JAERI-Data/Code
96-025

CRUSH1:
A SIMPLIFIED COMPUTER PROGRAM FOR IMPACT ANALYSIS OF RADIOACTIVE MATERIAL TRANSPORT CASKS

July 1996

Takeshi IKUSHIMA

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

本レポートは，日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合わせは，日本原子力研究所研究情報部研究情報課（〒319-11 茨城県那珂郡東海村）あて，お申し越しください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1996

編集兼発行　日本原子力研究所
印　　刷　㈱原子力資料サービス

CRUSH1: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks

Takeshi IKUSHIMA

Department of Fuel Cycle Safety Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received June 14, 1996)

In drop impact analyses for radioactive transport casks, it has become possible to perform them in detail by using interaction evaluation, computer programs, such as DYNA2D, DYNA3D, PISCES and HONDO. However, the considerable cost and computer time are necessitated to perform analyses by these programs. To meet the above requirements, a simplified computer program CRUSH1 has been developed. The CRUSH1 is a static calculation computer program capable of evaluating the maximum acceleration of cask bodies and the maximum deformation of shock absorbers using an Uniaxial Displacement Method (UDM).

The CRUSH1 is a revised version of the CRUSH. Main revisions of the computer program are as follows;

(1) not only main frame computer but also work stations (OS UNIX) and personal computer (OS Windows 3.1 or Windows NT) are available for use of the CRUSH1 and

(2) input data set are revised.

In the paper, brief illustration of calculation method using UDM is presented. The second section presents comparisons between UDM and the detailed method. The third section provides a use's guide for CRUSH1.

Keywords: Computer Program, Impact Analysis, Cask, Drop Impact, UDM, Transport Cask, Structural Analysis, Static Analysis

JAERI－Data/Code　96－025

ＣＲＵＳＨ１：放射性物質輸送容器の簡易衝突計算プログラム

日本原子力研究所東海研究所燃料サイクル安全工学部
幾島　毅

（1996年６月14日受理）

放射性物質輸送容器の落下衝突解析では，ＤＹＮＡ２Ｄ，ＤＹＮＡ３Ｄ，ＰＩＳＣＥＳおよびＨＯＮＤＯのような詳細計算プログラムを用いて計算されている。しかし，これらの計算プログラムによる計算は，多くの計算費用と計算時間が必要とされる。このような背景から，簡易計算プログラムＣＲＵＳＨ１を開発した。ＣＲＵＳＨ１は１次元変形法（ＵＤＭ法）を用いた静的計算プログラムであり，輸送容器本体の最大加速度およびショックアブソーバの最大変形量を計算するものである。

ＣＲＵＳＨ１はＣＲＵＳＨの改良版であり，主要な改良点は下記の通りである。

⑴　大型計算機以外にもワークステーション（OS UNIX）およびパーソナルコンピュータ（OS Windows 3.1またはWindows NT）によって使用できるプログラムが用意されている。

⑵　入力データの一部が変更された。

本報告書はＵＤＭ法の説明，ＵＤＭ法と詳細計算法とによる計算結果の比較およびＣＲＵＳＨ１の使用のための，入力データ等のユーザーガイドについても記述した。

東海研究所：〒319-11　茨城県那珂郡東海村白方白根２の４

Contents

目　　次

1. Introduction

In the drop impact analyses for radioactive transport casks, it has become possible to perform them in detail by using interaction evaluation computer programs, such as DYNA2D, DYNA3D, PISCES and HONDO. However, the considerable cost and computer time are necessitated to perform analyses by these programs. To meet the above requirements, a simplified computer program CRUSH1 as shown in Fig. 1.1 has been developed. The CRUSH1 is a static calculation computer program capable of evaluating the maximum acceleration of cask bodies and the maximum deformation of shock absorbers using a Uniaxial Displacement Method(UDM)(1) as shown in Fig. 1.2.

CRUSH1 is a revised version of CRUSH(2). Main revisions of the computer program are as follows;

(1) not only main frame computer but also work stations(OS UNIX) and personal computer(OS DOS or Windows) are available for use of CRUSH1 and

(2) input data set are revised.

In the paper, brief illustration of calculation method using UDM is presented. The second section presents comparisons between UDM and the detailed method. The third section provides a use's guide for CRUSH1.

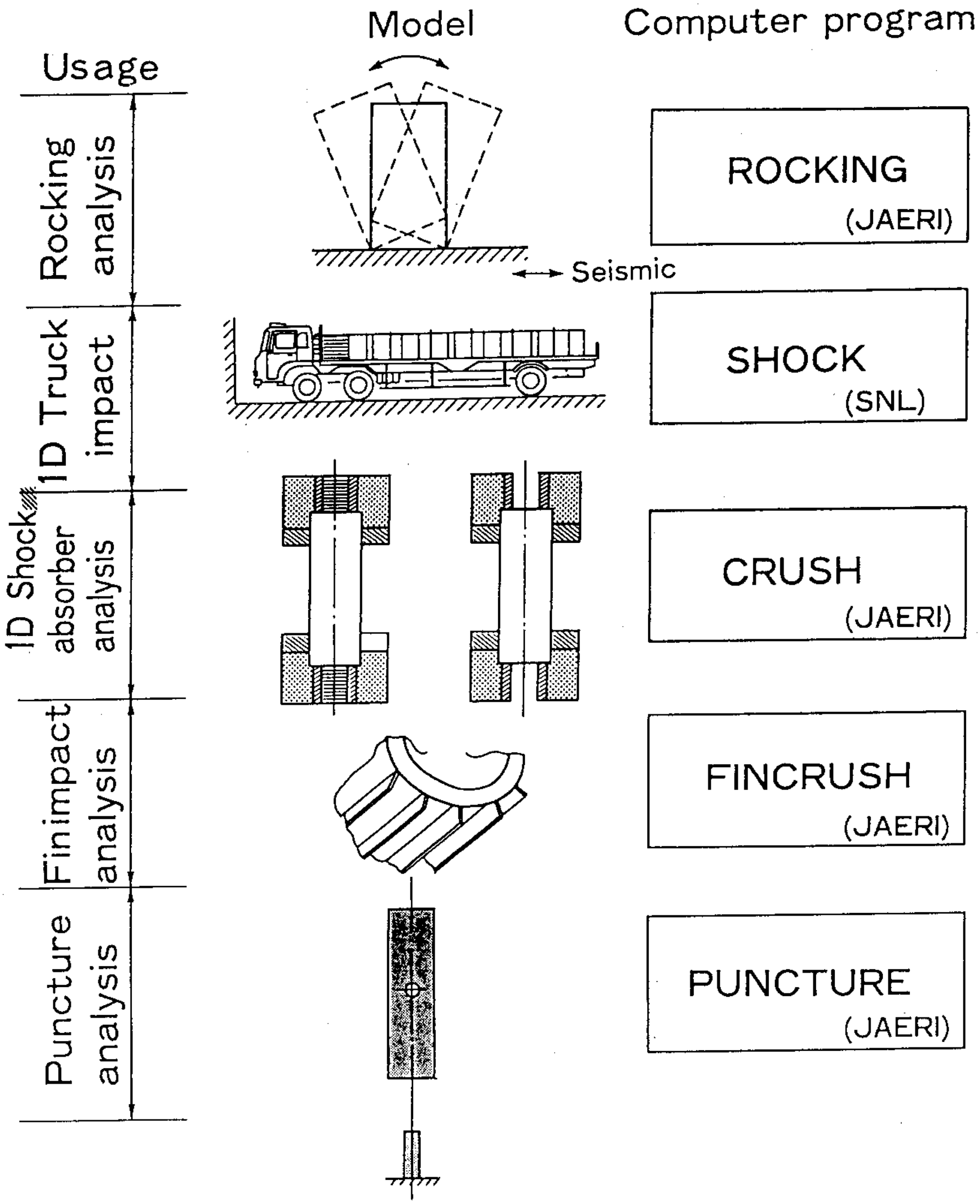
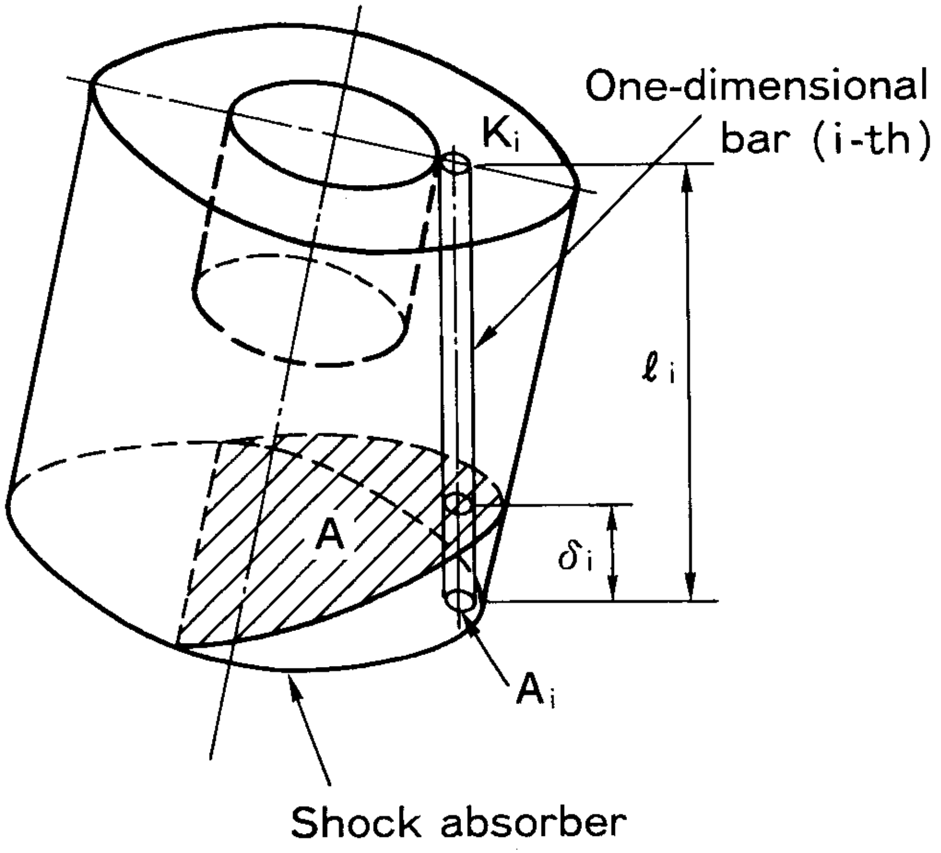


Fig.1.1 Simplified analysis computer programs



Strain : ε

$$\varepsilon_i = \frac{\delta_i}{\ell_i}$$

Force : F_i

$$F_i = K_i \sigma_i(\varepsilon_i) A_i$$

Total force : F

$$F = \Sigma_i F_i$$

Acceleration : α

$$\alpha = \frac{F}{M}$$

M : Mass,　A : Area
σ : Stress,　K : Boundary condition constant

Fig.1.2　Uni-axial displacement method

2. Calculation equation

2.1 Calculation model

In the modeling of shock absorber, it is assumed that the shock absorber consists of three or four species of material as shown in Figs. 2.1 to 2.3. In the figures, the symbols of K_i(i=1∼5) indicates boundary condition constants which are estimated by an overpack stiffness and boundary condition of the shock absorber.

When the shock absorber deforms by the displacement $\Delta \ell$ in a corner drop as shown in Fig.2.4, the strain ε of an one-dimensional bar is

$$\varepsilon = \frac{\Delta \ell}{\ell} . \tag{2.1}$$

The force f of the one-dimensional bar is

$$f = K\,\sigma(\varepsilon)\,\Delta A . \tag{2.2}$$

where ℓ, σ and ΔA are the length, stress and area of the one-dimensional bar. K is the boundary condition constant. The total force F of the shock absorber is

$$F = \sum_i f_i . \tag{2.3}$$

The dissipated energy E(δ) can also obtained using equation similar to above Eq.(2.3)

$$E(\delta) = \int_0^{\delta} F\,d\ell . \tag{2.4}$$

Therefore, when a cask whose weight of W is dropped from a height H with an oblique angle θ, the maximum displacement of the shock absorber δ and the maximum acceleration of the cask body α are given as follows.

$$E(\delta) = \gamma \cdot W \cdot H \quad , \tag{2.5}$$

$$\alpha = \frac{F(\delta)}{M} \quad , \tag{2.6}$$

where γ is the ratio of the energy absorbed in the primary impact to the total energy absorbed in the primary and secondary impacts. M is mass of the cask.

2.2 Vertical drop impact

In the modeling a shock absorber in CRUSH, it is assumed that the shock absorber consists of three or four species of shock absorber materials as shown in Fig. 2.5. In the figure, the symbols of σ_A, σ_B, σ_C, σ_D indicate their materials determined by the geometries of the cask and the shock absorber.

When the shock absorber deforms by a displacement Z in a vertical drop, the impact load F and the dissipated energy E are given by following equations.

$$F = F_{01} + F_{12} + F_{23} + F_{34} + F_{45} \quad , \tag{2.7}$$

$$E = \int_0^Z F \, dZ \quad , \tag{2.8}$$

where, for the region $0 \sim R_1$

$$F_{01} = K_1 \, \pi \, \sigma_D\left(\frac{Z}{Z_2}\right) R_1^2 \quad , \tag{2.9}$$

for the region $R_1 \sim R_2$

$$F_{12} = K_2 \, \pi \, \sigma_A\left(\frac{Z}{Z_2}\right)(R_2^2 - R_1^2) \quad , \tag{2.10}$$

for the region $R_2 \sim R_3$

$$F_{23} = K_3 \, \pi \, \sigma_B\left(\frac{Z}{Z_2}\right)(R_3^2 - R_1^2) \quad , \tag{2.11}$$

for the region $R_3 \sim R_4$

$$\left.\begin{aligned} F_{34} &= K_4 \pi \sigma_B \left(\frac{\Delta Z_B}{Z_3}\right)(R_4^2 - R_3^2)\,;\ (\text{for } \sigma_B \text{ material})\,, \\ F_{34} &= K_4 \pi \sigma_C \left(\frac{\Delta Z_C}{Z_4 - Z_3}\right)(R_4^2 - R_3^2)\,;\ (\text{for } \sigma_C \text{ material})\,, \end{aligned}\right\} \quad (2.12)$$

$$\left.\begin{aligned} Z &= \Delta Z_B + \Delta Z_C\,, \\ \sigma_B\left(\frac{\Delta Z_B}{Z_3}\right) &= \sigma_C\left(\frac{\Delta Z_C}{Z_4 - Z_3}\right)\,, \end{aligned}\right\} \quad (2.13)$$

for the region $R_4 \sim R_5$

$$\left.\begin{aligned} F_{45} &= K_5 \pi \sigma_B \left(\frac{\Delta Z_B}{Z_3}\right)(R_5^2 - R_4^2)\,;\ (\text{for } \sigma_B \text{ material})\,, \\ F_{45} &= K_5 \pi \sigma_C \left(\frac{\Delta Z_C}{Z_4 - Z_3}\right)(R_5^2 - R_4^2)\,;\ (\text{for } \sigma_C \text{ material})\,, \end{aligned}\right\} \quad (2.14)$$

$$\left.\begin{aligned} Z &= \Delta Z_B + \Delta Z_C\,, \\ \sigma_B\left(\frac{\Delta Z_B}{Z_3}\right) &= \sigma_C\left(\frac{\Delta Z_C}{Z_4 - Z_3}\right)\,, \end{aligned}\right\} \quad (2.15)$$

$$0 \leqq K_i \leqq 1\,.$$

$\sigma_i (i = A, B, C, D)$ are stress in these material regions.

2.3 Horizontal drop impact

When the shock absorber deforms by the displacement Z in a horizontal drop as shown in Figs. 2.6 to 2.8, the impact load and the dissipated energy are given by following equations.

$$F = \sum F_{ij} = F_{01} + F_{12} + F_{23} + F_{34} + F_{45} + F_{56} + F_{67} + F_{78} + F_{89}\,, \qquad (2.16)$$

$$F_{ij} = 2 \int_0^{R_5} \Delta F \, dx\,, \qquad (2.17)$$

$$E = \int_0^y F \, dy\,, \qquad (2.18)$$

where, for the region $0 \sim Z_1$ and $0 \sim R_1$

$$\left.\begin{aligned}
\Delta F &= K_1\ \sigma_D\left(\frac{\Delta y_1}{y_1}\right)dx\cdot Z_1;\ (\text{for } \sigma_D \text{ material})\ ,\\
\Delta F &= K_1\ \sigma_A\left(\frac{\Delta y_2}{y_2}\right)dx\cdot Z_1;\ (\text{for } \sigma_A \text{ material})\ ,\\
\Delta F &= K_1\ \sigma_B\left(\frac{\Delta y_3}{y_3}\right)dx\cdot Z_1;\ (\text{for } \sigma_B \text{ material})\ ,
\end{aligned}\right\} \tag{2.19}$$

$$\left.\begin{aligned}
y_1 &= 2\sqrt{R_1{}^2-x^2}\ ,\\
y_2 &= 2(\sqrt{R_2{}^2-x^2} - \sqrt{R_1{}^2-x^2})\ ,\\
y_3 &= 2(\sqrt{R_5{}^2-x^2} - \sqrt{R_2{}^2-x^2})\ .
\end{aligned}\right\} \tag{2.20}$$

$$\left.\begin{aligned}
&U = \Delta y_1 + \Delta y_2 + \Delta y_3\ ,\\
&\sigma_D\left(\frac{\Delta y_1}{y_1}\right) = \sigma_A\left(\frac{\Delta y_2}{y_2}\right) = \sigma_B\left(\frac{\Delta y_3}{y_3}\right)\ .
\end{aligned}\right\} \tag{2.21}$$

$$\left.\begin{aligned}
&U = y - h;\ \left\{\begin{matrix} U \leqq 0, & \text{for } \Delta F = 0 \\ U > 0, & \text{for } \Delta F \neq 0 \end{matrix}\right\}\ ,\\
&h = R_5 - \sqrt{R_5{}^2-x^2}\ .
\end{aligned}\right\} \tag{2.22}$$

For the region $0\sim Z_1$ and $R_1\sim R_2$

$$\left.\begin{aligned}
\Delta F &= K_1\ \sigma_A\left(\frac{\Delta y_1}{y_1}\right)dx\cdot Z_1;\ (\text{for } \sigma_A \text{ material})\ ,\\
\Delta F &= K_1\ \sigma_B\left(\frac{\Delta y_2}{y_2}\right)dx\cdot Z_1;\ (\text{for } \sigma_B \text{ material})\ .
\end{aligned}\right\} \tag{2.23}$$

$$y_1 = 2\sqrt{R_2{}^2-x^2}$$

$$y = 2(\sqrt{R_5{}^2-x^2} - \sqrt{R_2{}^2-x^2})\ .$$

$$U = \Delta y_1 + \Delta y_2\ ,$$

$$\sigma_A\left(\frac{\Delta y_1}{y_1}\right) = \sigma_B\left(\frac{\Delta y_2}{y_2}\right)\ . \tag{2.25}$$

For the region $0\sim Z_1$ and $R_2\sim R_3$

$$\Delta F = K_1\ \sigma_B\left(\frac{U}{y}\right)dx\cdot Z_1\ , \tag{2.26}$$

$$y = 2\sqrt{R_5{}^2-x^2}\ . \tag{2.27}$$

For the region $Z_1 \sim Z_2$ and $0 \sim R_5$ equations are similar as eqs.(2.9) through (2.25) when Z_1 changes to $(Z_2 - Z_1)$ and K_1 to K_4.

For the region $Z_2 \sim Z_3$ and $0 \sim R_3$

$$\Delta F = K_3\ \sigma_B\left(\frac{U}{y}\right) dx \cdot (Z_3 - Z_2)\ , \tag{2.28}$$

where

$$y = \sqrt{r_5^2 - x^2} - \sqrt{R_3^2 - x^2}\ . \tag{2.29}$$

For the region $Z_2 \sim Z_3$ and $R_3 \sim R_5$

$$\Delta F = K_3\ \sigma_B\left(\frac{U}{y}\right) dx \cdot (Z_3 - Z_2)\ , \tag{2.30}$$

where

$$y = 2\sqrt{R_5^2 - x^2\ .} \tag{2.31}$$

For the region $Z_3 \sim Z_4$ and $0 \sim R_3$

$$\Delta F = K_4\ \sigma_C\left(\frac{U}{y}\right) dx \cdot (Z_3 - Z_2)\ , \tag{2.32}$$

where

$$y = \sqrt{R_5^2 - x^2} - \sqrt{R_3^2 - x^2}\ . \tag{2.33}$$

For the region $Z_3 \sim Z_4$ and $R_3 \sim R_5$

$$\Delta F = K_4\ \sigma_C\left(\frac{U}{y}\right) dx \cdot (Z_4 - Z_3)\ , \tag{2.34}$$

where

$$y = 2\sqrt{R_5^2 - x^2\ .} \tag{2.35}$$

For the region $Z_5 \sim Z_6$ equations are same as the region $Z_3 \sim Z_4$, the region $Z_6 \sim Z_7$ same as the region $Z_2 \sim Z_3$, the region $Z_7 \sim Z_8$ same as the region $Z_1 \sim Z_2$ and the region $Z_8 \sim Z_9$ same as the region $0 \sim Z_1$.

2.4 Oblique drop impact

In the modeling a shock absorber in CRUSH at a oblique drop impact, it is assumed that shock absorber consists of one specy of shock absorber material as shown in Fig. 2.3. The stress of a wooden shock absorber can be written as following, including the effect of the wood grain angle θ.

$$\sigma_X = \sigma_A \cos^2\theta + \sigma_B \sin^2\theta \quad , \tag{2.36}$$

where σ_A, σ_B and σ_X are stresses in the wood whose grain direction is parallel, perpendicular and angle θ degree to the drop direction, respectively.

Let consider a cutway section of a shock absorber as shown in Figs. 2.9, 2.10 and 2.11. The impact load and the dissipated energy are given by following equations (see Fig. 2.11).

$$F_i = 2\int_0^{x_M} \Delta F \, dx \quad , \tag{2.37}$$

$$F = \int_S F_i \, dS \; , \tag{2.38}$$

$$E = \int_0^{y} F \, dy \quad , \tag{2.39}$$

$$U = y - h_0 - h; \quad \left\{ \begin{array}{ll} U \leqq 0, & \text{for } \Delta F = 0 \\ U > 0, & \text{for } \Delta F \neq 0 \end{array} \right\} \quad . \tag{2.40}$$

Sectional figures of the shock absorber in the case of oblique drop are shown in Figs. 2.12 and 2.13.

Let consider S-T coordinate as shown in Fig. 2.14.

The coordinates of points P_1~P_{12} and Q_1~Q_4 are as follows.

$$\left. \begin{array}{lllll} P_1 & \{ & -R_5\cos\theta & , & R_5\sin\theta \; \} \\ P_2 & \{ & -R_1\cos\theta & , & R_1\sin\theta \; \} \\ P_3 & \{ & R_1\cos\theta & , & -R_1\sin\theta \; \} \\ P_4 & \{ & R_5\cos\theta & , & -R_5\sin\theta \; \} \\ P_5 & \{ & -R_3\cos\theta + Z_2\sin\theta & , & R_3 cin\theta + Z_2\cos\theta \} \\ P_6 & \{ & -R_1\cos\theta + Z_2\sin\theta & , & R_1\sin\theta + Z_2\cos\theta \} \\ P_7 & \{ & R_1\cos\theta + Z_2\sin\theta & , & -R_1\sin\theta + Z_2\cos\theta \} \\ P_8 & \{ & R_3\cos\theta + Z_2\sin\theta & , & -R_3\sin\theta + Z_2\cos\theta \} \\ P_9 & \{ & -R_5\cos\theta + Z_4\sin\theta & , & R_5\sin\theta + Z_4\cos\theta \} \\ P_{10} & \{ & -R_3\cos\theta + Z_4\sin\theta & , & R_3\sin\theta + Z_4\cos\theta \} \end{array} \right\} \tag{2.41}$$

$$\left.\begin{array}{lll}
P_{11} & \{\ R_3\cos\theta + Z_4\sin\theta, & -R_3\sin\theta + Z_4\cos\theta\} \\
P_{12} & \{\ R_5\cos\theta + Z_4\sin\theta, & -R_5\sin\theta + Z_4\cos\theta\} \\
Q_2 & \{-R_1\cos\theta + Z_1\sin\theta, & R_1\sin\theta + Z_1\cos\theta\} \\
Q_3 & \{\ R_1\cos\theta + Z_1\sin\theta, & -R_1\sin\theta + Z_1\cos\theta\} \\
Q_0 & \{\qquad Z_2\sin\theta \qquad , & \qquad Z_2\cos\theta \qquad\} \\
Q_4 & \{\ R_4\cos\theta + Z_4\sin\theta, & -R_4\sin\theta + Z_4\cos\theta\}
\end{array}\right|$$

The length of h_0 as shown in Figs. 2.12 and 2.13 are as follows.

When $\sigma_D \neq 0$

$$\left.\begin{array}{l}
h_0 = R_5\sin\theta - \dfrac{\sin\theta}{\cos\theta}\,S \quad ;\ (\text{region } P_1\sim P_2 \text{ and } P_3\sim P_4)\ , \\
h_0 = \dfrac{\cos\theta}{\sin\theta}\,(S - R_5\cos\theta);\ (\text{region } P_4\sim P_{12})\ .
\end{array}\right\} \qquad (2.42)$$

When $\sigma_D = 0$

$$\left.\begin{array}{l}
h_0 = R_5\sin\theta - \dfrac{\sin\theta}{\cos\theta}\,S \quad ;\ (\text{region } P_1\sim P_2 \text{ and } P_3\sim P_4)\ , \\
h_0 = \dfrac{\cos\theta}{\sin\theta}\,(S - R_5\cos\theta);\ (\text{region } P_4\sim P_{12})\ , \\
h_0 = \dfrac{\cos\theta}{\sin\theta}\,(S + R_1\cos\theta) + (R_1 + R_5)\sin\theta; \\
\qquad\qquad\qquad\qquad (\text{region}(P_2\sim Q_2\sim P_6)\ , \\
h_0 = \infty\ ;\ (\text{region } P_6\sim Q_0\sim P_7)\ .
\end{array}\right\} \qquad (2.43)$$

The lengths of h_1, ℓ_1 and ℓ_2 are as follows.

(a) If ℓ_2 is known ($S<P_5$, $S\leqq 0$, $\ell_1=0$ and $K_h=0$), h_0, h_1 and ℓ_2 are shown in Fig. 2.15(a).

$$\left.\begin{array}{l}
\ell_2 = \dfrac{R_5}{\sin\theta} \qquad\qquad\qquad ;\ (\text{region } P_1\sim P_9)\ , \\
\ell_2 = \dfrac{1}{\sin\theta\cos\theta}\,(Z_4\sin\theta - S);\ (\text{region } P_9\sim P_{10}).
\end{array}\right\} \qquad (2.44)$$

When $\sigma_D \neq 0$

$$h_1 = -\frac{S}{\sin\theta\cos\theta}\ . \qquad (2.45)$$

When $\sigma_D = 0$

$$\left.\begin{aligned} h_1 &= -\frac{S}{\sin\theta\cos\theta}; \quad (\text{region } P_1 \sim P_2)\ , \\ h_1 &= \frac{R_1}{\sin\theta} \quad ; \quad (\text{Region } P_2 \sim P_6)\ . \end{aligned}\right\} \tag{2.46}$$

(b) If ℓ_1 and ℓ_2 are known ($S<P_5$, $S>0$, $\sigma_D \neq 0$, $h_1=0$ and $K_h=0$), h_0, ℓ_1 and ℓ_2 are shown in Fig. 2.15(b).

$$\left.\begin{aligned} \ell_2 &= \frac{R_5}{\sin\theta} \quad ; \quad (\text{region } P_1 \sim P_9)\ , \\ \ell_2 &= \frac{1}{\sin\theta\cos\theta}(Z_4\sin\theta - S); \quad (\text{region } P_9 \sim P_{10})\ , \end{aligned}\right\} \tag{2.47}$$

$$\left.\begin{aligned} \ell_1 &= \frac{S}{\sin\theta\cos\theta} \quad ; \quad (S \leqq P_4,\ \text{region } P_2 \sim P_4 \sim P_{12}), \\ \ell_1 &= \frac{R_5}{\sin\theta} \quad ; \quad (S > P_4,\ \text{region } P_2 \sim P_4 \sim P_{12}), \end{aligned}\right\} \tag{2.48}$$

(c) If ℓ_2 is known ($S<P_5$, $S>0$ and $\sigma_D=0$), h_0, h_1 and ℓ_1 are shown in Fig. 2.15(c) and are same as Eqs. (2.42), (2.43) and (2.43).

(d) If ℓ_2 is known ($P_5 \leqq S \leqq Q_0$ and $K_h=1$), h_0, h_1, ℓ_1 and ℓ_2 are shown in Fig. 2.16(d).

$$\ell_2 = \frac{1}{\sin\theta\cos\theta}(Z_1\sin\theta - S)\ . \tag{2.49}$$

When $\sigma_D \neq 0$

$$\left.\begin{aligned} h_1 &= -\frac{S}{\sin\theta\cos\theta}\ , \\ \ell_1 &= 0\ , \end{aligned}\right\}; \quad (\text{for } S \leqq 0)\ , \tag{2.50}$$

$$\left.\begin{aligned} h_1 &= 0\ , \\ \ell_1 &= \frac{S}{\sin\theta\cos\theta}\ , \end{aligned}\right\}; \quad (\text{for } 0 < S \leqq P_4)\ , \tag{2.51}$$

$$\left.\begin{aligned} h_1 &= 0\ , \\ \ell_1 &= \frac{R_5}{\sin\theta}\ , \end{aligned}\right\}; \quad (\text{for } S > P_4)\ . \tag{2.52}$$

When $\sigma_D = 0$

$$\left.\begin{aligned} h_1 &= \frac{R_1}{\sin\theta} , \\ \ell_1 &= 0 , \end{aligned}\right\} ; \text{ (for } S \leqq P_6\text{)} , \tag{2.53}$$

$$\left.\begin{aligned} h_1 &= \frac{1}{\sin\theta\cos\theta}(Z_2\sin\theta - S) , \\ \ell_1 &= 0 , \end{aligned}\right\} ; \text{ (for } S > P_6\text{)} . \tag{2.54}$$

(e) If ℓ_1 is known ($Q_0 \leqq S \leqq P_1$, $\ell_2=0$ and $K_h=1$), h_1 and ℓ_1 are shown in Fig. 2.16(e).

