

Finite Element Program EPITOK—II

Tetsuhiko MIYOSHI*, Ken KAMINISHI** and Shun-ichi KAWANO**

(Received July 15, 1986)

Abstract

A finite element program for elastic-plastic stress analysis based on a new approximation of the constitutive equations is presented. A brief introduction to the theory of this new procedure and a detailed explanation of the subroutines that are essential for this method are also described.

Introduction

The purpose of this paper is to explain the finite element program EPITOK—II developed at Yamaguchi University for elastic-plastic stress analysis. This program is based on a new method of approximation for plasticity problem and characterized by the following.

- (1) A new nonlinear scheme is used as the discrete constitutive equations instead of the conventional finite difference approximation. This scheme satisfies a consistency relation which does not hold in usual one step finite difference approximation of the constitutive equations.
- (2) An iterative method is applied at each loading step using LU decomposition of the total stiffness matrix.
- (3) An effective acceleration technique is employed.

Our main intention in publishing this program is to explain the essential features of the new procedure. Therefore the program presented in this paper can apply only to plane stress problems with isotropic hardening rule. However, the basic idea of the procedure is simple and general. Therefore it is applicable to all problems which can be treated by the usual iterative methods.

In §1 we summarize the theory of the method. §2 and §3 are devoted to the explanation of the main program and subroutines. In §4 the program itself is presented and a simple example of input data is shown for practical use of the program.

Notation.

u_i : displacements. We assume that the displacements of a certain portion of the boundary $\partial\Omega$ are prescribed so that a unique solution of the problem exists.

$$\begin{aligned}\sigma &= (\sigma_{11}, \sigma_{22}, \sigma_{12}) : \text{stresses} & (\sigma_{21} = \sigma_{12}) \\ \epsilon &= (\epsilon_{11}, \epsilon_{22}, \epsilon_{12}) : \text{strains}\end{aligned}$$

*Department of Applied Science

**Department of Mechanical Engineering

$$D = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{pmatrix} \quad \begin{array}{l} \nu : \text{Poisson's ratio} \\ E : \text{Young's modulus} \end{array}$$

$f^2(\sigma) = \sigma_{11}^2 + \sigma_{22}^2 - \sigma_{11}\sigma_{22} + 3\sigma_{12}^2$: yield function. The initial yielding is given by $f(\sigma) = z_0$.

$$u_{,i} = \partial u / \partial x_i$$

$$\partial f = \{\partial f / \partial \sigma\}$$

∂f^* : transposed ∂f

$$D' = D \partial f \partial f^* D / (H' + \partial f^* D \partial f)$$

$\dot{\sigma}$: time derivative of σ

$\bar{\epsilon}^p$: effective plastic strain

$H(\bar{\epsilon}^p)$: function to determine the yielding

$$S = \begin{pmatrix} 1 & -0.5 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix} \quad \text{: this matrix is used to express } \partial f \text{ as } \partial f = S \sigma / f.$$

B: The so called "B" matrix defined as follows. Let N be the piecewise linear function taking value 1 at node n and value 0 at other nodes. Let i, j, k be the nodes of an element. Then B is a 3×6 matrix of the form

$$B^{(n)} = \begin{pmatrix} N_{,1}^{(n)} & 0 \\ 0 & N_{,2}^{(n)} \\ N_{,2}^{(n)} & N_{,1}^{(n)} \end{pmatrix} \quad n = i, j, k$$

1. The theory of EPITOK—II

Let us consider a plane stress problem formulated as follows:

$$(1.1) \quad \sum_j \sigma_{ij,j} + F_i = 0 \quad \text{in } \Omega \quad (i = 1, 2)$$

$$(1.2) \quad \begin{cases} \dot{\sigma} = \dot{D} \dot{\epsilon} & \text{if } f(\sigma) < H(\bar{\epsilon}^p) \\ \dot{\sigma} = (D - D') \dot{\epsilon} & \text{if } f(\sigma) = H(\bar{\epsilon}^p) \text{ and } \partial f^* \dot{\sigma} > 0 \end{cases}$$

$$(1.3) \quad \epsilon_{11} = u_{1,1} \quad \epsilon_{22} = u_{2,2} \quad \epsilon_{12} = u_{1,2} + u_{2,1}$$

In EPITOK, unlike the conventional methods, the constitutive equations (1.2) are not approximated by the finite difference to increase the "accuracy" of the approximate solution. We introduce a new hardening parameter $\hat{\epsilon}^p$ and a new function $G(\hat{\epsilon}^p)$ by

$$(1.4) \quad G(\hat{\epsilon}^p) = H(\bar{\epsilon}^p) \quad \hat{\epsilon}^p = \int_0^{\bar{\epsilon}^p} \sqrt{H'(s)} ds$$

Then (1.2) can be written equivalently as

$$(1.5) \quad \begin{cases} \dot{\epsilon} - C \dot{\sigma} = 0, \quad \dot{\epsilon}^p = 0 & \text{if } f(\sigma) < G(\hat{\epsilon}^p) \\ \dot{\epsilon} - C \dot{\sigma} = \frac{1}{\{G'(\hat{\epsilon}^p)\}^2} \partial f \partial f^* \dot{\sigma}, \quad \dot{\epsilon}^p = \frac{\partial f^* \dot{\sigma}}{G'(\hat{\epsilon}^p)} & \text{if } f(\sigma) = G(\hat{\epsilon}^p) \\ & \text{and } \partial f^* \dot{\sigma} > 0. \end{cases}$$

The plastic strain increments $\dot{\epsilon} - C \dot{\sigma}$ and the hardening parameter $\dot{\epsilon}^p$ then satisfies, respectively,

$$(1.6) \quad \dot{\epsilon} - C \dot{\sigma} = \frac{\partial f^* \dot{\sigma}}{(G')^2} \partial f, \quad \dot{\epsilon}^p = \frac{\partial f^* \dot{\sigma}}{(G')^2} G'$$

which represent a normality condition in a wider sense (see [1] for details). Now we introduce a parameter λ defined by

$$\lambda = \frac{\partial f^* \dot{\sigma}}{(G')^2 f(\sigma)}$$

Then the equations in (1.6) are written as

$$(1.7) \quad \dot{\epsilon} - C \dot{\sigma} = \lambda S \sigma, \quad \dot{\epsilon}^p = \lambda f(\sigma) G'(\hat{\epsilon}^p).$$

Hence we approximate these equations by

$$(1.8) \quad \Delta \epsilon^n - C \Delta \sigma^n = \lambda^{n+1} S \sigma^{n+1}, \quad \Delta \hat{\epsilon}^{p,n} = \lambda^{n+1} f(\sigma^{n+1}) G'(\hat{\epsilon}^{p,n+1})$$

at n -th step of the integration, where $\Delta \epsilon^n = \epsilon^{n+1} - \epsilon^n$. The parameter λ^{n+1} is determined by the condition $f(\sigma^{n+1}) = G(\hat{\epsilon}^{p,n+1})$.

In EPITOK the increments of displacements, strains, stresses and $\Delta \hat{\epsilon}^{p,n}$ at n -th step of integration are determined by the following system of equations.

$$(1.9) \quad \Sigma B^* \Delta \sigma^n \cdot \text{area}(\text{element}) = \Delta F$$

$$(1.10) \quad \Delta \epsilon^n - C \Delta \sigma^n = 0, \quad \Delta \hat{\epsilon}^{p,n} = 0 \quad \text{if } f(D \Delta \epsilon^n + \sigma^n) < G(\hat{\epsilon}^{p,n})$$

$$(1.11) \quad \begin{cases} \Delta \epsilon^n - C \Delta \sigma^n = \lambda^{n+1} S \sigma^{n+1} \\ \Delta \hat{\epsilon}^{p,n} = \lambda^{n+1} f(\sigma^{n+1}) G'(\hat{\epsilon}^{p,n+1}) \\ f(\sigma^{n+1}) = G(\hat{\epsilon}^{p,n+1}), \end{cases} \quad \text{if } f(D \Delta \epsilon^n + \sigma^n) \geq G(\hat{\epsilon}^{p,n}),$$

where ΔF is the increments of the load at n -th step.

Since the above system is nonlinear we need some iterative method to solve it. There are several methods which are useful to this end. In EPITOK the following method is employed. Let $(u^n, \epsilon^n, \sigma^n, \hat{\epsilon}^{p,n}, \lambda^n)$ be the values at n -th step. Give a starting value $\Delta u_{(0)}^n$. Iterate (A)

~ (B) for $k = 0, 1, 2, \dots$

(A) Calculate $(\Delta\sigma_{(k)}^n, \Delta\hat{\epsilon}_{(k)}^{p,n}, \lambda_{(k)}^{n+1})$ by

$$(1.12) \quad \begin{cases} \Delta\epsilon_{(k)}^n - C\sigma_{(k)}^n = 0 \\ \Delta\hat{\epsilon}_{(k)}^{p,n} = 0 \quad \text{if } f(D\Delta\epsilon_{(k)}^n + \sigma^n) < G(\hat{\epsilon}_{(k)}^{p,n}) \end{cases}$$

$$(1.13) \quad \begin{cases} \Delta\epsilon_{(k)}^n - C\Delta\sigma_{(k)}^n = \lambda_{(k)}^{n+1}S\sigma_{(k)}^{n+1} \\ \Delta\hat{\epsilon}_{(k)}^{p,n} = \lambda_{(k)}^{n+1}f(\sigma_{(k)}^{n+1})G'(\hat{\epsilon}_{(k)}^{p,n+1}) \\ f(\sigma_{(k)}^{n+1}) = G(\hat{\epsilon}_{(k)}^{p,n+1}) \quad \text{if } f(D\Delta\epsilon_{(k)}^n + \sigma^n) \geq G(\hat{\epsilon}_{(k)}^{p,n}) \end{cases}$$

where $\Delta\epsilon_{(k)}^n = \epsilon_{(k)}^{n+1} - \epsilon^n$ etc., and $\epsilon_{(k)}^{n+1} = \epsilon(u_{(k)}^{n+1})$.

