA (*), R (*),
C           SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
C           DSIG (*), DALI (6, *), DEPS (*), DEPS(1), BP(3)
C   PARAMETER (NBPA=10)
C   DIMENSION SIGX (6), ALIX (6, NBPA), EPSEX (6),
C           DSIGX (6), DALIX (6, NBPA), DEPSX (6), DEPSPX (6),
C           DFDSX (6), ALX (6), ALSX (6)
C
C   IF (NB.GT.NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
C   BP(1) = 0D0
C   BP(2) = 8PX
C   BP(3) = 1D0
C
C ***** INITIALIZE OUTPUT VALUES
C
C   DO 110 I=1, NCOMP
C     DSIG (I) = 0.0
C     DEPS(1) = 0.0
C     DO 100 IB=1, NB
C       DALI (I, IB) = 0.0
C   100 CONTINUE
C   110 CONTINUE
C
C   DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
C   DO 210 I=1, NCOMP
C     SIGX (I) = SIG (I)
C     EPSEX(I) = EPSE(I)
C     DO 200 IB=1, NB
C       ALIX(I, IB) = ALI (I, IB)
C   200 CONTINUE
C   210 CONTINUE

```

```

C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C
C   DO 1000 IS=2, 3
C     .. SET INCREMENT
C     DBP = BP(IS) - BP(IS-1)
C
C     DTX = DT * DBP
C
C     DO 300 I=1, NCOMP
C       DEPSX(I) = DEPS(I) * DBP
C   300 CONTINUE
C     .. SET INTER VALUES
C     EX1 = E1 + (E2 - E1) * BP(IS-1)
C     EX2 = E1 + (E2 - E1) * BP(IS)
C     POIX1 = POI1 + (POI2 - POI1) * BP(IS-1)
C     POIX2 = POI1 + (POI2 - POI1) * BP(IS)
C     PHX1 = PHA11 + (PHA12 - PHA11) * BP(IS-1)
C     .. SUM BACK STRESS TO GET DF/DS
C     DO 400 I=1, NCOMP
C       ALX(I) = 0D0
C   400 CONTINUE
C     DO 420 IB=1, NB
C       DO 410 I=1, NCOMP
C         ALX(I) = ALX(I) + ALIX(I, IB)
C   410 CONTINUE
C   420 CONTINUE
C     DO 430 I=1, NCOMP
C       ALSX(I) = PHX1 * ALX(I)
C   430 CONTINUE
C     .. INTERNAL VARIABLES INC.
C
C     CALL YFLOW2( NCOMP, NNOR, SIGX, ALSX, DFDSX )
C
C     CALL YOWEPV( KFIELD, NCOMP, NNOR, NB, KM,
C     .           EX1, POIX1, EX2, POIX2, PHA12,
C     .           ALP, AM, ZETA, R, DTX,
C     .           DKAPDT, DALSDT, DFDSX, ALIX, EPSEX,
C     .           DEPSX, DSIGX, DALIX, DEPSPX, DEPBX )
C
C     .. CHECK NEXT BREAK POINT
C     IF (IS.EQ.3) THEN
C
C       CALL YOWBPX ( NCOMP, NNOR, NB, R, ALIX,

```



```

C          DALIX , KBP , BP2          )
C          .. TO NEXT BREAK POINT CAL.
C      IF (KBP.NE.0) THEN
C
C          CALL YOWBP8( KFIELD , NCOMP , NNOR , NB , KM ,
C                      EX1 , POIX1 , PHX1 , E2 , PO12 ,
C                      PHA12 , ALP , AM , ZETA , R ,
C                      DTX , DKAPDT , DALSDT , BP2 , SIGX ,
C                      ALIX , EPSEX , DEPSX , DSIGX , DALIX ,
C                      DEPSPX , DEPBX , )
C
C      ENDIF
C
C      ENDIF
C
C          .. SUM UP VARIABLES
C
C      DO 510 I=1,NCOMP
C          DSIG (I) = DSIG (I) + DSIGX (I)
C          DEPSP(I) = DEPSP(I) + DEPSPX(I)
C      DO 500 IB=1,NB
C          DALI (I,IB) = DALI (I,IB) + DALIX (I,IB)
500      CONTINUE
510      CONTINUE
C
C      DEPB = DEPB + DEPBX
C
C          .. SET NEXT VALUE
C      IF (IS.EQ.2) THEN
C
C          DO 610 I=1,NCOMP
C              SIGX(I) = SIGX(I) + DSIGX (I)
C          DO 600 IB=1,NB
C              ALIX(I,IB) = ALIX(I,IB) + DALIX (I,IB)
600      CONTINUE
610      CONTINUE
C
C          CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
C                    EPSEX , )
C
C      ENDIF
C
C      1000 CONTINUE
C
C      RETURN
C      9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B.STRESS COUNT EXCEED ',
C                    NBPA,NB

```

```

CALL PEXIT
STOP
END
SUBROUTINE YOWBP8( KFIELD , NCOMP , NNOR , NB , KM ,
                  E1 , PO11 , PHA11 , E2 , PO12 ,
                  PHA12 , ALP , AM , ZETA , R ,
                  DT , DKAPDT , DALSDT , BPX , SIG ,
                  ALI , EPSE , DEPS , DSIG , DALI ,
                  DEPSP , DEPB , )
C-----
C * FUNCTION OF THIS SUBROUTINE
C      GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C      WITH ONE BREAK POINT SUBSTEP INTEGRATION MEHOD FOR OHNO-WANG MODEL
C      PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C      KFIELD D. ARG (INT) : FINAS STRESS FIELD CODE
C      NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C      NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C      NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C      KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C                          (MODEL-2 USES M-VALUE)
C      E1 D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C      PO11 D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C      PHA11 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) BEFORE INC.
C      E2 D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C      PO12 D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C      PHA12 D. ARG (REAL*8) : TEMP. DEP. FUC PHAI(T) AFTER INC.
C      ALP D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C      AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C      ZETA(I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(I)
C      R (I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (I)
C      DT D. ARG (REAL*8) : TEMPERATURE INCREMENT
C      DKAPDT D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C      DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C      BP D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
C      SIG (I) D. ARG (REAL*8) : STRESS AT PREVIOUS STEP
C      ALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C      EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C      DEPS (I) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C      DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C      DALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C      DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C      DEPB D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT

```

```

-----
C * LOCAL VARIABLES
C   X-SUFFIXED VARIABLES ARE INTER SUBSTEPPING VALUES
-----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION ZETA (*), R (*),
            SIG (*), ALI (6, *), EPSE (*), DALSDT (*),
            DSIG (*), DALI (6, *), DEPS (*), DEPSX (*), BP (3)
  PARAMETER (NBPA=10)
  DIMENSION SIGX (6), ALIX (6, NBPA), EPSEX (6),
            DSIGX (6), DALIX (6, NBPA), DEPSX (6), DEPSPX (6),
            DFDSX (6), ALX (6), ALSX (6)
C
  IF (NB.GT.NBPA) GOTO 9010
C
C ***** SET BREAK POINT VALUE
C
  BP (1) = ODO
  BP (2) = BPX
  BP (3) = IDO
C
C ***** INITIALIZE OUTPUT VALUES
C
  DO 110 I=1, NCOMP
    DSIG (I) = 0.0
    DEPSX (I) = 0.0
    DO 100 IB=1, NB
      DALI (I, IB) = 0.0
  100 CONTINUE
  110 CONTINUE
C
  DEPB = 0.0
C
C ***** SET INITIAL VALUES
C
  DO 210 I=1, NCOMP
    SIGX (I) = SIG (I)
    EPSEX (I) = EPSE (I)
    DO 200 IB=1, NB
      ALIX (I, IB) = ALI (I, IB)
  200 CONTINUE
  210 CONTINUE
C
C ***** SUBSTEPPING ITERATION BY BREAK POINT
C

```

```

DO 1000 IS=2, 3
C
  DBP = BP (IS) - BP (IS-1) .. SET INCREMENT
C
  DTX = DT * DBP
C
  DO 300 I=1, NCOMP
    DEPSX (I) = DEPS (I) * DBP
  300 CONTINUE
C
  .. SET INTER VALUES
  EX1 = E1 + (E2 - E1) * BP (IS-1)
  EX2 = E1 + (E2 - E1) * BP (IS)
  POIX1 = POI1 + (POI2 - POI1) * BP (IS-1)
  POIX2 = POI1 + (POI2 - POI1) * BP (IS)
  PHX1 = PHA11 + (PHA12 - PHA11) * BP (IS-1)
C
  .. SUM BACK STRESS TO GET DF/DS
  DO 400 I=1, NCOMP
    ALX (I) = ODO
  400 CONTINUE
C
  DO 420 IB=1, NB
    DO 410 I=1, NCOMP
      ALX (I) = ALX (I) + ALIX (I, IB)
    410 CONTINUE
  420 CONTINUE
C
  DO 430 I=1, NCOMP
    ALSX (I) = PHX1 * ALX (I)
  430 CONTINUE
C
  .. INTERNAL VARIABLES INC.
C
  CALL YFLOW2 ( NCOMP , NNOR , SIGX , ALSX , DFDSX )
C
  CALL YOWEPV ( KFIELD , NCOMP , NNOR , NB , KM ,
              EX1 , POIX1 , EX2 , POIX2 , PHA12 ,
              ALP , AM , ZETA , R , DTX ,
              DKAPDT , DALSDT , DFDSX , ALIX , EPSEX ,
              DEPSX , DSIGX , DALIX , DEPSPX , DEPBX )
C
  .. CHECK NEXT BREAK POINT
  IF (IS.EQ.3) THEN
C
  CALL YOWBPX ( NCOMP , NNOR , NB , R , ALIX ,
              DALIX , KBP , BP2 )
C
  .. TO NEXT BREAK POINT CAL.
  IF (KBP.NE.0) THEN

```

```

C
  WRITE(6,*) ' ***** WARNING ON SUB. YOWBPB : '
  WRITE(6,*) '          SUB STEP COUNT EXCEED 8 ',
    'CAUSES LESS OF PRECISION'
C
  ENDIF
C
  ENDIF
C
  .. SUM UP VARIABLES
C
DO 510 I=1,NCOMP
  DSIG (I) = DSIG (I)  + DSIGX (I)
  DEPSP(I) = DEPSP(I)  + DEPSPX(I)
  DO 500 IB=1,NB
    DALI (I, IB) = DALI (I, IB) + DALIX (I, IB)
500  CONTINUE
510  CONTINUE
C
  DEPB = DEPB + DEPBX
C
  .. SET NEXT VALUE
C
  IF (IS.EQ.2) THEN
C
    DO 610 I=1,NCOMP
      SIGX(I) = SIGX(I)  + DSIGX (I)
      DO 600 IB=1,NB
        ALIX(I, IB) = ALIX(I, IB) + DALIX (I, IB)
600  CONTINUE
610  CONTINUE
C
    CALL YFEPSE( NCOMP , NNOR , EX2 , POIX2 , SIGX ,
      EPSEX )
C
  ENDIF
C
  1000 CONTINUE
C
  RETURN
9010 WRITE(6,*) ' ***** ERROR ON SUB. YOWEPB : B. STRESS COUNT EXCEED ',
    NBPA, NB
  CALL PEXIT
  STOP
  END
  SUBROUTINE YOWBPV ( NCOMP , NNOR , AL , DAL , R , KBP , BPV )
C-----
C * FUNCTION OF THIS SUBROUTINE

```

```

C   CHECK BREAK POIT OF F1=0 FUNCTION FOR OHNO-WANG MODEL
C   PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C   AL(1)   D. ARG (REAL*8) : ONE OF BACK STRESS ON TO
C   DAL(1)  D. ARG (REAL*8) : ONE OF BACK STRESS ON TO INCREMENT
C   R       D. ARG (REAL*8) : ONE OF STRESS PARAMETER R
C-----
C * OUTPUT VARIABLES
C   KBP     D. ARG (INT)   : KEY OF BREAK POINT (=0:NONE,=1,EXIST)
C   BPV     D. ARG (REAL*8) : BREAK POINT VALUE (0.0 < BPV < 1.0)
C-----
C
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION AL(*),DAL(*)
  DIMENSION AX(6),DAX(6)
C
C ***** EPS2 MEANS ADVANCE OF F1 SURFACE
C
  DATA EPS2 / 0.00100 /
C
C ***** EXTEND BACK STRESS TO 3-D STRESS FIELD
C
  DO 100 I=1,6
    AX(I) = 0.0
    DAX(I) = 0.0
100 CONTINUE
C
  .. NORMAL COMPONENTS
C
  IF (NNOR.EQ.1) THEN
    AX(1) = AL(1)
    AX(2) = -AL(1)/2D0
    AX(3) = AX(2)
    DAX(1) = DAL(1)
    DAX(2) = -DAL(1)/2D0
    DAX(3) = DAX(2)
  ELSE IF (NNOR.EQ.2) THEN
    AX(1) = AL(1)
    AX(2) = AL(2)
    AX(3) = -AL(1)-AL(2)
    DAX(1) = DAL(1)
    DAX(2) = DAL(2)
    DAX(3) = -DAL(1)-DAL(2)
  ELSE IF (NNOR.EQ.3) THEN
    AX(1) = AL(1)

```

```

      AX(2) = AL(2)
      AX(3) = AL(3)
      DAX(1) = DAL(1)
      DAX(2) = DAL(2)
      DAX(3) = DAL(3)
    ENDIF
  C          .. SHEAR COMPONENTS (DOUBLED)
  C
  C      J = 4
  C      DO 110 I=NNOR+1, NCOMP
  C          AX(J) = AL(I)
  C          DAX(J) = DAL(I)
  C          J = J + 1
  C      110 CONTINUE
  C
  C ***** CAL. INNER PRODUCT
  C
  C      AA = AX(1)**2 + AX(2)**2 + AX(3)**2 +
  C          ( AX(4)**2 + AX(5)**2 + AX(6)**2 ) * 2D0
  C
  C      ADA = AX(1)*DAX(1) + AX(2)*DAX(2) + AX(3)*DAX(3) +
  C          ( AX(4)*DAX(4) + AX(5)*DAX(5) + AX(6)*DAX(6) ) * 2D0
  C
  C      DADA = DAX(1)**2 + DAX(2)**2 + DAX(3)**2 +
  C          ( DAX(4)**2 + DAX(5)**2 + DAX(6)**2 ) * 2D0
  C
  C ***** SET F1 FUNCTION BEFORE AND AFTER INCREMENT
  C
  C      RX2 = (R*(1D0-EPS2))**2
  C      F1 = 1.5D0 * AA - RX2
  C      F2 = 1.5D0 * (AA+2D0*ADA+DADA) - RX2
  C
  C ***** FIND BREAK POINT VALUES
  C
  C      IF ( (F1.LT.0.0).AND.(F2.GT.0.0) ) THEN
  C          .. B.P. EXIST
  C
  C          A = DADA
  C          BD = ADA
  C          C = AA - (2D0/3D0) * R**2
  C          D = BD**2 - A*C
  C
  C          IF (A.EQ.0.0) GOTO 900
  C          IF (D.LT.0.0) GOTO 900
  C          IF (ABS(D).LT.1D-50) GOTO 900
  C
  C          BPV = (-BD+SQRT(D))/A
  C

```

```

      IF (BPV.GE.1.0) GOTO 900
      IF (BPV.LE.0.0) GOTO 900
  C
  C      KBP = 1
  C      ELSE
  C          KBP = 0
  C      ENDIF
  C
  C      RETURN
  C
  C      900 CONTINUE
  C          KBP = 0
  C          RETURN
  C      END
  C      SUBROUTINE YOWBPX ( NCOMP , NNOR , NB , R , ALI ,
  C                      DALI , KBP , BP )
  C-----
  C * FUNCTION OF THIS SUBROUTINE
  C   CHECK AND SET BREAK POINT VALUE OF F1=0 FOR OHNO-WANG MODEL
  C   PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
  C-----
  C * INPUT VARIABLES
  C   NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
  C   NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
  C   NB      D. ARG (INT)   : TOTAL BACK STRESS COUNT
  C   R (I)   D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R(I)
  C   ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
  C   DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
  C-----
  C * OUTPUT VARIABLES
  C   KBP     D. ARG (INT)   : BREAK POINT KEY (0:NONE, 1:EXIST)
  C   BP      D. ARG (REAL*8) : BREAK POINT VALUES (0<BP<1)
  C-----
  C * LOCAL VARIABLES
  C   EPSI    LOCAL (REAL*8) : OMIT BREAK POINT INTERVAL
  C-----
  C   IMPLICIT REAL*8 (A-H, O-Z)
  C   DIMENSION R(*), ALI(6, *), DALI(6, *)
  C
  C   DATA EPSI / 0.01D0 /
  C
  C ***** SEARCH ONE BREAK POINT AND SET VALUE FOR EACH TERM
  C
  C   DO 100 IB = 1, NB
  C

```

