

**JAERI-Data/Code**  
**97-001**



**CRUSH2:**  
**A SIMPLIFIED COMPUTER PROGRAM FOR IMPACT ANALYSIS**  
**OF RADIOACTIVE MATERIAL TRANSPORT CASKS**

February 1997

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, 1997

編集兼発行 日本原子力研究所  
印 刷 株原子力資料サービス

CRUSH2 : 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 January 17, 1997)

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 CRUSH2 has been developed. The CRUSH2 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 CRUSH2 is a revised version of the CRUSH1. 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) are available for use of the CRUSH1 and
- (2) add cover plate effect of shock absorber.

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 CRUSH2.

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

CRUSH2：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 豪

(1997年1月17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が用いられている。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である。このような背景から、簡易計算プログラム CRUSH2 を開発した。CRUSH2 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。

CRUSH2 は CRUSH1 の改良版であり、主要な改良点は下記の通りである。

- (1) 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) によっても使用できるようにした。
- (2) ショックアブソーバのカバープレートを計算モデルに追加した。

本報告書は UDM 法の説明、UDM 法と詳細計算法による計算結果の比較および CRUSH2 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。

## Contents

1. Introduction .....	1
2. Calculation Equation .....	4
2.1 Calculation Model .....	4
2.2 Vertical Drop Impact .....	5
2.3 Horizontal Drop Impact .....	6
2.4 Oblique Drop Impact .....	8
2.5 Convergence Method .....	13
2.6 Cover Plate of Shock Absorber .....	15
3. Benchmark Calculation .....	28
4. Computer Program .....	31
4.1 Program Description .....	31
4.2 Description of Input Data .....	32
4.3 Description of Output Data .....	32
5. Conclusions .....	40
Acknowledgments .....	40
Reference .....	40
Appendix A Sample Problem Input .....	41
Appendix B Sample Problem Output .....	42
Appendix C Graphical Output .....	43
Appendix D Job Control Data .....	49
Appendix E Program Abstract .....	50
Appendix F Program Source List .....	52

## 目 次

1. 緒 言 .....	1
2. 計 算 式 .....	4
2.1 計 算 モ デ ル .....	4
2.2 垂 直 落 下 衝 突 .....	5
2.3 水 平 落 下 衝 突 .....	6
2.4 傾 斜 落 下 衝 突 .....	8
2.5 収 束 計 算 法 .....	13
2.6 ショックアブソーバのカバープレート .....	15
3. ベンチマーク計算 .....	28
4. 計 算 プ ロ グ ラ ム .....	31
4.1 計 算 プ ロ グ ラ ム の 説 明 .....	31
4.2 入 力 デ タ .....	32
4.3 出 力 デ タ .....	32
5. 結 言 .....	40
謝 辞 .....	40
参考文献 .....	40
付録A 入 力 デ タ の 例 .....	41
付録B 出 力 デ タ の 例 .....	42
付録C 図 形 出 力 の 例 .....	43
付録D ジ ョ ブ 制 御 デ タ .....	49
付録E プ ロ グ ラ ム の 抄 錄 .....	50
付録F プ ロ グ ラ ム ソース リ ス ト .....	52

## 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 CRUSH2 as shown in Fig.1.1 has been developed. The CRUSH2 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) (1) as shown in Fig. 1.2.

CRUSH2 is a revised version of CRUSH1(2). Main revisions of the computer program are as follows;

- (1) not only main frame computer but also work stations(OS UNIX) and personalcomputer(OS Windows) are available for use of CRUSH2 and
- (2) add cover plate effect of shock absorber.

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 CRUSH2.