(B) Calculate $\Delta u_{(k+1)}^n$ by

$$(1.14) \quad \tilde{K}\Delta u_{(k+1)}^n = \tilde{K}\Delta u_{(k)}^n - (\Sigma B^* \Delta\sigma^n \cdot \text{area}(\text{element}) - \Delta F),$$

where \tilde{K} is the total elastic stiffness matrix with Poisson's ratio $\tilde{\nu}$ and Young's modulus \tilde{E} modified by the formulas

$$\begin{aligned} \tilde{\nu} &= \nu, \tilde{E} = E && \text{for elastic elements} \\ \tilde{\nu} &= \nu \frac{1 + \lambda^n E / 2\nu}{1 + \lambda^n E}, \tilde{E} = \frac{E}{1 + \lambda^n E} && \text{for plastic elements,} \end{aligned}$$

The above replacements of the elastic constants are effective for accelerating the convergence. It was motivated by the observation that the matrix

$$(I + \lambda_{(k)}^{n+1}DS)^{-1}D$$

is the only variable matrix to determine $\Delta\sigma_{(k)}^n$, and it will not be so different from the matrix

$$(I + \lambda^n DS)^{-1}D = \frac{\tilde{E}}{1 - \tilde{\nu}^2} \begin{pmatrix} 1 & \tilde{\nu} & 0 \\ \tilde{\nu} & 1 & 0 \\ 0 & 0 & (1 - \tilde{\nu})/2 \end{pmatrix}$$

We summarize the theoretical results on this type of approximation obtained up to the present.

- (1) The accuracy of the scheme (1.9) ~ (1.11) is better than those obtained by the finite difference approximation to (1.1) ~ (1.3).
- (2) $\lambda_{(k)}^{n+1}$, $\Delta\sigma_{(k)}^n$ and $\Delta\hat{\epsilon}_{(k)}^{p,n}$ in (1.12) ~ (1.13) are uniquely determined.

(3) If $\tilde{\nu} = \nu$ and $\tilde{E} = E$ for all elements, and the hardening is kinematic, then the iteration (1.14) converges for any load increments ΔF . For the proof, see [2]. Some numerical results obtained by using this program are reported in [3].

References

- 1) T. Miyoshi, Foundations of the numerical analysis of plasticity, North-Holland Mathematics Studies 107, Kinokuniya/North-Holland, (1985).
- 2) T. Miyoshi, A new iterative method for solving quasi-static problem of plasticity (to appear in Japan Journal of Applied Mathematics).
- 3) T. Miyoshi, K. Kaminishi, S. Kawano and S. Shimizu, an iterative method for elastic-plastic stress analysis, Computational Mechanics '86, Theory and Applications, Springer, IV-89 ~ IV-94 (1986).

2. The main program of EPITOK-II

The main program consists of four main blocks: Input and output, initiation, determination of yielding or unloading and inner iteration. Fig. 1 is the flow chart of the

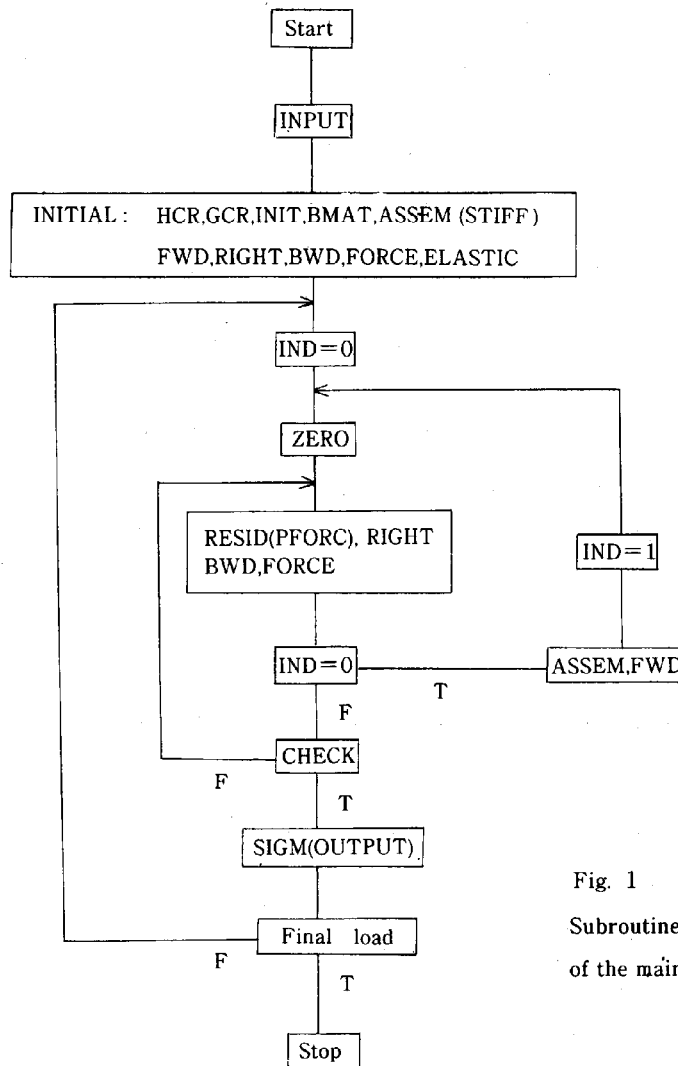


Fig. 1
Subroutines in the flow chart
of the main program

main program in which the disposition of the subroutines is shown.

00290-00380 Initiation and elastic solution.
 00400-00880 Computation of the increments of displacements, strains, stresses and plasticity parameters by the iteration

$$\tilde{K} \Delta u_{(k+1)}^n = \tilde{K} \Delta u_{(k)}^n - (\Sigma B^* \Delta \sigma_{(k)}^n \cdot \text{area}(\text{element}) - \Delta F)$$

00450 Compute $G^2(\hat{\epsilon}^{p,n})$, and clear the arrays of $D\Delta\epsilon^n$, $\Delta\epsilon^n$ and Δu^n .
 00470 QP: Parameter to check the convergence of the iteration 00490-00830. This check is done every ten iterations.
 00490-00580 Check if unloading occurs. This check is done by using old TK and new load increment during five extra iterations.
 00590-00630 Assemble the new iteration matrix TK associated with $(I + \lambda^{n+1}DS)^{-1}D$ and carry a new LU decomposition to start the next step.
 00650 If NUE is not a multiple of ten, the next convergence check is skipped.
 00670-00680 Convergence check by the criterion

$$\| \Delta u_{(k+1)}^n - \Delta u_{(k)}^n \| / \| \Delta u_{(k)}^n \| \leq \text{DELTA}.$$

00720-00750 Check if NUE is a multiple of ten.
 00770-00810 Store $u_{(k+1)}^{n+1}$ in AL1 and continue the iteration.
 00850 If the convergence criterion is satisfied, stop the iteration and add the increments Δu^n etc. to u^n etc. to get u^{n+1} etc..
 00900-00910 If NUE becomes greater than 300, the iteration is regarded as divergent.

3. The subroutines of EPITOK—II

In this section we explain the details of the subroutines used in EPITOK—II. For simplicity the explanation of the well known routines in the finite element technique is omitted.

SUBROUTINE INPUT

01080 Read NN, NE, NLN, NBN, E, ANU
 01190 Read ZO, DF, DELTA, ETA, NEQ, IFINAL
 01290 Read EQPSN, H
 01420 Read X, Y
 01530 Read ICON
 01650 Read IB, DISP
 01860 Read IZAL, ZAL, AL
 01990-02000 Read IS1, IS2, IS3, IS4.
 Read UN for nodes IS1 ~ IS2, STRESS etc. for elements IS3 ~ IS4 are printed.

SUBROUTINE INITIAL

Initiation to begin plastic computation

- 02270 Compute and store the vector HC (= H').
 02280 Compute and store the vector GOC (= G).
 02290 Initialise the variables of plastic parameters and compute the half bandwidth.
 02300 Compute and store "B" matrix.
 02320-02330 Assemble the total stiffness matrix associated with
 $(I + \lambda DS)^{-1}D$ ($\lambda = 0$ at this stage).
 and introduce the boundary conditions.
 02340 Forward elimination of the symmetric, banded matrix TK.
 02350 Modify the right side of the equations corresponding to the forward elimination.
 02360 Backward substitution.
 02370 Compute the stress $D \epsilon^p$.
 02380 Compute the minimum load and corresponding displacements, strains and stresses to cause the first yielding.

SUBROUTINE HCR

Computation and store of $H'(\bar{\epsilon}^p(i))$ under the assumption that $G'(\hat{\epsilon}^p)$ is piecewise linear. $H'(0)$ is determined by the derivative of the interpolating function of $H(1)$, $H(2)$ and $H(3)$ by a quadratic polynomial. $H'(i+1)$ is calculated by the formula

$$H'(i+1) = H'(i)(1 - ax)$$

where

$$a = [H(i+1) - H(i)] / [H'(i)(\bar{\epsilon}^p(i+1) - \bar{\epsilon}^p(i))]$$

and x is the solution of the equation

$$ax = 1 - \exp(-x).$$

To get the above formula, solve the following equation on H' assuming that $G'(\hat{\epsilon}^p)$ is linear in the same interval:

$$\sqrt{H'(\bar{\epsilon}^p)} = G'(\hat{\epsilon}^p)$$

$$\hat{\epsilon}^p = \int_0^{\bar{\epsilon}^p} \sqrt{H'(s)} ds \quad (\bar{\epsilon}^p(i) \leq \bar{\epsilon}^p \leq \bar{\epsilon}^p(i+1))$$

- 02660-02690 Newton's method applied to the above equation.

SUBROUTINE GCR

Compute and store $G'(\hat{\epsilon}^P(i))$ and $G(\hat{\epsilon}^P(i))$ by the formulas

$$\hat{\epsilon}^P = \int_0^{\bar{\epsilon}^P} \sqrt{H'(s)} ds, \quad G'(\hat{\epsilon}^P) = \sqrt{H'(\bar{\epsilon}^P)}$$

and

$$G(\hat{\epsilon}^P) = Z_0 + \int_0^{\bar{\epsilon}^P} G'(S) ds.$$

$G'(\hat{\epsilon}^P)$ is assumed to be linear in $[\hat{\epsilon}^P(i), \hat{\epsilon}^P(i+1)]$.