```

CALL YOWBPV ( NCOMP , NNOR , ALI(1,IB) , DALI(1,IB) , R(1B) ,
              KBP , BP )
C
C IF (KBP.EQ.1) THEN
C
C IF ( (BP.GT.EPS1) .AND. (BP.LT.1D0-EPS1) ) RETURN
C
C ENDIF
C
100 CONTINUE
C
KBP = 0
RETURN
END
SUBROUTINE YOWDA ( NCOMP , NNOR , NB , KM , AM ,
                  ZETA , R , ALI , DEPSP , DALI )
C-----
C * FUNCTION OF THIS SUBROUTINE
C GET BACK STRESS INCREMNT EXCLUDE TEMPERERATURE CHANGE
C PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C NCOMP D. ARG (INT) : STRESS OR STRAIN COMPONENTS COUNT
C NNOR D. ARG (INT) : NORMAL STRESS OR STRAIN COMP. COUNT
C NB D. ARG (INT) : TOTAL BACK STRESS COUNT
C KM D. ARG (INT) : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C (MODEL-2 USES M-VALUE)
C AM D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C ZETA (1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER ZETA(1)
C R (1) D. ARG (REAL*8) : 1-TH B. STRESS PARAMETER R (1)
C ALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C DEPSP(1) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C-----
C * OUTPUT VARIABLES
C DALI(6,J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C-----
C
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION ZETA(*),R (*),ALI (6,*),DEPSP(*),DALI (6,* )
C DIMENSION DEP3(6),AK(6)
C
C ***** EXTEND PLASTIC STRAIN INCREMENT TO 3-D SPACE (TENSOR TYPE)
C
C DO 100 I3D=1,6
C DEP3( I3D) = 0D0
100 CONTINUE

```

```

C .. NORMAL COMPONENTS
DO 110 IFLD=1,NNOR
DEP3(IFLD) = DEPSP(IFLD)
110 CONTINUE
C
C IF (NNOR.EQ.1) THEN
C DEP3( 2) = -DEP3(1)/2D0
C DEP3( 3) = -DEP3(2)
C ELSE IF (NNOR.EQ.2) THEN
C DEP3( 3) = -(DEP3(1)+DEP3(2))
C ENDIF
C .. SHEAR COMPONENTS
13D = 4
DO 120 IFLD=NNOR+1,NCOMP
DEP3( 13D) = DEPSP(IFLD) / 2D0
13D = 13D + 1
120 CONTINUE
C
C ***** GET EACH BACK STRESS INCREMENT
C
DO 1000 IB=1,NB
C .. GET FI AND H(FI)
C
C CALL YOWFI ( NCOMP , NNOR , ALI(1,IB) , R(1B) , ALIB , FI )
C
C IF (FI.GE.0.0) THEN
C HF1 = 1D0
C ELSE
C HF1 = 0D0
C ENDIF
C .. GET K
C
DO 200 I=1,6
AK(I) = 0D0
200 CONTINUE
C
C IF (ALIB.NE.0.0) THEN
C .. NORMAL COMPONENTS
DO 210 IFLD=1,NNOR
AK(IFLD) = ALI(IFLD,IB) / ALIB
210 CONTINUE
C
C IF (NNOR.EQ.1) THEN
C AK( 2) = - ALI( 1,IB) / 2D0 / ALIB
C AK( 3) = AK(2)
C ELSE IF (NNOR.EQ.2) THEN
C AK( 3) = -(ALI( 1,IB)+ALI( 2,IB)) / ALIB

```

```

C      ENDIF
C      .. SHEAR COMPONENTS
C      I3D = 4
C      DO 220 IFLD=NNOR+1, NCOMP
C          AK ( I3D ) = ALI ( IFLD, IB ) / ALIB
C          I3D = I3D + 1
220  CONTINUE
C      ENDIF
C      .. <DEP:K>
C      DEPK = DEP3 ( 1 ) * AK ( 1 ) + DEP3 ( 2 ) * AK ( 2 ) + DEP3 ( 3 ) * AK ( 3 ) +
C            ( DEP3 ( 4 ) * AK ( 4 ) + DEP3 ( 5 ) * AK ( 5 ) + DEP3 ( 6 ) * AK ( 6 ) ) * 2D0
C
C      IF ( DEPK .GE. 0.0 ) THEN
C          DEPK = DEPK
C      ELSE
C          DEPK = 0D0
C      ENDIF
C
C      IF ( KM .EQ. 1 ) THEN
C      C      .. MODEL-1
C      DO 300 I=1, NCOMP
C          IF ( I .LE. NNOR ) THEN
C              DALI ( I, IB ) = ZETA ( IB ) * ( 2D0/3D0 * R ( IB ) * DEPSP ( I ) -
C                                     HF1 * DEPK * ALI ( I, IB ) )
C          ELSE
C              DALI ( I, IB ) = ZETA ( IB ) * ( 2D0/3D0 * R ( IB ) * DEPSP ( I ) / 2D0 -
C                                     HF1 * DEPK * ALI ( I, IB ) )
C          ENDIF
300  CONTINUE
C      ELSE
C      C      .. MODEL-2 ( USE M )
C
C      XDPEK = ( ALIB / R ( IB ) ) ** AM * DEPK
C      DO 400 I=1, NCOMP
C          IF ( I .LE. NNOR ) THEN
C              DALI ( I, IB ) = ZETA ( IB ) * ( 2D0/3D0 * R ( IB ) * DEPSP ( I ) -
C                                     XDPEK * ALI ( I, IB ) )
C          ELSE
C              DALI ( I, IB ) = ZETA ( IB ) * ( 2D0/3D0 * R ( IB ) * DEPSP ( I ) / 2D0 -
C                                     XDPEK * ALI ( I, IB ) )
C          ENDIF
400  CONTINUE
C      ENDIF
C
C      ENDIF

```

```

1000 CONTINUE
C      RETURN
C      END
C      SUBROUTINE YOWDAT ( NCOMP , PHA11 , PHA12 , DT , AL ,
C                        DALSDT )
C-----
C * FUNCTION OF THIS SUBROUTINE
C   SET TEMPERATURE DIFFERENCE OF BACK STRESS FOR OHNO-WANG MODEL
C   PROGRAMED BY K. GARATANI ON DEC. 1995 ( VER 1.0 )
C-----
C * INPUT VARIABLES
C   NCOMP      D. ARG ( INT )      : STRESS OR STRAIN COMPONENTS COUNT
C   PHA11      D. ARG ( INT )      : TEMPEARTURE DEP. FUNC. BEFORE INC.
C   PHA12      D. ARG ( INT )      : TEMPEARTURE DEP. FUNC. AFTER INC.
C   DT         D. ARG ( INT )      : TEMPEARTURE INCREMENT
C   AL         ( I ) D. ARG ( INT ) : BACK STRESS ( ON TO )
C-----
C * OUTPUT VARIABLES
C   DALSDT ( I ) D. ARG ( REAL * 8 ) : D ( AL * ) / DT ( ON T )
C-----
C      IMPLICIT REAL * 8 ( A-H, O-Z )
C      DIMENSION AL ( * ) , DALSDT ( * )
C
C ***** SET VALUE ( CAREFUL TO AL ( TO ) AND AL * ( T ) )
C
C      IF ( DT .EQ. 0.0 ) THEN
C          DPDT = 0.0
C      ELSE
C          DPDT = ( PHA12 - PHA11 ) / DT
C      ENDIF
C
C      DO 100 I=1, NCOMP
C          DALSDT ( I ) = DPDT * AL ( I )
100  CONTINUE
C
C      RETURN
C      END
C      SUBROUTINE YOWEPV ( KFIELD , NCOMP , NNOR , NB , KM ,
C                        E1 , P011 , E2 , P012 , PHA12 ,
C                        ALP , AM , ZETA , R , DT ,
C                        DKAPDT , DALSDT , DFDS , ALI , EPSE ,
C                        DEPS , DSIG , DALI , DEPSP , DEPB )
C-----
C * FUNCTION OF THIS SUBROUTINE

```

```

C      GET EACH INNER VARIABLES INCREMENTS BY GIVEN STRAIN ON PLASTIC
C      FOR OHNO-WANG MODEL
C      PROGRAMED BY K. GARATANI ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C      KFIELD  D. ARG (INT)   : FINAS STRESS FIELD CODE
C      NCOMP   D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C      NNOR    D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C      NB      D. ARG (INT)   : TOTAL BACK STRESS COUNT
C      KM      D. ARG (INT)   : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C      (MODEL-2 USES M-VALUE)
C      E1      D. ARG (REAL*8) : ELASTIC MODULAS BEFORE INCREMENT
C      PO11    D. ARG (REAL*8) : POISSON'S RATIO BEFORE INCREMENT
C      E2      D. ARG (REAL*8) : ELASTIC MODULAS AFTER INCREMENT
C      PO12    D. ARG (REAL*8) : POISSON'S RATIO AFTER INCREMENT
C      PHA12   D. ARG (REAL*8) : TEMP. DEP. FUC PHA1(T) AFTER INC.
C      ALP     D. ARG (REAL*8) : THERMAL EXPANSION RATIO
C      AM      D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C      ZETA(I) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(I)
C      R (I)   D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (I)
C      DT      D. ARG (REAL*8) : TEMPERATURE INCREMENT
C      DKAPDT  D. ARG (REAL*8) : TEMP. DIFF. OF DRAG STRESS D(KAPPA)/DT
C      DALSDT(I) D. ARG (REAL*8) : TEMP. DIFF. OF BACK STRESS D(AL*)/DT
C      DFDS (I) D. ARG (REAL*8) : FLOW VECTOR (DF/DS)
C      ALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C      EPSE (I) D. ARG (REAL*8) : ELASTIC STRAIN
C      DEPS (I) D. ARG (REAL*8) : STRAI INCREMENT
C-----
C * OUTPUT VARIABLES
C      DSIG (I) D. ARG (REAL*8) : STRESS INCREMENT
C      DALI (6, J) D. ARG (REAL*8) : J-TH BACK STRESS INCREMENT ON TO
C      DEPSP(I) D. ARG (REAL*8) : PLASTIC STRAIN INCREMENT
C      DEPB    D. ARG (REAL*8) : EQUIVALENT PLASTIC STRAIN INCREMENT
C-----
C      IMPLICIT REAL*8 (A-H, O-Z)
C
C      DIMENSION ZETA(*), R(*), DALSDT(*), DFDS(*), EPSE(*), ALI (6, *),
C      DEPS(*), DSIG(*), DALI (6, *), DEPSP(*)
C      DIMENSION DEP(6, 6), DGEP(6), DEPSE(6), DEPST(6)
C
C ***** GET STRESS INCREMENT
C      .. S-S MATRIX 'DEP'
C
C      CALL YOWHD ( NCOMP , NNOR , NB , KM , AM ,
C      ZETA , R , DFDS , ALI , HD )

```

```

C      HD = HD * PHA12
C
C      CALL YFDEP ( KFIELD , NCOMP , 6 , E2 , PO12 ,
C      HD , DFDS , DEP )
C
C      .. PSEUDO STRESS VECOR 'DGEP'
C
C      CALL YFGEP ( KFIELD , NCOMP , 6 , E1 , PO11 ,
C      E2 , PO12 , ALP , HD , DT ,
C      DKAPDT , DALSDT, DFDS , EPSE , DGEP )
C
C      .. DSIG = DE*DEPS + DGEP
C
C      DO 110 I=1, NCOMP
C      DSIG(I) = 0.0
C      DO 100 J=1, NCOMP
C      DSIG(I) = DSIG(I) + DEP(I, J)*DEPS(J)
C 100 CONTINUE
C      DSIG(I) = DSIG(I) + DGEP(I)
C 110 CONTINUE
C
C ***** GET THERMAL STRAIN INCREMENT
C
C      DO 200 I=1, NCOMP
C      IF (I.LE.NNOR) THEN
C      DEPST(I) = ALP*DT
C      ELSE
C      DEPST(I) = 0.0
C      ENDIF
C 200 CONTINUE
C
C ***** GET PLASTIC STRAIN INCREMENT
C      .. ELASTIC STRAIN INCREMENT
C
C      CALL YFEPSE( NCOMP , NNOR , E2 , PO12 , DSIG ,
C      DEPSE )
C
C      .. PLASTIC STRAIN INCREMENT
C
C      DO 300 I=1, NCOMP
C      DEPSP(I) = DEPS(I) - DEPST(I) - DEPSE(I)
C 300 CONTINUE
C
C ***** GET EQUIVALENT PLASTIC STRAIN INCREMENT
C
C      CALL YFPBAR( KFIELD , DEPSP , DEPB )
C
C ***** GET EACH BACK STRESS INCREMENTS

```

```

C
C      CALL YOWDA ( NCOMP , NNOR , NB , KM , AM ,
C                  ZETA , R , ALI , DEPS , DALI )
C
C      RETURN
C      END
C      SUBROUTINE YOWFI ( NCOMP , NNOR , AL , R , ALBAR , FI )
C-----
C * FUNCTION OF THIS SUBROUTINE
C      CALCULATE DYNAMIC RECOVER TERM FUNCTION FI FOR OHNO-WANG MODEL
C      PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C      NCOMP  D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C      NNOR   D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C      AL(1)  D. ARG (REAL*8) : ONE OF BACK STRESS ON TO
C      R      D. ARG (REAL*8) : ONE OF STRESS PARAMETER R
C-----
C * OUTPUT VARIABLES
C      ALBAR  D. ARG (REAL*8) : EQ. B. STRESS ( 3/2 * AL:AL (EQ. 5) )
C      FI     D. ARG (REAL*8) :  $FI = ALBAR^{**2} - R^{**2}$ 
C-----
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION AL(*)
C
C ***** EPS2 FOR ALLOWANCE OF FI
C
C      DATA EPS2 / 0.0100 /
C
C ***** GET ALPHA-BAR**2 TERM
C      .. NORMAL COMPONENTS
C      IF (NNOR.EQ.0) THEN
C        ALB2 = 0.0
C      ELSE IF (NNOR.EQ.1) THEN
C        ALB2 = 1.500 * AL(1)**2
C      ELSE IF (NNOR.EQ.2) THEN
C        ALB2 = AL(1)**2 + AL(2)**2 + (AL(1)+AL(2))**2
C      ELSE IF (NNOR.EQ.3) THEN
C        ALB2 = AL(1)**2 + AL(2)**2 + AL(3)**2
C      ENDIF
C      .. SHEAR COMPONENTS (DOUBLED)
C      DO 100 I=NNOR+1,NCOMP
C        ALB2 = ALB2 + 2DO * AL(I)**2
C 100 CONTINUE
C

```

```

      ALB2 = 1.500 * ALB2
C
C ***** EQUIVALENT BACK STRESS
C
C      ALBAR = SQRT(ALB2)
C
C ***** GET DYNAMIC RECOVER TERM FUNCTION
C
C      FI = ALB2 - (R*(100-EPS2))**2
C
C      RETURN
C      END
C      SUBROUTINE YOWHD ( NCOMP , NNOR , NB , KM , AM ,
C                       ZETA , R , DFDS , ALI , HD )
C-----
C * FUNCTION OF THIS SUBROUTINE
C      SET PLASTIC HARDENING COEFF. FOR OHNO-WANG MODEL
C      PROGRAMED BY K. GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C      NCOMP  D. ARG (INT)   : STRESS OR STRAIN COMPONENTS COUNT
C      NNOR   D. ARG (INT)   : NORMAL STRESS OR STRAIN COMP. COUNT
C      NB     D. ARG (INT)   : TOTAL BACK STRESS COUNT
C      KM     D. ARG (INT)   : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C                                     (MODEL-2 USES M-VALUE)
C      AM     D. ARG (REAL*8) : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C      ZETA(1) D. ARG (REAL*8) : I-TH B. STRESS PARAMETER ZETA(1)
C      R (1)  D. ARG (REAL*8) : I-TH B. STRESS PARAMETER R (1)
C      DFDS(1) D. ARG (REAL*8) : FLOW VECTOR (DF/DS)
C      ALI (6,J) D. ARG (REAL*8) : J-TH BACK STRESS ON TO
C-----
C * OUTPUT VARIABLES
C      HD     D. ARG (REAL*8) : PLASTIC HARDENING COEFFICIENT
C-----
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION ZETA (*),R (*),DFDS(*),ALI (6,*)
C      DIMENSION DFDSN(6),ALIN(6)
C
C ***** INITIALIZE
C
C      HD = 0.0
C
C      .. NORMALIZE DF/DS
C
C      CALL YFNHVV ( 1 , NCOMP , NNOR , DFDS , DFDSN )
C

```