For region $Q_0 \sim P_8$

$$h_1 = \frac{1}{\sin\theta\cos\theta}(S - Z_2\sin\theta) , \tag{2.55}$$

$$\left.\begin{aligned} \ell_1 &= \frac{S}{\sin\theta\cos\theta} ; \text{ (for } S < P_4\text{)} , \\ \ell_1 &= \frac{R_5}{\sin\theta} \quad ; \text{ (for } S \geqq P_4\text{)} . \end{aligned}\right\} \tag{2.56}$$

For region $P_8 \sim P_1$

$$h_1 = \frac{R_3}{\sin\theta} , \tag{2.57}$$

$$\left.\begin{aligned} \ell_1 &= \frac{S}{\sin\theta\cos\theta} ; \text{ (for } S < P_4\text{)} , \\ \ell_1 &= \frac{R_5}{\sin\theta} \quad ; \text{ (for } S \geqq P_4\text{)} . \end{aligned}\right\} \tag{2.58}$$

(f) If ℓ_1 is known ($S>P_1$, $\ell_2=0$ and $K_h=1$), h_1 and ℓ_1 are shown in Fig. 2.16(f).

$$h_1 = \frac{1}{\sin\theta\cos\theta}(S - Z_4\sin\theta) , \tag{2.59}$$

$$\left.\begin{aligned} \ell_1 &= \frac{S}{\sin\theta\cos\theta} ; \text{ (for } S < P_4\text{)} , \\ \ell_1 &= \frac{R_5}{\sin\theta} \quad ; \text{ (for } S \geqq P_4\text{)} . \end{aligned}\right\} \tag{2.60}$$

The sectional fogure of the shock absorber in the case of the oblique drop impact as shown in Fig. 2.17, is an ellipoid. The equation of the ellipsoid is as follows.

$$\frac{x^2}{R_5{}^2} + \frac{y^2}{b^2} = 1 \quad , \tag{2.61}$$

where

$$b = \frac{R_5}{\sin\theta} \quad . \tag{6.62}$$

Rearranged eq. (2.61), using $y=h(x_M)$

$$x_M = \sqrt{b^2-h^2} \cdot \sin\theta \quad , \tag{2.63}$$

$$y = \frac{1}{\sin\theta} \sqrt{R_5{}^2-x^2} \quad . \tag{2.64}$$

Force ΔF at the sectional area (ΔS, Δx) is as follows.

$$\Delta F = K \cdot \sigma(\frac{U}{\ell}) \Delta S \cdot \Delta x ., \tag{2.65}$$

where K is the boundary condition constant as shown in Fig. 2.18.

$$0 \leqq K_i \leqq 1 \tag{2.66}$$

where

K_1 : for $\sigma_D=0$; region P_3~Q_3,

for $\sigma_D \neq 0$; region P_1~X_1,

K_2 : for $\sigma_D=0$; region Q_3~P_7,

for $\sigma_D \neq 0$; region X_1~X_2,

K_3 : for region P_{11}~Q_4,

K_4 : for region Q_4~P_{12}.

2.5 Convergence method

The convergence methods for two or three materials in one dimensional bar in the case of the oblique drop impact, are as follows.

(1) Two materials

According to the stress-strain relation as shown in Fig. 2.19, the following equations are derived.

$$\left.\begin{aligned} \sigma_A &= \sigma_A{}^0 + K_A\ \varepsilon_A = \sigma_A{}^0 + \frac{K_A}{\ell_A} U_A \quad , \\ \sigma_B &= \sigma_B{}^0 + K_B\ \varepsilon_B = \sigma_B{}^0 + \frac{K_B}{\ell_B} U_B \quad , \end{aligned}\right\} \qquad (2.67)$$

and

$$\left.\begin{aligned} U_A + U_B &= U \quad , \\ -\sigma_A + \sigma_B &= 0 \quad , \end{aligned}\right\} \qquad (2.68)$$

from above two equations

$$\begin{bmatrix} 1 & 1 \\ -\frac{K_A}{\ell_A} & \frac{K_B}{\ell_B} \end{bmatrix} \begin{Bmatrix} U_A \\ U_B \end{Bmatrix} = \begin{Bmatrix} U \\ \sigma_A{}^0 - \sigma_B{}^0 \end{Bmatrix} \ . \qquad (2.69)$$

solve Eq. (2.69), we obtain U_A and U_B.

(2) Three materials

$$\begin{bmatrix} 1 & 1 & 1 \\ -\frac{K_A}{\ell_A} & \frac{K_B}{\ell_B} & 0 \\ -\frac{K_A}{\ell_A} & 0 & \frac{K_K}{\ell_K} \end{bmatrix} \begin{Bmatrix} U_A \\ U_B \\ U_K \end{Bmatrix} = \begin{Bmatrix} U \\ \sigma_A{}^0 - \sigma_B{}^0 \\ \sigma_A{}^0 - \sigma_K{}^0 \end{Bmatrix} \ . \qquad (2.70)$$

(3) Strain, stiffness and stress

Strain

$$\varepsilon_i = \frac{U_i}{\ell_i} \ . \qquad (2.71)$$

Stiffness (see Fig. 2.20)

$$K_i = \frac{\sigma_2 - \sigma_1}{\varepsilon_2 - \varepsilon_1} \ . \qquad (2.72)$$

Stress (see Fig. 2.20)

$$\sigma_i = \sigma_1 - K_i \cdot \varepsilon_i \ . \qquad (2.73)$$

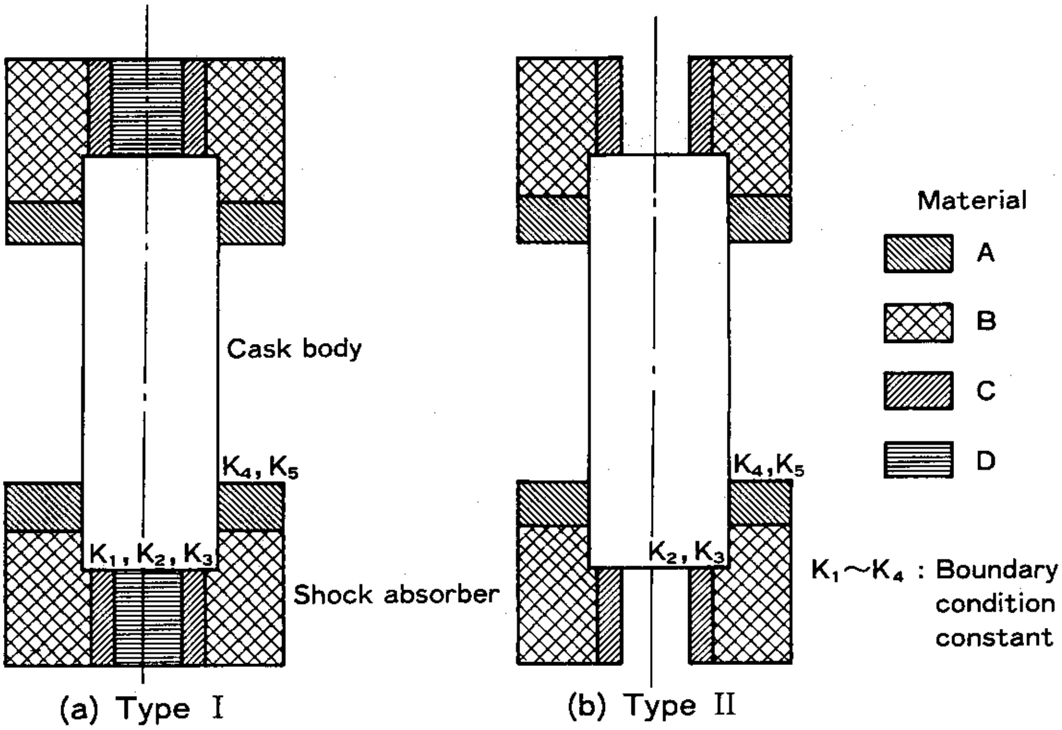


Fig.2.1 Vertical drop model

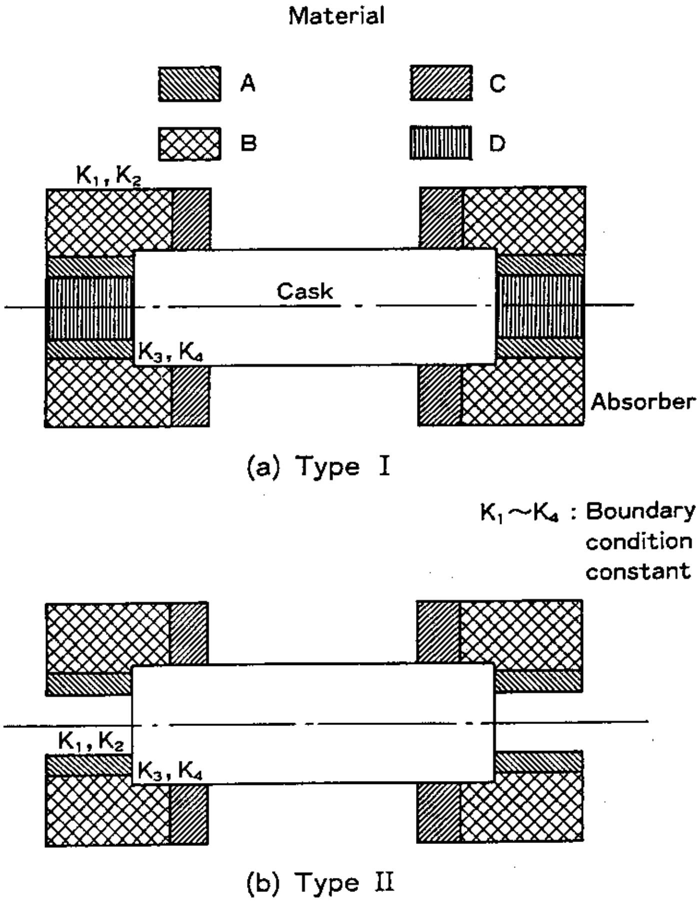
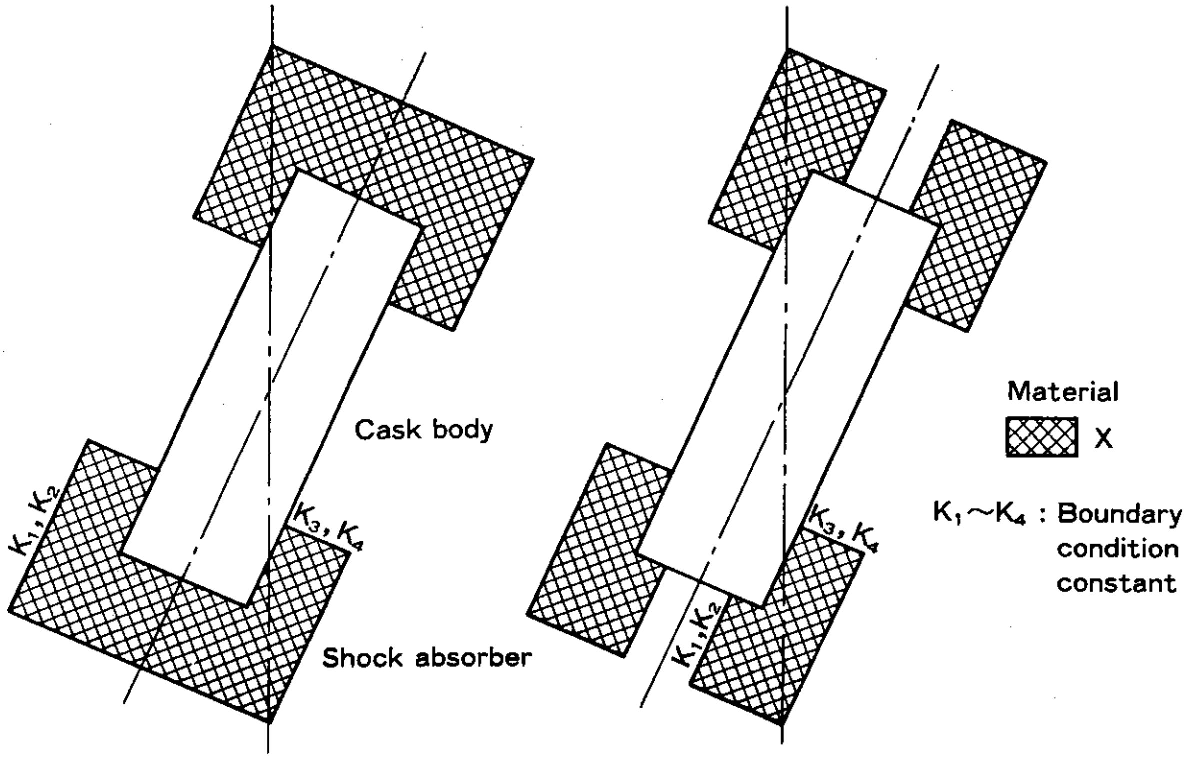
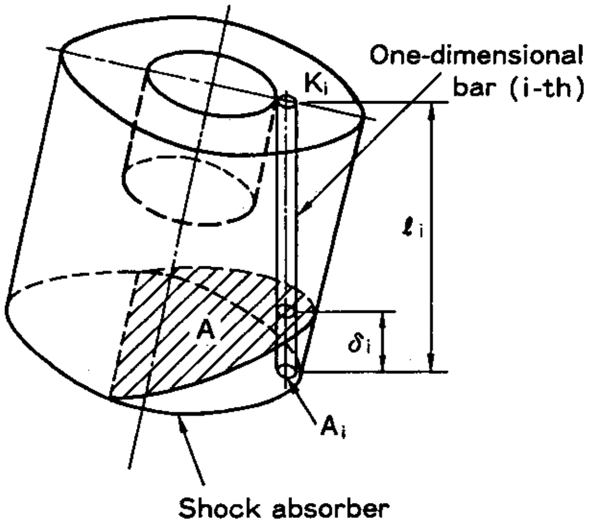


Fig.2.2 Horizontal drop model



(a) Type I (b) Type II

Fig.2.3 Oblique drop model



Strain : ε

$$\varepsilon_i = \frac{\delta_i}{\ell_i}$$

Force : F_i

$$F_i = K_i \sigma_i(\varepsilon_i) A_i$$

Total force : F

$$F = \Sigma_i F_i$$

Acceleration : α

$$\alpha = \frac{F}{M}$$

M : Mass, A : Area

σ : Stress, K : Boundary condition constant

Fig.2.4 Uni-axial displacement method

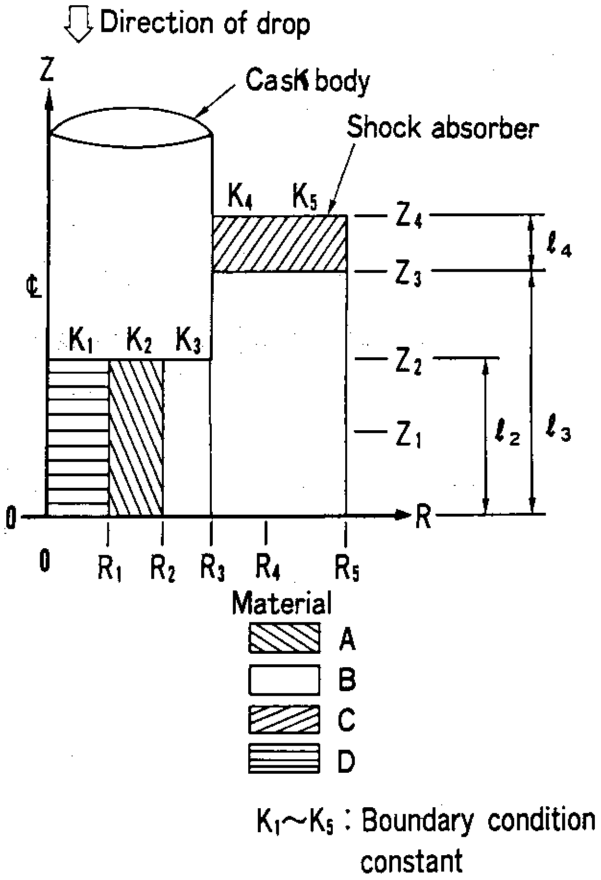


Fig.2.5 Geometry and material in the case of vertical drop model

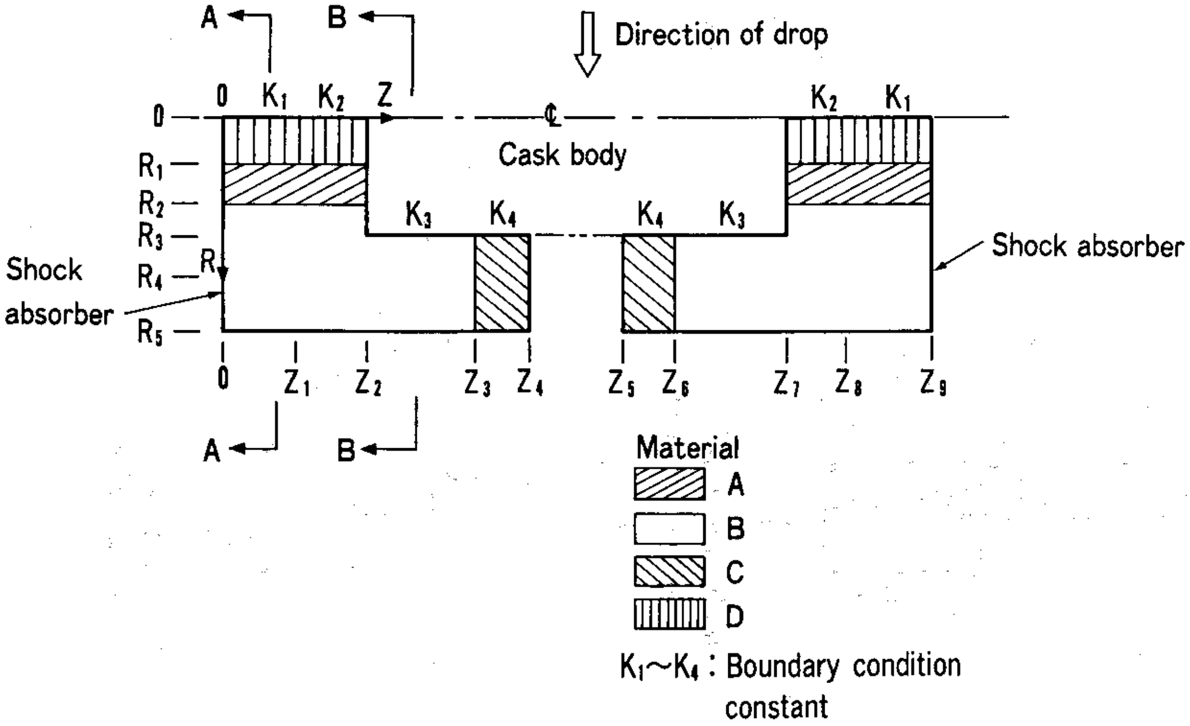
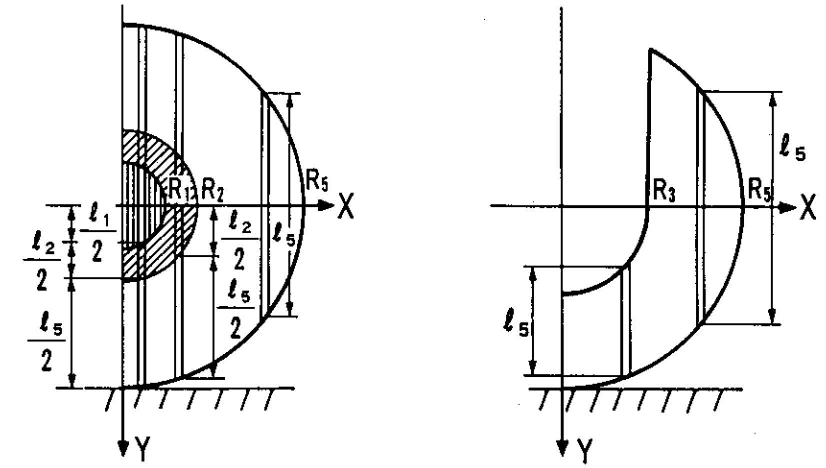
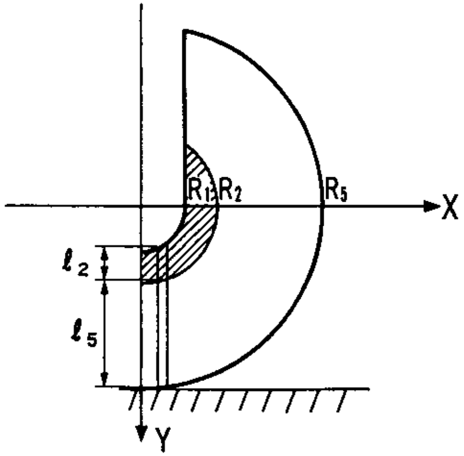


Fig.2.6 Geometry and material in the case of horizontal drop model



(a) Section A-A
(with Material D)

(c) Section B-B



(b) Section A-A
(without Material D)

Fig.2.7 Section view of horizontal drop model

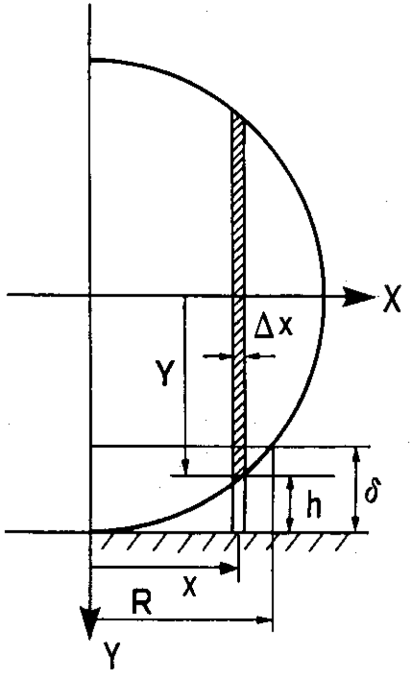


Fig.2.8 Sectional area of shock absorber

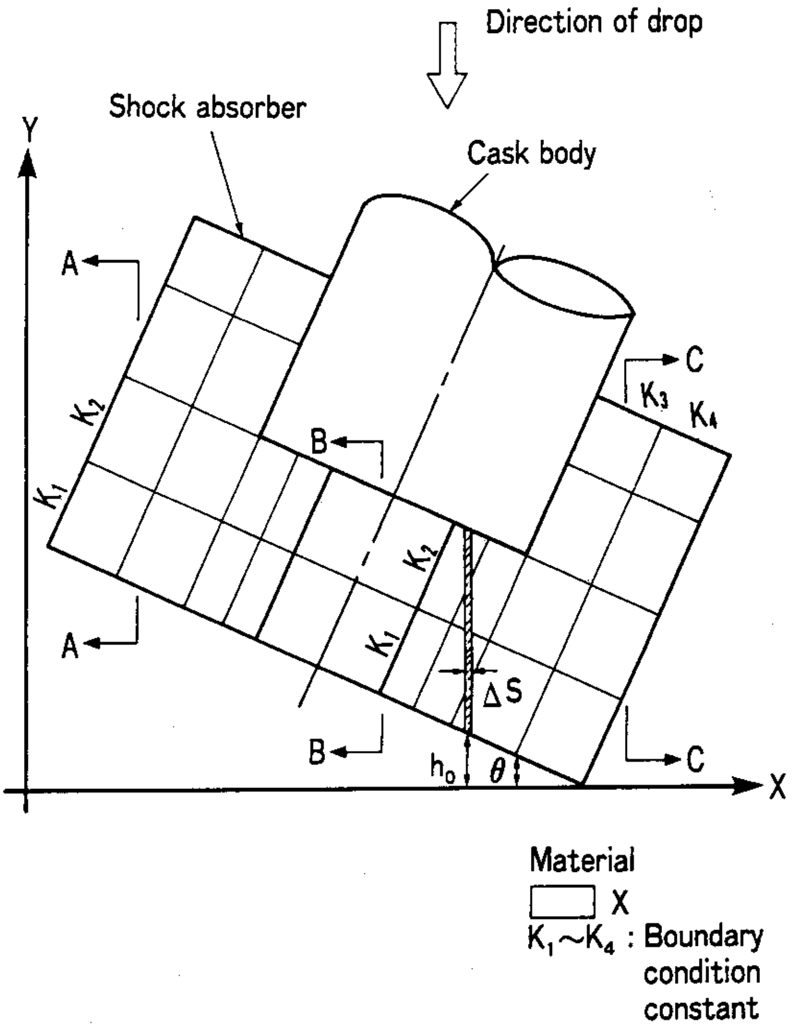


Fig.2.9 Geometry and material in the case of oblique drop model

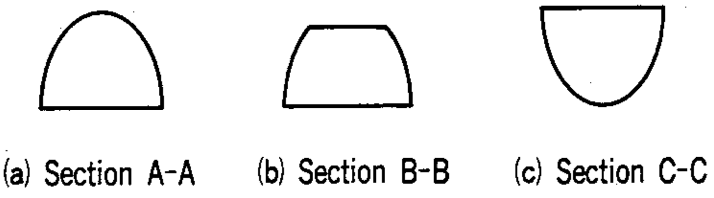


Fig.2.10 Sectional figure of shock absorber in the case of oblique drop (I)

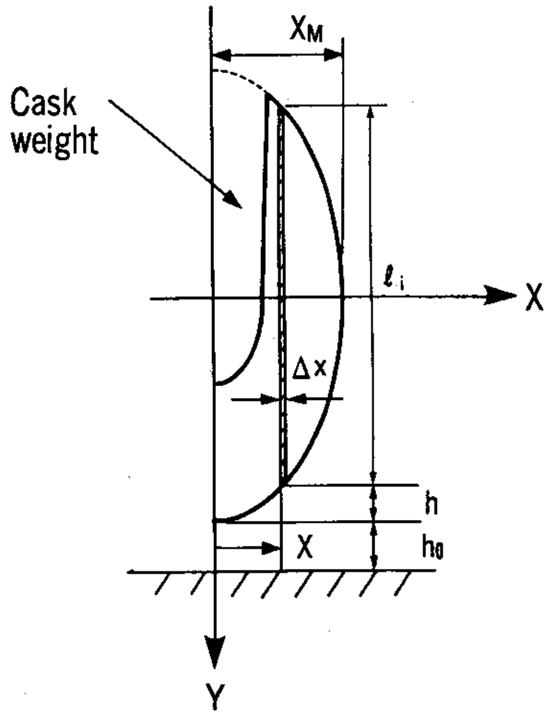


Fig.2.11 Sectional figure of shock absorber in the case of oblique drop (Ⅱ)

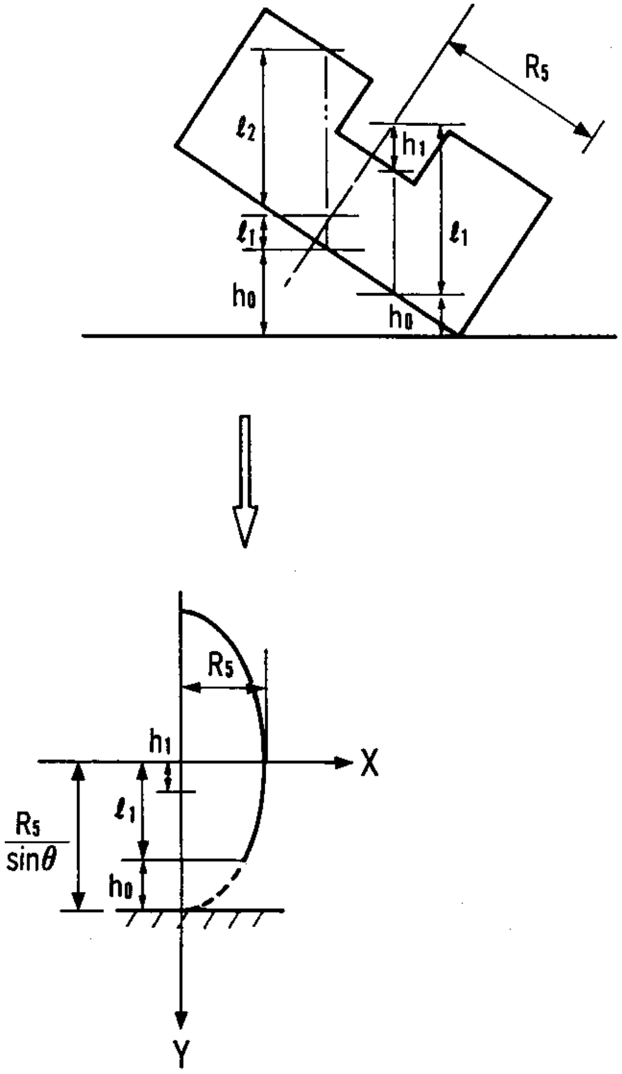
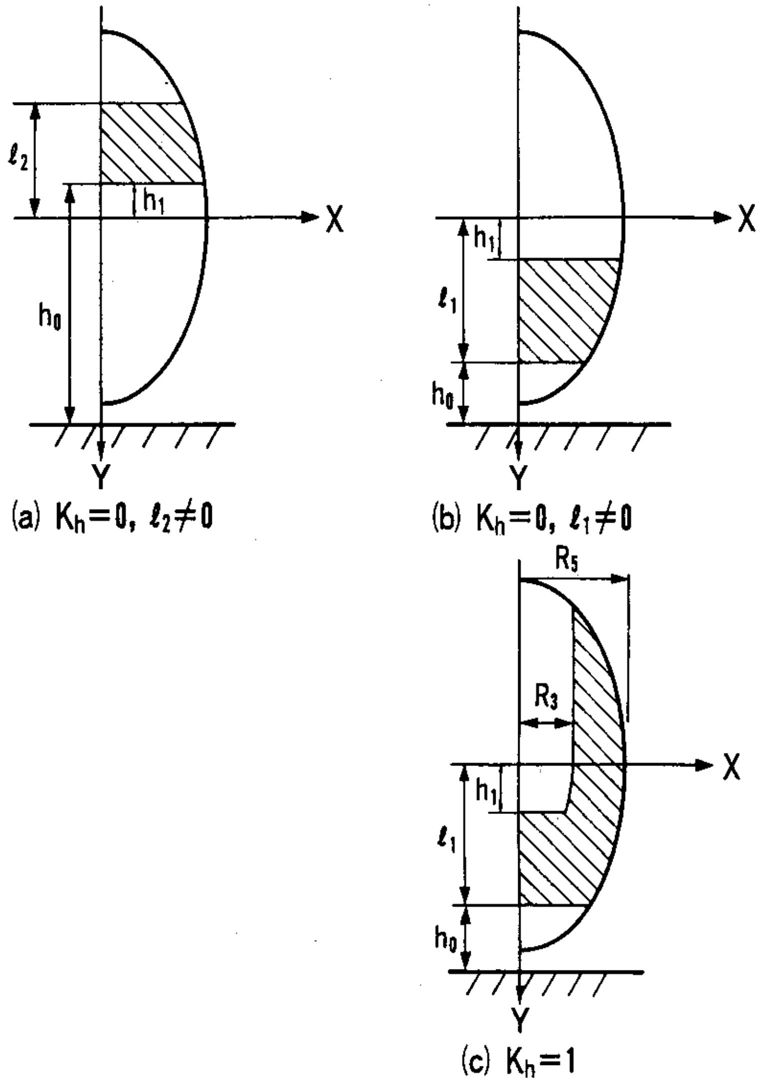