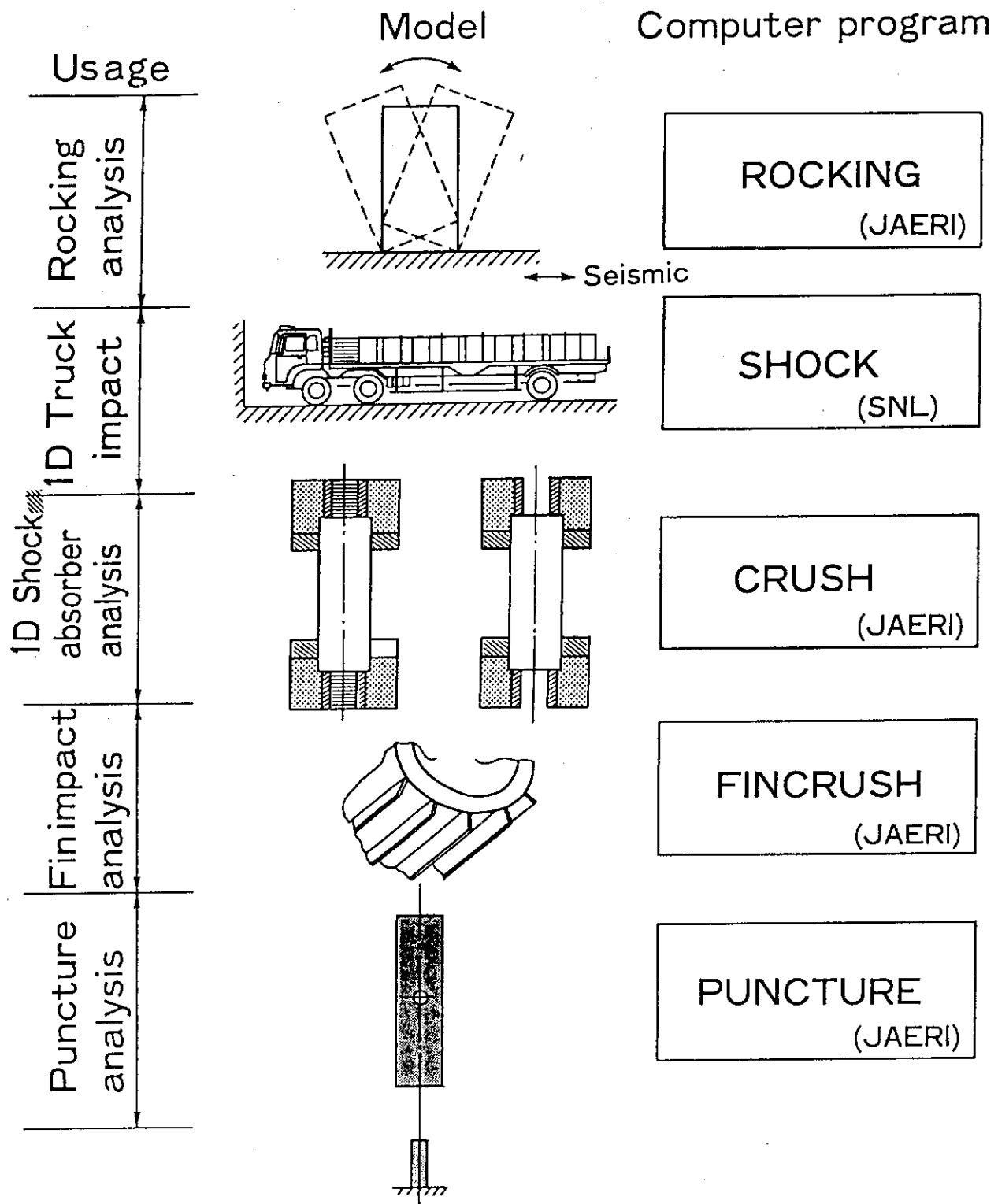


Fig. 1.1 Simplified analysis computer programs

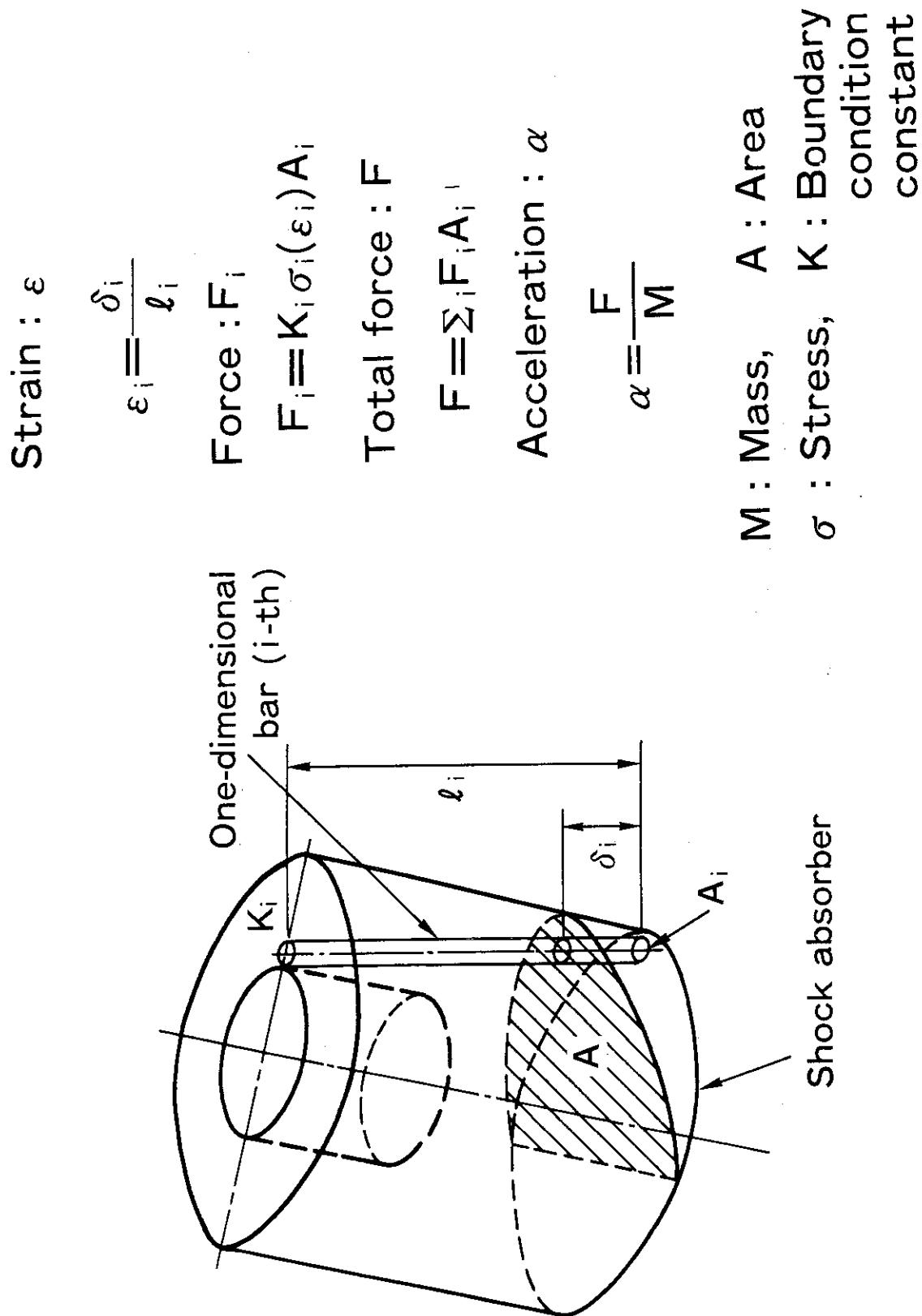


Fig. 1.2 Uni-axial displacement method

## 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  $K_i$  ( $i=1 \sim 5$ ) 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  $\varepsilon$  of an one-dimensional bar is

$$\varepsilon = \frac{\Delta \ell}{\ell} . \quad (2.1)$$

The force  $f$  of the one-dimensional bar is

$$f = K \sigma(\varepsilon) \Delta A . \quad (2.2)$$

where  $\ell$ ,  $\sigma$  and  $\Delta A$  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

$$F = \sum_i f_i . \quad (2.3)$$

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

$$E(\delta) = \int_0^\delta F d\ell . \quad (2.4)$$

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

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

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

where  $\gamma$  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  $\sigma_A$ ,  $\sigma_B$ ,  $\sigma_C$ ,  $\sigma_D$  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.