```

C ***** SUM UP EABC BACK STRESS HARDENING TERM FOR MODEL-1,2
C
C      DO 100 IB=1,NB
C          .. GET F1 FUNCTION
C
C      CALL YOWF1 ( NCOMP , NNOR , ALI(1,IB) , R(1B) , ALIB ,
C          F1 )
C
C          .. NORMALIZE ALI(1)
C
C      CALL YFNMV ( 0 , NCOMP , NNOR , ALI(1,IB) , ALIN )
C
C          .. GET LAMDA
C
C      RAMDA = DFDSN(1)*ALIN(1)+ DFDSN(2)*ALIN(2)+ DFDSN(3)*ALIN(3)+
C          ( DFDSN(4)*ALIN(4)+ DFDSN(5)*ALIN(5)+ DFDSN(6)*ALIN(6)
C          ) * 2DO
C
C          .. SET HD
C
C      IF (KM.EQ.1) THEN
C          .. MODEL-1
C
C          IF ( (F1.GE.0.0).AND.(RAMDA.GE.0.0) ) THEN
C              HD = HD + ZETA(1B)*R(1B) * (100-RAMDA**2)
C          ELSE
C              HD = HD + ZETA(1B)*R(1B)
C          ENDIF
C
C      ELSE
C          .. MODEL-2 (USE M)
C
C          IF ( RAMDA.GE.0.0 ) THEN
C
C              ALIBRM = (ALIB/R(1B))**(AM+100)
C              IF (ALIBRM.GT.100) ALIBRM = 100
C
C              HD = HD +
C              ZETA(1B)*R(1B) * ( 100- ALIBRM          * RAMDA**2 )
C              ZETA(1B)*R(1B) * ( 100-(ALIB/R(1B))**(AM+100) * RAMDA**2 )
C          ELSE
C              HD = HD + ZETA(1B)*R(1B)
C          ENDIF
C
C      ENDIF
C
C 100 CONTINUE

```

```

C
C      RETURN
C      END
C      SUBROUTINE YOWMAT( IEXN , IDMAT , TEMP , DTEMP ,
C          NB , KS , KT , KM ,
C          E1 , PO11 , SIGO1 , PHA11 ,
C          E2 , PO12 , SIGO2 , PHA12 ,
C          ALP , AM , ZETA , R )
C-----
C * FUNCTION OF THIS SUBROUTINE
C      SET MATERIAL CONSTATNTS FOR OHNO-WANG MODEL (MODEL-1,2)
C          PROGRAMED BY K.GARATANI ON DEC. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C      IEXN      D. ARG (INT)      : ELEMENT EXTERNAL ID.
C      IDMAT     D. ARG (INT)      : MATERIAL ID.
C      TEMP      D. ARG (REAL*8)   : TEMPERATURE AT PREVIOUS STEP
C      DTEMP     D. ARG (REAL*8)   : TEMPERATURE INCREMENT
C      *         /YBXMAT/         : MATERIAL DATA TABLE (SEE BLOW)
C-----
C * OUTPUT VARIABLES
C      NB        D. ARG (INT)      : TOTAL BACK STRESS COUNT
C      KS        D. ARG (INT)      : KEY OF SUBSTEP (=0:NO, =1:YES)
C      KT        D. ARG (INT)      : KEY OF TEMP. DEP. (=0:NO, =1:YES)
C      KM        D. ARG (INT)      : KEY OF MODEL (=1:MODEL-1, =2:MODEL-2)
C                                     (MODEL-2 USES M-VALUE)
C      E 1,2    D. ARG (REAL*8)   : YOUNG'S MODULAS
C      PO1 1,2  D. ARG (REAL*8)   : POISSON'S RATIO
C      SIGO1,2  D. ARG (REAL*8)   : INITIAL YIELD STRESS
C      PHA11,2  D. ARG (REAL*8)   : TEMPERATURE DEP. FUC PHAI(T)
C      ALP      D. ARG (REAL*8)   : THERMAL EXPANSION RATIO (BY SIMPSON)
C      AM       D. ARG (REAL*8)   : MODEL PARAMETER M (FOR MODEL-2 ONLY)
C      ZETA(1)  D. ARG (REAL*8)   : I-TH B. STRESS PARAMETER ZETA(I)
C      R (1)    D. ARG (REAL*8)   : I-TH B. STRESS PARAMETER R (I)
C                                     (SUFFIX 1,2 MEANS BEFORE OR AFTER INCREMENT)
C-----
C * CALLED SUBROUTINES
C      YFVEXP   : EXTERPOLATE OR INTERPOLATE TABLE FORM VALUE
C-----
C * DATA BLOCK /YBXMAT/ FOR MATERIAL DATA DEFINITION SETTED BY XREAD
C      NTBLPA   PARM. (INT)      : MAX. COUNT OF MATERIAL TABLE
C                                     NTBLPA MUSUT BE MULTIPLE OF 3
C      NMATPA   PARM. (INT)      : MAX. COUNT OF MATERIAL KINDS
C      NMIDPA   PARM. (INT)      : MAX. COUNT OF MATERIAL ID.
C      NKEYPA   PARM. (INT)      : MAX. COUNT OF OPTION KEY
C

```

```

C MID ( K ) /YBXMAT/(INT) : K-TH MATERIAL ID.
C KEY ( 10, K ) /YBXMAT/(INT) : K-TH OPTION KEY DATA
C NTBL( J, K ) /YBXMAT/(INT) : DATA LENGTH OF K-TH MAT. ID.
C AND J-TH KIND (=0:NONE DATA)
C TTBL(I, J, K) /YBXMAT/(REAL*8) : I-TH TEMPERATURE TABLE OF
C K-TH MAT. ID. AND J-TH KIND
C VTBL(I, J, K) /YBXMAT/(REAL*8) : I-TH MATERIAL VALUE TABLE OF
C K-TH MAT. ID. AND J-TH KIND
C -----
C * MATERIAL KIND KEY AS BELOW
C KIND (J) MNEMONIC PHYSICAL MEANING
C 1 E YOUNG'S MODULAS
C 2 NU POISSON'S RATIO
C 3 ALPHA INSTANTANEOUS THERMAL EXPANTION RATIO
C 4 SIGO INITIAL YIELD STRESS
C 5 M CONSTATNT FOR MODEL-2
C 6 PHA1A TEMPERATURE FUNC. CONSTANTS-1
C 7 PHA1B TEMPERATURE FUNC. CONSTANTS-2
C 8 ZETA01 BACK STRESS CONSTANTS FOR TERM-01
C 9 ZETA02 BACK STRESS CONSTANTS FOR TERM-02
C 10 ZETA03 BACK STRESS CONSTANTS FOR TERM-03
C 11 ZETA04 BACK STRESS CONSTANTS FOR TERM-04
C 12 ZETA05 BACK STRESS CONSTANTS FOR TERM-05
C 13 ZETA06 BACK STRESS CONSTANTS FOR TERM-06
C 14 ZETA07 BACK STRESS CONSTANTS FOR TERM-07
C 15 ZETA08 BACK STRESS CONSTANTS FOR TERM-08
C 16 ZETA09 BACK STRESS CONSTANTS FOR TERM-09
C 17 ZETA10 BACK STRESS CONSTANTS FOR TERM-10
C 18 R01 BACK STRESS CONSTANTS FOR TERM-01
C 19 R02 BACK STRESS CONSTANTS FOR TERM-02
C 20 R03 BACK STRESS CONSTANTS FOR TERM-03
C 21 R04 BACK STRESS CONSTANTS FOR TERM-04
C 22 R05 BACK STRESS CONSTANTS FOR TERM-05
C 23 R06 BACK STRESS CONSTANTS FOR TERM-06
C 24 R07 BACK STRESS CONSTANTS FOR TERM-07
C 25 R08 BACK STRESS CONSTANTS FOR TERM-08
C 26 R09 BACK STRESS CONSTANTS FOR TERM-09
C 27 R10 BACK STRESS CONSTANTS FOR TERM-10
C -----
C * OPTION KEYS
C KEY (1, K) /YBXMAT/(INT) : BACK STRESS COUNT (DEF.=10)
C KEY (2, K) /YBXMAT/(INT) : SUBSTEP METHOD (0:YES, 1:NO)
C KEY (3, K) /YBXMAT/(INT) : TEMP. DEPENDENCY (0:YES, 1:NO)
C KEY (4, K) /YBXMAT/(INT) : =0:MODEL-1, =1:MODEL-2 (M)
C -----
C

```

```

IMPLICIT REAL*8 (A-H, O-Z)
C
C DIMENSION ZETA(*), R(*)
C PARAMETER ( NTBLPA = 21 , NMATPA = 27 , NMIDPA = 4 ,
C NKEYPA = 10 )
C
C COMMON /YBXMAT/
C MID ( NMIDPA ) , KEY(NKEYPA, NMIDPA) ,
C NTBL( NMATPA, NMIDPA ) ,
C TTBL( NTBLPA, NMATPA, NMIDPA ) ,
C VTBL( NTBLPA, NMATPA, NMIDPA )
C .. LOCAL ARRAY
C DIMENSION VAL(2, NMATPA)
C
C ***** FIND AND SET MATERIAL ID
C
C DO 100 KMID = 1, NMIDPA
C IF ( 1DMAT.EQ. MID(KMID) ) GOTO 110
100 CONTINUE
C
C WRITE(6, *) ' ***** ERROR ON SUB. YOWMAT : NON-EXISTENT MAT ID. ',
C IDMAT, IEXN
C
C CALL PEXIT
C STOP
C
C 110 CONTINUE
C
C ***** SET KEYS
C .. BACK STRESS COUNT
C NB = KEY(1, KMID)
C .. SUBSTEP
C IF ( KEY(2, KMID).EQ.0 ) THEN
C KS = 1
C ELSE
C KS = 0
C ENDIF
C .. TEMP. DEP.
C IF ( KEY(3, KMID).EQ.0 ) THEN
C KT = 1
C ELSE
C KT = 0
C ENDIF
C .. MODEL-1, 2
C IF ( KEY(4, KMID).EQ.0 ) THEN
C KM = 1
C ELSE

```

```

      KM = 2
    ENDIF
  C
  C ***** SET MATERIAL VALUE AT BEFORE:1 AND AFTER:2 INCREMENT TO 'VAL'
  C
  C      ... TEMP. DEP. PARAMETER (EXCLUDE ALP)
  C      DO 300 INC = 1,2
  C          .. SET TEMPERATURE
  C          IF (INC.EQ.1) THEN
  C              T = TEMP
  C          ELSE
  C              T = TEMP + DTEMP
  C          ENDIF
  C
  C      DO 200 JMAT=1, NMATPA
  C      DO 200 JMAT=1, 2
  C          .. GET TABLE COUNT
  C          LTBL = NTBL(JMAT, KMID)
  C          IF (LTBL.EQ.0) GOTO 9100
  C          .. SET VALUE
  C          IF (LTBL.EQ.1) THEN
  C              VAL(INC, JMAT) = VTBL(1, JMAT, KMID)
  C          ELSE
  C              CALL YFVEXP( T, TTBL(1, JMAT, KMID), VTBL(1, JMAT, KMID),
  C                          LTBL, VAL(INC, JMAT) )
  C          ENDIF
  C
  C      200 CONTINUE
  C
  C      300 CONTINUE
  C      ... TEMP. NOT DEP. PARAMETER
  C      DO 400 JMAT=4, NMATPA
  C          .. GET TABLE COUNT
  C          LTBL = NTBL(JMAT, KMID)
  C          IF (LTBL.NE.0) GOTO 9100
  C          IF (LTBL.EQ.0) VAL( 1, JMAT) = 0.0
  C          .. SET VALUE
  C          IF (LTBL.EQ.1) THEN
  C              VAL( 1, JMAT) = VTBL(1, JMAT, KMID)
  C          ELSE
  C              VAL( 1, JMAT) = 0.0
  C              GOTO 9110
  C          ENDIF
  C
  C      400 CONTINUE
  C

```

```

  C ***** SET MATERIAL CONSTANTS TO EACH DUMMY ARGUMENT (EXCLUDE ALP)
  C
  C      E1 = VAL(1, 1)
  C      E2 = VAL(2, 1)
  C      PO11 = VAL(1, 2)
  C      PO12 = VAL(2, 2)
  C
  C      .. WHEN NO TEMP. DEP. PHA1=1
  C      IF (KT.EQ.0) THEN
  C          PHA11 = 1D0
  C          PHA12 = 1D0
  C      ELSE
  C1996,10,7 PHA11 = VAL(1,6) * EXP(VAL(1,7) / (TEMP +273.0D0) )
  C1996,10,7 PHA12 = VAL(1,6) * EXP(VAL(1,7) / (TEMP+DTEMP+273.0D0) )
  C          PHA11 = VAL(1,6) + VAL(1,7) * (TEMP -650.0D0)**2
  C          PHA12 = VAL(1,6) + VAL(1,7) * (TEMP+DTEMP-650.0D0)**2
  C      ENDIF
  C
  C      SIG01 = VAL(1,4) * PHA11
  C      SIG02 = VAL(1,4) * PHA12
  C
  C      AM = VAL(1,5)
  C
  C      DO 500 IB=1,NB
  C          ZETA(IB) = VAL(1, 7+IB)
  C          R (IB) = VAL(1, 17+IB)
  C      500 CONTINUE
  C
  C ***** SET THERMAL EXPANSION BY SIMPSON'S INTEGRATION RULE
  C
  C      JMAT = 3
  C
  C      LTBL = NTBL(JMAT, KMID)
  C
  C      IF (LTBL.EQ.0) THEN
  C
  C          ALP = 0.0
  C
  C      ELSE IF (LTBL.EQ.1) THEN
  C          ALP = VTBL(1, JMAT, KMID)
  C      ELSE
  C          T = TEMP
  C          CALL YFVEXP( T, TTBL(1, JMAT, KMID), VTBL(1, JMAT, KMID),
  C                      LTBL, ALP00 )
  C          T = TEMP + DTEMP/2.0
  C          CALL YFVEXP( T, TTBL(1, JMAT, KMID), VTBL(1, JMAT, KMID),
  C                      LTBL, ALP05 )
  C

```

```

      T = TEMP + DTEMP
      CALL YFVEXP( T, TTBL(1, JMAT, KMID), VTBL(1, JMAT, KMID),
                 LTBL, ALP10 )
      ALP = ( ALP00 + 4.0*ALP05 + ALP10 ) / 6.0
    ENDIF
  C
  C RETURN
  C
  C .. ERROR ..
9100 CONTINUE
      WRITE(6,*) ' ***** ERROR ON SUB. YOWMAT : NON-EXISTENT MATERIAL',
                ' CONSTANTS ', IEXN, IDMAT, JMAT
      CALL PEXIT
      STOP
  C
  C9110 CONTINUE
  C WRITE(6,*) ' ***** ERROR ON SUB. YOWMAT : ILLEGAL MATERIAL DATA'
  C CALL PEXIT
  C STOP
  C
  C END
  C SUBROUTINE YXINT5(CHR5, INT)
  C-----
  C * FUNCTION OF THIS SUBROUTINE
  C READ 5 COLUMNED INTEGER DATA (CALLED BY XREAD)
  C PROGRAMED BY K. GARATANI (CRC) ON NOV. 1995 (VER 1.0)
  C-----
  C * INPUT VARIABLES
  C CHR5 D. ARG (CHR*5) : 5 CLUMNED INTEGER CHARACTER
  C-----
  C * OUTPUT VARIABLES
  C INT D. ARG (INT) : INTEGER VALUE
  C-----
  C
  C CHARACTER CHR5*5
  C
  C ***** READ INTEGER DATA EXCEPT BLUNK
  C
  C DO 100 IC=5, 1, -1
  C
  C IF ( CHR5(IC:IC).NE.1H ) THEN
  C READ(CHR5(1:IC), '(15)') INT
  C GOTO 900
  C ENDIF
  C
  C 100 CONTINUE
  C
  C INT = 0

```

```

  C
  C 900 CONTINUE
  C
  C RETURN
  C END
  C SUBROUTINE YXREAD(LU5, LC, KEND, LINE)
  C-----
  C * FUNCTION OF THIS SUBROUTINE
  C READ 1 LINE DATA AND PRINT (CALLED BY XREAD)
  C BY SKIPPING '$' OR 40 BLUNK HEADED LINE
  C PROGRAMED BY K. GARATANI (CRC) ON NOV. 1995 (VER 1.0)
  C-----
  C * INPUT VARIABLES
  C LU5 D. ARG (INT) : FILE CODE TO READ
  C *LC D. ARG (INT) : CURRENT LINE COUNT
  C-----
  C * OUTPUT VARIABLES
  C *LC D. ARG (INT) : CURRENT LINE COUNT (ADDED)
  C KEND D. ARG (CHR*80) : KEY OF END OF FILE (0:NOT, 1:YES)
  C LINE D. ARG (CHR*80) : READING DATA
  C-----
  C * METHOD
  C-----
  C
  C CHARACTER LINE*80, BLUNK*40
  C
  C DATA BLUNK / ' /
  C
  C ***** READ AND PRINT LINE
  C
  C 10 CONTINUE
  C
  C READ (LU5, 501, END=9000) LINE
  C LC = LC + 1
  C CALL PAGE(1, IDUMMY)
  C WRITE( 6 , 601 ) LC, LINE
  C
  C .. SKIP WHEN ¥ OR BLUNK
  C IF ( (LINE(1: 1).EQ.1H$ ) .OR.
  C (LINE(1:40).EQ.BLUNK) ) GOTO 10
  C
  C .. NOMAL RETURN
  C KEND = 0
  C RETURN
  C
  C .. END OF FILE
  C 9000 CONTINUE
  C
  C KEND = 1

```