FUNCTION GOCR(X)

Function subroutine to define

$$G(\hat{\epsilon}^P) = Z_0 + \int_0^{\bar{\epsilon}^P} G'(s) ds.$$

FUNCTION G1CR(X)

Function subroutine to define

$$G'(\hat{\epsilon}^P) = dG(\hat{\epsilon}^P)/d\hat{\epsilon}^P.$$

SUBROUTINE INIT

Initialisation of the arrays for parameters and calculation of common variables. Half bandwidth MS is calculated in this subroutine.

SUBROUTINE BMAT

The so called "B" matrix is calculated and stored in arrays B and D. The area of each element is also calculated and stored.

SUBROUTINE ASSEM

Assembling of the total stiffness matrix associated with $(I + \lambda DS)^{-1}D$, and introduction of the fixed boundary condition.

SUBROUTINE STIFF

Computation of the element stiffness matrix associated with $(I + \lambda DS)^{-1}D$.

SUBROUTINE FWD

Forward elimination of the symmetric, banded matrix A.

SUBROUTINE RIGHT

Modification of the right side B of the equation $AU = B$ corresponding to the forward elimination.

SUBROUTINE BWD

SUBROUTINE BWD

Backward substitution for the equation $AU = B$.

SUBROUTINE FORCE

Computation of the vectors $D\Delta\epsilon_{(k)}^n$ and $\Delta\epsilon_{(k)}^n$.

SUBROUTINE ELASTIC

The minimum load to cause the first yielding is calculated with the corresponding displacements, strains and stresses.

SUBROUTINE ZERO

Computation of $G^2(\hat{\epsilon}^{p,n})$ and initiation of $D\Delta\epsilon^n$, $\Delta\epsilon^n$ and u^{n+1} .

SUBROUTINE RESID

Right side of the equation to determine $\Delta u_{(k+1)}^n$ is calculated and superposed in the array AL.

07120-07240 if $f(D\Delta\epsilon^n + \sigma^n) < G(\hat{\epsilon}^{p,n}) - 0.001$, then IEP = 0
 if $f(D\Delta\epsilon^n + \sigma^n) \geq G(\hat{\epsilon}^{p,n}) - 0.001$, then IEP = 1.

07320-07790 if IEP = 0, then skip this loop
 if IEP = 1, then calculate

ADF $(= (I + \lambda^n DS)^{-1} D\Delta\epsilon_{(k)}^n) - DDF (= \Delta\sigma_{(k)}^n)$.

DDF, $\lambda_{(k)}^{n+1}$ and EHNPI are stored in arrays DSIGM, ARAMDA and AEH.

07810-07870 Introduce the boundary conditions.

07890-07970 Add the next load increments.

SUBROUTINE PFORC

The vector $DDF = \Delta\sigma_{(k)}^n = (I + \lambda_{(k)}^{n+1} DS)^{-1} (D\Delta\epsilon_{(k)}^n + \sigma^n) - \sigma^n$ is calculated in this subroutine.

08210-08710 Newton's method applied to the equation
 FRAM(λ) = FSIGM(λ) - GEHP(λ)
 (= | $f(\sigma) - G(\hat{\epsilon}^p) | (\lambda)$)
 = 0

The iteration ends if the following is satisfied.

$ABS(f(\sigma) - G(\hat{\epsilon}^p)) / f(\sigma) \leq ETA$.

08210-08260 Compute

$$(I + \lambda DS)^{-1} = \begin{pmatrix} AA & BB & 0 \\ BB & AA & 0 \\ 0 & 0 & CC \end{pmatrix}$$

08270-08290	Compute $(I + \lambda DS)^{-1}(D \Delta \epsilon^n + \sigma^n) = \sigma^{n+1}$.
08300-08310	FSIGM = $f(\sigma^{n+1})$.
08320	$A = \lambda f(\sigma^{n+1})$
08340-08370	Find such i that $\hat{\epsilon}^P(i-1) \leq \hat{\epsilon}^{P,n} < \hat{\epsilon}^P(i)$.
08380-08450	Find such j that $\hat{\epsilon}^P(j) - \hat{\epsilon}^{P,n} > \lambda f(\sigma^{n+1})G'(\hat{\epsilon}^P(j))$.
08460	GK1 = $G'(j-1)$
08460-08500	Compute such $x = \hat{\epsilon}^P$ that $x - \hat{\epsilon}^{P,n} = \lambda f(\sigma^{n+1})G'(x)$.
08520	G1CRM = $G'(\hat{\epsilon}^P)$
08530	G2CRM = $G''(\hat{\epsilon}^P)$
08550	FRAM = $f(\sigma^{n+1}) - G(\hat{\epsilon}^P)$
08560-08580	Compute $S \sigma^{n+1}$
08590-08610	Compute $(I + \lambda DS)^{-1}S$
08620-08640	Compute $SD(I + \lambda DS)^{-1}S \sigma^{n+1}$
08650	$df/d\lambda = -\langle SD(I + \lambda DS)^{-1}S \sigma^{n+1}, \sigma^{n+1} \rangle / f(\sigma^{n+1})$
08660-08670	$d\hat{\epsilon}^P/d\lambda = \langle (I + \lambda DS)^{-1}S \sigma^{n+1}, \sigma^{n+1} \rangle G'(\hat{\epsilon}^P) / \{f(\sigma^{n+1})(1 - \lambda f(\sigma^{n+1})G''(\hat{\epsilon}^P))\}$
08680	FCR = $F'(\lambda) = df/d\lambda - G'(\hat{\epsilon}^P)d\hat{\epsilon}^P/d\lambda$
08690	$\lambda_{(k+1)} = \lambda_{(k)} - F(\lambda_{(k)})/F'(\lambda_{(k)})$
08710	Iterate until the following is satisfied. $ f(\sigma^{n+1}) - G(\hat{\epsilon}^P) / f(\sigma^{n+1}) \leq \text{ETA}$.
08730	EHNP1 = $\hat{\epsilon}^{P,n+1} = \hat{\epsilon}^{P,n} + \lambda f(\sigma^{n+1})G'(\hat{\epsilon}^{P,n+1})$
08740-08760	DDF = $\sigma^{n+1} - \sigma^n$.

SUBROUTINE CHECK

Convergence check by a relative error of displacement increments.

SUBROUTINE SIGM

Preparation to proceed to the next step of integration.

09230-09270	$u^{n+1} = u^n + \Delta u^n$.
09350-09640	if IEP = 0 $\sigma^{n+1} = \sigma^n + D \Delta \epsilon^n$ $\epsilon^{n+1} = \epsilon^n + \Delta \epsilon^n$ $\lambda = 0$ and calculate $f(\sigma^{n+1})$,
	if IEP = 1 $\sigma^{n+1} = \sigma^n + \Delta \sigma^n$ $\epsilon^{n+1} = \epsilon^n + \Delta \epsilon^n$ $\lambda^{n+1} = \text{ARAMADA(NEL)}$ $\hat{\epsilon}^{P,n+1} = \text{AEH(NEL)}$ $\hat{\epsilon}^{P,n+1} = \hat{\epsilon}^{P,n} + 0.5(1/G'(\hat{\epsilon}^{P,n+1}) + 1/G'(\hat{\epsilon}^{P,n})) \Delta \hat{\epsilon}^{P,n}$ (approximate value of

$$\bar{\epsilon}^{p,n+1} = \int_0^{\hat{\epsilon}^{p,n+1}} \frac{1}{G'(s)} ds$$

by the trapezoidal rule),

and calculate $f(\sigma^{n+1})$.

09660-09700 Compute the total load.

SUBROUTINE OUTPUT

Print: ISTAT, FX, FY, NUE
 IEP(I) (I = 1, NE)
 UN(2(IS1-1)+1) ~ UN(2(IS2-1)+2)
 FSO(IS3) ~ FSO(IS4)
 EEQ(IS3) ~ EEQ(IS4)
 ZRAMDA(IS3) ~ ZRAMDA(IS4)
 STRESS(k), STRAIN(k)
 (k = 3(IS3-1)+1, 3(IS4-1)+3)

4. The program of EPITOK-II

1. Comments

- C1. Problem incremental, quasi-static, plane stress problem
- C2. Hardening isotropic hardening
- C3. Finite element linear on triangular element
- C4. Numerical method accelerated iterative method
- C5. Maximum number of elements 300
- C6. Maximum number of loaded nodes 20
- C7. Maximum number of fixed nodes 20
- C8. Maximum half bandwidth 150
- C9. Maximum number of values of $H(\bar{\epsilon}^p)$ as input data 20
- C10. The change of the thickness is not considered.

2. Data, variables and arrays

NN : number of nodes
 NE : number of elements
 NLN : number of loaded nodes
 NBN : number of boundary nodes
 N : $2 \times NN$ (number of unknowns)
 MS : half bandwidth of total stiffness matrix
 E : Young's modulus
 ANU : Poisson's ratio
 ZO : yield stress
 ETA : relative error to stop Newton's iteration in PFORC
 NUE : number of iteration in the main program
 DELTA : relative error to stop iteration on NUE