Fig.2.12 Sectional figure of shock absorber in the case of oblique drop (Ⅲ)



(a) $K_h=0,\ \ell_2\neq0$

(b) $K_h=0,\ \ell_1\neq0$

(c) $K_h=1$

Fig.2.13 Sectional figure of shock absorber in the case of oblique drop (Ⅳ)

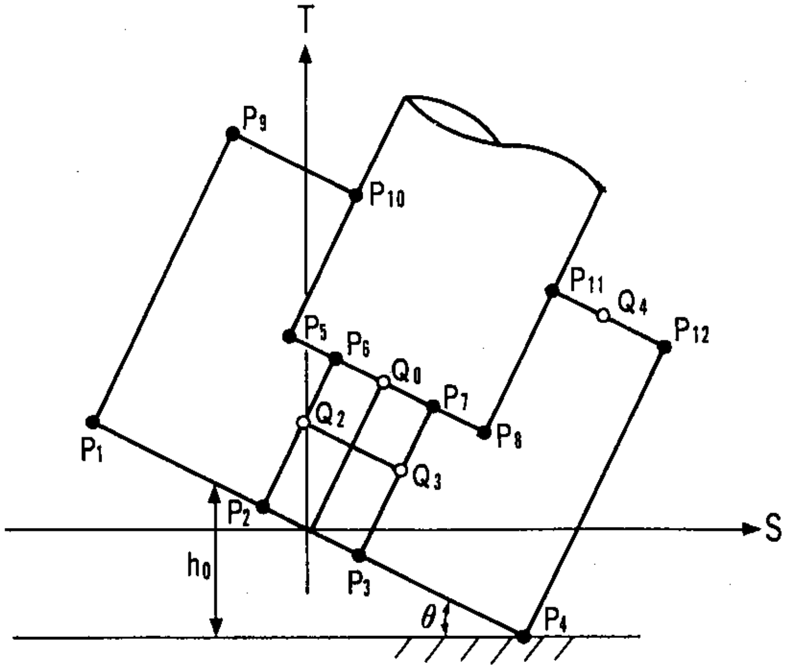


Fig.2.14 Geometry of shock absorber in the case of oblique drop and S-T coordinate

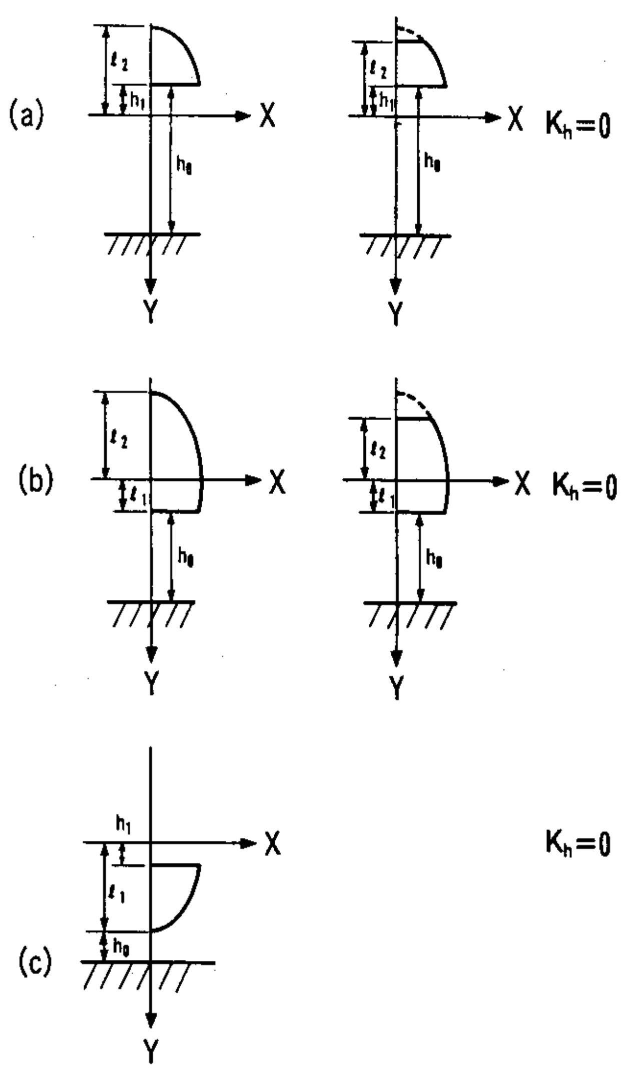


Fig.2.15 Various sectional figure of shock absorber in the case of oblique drop (I)

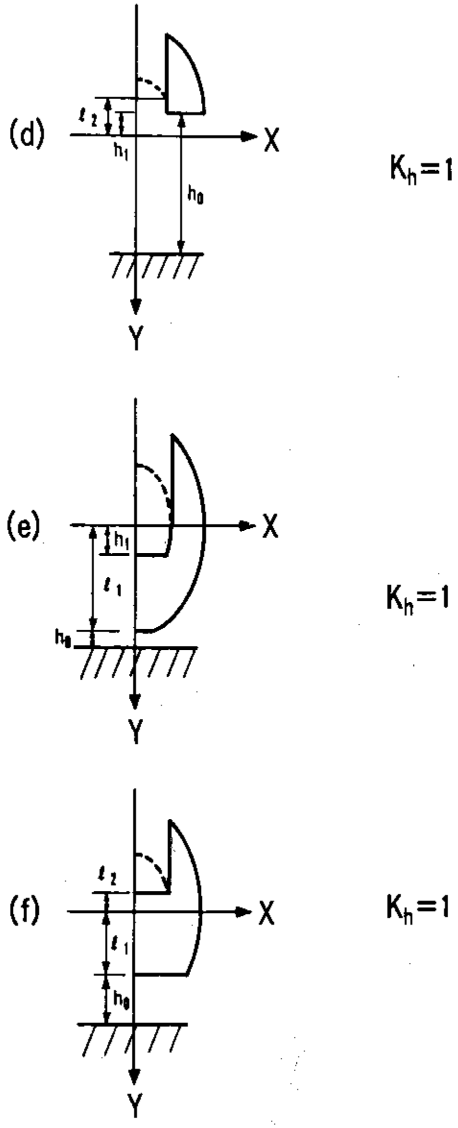


Fig.2.16 Various sectional figure of shock absorber in the case of oblique drop (II)

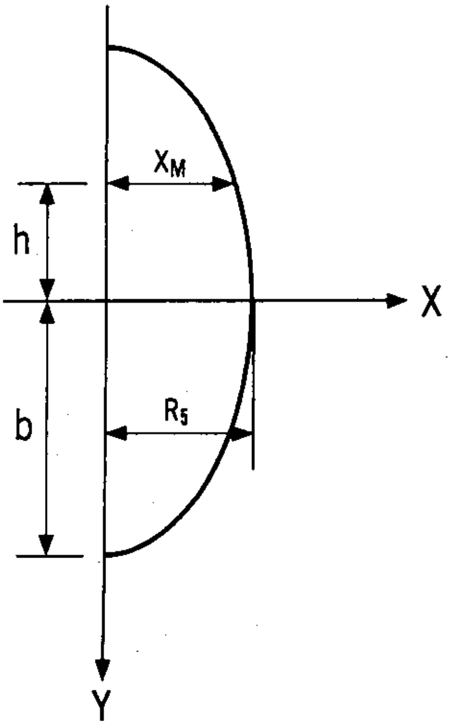


Fig.2.17 Ellipsoid

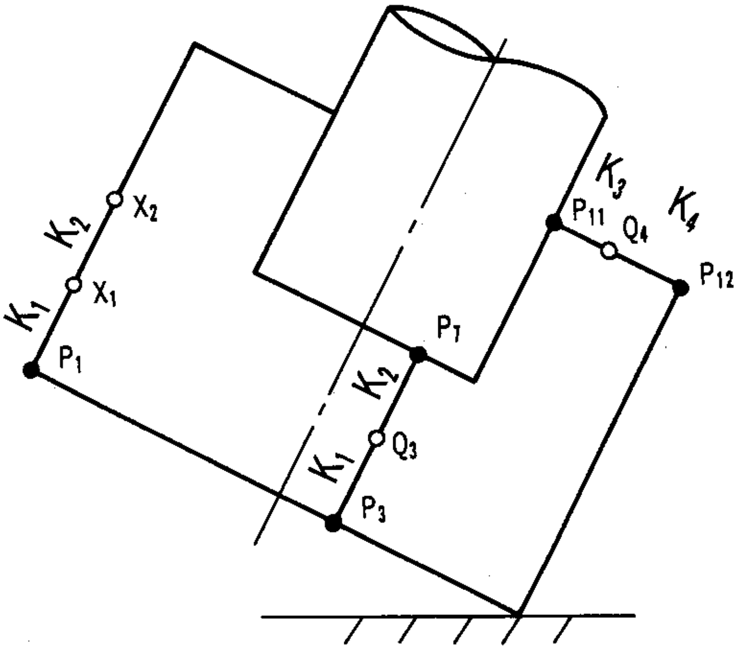


Fig.2.18 Boundary condition constant in the case of oblique drop

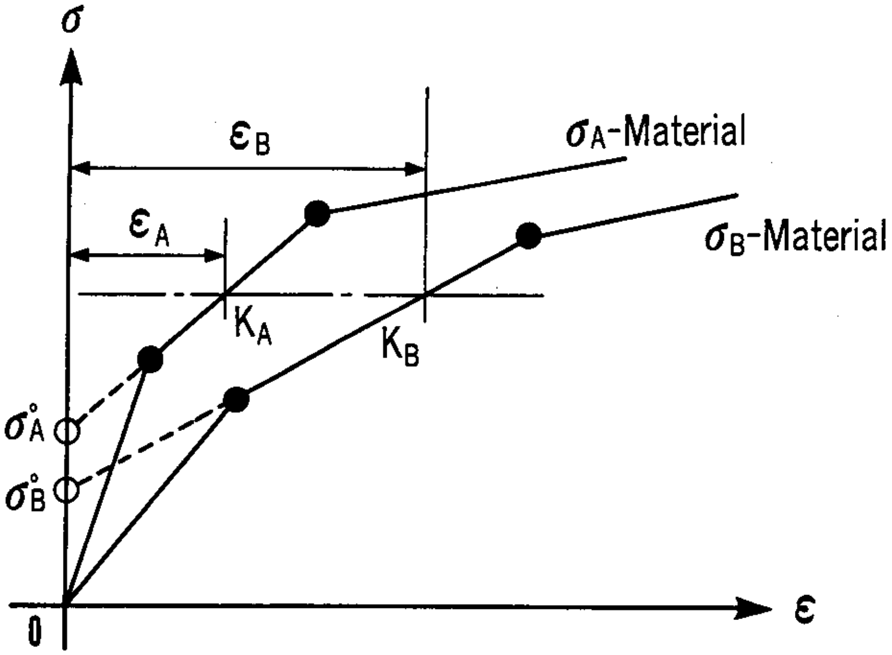


Fig.2.19 Stress-strain curves of shock absorber materials

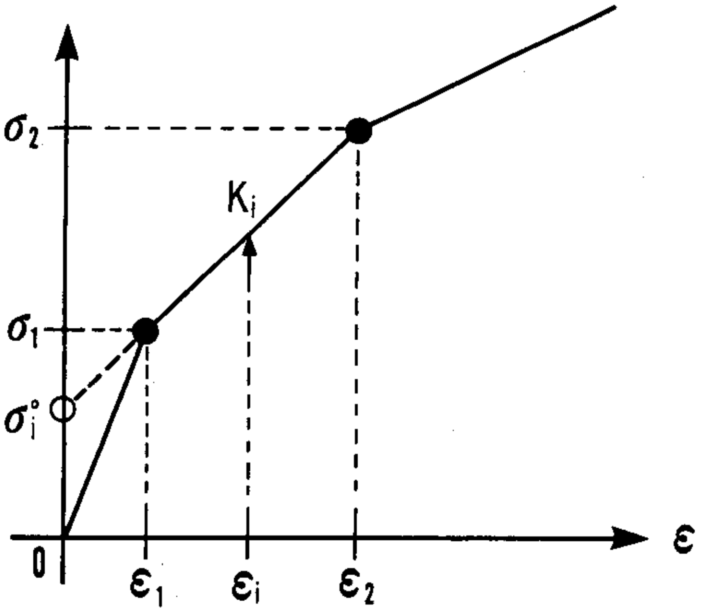


Fig.2.20 Stress-strain curve of shode absorber material

3.Benchmark Calculation

In order to demonstrate the adequacy of the simplified computer program CRUSH1, the benchmark calculations using experimental results of the 1/4 scale model of NUPAC 125B cask as shown in Fig. 3.1 have been performed.

Figure 3.2 is the deformed shapes of the cask after 9 meters drop impact to obtained by the detailed computer program DYNA3D. The comparison among the results obtained by the experiments, the simplified computer program CRUSH1 and the detailed computer program DYNA3D is shown in Table 3.1. The relation among the oblique angles of cask drop attitude and the maximum accelerations and the maximum deformations obtained by CRUSH1 are shown in Figs. 3.3 and 3.4. According to Table 3.1 and Figs. 3.3 and 3.4, results by the simplified computer program CRUSH1 agree with both the experimental results and that of the results of the detailed program DYNA3D.

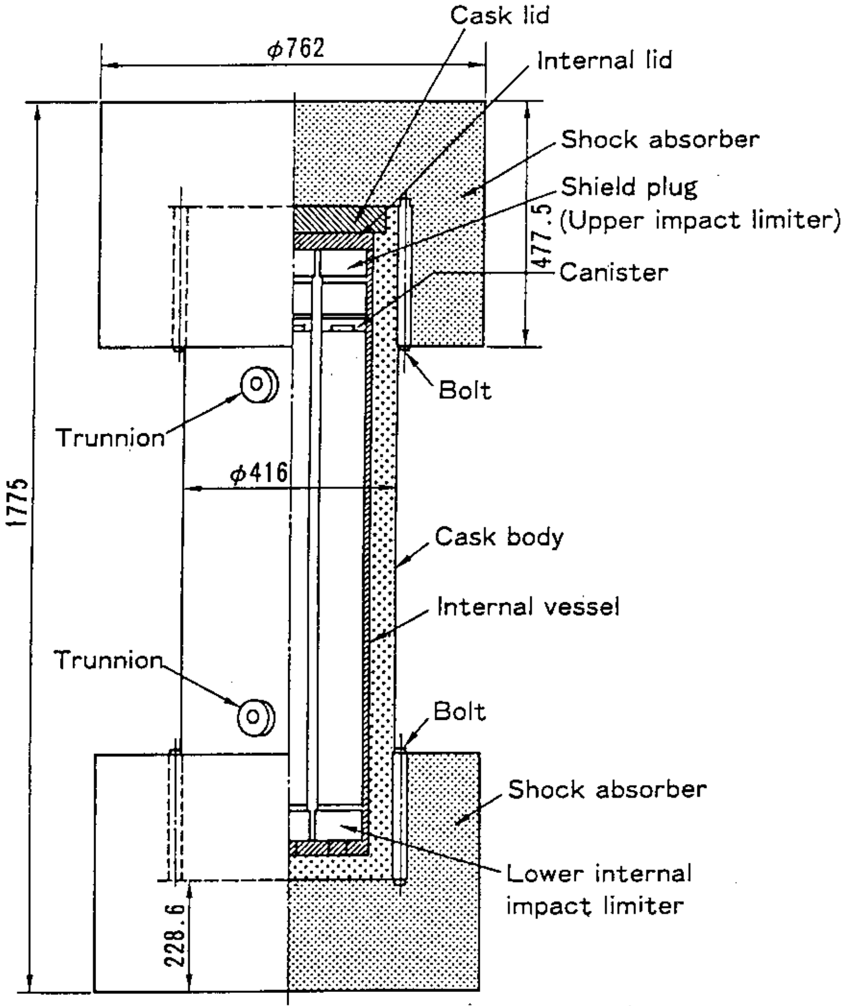
Table 3.1 Comparison between simplified and detailed analyses and experiment

Attitude	Acceleration (G)			Deformation (mm)		
	Experiment	Simplified analysis CRUSH	Detailed analysis DYNA3D	Experiment	Simplified analysis CRUSH	Detailed analysis DYNA3D
Vertical	200	179	271 * (200) **	51	52	50
Corner	106	125	130 (85)	127	151	132
Horizontal	180	183	216 (160)	61	63	64

* Value of low pass filter is 600 Hz.

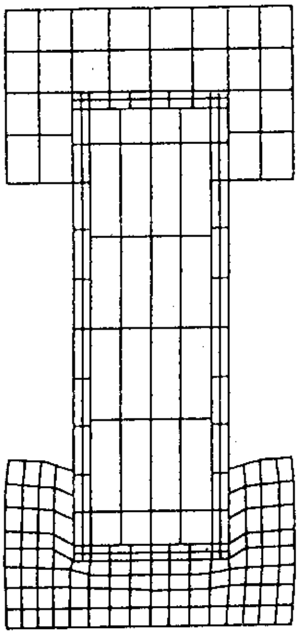
** Mean value $= \dfrac{\text{Impact velocity}}{\text{Rebound time}}$.

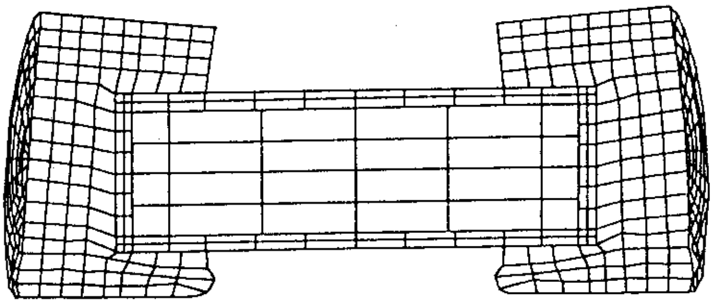
(NUPAC 125 B cask ¼ scale model).



(Dimension in mm)

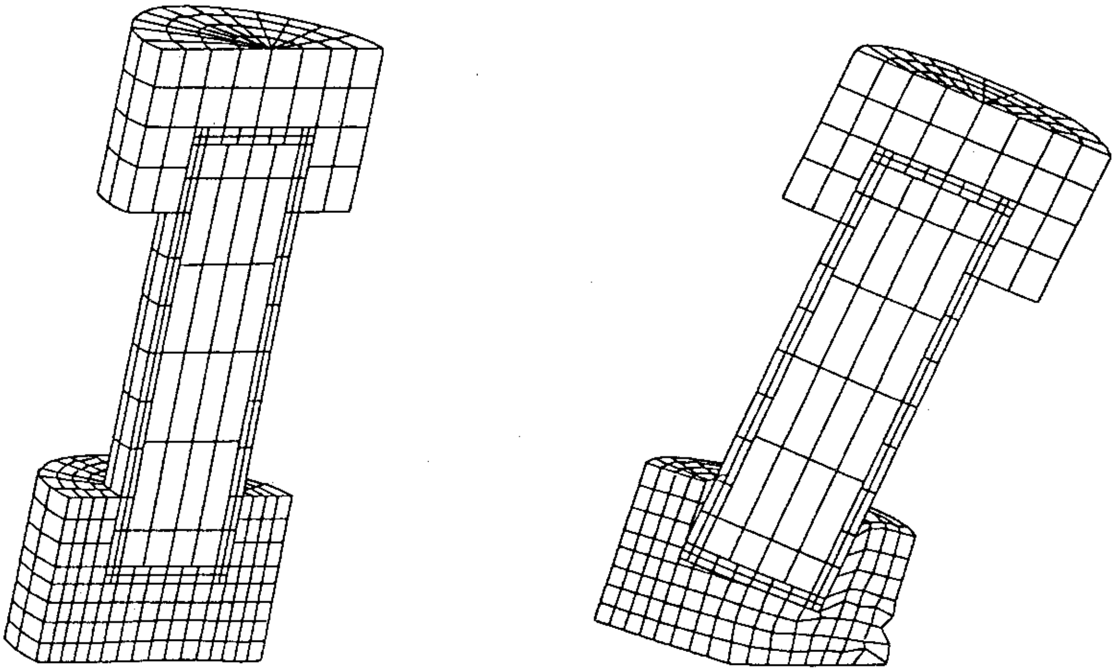
Fig.3.1 Shipping cask-NUPAC 125B (1/4 scale model)





(b) Horizontal drop

(a) Vertical drop



(c) Underformed mesh

(d) Corner drop

Fig.3.2 Deformed shape after 9m drop impact
(NUPAC 125B cask 1/4 scale model)

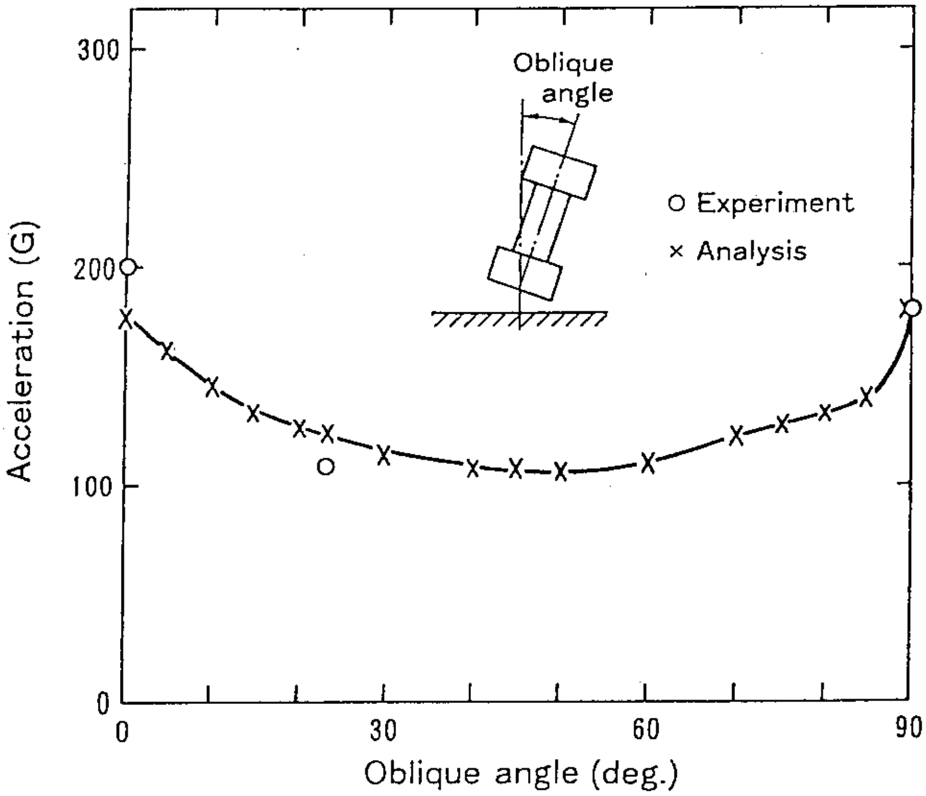


Fig.3.3 Comparison between simplified analysis and experiment on acceleration

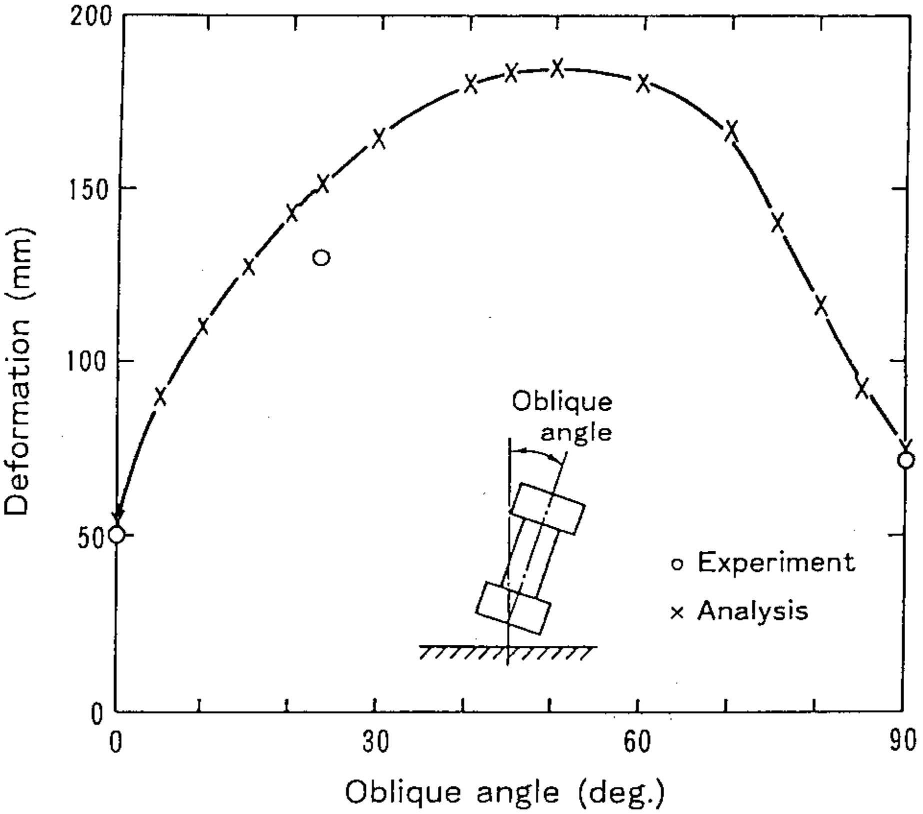


Fig.3.4 Comparison between simplified analysis and experiment on deformation

4. Computer Program

4.1 Program Description

The computer program CRUSH1 is static calculation program capable of evaluating the maximum acceleration of the cask body and the maximum deformation of the shock absorber using the uniaxial displacement method.

The computer program CRUSH1 consists of a main routine and thirteen subroutines that are MAIN, CARDIN, VERT, HORIZ, CORNER, MULTI, GETMAT, EQSOLV, GETSIG, FITSTP, MPLOT, DPLOT and UPLOT. Overall structure of CRUSH1 is shown in Fig. 4.1. Functions of subroutines are as follows:

MAIN : initiarizes start of run,
CARDIN : reads input data,
VERT : computes acceleration and deformation in the case of vertical(head-on or bottom-on) drop ompact,
HORIZ : computes acceleration and deformation in the case of horizontal (side-on) drop impact,
CORNER : computes acceleration and deformation in the case of oblique drop impact,
MULTI : computes relation of stress-strain data for multi-material layers,
GETMAT : computes relation of stress-strain data for single material,
EQSOLV : computes matrix data,
GETSIG : computes stress,
FITSTP : curve fitting for stress-strain data,
MPLOT : plots data,
DPLOT : plots data,
UPLOT : plots data.

A macroscopic flow chart of CRUSH1 is shown in Fig. 4.2.

4.2 Description of Input Data

This section describes the input data required by CRUSH1. The input data consists of the job description, the analysis type such as drop attitude, geometry, the cask weight, the initial condition, the boundary condition constants, the geometry unit selection and options for output plotting. The input instruction is simple and easy to follow. The input data forms are presented in Table 4.1.

4.3 Description of Output Data

This section describes the output data forms of CRUSH1. The contents of these various quantities are describes in the followings.

(1) Input data

The input data are printed in two formats. The first print format is exactly the same as they were read. Second, the computer program lists the input data as interpreted by CRUSH1.

(2) Calculation data

The calculation data; the deformation, the force, the energy and acceleration are printed at every interval steps.

(3) Graphical output

CRUSH1 provides users with graphical output of the deformation, relationship between acceleration and deformation, dissipation energy and deformation, acceleration and dissipation, energy and so forth.