```

RETURN
C
501 FORMAT(A80)
601 FORMAT(10X, 15, '-', 4X, A80)
END
SUBROUTINE YXRHED
C-----
C * FUNCTION OF THIS SUBROUTINE
C PRINT HEADER FOR USER MATERIAL SECTION (CALLED BY XREAD)
C PROGRAMED BY K. GARATANI (CRC) ON NOV. 1995 (VER 1.0)
C-----
C * INPUT VARIABLES
C NONE
C-----
C * OUTPUT VARIABLES
C NONE
C-----
C * UTILITY SUBROUTINES
C PAGE(11,12) : PAGE CONTROLLER (CALL BEFORE PRINT)
C WHERE 11: LINE COUNT, IF 11<=0 EJECT PAGE
C 12: DUMMY
C-----
C ***** EJECT PAGE AND PRINT HEADER LINE
C
CALL PAGE(0, IDUMMY)
CALL PAGE(4, IDUMMY)
WRITE(6, 6001)
C
RETURN
C
6001 FORMAT(1H, 37X, 'USER MATERIAL CONSTANTS' /
1H0, 28X, '1 2 3 4',
'5 6 7 8' /
1H, 19X, '1.....0.....0.....0.....',
'0.....0.....0.....0.....0')
END
SUBROUTINE DBMATN
IMPLICIT INTEGER (A-Z)
INTEGER LBNAME(2), KIND(2), K19OPT(9), FMTCHK(30)
INTEGER NDM2CR(5), NDMSWL(5)
INTEGER NMFMT(2), NMFMTS(11)
INTEGER LHKTD(2), LHKSWE(2), LHKVIS(2)
INTEGER NMOPT(4,6), NMPRO(2,32)
REAL BB(40), RC(8), SCALE, VALUE, TEMP
LOGICAL ALTER, CDALPH, CDNULL

```

DBMATN 2
DBMATN25
MU37 1
DBMATN28
DBMATN29
DBMATN30
MU37 2
DBMATN33
DBMATN35

```

COMMON /XF1NAS/ DC(8)
COMMON /IFPX2 / CBDICT(3,1)
COMMON /IFPCM1/ INP1, RDMODE, CRDSEQ, CROGRP(6)
COMMON /IFPCM3/ NZ, IBUF1, IBUF2, IBUF3, WKTOP, ZPNTRS(2,4)
COMMON /TBXTBL/ NUMTAB, BTBL(21), XTBL(20), ITBL(20), TBBOTM
COMMON /KBIT32/ KBIT(32)
COMMON /WKOCOM/ IB(40), INDEX(40)
COMMON /ZZZZZ/ Z(1)
EQUIVALENCE ( IB(1), BB(1) )
EQUIVALENCE ( DC(1), RC(1) )
COMMON /MXNUM/
* MXNDOF, MXNX, MXENOD, MXEDOF, MXEGED, MXEIP, /MXNUM/3
* MXSCM, MXECM, MXNRD, MXMDAT, MXEPD, MXCOML, /MXNUM/4
* MXEPS1, MXSIG1, MXINPE, MXPROP, MXLPRO, MXBDAT, /MXNUM/5
* MXSDF1, MXMDFG, MXMRAT, MXLDAT, MXED, MXEFP, /MXNUM/6
* MXFCM, /MXNUM/7
DATA NIND / 30 /
DATA NINDWD / 6 /
DATA LINDEX / 30 /
DATA NMAT / 0 /
DATA MLMATN / 0 /
DATA MLHARD / 0 /
DATA MLHRDC / 0 /
DATA KTDPT / 0 /
DATA KXMAT1 / 0 /
DATA BLANKS / 4H /
DATA LHMATN / 4HMATN /
DATA LHUSER / 4HUSER /
DATA LHKTD(2) / 4HKTDP, 4HT /
DATA LHKSWE(2) / 4HKSWE, 4HLL /
DATA LHKVIS(2) / 4HKVIS, 4HCO /
DATA K19OPT / 1, 2, 3, 4, 5, 6, 9, 12, 14 /
DATA LNMOP(2) / 6 /
DATA NMOPT / 4HEPLA, 4HST L, 10, 3, 4HCREE, 4HP L1, 11, 2
, 4HINTE, 4HRACT, 7, 1, 4HCC R, 4HULE, 8, 1
, 4HMU R, 4HULE, 13, 3, 4HR CR, 4HEEP, 13, 3
/
DATA LNMPRO / 32 /
DATA NMPRO / 4HE, 4H, 4HNU, 4H, 4HALPH, 4HA
, 4HRHO, 4H, 4HSIGY, 4H, 4HSIGY, 4HI0
, 4HHDAS, 4HH, 4HC, 4H, 4HC10, 4H
, 4HCREE, 4HP, 4HSWEL, 4HLING, 4HMU, 4H
, 4HK, 4H, 4HA1, 4H, 4HA2, 4H
, 4HA3, 4H, 4HA4, 4H, 4HA5, 4H
, 4HA6, 4H, 4HB1, 4H, 4HB2, 4H
, 4HB3, 4H, 4HB4, 4H, 4HB5, 4H

```

DBMATN37
DBMATN38
/IFPCM12
/IFPCM32
/TBXTBL2
DBMATN42
DBMATN43
DBMATN44
DBMATN46
DBMATN47
/MXNUM/2
/MXNUM/3
/MXNUM/4
/MXNUM/5
/MXNUM/6
/MXNUM/7
MU37 3
DBMATN57
DBMATN58
DBMATN60
DBMATN61
DBMATN62
DBMATN63
DBMATN64
MU48A 1
DBMATN66
DBMATN67
MU48A 2
DBMATN70
DBMATN71
DBMATN72
DBMATN75
DBMATN78
DBMATN79
DBMATN80
DBMATN81
DBMATN82
MU37 4
DBMATN86
DBMATN87
DBMATN88
DBMATN89
DBMATN90
DBMATN91
DBMATN92
DBMATN93

```

-          , 4HB6 ,4H      , 4HVISC,4HO  , 4HA7 ,4H      MU37  5
-          , 4HA8 ,4H      , 4HA9 ,4H      , 4HA10 ,4H     MU37  6
-          , 4HHDAS,4HHC  , 4HHDAS,4HH10 /    DBMATN95
DATA FMTCHK / 108, 108, 108, 108, 0, 108, 0, 0, 0, 0 DBMAT101
-          , 0, 108, 0,3982,3982,3982,3982,3982,3982, 0 MU37  7
-          , 0, 0, 0, 0, 0, 0, 3982,3982,3982,3982 / MU37  8
DATA NMFMTS / 4HF1 ,4HF2 ,4HF3 ,4HF4 ,4HF5 ,4HF6 ,4HF7    DBMAT104
-          , 4HF8 ,4HF9 ,4HF10 ,4HF11 /      DBMAT105
DATA NDM2CR / 3, 3, 5, 4, 5 /                MU31X 28
DATA NDMSWL / 6, 11, 8, 10, 6 /             DBMAT111
DATA IDSTBL / 1 /                           DBMAT121
DATA MATTBL / 2 /                           DBMAT122
DATA OPTTBL / 3 /                           DBMAT123
DATA ADRTBL / 4 /                           DBMAT124
DATA DATTBL / 5 /                           DBMAT125
IF( CDDICT(2,7).LE.1 ) GO TO 8000
A37777 = SHIFTR(COMPLF(0),1)
A40000 = COMPLF(A37777)
NUMTAB = 5
XTBL(IDSTBL) = 20
ITBL(IDSTBL) = 20
XTBL(MATTBL) = 20+NINDWD
ITBL(MATTBL) = 0
XTBL(OPTTBL) = 100
ITBL(OPTTBL) = 100
XTBL(ADRTBL) = 20
ITBL(ADRTBL) = 20
XTBL(DATTBL) = 200
ITBL(DATTBL) = 400
LW = IBUF2-WKTOP
CALL TBINIT( Z(WKTOP), LW )
CALL LFNMBR( LHMATN, MATN )
IF( MATN.GE.0 ) GO TO 9000
MATN = IABS(MATN)
CALL OPENS3( MATN, Z(IBUF2), 4HWT )
CALL CDREX( 7 )
40 CALL CDREAD( IEOR )
IF( IEOR.NE.0 ) GO TO 7000
50 CALL CDT01( 1,5, MATID, 0, JSTAT )
IF( MATID.GT.0 ) GO TO 80
IF( JSTAT.GE.0 ) CALL IFPMS1( 2082, MATID, 0 )
70 CALL CDREAD( IEOR )
IF( IEOR.NE.0 ) GO TO 7000
IF( .NOT.CDNULL(1,10) ) GO TO 50
GO TO 70
80 IF( NMAT.LE.0 ) GO TO 90

```

```

J = WKTOP
DO 88 I=1, NMAT
IF( MATID.NE. ANDF(Z(J), A37777) ) GO TO 85
CALL IFPMS1( 2081, MATID, 0 )
GO TO 70
85 J = J+1
88 CONTINUE
90 NOPT = 0
NPROP = 0
KPHARD = 0
KMLIB = 0
DO 95 I=1, LINDEX
INDEX(I) = 0
95 CONTINUE
CRDSQ1 = CRDSEQ
CALL CDT0H( 11, 10, LBNAME(I) )
IF( LBNAME(I).EQ.LHUSER ) GO TO 440
IF( LBNAME(I).EQ.BLANKS ) GO TO 120
CALL CDT0I( 6, 5, KMLIB, 0, DMY )
KMLIB = MAXO(1, KMLIB)
120 JCOL = 21
DO 155 I1=1, 9
CALL CDT0I( JCOL, 5, KK, 0, JSTAT )
IF( JSTAT.NE.1 ) GO TO 150
NOPT = NOPT+1
CODE = K19OPT(11)
IB(1) = LSHIFT( CODE, 16 ) + 1
IB(2) = KK
CALL TBADW( OPTTBL, IB(1), 2 )
IF( I1.NE.3 ) GO TO 133
KPHARD = KK
MLHARD = MAXO( MLHARD, KPHARD )
133 IF( I1.NE.5 ) GO TO 150
MLHRDC = MAXO( MLHRDC, KK )
150 JCOL = JCOL+5
155 CONTINUE
200 CALL CDREAD( IEOR )
IF( IEOR.NE.0 ) GO TO 3000
IF( CDNULL(1,10) ) GO TO 210
IEOR = -3
GO TO 3000
210 CALL CDT0H( 11, 10, KIND(1) )
DO 215 I1=1, LNMOPT
IF( KIND(1).EQ.NMOPT(1, I1) .AND.
- KIND(2).EQ.NMOPT(2, I1) ) GO TO 300
215 CONTINUE

```

```

DBMAT171
DBMAT172
MU48A  7
DBMAT175
DBMAT176
DBMAT177
DBMAT178
MU48A 10
MU48A 11
MU48A 12
MU48A 13
MU48A 14
MU48A 15
MU48A 16
MU48A 18
MU48A 19
MU48A 20
MU48A 22
MU48A 23
MU48A 24
MU48A 25
DBMAT198
DBMAT199
DBMAT200
DBMAT201
DBMAT202
DBMAT203
DBMAT204
DBMAT205
DBMAT207
DBMAT208
DBMAT209
DBMAT211
DBMAT212
DBMAT213
DBMAT214
DBMAT220
DBMAT221
MU48A 26
MU48A 27
MU48A 28
DBMAT224
DBMAT225
DBMAT226
DBMAT227
DBMAT228

```

```

DO 220 I1=1, LNMPRO
  IF( KIND(1).EQ.NMPRO(1, I1) .AND.
  -   KIND(2).EQ.NMPRO(2, I1) ) GO TO 500
220 CONTINUE
  CALL IFPMS1( 2104, 0, KIND(1) )
230 CALL CDREAD( IEOR )
  IF( IEOR.NE.0 ) GO TO 3000
  IF( CDNULL(1, 10) ) GO TO 233
  IEOR = -3
  GO TO 3000
233 IF( CDALPH(11, 10) ) GO TO 210
  GO TO 230
300 CODE = NMOPT(3, I1)
  LDAT = NMOPT(4, I1)
  GO TO (310, 320, 330, 340, 360, 360), I1
310 CALL CDTOR( 21, BB(2), 0.0, DMY )
  CALL CDT01( 31.5, IB(3), 10, DMY )
  CALL CDTOR( 41, BB(4), 1.0, DMY )
  GO TO 480
320 CALL CDTOR( 21, BB(2), 0.0, DMY )
  CALL CDTOR( 41, BB(3), 1.0, DMY )
  GO TO 480
330 CALL CDTOR( 21, BB(2), 0.3, DMY )
  GO TO 480
340 CALL CDTOR( 21, BB(2), 0.0, DMY )
  GO TO 480
360 CALL CDTOR( 21, BB(2), 0.5, DMY )
  CALL CDTOR( 31, BB(3), 1.0, DMY )
  CALL CDTOR( 41, BB(4), 0.0, DMY )
  GO TO 480
440 CODE = 15
  LDAT = 3
  KMLIB = -1
  KXMAT1 = 1
  CALL CDT01( 21, 5, IB(2), 0, DMY )
  CALL CDT01( 26, 5, IB(3), 0, DMY )
  CALL CDT01( 31, 5, IB(4), 0, DMY )
  IB(2) = MAXO(0, IB(2))
  IB(3) = MAXO(0, IB(3))
  IB(4) = MAXO(0, IB(4))
  IF( IB(2).GT.9 ) THEN
    WRITE(6, 450) 2871, IB(2)
450 FORMAT(1H0, '*** USER WARNING MESSAGE', 15/
  .      10X, 'USER VECTOR STATUS VARIABLE COUNT', 15,
  .      ' EXCEED 9.')
  ENDIF

```

```

DBMAT229
DBMAT230
DBMAT231
DBMAT232
DBMAT234
DBMAT235
DBMAT236
MU48A 30
MU48A 31
MU48A 32
MU48A 33
DBMAT239
DBMAT243
DBMAT244
DBMAT245
DBMAT248
DBMAT249
DBMAT250
DBMAT251
DBMAT254
DBMAT255
DBMAT256
DBMAT259
DBMAT260
DBMAT263
DBMAT264
DBMAT268
DBMAT269
DBMAT270
MU48A 34
MU48A 37
MU48A 38
MU48A 39
MU48A 40
MU48A 41
MU48A 42
MU48A 43
MU48A 44
MU48A 45
MU48A 46
KG15 1
KG15 2
KG15 3
KG15 4
KG15 5
KG15 6

```

```

  IF( IB(2).GT.19.OR. IB(3).GT.9 .OR. IB(4).GT.9 )
  -   CALL IFPMS1( 2870, 0, 0 )
480 NOPT = NOPT+1
  IB(1) = LSHIFT( CODE, 16 ) + LDAT
  CALL TBADW( OPTTBL, IB(1), 1+LDAT )
  IF( CODE.EQ.15 ) GO TO 3000
  GO TO 200
500 KPROP = I1
  JDATO = XTBL(DATTBL)
  IF( KPROP.EQ.10 ) GO TO 800
  IF( KPROP.EQ.11 ) GO TO 900
  IF( KPROP.EQ.26 ) GO TO 1100
  ALTER = .FALSE.
  SCALE = 1.0
  IF( KPHARD.LE.1 ) GO TO 550
  IF( KPROP.NE.31 ) GO TO 513
  IF( KPHARD.NE.3 ) GO TO 530
  KPROP = 8
  GO TO 525
513 IF( KPROP.NE.7 ) GO TO 516
  IF( KPHARD.EQ.3 ) GO TO 550
  KPROP = 8
  GO TO 525
516 IF( KPROP.NE.32 ) GO TO 550
  KPROP = 9
525 ALTER = .TRUE.
  SCALE = 2.0/3.0
  GO TO 550
530 CALL IFPMS1( 2519, KPHARD, KIND(1) )
  GO TO 230
550 IB(1) = NMPRO(1, KPROP)
  IB(2) = NMPRO(2, KPROP)
  IF( .NOT.CDALPH(21, 5) ) GO TO 610
  CALL CDT0H( 21, 5, NMFMT(1) )
  DO 555 KYFMT=2, 7
  IF( NMFMT(1).EQ.NMFMTS(KYFMT) ) GO TO 560
555 CONTINUE
  CALL IFPMS1( 2104, 0, NMFMT(1) )
  GO TO 230
560 IF( ANDF(FMTCHK(KPROP), KBIT(KYFMT)).EQ.0 ) GO TO 570
  CALL IFPMS1( 2522, NMFMT(1), KIND(1) )
570 CALL CDT01( 26, 5, M, 0, JSTAT )
  IF( KYFMT.GE.5 ) GO TO 572
  IF( M.GE.1 ) GO TO 575
  IF( JSTAT.GE.0 ) CALL IFPMS1( 2523, M, 0 )
  GO TO 230

```