- DF : ratio to determine the load increment
 IFINAL : maximum number of integration step
 NEQ : number of $\bar{\epsilon}^p(i)$ as input data (≤ 20)
 EQPSN(i) : input data $\bar{\epsilon}^p(i)$
 H(i) : input data $H(\bar{\epsilon}^p(i))$
 HC(i) : $H'(i)$ calculated from $H(i)$
 EQPSNH(i) : calculated $\hat{\epsilon}^p(i)$ from input data $\bar{\epsilon}^p(i)$
 GOC(i) : calculated $G(\hat{\epsilon}^p(i))$ from $G'(\hat{\epsilon}^p(i))$
 G1C(i) : calculated $G'(\hat{\epsilon}^p(i))$ from $HC(i)$
 X(i) : x-coordinate of i-th node
 Y(i) : y-coordinate of i-th node
 ICON(i) : one dimensional array of node. $ICON(3(k-1)+1)$, $ICON(3(k-1)+2)$ and $ICON(3(k-1)+3)$ denote the first, second and third nodes of k-th elements placed counterclockwise.
 IB(i) : one dimensional array of boundary data. $IB(3(k-1)+1)$ denotes the node number of k-th fixed node, and $IB(3(k-1)+2)$ and $IB(3(k-1)+3)$ denote the status of the displacements in x and y directions. 1 denotes free and 0 denotes prescribed.
 (BA, DA) : "B" matrix of an element
 IZAL(i) : node number of i-th loaded node
 ZAL(i) : one dimensional array of the load vector given initially. $ZAL(2(k-1)+1)$ and $ZAL(2(k-1)+2)$ include x and y components of the load at k-th node. $DF * ZAL$ is the load increment at one step.
 AL(i) : one dimensional array of load or increments of displacements. $AL(2(k-1)+1)$ and $AL(2(k-1)+2)$ denote x and y components of the load. After solving the equilibrium equation this array is used to store the increments of displacements.
 TK(i, j) : total stiffness matrix
 FORC(i) : one dimensional array of $D\Delta\epsilon^n$
 DEPS(i) : one dimensional array of strain increments $\Delta\epsilon^n$ at each step.
 ELST(i, j) : element stiffness matrix
 EH(i) : $\hat{\epsilon}^{p,n}$ of element i
 AEH(i) : $\hat{\epsilon}^{p,n+1}$ of element i
 STRAIN(i) : one dimensional array of strains
 STRESS(i) : one dimensional array of stresses
 DSIGM(i) : one dimensional array of stress increments
 UN(i) : one dimensional array of displacements
 FSO(i) : $f(\sigma)$ of element i
 ZN2(i) : $f^2(\sigma) - 0.001$. $ZN2(i) \geq G(\hat{\epsilon}^p)$ is used to determine yielding.
 EEQ(i) : $\bar{\epsilon}^{p,n}$ of element i
 ZRAMDA(i) : λ^n of element i
 ARAMDA(i) : λ^{n+1} of element i
 IPE(i) : parameter to distinct elastic and plastic. 1 denotes plastic and 0 elastic.
 ADF(i) : the vector $(I + \lambda^n DS)^{-1} D\Delta\epsilon^n$ of an element
 DDF(i) : the vector $\Delta\sigma^n$ of an element

FX : x—component of total load
FY : y—component of total load

Finite Element Program EPITOK-II

```

00010C-----
00020C
00030C
00040C          *****      EPITOK-II      *****
00050C
00060C
00070C          ELASTIC-PLASTIC ITERATIVE
00080C          METHOD TOKIWA-II
00090C
00100C-----
00110C
00120C
00130C      COMMON /INP/ NN,NE,NLN,NBN,N,MS
00140C      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
00150C      COMMON /RR/  RA,RB,RC
00160C      COMMON /EQ/  NEQ,EQPSN(20),H(20)
00170C      COMMON /GC/  HC(20),EQPSNH(20),GOC(20),G1C(20)
00180C      COMMON /OUT/ IN,IO,IS1,IS2,IS3,IS4,ISTAT
00190C      COMMON /ELT/ NUE,FX,FY
00200C
00210C      DIMENSION X(300),Y(300),ICON(900),IB(60),ZRAMDA(300),
00220C      &          TK(600,150),AL(600),FORC(900),ELST(20,20),
00230C      &          IEP(300),AL1(600),ARAMDA(300),DSIGM(900),
00240C      &          EH(300),DEPS(900),STRAIN(900),ZAL(40),AEH(300),
00250C      &          STRESS(900),UN(600),FSO(300),EEQ(300),ZN2(300),
00260C      &          B(300,3),D(300,3),BA(300,3),DA(300,3),IZAL(20)
00270C
00280C
00290C      NRMX=600
00300C      NCMX=150
00310C      IN=5
00320C      IO=6
00330C
00340C
00350C      CALL INPUT(X,Y,ICON,AL,IB,ZAL,IZAL)
00360C      CALL INITIAL(X,Y,ICON,AL,IB,ZAL,ZRAMDA,EH,ISTAT,
00370C      &          B,D,BA,DA,TK,ELST,IEP,NRMX,NCMX,FORC,
00380C      &          DEPS,STRESS,FSO,STRAIN,UN)
00390C
00400C      10  NUE=0
00410C          IND=0
00420C          KK=0
00430C          ISTAT=ISTAT+1
00440C
00450C      20  CALL ZERO(EH,ZN2,FORC,DEPS,AL1)
00460C
00470C          QP=1.0
00480C
00490C      30  CALL RESID(FORC,STRESS,ZN2,IEP,AL,ZRAMDA,EH,ARAMDA,EHNP1,
00500C      &          ICON,IB,ZAL,B,D,DSIGM,ARAMDA,AEH,IZAL)
00510C
00520C      CALL RIGHT(TK,AL,N,MS,NRMX,NCMX)
00530C      CALL BWD(TK,AL,N,MS,NRMX,NCMX)
00540C      CALL FORCE(ICON,FORC,AL,DEPS,BA,DA)
00550C
00560C      IF(IND.NE.0) GOTO 50
00570C      KK=KK+1

```

```

00580      1F(KK-5) 30,30,40
00590  40 CALL ASSEM(ICON,TK,ELST,ZRAMDA,B,D,BA,DA,IEP,
00600      &          AL,IB)
00610      CALL FWD(TK,N,MS,NRMX,NCMX)
00620      IND=1
00630      GOTO 20
00640C
00650  50 IF(QP.GT.0.01) GOTO 60
00660C
00670      CALL CHECK(AL,AL1,RSD,RDEL)
00680      IF(RSD-RDEL) 90,90,60
00690C
00700  60 IF(NUE-300) 70,100,100
00710C
00720  70 NUE=NUE+1
00730      RN=FLOAT(NUE)
00740      QP=RN*0.1-AINT(RN*0.1)
00750      IF(QP.GT.0.01) GOTO 30
00760C
00770      DO 80 I=1,NN
00780          L=2*(I-1)+1
00790          AL1(L)=AL(L)
00800          AL1(L+1)=AL(L+1)
00810  80 CONTINUE
00820C
00830      GO TO 30
00840C
00850  90 CALL SIGM(UN,AL,IEP,FORC,STRESS,EH,ZRAMDA,DEPS,STRAIN,
00860      &          EEQ,ZAL,DSIGM,ARAMDA,AEH)
00870C
00880      IF(ISTAT-IFINAL) 10,120,120
00890C
00900  100 WRITE(10,110)
00910  110 FORMAT(5X,"DIVERGENT")
00920C
00930  120 STOP
00940      END
00950C
00960C
00970      SUBROUTINE INPUT(X,Y,ICON,AL,IB,ZAL,IZAL)
00980C
00990      COMMON /INP/ NN,NE,NLN,NBN,N,MS
01000      COMMON /OUT/ IN,IO,IS1,IS2,IS3,IS4,ISTAT
01010      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
01020      COMMON /EQ/  NEQ,EQPSN(20),H(20)
01030C
01040      DIMENSION X(300),Y(300),ICON(900),AL(600),IB(60),
01050      &          W(3),IC(3),ZAL(40),IZAL(20)
01060C
01070C
01080      READ(IN,10) NN,NE,NLN,NBN,E,ANU
01090      WRITE(10,20) NN,NE,NLN,NBN,E,ANU
01100      10 FORMAT(4I5,2F10.0)
01110      20 FORMAT(///57X," FUNDAMENTAL DATA  "
01120      &          //15X,"      NUMBER OF NODES      = ",15,
01130      &          30X,"      NUMBER OF ELEMENTS      = ",15,
01140      &          //15X,"      NUMBER OF LOADED NODES = ",15,
01150      &          30X,"      NUMBER OF BOUNDARY NODES = ",15,
01160      &          //15X,"      YOUNG'S MODULUS      = ",1PE12.6,
01170      &          22X,"      POISSON'S RATIO      = ",1PE12.6)

```

```

01180C
01190 READ(IN,30) ZO,DF,DELTA,ETA,NEQ,IFINAL
01200 30 FORMAT(4E10.3,2I10)
01210 WRITE(10,40) ZO,DF,DELTA,ETA,NEQ,IFINAL
01220 40 FORMAT(/17X," YIELD STRESS      =",1PE14.7,
01230 &      23X," INCREMENT OF LOAD    =",1PE14.7,
01240 &      //17X," CONVERGENCE OF U     =",1PE14.7,
01250 &      23X,"CONVERGENCE OF RAMDA   =",1PE14.7,
01260 &      //26X,"NEG",11X,"=",14,40X," IFINAL",10X,"=",14)
01270C
01280 DO 50 I=1,NEQ
01290 READ(IN,60) J,EQPSN(I),H(I)
01300 50 CONTINUE
01310 60 FORMAT(15,2F10.0)
01320 WRITE(10,70)
01330 70 FORMAT(/3(5X,"NO.",5X,"EQ.P.STRAIN",5X,"EQ.STRESS"))
01340 WRITE(10,80) (I,EQPSN(I),H(I),I=1,NEQ)
01350 80 FORMAT(/3(18,6X,F10.3,4X,F10.3))
01360C
01370 WRITE(10,90)
01380 90 FORMAT(///48X," COORDINATES OF NODES "
01390 &      //14X,"NODE",8X,"X",9X,"Y",17X,"NODE",8X,"X",
01400 &      9X,"Y",17X,"NODE",8X,"X",9X,"Y"//)
01410 DO 100 I=1,NN
01420 READ(IN,110) J,X(I),Y(I)
01430 100 CONTINUE
01440 110 FORMAT(15,2F10.0)
01450 WRITE(10,120) (I,X(I),Y(I),I=1,NN)
01460 120 FORMAT(8X,110,2F10.2,10X,110,2F10.2,10X,110,2F10.2)
01470C
01480 WRITE(10,130)
01490 130 FORMAT(///62X," ELEMENT CONNECTIONS "
01500 &      //8X,3(2X,"ELEMENT NO.",5X,"I",5X,"J",5X,"K",8X)//)
01510 DO 140 I=1,NE
01520 L=3*(I-1)+1
01530 READ(IN,150) J,ICON(L),ICON(L+1),ICON(L+2)
01540 140 CONTINUE
01550 WRITE(10,160) (J,ICON(3*J-2),ICON(3*J-1),ICON(3*J),J=1,NE)
01560 150 FORMAT(4I5)
01570 160 FORMAT(3(11X,16,5X,3I6))
01580C
01590 WRITE(10,170)
01600 170 FORMAT(///43X," BOUNDARY CONDITIONS ",
01610 &      "(FREE=1,FIXED=0) "
01620 &      //10X,3("NODE",9X,"X",9X,"Y",16X)//)
01630C
01640 DO 180 I=1,NBN
01650 READ(IN,190) J,(IC(K),K=1,2)
01660 L1=3*(I-1)+1
01670 L2=2*(J-1)
01680 IB(L1)=J
01690 DO 180 K=1,2
01700 N1=L1+K
01710 N2=L2+K
01720 IB(N1)=IC(K)
01730 180 CONTINUE
01740 190 FORMAT(3I10)
01750 WRITE(10,200) (IB(3*(K-1)+1),IB(3*(K-1)+2),
01760 &      IB(3*(K-1)+3),K=1,NBN)
01770 200 FORMAT(10X,14,2I10,16X,14,2I10,16X,14,2I10)