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

$$E = \int_0^Z F \, dZ , \quad (2.8)$$

where, for the region  $0 \sim R_1$

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

for the region  $R_1 \sim R_2$

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

for the region  $R_2 \sim R_3$

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

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  $R_4 \sim R_5$

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

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

$$0 \leq K_i \leq 1.$$

$\sigma_i$  ( $i = A, B, C, D$ ) 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.

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

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

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

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

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

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

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

$$\left. \begin{array}{l} U = y - h; \left\{ \begin{array}{l} U \leq 0, \text{ for } \Delta F = 0 \\ U > 0, \text{ for } \Delta F \neq 0 \end{array} \right. , \\ h = R_5 - \sqrt{R_5^2 - x^2} . \end{array} \right\} \quad (2.22)$$

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

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

$$\left. \begin{array}{l} y_1 = 2\sqrt{R_2^2 - x^2} \\ y = 2(\sqrt{R_5^2 - x^2} - \sqrt{R_2^2 - x^2}) . \\ U = \Delta y_1 + \Delta y_2 , \\ \sigma_A \left( \frac{\Delta y_1}{y_1} \right) = \sigma_B \left( \frac{\Delta y_2}{y_2} \right) . \end{array} \right\} \quad (2.25)$$

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

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

$$y = 2\sqrt{R_5^2 - x^2} . \quad (2.27)$$

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  $0 \sim R_3$ ,

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

where

$$y = \sqrt{r_5^2 - x^2} - \sqrt{R_3^2 - x^2}. \quad (2.29)$$

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

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

where

$$y = 2\sqrt{R_5^2 - x^2}. \quad (2.31)$$

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

$$\Delta F = K_4 \sigma_C \left( \frac{U}{y} \right) dx \cdot (Z_4 - Z_3), \quad (2.32)$$

where

$$y = \sqrt{R_5^2 - x^2} - \sqrt{R_3^2 - x^2}. \quad (2.33)$$

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

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

where

$$y = 2\sqrt{R_5^2 - x^2}. \quad (2.35)$$

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  $\theta$ .

$$\sigma_X = \sigma_A \cos^2\theta + \sigma_B \sin^2\theta , \quad (2.36)$$

where  $\sigma_A$ ,  $\sigma_B$  and  $\sigma_X$  are stresses in the wood whose grain direction is parallel, perpendicular and angle  $\theta$  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).

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

$$F = \int_S F_i dS , \quad (2.38)$$

$$E = \int_0^y F dy , \quad (2.39)$$

$$U = y - h_0 - h; \quad \begin{cases} U \leq 0, & \text{for } \Delta F = 0 \\ U > 0, & \text{for } \Delta F \neq 0 \end{cases} . \quad (2.40)$$

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  $P_1 \sim P_{12}$  and  $Q_1 \sim Q_4$  are as follows.

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

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

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

When  $\sigma_D \neq 0$

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

When  $\sigma_D = 0$

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

The lengths of  $h_1$ ,  $\ell_1$  and  $\ell_2$  are as follows.

- (a) If  $\ell_2$  is known ( $S < P_5$ ,  $S \leq 0$ ,  $\ell_1 = 0$  and  $K_h = 0$ ),  $h_0$ ,  $h_1$  and  $\ell_2$  are shown in Fig. 2.15(a).

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

When  $\sigma_D \neq 0$

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

When  $\sigma_D = 0$

$$\left. \begin{array}{l} h_1 = -\frac{S}{\sin\theta\cos\theta}; \text{ (region } P_1 \sim P_2), \\ h_1 = \frac{R_5}{\sin\theta}; \text{ (Region } P_2 \sim P_6). \end{array} \right\} \quad (2.46)$$

(b) If  $\ell_1$  and  $\ell_2$  are known ( $S < P_5$ ,  $S > 0$ ,  $\sigma_D \neq 0$ ,  $h_1 = 0$  and  $K_h = 0$ ),  $h_0$ ,  $\ell_1$  and  $\ell_2$  are shown in Fig. 2.15(b).

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

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

(c) If  $\ell_2$  is known ( $S < P_5$ ,  $S > 0$  and  $\sigma_D = 0$ ),  $h_0$ ,  $h_1$  and  $\ell_1$  are shown in Fig. 2.15(c) and are same as Eqs. (2.42), (2.43) and (2.43).

(d) If  $\ell_2$  is known ( $P_5 \leq S \leq Q_0$  and  $K_h = 1$ ),  $h_0$ ,  $h_1$ ,  $\ell_1$  and  $\ell_2$  are shown in Fig. 2.16(d).

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

When  $\sigma_D \neq 0$

$$\left. \begin{array}{l} h_1 = -\frac{S}{\sin\theta\cos\theta}, \\ \ell_1 = 0, \end{array} \right\}; \text{ (for } S \leq 0), \quad (2.50)$$

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

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

When  $\sigma_D = 0$

$$\left. \begin{array}{l} h_1 = \frac{R_1}{\sin\theta}, \\ l_1 = 0, \end{array} \right\}; \text{ (for } S \leq P_6) . \quad (2.53)$$

$$\left. \begin{array}{l} h_1 = \frac{1}{\sin\theta\cos\theta} (Z_2\sin\theta - S), \\ l_1 = 0, \end{array} \right\}; \text{ (for } S > P_6) . \quad (2.54)$$

(e) If  $l_1$  is known ( $Q_0 \leq S \leq P_1$ ,  $l_2=0$  and  $K_h=1$ ),  $h_1$  and  $l_1$  are shown in Fig. 2.16(e).