```

KG15 7
KG15 8
DBMAT273
DBMAT274
DBMAT275
MU48A 50
DBMAT276
DBMAT280
DBMAT282
DBMAT284
DBMAT285
DBMAT286
DBMAT292
DBMAT293
DBMAT294
MU37 9
DBMAT296
DBMAT297
DBMAT298
DBMAT299
DBMAT300
DBMAT301
DBMAT302
MU37 10
DBMAT304
DBMAT305
DBMAT306
DBMAT307
DBMAT309
DBMAT310
DBMAT314
DBMAT315
DBMAT316
DBMAT318
DBMAT319
DBMAT320
DBMAT321
DBMAT323
DBMAT324
DBMAT327
DBMAT329
DBMAT332
DBMAT334
DBMAT335
DBMAT337
DBMAT338

```

```

572 IF( M.GE.2 ) GO TO 575
      IF( JSTAT.GE.0 ) CALL IFPMS1( 2524,M,0 )
      GO TO 230
575 IB(3) = KYFMT
      GO TO ( 610,620,630,640,650,660,670), KYFMT
610 IF( ANDF(FMCHK(KPROP),KBIT(1)).NE.0 )
      - CALL IFPMS1( 2522,NMFMTS(1),KIND(1) )
      CALL CDTOR( 21, VALUE, 0.0, DMY )
      IB(3) = 1
      IB(4) = 0
      IB(5) = 0
      BB(6) = VALUE*SCALE
      CALL TBADW( DATTBL, IB(1), 6 )
      GO TO 1800
620 KTDPT = 1
      IB(4) = M
      CALL TBADW( DATTBL, IB(1), 4 )
      M1A = M+1
      CALL TBADW( DATTBL, 0, -M1A )
      GO TO 633
630 IB(4) = 0
      IB(5) = M
      CALL TBADW( DATTBL, IB(1), 5 )
      M1A = M+1
633 CALL MATCDV( DATTBL, 2, M1A, .TRUE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      IF( .NOT.ALTER ) GO TO 1800
      K = XTBL(DATTBL)-M1A
      DO 638 I=1,M1A
          RC(K) = RC(K)*SCALE
          K = K+1
638 CONTINUE
      GO TO 1800
640 KTDPT = 1
      IB(4) = M
      CALL TBADW( DATTBL, IB(1), 4 )
      M1A = M+1
      CALL MATCDV( DATTBL, 1, M1A, .FALSE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      JN = XTBL(DATTBL)-M1A
      NWD = 0
      DO 645 I=1,M1A
          N1A = DC(JN)+1
          CALL MATCDV( DATTBL, 2, N1A, .TRUE., IEOR )
          IF( IEOR.NE.0 ) GO TO 1930
          JN = JN+1

```

```

DBMAT339
DBMAT341
DBMAT342
DBMAT343
DBMAT344
DBMAT348
DBMAT349
DBMAT350
DBMAT351
DBMAT352
DBMAT353
DBMAT354
DBMAT355
DBMAT356
DBMAT360
DBMAT361
DBMAT362
DBMAT364
DBMAT365
DBMAT366
DBMAT370
DBMAT371
DBMAT372
DBMAT373
DBMAT375
DBMAT376
DBMAT378
DBMAT379
DBMAT380
DBMAT381
DBMAT382
DBMAT383
DBMAT384
DBMAT388
DBMAT389
DBMAT390
DBMAT392
DBMAT393
DBMAT394
DBMAT396
DBMAT397
DBMAT398
DBMAT399
DBMAT400
DBMAT401
DBMAT402

```

```

      NWD = NWD+M1A
645 CONTINUE
      IF( .NOT.ALTER ) GO TO 1800
      K = XTBL(DATTBL)-NWD
      DO 648 I=1,NWD
          RC(K) = RC(K)*SCALE
          K = K+1
648 CONTINUE
      GO TO 1800
650 KTDPT = 1
660 IB(4) = M
      CALL TBADW( DATTBL, IB(1), 4 )
      M2 = 2*M
      CALL MATCDV( DATTBL, 5, M2, .TRUE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      IF( .NOT.ALTER ) GO TO 1800
      K = XTBL(DATTBL)-M2
      DO 668 I=1,M
          RC(K+1) = RC(K+1)*SCALE
          K = K+2
668 CONTINUE
      GO TO 1800
670 KTDPT = 1
      IB(4) = M
      CALL TBADW( DATTBL, IB(1), 4 )
      CALL MATCDV( DATTBL, 1, M, .FALSE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      JN = XTBL(DATTBL)-M
      DO 678 J=1,M
          N = DC(JN)
          N2A = 2*N+1
          CALL MATCDV( DATTBL, 7, N2A, .TRUE., IEOR )
          IF( IEOR.NE.0 ) GO TO 1930
          JN = JN+1
          IF( .NOT.ALTER ) GO TO 678
          K = XTBL(DATTBL)-N2A+1
          DO 676 I=1,N
              RC(K+1) = RC(K+1)*SCALE
              K = K+2
676 CONTINUE
678 CONTINUE
      GO TO 1800
800 CALL CDTOR( 21, 5, NMFMT(1) )
      CALL CDTOI( 26, 5, KUSER, 0, DMY )
      CALL CDTOR( 51, TEMP, 0., DMY )
      IF( NMFMT(1).EQ.BLANKS .OR. NMFMT(1).EQ.NMFMTS(8) ) GO TO 805

```

```

DBMAT403
DBMAT404
DBMAT406
DBMAT407
DBMAT408
DBMAT409
DBMAT410
DBMAT411
DBMAT412
DBMAT416
DBMAT417
DBMAT418
DBMAT420
DBMAT421
DBMAT422
DBMAT424
DBMAT425
DBMAT426
DBMAT427
DBMAT428
DBMAT429
DBMAT430
DBMAT434
DBMAT435
DBMAT436
DBMAT438
DBMAT439
DBMAT441
DBMAT442
DBMAT443
DBMAT444
DBMAT445
DBMAT446
DBMAT447
DBMAT449
DBMAT450
DBMAT451
DBMAT452
DBMAT453
DBMAT454
DBMAT455
DBMAT456
DBMAT460
DBMAT461
DBMAT462
DBMAT463

```



```

      CALL IFPMS1 ( 2522, NMFMT(1), KIND(1) )
805 IB(1) = KIND(1)
      IB(2) = KIND(2)
      IB(3) = 8
      IB(4) = 1
      BB(5) = TEMP
      IB(6) = 0
      IF( KUSER.EQ.0 ) GO TO 810
      IB(4) = 3
      CALL TBADW( DATTBL, IB(1), 6 )
      GO TO 1800
810 CALL CDT01( 31, 5, IEQ1, 0, DMY )
      CALL CDT01( 36, 5, IEQ2, 0, DMY )
      IF( IEQ1.GE.0 .AND. IEQ1.LE.2 ) GO TO 813
      CALL IFPMS1( 2166, IEQ1, 0 )
      GO TO 1930
813 IF( IEQ2.GE.0 .AND. IEQ2.LE.5 ) GO TO 816
      CALL IFPMS1( 2166, IEQ2, 0 )
      GO TO 1930
816 IB(6) = 1
      IF( IEQ2.NE.0 ) IB(6) = 2
      CALL TBADW( DATTBL, IB(1), 6 )
      N1 = 0
      IF( IEQ1.EQ.0 ) GO TO 830
      CALL CDREAD( IEOR )
      IF( IEOR.NE.0 ) GO TO 1900
      IF( IEQ1.NE.2 ) GO TO 826
      CALL CDTOR( 11, VALUE, 0.0, DMY )
      N1 = INT( VALUE+0.5 )
      IF( N1.GE.1 .AND. N1.LE.4 ) GO TO 824
      CALL IFPMS1( 2167, VALUE, 0 )
      GO TO 1930
824 N1 = 5*N1 + 1
      GO TO 830
826 N1 = 4
830 IB(1) = IEQ1
      IB(2) = N1
      CALL TBADW( DATTBL, IB(1), 2 )
      IF( N1.LE.0 ) GO TO 840
      CALL MATCDV( DATTBL, 2, N1, .FALSE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
840 IF( IEQ2.EQ.0 ) GO TO 1800
      N2 = NDM2CR(IEQ2)
      IB(1) = IEQ2
      IB(2) = N2
      CALL TBADW( DATTBL, IB(1), 2 )

```

```

DBMAT465
DBMAT466
DBMAT467
DBMAT468
DBMAT469
DBMAT470
DBMAT471
DBMAT473
DBMAT474
DBMAT475
DBMAT476
DBMAT479
DBMAT480
DBMAT481
DBMAT483
DBMAT484
DBMAT485
DBMAT487
DBMAT488
DBMAT489
DBMAT490
DBMAT491
DBMAT494
DBMAT495
DBMAT496
DBMAT497
DBMAT498
DBMAT499
DBMAT500
DBMAT501
DBMAT503
DBMAT504
DBMAT505
DBMAT506
DBMAT507
DBMAT510
DBMAT511
DBMAT512
DBMAT514
DBMAT515
DBMAT516
DBMAT519
DBMAT520
DBMAT521
DBMAT522
DBMAT523

```

```

      CALL MATCDV( DATTBL, 2, N2, .TRUE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      GO TO 1800
900 CALL CDT0H( 21, 5, NMFMT(1) )
      CALL CDT01( 26, 5, KUSER, 0, DMY )
      CALL CDTOR( 51, TEMP, 0., DMY )
      IF( NMFMT(1).EQ.BLANKS .OR. NMFMT(1).EQ.NMFMTS(9) ) GO TO 905
      CALL IFPMS1( 2522, NMFMT(1), KIND(1) )
905 IB(1) = KIND(1)
      IB(2) = KIND(2)
      IB(3) = 9
      IB(4) = 1
      BB(5) = TEMP
      CALL PVTPUT( LHKSWE, 1, 1 )
      IF( KUSER.EQ.0 ) GO TO 910
      IB(4) = 3
      CALL TBADW( DATTBL, IB(1), 5 )
      GO TO 1800
910 CALL CDT01( 31, 5, IEQ1, 0, DMY )
      IF( IEQ1.GE.0 .AND. IEQ1.LE.5 ) GO TO 916
      CALL IFPMS1( 2169, IEQ1, 0 )
      GO TO 1930
916 N1 = NDM5WL(IEQ1)
      IB(6) = IEQ1
      IB(7) = N1
      CALL TBADW( DATTBL, IB(1), 7 )
      CALL MATCDV( DATTBL, 2, N1, .TRUE., IEOR )
      IF( IEOR.NE.0 ) GO TO 1930
      GO TO 1800
1100 CALL CDT0H( 21, 5, NMFMT(1) )
      CALL CDT01( 26, 5, KUSER, 0, DMY )
      CALL CDTOR( 51, TEMP, 0., DMY )
      IF( NMFMT(1).EQ.BLANKS .OR. NMFMT(1).EQ.NMFMTS(11) ) GO TO 1105
      CALL IFPMS1( 2522, NMFMT(1), KIND(1) )
1105 IF( KUSER.NE.0 ) GO TO 1110
      CALL IFPMS1( 2165, 0, 0 )
1110 IB(1) = KIND(1)
      IB(2) = KIND(2)
      IB(3) = 11
      IB(4) = 3
      BB(5) = TEMP
      CALL TBADW( DATTBL, IB(1), 5 )
      CALL PVTPUT( LHKVIS, 1, 1 )
      GO TO 1800
1800 IF( INDEX(KPROP).EQ.0 ) GO TO 1810
      CALL IFPMS1( 2526, MATID, KIND(1) )

```

```

DBMAT525
DBMAT526
DBMAT527
DBMAT531
DBMAT532
DBMAT533
DBMAT534
DBMAT536
DBMAT537
DBMAT538
DBMAT539
DBMAT540
DBMAT541
DBMAT543
DBMAT545
DBMAT546
DBMAT547
DBMAT548
DBMAT551
DBMAT552
DBMAT554
DBMAT555
DBMAT556
DBMAT557
DBMAT558
DBMAT559
DBMAT561
DBMAT562
DBMAT563
DBMAT567
DBMAT568
DBMAT569
DBMAT570
DBMAT572
DBMAT573
DBMAT575
DBMAT577
DBMAT578
DBMAT579
DBMAT580
DBMAT581
DBMAT582
DBMAT584
DBMAT585
DBMAT590
DBMAT592

```

```

GO TO 1930
1810 NPROP = NPROP+1
INDEX(KPROP) = NPROP
JADD = JDATO-BTBL(DATTBL)
CALL TBADW( ADRTBL, JADD, 1 )
GO TO 200
1900 CALL IFPMS1( 2518, 0, 0 )
1930 XTBL(DATTBL) = JDATO
IF( IEOR.EQ.0 ) GO TO 200
IF( IEOR.EQ.-1 ) GO TO 230
IF( IEOR.EQ.-2 ) GO TO 210
3000 LOPT = XTBL(OPTTBL) - BTBL(OPTTBL)
LPROP = XTBL(DATTBL) - BTBL(DATTBL)
LMATN = 8 + (1+LOPT) + (3+NINDWD) + NPROP + LPROP
JOPT = 10
JMATN = BTBL(MATTBL)
DC(JMATN ) = LMATN
DC(JMATN+1) = MATID
DC(JMATN+2) = LBNAME(1)
DC(JMATN+3) = LBNAME(2)
DC(JMATN+4) = KMLIB
DC(JMATN+5) = JOPT
DC(JMATN+6) = 0
DC(JMATN+7) = 0
DC(JMATN+8) = 0
DC(JMATN+9) = NOPT
JM = JMATN+10
IF( LOPT.LE.0 ) GO TO 3030
K = BTBL(OPTTBL)
DO 3025 I=1, LOPT
DC(JM) = DC(K)
JM = JM+1
K = K+1
3025 CONTINUE
3030 IF( NPROP.LE.0 ) GO TO 3100
JPROP = JOPT + (1+LOPT)
JIND = JPROP+2
JADD = JPROP+(3+NINDWD)
DC(JMATN+6) = JPROP
DC(JM ) = JIND
DC(JM+1) = JADD
DC(JM+2) = NIND
JM = JM+3
K = 1
DO 3035 I=1, NINDWD
MSK = 0

```

```

DBMAT593
DBMAT594
DBMAT595
DBMAT596
DBMAT597
DBMAT598
DBMAT599
DBMAT600
DBMAT601
DBMAT602
DBMAT603
MU48A 51
MU48A 52
DBMAT606
DBMAT616
DBMAT617
DBMAT618
DBMAT619
DBMAT620
DBMAT621
DBMAT622
DBMAT623
DBMAT624
DBMAT625
DBMAT626
DBMAT627
DBMAT628
DBMAT629
DBMAT631
DBMAT632
DBMAT633
DBMAT634
DBMAT636
DBMAT637
DBMAT638
DBMAT639
DBMAT640
DBMAT643
DBMAT645
DBMAT646
DBMAT647
DBMAT648
DBMAT649
DBMAT650
DBMAT651
DBMAT652
DBMAT653
DBMAT654
DBMAT655

```

```

SFT = 0
DO 3033 J=1, 5
MSK = ORF( MSK, LSHIFT(INDEX(K), SFT) )
K = K+1
SFT = SFT+6
3033 CONTINUE
DC(JM) = MSK
JM = JM+1
3035 CONTINUE
K = BTBL(ADRTBL)
DO 3045 I=1, NPROP
DC(JM) = DC(K) + (JADD+NPROP)
JM = JM+1
K = K+1
3045 CONTINUE
K = BTBL(DATTBL)
DO 3055 I=1, LPROP
DC(JM) = DC(K)
JM = JM+1
K = K+1
3055 CONTINUE
3100 CSEQSV = CRDSEQ
CRDSEQ = CRDSQ1
IF( KMLIB.LE.0 ) GO TO 3200
CALL MATLIB( DC(JMATN), IER, KTDPT )
IF( IER.EQ.0 ) GO TO 3130
CALL IFPMS1( 2143, 0, LBNAME(1) )
GO TO 3200
3130 L = DC(JMATN)+1
IF( L.GT.(TBBOTM-JMATN) ) GO TO 9030
3200 NMAT = NMAT+1
LMATN = DC(JMATN)+1
MLMATN = MAXO( MLMATN, LMATN )
CALL WRITSQ( MATN, DC(JMATN), LMATN, 1 )
CALL PVTPUT( LHKTD, KTDPT, 1 )
IF( KMLIB.LT.0 ) GO TO 3270
CRDSEQ = MATID
CALL CKMATN( DC(JMATN) )
3270 CRDSEQ = CSEQSV
XTBL(OPTTBL) = BTBL(OPTTBL)
XTBL(ADRTBL) = BTBL(ADRTBL)
XTBL(DATTBL) = BTBL(DATTBL)
IF( KMLIB.LT.0 ) MATID = ORF( A40000, MATID )
CALL TBADW( IDSTBL, MATID, 1 )
IF( IEOR.LT.0 ) THEN
IEOR = 0

```