```



```

01780C
01790      WRITE(10,210)
01800 210 FORMAT(//50X," LOADED CONDITIONS  "
01810      &      //7X,3("NODE",9X,"FX",9X,"FY",16X)//)
01820      N=NN*2
01830      DO 220 I=1,N
01840 220 AL(I)=0.
01850      DO 240 I=1,NLN
01860      READ(IN,230) J, (W(K),K=1,2)
01870 230 FORMAT(110,2F10.0)
01880      N1=2*(I-1)+1
01890      IZAL(I)=J
01900      ZAL(N1)=W(1)
01910      ZAL(N1+1)=W(2)
01920      DO 240 K=1,2
01930      L=2*(J-1)+K
01940 240 AL(L)=W(K)
01950      WRITE(10,250) (IZAL(J),ZAL(2*(J-1)+1),
01960      &      ZAL(2*(J-1)+2),J=1,NLN)
01970 250 FORMAT(3(111,5X,1PE10.4,1X,1PE10.4,5X))
01980C
01990      READ(IN,260) IS1,IS2
02000      READ(IN,260) IS3,IS4
02010 260 FORMAT(2110)
02020C
02030      WRITE(10,270)
02040 270 FORMAT(//5X,12("-----"))//)
02050C
02060      RETURN
02070      END
02080C
02090C
02100      SUBROUTINE INITIAL(X,Y,ICON,AL,IB,ZAL,ZRAMDA,EH,ISTAT,
02110      &      B,D,BA,DA,TK,ELST,IEP,NRMX,NCMX,
02120      &      FORC,DEPS,STRESS,FSO,STRAIN,UN)
02130C
02140      COMMON /INP/ NN,NE,NLN,NBN,N,MS
02150      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
02160      COMMON /RR/  RA,RB,RC
02170      COMMON /EQ/  NEQ,EQPSN(20),H(20)
02180      COMMON /GC/  HC(20),EQPSNH(20),GOC(20),G1C(20)
02190      COMMON /ELT/ NUE,FX,FY
02200C
02210      DIMENSION X(300),Y(300),ICON(900),IB(60),ZRAMDA(300),
02220      &      TK(600,150),AL(600),FORC(900),ELST(20,20),
02230      &      EH(300),DEPS(900),STRAIN(900),ZAL(40),
02240      &      STRESS(900),UN(600),FSO(300),IEP(300),
02250      &      B(300,3),D(300,3),BA(300,3),DA(300,3)
02260C
02270      CALL HCR
02280      CALL GCR
02290      CALL INIT(ICON,ZRAMDA,EH,ISTAT)
02300      CALL BMAT(X,Y,ICON,B,D,BA,DA)
02310C
02320      CALL ASSEM(ICON,TK,ELST,ZRAMDA,B,D,BA,DA,IEP,AL,IB)
02330C
02340      CALL FWD(TK,N,MS,NRMX,NCMX)
02350      CALL RIGHT(TK,AL,N,MS,NRMX,NCMX)
02360      CALL BWD(TK,AL,N,MS,NRMX,NCMX)
02370      CALL FORCE(ICON,FORC,AL,DEPS,BA,DA)

```

```

02380      CALL ELASTIC(FORC,STRESS,FSO,DEPS,STRAIN,AL,UN,ZAL)
02390C
02400      RETURN
02410      END
02420C
02430C
02440      SUBROUTINE HCR
02450C
02460      COMMON /EQ/  NEQ,EQPSN(20),H(20)
02470      COMMON /GC/  HC(20),EQPSNH(20),GOC(20),G1C(20)
02480C
02490      A1=H(2)-H(1)
02500      A2=H(3)-H(1)
02510      EQ1=EQPSN(2)-EQPSN(1)
02520      EQ2=EQPSN(3)-EQPSN(1)
02530      EQ3=EQPSN(3)-EQPSN(2)
02540C
02550      HC(1)=(A1/EQ1)*(EQ2/EQ3)-(A2/EQ2)*(EQ1/EQ3)
02560C
02570      I=0
02580      10 I=I+1
02590      A1=H(I+1)-H(I)
02600      EQ1=EQPSN(I+1)-EQPSN(I)
02610      A=A1/A
02620      A=A1/A
02630      IF(A.GE.1.0-0.0001) GOTO 30
02640      RLOG=10.0
02650      DO 20 I=1,6
02660      REX=EXP(-RLOG)
02670      RLOG=RLOG-(A*RLOG-1.0+REX)/(A-REX)
02680      20 CONTINUE
02690      HC(I+1)=HC(I)*(1.0-A*RLOG)
02700      IF(I.LT.NEQ-1) GOTO 10
02710      GOTO 40
02720      30 HC(I+1)=HC(I)
02730      IF(I.LT.NEQ-1) GOTO 10
02740C
02750      40 RETURN
02760      END
02770C
02780C
02790      SUBROUTINE GCR
02800C
02810      COMMON /EQ/  NEQ,EQPSN(20),H(20)
02820      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
02830      COMMON /GC/  HC(20),EQPSNH(20),GOC(20),G1C(20)
02840C
02850      EQPSNH(1)=0.0
02860      DO 20 I=2,NEQ
02870      Z=0.0
02880      IM1=I-1
02890      DO 10 K=1,IM1
02900      Z=Z+0.5*(SQRT(HC(K+1))+SQRT(HC(K)))*(EQPSN(K+1)-EQPSN(K))
02910      10 CONTINUE
02920      EQPSNH(I)=Z
02930      20 CONTINUE
02940      DO 30 I=1,NEQ
02950      G1C(I)=SQRT(HC(I))
02960      30 CONTINUE
02970      GOC(1)=ZO

```

```

02980      DO 50 I=2,NEQ
02990      Z=0.0
03000      IM1=I-1
03010      DO 40 K=1,IM1
03020      Z=Z+0.5*(G1C(K+1)+G1C(K))*(EQPSNH(K+1)-EQPSNH(K))
03030      40 CONTINUE
03040      GOC(I)=Z+ZO
03050      50 CONTINUE
03060C
03070      RETURN
03080      END
03090C
03100C
03110      FUNCTION GOCR(X)
03120C
03130      COMMON /EQ/ NEQ,EQPSN(20),H(20)
03140      COMMON /GC/ HC(20),EQPSNH(20),GOC(20),G1C(20)
03150C
03160      E=EQPSNH(NEQ)
03170      IF(X.GE.E) GOTO 30
03180      K=1
03190      10 EH=EQPSNH(K+1)
03200      IF(X.LT.EH) GOTO 20
03210      K=K+1
03220      GOTO 10
03230      20 A=G1C(K)
03240      B=EQPSNH(K)
03250      C=X-B
03260      GOCR=GOC(K)+A*C
03270      GOCR=GOCR+0.5*(G1C(K+1)-A)*C*C/(EH-B)
03280      GOTO 40
03290      30 GOCR=GOC(NEQ)+G1C(NEQ)*(X-EQPSNH(NEQ))
03300C
03310      40 RETURN
03320      END
03330C
03340C
03350      FUNCTION G1CR(X)
03360C
03370      COMMON /EQ/ NEQ,EQPSN(20),H(20)
03380      COMMON /GC/ HC(20),EQPSNH(20),GOC(20),G1C(20)
03390C
03400      E=EQPSNH(NEQ)
03410      IF(X.GE.E) GOTO 30
03420      K=1
03430      10 EH=EQPSNH(K+1)
03440      IF(X.LT.EH) GOTO 20
03450      K=K+1
03460      GOTO 10
03470      20 A=G1C(K)
03480      B=EQPSNH(K)
03490      G1CR=A+(G1C(K+1)-A)*(X-B)/(EH-B)
03500      GOTO 40
03510      30 G1CR=G1C(NEQ)
03520C
03530      40 RETURN
03540      END
03550C
03560C
03570      SUBROUTINE INIT(ICON,ZRAMDA,EH,ISTAT)

```

```

03580C
03590      COMMON /INP/ NN,NE,NLN,NBN,N,MS
03600      COMMON /ELT/ NUE,FX,FY
03610      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
03620      COMMON /RR/ RA,RB,RC
03630C
03640      DIMENSION ICON(900),ZRAMDA(300),EH(300)
03650C
03660      ISTAT=0
03670      F=0.0
03680      FX=0.0
03690      FY=0.0
03700      RA=E*(1.0-0.5*ANU)/(1.0-ANU*ANU)
03710      RB=E*(ANU-0.5)/(1.0-ANU*ANU)
03720      RC=E*1.5*(1.0-ANU)/(1.0-ANU*ANU)
03730C
03740      DO 10 NEL=1,NE
03750      ZRAMDA(NEL)=0.0
03760      EH(NEL)=0.0
03770 10 CONTINUE
03780C
03790      N1=2
03800      MS=0
03810C
03820      DO 30 I=1,NE
03830      L1=3*(I-1)
03840      DO 30 J=1,N1
03850      L2=L1+J
03860      J1=J+1
03870      DO 30 K=J1,3
03880      L3=L1+K
03890      L=IABS(ICON(L2)-ICON(L3))
03900      IF(MS-L)20,30,30
03910 20 MS=L
03920 30 CONTINUE
03930      MS=2*(MS+1)
03940C
03950      RETURN
03960      END
03970C
03980C
03990      SUBROUTINE BMAT(X,Y,ICON,B,D,BA,DA)
04000C
04010      COMMON /INP/ NN,NE,NLN,NBN,N,MS
04020C
04030      DIMENSION X(300),Y(300),ICON(900),B(300,3),D(300,3),
04040      & BA(300,3),DA(300,3)
04050C
04060      DO 20 I=1,NE
04070      L=3*(I-1)+1
04080      N1=ICON(L)
04090      N2=ICON(L+1)
04100      N3=ICON(L+2)
04110C
04120      B(I,1)=Y(N2)-Y(N3)
04130      B(I,2)=Y(N3)-Y(N1)
04140      B(I,3)=Y(N1)-Y(N2)
04150      D(I,1)=X(N3)-X(N2)
04160      D(I,2)=X(N1)-X(N3)
04170      D(I,3)=X(N2)-X(N1)