For region  $Q_0 \sim P_8$

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

$$\left. \begin{array}{l} l_1 = \frac{S}{\sin\theta\cos\theta}; \text{ (for } S < P_4) , \\ l_1 = \frac{R_5}{\sin\theta}; \text{ (for } S \geq P_4) . \end{array} \right\} \quad (2.56)$$

For region  $P_8 \sim P_1$

$$h_1 = \frac{R_3}{\sin\theta} , \quad (2.57)$$

$$\left. \begin{array}{l} l_1 = \frac{S}{\sin\theta\cos\theta}; \text{ (for } S < P_4) , \\ l_1 = \frac{R_5}{\sin\theta}; \text{ (for } S \geq P_4) . \end{array} \right\} \quad (2.58)$$

(f) If  $l_1$  is known ( $S > P_1$ ,  $l_2=0$  and  $K_h=1$ ),  $h_1$  and  $l_1$  are shown in Fig. 2.16(f).

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

$$\left. \begin{array}{l} l_1 = \frac{S}{\sin\theta\cos\theta}; \text{ (for } S < P_4) , \\ l_1 = \frac{R_5}{\sin\theta}; \text{ (for } S \geq P_4) . \end{array} \right\} \quad (2.60)$$

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

equation of the ellipsoid is as follows.

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

where

$$b = \frac{R_s}{\sin\theta} . \quad (6.62)$$

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

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

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

Force  $\Delta F$  at the sectional area ( $\Delta S$ ,  $\Delta x$ ) is as follows.

$$\Delta F = K \cdot \sigma \left( \frac{U}{\ell} \right) \Delta S \cdot \Delta x , \quad (2.65)$$

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

$$0 \leq K_i \leq 1 \quad (2.66)$$

where

- $K_1$  : for  $\sigma_D=0$ ; region  $P_3 \sim Q_3$ ,
- for  $\sigma_D \neq 0$ ; region  $P_1 \sim X_1$ ,
- $K_2$  : for  $\sigma_D=0$ ; region  $Q_3 \sim P_7$ ,
- for  $\sigma_D \neq 0$ ; region  $X_1 \sim X_2$ ,
- $K_3$  : for region  $P_{11} \sim Q_4$ ,
- $K_4$  : for region  $Q_4 \sim P_{12}$ .

## 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.

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

and

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

from above two equations

$$\left[ \begin{array}{cc} 1 & 1 \\ -\frac{K_A}{\ell_A} & \frac{K_B}{\ell_B} \end{array} \right] \left\{ \begin{array}{c} U_A \\ U_B \end{array} \right\} = \left\{ \begin{array}{c} U \\ \sigma_A^0 - \sigma_B^0 \end{array} \right\} . \quad (2.69)$$

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

## (2) Three materials

$$\left[ \begin{array}{ccc} 1 & 1 & 1 \\ -\frac{K_A}{\ell_A} & \frac{K_B}{\ell_B} & 0 \\ -\frac{K_A}{\ell_A} & 0 & \frac{K_K}{\ell_K} \end{array} \right] \left\{ \begin{array}{c} U_A \\ U_B \\ U_K \end{array} \right\} = \left\{ \begin{array}{c} U \\ \sigma_A^0 - \sigma_B^0 \\ \sigma_A^0 - \sigma_K^0 \end{array} \right\} . \quad (2.70)$$

## (3) Strain, stiffness and stress

Strain

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

Stiffness (see Fig. 2.20)

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

Stress (see Fig. 2.20)

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

2.6 Cover Plate of Shock Absorber

Modeling of cover plate as shown in Figs. from 2.21 to 2.23 are same as shock absorber.