Table 4.1 Input data for CRUSH1

Columns	Format	Variables	Descriptions
Data set No.1:Job description.			
1 - 8	2A4	NAME	Flag for data type. 'TITLE '.
9 - 10	2X	—	Blank,
11 - 78	70A1	NTITLE	Job description.
Data set No.2:Calculation model data.			
1 - 8	2A4	NAME	Flag for type of drop. 'VERTICAL':Vertical drop(Head-on drop or Bottom-on drop). 'HORIZONT':Horizontal drop(Side-on drop). 'CORNER ':Oblique drop.
9 - 10	2X	—	Blank.
11 - 20	F10.0	WEIGHT	Weight of cask(kg).
21 - 30	F10.0	HEIGHT	Height of cask drop(mm or cm).
31 - 40	F10.0	ANGLE	Oblique angle in the case of oblique drop (deg.),(ANGLE is 5～85deg.).
41 - 50	F10.0	DISP	Incremental deformation for calculation(mm or cm), if DISP is blank or zero, DISP equal to 1.0.
51 - 60	F10.0	CONV	Truncation error, if CONV is blank or zero, CONV equal to 1.0×10^{-5}.
61 - 64	A4	NDATA	Flag for unit. 'MM ':unit is mm. 'CM ':unit is cm.
65 - 70	6X	—	Blank.
71 - 80	F10.0	EPMAX	Limit strain for calculation,$0.0<\mathrm{EPMAX}<1.0$. If EPMAX is blank or zero, EPMAX equal to 1.0,

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
Data set No.3:Geometry data.			
1 - 8	2A4	NAME	Flag for data type. 'LENGTH '
9 - 10	2X	−	Blank,
11 - 20	F10.0	CLENGTH	Length of cask body.
21 - 30	F10.0	RADIUS	Radius of cask body.
31 - 35	5X	−	Blank.
36 - 40	A4	KDTYPE	Flag of material data. 'SIGD':material σ_D exist. 'NONE':material σ_D do not exist.
Data set No.4:Options for input data check and plotting.			
1 - 8	2A4	NAME	Flag for input data check or execution. 'MODEL ':input data check. 'RUN ':execution.
9 - 10	2X	−	Blank.
11 - 14	A4	NAME1	Plotting of calculation results. 'PLOT':plotting of calculation results. ' ':no plotting.
15	1X	−	Blank.
16 - 19	A4	NDATA	Flag for debugging of program. 'CHEC':debugging of program. ' ':no debugging.
Data set No.5:Geometry of shock absorber(I).			
1 - 8	2A4	NAME	Flag for coordinate of radial direction. 'RCOOR '.
9 - 10	2X	−	Blank.
11 - 20	F10.0	R1	Radius of shock absorber(see Fig.2.5, 2.6 and 2.9).
21 - 30	F10.0	R2	Same as above.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
31 - 40	F10.0	R3	Same as above.
41 - 50	F10.0	R4	Same as above.
51 - 60	F10.0	R5	Same as above.
Data set No.6A:Geometry of shock absorber(IIA).			
1 - 8	2A4	NAME	Flag for coordinate of axial direction. 'ZCOOR '.
9 - 10	2X	−	Blank.
11 - 20	F10.0	Z1	Axial lenght of shock absorber(see Fig.2.5 and 2.6).
21 - 30	F10.0	Z2	Same as above.
31 - 40	F10.0	Z3	Same as above.
41 - 50	F10.0	Z4	Same as above.
51 - 60	F10.0	Z5	Same as above.
61 - 70	F10.0	Z6	Same as above.
71 - 80	F10.0	Z7	Same as above.
Data set No.6B:Geometry of shock absorber(IIB).			
1 - 8	2A4	NAME	Flag for coordinate of axial direction. 'ZCOOR '.
9 - 10	2X	−	Blank.
11 - 20	F10.0	Z8	Axial lenght of shock absorber(see Fig.2.5 and 2.6).
21 - 30	F10.0	Z9	Same as above.
Data set No.7:Boundary condition constant.			
1 - 4	2A4	NAME	Flag for data type. 'MESH '.
9 - 10	2X	−	Blank.
11 - 15	I5	NPART	Number of partitions in the X-Y plain of horizontal or oblique drop case.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
			Maximum NPART is 300. If NPART is blank or zero, NPART equal to 100.
15 - 20	I5	KPART	Number of partitions in the R-Z plain of oblique drop case. Maximum KPART is 400. If KPART is blank or zero, KPART equal to 100.
21 - 30	F10.0	K1	Boundary condition constant(see Fig.2.5, 2.6 and 2.9).
31 - 40	F10.0	K2	Same as above.
41 - 50	F10.0	K3	Same as above.
51 - 60	F10.0	K4	Same as above.
61 - 70	F10.0	K5	Same as above.
Data set No.8A:Calculation model data.			
1 - 8	2A4	NAME	Flag for material data identification. 'SIGA ': σ_A stress-strain data(see Fig. 2.5, 2.6 and 2.9). 'SIGB ': σ_B stress-strain data. 'SIGC ': σ_C stress-strain data. 'SIGD ': σ_D stress-strain data. 'SIGX ': σ_X stress-strain data.
9 - 10	2X	−	Blank.
11 - 15	I5	NN	Number of stress-strain data.
16 - 20	5X	−	Blank.
21 - 30	F10.0	FACT	Multiplication factor of stress data.
Data set No.8B:Stress-strain-data.			
1 - 10	F10.0	STRAIN(1)	Strain(see Fig.2.19 and 2.20).
11 - 20	F10.0	STRESS(1)	Stress(see Fig.2.19 and 2.20).
21 - 30	F10.0	STRAIN(2)	Strain.
31 - 40	F10.0	STRESS(2)	Stress.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
41 - 50	F10.0	STRAIN(3)	Strain.
51 - 60	F10.0	STRESS(3)	Stress.
61 - 70	F10.0	STRAIN(4)	Strain.
71 - 80	F10.0	STRESS(4)	Stress.
			Rpeat 8B data set for NN data.
Repeat 8A and 8B data set for SIGA, SIGB, SIGD and SIGX stress-strain data.			

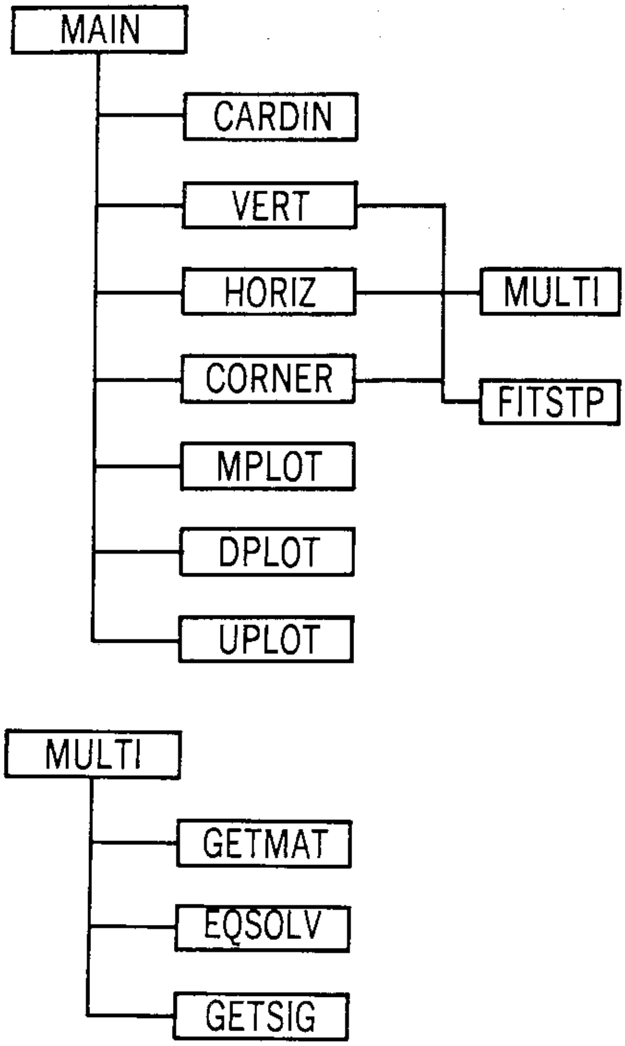


Fig.4.1 Sturcture of computer program CRUSH1

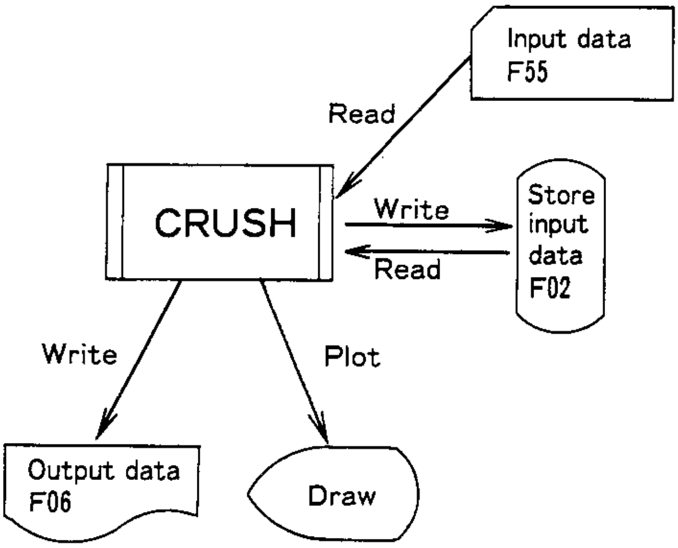


Fig.4.2 Program flow

5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask bodys and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH1 make to analyze economical and by shortening input and computer time to about 1/20 or less as compared with other detailed computer programs to analyze dynamic interactions. The results obtained by the simplified computer program CRUSH1 has an enough adequacy for its practical use. CRUSH1 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

Ackowledgements

The author is indebted Dr. Kazuo Asada of Mitubishi Heavy Industries, Ltd. for providing the sample problems and valuable discussions. He is also indebted Mr. Yutaka Hasegawa, Junji Oshika and Takashi Ishiwata of CRC Research Institute, Inc. for assistance of making the computer program.

References

(1) Asada, K. et al.,"Development of Simplified Analysis Codes for Nine Meter Drop and One Meter Puncture Tests for a Radioactive Material Transport Cask", Waste Management '88(Tuscon)(1988).

(2) Ikushima, T.,"CRUSH: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-M 90-004(1990).

5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask bodys and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH1 make to analyze economical and by shortening input and computer time to about 1/20 or less as compared with other detailed computer programs to analyze dynamic interactions. The results obtained by the simplified computer program CRUSH1 has an enough adequacy for its practical use. CRUSH1 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

Ackowledgements

The author is indebted Dr. Kazuo Asada of Mitubishi Heavy Industries, Ltd. for providing the sample problems and valuable discussions. He is also indebted Mr. Yutaka Hasegawa, Junji Oshika and Takashi Ishiwata of CRC Research Institute, Inc. for assistance of making the computer program.

References

(1) Asada, K. et al.,"Development of Simplified Analysis Codes for Nine Meter Drop and One Meter Puncture Tests for a Radioactive Material Transport Cask", Waste Management '88(Tuscon)(1988).

(2) Ikushima, T.,"CRUSH: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-M 90-004(1990).

5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask bodys and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH1 make to analyze economical and by shortening input and computer time to about 1/20 or less as compared with other detailed computer programs to analyze dynamic interactions. The results obtained by the simplified computer program CRUSH1 has an enough adequacy for its practical use. CRUSH1 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

Ackowledgements

The author is indebted Dr. Kazuo Asada of Mitubishi Heavy Industries, Ltd. for providing the sample problems and valuable discussions. He is also indebted Mr. Yutaka Hasegawa, Junji Oshika and Takashi Ishiwata of CRC Research Institute, Inc. for assistance of making the computer program.

References

(1) Asada, K. et al.,"Development of Simplified Analysis Codes for Nine Meter Drop and One Meter Puncture Tests for a Radioactive Material Transport Cask", Waste Management '88(Tuscon)(1988).

(2) Ikushima, T.,"CRUSH: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-M 90-004(1990).

Appendix A Sample Problem Input

```
                            C R U S H 1   M O D E L   D A T A

                  1         2         3         4         5         6         7         8
         ....:....0....:....0....:....0....:....0....:....0....:....0....:....0....:....0
   1-    TITLE     NUPAC-125B CASK VERTICAL IMPACT MODEL
   2-    VERTICAL  1283.6    9000.0      0.0   5.0       0.0       MM
   3-    LENGTH    1775.0    208.0               SIGD
   4-    RUN       PLOT
   5-    RCOOR       0.0     220.0     247.0     350.0     380.0
   6-    ZCOOR     224.0     224.0     350.0     474.0
   7-    ZCOOR
   8-    MESH                  1.0     1.0       1.0       1.0       1.0
   9-    SIGA          24      1.0
  10-    0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
  11-    0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
  12-    0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
  13-    0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
  14-    0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
  15-    0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
  16-    SIGB          24      1.0
  17-    0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
  18-    0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
  19-    0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
  20-    0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
  21-    0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
  22-    0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
  23-    SIGC          24      1.0
  24-    0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
  25-    0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
  26-    0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
  27-    0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
  28-    0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
  29-    0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
  30-    SIGD          24      1.0
  31-    0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
  32-    0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
  33-    0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
  34-    0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
  35-    0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
  36-    0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5


         ...... MODEL WEIGHT  =    1283.60  (KG)
         ...... MODEL HEIGHT  =    9000.00  (MM)
         ...... CORNER ANGLE  =       0.00  (DEG.)
         ...... INCREMENT DISP=       5.00  (MM)


         ...... K-FACTOR      =     1.0000     1.0000     1.0000     1.0000     1.0000
```

Appendix B Sample Problem Output

```
                                   INPUT DATA ECHO

   DATA                1         2         3         4         5         6         7         8
 SEQ. NO.     ----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0

     1        TITLE     NUPAC-125B CASK VERTICAL IMPACT MODEL
     2        VERTICAL  1283.6    9000.0      0.0   5.0       0.0       MM
     3        LENGTH    1775.0    208.0               SIGD
     4        RUN       PLOT
     5        RCOOR       0.0     220.0     247.0     350.0     380.0
     6        ZCOOR     224.0     224.0     350.0     474.0
     7        ZCOOR
     8        MESH                  1.0     1.0       1.0       1.0       1.0
     9        SIGA          24      1.0
    10        0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
    11        0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
    12        0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
    13        0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
    14        0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
    15        0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
    16        SIGB          24      1.0
    17        0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
    18        0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
    19        0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
    20        0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
    21        0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
    22        0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
    23        SIGC          24      1.0
    24        0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
    25        0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
    26        0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
    27        0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
    28        0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
    29        0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5
    30        SIGD          24      1.0
    31        0.0       0.0      0.02      0.2       0.03      0.27      0.04      0.29
    32        0.05      0.3      0.25      0.3       0.33      0.33      0.4       0.36
    33        0.45      0.4      0.5       0.44      0.55      0.52      0.58      0.57
    34        0.6       0.61     0.62      0.68      0.64      0.74      0.66      0.82
    35        0.68      0.91     0.7       1.06      0.72      1.20      0.74      1.42
    36        0.76      1.75     0.78      2.18      0.8       2.78      0.82      3.5

              ----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
                       1         2         3         4         5         6         7         8

                    * * *  INPUT DATA END  * * *
```

Appendix B (Continued)

```
                                   Appendix B  (Continued)
NUPAC-125B CASK VERTICAL IMPACT MODEL

MODEL TYPE = VERTICAL              TOTAL ENERGY =     11552.40 (KG-M)

     STEP         DEPTH          FORCE          ENERGY    ACCELERATION
                   (MM)           (KG)          (KG-M)            (G)

        1          5.00       69082.74          345.41          53.82
        2         10.00      110880.80          899.82          86.38
        3         15.00      129096.61         1545.30         100.57
        4         20.00      134048.81         2215.54         104.43
        5         25.00      136093.81         2896.01         106.03
        6         30.00      136093.81         3576.48         106.03
        7         35.00      136093.81         4256.95         106.03
        8         40.00      136093.81         4937.42         106.03
        9         45.00      136093.81         5617.89         106.03
       10         50.00      136093.81         6298.36         106.03
       11         55.00      136093.81         6978.83         106.03
       12         60.00      137377.28         7665.71         107.03
       13         65.00      138981.62         8360.62         108.27
       14         70.00      140585.97         9063.55         109.52
       15         75.00      142239.82         9774.75         110.81
       16         80.00      144073.35        10495.12         112.24
       17         85.00      145906.88        11224.65         113.67
       18         85.50      146090.24        11297.70         113.81
       19         86.00      146273.59        11370.83         113.96
       20         86.50      146456.94        11444.06         114.10
       21         87.00      146640.30        11517.38         114.24
       22         87.24      146727.76        11552.40         114.31
```

Appendix C Graphical Output

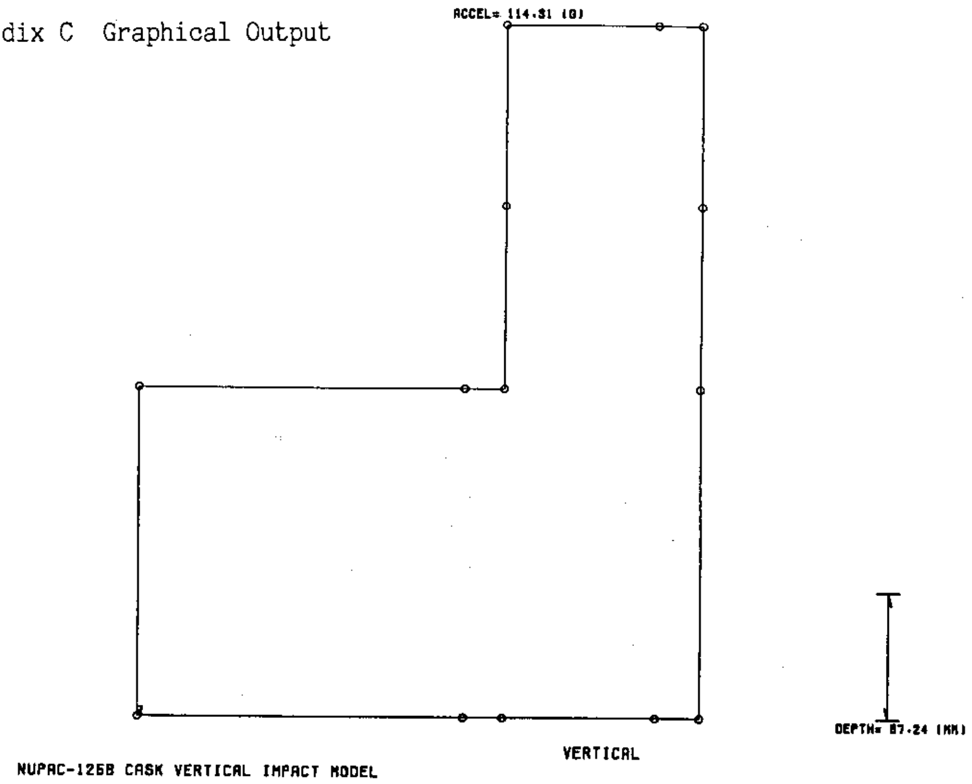


Fig.C.1 Graphical Output of CRUSH1(1)

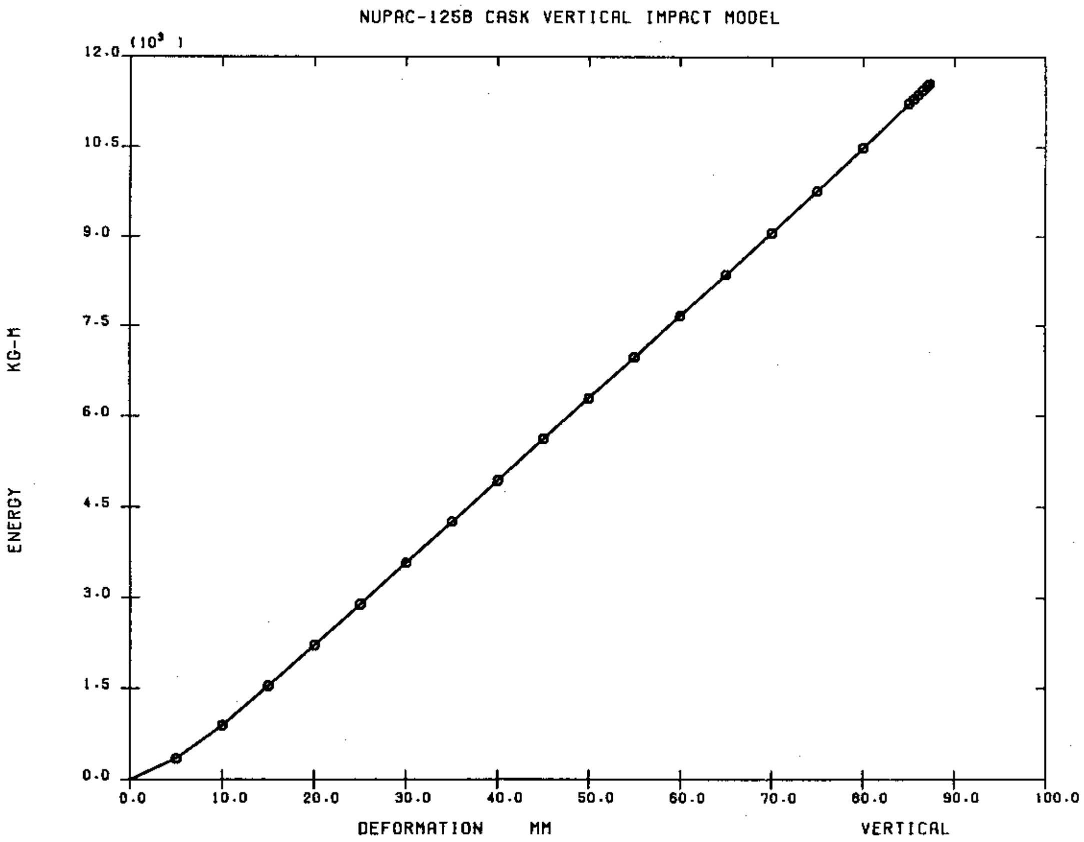


Fig.C.2 Graphical Output of CRUSH1(2)

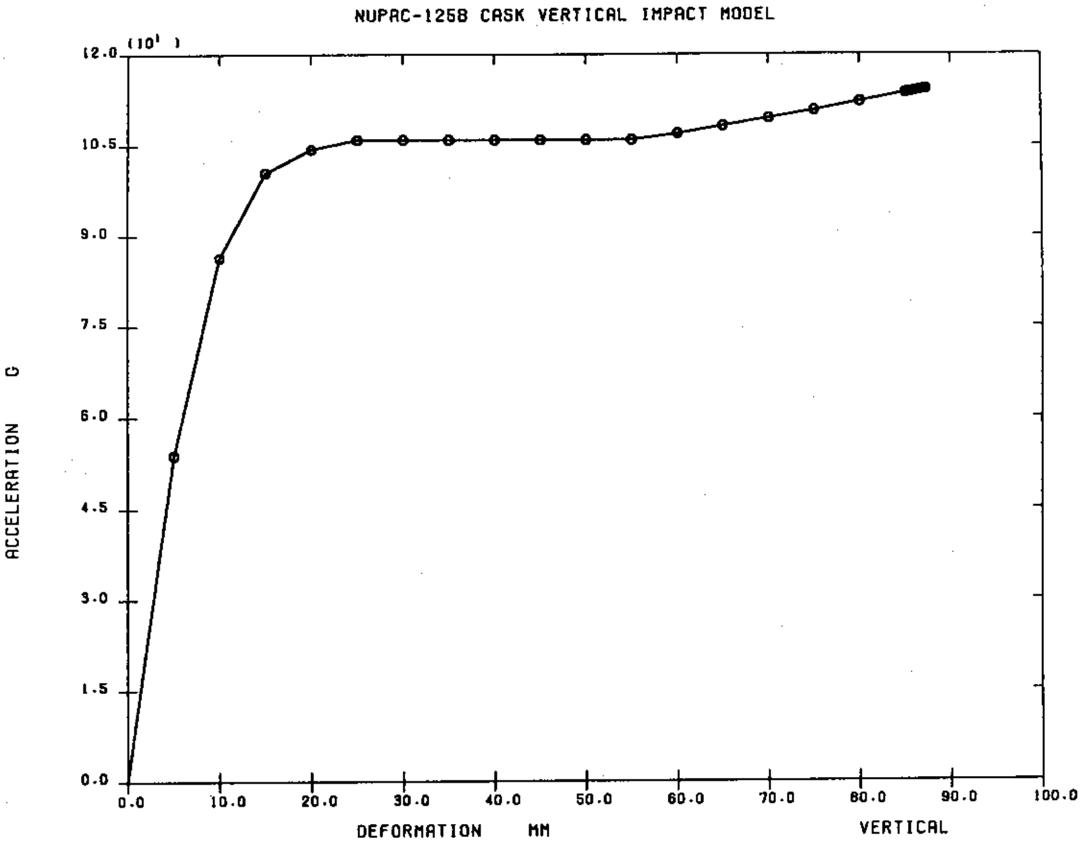


Fig.C.3 Graphical Output of CRUSH1(3)

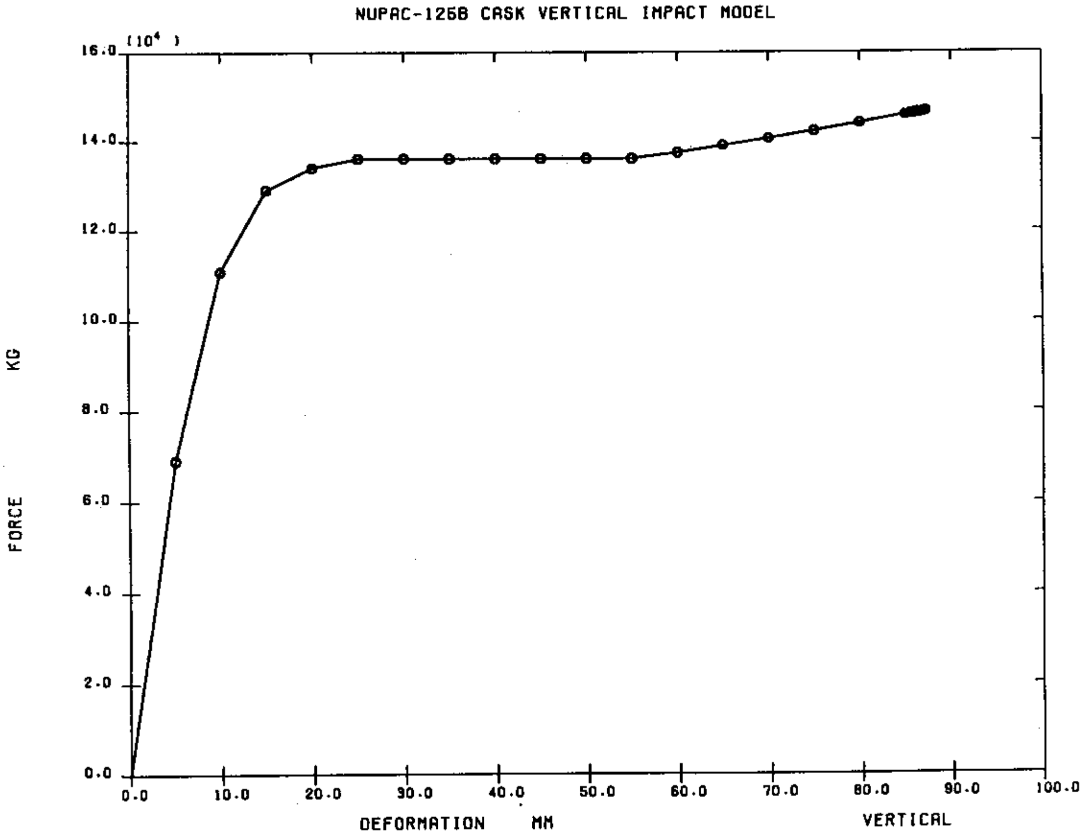


Fig.C.4 Graphical Output of CRUSH1(4)

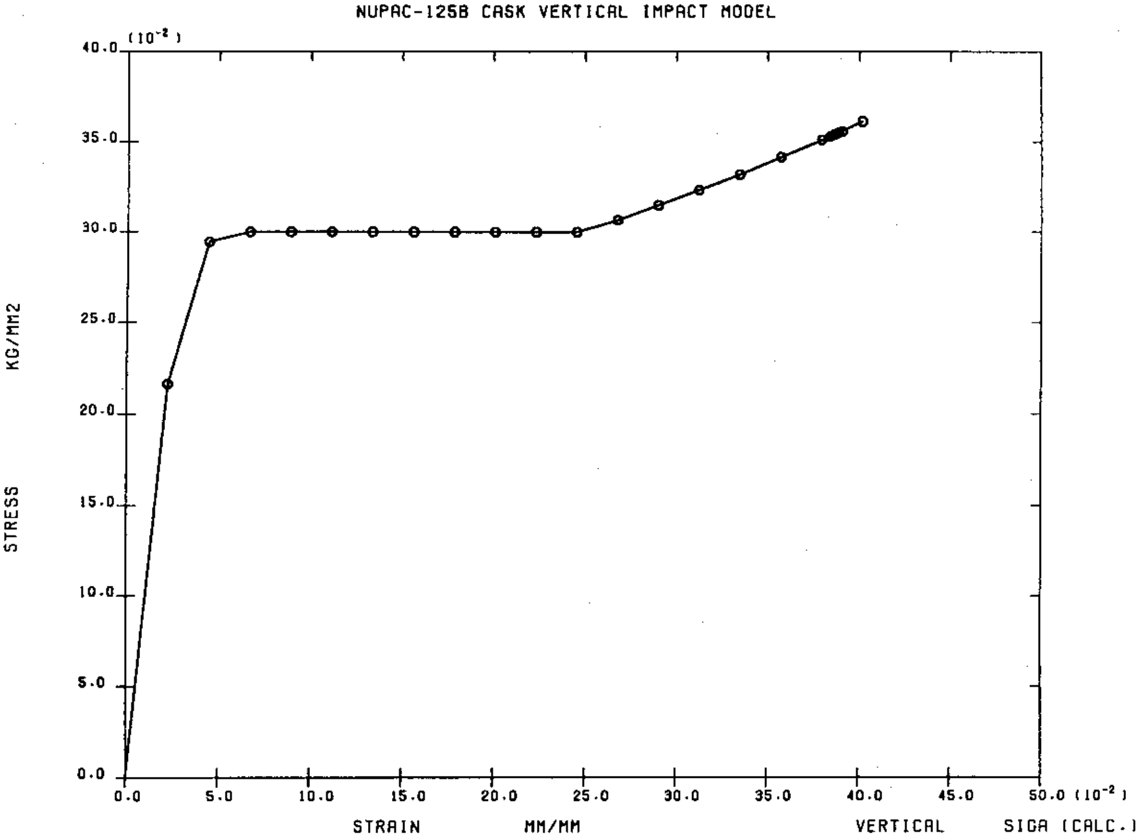


Fig.C.5 Graphical Output of CRUSH1(5)