```

```

04180     ARE=(B(1,1)*D(1,2)-B(1,2)*D(1,1))
04190     DO 10 J=1,3
04200     BA(1,J)=B(1,J)/ARE
04210     DA(1,J)=D(1,J)/ARE
04220     10 CONTINUE
04230     20 CONTINUE
04240C
04250     RETURN
04260     END
04270C
04280C
04290     SUBROUTINE ASSEM(ICON,TK,ELST,ZRAMDA,B,D,BA,DA,IEP,AL,IB)
04300C
04310     COMMON /INP/ NN,NE,NLN,NBN,N,MS
04320     COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
04330C
04340     DIMENSION ICON(900),TK(600,150),ELST(6,6),BA(300,3),ND(3),
04350     &          ZRAMDA(300),B(300,3),D(300,3),DA(300,3),IEP(300),
04360     &          AL(600),IB(60)
04370C
04380     DO 10 I=1,N
04390     DO 10 J=1,MS
04400     10 TK(I,J)=0.
04410C
04420     DO 30 I=1,NE
04430     NEL=I
04440C
04450     CALL STIFF(NEL,ELST,ZRAMDA,B,D,BA,DA,IEP)
04460C
04470     N1=3*(NEL-1)
04480     ND(1)=ICON(N1+1)
04490     ND(2)=ICON(N1+2)
04500     ND(3)=ICON(N1+3)
04510C
04520     DO 20 J=1,6
04530     JJ=(J+1)/2
04540     JM=2*JJ-J
04550     K=2*ND(JJ)-JM
04560     DO 20 L=1,6
04570     LL=(L+1)/2
04580     LM=2*LL-L
04590     M=2*ND(LL)-LM
04600     IF(K.GT.M) GOTO 20
04610     MM=M-K+1
04620     IF(J.GT.L) ELST(J,L)=ELST(L,J)
04630     TK(K,MM)=TK(K,MM)+ELST(J,L)
04640     20 CONTINUE
04650     30 CONTINUE
04660C
04670     DO 90 L=1,NBN
04680     L1=3*(L-1)+1
04690     NO=IB(L1)
04700     K1=2*(NO-1)
04710     DO 90 I=1,2
04720     L2=L1+I
04730     IF(IB(L2))90,40,90
04740     40 KR=K1+I
04750     DO 80 J=2,MS
04760     KV=KR+J-1
04770     IF(N-KV)60,50,50

```

```

04780 50 TK(KR,J)=0.
04790 60 KV=KR-J+1
04800 IF(KV)80,80,70
04810 70 TK(KV,J)=0.
04820 80 CONTINUE
04830 TK(KR,I)=1.
04840 AL(KR)=0.
04850 90 CONTINUE
04860C
04870 RETURN
04880 END
04890C
04900C
04910 SUBROUTINE STIFF(NEL,ELST,ZRAMDA,B,D,BA,DA,IEP)
04920C
04930 COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
04940C
04950 DIMENSION ELST(6,6),BA(300,3),IEP(300),
04960 & B(300,3),D(300,3),ZRAMDA(300),DA(300,3)
04970C
04980 DO 10 I=1,6
04990 DO 10 J=1,6
05000 10 ELST(I,J)=0.
05010C
05020 Q=ZRAMDA(NEL)
05030 Q=Q*E
05040 NA=IEP(NEL)
05050 IF(NA.EQ.0) Q=0.0
05060 ANUU=(ANU+Q/2.0)/(1.0+Q)
05070 ED=E/(1.0+Q)
05080 ANUP=(1.-ANUU)/2.
05090 C=0.5*ED/(1.0-ANUU*ANUU)
05100C
05110 DO 20 I=1,3
05120 I2=2*I
05130 DO 20 J=1,3
05140 J2=2*J
05150 ELST(I2-1,J2-1)=C*(B(NEL,I)*BA(NEL,J)+ANUP*D(NEL,I)*DA(NEL,J))
05160 ELST(I2-1,J2)=C*(B(NEL,I)*DA(NEL,J)*ANU+D(NEL,I)*BA(NEL,J)*ANUP)
05170 ELST(I2,J2-1)=C*(B(NEL,J)*DA(NEL,I)*ANU+D(NEL,J)*BA(NEL,I)*ANUP)
05180 20 ELST(I2,J2)=C*(B(NEL,I)*BA(NEL,J)*ANUP+D(NEL,I)*DA(NEL,J))
05190C
05200 RETURN
05210 END
05220C
05230C
05240 SUBROUTINE FWD(A,N,MS,NX,MX)
05250C
05260 DIMENSION A(NX,MX)
05270C
05280 N1=N-1
05290 NM2=N-MS+2
05300 DO 10 I=NM2,N
05310 N12=N-1+2
05320 DO 10 J=N12,MS
05330 10 A(I,J)=0.0
05340 DO 40 I=1,N1
05350 IL=MINO(MS+I-1,N)
05360 I1=I+1
05370 DO 30 K=I1,IL

```

```

05380      DO 20 JD=1,MS
05390      KJDI=K+JD-1
05400      K11=K-1+1
05410      20 A(K,JD)=A(K,JD)-A(I,KJDI)*A(I,K11)/A(I,1)
05420      30 CONTINUE
05430      40 CONTINUE
05440      RETURN
05450      END
05460C
05470C
05480      SUBROUTINE RIGHT(A,B,N,MS,NX,MX)
05490C
05500      DIMENSION A(NX,MX),B(NX)
05510C
05520      N1=N-1
05530      DO 20 I=1,N1
05540      IL=MIN(MS+I-1,N)
05550      I1=I+1
05560      DO 10 K=I1,IL
05570      K11=K-1+1
05580      10 B(K)=B(K)-B(I)*A(I,K11)/A(I,1)
05590      20 CONTINUE
05600      RETURN
05610      END
05620C
05630C
05640      SUBROUTINE BWD(A,B,N,MS,NX,MX)
05650C
05660      DIMENSION A(NX,MX),B(NX)
05670C
05680      B(N)=B(N)/A(N,1)
05690      N1=N-1
05700      M1=MS-1
05710      DO 20 I=1,M1
05720      N1=N-1
05730      RX=0.0
05740      DO 10 J=1,I
05750      J1=J+1
05760      N1J=N-1+J
05770      10 RX=RX+A(N1,J1)*B(N1J)
05780      20 B(N1)=(B(N1)-RX)/A(N1,1)
05790      DO 40 I=MS,N1
05800      N1=N-1
05810      RX=0.0
05820      DO 30 J=1,M1
05830      J1=J+1
05840      N1J=N-1+J
05850      30 RX=RX+A(N1,J1)*B(N1J)
05860      40 B(N1)=(B(N1)-RX)/A(N1,1)
05870      RETURN
05880      END
05890C
05900C
05910      SUBROUTINE FORCE(ICON,FORC,AL,DEPS,BA,DA)
05920C
05930      COMMON /INP/ NN,NE,NLN,NBN,N,MS
05940      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
05950C
05960      DIMENSION ICON(900),FORC(900),BA(300,3),DA(300,3),
05970      & AL(600),DEPS(900)

```

```

05980C      C=E/(1.-ANU*ANU)
05990
06000C
06010      DO 10 I=1,NE
06020          L=3*(I-1)+1
06030          K1=2*(ICON(L)-1)
06040          K2=2*(ICON(L+1)-1)
06050          K3=2*(ICON(L+2)-1)
06060          L=3*(I-1)
06070          DEPS(L+1)=BA(1,1)*AL(K1+1)+BA(1,2)*AL(K2+1)+BA(1,3)*AL(K3+1)
06080          DEPS(L+2)=DA(1,1)*AL(K1+2)+DA(1,2)*AL(K2+2)+DA(1,3)*AL(K3+2)
06090          DEPS(L+3)=DA(1,1)*AL(K1+1)+BA(1,1)*AL(K1+2)+DA(1,2)*AL(K2+1)+
06100          &          BA(1,2)*AL(K2+2)+DA(1,3)*AL(K3+1)+BA(1,3)*AL(K3+2)
06110          FORC(L+1)=C*(DEPS(L+1)+ANU*DEPS(L+2))
06120          FORC(L+2)=C*(ANU*DEPS(L+1)+DEPS(L+2))
06130          FORC(L+3)=C*(1.-ANU)*0.5*DEPS(L+3)
06140      10 CONTINUE
06150C
06160          RETURN
06170          END
06180C
06190C
06200          SUBROUTINE ELASTIC(FORC,STRESS,FSO,DEPS,STRAIN,AL,UN,ZAL)
06210C
06220          COMMON /INP/ NN,NE,NLN,NBN,N,MS
06230          COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
06240          COMMON /ELT/ NUE,FX,FY
06250          COMMON /OUT/ IN,IO,IS1,IS2,IS3,IS4,ISTAT
06260C
06270          DIMENSION FORC(900),STRESS(900),FSO(300),DEPS(900),STRAIN(900),
06280          &          AL(600),UN(600),ZAL(60)
06290          DO 10 I=1,NE
06300              N1=3*(I-1)+1
06310              FS1=FORC(N1)*FORC(N1)+FORC(N1+1)*FORC(N1+1)
06320              &          -FORC(N1)*FORC(N1+1)+3.0*FORC(N1+2)*FORC(N1+2)
06330              IF (FS1.GT.F) F=FS1
06340          10 CONTINUE
06350C
06360              F=SQRT(F)
06370              RATIO=ZO/F
06380C
06390              DO 20 I=1,NE
06400                  N1=3*(I-1)+1
06410                  STRESS(N1)=RATIO*FORC(N1)
06420                  STRESS(N1+1)=RATIO*FORC(N1+1)
06430                  STRESS(N1+2)=RATIO*FORC(N1+2)
06440                  FSO(I)=SQRT(STRESS(N1)*STRESS(N1)+STRESS(N1+1)*STRESS(N1+1)-
06450                  &          STRESS(N1)*STRESS(N1+1)+3.0*(STRESS(N1+2)*STRESS(N1+2)))
06460C
06470                  STRAIN(N1)=RATIO*DEPS(N1)
06480                  STRAIN(N1+1)=RATIO*DEPS(N1+1)
06490                  STRAIN(N1+2)=RATIO*DEPS(N1+2)
06500          20 CONTINUE
06510C
06520              DO 30 I=1,NN
06530                  N1=2*(I-1)+1
06540                  UN(N1)=RATIO*AL(N1)
06550                  UN(N1+1)=RATIO*AL(N1+1)
06560          30 CONTINUE
06570C