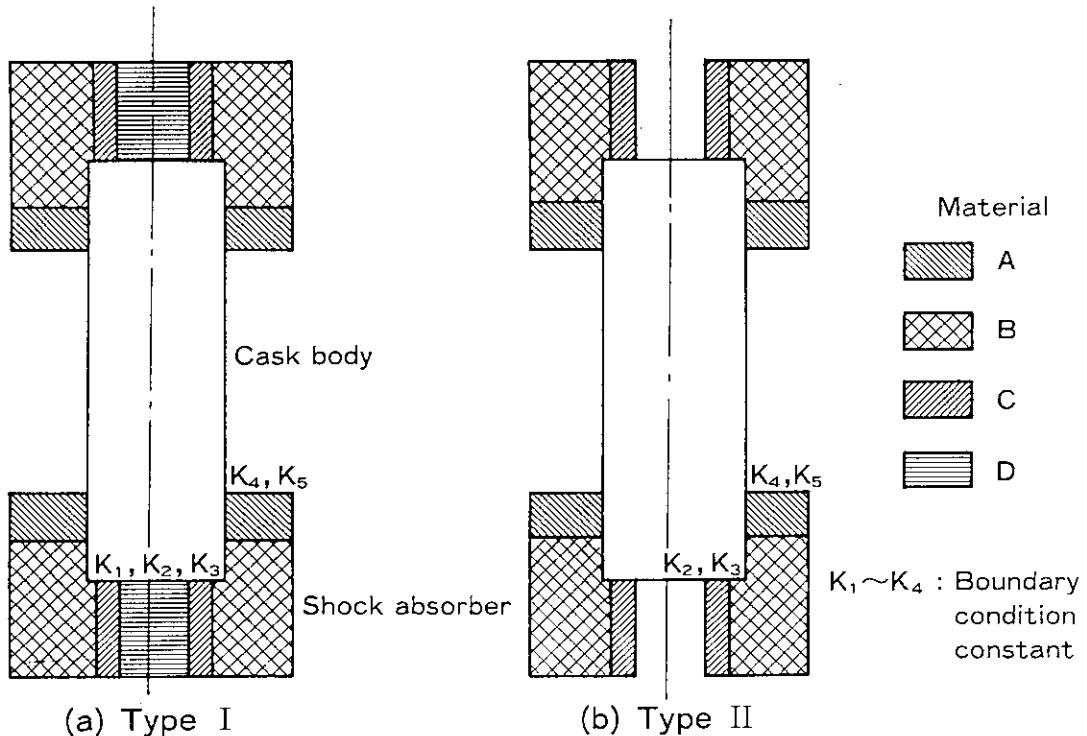


Fig. 2.1 Vertical drop model

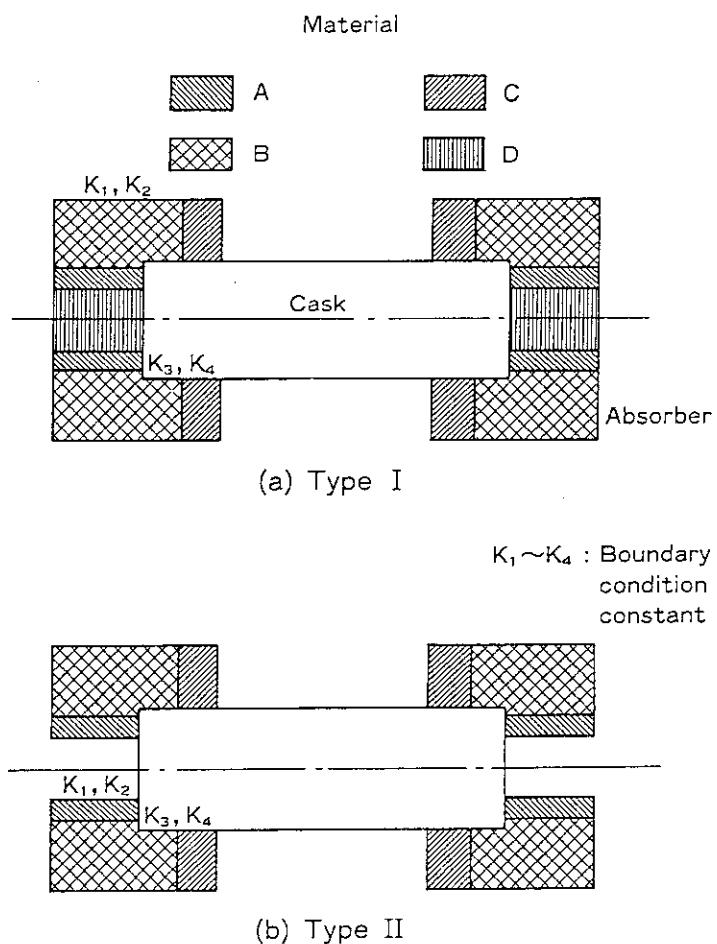


Fig. 2.2 Horizontal drop model