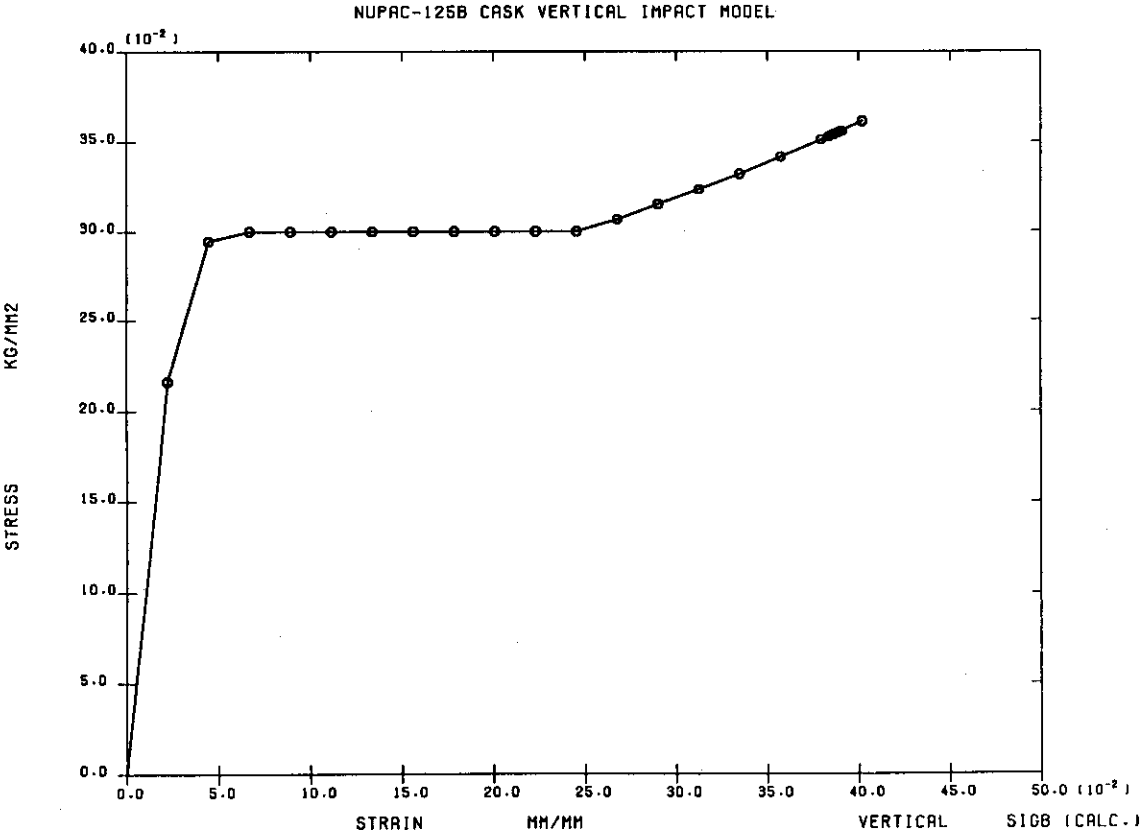


Fig.C.6 Graphical Output of CRUSH1(6)

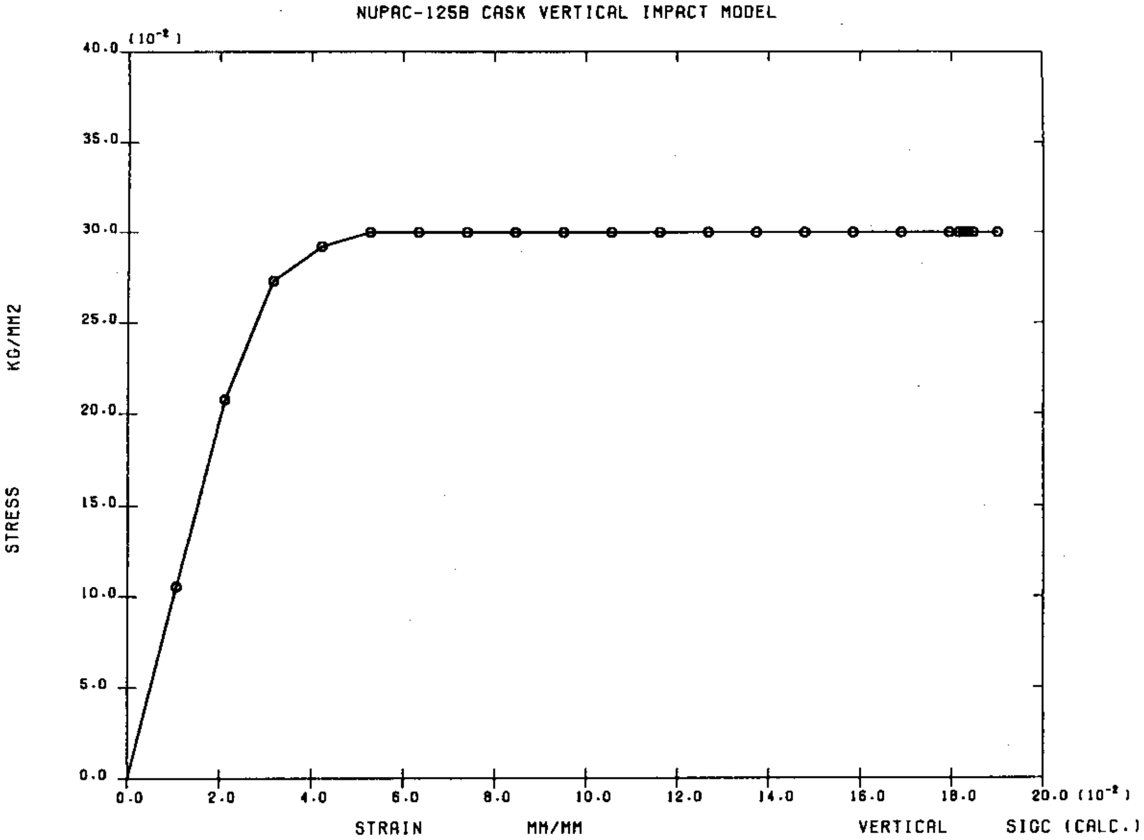


Fig.C.7 Graphical Output of CRUSH1(7)

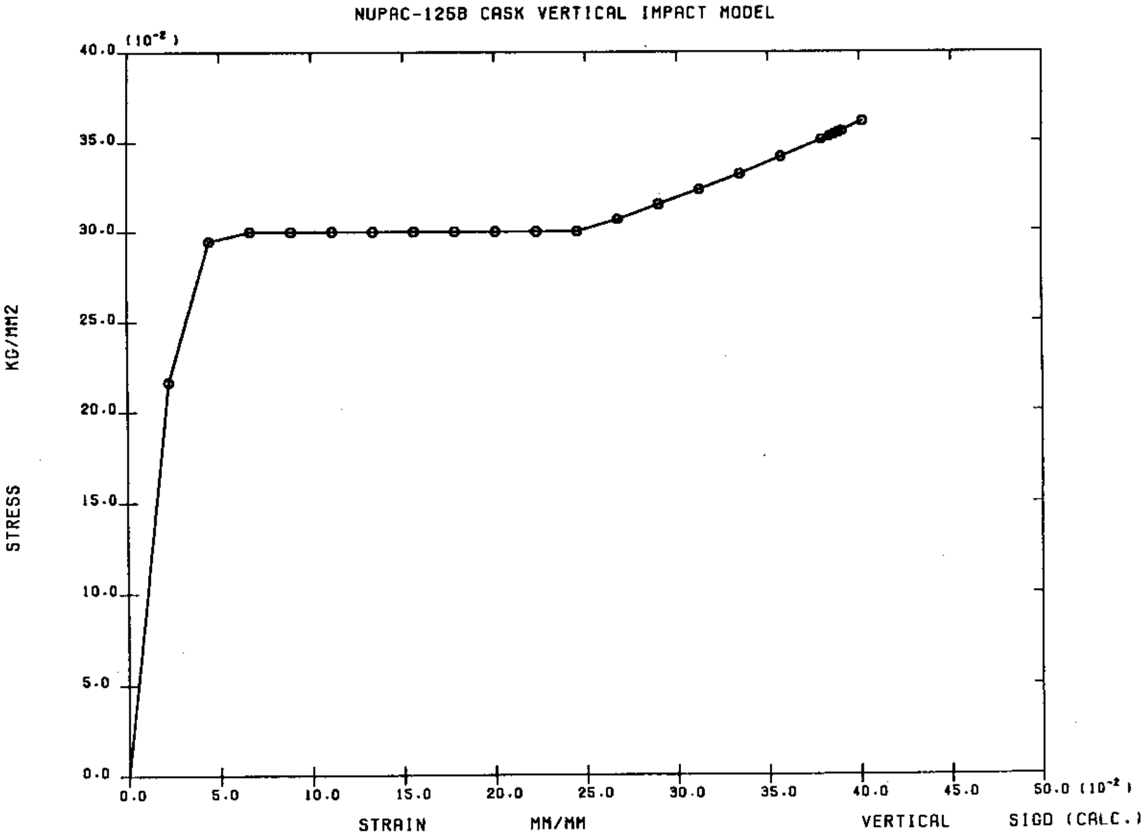


Fig.C.8 Graphical Output of CRUSH1(8)

Appendix D Job Control Data

The job control data for CRUSH1 execution on the computer FACOM M-780 in JAERI is as follows:

```
//JCLG JOB
// EXEC JCLG
//SYSIN DD DATA,DLM='++'
// JUSER XXXXXXXX.XX,XXXXXXXX,XXXX.XX,CRUSH1
// T.03 C.02 W.01 I.02 CLS GRP
// OPTP MSGCLASS=A,MSGLEVEL=(2,0,1),CLASS=B,NOTIFY=JXXXX
// OPTP PASSWORD=XXXXXXXX
// EXEC LMGOEX,LM=J2322.LMCRUSH1,PNM=CRUSH1
// EXPAND GRNLP
//FT55F001 DD DSN=JXXXX.DTCRUSH1.DATA,DISP=SHR
//FT02F001 DD DSN=SPACE=(TRK,(5,5)),UNIT=TSSWK
++
//
```

Appendix E Program Abstract

1. Name :

   CRUSH1.

2. Computer for which the program is designed and others upon which it is possible:

   FACOM M-780, SUN4 or IBM-PC.

3. Nature of physical problem solved:

   Drop impact analysis of radioactive material transport casks.

4. Method of solutions:

   One dimensional static analysis.

5. Restrictions on the complexity of the problem:

   None.

6. Typical running time:

   FACOM-M780 : 2 seconds.
   SUN4 : 5 seconds.
   IBM-PC : 10 seconds.

7. Unusual features of the program:

   None.

8. Related and auxiliary program:

   None.

9. Status:

   -

10. References:

    "Simplified Analysis Computer Program and Their Adequacy for Radioactive Materials Shipping casks", PATRAM'89 (Washington) DC, USA June 11-16, 1989).

11. Machine requirement:

    Required 1100 k bytes of core memory.

12. Program language used:

    FORTRAN-77.

13.Operating system or monitor under which the program is executed:

FACOM M-780 : MSP.
SUN4 : Solaris 2.1.
IBM PC : Windows 3.1, Windows 95 or Windows NT.

14.Any other programming or operating information or restrictions:

The program is approximately 2800 source steps(include comment lines). The graphical programs are used as follows:

FACOM M-780 : CALCOMP pltter or the compatible ones.
SUN4 : X-windows.
IBM PC : windows 3.1, Windows 95 or Windows NT.

15.Name and establishment of author:

T. Ikushima
Japan Atomic Energy Research Institute,
Tokai Research Establishment,
Department of Fuel Cycle Safety Research,
Tokai-mura, Naka-gun, Ibaraki-ken, 319-11
Japan

16.Material available:

Source.

Appendix F Program Source List

```
C......................................................
C
C         C R U S H 1    C O D E   (1989-01-05)
C
C......................................................
C
C***    /PROGRAMMED BY JAERI                    1988.01.29
C***     MODIFIED   BY JAERI                    1988.09.19
C***     MODIFIED   BY JAERI                    1988.10.31
C***     MODIFIED   BY JAERI                    1989.01.05
C***     MODIFIED   BY JAERI                    1994.06.08
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +          RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +          DISP,     FACTK(5), ENERGO,   GRAVIT,   EPMAX,
     +          MTINN,    MTOUT,    MODEL,    IFLAG,    KSOLV,
     +          KOUT,     NPART,    KPART,    NTITLE(18)
      COMMON /MATDAT/     TABLE(100,2,5),     MLENG(5)
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
      COMMON /SWITCH/     KEOF,     KKTEST
C
      DIMENSION LABEL(3,3), NDIMM(2)
C
      DATA LABEL/ 4HVERT, 4HICAL, 4H    ,   4HHORI, 4HZONT, 4HAL  ,
     +            4HCORN, 4HER  , 4H    /
C
      DATA NDIMM/ 4HMM  , 4HCM  /
      DATA NDIM / 4H    /
C
C...   GET MODEL
C
      KSWPLT = 0
      KEOF   = 0
C
  100 CONTINUE
      CALL CARDIN( MEOF )
      IF( MEOF .NE. 0 )             GO TO 8000
C
C
C...   PLOT  OR  ANALYSIS
C
      IF( KSOLV.EQ.0 )              GO TO 300
C
      IF( KOUT.EQ.0 )               GO TO 8000
      KSWPLT = KSWPLT + 1
      IF ( KSWPLT .EQ. 1 )  CALL PLTBGN
      CALL MPLOT   ( LABEL(1,MODEL) )
      DO 200 J= 1, 5
      IF( MLENG(J).EQ.0 )           GO TO 200
      CALL DPLOT   ( LABEL(1,MODEL), NTITLE, TABLE(1,1,J), TABLE(1,2,J),
     +               MLENG(J), 3 , IFLAG, J )
  200 CONTINUE
```

Appendix F (Continued)

```
      GO TO 8000
C
C...    CALCULATION
C
  300 CONTINUE
      GO TO ( 1000, 2000, 3000)   ,  MODEL
C
C...  VERTICAL
 1000 CONTINUE
      CALL  VERT
      GO TO 5000
C
C... HORIZONTAL
 2000 CONTINUE
      CALL  HORIZ
      GO TO 5000
C
C... CORNER
 3000 CONTINUE
      CALL  CONER
C
C
C...      END OF ANALYSIS
C
 5000 CONTINUE
      IF ( IFLAG .EQ. 0 ) NDIM = NDIMM(1)
      IF ( IFLAG .EQ. 1 ) NDIM = NDIMM(2)
      IF ( IFLAG .EQ. 0 ) FMETER = 0.001
      IF ( IFLAG .EQ. 1 ) FMETER = 0.01
      ENERGO = ENERGO * FMETER
      WRITE(MTOUT,9010)       NTITLE
 9010 FORMAT(1H1/5X,18A4/)
      WRITE(MTOUT,9020)       (LABEL(I,MODEL),I=1,3), ENERGO
 9020 FORMAT(5X,13HMODEL TYPE = ,3A4,10X,14HTOTAL ENERGY =, F12.2,
     +    1X, 6H(KG-M) // )
C
      WRITE(MTOUT,9030)
 9030 FORMAT(10X,4HSTEP,12H      DEPTH ,4X,11H      FORCE,9X,6HENERGY,
     +      3X,12HACCELERATION )
      WRITE(MTOUT,9035)  NDIM
 9035 FORMAT(22X, 1H(, A2, 1H), 11X, 4H(KG), 9X, 6H(KG-M), 12X, 3H(G)/)
      DO 5200 J= 1, MXSTEP
      ENERG(J) = ENERG(J) * FMETER
      WRITE(MTOUT,9040)       J, DEPTH(J), FORCE(J), ENERG(J), ACCEL(J)
 9040 FORMAT(10X,I4,F12.2,3X,F12.2,3X,F12.2)
      IF ( MOD( J ,50) .NE. 0 ) GO TO 5200
      WRITE(MTOUT,9050)       NTITLE
 9050 FORMAT(1H1/5X,18A4//
     +       10X,4HSTEP,12H      DEPTH ,4X,11H      FORCE,9X,6HENERGY,
     +       3X,12HACCELERATION )
      WRITE(MTOUT,9035)  NDIM
 5200 CONTINUE
C
C...    PLOT RESULT
C
 5500 CONTINUE
      IF( KOUT.EQ.0 )               GO TO 100
      KSWPLT = KSWPLT + 1
```

Appendix F (Continued)

```
      IF ( KSWPLT .EQ. 1 )           CALL PLTBGN
      CALL UPLOT  ( LABEL(1,MODEL), DEPTH(MXSTEP), ACCEL(MXSTEP) )
C
      CALL DPLOT  ( LABEL(1,MODEL), NTITLE, DEPTH, ENERG, MXSTEP, 1 ,
     +              IFLAG, 0 )
      CALL DPLOT  ( LABEL(1,MODEL), NTITLE, DEPTH, ACCEL, MXSTEP, 2 ,
     +              IFLAG, 0 )
      CALL DPLOT  ( LABEL(1,MODEL), NTITLE, DEPTH, FORCE, MXSTEP, 5 ,
     +              IFLAG, 0 )
C
      DO 5700 J= 1, 6
      IF( MSUSE(J).EQ.0 )            GO TO 5700
      CALL DPLOT  ( LABEL(1,MODEL), NTITLE, EPSU(1,J), SIGU(1,J),
     +              MXSTEP, 4 , IFLAG, J )
 5700 CONTINUE
      GO TO 100
C
 8000 CONTINUE
      IF( KSWPLT .GT. 0 ) CALL PLTEND
      STOP
      END


      SUBROUTINE    CARDIN (MEOF)
C
C****   /PURPOSE/
C         READ MODEL DATA
C
C***    /OUTPUT/
C         MEOF        = 0  :NORMAL RUN
C                     = 1  :END OF INPUT DATA
C
C***     PROGRAMMED BY JAERI                    1988.01.29
C***     MODIFIED   BY JAERI                    1988.09.19
C***     MODIFIED   BY JAERI                    1989.01.05
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +           RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +           DISP,     FACTK(5), ENERGO,   GRAVIT,   EPMAX,
     +           MTINN,    MTOUT,    MODEL,    IFLAG,    KSOLV,
     +           KOUT,     NPART,    KPART,    NTITLE(18)
      COMMON /MODEL2/
     +           CLENGT,   RADIUS
      COMMON /MATDAT/      TABLE(100,2,5),     MLENG(5)
      COMMON /RESLT/
     +           DEPTH(500),         FORCE(500),         ACCEL(500),
     +           ENERG(500),         ENERGY,             TODISP,
     +           EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +           MSUSE(6)
      COMMON /SIGCNV/      CONV,     MXITER
      COMMON /MODEL1/
     +           S,        C,        S2,       C2
      COMMON /SWITCH/      KEOF,     KKTEST
C
C
```

Appendix F (Continued)

```
      DIMENSION LABEL(2,7),            KOPT(2,3),          KTYPE(2,5),
     +          NAME(2),   KEY(5),     NNVAL(2), NCARD(20), NDIMM(2)
C
      DATA NCARD/ 20*4H     /
      DATA LABEL/ 4HTITL, 4HE    ,    4HRCOO, 4HR    ,
     +            4HZCOO, 4HR    ,    4HTRUS, 4HS    ,
     +            4HRUN , 4H     ,    4HMODE, 4HL    ,
     +            4HMESH, 4H     /
      DATA KOPT/  4HVERT, 4HICAL,    4HHORI, 4HZONT,
     +            4HCORN, 4HER  /
      DATA KTYPE/ 4HSIGA, 1,   4HSIGB, 2,   4HSIGC, 3,   4HSIGD, 4,
     +            4HSIGX, 5/
      DATA KEY/   4HMM  , 4HCM  , 4HPLOT, 4H    , 4HCHEC/
      DATA LCOMM/ 4H****   /
      DATA MFIN/  4HFIN    /
      DATA RAD/ 0.01745329 /
      DATA EPSS/ 1.0E-8 /
      DATA NDIMM / 4HMM  , 4HCM  /
      DATA NDIM  / 4H    /
      DATA KSIGNN/ 4HNONE/
C
C...    FUNCTION
      ABS(Z) = DABS(Z)
      SIN(Z) = DSIN(Z)
      COS(Z) = DCOS(Z)
C
C...     INITIAL VALUES
C
      MTINN   = 2
      MTOUT   = 6
      MXITER  = 20
      DO 80 J= 1, 5
      MSUSE(J) = 0
   80 MLENG(J) = 0
      MSUSE(6) = 0
C
      DO 90 J= 1, 6
      DO 90 I= 1, 100
      EPSU(I,J)= 0.0
   90 SIGU(I,J)= 0.0
C
C...     CARD IMAGE
C
      LINE    = 0
      REWIND MTINN
C
      IF( KEOF.EQ.0 )               GO TO 110
      MEOF    = 1
      RETURN
C
  110 CONTINUE
      READ(55,120,END=150)      NCARD
  120 FORMAT(20A4)
      IF( NCARD(1).EQ.MFIN )        GO TO 160
C
      LINE    = LINE+1
      IF( MOD(LINE,50).NE.1 )       GO TO 130
```

Appendix F (Continued)

```
      WRITE(MTOUT,100)          (I,I=1,8)
  100 FORMAT(1H1 ///34X,34HC R U S H 1   M O D E L   D A T A //
     +              10X,8I10/10X,10H....:....0,7(10H....:....0) )
C
  130 WRITE(MTOUT,140)          LINE, NCARD
  140 FORMAT(1X,I4,5H-     ,20A4)
C
      IF( NCARD(1).EQ.LCOMM )        GO TO 110
      WRITE(MTINN,120)          NCARD
      GO TO 110
C
C
C...    GET A COMMAND.
C
  150 KEOF    = 1
  160 IF( LINE.GT.0 )                GO TO 170
      MEOF    = 1
      RETURN
C
C. (1)
  170 REWIND MTINN
      READ(MTINN,211)           NAME, NTITLE
  211 FORMAT(2A4,2X,17A4,A2)
      LINE    = 1
      IF( NAME(1).NE.LABEL(1,1) )        GO TO 999
      IF( NAME(2).NE.LABEL(2,1) )        GO TO 999
C
C. (2)
      READ(MTINN,221)           NAME, WEIGHT, HEIGHT, ANGLE, DISP, CONV,
     +                          NDATA, EPMAX
  221 FORMAT(2A4,2X,5F10.0,A4,6X,F10.0)
      LINE    = LINE+1
      DO 222 J= 1, 3
      IF( NAME(1).NE.KOPT(1,J) )         GO TO 222
      IF( NAME(2).EQ.KOPT(2,J) )         GO TO 224
  222 CONTINUE
      GO TO 999
C
  224 MODEL   = J
      IF( NDATA.EQ.KEY(1) )              IFLAG   = 0
      IF( NDATA.EQ.KEY(2) )              IFLAG   = 1
      IF( NDATA.EQ.KEY(4) )              IFLAG   = 0
C
      IF( DISP.EQ.0.0 )                  DISP    = 1.0
      IF( CONV.EQ.0.0 )                  CONV    = 1.0D-5
      ENERGO = WEIGHT*HEIGHT
      IF( IFLAG.EQ.0 )                   GRAVIT = 9800.0
      IF( IFLAG.EQ.1 )                   GRAVIT = 980.0
      IF( IFLAG.EQ.0 )                   NDIM    = NDIMM(1)
      IF( IFLAG.EQ.1 )                   NDIM    = NDIMM(2)
      IF( EPMAX.EQ.0.0 )                 EPMAX   = 1.0
      IF( EPMAX.LE.0.0 .OR. EPMAX.GT.1.0 )       GO TO 996
C
C. (2A)
      READ(MTINN,225)           CLENGT, RADIUS, KTYPED
  225 FORMAT( 10X,  2F10.0,5X,A4)
      LINE    = LINE+1
      IF( CLENGT.LT.0.0 )                GO TO 992
```

Appendix F (Continued)

```
      IF( RADIUS.LT.0.0 )                GO TO 994
      KSIGDD =0
      IF( KTYPED.EQ.KTYPE(1,4) )         KSIGDD =0
      IF( KTYPED.EQ.KSIGNN )             KSIGDD =1
C
C. (3)
      READ(MTINN,227)           NAME, NAME1, NDATA
  227 FORMAT(2A4,2X,A4,1X,A4)
      LINE    = LINE+1
      DO 228 J= 5, 6
      IF( NAME(1).NE.LABEL(1,J) )        GO TO 228
      IF( NAME(2).EQ.LABEL(2,J) )        GO TO 229
  228 CONTINUE
      GO TO 999
C
  229 KSOLV   = J-5
      KOUT    = 0
      IF( NAME1.EQ.KEY(3) )              KOUT    = 1
      KKTEST = 0
      IF( NDATA.EQ.KEY(5) )              KKTEST = 1
C
C. (4)
      READ(MTINN,231)           NAME, (RCOOR(I),I=1,5)
  231 FORMAT(2A4,2X,7F10.0/10X,7F10.0)
      LINE    = LINE+1
      IF( NAME(1).NE.LABEL(1,2) )        GO TO 999
      IF( NAME(2).NE.LABEL(2,2) )        GO TO 999
C
C. (5)
      READ(MTINN,231)           NAME, (ZCOOR(I),I=1,9)
      LINE    = LINE+2
      IF( NAME(1).NE.LABEL(1,3) )        GO TO 999
      IF( NAME(2).NE.LABEL(2,3) )        GO TO 999
      DO 233 J= 2, 4
      IF( RCOOR(J).LT.RCOOR(J-1) )       GO TO 999
      IF( ZCOOR(J).LT.ZCOOR(J-1) )       GO TO 999
  233 CONTINUE
      IF( RCOOR(5).LT.RCOOR(4) )         GO TO 999
C
C. (6)
      READ(MTINN,251)           NAME, NNVAL, FACTK
  251 FORMAT(2A4,2X,2I5,5F10.0)
      LINE    = LINE+1
      IF( NAME(1).NE.LABEL(1,7) )        GO TO 999
      IF( NAME(2).NE.LABEL(2,7) )        GO TO 999
C
      IF( MODEL.NE.1 )                   GO TO 253
      DO 252 J= 1, 5
      IF( FACTK(J).LT.0.0 )              GO TO 997
      IF( FACTK(J).GT.1.0 )              GO TO 997
      IF( FACTK(J).EQ.0.0 )              FACTK(J) =1.0
  252 CONTINUE
      MXFCTK = 5
      GO TO 260
C
  253 IF( MODEL.NE.2 )                   GO TO 255
      NPART   = NNVAL(1)
      IF( NPART.EQ.0 )                   NPART   = 100
```

```
      IF( NPART.GT.300 )                   NPART  = 300
      DO 254 J= 1, 4
      IF( FACTK(J).LT.0.0 )                GO TO 997
      IF( FACTK(J).GT.1.0 )                GO TO 997
      IF( FACTK(J).EQ.0.0 )                FACTK(J) =1.0
  254 CONTINUE
      MXFCTK = 4
      GO TO 260
C
  255 NPART  = NNVAL(1)
      KPART  = NNVAL(2)
      IF( NPART.EQ.0 )                     NPART  = 100
      IF( NPART.GT.300 )                   NPART  = 300
      IF( KPART.EQ.0 )                     KPART  = 200
      IF( KPART.GT.400 )                   KPART  = 400
      DO 256 J= 1, 4
      IF( FACTK(J).LT.0.0 )                GO TO 997
      IF( FACTK(J).GT.1.0 )                GO TO 997
      IF( FACTK(J).EQ.0.0 )                FACTK(J) =1.0
  256 CONTINUE
      FACTK(5) = 1.0
      MXFCTK = 4
C
C. (7)
  260 CONTINUE
      READ(MTINN,271,END=280) NAME, NN, FACT
  271 FORMAT(2A4,2X,I5,5X,F10.0)
      LINE    = LINE+1
      DO 272 J= 1, 5
      IF( NAME(1).EQ.KTYPE(1,J) )          GO TO 274
  272 CONTINUE
      GO TO 999
C
  274 ID      = KTYPE(2,J)
      IF( NN.EQ.0 )                        GO TO 999
      IF( FACT.LT.0.0 )                    GO TO 999
      IF( FACT.EQ.0.0 )                    FACT   = 1.0
      MLENG(ID) = NN
C
      READ(MTINN,275)           (TABLE(I,1,ID),TABLE(I,2,ID), I=1,NN)
  275 FORMAT(8F10.0)
      LINE    = LINE+1+(NN-1)/4
      DO 276 I= 1, NN
      TABLE(I,2,ID) = FACT*TABLE(I,2,ID)
  276 CONTINUE
C. DATA OF TABLE
      DO 277 I= 2, NN
      IF( TABLE(I,1,ID).LE.TABLE(I-1,1,ID) )      GO TO 998
  277 CONTINUE
      DO 278 I= 2, NN
      DS      = ABS(TABLE(I,2,ID)-TABLE(I-1,2,ID))
      IF( DS.GT.EPSS )                     GO TO 278
      TABLE(I,2,ID) = TABLE(I,2,ID)+TABLE(I,2,ID)*EPSS
  278 CONTINUE
C
      IF( ID.NE.4 )                        GO TO 260
      IF( KSIGDD.EQ.0 )                    GO TO 260
      MLENG(ID) =0
```

```
      GO TO 260
C
C...
  280 IF( MODEL.EQ.3 )                          GO TO 281
      IF( MLENG(1).EQ.0 )                       GO TO 999
      IF( MLENG(2).EQ.0 )                       GO TO 999
      IF( MLENG(3).EQ.0 )                       GO TO 999
      GO TO 900
C
  281 CONTINUE
      S       = SIN(RAD*ANGLE)
      C       = COS(RAD*ANGLE)
      S2      = S*S
      C2      = C*C
      NEWVVV = 0
      IF( MLENG(5).GT.0 )                       GO TO 900
      NEWVVV = 1
      N1      = MLENG(1)
      N2      = MLENG(2)
      TABLE(N1+1,1,1) = TABLE(N2,1,2)+1.0
      TABLE(N2+1,1,2) = TABLE(N1,1,1)+1.0
      NN      = 0
      N1      = 1
      N2      = 1
C..
  285 IF( N1.GT.MLENG(1) .AND. N2.GT.MLENG(2) )          GO TO 290
      NN      = NN+1
      EPS1    = TABLE(N1,1,1)
      SIG1    = TABLE(N1,2,1)
      EPS2    = TABLE(N2,1,2)
      SIG2    = TABLE(N2,2,2)
      IF( ABS(EPS1-EPS2).GT.EPSS )              GO TO 287
      TABLE(NN,1,5) = EPS1
      TABLE(NN,2,5) = C2*SIG1 + S2*SIG2
      N1      = N1+1
      N2      = N2+1
      GO TO 285
  287 IF( EPS2.GT.EPS1 )                        GO TO 288
      K1      = N1
      IF( N1.GT.MLENG(1) )              K1 = MLENG(1)
      KO      = K1-1
      DX      = TABLE(K1,1,1)-TABLE(KO,1,1)
      DY      = TABLE(K1,2,1)-TABLE(KO,2,1)
      SIG1    = TABLE(KO,2,1) + DY/DX*(EPS2-TABLE(KO,1,1))
      TABLE(NN,1,5) = EPS2
      TABLE(NN,2,5) = C2*SIG1 + S2*SIG2
      N2      = N2+1
      GO TO 285
  288 K1      = N2
      IF( N2.GT.MLENG(2) )              K1 = MLENG(2)
      KO      = K1-1
      DX      = TABLE(K1,1,2)-TABLE(KO,1,2)
      DY      = TABLE(K1,2,2)-TABLE(KO,2,2)
      SIG2    = TABLE(KO,2,2) + DY/DX*(EPS1-TABLE(KO,1,2))
      TABLE(NN,1,5) = EPS1
      TABLE(NN,2,5) = C2*SIG1 + S2*SIG2
      N1      = N1+1
      GO TO 285
```