```



```

06580      DO 40 I=1,NLN
06590      N1=2*(I-1)+1
06600      FX=FX+RATIO*ZAL(N1)
06610      FY=FY+RATIO*ZAL(N1+1)
06620 40 CONTINUE
06630C
06640      CALL OUTPUT (STRESS,STRAIN,ZRAMDA,FSO,UN,IEP,EEQ)
06650C
06660      RETURN
06670      END
06680C
06690C
06700      SUBROUTINE ZERO(EH,ZN2,FORC,DEPS,AL1)
06710C
06720      COMMON /INP/ NN,NE,NLN,NBN,N,MS
06730C
06740      DIMENSION EH(300),ZN2(300),FORC(900),DEPS(900),AL1(600)
06750C
06760      DO 10 NEL=1,NE
06770      EHAT=EH(NEL)
06780      GNEP=GOCR(EHAT)
06790      ZN2(NEL)=GNEP*GNEP-0.001
06800      L=3*(NEL-1)+1
06810      FORC(L)=0.0
06820      FORC(L+1)=0.0
06830      FORC(L+2)=0.0
06840      DEPS(L)=0.0
06850      DEPS(L+1)=0.0
06860      DEPS(L+2)=0.0
06870 10 CONTINUE
06880C
06890      DO 20 I=1,NN
06900      L=2*(I-1)+1
06910      AL1(L)=0.0
06920      AL1(L+1)=0.0
06930 20 CONTINUE
06940C
06950      RETURN
06960      END
06970C
06980C
06990      SUBROUTINE RESID(FORC,STRESS,ZN2,IEP,AL,ZRAMDA,EH,ARAMDA,
07000 &      EHNPI,ICON,IB,ZAL,B,D,DSIGM,ARAMDA,AEH,IZAL)
07010C
07020      COMMON /INP/ NN,NE,NLN,NBN,N,MS
07030      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
07040      COMMON /RR/  RA,RB,RC
07050      COMMON /GC/  HC(20),EQPSNH(20),GOC(20),GIC(20)
07060C
07070      DIMENSION FORC(900),STRESS(900),ZN2(300),IEP(300),AL(600),
07080 &      ZRAMDA(300),EH(300),ICON(900),IB(60),
07090 &      ZAL(60),B(300,3),D(300,3),ADF(3),DDF(3),
07100 &      DSIGM(900),ARAMDA(300),AEH(300),IZAL(20)
07110C
07120      DO 30 NEL=1,NE
07130      L=3*(NEL-1)+1
07140      SIG11=FORC(L)+STRESS(L)
07150      SIG22=FORC(L+1)+STRESS(L+1)
07160      SIG12=FORC(L+2)+STRESS(L+2)
07170      ZETO=-SIG11*SIG11+SIG22*SIG22-SIG11*SIG22+3.0*(SIG12*SIG12)

```

```

07180      GN2EP=ZN2(NEL)
07190      IF(ZETO-GN2EP) 20,10,10
07200C
07210      10 IEP(NEL)=1
07220      GO TO 30
07230      20 IEP(NEL)=0
07240      30 CONTINUE
07250C
07260      DO 40 L=1,NN
07270      N1=2*(L-1)+1
07280      AL(N1)=0.0
07290      AL(N1+1)=0.0
07300      40 CONTINUE
07310C
07320      DO 50 IQ=1,NE
07330      IQ=IQ1
07340      ID=IEP(IQ)
07350      IF(ID.EQ.0) GO TO 50
07360      L=3*(IQ-1)+1
07370      ZR=ZRAMDA(IQ1)
07380C
07390      AP=1.0+ZR*RA
07400      BP=ZR*RB
07410      CP=1.0+ZR*RC
07420      AA=AP/(AP*AP-BP*BP)
07430      BB=BP/(BP*BP-AP*AP)
07440      CC=1.0/CP
07450      ADF(1)=AA*FORC(L)+BB*FORC(L+1)
07460      ADF(2)=BB*FORC(L)+AA*FORC(L+1)
07470      ADF(3)=CC*FORC(L+2)
07480C
07490      CALL PFORC(IQ,FORC,STRESS,EH,DDF,RAMDA,EHNP1)
07500C
07510      L=3*(IQ-1)+1
07520      DSIGM(L)=DDF(1)
07530      DSIGM(L+1)=DDF(2)
07540      DSIGM(L+2)=DDF(3)
07550      ARAMDA(IQ)=RAMDA
07560      AEH(IQ)=EHNP1
07570C
07580      NN1=2*(ICON(L)-1)+1
07590      NN2=2*(ICON(L+1)-1)+1
07600      NN3=2*(ICON(L+2)-1)+1
07610C
07620      AD1=ADF(1)-DDF(1)
07630      AD2=ADF(2)-DDF(2)
07640      AD3=ADF(3)-DDF(3)
07650C
07660      RX1=(AD1*B(IQ,1)+AD3*D(IQ,1))*0.5
07670      RY1=(AD3*B(IQ,1)+AD2*D(IQ,1))*0.5
07680      RX2=(AD1*B(IQ,2)+AD3*D(IQ,2))*0.5
07690      RY2=(AD3*B(IQ,2)+AD2*D(IQ,2))*0.5
07700      RX3=(AD1*B(IQ,3)+AD3*D(IQ,3))*0.5
07710      RY3=(AD3*B(IQ,3)+AD2*D(IQ,3))*0.5
07720C
07730      AL(NN1)=RX1+AL(NN1)
07740      AL(NN1+1)=RY1+AL(NN1+1)
07750      AL(NN2)=RX2+AL(NN2)
07760      AL(NN2+1)=RY2+AL(NN2+1)
07770      AL(NN3)=RX3+AL(NN3)

```

```

07780     AL(NN3+1)=RY3+AL(NN3+1)
07790     50 CONTINUE
07800C
07810     DO 60 L=1,NBN
07820     L1=3*(L-1)+1
07830     NB=IB(L1)
07840     NB1=2*(NB-1)+1
07850     AL(NB1)=AL(NB1)*FLOAT(1B(L1+1))
07860     AL(NB1+1)=AL(NB1+1)*FLOAT(1B(L1+2))
07870     60 CONTINUE
07880C
07890     DO 70 K=1,NLN
07900     L=2*(K-1)+1
07910     I=1ZAL(K)
07920     N1=2*(I-1)+1
07930     A1=ZAL(L)
07940     A2=ZAL(L+1)
07950     AL(N1)=AL(N1)+DF*A1
07960     AL(N1+1)=AL(N1+1)+DF*A2
07970     70 CONTINUE
07980C
07990     RETURN
08000     END
08010C
08020C
08030     SUBROUTINE PFORC(NEL,FORC,STRESS,EH,DDF,RAMDA,EHNP1)
08040C
08050     COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
08060     COMMON /RR/  RA,RB,RC
08070     COMMON /GC/  HC(20),EQPSNH(20),GOC(20),G1C(20)
08080C
08090     DIMENSION DDF(3),FORC(900),STRESS(900),EH(300)
08100C
08110     L=3*(NEL-1)+1
08120     EHAT=EH(NEL)
08130     RAMDA=0.0
08140     S1N=STRESS(L)
08150     S2N=STRESS(L+1)
08160     S3N=STRESS(L+2)
08170     SA1=FORC(L)+S1N
08180     SA2=FORC(L+1)+S2N
08190     SA3=FORC(L+2)+S3N
08200C
08210     10 A=1.0+RAMDA*RA
08220     B=RAMDA*RB
08230     C=1.0+RAMDA*RC
08240     AA=A/(A*A-B*B)
08250     BB=B/(B*B-A*A)
08260     CC=1.0/C
08270     SMA1=AA*SA1+BB*SA2
08280     SMA2=BB*SA1+AA*SA2
08290     SMA3=CC*SA3
08300     FSIGM=SMA1*SMA1+SMA2*SMA2-SMA1*SMA2+3.0*SMA3*SMA3
08310     FSIGM=SQRT(FSIGM)
08320     A=RAMDA*FSIGM
08330     I=2
08340     20 B=EQPSNH(I)
08350     IF(EHAT.LT.B) GOTO 30
08360     I=I+1
08370     GOTO 20