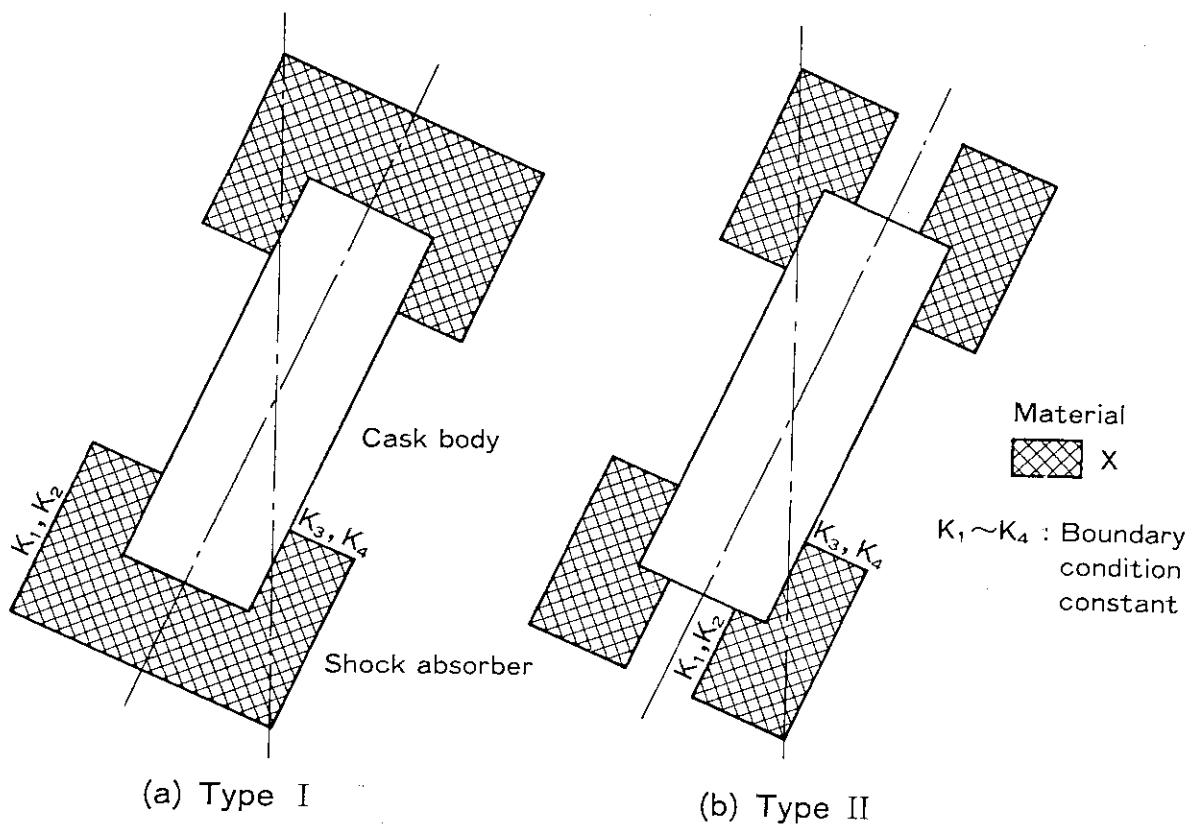


Fig. 2.3 Oblique drop model

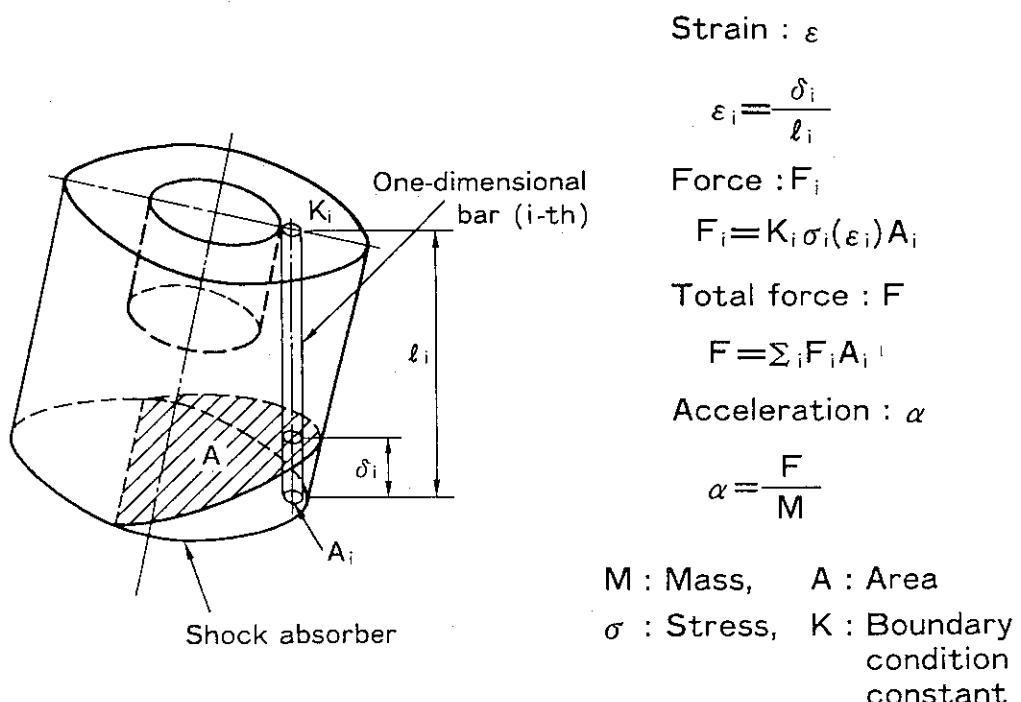
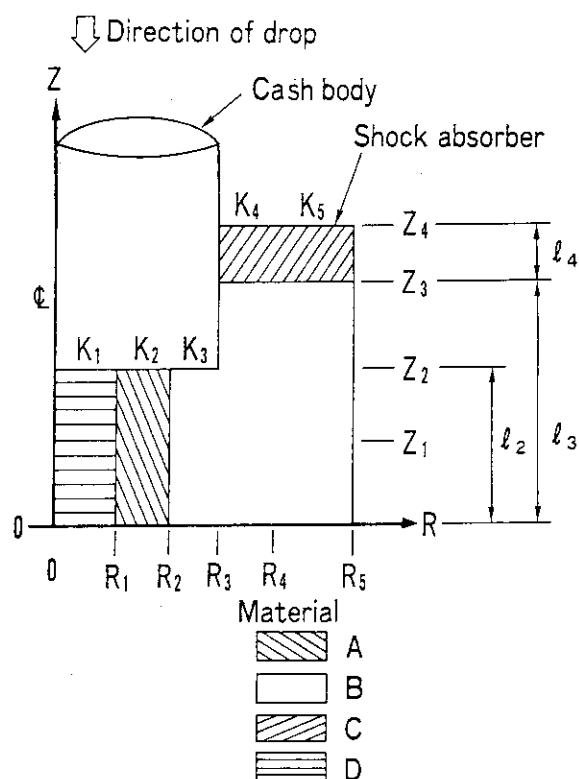