```

DBMAT656
DBMAT657
DBMAT658
DBMAT659
DBMAT660
DBMAT661
DBMAT662
DBMAT663
DBMAT664
DBMAT666
DBMAT668
DBMAT669
DBMAT670
DBMAT671
DBMAT672
DBMAT674
DBMAT676
DBMAT677
DBMAT678
DBMAT679
DBMAT680
DBMAT686
DBMAT687
MU48A 53
MU71U 1
DBMAT691
DBMAT693
DBMAT694
DBMAT695
DBMAT696
DBMAT700
DBMAT701
DBMAT702
DBMAT703
MU40X 9
MU48A 54
DBMAT706
DBMAT707
MU48A 55
DBMAT712
DBMAT713
DBMAT714
MU48A 56
DBMAT716
MU86S 22
MU86S 23

```

```
GO TO 50
ENDIF
IF( 1EOR.EQ.0 ) GO TO 40
7000 IF( NMAT.GT.0 ) GO TO 7020
CALL CLOSSQ( MATN,4HUNLD )
GO TO 8000
7020 IB(1) = NMAT
IB(2) = MLMATN
IB(3) = MLHARD
IB(4) = MLHRDC
IB(5) = 0
IB(6) = KXMAT1
CALL WRTHDR( MATN, IB(1) )
CALL CLOSSQ( MATN,4HR )
ZPNTRS(1,2) = WKTOP
ZPNTRS(2,2) = NMAT
WKTOP = WKTOP+NMAT
MXMDAT = MLMATN
8000 RETURN
9000 CALL MESSAGE(-201, LHMATN, 0)
9030 CALL MESSAGE(-4, LHMATN, NZ)
RETURN
END
```

```
MU86S 24
MU86S 25
MU48A 58
DBMAT723
DBMAT724
DBMAT725
DBMAT727
DBMAT728
DBMAT729
DBMAT730
DBMAT731
MU48A 59
DBMAT733
DBMAT734
DBMAT735
DBMAT736
DBMAT737
DBMAT739
DBMAT743
DBMAT749
DBMAT752
DBMAT753
DBMAT754
```

C. FASTENINGおよび縮合における計算効率の改善

1. 改良の概要

FINASでは、解析モデルの節点自由度の線形な従属関係を与えるデータとして、FASTENGデータが用意され、ユーザーは容易に線形結合の条件を定義することが可能となっている。一方、解析モデルの自由度数が大きい問題に対しては、実際に解くべき自由度数を縮小することにより、計算コストを低減するような静的縮合解析機能が組み込まれている。

両機能ともに、解析モデルの形状、境界条件等により計算時間が多く費やされる場合があるため、両機能にとって処理過程の主要部分であるソルバーモジュールの見直が行われ、前進・後退代入演算の高速化を図るFBSQルーチン(Foward-Backward Substitution, Quick Method)が開発された。

本報告では、前進・後退代入演算の内部ロジックについて、従来の方法とその問題点および改良された新しい方法が述べられている、さらに、流体構造連成問題および静的縮合問題の解析に適用され、計算効率の成果が検討された。

2. 前進・後退代入演算

前進・後退代入演算とは、連立一次方程式 $[A][X]=[B]$ の係数行列が

$$[A]=[L][D][L]^T \quad (1)$$

という形に3角分解されているとき、次の2つの方程式を順次解いて、元の方程式の解 $[X]$ を求めるものである。

$$[L][Y]=[B] \quad (2)$$

$$[L]^T[X]=[D]^{-1}[Y] \quad (3)$$

ここで、 $[L]$ は下3角行列、 $[D]$ は対角行列である。
 $[Y]$ と $[X]$ の各項は次式で得られる。

$$y_{ij} = b_{ij} - \sum_{k=1}^{i-1} l_{ik} y_{kj} \quad (4)$$

$$x_{ij} = y_{ij} - \sum_{k=i+1}^N y_{ij} x_{kj} \quad (5)$$

3. 従来の方法

従来の方法では、メモリーは、図1に示されるように、できるだけ多くの[X]のアンパックされた列のために割り付けられる。このエリアには、初期値として[B]の各列がアンパックされる。以下、

```

┌── K=1,2,3,...,N-1 の順に；
|   [L]の第k列を GETSTR により、1要素ずつ取り出す。取り出した非ゼロ項の
|   行位置を i とすると、i は k ≤ i ≤ N の範囲にある。
|   i = k であれば、それは対角項であるから、Dii として一時的に保存しておく。
|   以下、
|   ┌── k + 1 ≤ i ≤ N の範囲にある Lik について；
|   |   ┌── j=1,2,3,...,p の順に；
|   |   |   Xij := Xij - Lik * Xkj
|   |   └──
|   └──
|   ┌── j=1,2,3,...,p の順に；
|   |   Xkj = Xkj / Dii
|   └──
└──

```

という手順で前進代入を行う。ただし、pはメモリーに割り付けられている[X]の列数である。

前進代入を終えた時点では、[X]の最終行だけが完成している。

一方、後退代入の方は