Appendix F (Continued)

```
  290 MLENG(5) = NN
C
C...     END OF INPUT CARD.
C
  900 CONTINUE
      WRITE(MTOUT,910)         WEIGHT, HEIGHT, NDIM,  ANGLE, DISP, NDIM
  910 FORMAT(/////11X,22H...... MODEL WEIGHT  =,F10.2,2X,4H(KG)      /
     +            11X,22H...... MODEL HEIGHT  =,F10.2,2X,1H(,A2,1H) /
     +            11X,22H...... CORNER ANGLE  =,F10.2,2X,6H(DEG.)    /
     +            11X,22H...... INCREMENT DISP=,F10.2,2X,1H(,A2,1H) /)
      WRITE(MTOUT,920)         (FACTK(J),J=1,MXFCTK)
  920 FORMAT(/    11X,22H...... K-FACTOR       =,5F10.4)
      IF( MODEL.NE.3 )                    RETURN
      IF( NEWVVV.EQ.0 )                   RETURN
      WRITE(MTOUT,930)
      WRITE(MTOUT,940)     (TABLE(I,1,5),TABLE(I,2,5),I=1,NN)
  930 FORMAT(//)
CJ940 FORMAT(12H (EPS,SIG) =,2F12.5,3X,2F12.5,3X,2F12.5,3X,2F12.5)
  940 FORMAT(12H (EPS,SIG) =,2F12.5,3X,2F12.5,3X,2F12.5)
      RETURN
C
C...     ERROR LENGTH
  992 WRITE(MTOUT,9921)        CLENGT
 9921 FORMAT(40H **** ERROR  LENGTH VALUE.        LENGTH=,F10.4)
      CALL EXIT
      RETURN
C
C...     ERROR RADIUS
  994 WRITE(MTOUT,9924)        RADIUS
 9924 FORMAT(40H **** ERROR  RADIUS VALUE.        RADIUS=,F10.4)
      CALL EXIT
      RETURN
C
C...     ERROR EPMAX
  996 WRITE(MTOUT,9961)        EPMAX
 9961 FORMAT(40H **** ERROR  EPMAX VALUE.         EPMAX=,F10.4)
      CALL EXIT
      RETURN
C
C...     ERROR K-FACTOR
  997 WRITE(MTOUT,9971)        J, FACTK(J)
 9971 FORMAT(40H **** ERROR K-FACTOR VALUE.    POSITION=,I2,5X,6HVALUE=,
     +  F10.4)
      CALL EXIT
      RETURN
C
C...     ERROR TABLE
  998 WRITE(MTOUT,9981)        KTYPE(1,J), I
 9981 FORMAT(40H **** ERROR STRAIN TABLE.         LABEL=,A4,5X,6HCOLUM=,
     +  I4)
      CALL EXIT
      RETURN
C
C...     ERROR FORMAT CARD.
  999 WRITE(MTOUT,9991)        LINE
 9991 FORMAT(40H **** ERROR CRAD FOUND.        CARD NO. =,I5)
      CALL EXIT
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    VERT
C
C****   /PURPOSE/
C         CALCULATION ENERGRY OF VERTICAL SHPAE
C
C***     PROGRAMMED BY JAERI                     1988.01.29
C***     MODIFIED   BY JAERI                     1988.09.19
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +          RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +          DISP,     FACTK(5), ENERGO,   GRAVIT,   EPMAX,
     +          MTINN,    MTOUT,    MODEL,    IFLAG,    KSOLV,
     +          KOUT,     NPART,    KPART,    NTITLE(18)
      COMMON /MATDAT/     TABLE(100,2,5),     MLENG(5)
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
C
      DIMENSION P(5),     DL(2),    A(5),     ID(2)
      DATA PAI/ 3.14159265 /
C
C...     INITIAL STATUS
C
      MXSTEP = 0
      TODISP = 0.0
      MINI   = 0
      DISPX  = DISP
C
      A(1)   = PAI*RCOOR(1)**2
      A(2)   = PAI*(RCOOR(2)**2-RCOOR(1)**2)
      A(3)   = PAI*(RCOOR(3)**2-RCOOR(2)**2)
      A(4)   = PAI*(RCOOR(4)**2-RCOOR(3)**2)
      A(5)   = PAI*(RCOOR(5)**2-RCOOR(4)**2)
CRC   WRITE(6,1)      A
CRC 1 FORMAT(6H AREA=,F12.2)
C
      DL(1)  = ZCOOR(3)
      DL(2)  = ZCOOR(4)-ZCOOR(3)
      ID(1)  = 2
      ID(2)  = 3
      KS3    = 2
C
C...     GET CURRENT FORCE
C
  100 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )          GO TO 1000
      TODISP = TODISP + DISPX
C
C. (O-R1)
      EPS     = TODISP/ZCOOR(2)
      CALL GETSIG ( EPS, 4, SIGM )
      P(1)    = A(1)*SIGM
C
```

Appendix F (Continued)

```
C. (R1-R2)
      EPS    = TODISP/ZCOOR(2)
      CALL GETSIG ( EPS, 1, SIGM )
      P(2)   = A(2)*SIGM
C
C. (R2-R3)
      CALL GETSIG ( EPS, 2, SIGM )
      P(3)   = A(3)*SIGM
C
C. (R3-R4)
      CALL MULTI  ( TODISP, DL, ID, KS3, SIGM , IERR )
      IF( IERR.NE.0 )              GO TO 2000
      P(4)   = A(4)*SIGM
C
C. (R4-R5)
      P(5)   = A(5)*SIGM
C
C...    SUM OF EACH FORCES
C
      SUM    = 0.0
      DO 300 J= 1, 5
  300 SUM    = SUM + P(J)*FACTK(J)
      ENERGY = SUM*DISPX
C
      DEPTH(MXSTEP) = TODISP
      FORCE(MXSTEP) = SUM
      ACCEL(MXSTEP) = SUM/WEIGHT
      ENERG(MXSTEP) = ENERGY
      IF( MXSTEP.NE.1 )            ENERG(MXSTEP) =ENERG(MXSTEP-1)+ENERGY
C
      IF( ENERG(MXSTEP).LT.ENERGO )GO TO 100
      IF( MINI.NE.0 )              GO TO 800
      MINI   = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      GO TO 100
C
C...    INTERPOLATION OF LAST STEP
C
  800 CONTINUE
      CALL FITSTP
C
C...    END OF VERTICAL
C
 1000 CONTINUE
      IF( MXSTEP.GT.500 )          MXSTEP  = 500
      RETURN
C
C***    ERROR
 2000 CONTINUE
      WRITE(MTOUT,2099)       MXSTEP, TODISP, KS3, ID
 2099 FORMAT(// 29H ***  MULTI-MATERIAL ERROR.  ,5X,7H STEP =,I5,
     + 5X,7HDEPTH =,F10.4,5X,11HNO. OF MAT=,I3,5X,5I3)
      CALL EXIT
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    HORIZ
C
C****   /PURPOSE/
C         CALCULATION ENERGRY OF HORIZONTAL SHAPE
C
C***     PROGRAMMED BY JAERI                  1988.02.05
C***     MODIFIED   BY JAERI                  1988.09.19
C***     MODIFIED   BY JAERI                  1994.08.01
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +          RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +          DISP,     FACTK(5), ENERGO,   GRAVIT,   EPMAX,
     +          MTINN,    MTOUT,    MODEL,    IFLAG,    KSOLV,
     +          KOUT,     NPART,    KPART,    NTITLE(18)
      COMMON /MATDAT/     TABLE(100,2,5),     MLENG(5)
      COMMON /SWITCH/     KEOF,     KKTEST
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
C
      DIMENSION P(4),     E(4),     DZ(4),    R(5),     RO(3),
     +          ID(3),    MESH1(3), MESH2(2), XLENG(5,300,4)
C
C...    FUNCTION
      SQRT(Z) = DSQRT(Z)
C
C...    INITIAL STATUS
C
      MXSTEP = 0
      TODISP = 0.0
      MINI   = 0
      DISPX  = DISP
C
C..  GET (Z) LENGTH
      DZ(1)   = ZCOOR(1)
      DZ(2)   = ZCOOR(2)-ZCOOR(1)
      DZ(3)   = ZCOOR(3)-ZCOOR(2)
      DZ(4)   = ZCOOR(4)-ZCOOR(3)
      IF( ZCOOR(5).GT.ZCOOR(4) )   GO TO 30
      DO 20 I= 1, 4
   20 DZ(I)   = 2.0*DZ(I)
      GO TO 50
   30 DZ(1)   = DZ(1) + ZCOOR(9)-ZCOOR(8)
      DZ(2)   = DZ(2) + ZCOOR(8)-ZCOOR(7)
      DZ(3)   = DZ(3) + ZCOOR(7)-ZCOOR(6)
      DZ(4)   = DZ(4) + ZCOOR(6)-ZCOOR(5)
C
C..  GET (R) LENGTH
   50 R(1)    = RCOOR(1)
      R(2)    = RCOOR(2)-RCOOR(1)
      R(3)    = RCOOR(5)-RCOOR(2)
      R(4)    = RCOOR(3)
      R(5)    = RCOOR(5)-RCOOR(3)
      DPART   = NPART
```

Appendix F (Continued)

```
      DX      = RCOOR(4)/DPART
C
      MESH1(1) = R(1)/DX + 1
      MESH1(2) = R(2)/DX + 1
      MESH1(3) = NPART - (MESH1(1)+MESH1(2))
      MESH2(1) = R(4)/DX + 1
      MESH2(2) = NPART - MESH2(1)
      IF( KKTEST.NE.0 )              WRITE(6,1)     MESH1, MESH2, DZ
    1 FORMAT(6H MESH=,3I5,5X,2I5,5X,4F10.3)
C
C..   (C-C) MESH DATA
      MLOOP   = 0
      RO(1)   = 0.0
      RO(2)   = RCOOR(1)
      RO(3)   = RCOOR(2)
C
      DO 150 J= 1, 3
      LOOP    = MESH1(J)
      DPART   = LOOP
      DX      = R(J)/DPART
      XX      = RO(J) + 0.5*DX
      DO 100 I= 1, LOOP
      XR5     = SQRT( RCOOR(5)**2-XX**2 )
      H       = RCOOR(5)-XR5
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP,1) = DX
      XLENG(2,MLOOP,1) = H
      XLENG(1,MLOOP,2) = DX
      XLENG(2,MLOOP,2) = H
C
      GO TO ( 91, 92, 93)     , J
   91 XR1     = SQRT( RCOOR(1)**2-XX**2 )
      XR2     = SQRT( RCOOR(2)**2-XX**2 )
      IF( MLENG(4).EQ.0 )            GO TO 99
      XLENG(3,MLOOP,1) = 2.0*(XR5-XR2)
      XLENG(4,MLOOP,1) = 2.0*(XR2-XR1)
      XLENG(5,MLOOP,1) = 2.0*XR1
      XLENG(3,MLOOP,2) = 2.0*(XR5-XR2)
      XLENG(4,MLOOP,2) = 2.0*(XR2-XR1)
      XLENG(5,MLOOP,2) = 2.0*XR1
      GO TO 100
   99 XLENG(3,MLOOP,1) = XR5-XR2
      XLENG(4,MLOOP,1) = XR2-XR1
      XLENG(5,MLOOP,1) = 0.0
      XLENG(3,MLOOP,2) = XR5-XR2
      XLENG(4,MLOOP,2) = XR2-XR1
      XLENG(5,MLOOP,2) = 0.0
      GO TO 100
   92 XR2     = SQRT( RCOOR(2)**2-XX**2 )
      XLENG(3,MLOOP,1) = 2.0*(XR5-XR2)
      XLENG(4,MLOOP,1) = 2.0*XR2
      XLENG(5,MLOOP,1) = 0.0
      XLENG(3,MLOOP,2) = 2.0*(XR5-XR2)
      XLENG(4,MLOOP,2) = 2.0*XR2
      XLENG(5,MLOOP,2) = 0.0
      GO TO 100
   93 XLENG(3,MLOOP,1) = 2.0*XR5
      XLENG(4,MLOOP,1) = 0.0
```

Appendix F (Continued)

```
      XLENG(5,MLOOP,1) = 0.0
      XLENG(3,MLOOP,2) = 2.0*XR5
      XLENG(4,MLOOP,2) = 0.0
      XLENG(5,MLOOP,2) = 0.0
C
  100 XX      = XX + DX
  150 CONTINUE
C
C..   (D-D) MESH
      MLOOP   = 0
      RO(2)   = RCOOR(3)
C
      DO 250 J= 1, 2
      LOOP    = MESH2(J)
      DPART   = LOOP
      DX      = R(J+3)/DPART
      XX      = RO(J) + 0.5*DX
      DO 200 I= 1, LOOP
      XR5     = SQRT( RCOOR(5)**2-XX**2 )
      H       = RCOOR(5)-XR5
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP,3) = DX
      XLENG(2,MLOOP,3) = H
      XLENG(1,MLOOP,4) = DX
      XLENG(2,MLOOP,4) = H
C
      IF( J.NE.1)                    GO TO 190
      XR3     = SQRT( RCOOR(3)**2-XX**2 )
      XLENG(3,MLOOP,3) = XR5-XR3
      XLENG(4,MLOOP,3) = 0.0
      XLENG(5,MLOOP,3) = 0.0
      XLENG(3,MLOOP,4) = XR5-XR3
      XLENG(4,MLOOP,4) = 0.0
      XLENG(5,MLOOP,4) = 0.0
      GO TO 200
  190 XLENG(3,MLOOP,3) = 2.0*XR5
      XLENG(4,MLOOP,3) = 0.0
      XLENG(5,MLOOP,3) = 0.0
      XLENG(3,MLOOP,4) = 2.0*XR5
      XLENG(4,MLOOP,4) = 0.0
      XLENG(5,MLOOP,4) = 0.0
C
  200 XX      = XX + DX
  250 CONTINUE
C
C
C...     GET CURRENT FORCE
C
 1000 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )            GO TO 2000
      TODISP = TODISP + DISPX
C
C
C..   Z = 0.0 - Z1
      P(1)    = 0.0
      E(1)    = 0.0
      MLOOP   = 0
```

```
      DO 600 J= 1, 3
      LOOP    = MESH1(J)
      GO TO ( 571, 581, 591 )      , J
C
  571 ID(1)   = 2
      ID(2)   = 1
      ID(3)   = 4
      KS3     = 3
      IF( MLENG(4).EQ.0 )          KS3    = 2
      DO 572 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,1)
      IF( U.LE.0.0 )               GO TO 572
      CALL MULTI  ( U, XLENG(3,MLOOP,1), ID, KS3, SIGM, IERR )
      IF( IERR.NE.0 )              GO TO 3000
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(1)    = P(1) + SIGM*XLENG(1,MLOOP,1)
      E(1)    = E(1) + SIGM*XLENG(1,MLOOP,1)*UO
  572 CONTINUE
      GO TO 600
  581 ID(1)   = 2
      ID(2)   = 1
      KS3     = 2
      DO 582 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,1)
      IF( U.LE.0.0 )               GO TO 582
      CALL MULTI  ( U, XLENG(3,MLOOP,1), ID, KS3, SIGM, IERR )
      IF( IERR.NE.0 )              GO TO 3000
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(1)    = P(1) + SIGM*XLENG(1,MLOOP,1)
      E(1)    = E(1) + SIGM*XLENG(1,MLOOP,1)*UO
  582 CONTINUE
      GO TO 600
  591 CONTINUE
      DO 592 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,1)
      IF( U.LE.0.0 )               GO TO 592
      EPS     = U/XLENG(3,MLOOP,1)
      CALL GETSIG ( EPS, 2, SIGM )
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(1)    = P(1) + SIGM*XLENG(1,MLOOP,1)
      E(1)    = E(1) + SIGM*XLENG(1,MLOOP,1)*UO
  592 CONTINUE
C
  600 CONTINUE
C
C
C..   Z = Z1- Z2
      P(2)    = 0.0
      E(2)    = 0.0
      MLOOP   = 0
      DO 700 J= 1, 3
      LOOP    = MESH1(J)
```

```
      GO TO ( 671, 681, 691 )      , J
C
  671 ID(1)   = 2
      ID(2)   = 1
      ID(3)   = 4
      KS3     = 3
      IF( MLENG(4).EQ.0 )          KS3    = 2
      DO 672 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,2)
      IF( U.LE.0.0 )               GO TO 672
      CALL MULTI  ( U, XLENG(3,MLOOP,2), ID, KS3, SIGM, IERR )
      IF( IERR.NE.0 )              GO TO 3000
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(2)    = P(2) + SIGM*XLENG(1,MLOOP,2)
      E(2)    = E(2) + SIGM*XLENG(1,MLOOP,2)*UO
  672 CONTINUE
      GO TO 700
  681 ID(1)   = 2
      ID(2)   = 1
      KS3     = 2
      DO 682 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,2)
      IF( U.LE.0.0 )               GO TO 682
      CALL MULTI  ( U, XLENG(3,MLOOP,2), ID, KS3, SIGM, IERR )
      IF( IERR.NE.0 )              GO TO 3000
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(2)    = P(2) + SIGM*XLENG(1,MLOOP,2)
      E(2)    = E(2) + SIGM*XLENG(1,MLOOP,2)*UO
  682 CONTINUE
      GO TO 700
  691 CONTINUE
      DO 692 I= 1, LOOP
      MLOOP   = MLOOP+1
      U       = TODISP-XLENG(2,MLOOP,2)
      IF( U.LE.0.0 )               GO TO 692
      EPS     = U/XLENG(3,MLOOP,2)
      CALL GETSIG ( EPS, 2, SIGM )
      UO      = DISPX
      IF( U.LT.DISPX )             UO      = U
      P(2)    = P(2) + SIGM*XLENG(1,MLOOP,2)
      E(2)    = E(2) + SIGM*XLENG(1,MLOOP,2)*UO
  692 CONTINUE
C
  700 CONTINUE
C
C
C..   Z = Z2- Z3
      P(3)    = 0.0
      E(3)    = 0.0
      MLOOP   = 0
C
      DO 800 J= 1, 2
      LOOP    = MESH2(J)
      DO 782 I= 1, LOOP
```

Appendix F (Continued)

```
      MLOOP   = MLOOP+1
      U        = TODISP-XLENG(2,MLOOP,3)
      IF( U.LE.0.0 )                   GO TO 782
      EPS     = U/XLENG(3,MLOOP,3)
      CALL GETSIG ( EPS, 2, SIGM )
      UO      = DISPX
      IF( U.LT.DISPX )                 UO      = U
      P(3)    = P(3) + SIGM*XLENG(1,MLOOP,3)
      E(3)    = E(3) + SIGM*XLENG(1,MLOOP,3)*UO
  782 CONTINUE
  800 CONTINUE
C
C
C..  Z = Z3- Z4
      P(4)    = 0.0
      E(4)    = 0.0
      MLOOP   = 0
C
      DO 900 J= 1, 2
      LOOP    = MESH2(J)
      DO 882 I= 1, LOOP
      MLOOP   = MLOOP+1
      U        = TODISP-XLENG(2,MLOOP,4)
      IF( U.LE.0.0 )                   GO TO 882
      EPS     = U/XLENG(3,MLOOP,4)
      CALL GETSIG ( EPS, 3, SIGM )
      UO      = DISPX
      IF( U.LT.DISPX )                 UO      = U
      P(4)    = P(4) + SIGM*XLENG(1,MLOOP,4)
      E(4)    = E(4) + SIGM*XLENG(1,MLOOP,4)*UO
  882 CONTINUE
  900 CONTINUE
C
C
C...   SUM OF EACH FORCES
C
      SUM     = 0.0
      TUM     = 0.0
      DO 1300 J= 1, 4
      SUM     = SUM + 2.0*P(J)*DZ(J)*FACTK(J)
 1300 TUM     = TUM + 2.0*E(J)*DZ(J)*FACTK(J)
      ENERGY = TUM
C
      DEPTH(MXSTEP) = TODISP
      FORCE(MXSTEP) = SUM
      ACCEL(MXSTEP) = SUM/WEIGHT
      ENERG(MXSTEP) = ENERGY
      IF( MXSTEP.NE.1 )                ENERG(MXSTEP) =ENERG(MXSTEP-1)+ENERGY
C
      IF( ENERG(MXSTEP).LT.ENERGO )GO TO 1000
      IF( MINI.NE.0 )                  GO TO 1800
      MINI    = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      GO TO 1000
C
C...    INTERPOLATION OF LAST STEP
```

Appendix F (Continued)

```
C
 1800 CONTINUE
      CALL FITSTP
C
C...    END OF HORIZONTAL
C
 2000 CONTINUE
      IF( MXSTEP.GT.500 )           MXSTEP  = 500
      RETURN
C
C***    ERROR
 3000 CONTINUE
      WRITE(MTOUT,3099)        MXSTEP, TODISP, KS3, ID
 3099 FORMAT(// 29H ***  MULTI-MATERIAL ERROR.  ,5X,7H STEP =,I5,
     + 5X,7HDEPTH =,F10.4,5X,11HNO. OF MAT=,I3,5X,5I3)
      CALL EXIT
      RETURN
      END


      SUBROUTINE    CONER
C
C****    /PURPOSE/
C          CALCULATION ENERGRY OF CORNER SHAPE
C
C***     PROGRAMMED BY JAERI                    1988.02.09
C***     MODIFIED   BY JAERI                    1988.09.19
C***     MODIFIED   BY JAERI                    1989.01.05
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +          RCOOR(5), ZCOOR(9), WEIGHT,  HEIGHT,   ANGLE,
     +          DISP,     FACTK(5), ENERGO,  GRAVIT,   EPMAX,
     +          MTINN,    MTOUT,    MODEL,   IFLAG,    KSOLV,
     +          KOUT,     NPART,    KPART,   NTITLE(18)
      COMMON /MATDAT/     TABLE(100,2,5),    MLENG(5)
      COMMON /MODEL1/
     +          S,        C,        S2,      C2
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
      COMMON /SWITCH/     KEOF,     KKTEST
C
      DIMENSION ST(2,18),
     +          MESH(3,400),        SLICE(6,400),       XLENG(3,40000)
C
C...     FUNCTION
      SQRT(Z) = DSQRT(Z)
C
C...     INITIAL STATUS
C
      MXSTEP = 0
      TODISP = 0.0
      MINI   = 0
```

```
      DISPX  = DISP
C
C...    GET (S,T) COORD
C
      ST(1, 1) = -RCOOR(5)*C
      ST(2, 1) =  RCOOR(5)*S
      ST(1, 2) = -RCOOR(1)*C
      ST(2, 2) =  RCOOR(1)*S
      ST(1, 3) =  RCOOR(1)*C
      ST(2, 3) = -RCOOR(1)*S
      ST(1, 4) =  RCOOR(5)*C
      ST(2, 4) = -RCOOR(5)*S
      ST(1, 5) = -RCOOR(3)*C+ZCOOR(2)*S
      ST(2, 5) =  RCOOR(3)*S+ZCOOR(2)*C
      ST(1, 6) = -RCOOR(1)*C+ZCOOR(2)*S
      ST(2, 6) =  RCOOR(1)*S+ZCOOR(2)*C
      ST(1, 7) =  RCOOR(1)*C+ZCOOR(2)*S
      ST(2, 7) = -RCOOR(1)*S+ZCOOR(2)*C
      ST(1, 8) =  RCOOR(3)*C+ZCOOR(2)*S
      ST(2, 8) = -RCOOR(3)*S+ZCOOR(2)*C
      ST(1, 9) = -RCOOR(5)*C+ZCOOR(4)*S
      ST(2, 9) =  RCOOR(5)*S+ZCOOR(4)*C
      ST(1,10) = -RCOOR(3)*C+ZCOOR(4)*S
      ST(2,10) =  RCOOR(3)*S+ZCOOR(4)*C
      ST(1,11) =  RCOOR(3)*C+ZCOOR(4)*S
      ST(2,11) = -RCOOR(3)*S+ZCOOR(4)*C
      ST(1,12) =  RCOOR(5)*C+ZCOOR(4)*S
      ST(2,12) = -RCOOR(5)*S+ZCOOR(4)*C
      ST(1,13) = -RCOOR(1)*C+ZCOOR(1)*S
      ST(2,13) =  RCOOR(1)*S+ZCOOR(1)*C
      ST(1,14) =  RCOOR(1)*C+ZCOOR(1)*S
      ST(2,14) = -RCOOR(1)*S+ZCOOR(1)*C
      ST(1,15) =  ZCOOR(2)*S
      ST(2,15) =  ZCOOR(2)*C
      ST(1,16) =  RCOOR(4)*C+ZCOOR(4)*S
      ST(2,16) = -RCOOR(4)*S+ZCOOR(4)*C
      ST(1,17) = -RCOOR(5)*C+ZCOOR(1)*S
      ST(2,17) =  RCOOR(5)*S+ZCOOR(1)*C
      ST(1,18) = -RCOOR(5)*C+ZCOOR(2)*S
      ST(2,18) =  RCOOR(5)*S+ZCOOR(2)*C
      IF( KKTEST.NE.0 )             WRITE(6,1)     ST
    1 FORMAT(6H ST  =,10F12.4)
C
C..  DS MESH SIZE
      S1     = ST(1,1)
      S2     = ST(1,4)
      S3     = ST(1,12)
      DPART  = KPART
      DX     = (S3-S1)/DPART
      M1     = (S2-S1)/DX
      IF( ANGLE.LE.45.0 )           M1   =  M1+1
      IF( ANGLE.GT.45.0 )           M1   =  M1-1
C
C.   C:S = M1:(KPART-M1)
      IF( ANGLE.GT.20.0 .AND. ANGLE.LT.70.0 )     GO TO 10
      DX1    = C*DPART/(C+S)
      M1     = DX1+0.5
   10 M2     = KPART-M1
C
C..  SIN/COS
       CS      = C/S
       SC      = S/C
       S2X     = 1.0/(S*C)
       SX      = 1.0/S
       R5      = RCOOR(5)**2
       R5MAX   = RCOOR(5)*SX
       B       = R5MAX**2
       R3      = RCOOR(3)**2
       R3MAX   = RCOOR(3)*SX
       G       = R3MAX**2
C
       DPART   = M1
       DX1     = (S2-S1)/DPART
       DPART   = M2
       DX2     = (S3-S2)/DPART
       DPART   = NPART
       DX0     = RCOOR(5)/DPART
C
       M6M1    = M1
       M6M2    = M1+1
C
       WRITE(6,2)       M1, DX1, M2, DX2, DX0
     2 FORMAT(1H0, 8H L-PART=,I5,F10.4,10H   R-PART=,I5,F10.4,10X,
     +        6H   DX=,F10.4  //)
C
C
C..  MESH DATA
       MFACE   = 0
CRC    XX      = S1 + 0.5*DX1
       XX      = S1 + 0.9*DX1
       DO 100 J= 1, M1
C. (H0)
       IF( XX.GE.ST(1,3) )           GO TO 40
       IF( XX.LE.ST(1,2) )           GO TO 40
       IF( MLENG(4).GT.0 )           GO TO 40
       H0      = CS*(XX+RCOOR(1)*C) + (RCOOR(1)+RCOOR(5))*S
       IF( XX.GT.ST(1,6) )           H0      = 100.0*ZCOOR(4)
       GO TO 50
    40 H0      = RCOOR(5)*S - SC*XX
C
C. (KTYPE) AND (XL)
    50 CONTINUE
       IF( XX.GT.ST(1,5) )           GO TO 60
       IF( MLENG(4).GT.0 )           GO TO 55
C
       IF( XX.GE.ST(1,2) .AND. XX.LE.ST(1,3) )         GO TO 100
       IF( XX.GT.0.0 )               GO TO 53
       KTYPE   = 1
       KFNO    = 5
       XL1     = 0.0
       IF( XX.LE.ST(1,9) )           XL2     = R5MAX
       IF( XX.GT.ST(1,9) )           XL2     = S2X*(ZCOOR(4)*S-XX)
       H1      = -S2X*XX
       IF( XX.GT.ST(1,2) )           H1      = RCOOR(1)*SX
       GO TO 90
    53 KTYPE   = 3
```

```
      KFNO   = 1
      IF( XX.GT.ST(1,14) )          KFNO   = 2
      XL2    = 0.0
      XL1    = S2X*XX
      H1     = RCOOR(1)*SX
      GO TO 90
C
   55 KFNO   = 5
      IF( XX.LE.ST(1,18) )          KFNO   = 2
      IF( XX.LE.ST(1,17) )          KFNO   = 1
      IF( XX.GT.0.0 )               GO TO 57
      KTYPE  = 1
      XL1    = 0.0
      H1     = -S2X*XX
      GO TO 58
   57 KTYPE  = 2
      H1     = 0.0
      XL1    = S2X*XX
   58 IF( XX.LE.ST(1,9) )           XL2    = R5MAX
      IF( XX.GT.ST(1,9) )           XL2    = S2X*(ZCOOR(4)*S-XX)
      GO TO 90
C..   ( P5(S)<S<Q0(S) )
   60 IF( XX.GT.ST(1,15) )          GO TO 70
      XL2    = S2X*(ZCOOR(2)*S-XX)
      KFNO   = 5
      IF( MLENG(4).GT.0 )           GO TO 65
C
      IF( XX.GE.ST(1,2) .AND. XX.LE.ST(1,3) )        GO TO 100
      IF( XX.GT.0.0 )               GO TO 63
      KTYPE  = 4
      XL1    = 0.0
      IF( XX.LE.ST(1,6) )           GO TO 61
      H1     = XL2
      GO TO 90
   61 H1     = RCOOR(1)*SX
      IF( XX.LT.ST(1,2) )           H1     = -S2X*XX
      GO TO 90
   63 KTYPE  = 3
      KFNO   = 1
      IF( XX.GT.ST(1,14) )          KFNO   = 2
      XL2    = 0.0
      XL1    = S2X*XX
      H1     = RCOOR(1)*SX
      GO TO 90
C
   65 IF( XX.GT.0.0 )               GO TO 67
      KTYPE  = 4
      XL1    = 0.0
      H1     = -S2X*XX
      GO TO 90
   67 KTYPE  = 6
      H1     = 0.0
      XL1    = S2X*XX
      GO TO 90
C..   ( Q0(S)<S<P11(S) )
   70 IF( XX.GT.ST(1,11) )          GO TO 80
      KTYPE  = 5
      KFNO   = 5
```

```
      XL2    = 0.0
      XL1    = S2X*XX
      IF( MLENG(4).GT.0 )           GO TO 75
      IF( XX.GT.ST(1,7) )           GO TO 75
C
      IF( XX.GE.ST(1,2) .AND. XX.LE.ST(1,3) )        GO TO 100
      IF( XX.LE.ST(1,3) )           GO TO 73
      IF( XX.LE.ST(1,7) )           KFNO   = 2
      IF( XX.LE.ST(1,14) )          KFNO   = 1
      KTYPE  = 3
      H1     = RCOOR(1)*SX
      GO TO 90
   73 KTYPE  = 4
      H1     = RCOOR(1)*SX
      XL2    = H1
      XL1    = 0.0
      GO TO 90
C
   75 H1     = S2X*(XX-ZCOOR(2)*S)
      IF( XX.GT.ST(1,8) )           H1     = RCOOR(3)*SX
      GO TO 90
C..
   80 KTYPE  = 3
      XL2    = 0.0
      H1     = S2X*(XX-ZCOOR(4)*S)
      XL1    = S2X*XX
      IF( XX.LE.ST(1,12) )          KFNO   = 4
      IF( XX.LE.ST(1,16) )          KFNO   = 3
C..
   90 CONTINUE
      MFACE  = MFACE+1
      MESH(2,MFACE)  = KTYPE
      MESH(3,MFACE)  = KFNO
      SLICE(1,MFACE) = HO
      SLICE(2,MFACE) = XL1
      SLICE(3,MFACE) = XL2
      SLICE(4,MFACE) = H1
      SLICE(5,MFACE) = DX1
      SLICE(6,MFACE) = XX
  100 XX     = XX + DX1
C
C..
CRC   XX     = S2 + 0.5*DX2
      XX     = S2 + 0.1*DX2
      DO 200 J= 1, M2
      HO     = CS*(XX-RCOOR(5)*C)
C..
      IF( XX.GT.ST(1,5) )           GO TO 160
      IF( MLENG(4).GT.0 )           GO TO 155
      KTYPE  = 3
      KFNO   = 1
      IF( XX.GT.ST(1,14) )          KFNO   = 2
      XL2    = 0.0
      XL1    = R5MAX
      H1     = RCOOR(1)*SX
      GO TO 190
C
  155 KFNO   = 5
```

```
      IF( XX.LE.ST(1,18) )         KFNO    = 2
      IF( XX.LE.ST(1,17) )         KFNO    = 1
      KTYPE  = 2
      H1     = 0.0
      XL1    = R5MAXX
      IF( XX.LE.ST(1,9) )          XL2     = R5MAX
      IF( XX.GT.ST(1,9) )          XL2     = S2X*(ZCOOR(4)*S-XX)
      GO TO 190
C..   ( P5(S)<S<Q0(S) )
  160 IF( XX.GT.ST(1,15) )         GO TO 170
      IF( MLENG(4).GT.0 )          GO TO 165
      KTYPE  = 3
      H1     = RCOOR(1)*SX
      XL1    = R5MAX
      XL2    = 0.0
      KFNO   = 1
      IF( XX.GT.ST(1,14) )         KFNO    = 2
      GO TO 190
C
  165 KTYPE  = 6
      KFNO   = 5
      H1     = 0.0
      XL1    = R5MAX
      XL2    = S2X*(ZCOOR(2)*S-XX)
      GO TO 190
C..   ( Q0(S)<S<P11(S) )
  170 IF( XX.GT.ST(1,11) )         GO TO 180
      KTYPE  = 5
      KFNO   = 5
      XL2    = 0.0
      XL1    = R5MAX
      IF( MLENG(4).GT.0 )          GO TO 175
      IF( XX.GT.ST(1,7) )          GO TO 175
C
      KTYPE  = 3
      KFNO   = 2
      IF( XX.LE.ST(1,14) )         KFNO    = 1
      H1     = RCOOR(1)*SX
      GO TO 190
C
  175 H1     = S2X*(XX-ZCOOR(2)*S)
      IF( XX.GT.ST(1,8) )          H1      = RCOOR(3)*SX
      GO TO 190
C..
  180 KTYPE  = 3
      XL2    = 0.0
      H1     = S2X*(XX-ZCOOR(4)*S)
      XL1    = R5MAX
      IF( XX.LE.ST(1,12) )         KFNO    = 4
      IF( XX.LE.ST(1,16) )         KFNO    = 3
C..
  190 CONTINUE
      MFACE  = MFACE+1
      MESH(2,MFACE)   = KTYPE
      MESH(3,MFACE)   = KFNO
      SLICE(1,MFACE) = H0
      SLICE(2,MFACE) = XL1
      SLICE(3,MFACE) = XL2
```

```
      SLICE(4,MFACE) = H1
      SLICE(5,MFACE) = DX2
      SLICE(6,MFACE) = XX
  200 XX     = XX + DX2
C
C...   GET (MLOOP)
C
      MLOOP  = 0
      DO 600 J= 1, MFACE
      IF( MLOOP.GT.18000 )         GO TO 4000
C
      IF( KKTEST.NE.0 )            WRITE(6,6)     J, MESH(2,J),
     +                             MESH(3,J), (SLICE(I,J),I=1,6)
    6 FORMAT(/6H FACE=,I3,7H KTYPE=,I3,7H KFACT=,I3,7H SLICE=,6F13.4)
      KTYPE  = MESH(2,J)
      MLOOP1 = MLOOP+1
      GO TO ( 310, 320, 330, 340, 350, 360 )         , KTYPE
C
C. (1)-TYPE
  310 CONTINUE
      H      = SLICE(4,J)
      XM     = S*SQRT( B-H**2 )
      IF( SLICE(3,J).NE.R5MAX )    GO TO 315
      ML     = XM/DX0
      IF( ML.EQ.0 )                ML      = 1
      DPART  = ML
      DX     = XM/DPART
      XX     = 0.5*DX
      MESH(1,J) = ML
      DO 311 I= 1, ML
      XL     = SX*SQRT( R5-XX**2 )
      MLOOP  = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
  311 XX     = XX+DX
      GO TO 500
C
  315 XM0    = S*SQRT( B-SLICE(3,J)**2 )
      MLOOP  = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = SLICE(3,J)-H
      XLENG(3,MLOOP) = XM0
C
      ML     = (XM-XM0)/DX0
      IF( ML.EQ.0 )                ML      = 1
      DPART  = ML
      DX     = (XM-XM0)/DPART
      XX     = XM0 + 0.5*DX
      MESH(1,J) = ML+1
      DO 316 I= 1, ML
      XL     = SX*SQRT( R5-XX**2 )
      MLOOP  = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
  316 XX     = XX+DX
      GO TO 500
```