Fig. 2.4 Uni-axial displacement method



K<sub>1</sub>~K<sub>5</sub> : Boundary condition constant

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

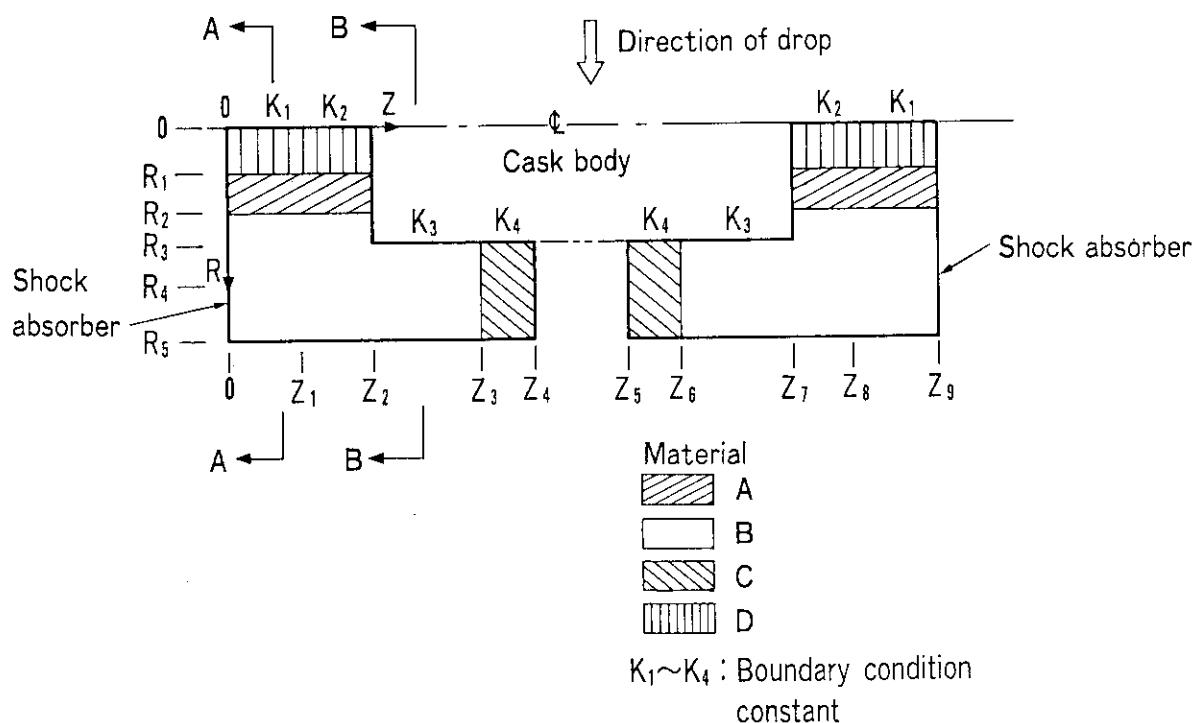


Fig. 2.6 Geometry and material in the case of horizontal drop model

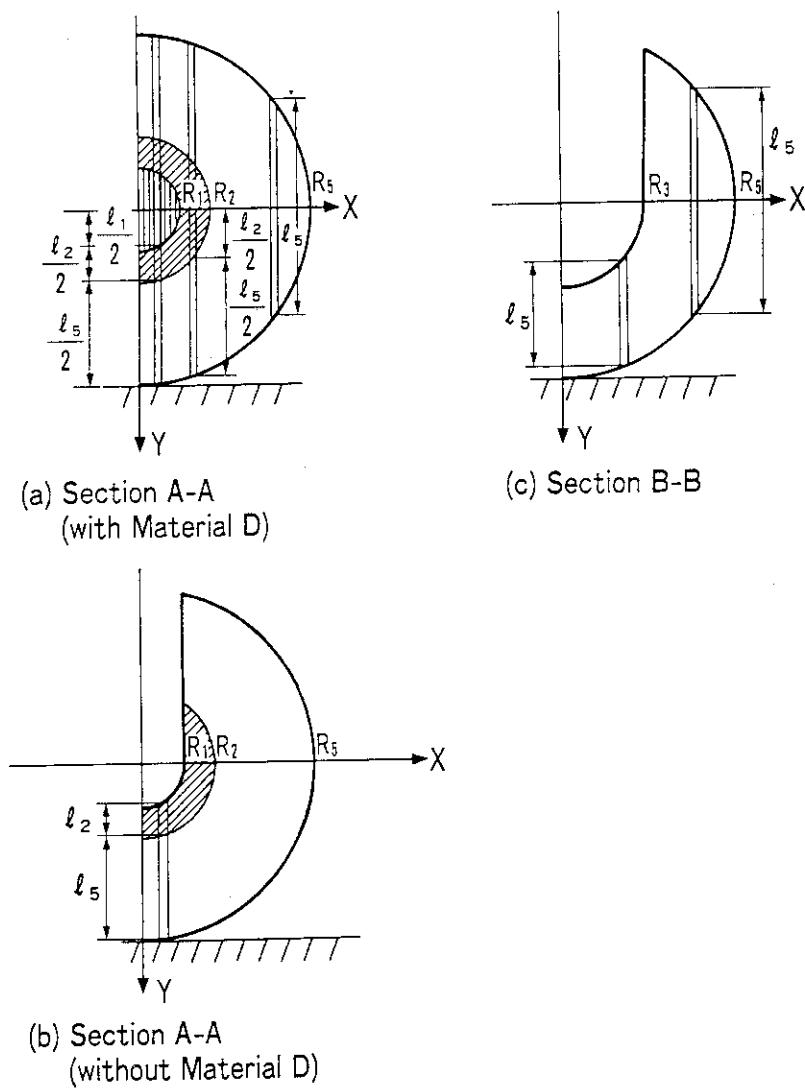


Fig. 2.7 Section view of horizontal drop model

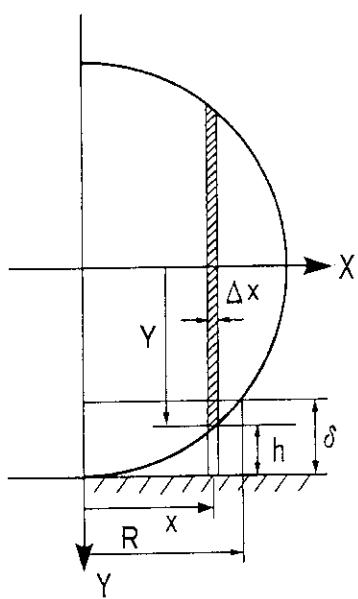


Fig. 2.8 Sectional area of shock absorber

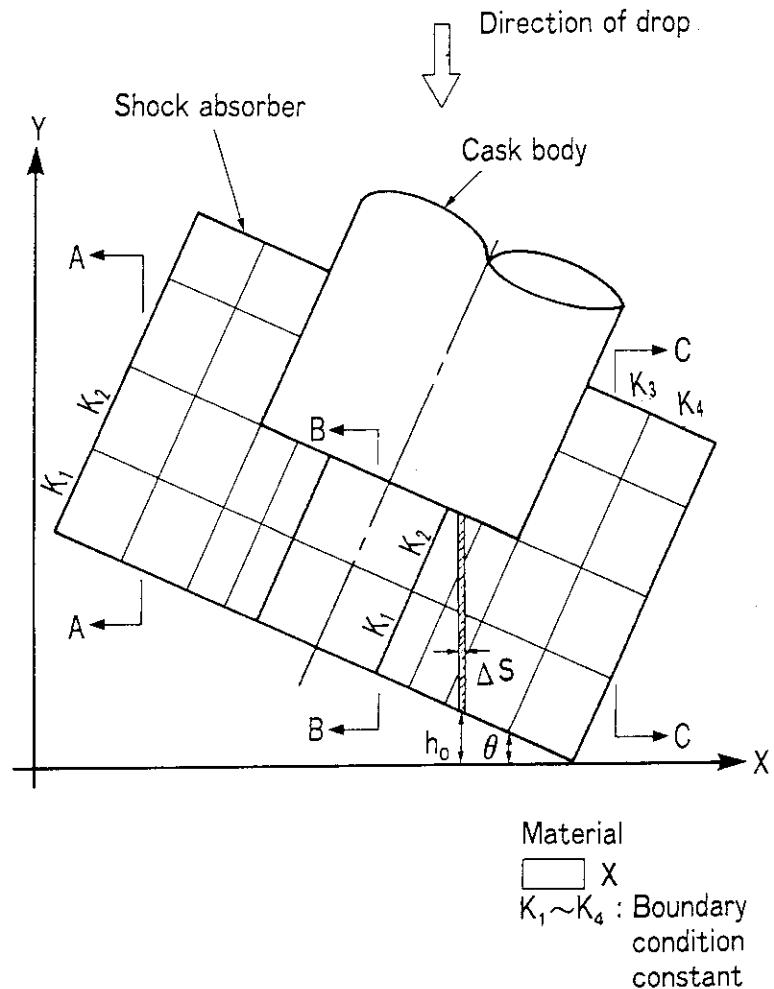


Fig. 2.9 Geometry and material in the case of oblique drop model

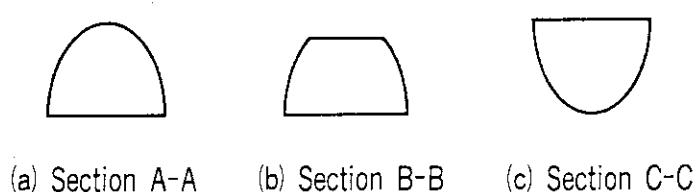


Fig. 2.10 Sectional figure of shock absorber in the case of oblique drop (I)