```

┌── i=N-1,N-2,...,1 の順に；
|   [L]の第 i 列を GETSTB により、1 要素ずつ逆順に（行Nから1に向かって）
|   取り出す。これは[L]Tの第 i 行を逆順に取り出すkotoに相当する。
|   取り出した非ゼロ項の行位置を k とすると
|   ┌── N ≥ k ≥ i + 1 の範囲にあるLkiについて；
|   |   ┌── j=1,2,3,...,p の順に；
|   |   |    $X_{ij} = X_{ij} - L_{ki} * X_{kj}$ 
|   |   └──
|   └──
└──

```

という手順になる。

[B]の列数が多くて、1回の処理で完結しない場合には、上記の手続きを繰り返す。繰り返しの回数（パスの数）は $\text{ceil}(\text{NCX}/p)$ となる。

ここで、

NCX : [X]の列数

p : メモリーに割り付け得る[X]の列数 = $\text{int}(\text{LW}/\text{NRB})$

LW : 作業用メモリーサイズ（単位：[X]の算術タイプ）

NRX : [X]の行数

である。

4. 従来の方法の問題点と新しい方法のねらい

従来の方法では、[X]の列全体（行1からNRXまで）をメモリーに割り付けるため、1回のパスで処理できる列の数は、 LW/NRB となる。

したがって、メモリーサイズが一定であれば、[X]の行数、列数が増すに比例して、[L]に対するパスの数が増える。

もともと[L]は相当に密なマトリックスであり、これに対するパスが100とか200にも及ぶと、I/O負荷およびI/Oに関わるCPU時間は、実用範囲を越えてしまう。

新しい方法では、[L]に対するパスの数を減らすために、ウエーブフロント法のテクニックが導入されている。[L]の最大アクティブカラム数をMaxACとする。

いま、[L]のピボットがk行k列にある時、当座の計算に必要な[X]の行は、k以下MaxAC行だけで、残りの行は当座の計算には無関係である。

したがって、新しい方法では[X]のMaxAC行分だけをメモリーに割り付け、残りはkの移動とともにファイルから読み出し、また計算の終わったものはその都度ファイルに書き出すという方針が取とられている。この方法によれば、1回のパスで処理できる[X]の列数は $LW/MaxAC$ となる。

一般にMaxACはNRXの1割程度であるので、新しい方法では[X]に対するI/Oが追加されるが、全体のパスの数は従来の方法の1/10程度になる。

5. 新しい方法の詳細

新しい方法では、先ず1回のパスで処理できる[B]の列数MAXCOL ($=LW/MaxAC$)が求められる。1回のパスは3つのフェーズで構成され、最初のフェーズでは、[B]のMAXCOL列が転置され、スクラッチファイルに書かれる。これは、[B]のMAXCOL列分の情報を、行単位に読み出す必要性からである。

第2のフェーズでは、実際の前進・後退代入演算が行われ、結果が行番号の逆順に、転置された状態で、スクラッチファイル上に得られる。

最後のフェーズでは、スクラッチファイル上の結果が再度転置されて、解ベクトルが出力ファイルに書かれる。

5.1 前段の転置

[B]のMAXCOL列が、GETSTRにより読み込まれ、それぞれの非ゼロ項は、2ワードのデスクリプターを付して、作業領域に格納される。デスクリプターには転置行列における行番号と列番号が記されている。これらの非ゼロ項は、SORT2Kによってソートされる。そして転置された結果は、BLDPKによりスクラッチファイルに書かれる。

MAXCOL列分の非ゼロ項が作業領域に収まらない場合には、作業領域に入るだけの非ゼロ項が格納され、ソートされて、第2のスクラッチファイルに一つの区画として書かれる。この時、1区画の最初のレコードアドレスを記録しておく。

[B]のMAXCOL列分の非ゼロ項全てを読み終えた時、第2のスクラッチファイルにはm個の区画が出来る。これらの区画は、FFCPのダイレクトアクセス機能 (FFCPDA) を用いて、あたかもm本のファイルをマージするかのように読み込まれ、[B]の転置行列が完成される。

5.2 前進・後退代入演算

このフェーズで使用する配列には、次のものがある。

PIVOT	: 現在のピボット行。[B] ^T あるいは[X] ^T の第k行を収容する。 (NCOLB*IPRECワード)
VEC1	: [L]の第k列の非ゼロ項 (MaxAC*IPRECワード)
IACT	: [L]の行それぞれについて、最初に現れる非ゼロ項の列番号 (NCOLLワード)
IAP1	: 現在の (行Kにおける) アクティブポジションベクトル (MaxACワード)
IAP2	: ‘一つ前’ のアクティブポジションベクトル 前進代入では行k-1の、後退代入では行k+1のアクティブ ポジションベクトルとなる。 (MaxACワード)
W	: 前進代入時には現在のピボット行からの寄与分を保存する ための、後退代入時には解ベクトルを保存するための、作 業領域 ((MaxAC-1)*NCOLB*IPRECワード)
NCOLB	- 今回のパスで処理する[B] ^T の列数 (通常はMAXCOL)
MaxAC	- [L]の各列の非ゼロ項の最大個数
IPREC	- 計算精度(1または2)
NCOLL	- [L]の列数

前進代入における計算ステップは次の通りである。

- (1) [L]の一行分の非ゼロ項を、配列VEC1に読み込む。同時にこれらの非ゼロ項の行番号をIAP1に記録する。
- (2) 第1列目であれば、IAP1の全要素を負にして、最初に現れたものであることを表示する。2列目以降であれば、IAP1とIAP2を比較し、新しい (IAP2に無く、IAP1に初めて現れた) 要素を負にする。同時に[L]の行毎に、最初に現れる非ゼロ項の列番号をIACTに記録する。

- (3) $[B]^T$ の一行分をPIVOTにアンパックする。
- (4) もし前段からの寄与（式(4)の右辺第2項）がWにあれば、それらをPIVOTに加える。
- (5) 現在のピボット行から後続の行への寄与分を計算し、作業領域Wに蓄積する。
- (6) ピボット行の各要素を[L]の対角項で除す。
- (7) 最終行を除いてピボット行をスクラッチファイルに書き出す（by BLDPK）。
- (8) 最終行でなければ、IAP1をIAP2にリネームして、ステップ(1)に戻る。

後退代入における計算ステップは次の通りである。

- (1) ステップ(6)へジャンプする。
- (2) 前進代入の結果のピボット行を、書いたときとは逆順に（後ろから）PIVOTに読み込む（by GETSTB）。
- (3) [L]の一行分の非ゼロ項を、配列VEC1に読み込む（by GETSTB）。同時にこれらの非ゼロ項の行番号をIAP1に記録する。この時IAP1とIAP2とを比較し、パツシブエレメントがあればゼロを挿入する。
- (4) IAP1内の行番号をIACTと突き合わせる。現在の[L]の列番号とIAP1内の各行が最初にアクティベートされた列番号とが一致すると、その行はもはや後続の計算には不要となるので、IAP1の要素を負にする。
- (5) Wに残してある結果行とVEC1とを用いて現在のピボット行を完成させる。
- (6) 完成した結果行をスクラッチファイルに書き出す。
- (7) 現在の結果行が後続の計算に必要なならば（それはIAP1の第1要素の符号で判別される）、結果行をWに保存する。また、Wの中の結果行の内、後続の計算に必要なくなったものは取り除く。
- (8) 最初の行でなければ、IAP1をIAP2にリネームして、ステップ(2)に戻る。

5.3 後段の転置

前進・後退代入演算を実施した結果、解は求められた。しかし、それらの解は行方向に、しかも行番号の逆順に記録されている。したがって、解ベクトルをファイルに出力する前に、解の転置と行の入れ替え（第1行が最終行に、第2行が最終の1つ前になるように、以下同様）を行う必要がある。

ここでの手続きは、メモリーに入るだけのアンパックされた結果行を1区画とし、区画単位に転置と行の入れ替えを行う。結果は第3のスクラッチファイルに記録する。この時、各区画の最初のレコードアドレスを記録しておく。

全ての結果行を処理し終えた後には、第3のスクラッチファイルにはm個の区画が出来上がる。このm個の区画を、FFCPのダイレクトアクセス機能（FFCPDA）を用いて、あたかもm本のファイルをつなぎ合わせる如くに読み出し、解ベクトルを完成させる。

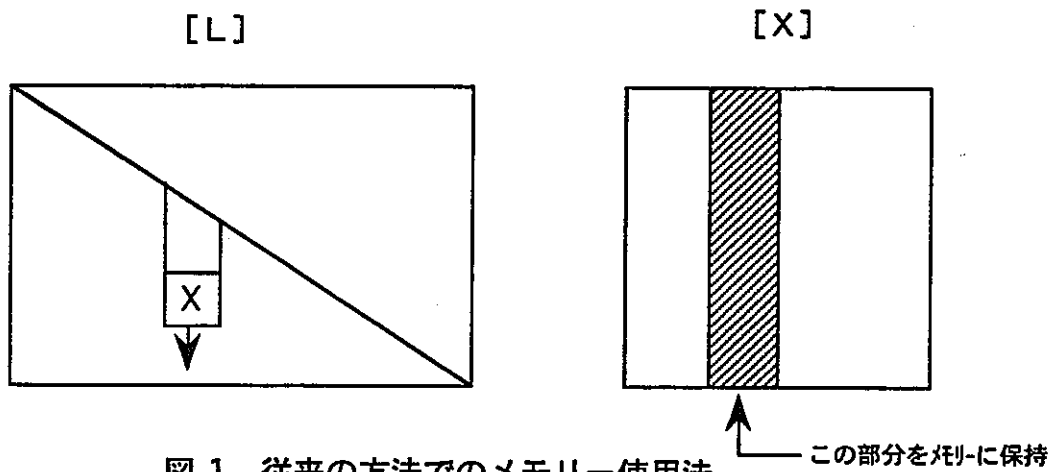


図.1 従来の方法でのメモリー使用法

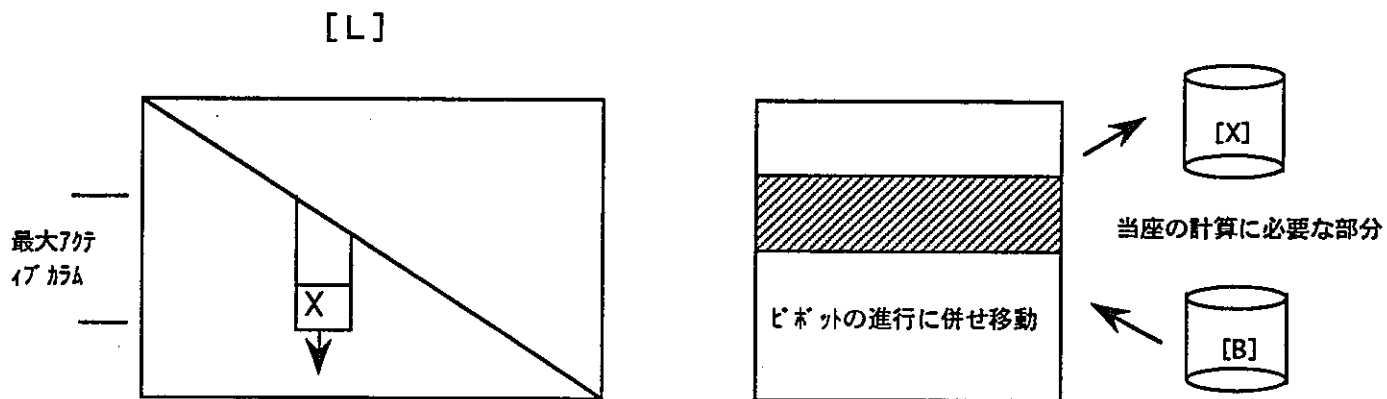


図.2 新しい方法でのメモリー使用方法

6. 性能評価

新しく開発したFBSQ(Forward-Backward Substitution, Quick method) およびMPYA3T (Multiply&Add, Method3, Transposed case)の性能を評価するために、流体構造連成問題と静的縮合問題を例に採り、CPU時間の計測を行った。

・ 流体構造連成問題

海遊館大型水槽モデル

2690 節点

4272 要素

(224*SQFAC4)

(1016*IQFAC4)

(1016*QFLA4S)

(2016*LHEX8)

7710 未知自由度

・ 静的縮合問題

64x64メッシュの矩形板の振動問題 (VP723類似)

このうち4点置きに15x15、都合225点に静的縮合を掛ける。

4225 節点

4096 要素

(4096*QFLA4R)

24834 未知自由度

6.1 演算内容とマトリックスサイズ

・流体構造連成問題

- ① マトリックス分割 (PARTNS) $[M_{ff}] \Rightarrow \begin{array}{|c|c|} \hline M_{pp} & M_{px} \\ \hline \text{---} & \text{---} \\ \hline & M_{xx} \\ \hline \end{array}$
- ② マトリックス分割 (PARTNS) $[K_{ff}] \Rightarrow \begin{array}{|c|c|} \hline K_{pp} & K_{px} \\ \hline \text{---} & \text{---} \\ \hline K_{xp} & K_{xx} \\ \hline \end{array}$
- ③ 三角分解 (SDCOMP) $[K_{pp}] \Rightarrow [L_{pp}] [L_{pp}] [L_{pp}]^T$
- ④ 前進後退代入 (FBSQ) $[K_{pp}]^{-1} [K_{px}] \Rightarrow [\bar{K}_{px}]$
- ⑤ 乗和算 (MPYA3T) $[K_{px}]^T [K_{px}] + [M_{xx}] \Rightarrow [\bar{M}_{xx}]$

$[M_{ff}]$: 7710 行	7710 列、	(null)
$[K_{ff}]$: 7710 行	7710 列、	密度 0.46%
$[K_{pp}]$: 2689 行	2689 列、	密度 ? %
$[K_{px}]$: 2689 行	5021 列、	密度 ? %
$[K_{xx}]$: 5021 行	5021 列、	密度 0.68%
$[\bar{K}_{px}]$: 2689 行	5021 列、	密度 52.68%
$[\bar{M}_{xx}]$: 5021 行	5021 列、	密度 27.75%

・ 静的縮合問題

- ① マトリックス分割 (PARTNS)
$$[K_{xx}] \Rightarrow \begin{array}{c} | K_{oo} | K_{oa} | \\ | \text{-----} + \text{-----} | \\ | K_{ao} | K_{aa} | \end{array}$$
- ② 三角分解 (SDCOMP)
$$[K_{oo}] \Rightarrow [L_{oo}] [D] [L_{oo}]^T$$
- ③ 前進後退代入 (FBSQ)
$$[K_{oo}]^{-1} [K_{oa}] \Rightarrow [G_{oa}]$$

$$[K_{aa}] + [K_{ao}] [G_{oa}] \Rightarrow [\bar{K}_{aa}]$$
- ④ マトリックス分割 (PARTNS)
$$[M_{xx}] \Rightarrow \begin{array}{c} | M_{oo} | M_{oa} | \\ | \text{-----} + \text{-----} | \\ | M_{ao} | M_{aa} | \end{array}$$
- ⑤ 乗和算 (MPYA2)
$$[M_{oo}] [G_{oa}] + [M_{oa}] \Rightarrow [scr1]$$
- ⑥ 乗和算 (MPYA2)
$$[M_{ao}] [G_{oa}] + [M_{aa}] \Rightarrow [scr2]$$
- ⑦ 乗和算 (MPYA3T)
$$[G_{oa}]^T [scr1] + [scr2] \Rightarrow [\bar{M}_{aa}]$$

$[K_{xx}]$: 24834 行	24834 列、	密度 0.10%
$[G_{oa}]$: 23484 行	1350 列、	密度 71.98%
$[\bar{K}_{aa}]$: 1350 行	1350 列、	密度 72.22%
$[M_{xx}]$: 24834 行	24834 列、	密度 0.02%
$[scr1]$: 23834 行	1350 列、	密度 69.13%
$[scr2]$: 1350 行	1350 列、	密度 69.44%
$[\bar{M}_{aa}]$: 1350 行	1350 列、	密度 69.44%

6.2 CPU時間計測結果

流体構造連成問題と静的縮合問題に対し、それぞれの演算過程におけるCPU時間を表.1、2に示す。

表.1 流体構造連成問題でのCPU時間

演算内容	CPU時間 (単位：秒)	
	変更前	変更後
①	0.160	0.140
②	1.220	1.120
③	12.570	13.880
④	952.570	381.230
⑤	34.020	74.990
合計	1000.540	471.360

(使用機種：RS6000-580)

表.2 静的縮合問題でのCPU時間

演算内容	CPU時間 (単位：秒)	
	変更前	変更後
①	2.440	2.230
②	280.190	278.160
③	1985.680 42.250	4309.180 38.420
④	1.490	1.330
⑤	155.960	151.320
⑥	32.990	30.530
⑦	9301.750	9037.699
合計	11802.750	13848.869

(使用機種：RS6000-580)

6.3 考察

新しく開発したFBSQ(Forward-Backward Substitution, Quick method)は、流体構造連成問題においては効果を見せているが、静的縮合問題では逆効果になることが判った。

MPYA3Tについては、これらの問題に対して効果が見られなかった。しかし、表.3に示すように、マトリックスのパターンによっては、表.4のように、MPYA3Tが効果をあげる場合もある。今後、MPYA3Tの効果的な使用場所を探す必要がある。

なお、静的縮合問題の演算⑦でのマトリックスは [A]、[B] とともにほぼフルマトリックスになっているために、マトリックスのスパース性を利用した計算効率の改善は期待できない。

表.3 計測に用いたマトリックスパターン

パターンNo.		[A] ^T	[B]	[C]
①	行数	3000	3000	3000
	列数	3000	3000	3000
	密度(%)	5	5	10
②	行数	1000	3000	1000
	列数	3000	500	500
	密度(%)	3	10	10
③	行数	1000	3000	1000
	列数	3000	500	500
	密度(%)	30	10	10
④	行数	3000	1000	3000
	列数	1000	500	500
	密度(%)	30	10	10

表.4 マトリックスパターン別乗和算手法の比較

	パターンNo.	計算手法		
		MPYA1T	MPYA2T	MPYA3T
CPU時間 (単位：秒)	①	1981.44	874.60	602.60
	②	81.09	31.93	43.84
	③	588.59	236.82	121.53
	④	546.11	237.40	123.31
入出力回数	①	52545	24274	59662
	②	2405	2056	4026
	③	7699	6710	7225
	④	9673	11227	12760

C descriptor which contains the packed column and row position
 C in the transposed matrix. The nonzero terms are sorted
 C (by SORTIC) and the transposed result is written (by BLDPK)
 C on a scratch file. If the working space will not contain
 C all nonzero terms in MAXCOL columns, as many terms as can be
 C held are stored, sorted and written on an intermediate
 C scratch file and the position (FFCP address) of the first
 C column is noted. When all terms have been written, the
 C intermediate scratch file is read (by FFCPPDA) and the
 C transposed matrix is written as a result of an n-way merge
 C from each of the sorted partitions. Null columns of B are
 C squeezed out of B_T. Each null column is noted in a bit
 C array which is reserved at the end of working space.

3. Forward/Backward Substitution

Allocation of the working space is as follows:

JPIVOT	Current pivot row	NCOLB*PREC
JVEC1	Nonzero terms in current column of L	PMAX*PREC
IACT	Row numbers for each column of L at which the column is activated.	NCOLL
IAP1	Current active position vector	PMAX
IAP2	Previous (forward pass) or next (back pass) active position vector	PMAX
JW	Working storage to hold contributions (forward pass) or actual solution vectors (back pass)	(PMAX-1) *NCOLB *PREC
	:	
	:	
IBUF3	3 FFCP buffers	3*LBUFSZ

- C NCOLB - Number of columns in B_T (usually MAXCOL).
 C PMAX - Maximum number of nonzero terms in a column of L
 C (as defined in the 5-th word of the header).
 C PREC - Precision (1 or 2).
 C NCOLL - Number of columns in L.
 C LBUFSZ - Number of words in a FFCP buffer.

The steps in the forward pass are as follows:

1. The nonzero terms in the current column of L are read and stored in the area pointed to by JVEC1. At the same time, the row positions of these terms are stored in the area pointed to by IAP1.
2. If first column, all terms in the active position vector are set negative to indicate first appearance. Otherwise, the current active position vector is compared to the previous active position vector and terms which are new (did not appear in the previous active position vector) are set negative. At the same time, the row positions for each column of L at which the column is activated are stored in the area pointed to by IACT.
3. The current row of B_T is unpacked into the area pointed to by JPIVOT.
4. If contributions exist in working storage, they are

- C added to the pivot.
- C 5. Contributions from current pivot row to subsequent rows