```

```

08380 30 J=1
08390 40 EQK=EQPSNH(J)
08400 X=EQK-EHAT
08410 GK=G1C(J)
08420 Y=A*GK
08430 IF(X.GT.Y) GOTO 50
08440 J=J+1
08450 GOTO 40
08460 50 GK1=G1C(J-1)
08470 EQK1=EQPSNH(J-1)
08480 CD=(GK1-GK)/(EQK-EQK1)
08490 X=EHAT+A*(GK1+CD*EQK1)
08500 X=X/(1.0+A*CD)
08510 XEQ=X-EQK1
08520 G1CRM=GK1-CD*XEQ
08530 G2CRM=-CD
08540 GEHP=GOC(J-1)+(GK1-0.5*CD*XEQ)*XEQ
08550 FRAM=FSIGM-GEHP
08560 SSIG1=SMA1-0.5*SMA2
08570 SSIG2=-0.5*SMA1+SMA2
08580 SSIG3=3.0*SMA3
08590 A=AA*SSIG1+BB*SSIG2
08600 B=BB*SSIG1+AA*SSIG2
08610 C=CC*SSIG3
08620 SDA=RA*A+RB*B
08630 SDB=RB*A+RA*B
08640 SDC=RC*C
08650 DFDRAM=- (SDA*SMA1+SDB*SMA2+SDC*SMA3)/FSIGM
08660 DXDRAM=(A*SMA1+B*SMA2+C*SMA3)/FSIGM
08670 DXDRAM=DXDRAM*G1CRM/(1.0-RAMDA*FSIGM*G2CRM)
08680 FCR=DFDRAM-G1CRM*DXDRAM
08690 RAMDA=RAMDA-FRAM/FCR
08700 D=ETA*FSIGM
08710 IF(FRAM.GT.D) GOTO 10
08720C
08730 EHNP1=RAMDA*FSIGM*G1CRM+EHAT
08740 DDF(1)=SMA1-S1N
08750 DDF(2)=SMA2-S2N
08760 DDF(3)=SMA3-S3N
08770C
08780 RETURN
08790 END
08800C
08810C
08820 SUBROUTINE CHECK(AL,AL1,RSD,RDEL)
08830C
08840 COMMON /INP/ NN,NE,NLN,NBN,N,MS
08850 COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
08860C
08870 DIMENSION AL(600),AL1(600)
08880C
08890 RSD=0.0
08900 RSE=0.0
08910C
08920 DO 10 I=1,NN
08930 N1=2*(I-1)+1
08940 ALN1=AL(N1)
08950 ALN11=AL1(N1)
08960 ALN2=AL(N1+1)
08970 ALN12=AL1(N1+1)

```

```

08980      AB=ABS(ALN1-ALN11)+ABS(ALN2-ALN12)
08990      IF (AB.GE.RSD) RSD=AB
09000      AB=ABS(ALN1)+ABS(ALN2)
09010      IF (AB.GE.RSE) RSE=AB
09020      10 CONTINUE
09030C
09040      RDEL=RSE*DELTA
09050C
09060      RETURN
09070      END
09080C
09090C
09100      SUBROUTINE SIGM(UN,AL,IEP,FORC,STRESS,EH,ZRAMDA,DEPS,
09110      & STRAIN,EEQ,ZAL,DSIGM,ARAMDA,AEH)
09120C
09130      COMMON /INP/ NN,NE,NLN,NBN,N,MS
09140      COMMON /OUT/ IN,IO,IS1,IS2,IS3,IS4,ISTAT
09150      COMMON /PRO/ E,ANU,ZO,ETA,DELTA,DF,IFINAL
09160      COMMON /ELT/ NUC,FX,FY
09170C
09180      DIMENSION FSO(300),UN(600),AL(600),IEP(300),
09190      & FORC(900),STRESS(900),EH(300),ZRAMDA(300),
09200      & DEPS(900),STRAIN(900),EEQ(300),ZAL(60),
09210      & DSIGM(900),ARAMDA(300),AEH(300)
09220C
09230      DO 10 I=1,NN
09240      NI=2*(I-1)+1
09250      UN(NI)=UN(NI)+AL(NI)
09260      UN(NI+1)=UN(NI+1)+AL(NI+1)
09270      10 CONTINUE
09280C
09290      DO 50 I=1,NE
09300      NEL=I
09310      NI=3*(NEL-1)+1
09320      INDEX=IEP(NEL)
09330      IF (INDEX) 30,30,20
09340C
09350      20 STRESS(NI)=STRESS(NI)+DSIGM(NI)
09360      STRESS(NI+1)=STRESS(NI+1)+DSIGM(NI+1)
09370      STRESS(NI+2)=STRESS(NI+2)+DSIGM(NI+2)
09380      STRAIN(NI)=STRAIN(NI)+DEPS(NI)
09390      STRAIN(NI+1)=STRAIN(NI+1)+DEPS(NI+1)
09400      STRAIN(NI+2)=STRAIN(NI+2)+DEPS(NI+2)
09410C
09420      ZRAMDA(NEL)=ARAMDA(NEL)
09430      EHN=EH(NEL)
09440      EHN1=AEH(NEL)
09450      DEPN=EHN1-EHN
09460      EH(NEL)=EHN1
09470      GC=G1CR(EHN)
09480      GC1=G1CR(EHN1)
09490      EEQ(I)=EEQ(I)+0.5*(1.0/GC1+1.0/GC)*DEPN
09500      GO TO 40
09510C
09520      30 STRESS(NI)=STRESS(NI)+FORC(NI)
09530      STRESS(NI+1)=STRESS(NI+1)+FORC(NI+1)
09540      STRESS(NI+2)=STRESS(NI+2)+FORC(NI+2)
09550      STRAIN(NI)=STRAIN(NI)+DEPS(NI)
09560      STRAIN(NI+1)=STRAIN(NI+1)+DEPS(NI+1)
09570      STRAIN(NI+2)=STRAIN(NI+2)+DEPS(NI+2)

```

```

09580      ZRAMDA(NEL)=0.0
09590C
09600      40 FSO(1)=SQRT(STRESS(N1)*STRESS(N1)+STRESS(N1+1)*
09610      &          STRESS(N1+1)-STRESS(N1)*STRESS(N1+1)+
09620      &          3.0*(STRESS(N1+2)*STRESS(N1+2)))
09630C
09640      50 CONTINUE
09650C
09660      DO 60 K=1,NLN
09670      N1=2*(K-1)+1
09680      FX=FX+ZAL(N1)*DF
09690      FY=FY+ZAL(N1+1)*DF
09700      60 CONTINUE
09710C
09720      CALL OUTPUT (STRESS,STRAIN,ZRAMDA,FSO,UN,IEP,EEQ)
09730C
09740      RETURN
09750      END
09760C
09770C
09780      SUBROUTINE OUTPUT (STRESS,STRAIN,ZRAMDA,FSO,UN,IEP,EEQ)
09790C
09800      COMMON /INP/ NN,NE,NLN,NBN,N,MS
09810      COMMON /ELT/ NUE,FX,FY
09820      COMMON /OUT/ IN,IO,IS1,IS2,IS3,IS4,ISTAT
09830C
09840      DIMENSION STRESS(900),STRAIN(900),ZRAMDA(300),
09850      &          FSO(300),UN(600),IEP(300),EEQ(300)
09860C
09870      IF(ISTAT) 10,10,30
09880C
09890      10 WRITE(10,20) FX,FY
09900      20 FORMAT(///2X,"***** ELASTIC SOLUTION *****"
09910      &          ///35X,"X - LOAD = ",1PE12.6,14X,"Y - LOAD = ",1PE12.6)
09920      GOTO 40
09930      30 WRITE(10,50) ISTAT,FX,FY,NUE
09940      WRITE(10,60)(IEP(I),I=1,NE)
09950      40 WRITE(10,70)
09960      WRITE(10,80) (1,UN(2*(I-1)+1),UN(2*(I-1)+2),I=1S1,IS2)
09970      WRITE(10,90)
09980      WRITE(10,100) (K,STRESS(3*(K-1)+1),STRESS(3*(K-1)+2),
09990      &          STRESS(3*(K-1)+3),FSO(K),
10000      &          STRAIN(3*(K-1)+1),STRAIN(3*(K-1)+2),
10010      &          STRAIN(3*(K-1)+3),EEQ(K),
10020      &          ZRAMDA(K),K=1S3,IS4)
10030      WRITE(10,110)
10040C
10050      50 FORMAT(///2X,"***** STEP ",I3," *****"
10060      &          ///35X,"X - LOAD = ",1PE12.6,14X,"Y - LOAD = ",1PE12.6
10070      &          ///51X,"ITERATION NUMBER = ",I7//)
10080      60 FORMAT(13X,5I3,2X,5I3,2X,5I3,2X,5I3,2X,5I3,2X,5I3)
10090      70 FORMAT(///53X," DISPLACEMENT "
10100      &          ///15X,"NODE",10X,"X",15X,"Y",30X,
10110      &          "NODE",10X,"X",15X,"Y",//)
10120      80 FORMAT(15X,14,4X,1PE12.5,4X,1PE12.5,25X,
10130      &          14,4X,1PE12.5,4X,1PE12.5)
10140      90 FORMAT(///53X," STRESS AND STRAIN "
10150      &          ///2X,"ELEMENT NO. ",6X,"SX",11X,"SY",10X,"SXY"
10160      &          10X,"SEQ",11X,"EX",11X,"EY",10X,"EXY",
10170      &          10X,"EEQ",9X,"RAMDA"//)

```

```

10180 100 FORMAT(5X,14,6X,1PE11.4,2X,1PE11.4,2X,1PE11.4,2X,1PE11.4,
10190 &          2X,1PE11.4,2X,1PE11.4,2X,1PE11.4,2X,1PE11.4,
10200 &          2X,1PE11.4)
10210 110 FORMAT(///5X,12("-----"))//
10220C
10230      RETURN
10240      END
    
```

Table 1. An example of input data

0010	5	4	2	4	21000.0	0.3		
0020		32.0		4.0	0.1E-2	0.01	8	20
0030	1		0.0		32.000			
0040	2		0.01		37.000			
0050	3		0.025		41.600			
0060	4		0.045		46.400			
0070	5		0.070		50.900			
0080	6		0.100		55.400			
0090	7		0.130		58.658			
0100	8		0.195		61.908			
0110	1		0.0		0.0			
0120	2		0.0		5.0			
0130	3		2.5		2.5			
0140	4		5.0		0.0			
0150	5		5.0		5.0			
0160	1	1	3	2				
0170	2	1	4	3				
0180	3	3	5	2				
0190	4	3	4	5				
0200	1			0				
0210	2			0				
0220	4			0				
0230	5			0				
0240	2			0.0	1.0			
0250	5			0.0	1.0			
0260	1			4				
0270	1			2				