Appendix F (Continued)

```
C
C. (2)-TYPE
  320 CONTINUE
      ML      = RCOOR(5)/DXO
      IF( ML.EQ.O )             ML      = 1
      DPART   = ML
      DX      = RCOOR(5)/DPART
      XX      = 0.5*DX
      MESH(1,J) = ML
C
      IF( SLICE(2,J).NE.R5MAX )    GO TO 325
      IF( SLICE(3,J).NE.R5MAX )    GO TO 322
      DO 321 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = R5MAX-XL
      XLENG(2,MLOOP) = 2.0*XL
      XLENG(3,MLOOP) = DX
  321 XX      = XX+DX
      GO TO 500
C
  322 XM1     = S*SQRT( B-SLICE(3,J)**2 )
      DO 323 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = R5MAX-XL
      XLENG(2,MLOOP) = XL+SLICE(3,J)
      XLENG(3,MLOOP) = DX
      IF( XX.GT.XM1 )               XLENG(2,MLOOP) = 2.0*XL
  323 XX      = XX+DX
      GO TO 500
C..
  325 XMO     = S*SQRT( B-SLICE(2,J)**2 )
      IF( SLICE(3,J).NE.R5MAX )    GO TO 327
      DO 326 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL+SLICE(2,J)
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XMO )               GO TO 326
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = 2.0*XL
  326 XX      = XX+DX
      GO TO 500
C
  327 XM1     = S*SQRT( B-SLICE(3,J)**2 )
      DO 329 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      IF( XX.GT.XMO )               GO TO 328
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL+SLICE(2,J)
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XM1 )               XLENG(2,MLOOP) =SLICE(2,J)+SLICE(3,J)
      GO TO 329
  328 XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = 2.0*XL
```

Appendix F (Continued)

```
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XM1 )               XLENG(2,MLOOP) =SLICE(3,J)+XL
  329 XX      = XX+DX
      GO TO 500
C
C. (3)-TYPE
  330 CONTINUE
      H       = SLICE(4,J)
      XM      = S*SQRT( B-H**2 )
      IF( SLICE(2,J).NE.R5MAX )    GO TO 335
      ML      = XM/DXO
      IF( ML.EQ.O )                ML      = 1
      DPART   = ML
      DX      = XM/DPART
      XX      = 0.5*DX
      MESH(1,J) = ML
      DO 331 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = R5MAX-XL
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
  331 XX      = XX+DX
      GO TO 500
C
  335 XMO     = S*SQRT( B-SLICE(2,J)**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = SLICE(2,J)-H
      XLENG(3,MLOOP) = XMO
C
      ML      = (XM-XMO)/DXO
      IF( ML.EQ.O )                ML      = 1
      DPART   = ML
      DX      = (XM-XMO)/DPART
      XX      = XMO + 0.5*DX
      MESH(1,J) = ML+1
      DO 336 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
  336 XX      = XX+DX
      GO TO 500
C
C. (4)-TYPE
  340 CONTINUE
      XMM     = S*SQRT( G-SLICE(3,J)**2 )
      H       = SLICE(4,J)
      XM      = S*SQRT( B-H**2 )
      ML      = (XM-XMM)/DXO
      IF( ML.EQ.O )                ML      = 1
      DPART   = ML
      DX      = (XM-XMM)/DPART
      XX      = XMM + 0.5*DX
      MESH(1,J) = ML
      DO 341 I= 1, ML
```

```
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
  341 XX      = XX+DX
      GO TO 500
C
C. (5)-TYPE
  350 CONTINUE
      H       = SLICE(4,J)
      XMO     = S*SQRT( B-SLICE(2,J)**2 )
      XMM     = S*SQRT( G-H**2 )
      XXX     = XMO
      IF( XMM.LT.XMO )                XXX     = XMM
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = SLICE(2,J)-H
      XLENG(3,MLOOP) = XXX
C
      ML1     = 0
      IF( XMO.GE.XMM )                GO TO 355
      ML1     = (XMM-XMO)/DXO
      IF( ML1.EQ.0 )                  ML1     = 1
      DPART   = ML1
      DX      = (XMM-XMO)/DPART
      XX      = XMO + 0.5*DX
      DO 351 I= 1, ML1
      XL      = SX*SQRT( R5-XX**2 )
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = XL-SLICE(4,J)
      XLENG(3,MLOOP) = DX
  351 XX      = XX+DX
      XXX     = XMM
C
  355 ML2     = 0
      IF( XXX.GE.RCOOR(3) )           GO TO 357
      ML2     = (RCOOR(3)-XXX)/DXO
      IF( ML2.EQ.0 )                  ML2     = 1
      DPART   = ML2
      DX      = (RCOOR(3)-XXX)/DPART
      XX      = XXX + 0.5*DX
      DO 352 I= 1, ML2
      XL      = SX*SQRT( R5-XX**2 )
      XLM     = SX*SQRT( R3-XX**2 )
      IF( XL.GT.SLICE(2,J) )          XL      = SLICE(2,J)
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = XL-XLM
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XMO )                 XLENG(2,MLOOP) = SLICE(2,J)-XLM
  352 XX      = XX+DX
C
  357 ML3     = (RCOOR(5)-RCOOR(3))/DXO
      IF( ML3.EQ.0 )                  ML3     = 1
      DPART   = ML3
      DX      = (RCOOR(5)-RCOOR(3))/DPART
      XX      = RCOOR(3) + 0.5*DX
      DO 353 I= 1, ML3
      XL      = SX*SQRT( R5-XX**2 )
      IF( XL.GT.SLICE(2,J) )          XL      = SLICE(2,J)
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = 2.0*XL
      XLENG(3,MLOOP) = DX
  353 XX      = XX+DX
C
      MESH(1,J) = ML1+ML2+ML3+1
      GO TO 500
C
C. (6)-TYPE
  360 CONTINUE
      XMO     = S*SQRT( B-SLICE(2,J)**2 )
      XMM     = S*SQRT( G-SLICE(3,J)**2 )
      XXX     = XMO
      IF( XMM.LT.XMO )                XXX     = XMM
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = SLICE(2,J)+SLICE(3,J)
      XLENG(3,MLOOP) = XXX
C
      ML      = (RCOOR(5)-XXX)/DXO
      IF( ML.EQ.0 )                   ML      = 1
      DPART   = ML
      DX      = (RCOOR(5)-XXX)/DPART
      XX      = XXX + 0.5*DX
      DO 363 I= 1, ML
      XL      = SX*SQRT( R5-XX**2 )
      IF( XX.GT.XMO )                 GO TO 361
      MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL+SLICE(2,J)
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XMM )                 XLENG(2,MLOOP) =SLICE(2,J)+SLICE(3,J)
      GO TO 363
  361 MLOOP   = MLOOP+1
      XLENG(1,MLOOP) = SLICE(2,J)-XL
      XLENG(2,MLOOP) = 2.0*XL
      XLENG(3,MLOOP) = DX
      IF( XX.LE.XMM )                 XLENG(2,MLOOP) =SLICE(3,J)+XL
  363 XX      = XX+DX
      MESH(1,J) = ML+1
C
C
  500 CONTINUE
      MLOOP2 = MLOOP
      XMIN    = 1.0D10
      XMAX    = 0.0
      DO 550 I =MLOOP1,MLOOP2
      IF( XLENG(2,I).LE.0.0 )         WRITE(6,7)  I, (XLENG(K,I),K=1,3)
    7 FORMAT(6H XLNG=,I5,3F12.5)
      IF( XLENG(2,I).GE.XMIN )        GO TO 540
      XMIN    = XLENG(2,I)
      HMIN    = XLENG(1,I)
  540 IF( XLENG(2,I).LE.XMAX )        GO TO 550
```

Appendix F (Continued)

```
      XMAX    = XLENG(2,I)
      HMAX    = XLENG(1,I)
  550 CONTINUE
      IF( KKTEST.NE.0 )             WRITE(6,8)     MLOOP1, MLOOP2,
     +                              XMIN, HMIN, XMAX, HMAX
    8 FORMAT(6H LOOP=,2I5,6H LMIN=,F12.5,6H H1  =,F12.5,
     +       6H LMAX=,F12.5,6H H1  =,F12.5)
  600 CONTINUE
C
      WRITE(6,9)        MFACE, MLOOP
    9 FORMAT(14H *TOTAL MESH *,5X,18HNO. OF (S-T) FACE=,I6,10X,
     +   18HNO. OF (X-Y) FACE=,I6)
C
C
C...    GET CURRENT FORCE
C
 1000 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )           GO TO 2000
      TODISP = TODISP + DISPX
C
C...
      SUM     = 0.0
      TUM     = 0.0
      MLOOP   = 0
C
      DO 1500 J= 1, MFACE
      UO      = TODISP - SLICE(1,J)
      IF( UO.LE.0.0 )               GO TO 1450
C
      ML      = MESH(1,J)
      KFNO    = MESH(3,J)
      PSUM    = 0.0
      ESUM    = 0.0
C
      M6USE   = 0
      IF( J.LT.M6M1 )               GO TO 1100
      IF( J.GT.M6M2 )               GO TO 1100
      M6USE   = 1
 1100 CONTINUE
      DO 1400 I= 1, ML
      MLOOP   = MLOOP+1
      U       = UO-XLENG(1,MLOOP)
      IF( U.LE.0.0 )                GO TO 1400
C
      EPS     = U/XLENG(2,MLOOP)
CRC   IF( MXSTEP.GT.5 )             GO TO 1190
CRC   IF( EPS.LE.0.30 )             GO TO 1190
CRC   WRITE(6,19)  MXSTEP, MLOOP, (XLENG(K,MLOOP),K=1,3), U, TODISP, UO
CRC19 FORMAT(6H OVER=,2I5,8F12.5)
C1190 CONTINUE
      IF( EPS.LT.EPMAX )            GO TO 1200
      EPS     = EPMAX
      U       = EPMAX*XLENG(2,MLOOP)
C
 1200 CONTINUE
      CALL GETSIG ( EPS, 5, SIGM )
      UU      = DISPX
```

Appendix F (Continued)

```
      IF( U.LT.DISPX )              UU     = U
      PSUM    = PSUM + SIGM*XLENG(3,MLOOP)
      ESUM    = ESUM + SIGM*XLENG(3,MLOOP)*UU
C
      IF( M6USE.EQ.0 )              GO TO 1400
      IF( EPSU(MXSTEP,6).GT.EPS )   GO TO 1400
      MSUSE(6)        = MXSTEP
      EPSU(MXSTEP,6) = EPS
      SIGU(MXSTEP,6) = SIGM
 1400 CONTINUE
      SUM     = SUM + 2.0*PSUM*SLICE(5,J)*FACTK(KFNO)
      TUM     = TUM + 2.0*ESUM*SLICE(5,J)*FACTK(KFNO)
      GO TO 1500
C..  (UPDATE MLOOP)
 1450 ML      = MESH(1,J)
      MLOOP   = MLOOP+ML
 1500 CONTINUE
C
C...    STEP RESULTANT
C
      ENERGY = TUM
C
      DEPTH(MXSTEP) = TODISP
      FORCE(MXSTEP) = SUM
      ACCEL(MXSTEP) = SUM/WEIGHT
      ENERG(MXSTEP) = ENERGY
      IF( MXSTEP.NE.1 )             ENERG(MXSTEP) =ENERG(MXSTEP-1)+ENERGY
C
      IF( ENERG(MXSTEP).LT.ENERGO )GO TO 1000
      IF( MINI.NE.0 )               GO TO 1800
      MINI    = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      GO TO 1000
C
C...    INTERPOLATION OF LAST STEP
C
 1800 CONTINUE
      CALL FITSTP
C
C...    END OF CORNER
C
 2000 CONTINUE
      IF( MXSTEP.GT.500 )           MXSTEP  = 500
      RETURN
C
C***    ERROR
 4000 CONTINUE
      WRITE(MTOUT,4099)
 4099 FORMAT(// 29H *** (XLENG) CAPACITY OVER.  )
      CALL EXIT
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    MULTI ( DISP, DL, ID, LENGTH, S I G M , IERR )
C
C****   /PURPOSE/
C         GET STRESS FROM MULTI-MATERIAL TRUSS
C
C***    /OUTPUT/
C         SIGM       = TRUSS STRESS
C         IERR       = ERROR FLAG
C
C***    /INPUT/
C         DISP       = TOTAL DISPLACEMENT
C         DL         = ELEMENT LENGTH
C         ID         = MATERIAL IDENT
C         LENGTH     = NO. OF MATERIALS
C
C***    /PROGRAMMED BY JAERI                     1988.01.29
C***     MODIFIED   BY JAERI                     1988.09.19
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /SIGCNV/     CONV,     MXITER
      DIMENSION DL(1),    ID(1)
      DIMENSION EPS(3),   B(3),     SO(3),    EPS1(3),
     +          SK(9),    DU(3),    F(3),     SAME(2),
     +          KSAME(2)
      DATA DLT/ 1.0E-10 /
C
C...    FUNCTION
      ABS(Z) = DABS(Z)
C
C...    INITIAL STATUS
C
      EPS(1) = 0.0
      EPS(2) = 0.0
      EPS(3) = 0.0
      IERR   = 0
      ITER   = 0
C
C...    INITIAL STATUS
C
  100 CONTINUE
      ITER   = ITER+1
      DO 200 J= 1, LENGTH
      CALL GETMAT ( EPS(J), ID(J), B(J) , SO(J) )
  200 CONTINUE
C
C...    FIND SAME MATERIAL
C
      DO 300 J= 2, LENGTH
      I      = J-1
      SAME(I) =ABS(B(J)-B(I))
      KSAME(I) =0
      IF( SAME(I).LE.DLT )          KSAME(I) =1
  300 CONTINUE
C
C...    CALCULATION OF TRUSS DISP
C
      IF( LENGTH.NE.2 )             GO TO 400
```

Appendix F (Continued)

```
      IF( KSAME(1).NE.0 )           GO TO 3000
      SK( 1) = 1.0
      SK( 3) = 1.0
      SK( 2) =-B(1)/DL(1)
      SK( 4) = B(2)/DL(2)
      F( 1)  = DISP
      F( 2)  = SO(1)-SO(2)
      CALL EQSOLV ( SK, F, 2, D U , IERR )
      GO TO 500
C...  (3)-TRUSS
  400 CONTINUE
      IF( KSAME(1).NE.0 )           GO TO 3000
      IF( KSAME(2).NE.0 )           GO TO 3000
      SK( 1) = 1.0
      SK( 4) = 1.0
      SK( 7) = 1.0
      SK( 2) =-B(1)/DL(1)
      SK( 5) = B(2)/DL(2)
      SK( 8) = 0.0
      SK( 3) =-B(1)/DL(1)
      SK( 6) = 0.0
      SK( 9) = B(3)/DL(3)
      F( 1)  = DISP
      F( 2)  = SO(1)-SO(2)
      F( 3)  = SO(1)-SO(3)
      CALL EQSOLV ( SK, F, 3, D U , IERR )
C
C
C...    CONVERGENCE
C
  500 CONTINUE
      IF( IERR.NE.0 )               GO TO 3000
      NCV     = 0
      DO 600 J= 1, LENGTH
      EPS1(J) = DU(J)/DL(J)
      DCONV   = EPS(J)-EPS1(J)
      IF( EPS1(J).NE.0.0 )          DCONV = DCONV/EPS1(J)
      IF( ABS(DCONV).LT.CONV )      NCV     = NCV+1
  600 EPS(J)  = EPS1(J)
      IF( NCV.EQ.LENGTH )           GO TO 700
C...
      IF( ITER.LE.MXITER )          GO TO 100
      GO TO 1000
C
C...     END OF MULTI
C
  700 CONTINUE
CRC   WRITE(6,1)      (ID(I),DL(I),DU(I),EPS1(I),I=1,LENGTH)
CRC 1 FORMAT(6H MULT=,I4,6H  DL =,F12.4,6H  DU =,F12.4,6H  EPS=,F12.4)
CRC
C
      DO 800 J= 1, LENGTH
      CALL GETSIG ( EPS(J), ID(J), S I G M )
  800 CONTINUE
      RETURN
C
C...     ERROR  MULTI
 1000 IERR   = 1
```

Appendix F (Continued)

```
      SIGM   = 0.0
      RETURN
C
C
C...    SAME MATERIAL OPTION
C
 3000 CONTINUE
      BLENGX = 0.0
      DO 3100 J= 1, LENGTH
 3100 BLENGX = BLENGX+DL(J)
      EPS(1) = DISP/BLENGX
C
      DO 3200 J= 1, LENGTH
      CALL GETSIG ( EPS(1), ID(J), S I G M )
 3200 CONTINUE
      IERR   = 0
      RETURN
      END



      SUBROUTINE    GETMAT ( EPS, ID, B K , S I G O )
C
C****   /PURPOSE/
C         GET STIFFNESS AND INITIAL SIGM
C
C***    /OUTPUT/
C         BK        = STIFFNESS
C         SIGO      = INITIAL SIGM
C
C***    /INPUT/
C         EPS       = STRAIN
C         ID        = MATERIAL IDENT
C
C***    /PROGRAMMED BY JAERI                   1988.01.29
C***     MODIFIED   BY JAERI                   1988.09.19
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MATDAT/     TABLE(100,2,5),    MLENG(5)
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
      DATA DLT/ 1.0E-20 /
C
C...    FUNCTION
      ABS(Z) = DABS(Z)
C
C...    INITIAL VALUE
C
      LENGTH = MLENG(ID)
      IF( LENGTH.GT.1 )              GO TO 300
      BK     = 1.0
      SIGO   = 0.0
      RETURN
C
```

Appendix F (Continued)

```
C...    STRAIN RANGE
C
  300 CONTINUE
      IF( EPS.LE.0.0 )              GO TO 600
      DO 400 J2= 2, LENGTH
      IF( EPS.LT.TABLE(J2,1,ID) )   GO TO 500
  400 CONTINUE
      J2     = LENGTH
  500 CONTINUE
      J1     = J2-1
      DX     = TABLE(J2,1,ID)-TABLE(J1,1,ID)
      DY     = TABLE(J2,2,ID)-TABLE(J1,2,ID)
      IF( ABS(DX).LT.DLT )          GO TO 1000
C
      BK     = DY/DX
      SIGO   = TABLE(J1,2,ID)-BK*TABLE(J1,1,ID)
      RETURN
C
C..
  600 IF( TABLE(1,1,ID).EQ.0.0 )    GO TO 650
      BK     = TABLE(1,2,ID)/TABLE(1,1,ID)
      SIGO   = 0.0
      RETURN
C
  650 J2     = 2
      GO TO 500
C
C***     ERROR DATA
C
 1000 WRITE(6,1010)             ID, J1, J2
 1010 FORMAT(24H *** ERROR MATERIAL DATA,10X,4H ID=,I5,8H COLUMN=,2I5)
      CALL EXIT
      RETURN
C
C
C... ENTRY
C
      ENTRY         GETSIG ( EPS, ID, S I G O )
C
      LENGTH = MLENG(ID)
      IF( LENGTH.EQ.0 )             GO TO 1600
C
      DO 1400 J2= 2, LENGTH
      IF( EPS.LT.TABLE(J2,1,ID) )   GO TO 1500
 1400 CONTINUE
      J2     = LENGTH
 1500 CONTINUE
      J1     = J2-1
      DX     = TABLE(J2,1,ID)-TABLE(J1,1,ID)
      DY     = TABLE(J2,2,ID)-TABLE(J1,2,ID)
C
      SIGO   = TABLE(J1,2,ID)+DY/DX*(EPS-TABLE(J1,1,ID))
CRC   WRITE(6,5)        ID, EPS, SIGO
CRC 5 FORMAT(8H GETSIG=,I5,6H   EPS=,F12.4,6H   SIG=,F12.4)
C
      MSUSE(ID)        = MXSTEP
      EPSU(MXSTEP,ID) = DMAX1 ( EPSU(MXSTEP,ID), EPS )
      SIGU(MXSTEP,ID) = DMAX1 ( SIGU(MXSTEP,ID), SIGO )
```

Appendix F (Continued)

```
C
      RETURN
C
C***   ERROR
 1600 CONTINUE
      SIGO    = 0.0
      RETURN
      END


      SUBROUTINE       EQSOLV ( BK, F, NN, D , IERR )
C
C****   /PURPOSE/
C         EQUATION SOLVER   (BK)*(D) = (F)
C
C***    /OUTPUT/
C         D          = DISP VECTOR
C         IERR       = ERROR FLAG
C
C***    /INPUT/
C         BK         = MATRIX
C         F          = LOAD VECTOR
C
C***    /PROGRAMMED BY JAERI                    1988.01.29
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION BK(NN,NN), F(NN),    D(NN)
      DATA EPS/ 1.0D-20 /
C
C...
C
      DO 500 J= 1, NN
      PIVOT  = BK(J,J)
      IF( DABS(PIVOT).LE.EPS )     GO TO 1000
C
      DO 300 I= 1, NN
      IF( I.EQ.J )                GO TO 300
      IF( DABS(BK(I,J)).LT.EPS )  GO TO 300
      SHIFT  = BK(I,J)/PIVOT
C
      DO 100 K= 1, NN
  100 BK(I,K) = BK(I,K) - SHIFT*BK(J,K)
C
      F(I)    = F(I) - SHIFT*F(J)
  300 CONTINUE
  500 CONTINUE
C
C...
      DO 600 J= 1, NN
  600 D(J)    = F(J)/BK(J,J)
      IERR    = 0
      RETURN
C
C***   ERROR MATRIX
C
 1000 CONTINUE
```

Appendix F (Continued)

```
      IERR    = 1
      RETURN
      END


      SUBROUTINE    FITSTP
C
C****   /PURPOSE/
C         RESET CALCULATED ENERGRY OF LAST STEP
C
C***    PROGRAMMED BY JAERI                    1988.09.19
C
C
      IMPLICIT    REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +          RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +          DISP,     FACTK(5), ENERGO,   GRAVIT,   MTINN,
     +          MTOUT,    MODEL,    IFLAG,    KSOLV,    KOUT,
     +          NPART,    KPART,    NTITLE(18)
      COMMON /RESLT/
     +          DEPTH(500),         FORCE(500),         ACCEL(500),
     +          ENERG(500),         ENERGY,             TODISP,
     +          EPSU(500,6),        SIGU(500,6),        MXSTEP,
     +          MSUSE(6)
      DATA EPS/ 0.05 /
C
C...    FUNCTION
      ABS(Z) = DABS(Z)
C
C...
C
 -    GO TO 200
C
  100 MXSTEP = MXSTEP-1
C
  200 CONTINUE
      NXSTEP = MXSTEP-1
      DL1    = ENERG(MXSTEP)-ENERG(NXSTEP)
      DL2    = ENERGO       -ENERG(NXSTEP)
      IF( ABS(DL2).LE.EPS )        GO TO 100
      DLT    = DL2/DL1
C
      DEPTH(MXSTEP) = DLT*(DEPTH(MXSTEP)-DEPTH(NXSTEP)) + DEPTH(NXSTEP)
      FORCE(MXSTEP) = DLT*(FORCE(MXSTEP)-FORCE(NXSTEP)) + FORCE(NXSTEP)
      ACCEL(MXSTEP) = DLT*(ACCEL(MXSTEP)-ACCEL(NXSTEP)) + ACCEL(NXSTEP)
      ENERG(MXSTEP) = ENERGO
C
      RETURN
      END


      SUBROUTINE    MPLOT ( LABEL )
C
C****   /PURPOSE/
C         GEOMETRY PLOT
C
C***    /INPUT/
```