- C are accumulated in working storage (subr. FBSQF1).
- C 6. Each term in the pivot row is divided by the diagonal
- C term in the current column of L.
- C 7. For each row except the last, the pivot row is written
- C on a scratch file (by BLDPK).
- C 8. If not the last row, pointers are adjusted for the next
- C row and the process returns to step 1.

C The steps in the backward pass are as follows:

- C 1. Control passes to step 6.
- C 2. The current row of the results of forward pass are read
- C into the pivot row (by GETSTB).
- C 3. The current column of L is read into the area pointed to
- C by JVEC1. At the same time its positions are stored in
- C the area pointed to by IAP1. Passive terms are also
- C restored by comparing current vector (IAP1) and previous
- C (in the backward sense) vector (IAP2).
- C 4. The terms in the current active positions are tested
- C against the positions where it was activated (IACT), and
- C terms in the current vector (IAP1) are set negative if
- C they no longer be needed for subsequent computations.
- C 5. The current solution row is formed (subroutine FBSQB1).
- C 6. The solution row is written on a scratch file.
- C 7. If the solution row is needed for subsequent computations
- C (as determined by the sign of the first term in the
- C active position vector), the row is saved in the working
- C storage.
- C 8. If not the first row, pointers are adjusted for the next
- C row and the process returns to stpe 2.

C 4. Final Transpose

C As a result of the forward/backward substitution, the
 C solution is now complete. However, the terms are
 C "scrambled", i.e., the solution is written by rows and the
 C rows are in reverse order. It is required to transpose
 C the solution and then, prior to writing each column on the
 C solution file, interchange the terms in each column (so that
 C first is last, second is next to last, etc.).

C The method used is that which is used in method 3T of MPYAD
 C (see subroutine MPYA3T).

C
 C

D. 有限要素の修正

1. 修正の概要

FINAS では、現在4節点四辺形のシェル要素として、4節点四辺形シェル選択的低積分要素 QFLA4R、4節点四辺形シェル要素QFLA4S、4節点四辺形シェル低積分要素QFLA4RTが組み込まれている。

QFLA4Rは要素内での縦せん断ひずみ分布を一定とした低積分要素であり、面外せん断ひずみを要素中央の1点により評価をしているため、ねじり変形に関してアワーグラスモードの変形を生じる場合がある。

これを防ぐため QFLA4S では、面外せん断ひずみ成分について、ひずみを方向別に線形分布を仮定する方法で定式が行われている。

また、QFLA4RTにおいても、変位-ひずみ関係式のTaylor級数展開および QFLA4S と同様に面外せん断ひずみ分布仮定をベースに定式が行われている。

これにより、アワーグラスモードの発生を防ぐことができたが、QFLA4S、QFLA4RTに関して、要素形状が長方形でない場合には、曲げ剛性が過大に評価されることがわかった。

したがって、今回は、QFLA4SおよびQFLA4RT に対し、要素形状が長方形でない場合でも曲げ剛性が過大に評価されないように定式を見直し、修正を行った。

2. 現状の定式

現状の QFLA4Sでは図1のように面外せん断ひずみを要素座標系（物理座標系： $x - y$ 座標系）について、 γ_{yz} を y 方向に線形分布、 γ_{xz} を x 方向に線形分布で仮定している。

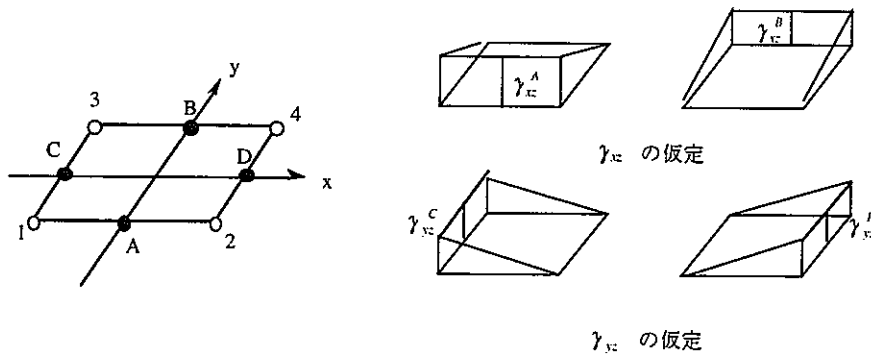


図1 現状の面外せん断ひずみ分布仮定

しかし、この方向別の線形分布仮定は正規化された局所座標系（ $\xi - \eta$ 座標系）でのひずみ成分 $\gamma_{\xi z}$ 、 $\gamma_{\eta z}$ について行うべきである（図2参照）。

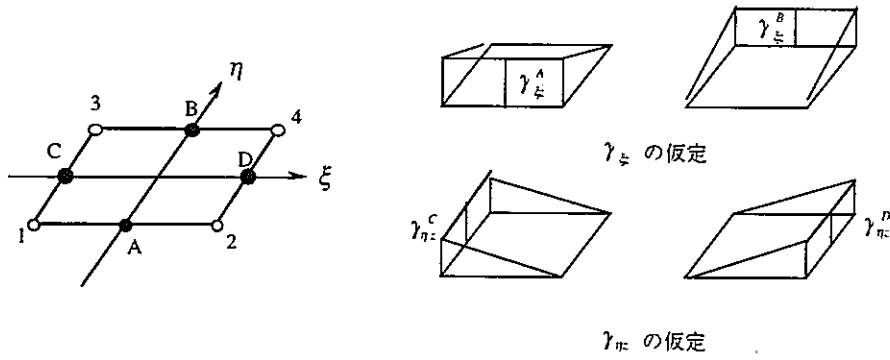


図2 正規化座標系での面外せん断ひずみ分布仮定

現状のQFLA4Sでは、一般的な形状で $\xi - \eta$ 軸が物理座標系で直交しなくなる場合、 γ_{ξ} 、 γ_{η} は、 γ_{ξ} 、 γ_{η} の両方の成分が含まれるため、ロッキングの原因となり、曲げ剛性が過大に評価されることになる。

3. 定式の改良

面外せん断ひずみの評価を以下のように行うように改良した。

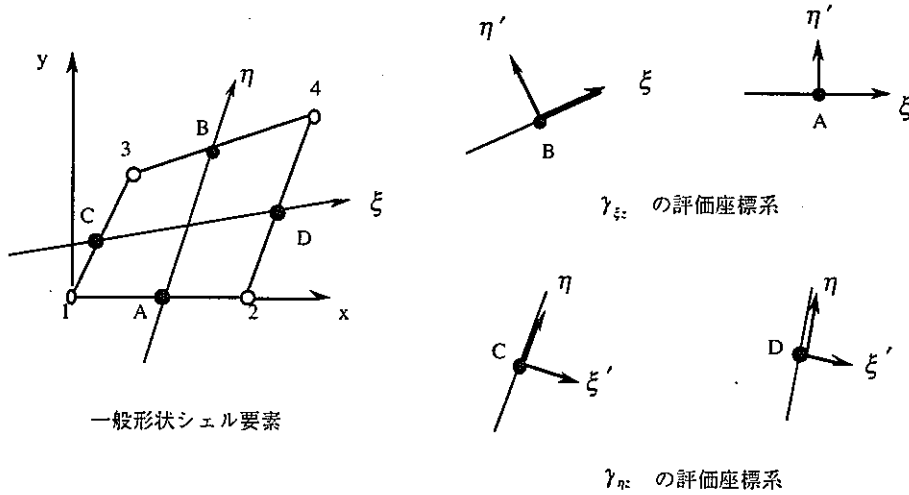


図3 一般形状要素の面外せん断ひずみ仮定

図3において、点A,Bについて図中の $\xi - \eta' - z$ 座標系で γ_{ξ} 、点C,Dについて図中の $\xi' - \eta - z$ 座標系で γ_{η} を評価する。これらの点のひずみは仮定した変位関数から通常の方法で求められる。要素内の各位置における γ_{ξ} 、 γ_{η} は次のように仮定する。

$$\gamma_{\xi} = \frac{1}{2}(1 - \eta)\gamma_{\xi}^A + \frac{1}{2}(1 + \eta)\gamma_{\xi}^B \tag{1}$$

$$\gamma_{\eta} = \frac{1}{2}(1 - \xi)\gamma_{\eta}^B + \frac{1}{2}(1 + \xi)\gamma_{\eta}^D \quad (2)$$

積分点における $x - y$ 軸と $\xi - \eta$ 軸の関係が図4のようであるとすると、物理座標系での面外せん断ひずみ座標系と物理座標でのひずみ γ_{ξ} 、 γ_{η} の関係は次のようになる。

$$\gamma_{\xi} = \gamma_{\eta} \cos \alpha + \gamma_{\eta} \sin \alpha \quad (3)$$

$$\gamma_{\eta} = \gamma_{\xi} \cos \beta + \gamma_{\eta} \sin \beta \quad (4)$$

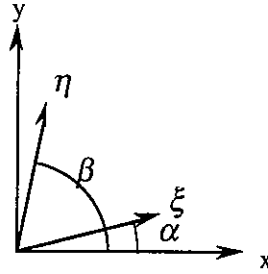


図4 積分点での物理座標系と正規化座標系の関係

これから、 γ_{ξ} 、 γ_{η} の関係は次のように評価される。

$$\gamma_{\xi} = \frac{1}{\cos \alpha \sin \beta - \sin \alpha \cos \beta} (\gamma_{\xi} \sin \beta - \gamma_{\eta} \sin \beta) \quad (5)$$

$$\gamma_{\eta} = \frac{1}{\cos \alpha \sin \beta - \sin \alpha \cos \beta} (-\gamma_{\xi} \cos \beta + \gamma_{\eta} \cos \alpha) \quad (6)$$

参考文献

Klaus-Jurgen Bathe, " Finite Element Procedure", 1996, PRENCE HALL.

4. 検証計算

3. で示したように面外せん断ひずみを計算するよう改良し、次ページのような片持ちはりモデルにより検証計算を行った。

解析モデルは断面 0.2×0.1 の長方形で長さ 6 の片持ちはりで、自由端に引張、面内曲げ、面外曲げ、ねじりの単位荷重を加えるものである。

検証問題 1 は要素を長さ方向 6 個の長方形要素で分割したモデルであり、検証問題 2 は要素形状が台形、平行四辺形となるよう分割したモデルである。

材料定数、各荷重条件の載荷点の変位の理論解は次ページの示している。表に示す値は FINAS による解析結果を理論解で割った値であり、1 に近い程理論解に近いことを示している。

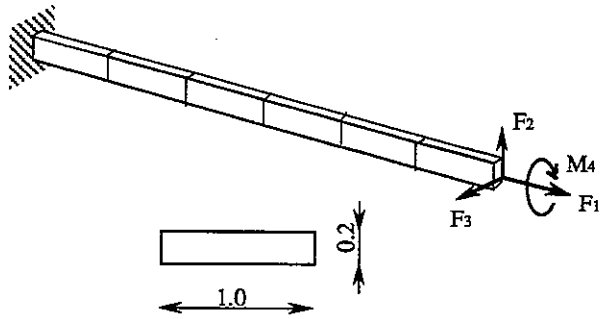
要素名に（修正）を付けたものが、3. の改良を行ったものであり、付いていないものが現状の FINAS の要素である。

検証問題 1 では、QFLA4R のねじりが大きくなる以外は良好な解となっている。

検証問題 2 では、面外曲げについて、現状の QFLA4S, QFLA4RT の変形が小さくなっており、剛性が過大に評価されていることを示している。これに対し、今回の改良を加えたものは良好な解となっている。

QFLA4S の台形要素モデルのねじりに関して若干精度が落ちているが、重大な問題ではないと思われる。

検証問題 1. 長方形要素 (FINAS)



長さ 6.0 × 幅 0.2 × 厚さ 0.1

$E = 1.E7$ $\nu = 0.3$

メッシュ 6 × 1

拘束条件：片持ち

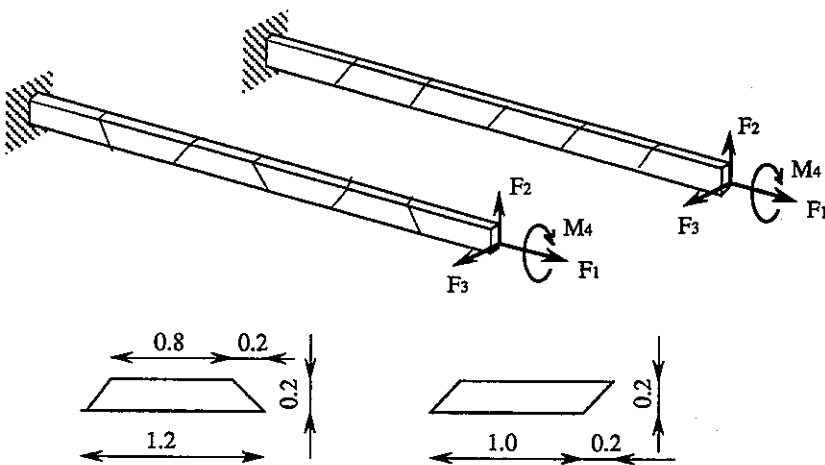
荷重条件： 理論最大変位：

- 1) 引張 3.0 E-5
- 2) 面内曲げ 0.1081
- 3) 面外曲げ 0.4321
- 4) 捩り 0.03208 [rad]

計算結果 (計算値/理論値)

4 節点 シェル要素	長方形要素							
	引張	面内曲げ	面外曲げ	捩り				
QFLA4R	0.995	0.904	0.980	377.400				
QFLA4RT	0.995	0.904	0.980	0.940				
QFLA4RT (修正)	0.995	0.904	0.980	0.940				
QFLA4S	0.995	0.904	0.980	0.940				
QFLA4S (修正)	0.995	0.904	0.980	0.940				

検証問題 2. 歪んだ要素形状 (FINAS)



長さ 6.0 × 幅 0.2 × 厚さ 0.1

$E = 1.E7$ $\nu = 0.3$

メッシュ 6 × 1

拘束条件：片持ち

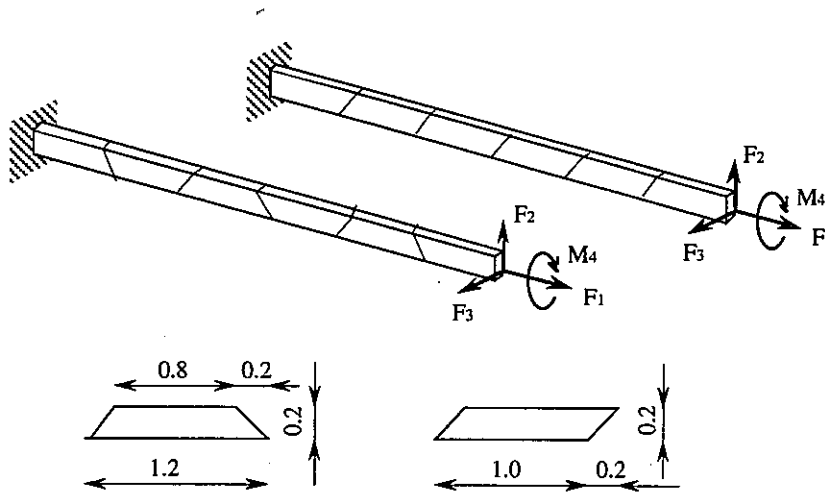
荷重条件： 理論最大変位：

- 1) 引張 3.0 E-5
- 2) 面内曲げ 0.1081
- 3) 面外曲げ 0.4321
- 4) 捩り 0.03208 [rad]

計算結果 (計算値/理論値)

4 節点 シェル要素	台形要素				平行四辺形要素			
	引張	面内曲げ	面外曲げ	捩り	引張	面内曲げ	面外曲げ	捩り
QFLA4R	1.000	0.057	0.996	109.800	1.000	0.080	0.992	213.700
QFLA4RT	0.996	0.071	0.054	0.941	0.996	0.081	0.067	0.940
QFLA4RT (修正)	0.996	0.071	0.965	0.942	0.996	0.081	0.978	0.941
QFLA4S	1.000	0.057	0.042	0.941	1.000	0.080	0.060	0.816
QFLA4S (修正)	0.996	0.071	0.964	0.883	0.996	0.081	0.978	0.848

検証問題 2. 歪んだ要素形状 (FINAS)



長さ 6.0 × 幅 0.2 × 厚さ 0.1

$E = 1.E7$ $\nu = 0.3$

メッシュ 6 × 1

拘束条件：片持ち

荷重条件：理論最大変位：

- 1) 引張 3.0 E-5
- 2) 面内曲げ 0.1081
- 3) 面外曲げ 0.4321
- 4) 振り 0.03208 [rad]

計算結果 (計算値/理論値)

4 節点 シェル要素	台形要素				平行四辺形要素			
	引張	面内曲げ	面外曲げ	振り	引張	面内曲げ	面外曲げ	振り
QFLA4R	1.000	0.057	0.996	109.800	1.000	0.080	0.992	213.700
QFLA4RT	0.996	0.071	0.054	0.941	0.996	0.081	0.067	0.940
QFLA4RT (修正)	0.996	0.071	0.965	0.942	0.996	0.081	0.978	0.941
QFLA4S	1.000	0.057	0.042	0.941	1.000	0.080	0.060	0.816
QFLA4S (修正)	0.996	0.071	0.964	0.883	0.996	0.081	0.978	0.848

8 節点 シェル要素	台形要素				平行四辺形要素			
	引張	面内曲げ	面外曲げ	振り	引張	面内曲げ	面外曲げ	振り

8 節点 ソリッド要素	台形要素				平行四辺形要素			
	引張	面内曲げ	面外曲げ	振り	引張	面内曲げ	面外曲げ	振り
HEXA (6分割)	0.902	0.032	0.024	0.880	0.980	0.064	0.048	0.909
HEXA (32分割)	0.966	0.017	0.045	0.870	0.995	0.293	0.399	0.928

20 節点 ソリッド要素	台形要素				平行四辺形要素			
	引張	面内曲げ	面外曲げ	振り	引張	面内曲げ	面外曲げ	振り
HEXA (6分割)	0.995	0.886	0.920	0.906	0.995	0.967	0.941	0.906
HEXA (32分割)								

E. F I N A S (V13.0)の整備と公開

1. F I N A S (V13.0)の整備と公開

汎用非線形構造解析システム F I N A S (Finite Nonlinear Structural Analysis System)は、現在バージョン12.0が公開され、動燃事業団はもとより多数のユーザーに幅広い分野で利用されている。

F I N A Sが他の汎用構造解析コードと競合ができ、更に多くのユーザーを獲得し、利用されていくためには、既存ユーザーの要望への迅速な対応、適用分野の拡充、新しい理論、手法の導入等を行っていかねばならない。

本年度は、F I N A Sがバージョン12.0から13.0に拡張され、使用説明書も整備された。ここでは、バージョン13.0の改良、拡張項目について概要を述べる。

なお、改良・拡張項目の詳細については「汎用非線形構造解析システム F I N A S Version 13.0における機能拡張報告書」（平成7年8月）で述べられている。

2.改良・拡張項目

2.1静的応力解析機能

・有限要素の追加

ソリッド要素の面または2次元要素の辺に弾性支床境界を定義する3種類の要素が新たに追加された。また、平面ひずみ要素として高次要素のほかに、低次の4節点四辺形要素も加えられた。今回追加された応力解析用の要素以下の4種類である。

3節点三角形弾性支床要素	TSPRG3
4節点四辺形弾性支床要素	QSPRG4
3次元弾性支床梁要素	LSPRG2
4節点平面ひずみ要素	QPLAN4

・BEAM2要素にオフセット機能を追加

3次元梁要素BEAM2にオフセット機能が追加され、偏心梁の挙動を扱うことが可能となった。

・遠心力荷重の定義

ROD2,QAX4,QAX8,QFLA4R,QFLA4S,QFLA4RT,HEX8,HEX20の各要素において、遠心力荷重の定義が可能となった。

- ・面接触解析機能の効率化

面接触解析で大きな計算時間を消費する接触力マトリクスの算出過程が改良され、この過程での計算時間は平均1/2～1/3になり、面接触解析が高速化された。

2.2熱伝導解析機能

- ・熱流動解析機能

流体の熱移送を考慮した熱伝導解析を行うために、非対称ソルバーの開発と熱流路要素が組み込みが行われた。追加されたのは以下の熱伝導解析用要素である。

2節点熱流路要素

HFLOW2

- ・前処理プログラムFNSRADの開発

輻射境界を有する軸対称構造物の定常または非定常熱伝導解析を行うために、輻射形態係数の計算とFINASの入力データをジェネレートする前処理プログラムが開発された。

2.3動的解析機能

- ・スペクトル応答解析のモード重ね合わせ方法の追加

スペクトル応答解析におけるモード重ね合わせは、従来、SRSS法(Square Root of Sum of Square)を用いているが、新たにCQC法(Complete Quadratic Combination)とABS法(Abtract)が追加された。

- ・スペクトル応答解析に対数補間機能の追加

スペクトル応答解析では、解析に必要な応答スペクトルを各周期に対する値として表形式で入力したデータから線形補間する方法が用いられているが、さらに対数補間する方法も組込まれた。

- スペクトル応答解析に多入力床応答スペクトル解析機能の追加
従来のFINASでは単一入力の床応答スペクトル解析機能のみが用意されているが、新たに多入力の床応答スペクトルを取扱うことができる解析機能が追加された。
- 全固有値計算機能
固有値計算では、従来、最低次からn個およびある範囲内の全ての固有値が求められているが、さらにある固有値の近傍でのn個および全固有値が求められようになった。

2.4入出力機能

- XYプロットにスケーリング機能の追加
XYプロット図において出力されるX軸とY軸の値に各々倍率を与えることが可能となった。
- ベクトルプロット図に出力項目の追加
ベクトルプロット図において、従来の主応力、最大せん断応力に加え、応力直成分およびせん断成分の出力が可能となった。
- シェル要素に部材力の出力機能の追加
部材力のプリント、ポストテープ出力機能は、応力を出力しない2節点配管要素(PIPE2)および3次元弾性梁要素(SBEAM2)に適用されているが、新たにシェル要素(TFLA3,QFLA4R,QFLA4RT,QFLA4S)においても部材力の出力が可能となった。ただし、この機能は線形弾性解析の範囲に限定される。
- スペクトル応答解析に最大応答反力の出力機能の追加
スペクトル応答解析において最大応答反力のプリントおよびポストテープ出力が可能となった。

2.5その他

- ・リナンバー機能にSloanの方法の追加

F I N A Sでは連立一次方程式の解法としてウェーブフロント法が採用され、その効率化のためにリナンバーが行われている。このリナンバーアルゴリズムは従来Hoitの方法が用いられているが、さらにSloanの方法が追加された。

- ・解析規模の制限を緩和

従来のF I N A Sは1ワード32ビット型の計算機を用いた場合、自由度数の上限は $2^{15}=32768$ に制限されていた。F I N A Sの内部データテーブルを拡張することにより、自由度数の上限が 2^{22} =(約400万)になった。ただし、入力データとして節点番号、要素番号の入力フィールドが5桁しかないために、実際には節点数、要素数が99999までという制限となる。また、現状ではF I N A Sの構造図プロット機能を用いる場合には節点数が65536以下でなければならない。

- ・ユーザーが定義するサブプログラムの追加

サブプログラムにより任意の構成モデルまたは任意の有限要素を定義することが可能となった。

結 言

本報告書は平成7年10月から平成8年3月までに実施された汎用非線形構造システムF I N A Sの拡張、整備等の内容についてまとめたものである。今回の機能拡張は、現バージョンにおいて改良されるべき機能、動燃事業団からの要望等を中心に行われた。また、公開されたバージョンV13.0の内容についても報告が記されている。

F I N A Sは今後とも多くのユーザーに利用されて行くためには、ユーザーのニーズに応えるばかりではなく、適用分野の拡大および新しい解析手法、技術の調査と導入、各種計算機環境への対応、汎用プリ/ポストプロセッサ-との連結等を積極的に行っていく必要がある。今回の改良、拡張にあたり、動燃事業団の技術協力部/技術協力室/室長川原敏夫氏および菊地隆雄氏、大洗工学センター基盤技術開発部/部長代理 岩田耕司氏、構造・材料技術開発室の月森和之氏にはいろいろとご助言、ご指導をいただき、ここに謝辞を申し上げます。