Appendix F (Continued)

```
C          LABEL      = ANALYSIS TYPE LABEL
C
C***      PROGRAMMED BY JAERI                    1988.02.01
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +           RCOOR(5), ZCOOR(9), WEIGHT,   HEIGHT,   ANGLE,
     +           DISP,     FACTK(5), ENERGO,   GRAVIT,   EPMAX,
     +           MTINN,    MTOUT,    MODEL,    IFLAG,    KSOLV,
     +           KOUT,     NPART,    KPART,    NTITLE(18)
      COMMON /MODEL1/
     +           S,        C,        S2,       C2
C
      DIMENSION LABEL(3)
      DIMENSION X(40),     Y(40),    XV(6),    YV(6),    LDIR(2),
     +           NP(2,17)
      DATA NP/  1, 0,    2, 0,    3, 0,    4, 0,    5, 0,    5, 1,
     +           5, 2,    5, 3,    5, 4,    4, 4,    3, 4,    3, 3,
     +           3, 2,    2, 2,    1, 2,    1, 1,    1, 0/
      DATA NODE/ 17 /
      DATA LDIR/   4HR    , 4HZ     /
      DATA XL,YL/  3.8637, 1.0353 /
      DATA XO,YO/     40.0,   20.0 /
      DATA HH/  2.5 /
C
C...     FUNCTION
      SQRT(Z) = DSQRT(Z)
C
C...     CLEAR (X,Y)
C
      DO 100 J= 1, 35
      X(J)    = 0.0
  100 Y(J)    = 0.0
C
C...     GET GEOMETRY
C
      DO 200 J= 1, NODE
      IX      = NP(1,J)
      IY      = NP(2,J)
      X(J)    = RCOOR(IX)
      IF( IY.EQ.0 )            GO TO 200
      Y(J)    = ZCOOR(IY)
  200 CONTINUE
      NN      = NODE
      ANGXX   = 0.0
C
      IF( MODEL.NE.2 )         GO TO 400
      IF( ZCOOR(5).LT.ZCOOR(4) )   GO TO 310
      NN      = 2*NODE
      DO 300 J= 1, NODE
      IX      = NP(1,J)
      IY      = NP(2,J)
      X(J+NODE) = RCOOR(IX)
      Y(J+NODE) = ZCOOR(9-IY)
  300 CONTINUE
      GO TO 330
  310 ZWIDE   = 3.0*ZCOOR(4)
      DO 320 J= 1, NODE
```

Appendix F (Continued)

```
      X(J+NODE) = X(J)
      Y(J+NODE) = ZWIDE-Y(J)
  320 CONTINUE
C
  330 MM      = NN+4
      X(NN+1)= 0.0
      X(NN+2)= 0.5*RCOOR(1)
      X(NN+3)= 0.0
      X(NN+4)= 0.0
      Y(NN+1)= 0.0
      Y(NN+2)= 0.0
      Y(NN+3)= 0.0
      Y(NN+4)= 0.5*RCOOR(1)
      DO 350 J= 1, MM
      R       = X(J)
      Z       = Y(J)
      X(J)    =  Z
      Y(J)    = -R
  350 CONTINUE
      GO TO 500
C
C...     CORNER MODEL
C
  400 MM      = NN+4
      X(NN+1)= 0.0
      X(NN+2)= 0.5*RCOOR(1)
      X(NN+3)= 0.0
      X(NN+4)= 0.0
      Y(NN+1)= 0.0
      Y(NN+2)= 0.0
      Y(NN+3)= 0.0
      Y(NN+4)= 0.5*RCOOR(1)
      IF( MODEL.NE.3 )         GO TO 500
      ANGXX   = -ANGLE
      DO 450 J= 1, MM
      R       = X(J)
      Z       = Y(J)
      X(J)    =  R*C + Z*S
  450 Y(J)    = -R*S + Z*C
C
C...     GEOMETRY SCALING
C
  500 CONTINUE
      CALL GOSCAL ( MM, X, Y, XMN, YMN, XMX, YMX, FACTOR )
C
C...     PLOT COORDINATES
C
      DO 600 J= 1, MM
      X(J)    = FACTOR*(X(J)-XMN) + XO
      Y(J)    = FACTOR*(Y(J)-YMN) + YO
  600 CONTINUE
C
C...     GEOMETRY PLOT
C
      CALL PLPEN ( 1 )
      CALL PLINES ( X, Y, NODE )
      RR      = 1.0
      DO 700 J= 1, NODE
```

Appendix F (Continued)

```
      CALL PLMARK ( X(J), Y(J), RR )
  700 CONTINUE
C
      IF( NN.EQ.NODE )         GO TO 1000
      MM     = NODE+1
      CALL PLINES ( X(MM), Y(MM), NODE )
      DO 800 J= MM, NN
      CALL PLMARK ( X(J), Y(J), RR )
  800 CONTINUE
C
C...    AXES PLOT
C
 1000 CONTINUE
      DO 1200 J= 1, 2
      XV(1)   = X(NN+1)
      YV(1)   = Y(NN+1)
      XV(2)   = X(NN+2)
      YV(2)   = Y(NN+2)
      XV(5)   = X(NN+2)
      YV(5)   = Y(NN+2)
      NN      = NN+2
C
      X(1)    = XV(2)-XV(1)
      Y(1)    = YV(2)-YV(1)
      DLL     = SQRT( X(1)**2+Y(1)**2 )
      F1      = (DLL-XL)/DLL
      F2      = (DLL+XL)/DLL
      X(2)    = XV(1) + F1*X(1)
      Y(2)    = YV(1) + F1*Y(1)
      XV(6)   = XV(1) + F2*X(1)
      YV(6)   = YV(1) + F2*Y(1)
      X(1)    = X(1)/DLL
      Y(1)    = Y(1)/DLL
C
      XV(3)   = X(2) - Y(1)*YL
      XV(4)   = X(2) + Y(1)*YL
      YV(3)   = Y(2) + X(1)*YL
      YV(4)   = Y(2) - X(1)*YL
      CALL PLINES ( XV, YV, 5 )
      CALL PSYM   ( XV(6), YV(6), HH, LDIR(J), ANGXX, 1 )
 1200 CONTINUE
C
C...    HEADER PLOT
C
      XX      = 10.0
      YY      = 0.25*YO
      CALL PSYM   ( XX, YY, HH, NTITLE, 0.0D0, 70 )
C
      XX      = 150.0
      YY      = 0.50*YO
      CALL PSYM   ( XX, YY, HH, LABEL, 0.0D0, 12 )
C
C...
      CALL PLTEOR
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    OPLOT ( LABEL, NTITLE, XVAL, YVAL, LENGTH, KTYPE,
     +                      IFLAG, ID )
C
C****    /PURPOSE/
C          XYPLOT PROGRAM
C
C***     /INPUT/
C          LABEL       = ANALYSIS TYPE LABEL
C          NTITLE      = MODEL TITLE MESSAGE
C          XVAL        = X VARIABLES
C          YVAL        = Y VARIABLES
C          LENGTH      = NO. OF VARIABLES
C          KTYPE       = 1 ENERGY PLOT
C                      = 2 ACCELERATION PLOT
C                      = 3 STRAIN-STRESS PLOT
C                      = 4 STRAIN-STRESS PLOT (WITH CALCULATED)
C                      = 5 FORCE PLOT
C          IFLAG       = 0, MM        = 1, CM
C          ID          = MATERIAL IDENT
C
C***      PROGRAMMED BY JAERI                    1988.02.01
C***      MODIFIED   BY JAERI                    1988.09.19
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      DIMENSION LABEL(3), NTITLE(18)
      DIMENSION XVAL(1),  YVAL(1)
      DIMENSION X(5),     Y(5),     NP(2,5),  KSCL(2),  KHEAD(3,6),
     +          MLAB(6),  KKMCM(2), MH2H(2),  MHEAD(2,6), MCALC(2),
     +          KKMCM2(2)
      DATA NP/        1, 2,    1, 3,    4, 5,    4, 5,    1,6 /
      DATA KHEAD/  4HDEFO, 4HRMAT, 4HION ,
     +             4HENER, 4HGY  , 4H    ,
     +             4HACCE, 4HLERA, 4HTION,
     +             4HSTRA, 4HIN  , 4H    ,
     +             4HSTRE, 4HSS  , 4H    ,
     +             4HFORC, 4HE   , 4H    /
      DATA MHEAD/  4H    , 4H    ,             4H KG-, 4HM    ,
     +             4H G  , 4H    ,             4H    , 4HM    ,
     +             4H KG/, 4H    ,             4H KG , 4H     /
      DATA KSCL/   4H(10 , 4H )  /
      DATA MLAB/   4HSIGA, 4HSIGB, 4HSIGC, 4HSIGD, 4HSIG , 4HSIGX/
      DATA KKMCM/  4HMM  , 4HCM  /
      DATA KKMCM2/ 4HMM2 , 4HCM2 /
      DATA MH2H/   4HMM/M, 4HCM/C/
      DATA MCALC/  4H(CAL, 4HC.) /
      DATA NPARTX,NPARTY/  10, 8 /
      DATA XL,YL/  200.0,  160.0 /
      DATA XO,YO,DH/ 40.0, 30.0, 20.0 /
      DATA HH,H2,H3/  2.5,  2.0,  1.5 /
C
C...    DATA RANGE
C
      CALL GOSCAL ( LENGTH, XVAL, YVAL, XMN, YMN, XMX, YMX, FACT )
C
      CALL XYSCAL ( XMN, XMX, NPARTX, DX, NEXPX )
      CALL XYSCAL ( YMN, YMX, NPARTY, DY, NEXPY )
C
```

Appendix F (Continued)

```
C...    GRID MESH
C
      CALL PLPEN( 1 )
      X(1)    = XO
      X(2)    = XO+XL
      X(3)    = XO+XL
      X(4)    = XO
      X(5)    = XO
      Y(1)    = YO
      Y(2)    = YO
      Y(3)    = YO+YL
      Y(4)    = YO+YL
      Y(5)    = YO
      CALL PLINES ( X, Y, 5 )
C. (X) SCALE
      Y(1)    = YO-2.0
      Y(2)    = YO+2.0
      Y(3)    = YO-5.0
      Y(4)    = YO+YL-2.0
      Y(5)    = YO+YL
      X(1)    = XO
      X(2)    = XO
      DNO     = 0.0
      LPARTX = NPARTX+1
      DO 300 J= 1, LPARTX
      CALL PLINES ( X, Y, 2 )
      CALL PLINES ( X, Y(4), 2)
      X(3)    = X(1)-H2
      CALL PNUMB  ( X(3), Y(3), H2, DNO, 0.0DO, 1 )
      X(1)    = X(1)+DH
      X(2)    = X(2)+DH
      DNO     = DNO+DX
  300 CONTINUE
      IF( NEXPX.EQ.0 )                  GO TO 350
      X(3)    = XO+XL+4.0*H2
      CALL PSYM   ( X(3), Y(3), H2, KSCL, 0.0DO, 6 )
      X(3)    = XO+XL+7.0*H2
      Y(3)    = Y(3)+H3
      CALL PNUMBR ( X(3), Y(3), H3, NEXPX, 0.0DO )
C. (Y) SCALE
  350 X(1)    = XO-2.0
      X(2)    = XO+2.0
      X(3)    = XO-5.0*H2
      X(4)    = XO+XL-2.0
      X(5)    = XO+XL
      Y(1)    = YO
      Y(2)    = YO
      DNO     = 0.0
      LPARTY = NPARTY+1
      DO 400 J= 1, LPARTY
      CALL PLINES ( X, Y, 2 )
      CALL PLINES ( X(4), Y, 2 )
      Y(3)    = Y(1)
      CALL PNUMB  ( X(3), Y(3), H2, DNO, 0.0DO, 1 )
      Y(1)    = Y(1)+DH
      Y(2)    = Y(2)+DH
      DNO     = DNO+DY
  400 CONTINUE
```

Appendix F (Continued)

```
      IF( NEXPY.EQ.0 )                 GO TO 500
      X(3)    = XO
      Y(3)    = YO+YL+H2
      CALL PSYM   ( X(3), Y(3), H2, KSCL, 0.0DO, 6 )
      X(3)    = XO+3.0*H2
      Y(3)    = Y(3)+H3
      CALL PNUMBR ( X(3), Y(3), H3, NEXPY, 0.0DO )
C
C...    HEADER PLOT
C
  500 CONTINUE
      XX      = XO+20.0*HH
      YY      = YO+YL+3.0*HH
      CALL PSYM   ( XX, YY, HH, NTITLE, 0.0DO, 70 )
C. (X-TITLE)
      IPX     = NP(1,KTYPE)
      XX      = XO+20.0*HH
      YY      = 0.6*YO
      CALL PSYM   ( XX, YY, HH, KHEAD(1,IPX), 0.0DO, 12 )
C
      XX      = XX+15.0*HH
      IF( IPX.EQ.1 )           MHEAD(1,IPX) = KKMCM(IFLAG+1)
      IF( IPX.EQ.4 )           MHEAD(1,IPX) = MH2H(IFLAG+1)
      CALL PSYM   ( XX, YY, HH, MHEAD(1,IPX), 0.0DO, 8 )
C. (Y-TITLE)
      IPY     = NP(2,KTYPE)
      XX      = 0.4*XO
      YY      = YO+20.0*HH
      CALL PSYM   ( XX, YY, HH, KHEAD(1,IPY), 90.0DO, 12 )
C
      YY      = YY+15.0*HH
      IF( KTYPE .EQ. 1) GO TO 510
      IF( IPY.EQ.2 )           MHEAD(2,IPY) = KKMCM(IFLAG+1)
      IF( IPY.EQ.5 )           MHEAD(2,IPY) = KKMCM2(IFLAG+1)
  510 CONTINUE
      CALL PSYM   ( XX, YY, HH, MHEAD(1,IPY), 90.0DO, 8 )
C. (ANALYSIS LABEL)
      XX      = XO+160.0
      YY      = 0.60*YO
      CALL PSYM   ( XX, YY, HH, LABEL, 0.0DO, 12 )
      IF( KTYPE.LE.2 )                 GO TO 600
      IF( KTYPE.GE.5 )                 GO TO 600
C. (MATERIAL ID)
      XX      = XX+13.0*HH
      CALL PSYM   ( XX, YY, HH, MLAB(ID), 0.0DO, 4 )
      IF( KTYPE.EQ.3 )                 GO TO 600
      XX      = XX+5.0*HH
      CALL PSYM   ( XX, YY, HH, MCALC, 0.0DO, 7 )
C
C
C...     XYPLOT
C
  600 CONTINUE
      CALL PLPEN( 2 )
      PX      = DX* 10.0**NEXPX
      PY      = DY* 10.0**NEXPY
      XFACT   = DH/PX
      YFACT   = DH/PY
```

```
C
      X(1)    = XO
      Y(1)    = YO
      R       = 1.0
      DO 700 J= 1, LENGTH
      X(2)    = XFACT*XVAL(J) + XO
      Y(2)    = YFACT*YVAL(J) + YO
      CALL PLINES ( X, Y, 2 )
      CALL PLMARK ( X(2), Y(2), R )
      X(1)    = X(2)
      Y(1)    = Y(2)
  700 CONTINUE
C
      CALL PLTEOR
      RETURN
      END


      SUBROUTINE    UPLOT ( LABEL, DEPTH, ACCEL )
C
C****   /PURPOSE/
C         GEOMETRY PLOT AND MAXIMUM DISPLACEMENT VALUE PLOT
C
C***    /INPUT/
C         LABEL      = ANALYSIS TYPE LABEL
C         DEPTH      = MAXIMUM DISPLACEMENT VALUE
C         ACCEL      = MAXIMUM ACCELERATION VALUE
C
C***     PROGRAMMED BY JAERI                    1988.09.20
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /MODELS/
     +           RCOOR(5), ZCOOR(9), WEIGHT,    HEIGHT,   ANGLE,
     +           DISP,     FACTK(5), ENERGO,    GRAVIT,   EPMAX,
     +           MTINN,    MTOUT,    MODEL,     IFLAG,    KSOLV,
     +           KOUT,     NPART,    KPART,     NTITLE(18)
      COMMON /MODEL1/
     +           S,        C,        S2,        C2
C
      DIMENSION LABEL(3)
      DIMENSION X(40),     Y(40),    XV(6),     YV(6),    LDIR(2),
     +           NP(2,17), KKMCM(2),
     +           KDEPTH(2),          KACCEL(2)
      DATA NP/  1, 0,   2, 0,   3, 0,   4, 0,   5, 0,   5, 1,
     +           5, 2,   5, 3,   5, 4,   4, 4,   3, 4,   3, 3,
     +           3, 2,   2, 2,   1, 2,   1, 1,   1, 0/
      DATA NODE/ 17 /
      DATA LDIR/   4HR    , 4HZ    /
      DATA KDEPTH/ 4HDEPT, 4HH=    /
      DATA KACCEL/ 4HACCE, 4HL=    /
      DATA KKMCM/  4H(MM), 4H(CM) /
      DATA KKACCG/ 4H(G) /
      DATA XL,YL/  3.8637, 1.0353 /
      DATA XO,YO/       40.0,  20.0 /
      DATA HH/  2.5 /
      DATA H2/  2.0 /
```

```
C
C...      FUNCTION
      SQRT(Z) = DSQRT(Z)
C
C...      CLEAR (X,Y)
C
      DO 100 J= 1, 35
      X(J)    = 0.0
  100 Y(J)    = 0.0
C
C...      GET GEOMETRY
C
      DO 200 J= 1, NODE
      IX      = NP(1,J)
      IY      = NP(2,J)
      X(J)    = RCOOR(IX)
      IF( IY.EQ.0 )             GO TO 200
      Y(J)    = ZCOOR(IY)
  200 CONTINUE
      NN      = NODE
      ANGXX   = 0.0
C
      IF( MODEL.NE.2 )          GO TO 400
      IF( ZCOOR(5).LT.ZCOOR(4) )    GO TO 310
      NN      = 2*NODE
      DO 300 J= 1, NODE
      IX      = NP(1,J)
      IY      = NP(2,J)
      X(J+NODE) = RCOOR(IX)
      Y(J+NODE) = ZCOOR(9-IY)
  300 CONTINUE
      GO TO 330
  310 ZWIDE   = 3.0*ZCOOR(4)
      DO 320 J= 1, NODE
      X(J+NODE) = X(J)
      Y(J+NODE) = ZWIDE-Y(J)
  320 CONTINUE
C
  330 MM      = NN+4
      X(NN+1)= 0.0
      X(NN+2)= 0.5*RCOOR(1)
      X(NN+3)= 0.0
      X(NN+4)= 0.0
      Y(NN+1)= 0.0
      Y(NN+2)= 0.0
      Y(NN+3)= 0.0
      Y(NN+4)= 0.5*RCOOR(1)
      DO 350 J= 1, MM
      R       = X(J)
      Z       = Y(J)
      X(J)    =  Z
      Y(J)    = -R
  350 CONTINUE
      GO TO 500
C
C...       CORNER MODEL
C
  400 MM      = NN+4
```

Appendix F (Continued)

```
      X(NN+1)= 0.0
      X(NN+2)= 0.5*RCOOR(1)
      X(NN+3)= 0.0
      X(NN+4)= 0.0
      Y(NN+1)= 0.0
      Y(NN+2)= 0.0
      Y(NN+3)= 0.0
      Y(NN+4)= 0.5*RCOOR(1)
      IF( MODEL.NE.3 )           GO TO 500
      ANGXX  = -ANGLE
      DO 450 J= 1, MM
      R       = X(J)
      Z       = Y(J)
      X(J)    =  R*C + Z*S
  450 Y(J)    = -R*S + Z*C
C
C...    GEOMETRY SCALING
C
  500 CONTINUE
      CALL GOSCAL ( MM, X, Y, XMN, YMN, XMX, YMX, FACTOR )
C
C...    PLOT COORDINATES
C
      DO 600 J= 1, MM
      X(J)    = FACTOR*(X(J)-XMN) + XO
      Y(J)    = FACTOR*(Y(J)-YMN) + YO
  600 CONTINUE
C
C...    GEOMETRY PLOT
C
      CALL PLPEN (1)
      CALL PLINES ( X, Y, NODE )
      RR      = 1.0
      DO 700 J= 1, NODE
      CALL PLMARK ( X(J), Y(J), RR )
  700 CONTINUE
C
      IF( NN.EQ.NODE )           GO TO 1000
      MM      = NODE+1
      CALL PLINES ( X(MM), Y(MM), NODE )
      DO 800 J= MM, NN
      CALL PLMARK ( X(J), Y(J), RR )
  800 CONTINUE
C
C...    AXES PLOT
C
 1000 CONTINUE
      DO 1200 J= 1, 2
      XV(1)   = X(NN+1)
      YV(1)   = Y(NN+1)
      XV(2)   = X(NN+2)
      YV(2)   = Y(NN+2)
      XV(5)   = X(NN+2)
      YV(5)   = Y(NN+2)
      NN      = NN+2
C
      X(1)    = XV(2)-XV(1)
      Y(1)    = YV(2)-YV(1)
```

Appendix F (Continued)

```
      DLL     = SQRT( X(1)**2+Y(1)**2 )
      IF( DLL .LE. 1.0D-2 ) DLL = 1.0D-2
      F1      = (DLL-XL)/DLL
      F2      = (DLL+XL)/DLL
      X(2)    = XV(1) + F1*X(1)
      Y(2)    = YV(1) + F1*Y(1)
      XV(6)   = XV(1) + F2*X(1)
      YV(6)   = YV(1) + F2*Y(1)
      X(1)    = X(1)/DLL
      Y(1)    = Y(1)/DLL
C
      XV(3)   = X(2) - Y(1)*YL
      XV(4)   = X(2) + Y(1)*YL
      YV(3)   = Y(2) + X(1)*YL
      YV(4)   = Y(2) - X(1)*YL
      CALL PLINES ( XV, YV, 5 )
      CALL PSYM   ( XV(6), YV(6), HH, LDIR(J), ANGXX, 1 )
 1200 CONTINUE
C
C...    DISPLACEMENT VALUE PLOT
C
      DLL     = 0.1*DEPTH*FACTOR
      X(1)    = 230.0
      X(2)    = X(1)+6.0
      X(3)    = X(1)
      X(4)    = X(2)
      X(6)    = X(1)+3.0
      X(7)    = X(6)
      X(5)    = X(6)-0.25*YL*DLL
      X(8)    = X(6)+0.25*YL*DLL
      Y(1)    = YO
      Y(2)    = Y(1)
      Y(3)    = Y(1)+DEPTH*FACTOR
      Y(4)    = Y(3)
      Y(6)    = Y(1)
      Y(7)    = Y(3)
      Y(5)    = Y(6)+0.25*XL*DLL
      Y(8)    = Y(7)-0.25*XL*DLL
C
      CALL PLPEN (2)
      CALL PLINES ( X(1), Y(1), 2 )
      CALL PLINES ( X(3), Y(3), 2 )
      CALL PLINES ( X(5), Y(5), 4 )
      XX      = X(1)-10.0
      YY      = YO-1.5*H2
      CALL PSYM   ( XX, YY, H2, KDEPTH, 0.0D0, 7 )
      XX      = XX+7.0*H2
      CALL PNUMB  ( XX, YY, H2, DEPTH,  0.0D0, 2 )
      FXU     = 6.0
      IF( DEPTH.GE.100.0  )           FXU     = 7.0
      IF( DEPTH.GE.1000.0 )           FXU     = 8.0
      XX      = XX+FXU*H2
      CALL PSYM   ( XX, YY, H2, KKMCM(IFLAG+1), 0.0D0, 4 )
C
      XX      = 120.0
      YY      = YO+H2+ FACTOR*(YMX-YMN)
      CALL PSYM   ( XX, YY, H2, KACCEL, 0.0D0, 7 )
      XX      = XX+7.0*H2
```

Appendix F (Continued)

```
      CALL PNUMB  ( XX, YY, H2, ACCEL,  0.0D0, 2 )
      XX      = XX+7.0*H2
      CALL PSYM   ( XX, YY, H2, KKACCG, 0.0D0, 3 )
C
C...    HEADER PLOT
C
      XX      = 10.0
      YY      = 0.25*Y0
      CALL PSYM   ( XX, YY, HH, NTITLE, 0.0D0, 70 )
C
      XX      = 150.0
      YY      = 0.50*Y0
      CALL PSYM   ( XX, YY, HH, LABEL, 0.0D0, 12 )
C
C...
      CALL PLTEOR
      RETURN
      END


      SUBROUTINE    GOSCAL ( ND, X, Y, XMN, YMN, XMX, YMX, SCALEG )
C
C****    /PURPOSE/
C          GET SCALING FACTOR OF THE GEOMETRY PLOT
C
C***     /OUTPUT/
C          XMN         = X MINIMUM
C          YMN         = Y MINIMUM
C          XMX         = X MAXIMUM
C          YMX         = Y MAXIMUM
C          SCALEG      = SCALING FACTOR
C
C***     /INPUT/
C          ND          = NO. OF NODES
C          X           = X COORNINATES
C          Y           = Y COORNINATES
C
C***      PROGRAMMED BY JAERI                  1988.02.01
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      DIMENSION X(1),      Y(1)
      DATA PAPER/ 180.0 /
C
C...    FUNCTION
      AMIN1 (Z1,Z2) = DMIN1 (Z1,Z2)
      AMAX1 (Z1,Z2) = DMAX1 (Z1,Z2)
C
C...    GEOMETRY RANGE
C
      XMN     =  1.0D20
      YMN     =  1.0D20
      XMX     = -1.0D20
      YMX     = -1.0D20
C
      DO 100 I= 1, ND
      XI      = X(I)
```

Appendix F (Continued)

```
      XMN     = AMIN1 (XMN,XI)
      XMX     = AMAX1 (XMX,XI)
      YI      = Y(I)
      YMN     = AMIN1 (YMN,YI)
      YMX     = AMAX1 (YMX,YI)
  100 CONTINUE
C
      XS      = XMX-XMN
      YS      = YMX-YMN
      AS      = AMAX1 (XS,YS)
      SCALEG  = PAPER/AS
      RETURN
      END


      SUBROUTINE    XYSCAL ( VMIN, VMAX, NPART, DSIZE , NEXP )
C
C****    /PURPOSE/
C          XYPLOT SCALING
C
C***     /OUTPUT/
C          DSIZE       = ADDITIONAL VALUE
C          NEXP        = EXPONENT NUMBER
C
C***     /INPUT/
C          VMIN        = MINIMUM VALUE
C          VMAX        = MAXIMUM VALUE
C          NPART       = NO. OF DIVISION
C
C***      PROGRAMMED BY JAERI                  1988.02.01
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      DIMENSION DVAL(8)
      DATA DVAL/ 0.5, 1.0, 1.5, 2.0, 2.5, 5.0, 10.0, 15.0 /
C
C...    FUNCTION
      ALOG10(Z) = DLOG10(Z)
C
C...
C
      DPART   = NPART
      XMIN    = 0.0
      DS      = (VMAX-XMIN)/DPART
      NEXP    = ALOG10 (DS)
      IF( NEXP.LT.0 )          NEXP   = NEXP-1
      GVAL    = 10.0**NEXP
C
      HWIDE   = DS/GVAL
      DO 100 J= 1, 7
      IF( HWIDE.LE.DVAL(J) )  GO TO 200
  100 CONTINUE
      J       = 7
  200 DSIZE   = DVAL(J)
C
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    PLTBGN
      IMPLICIT     REAL*8 (A-H,O-Z)
      CALL  PLOTS (0.0D0,0.0D0,10)
      RETURN
      END


      SUBROUTINE    PLPEN (NO)
      CALL  NEWPEN (NO)
      RETURN
      END


      SUBROUTINE    PLINES ( X,Y,NN )
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION   X(1), Y(1)
      CALL  PLOT (X(1),Y(1),3)
      IF( NN.EQ.1 )           RETURN
      DO 10 I=2,NN
      CALL  PLOT (X(I),Y(I),2)
   10 CONTINUE
      RETURN
      END


      SUBROUTINE    PLMARK ( XO,YO,R )
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION   CR(2,9), X(9), Y(9)
      DATA CR/ 0.92388,  0.38268,       0.38268,  0.92388,
     +        -0.38268,  0.92388,      -0.92388,  0.38268,
     +        -0.92388, -0.38268,      -0.38268, -0.92388,
     +         0.38268, -0.92388,       0.92388, -0.38268,
     +         0.92388,  0.38268/
C
      DO 10 J=1,9
      X(J) =R*CR(1,J) + XO
   10 Y(J) =R*CR(2,J) + YO
C
      CALL  PLINES ( X, Y, 9 )
      RETURN
      END


      SUBROUTINE    PSYM   ( X,Y,H,TIT,ANG,NL )
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION   TIT(1)
      CALL  SYMBOL (X,Y,H,TIT,ANG,NL)
      RETURN
      END


      SUBROUTINE    PNUMB   ( X,Y,H,AN,ANG,N )
      IMPLICIT     REAL*8 (A-H,O-Z)
      CALL  NUMBER (X,Y,H,AN,ANG,N)
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE    PNUMBR ( X,Y,H,N,ANG )
      IMPLICIT     REAL*8 (A-H,O-Z)
      AN =N
      CALL  NUMBER (X,Y,H,AN,ANG,-1)
      RETURN
      END


      SUBROUTINE    PLTEOR
      IMPLICIT     REAL*8 (A-H,O-Z)
C     CALL  PLOT (300.0D0,0.0D0,-3)
      CALL  PLOT ( 0.0D0, 0.0D0, 666 )
      CALL  PLOT ( 0.0D0, 0.0D0, 777 )
      CALL  PLOT ( 0.0D0, 0.0D0, 888 )
      RETURN
      END


      SUBROUTINE    PLTEND
      IMPLICIT     REAL*8 (A-H,O-Z)
      CALL  PLOT (0.0D0,0.0D0,999)
      RETURN
      END
```