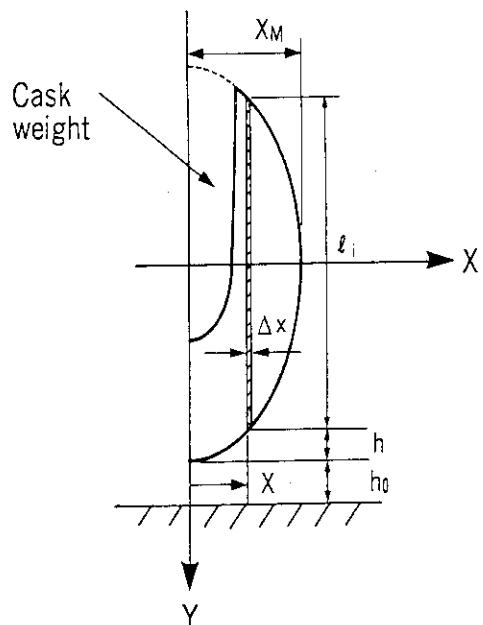


Fig. 2.11 Sectional figure of shock absorber in the case of oblique drop (II)

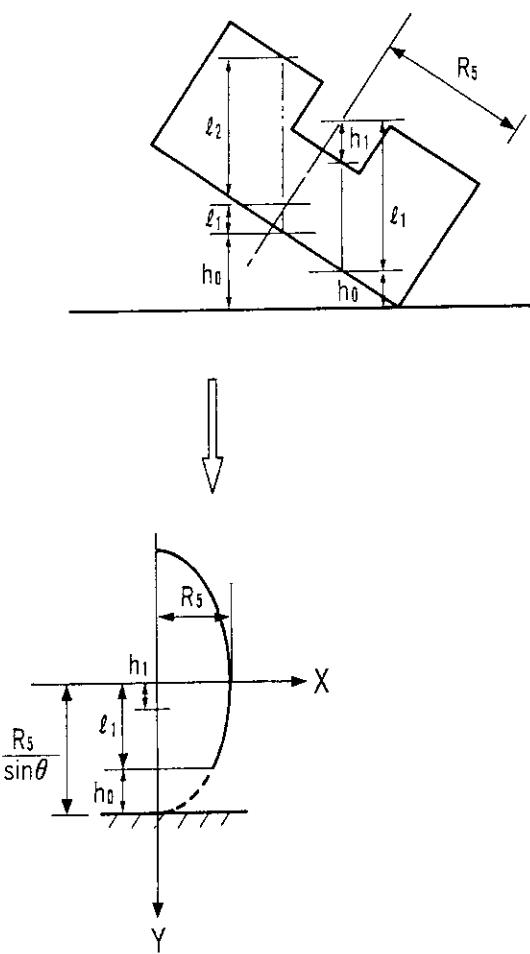


Fig. 2.12 Sectional figure of shock absorber in the case of oblique drop (III)

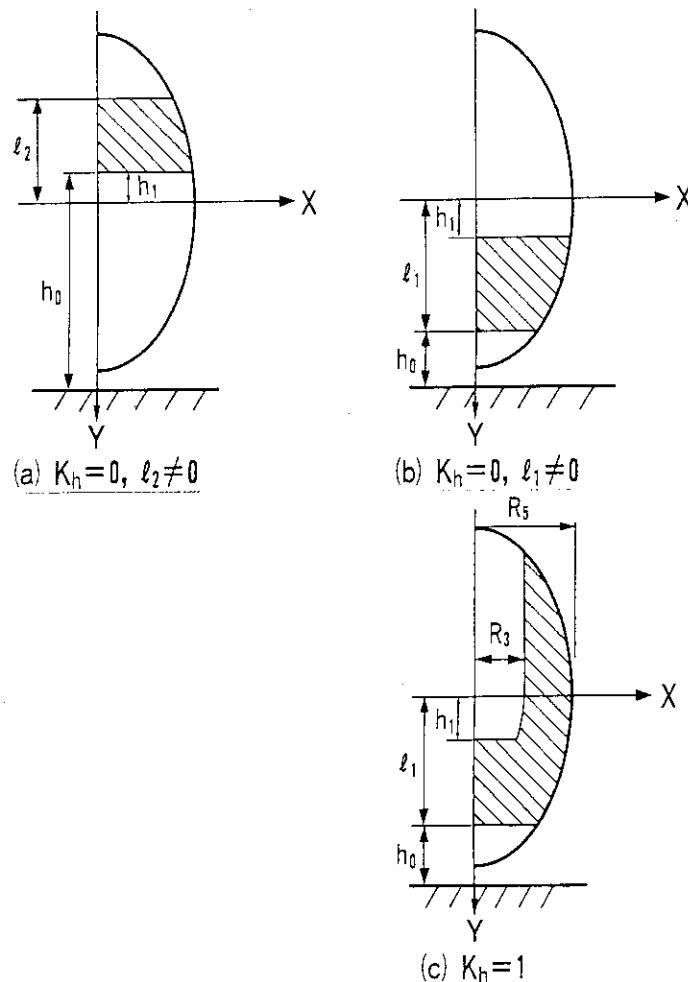


Fig. 2.13 Sectional figure of shock absorber in the case of oblique drop (IV)

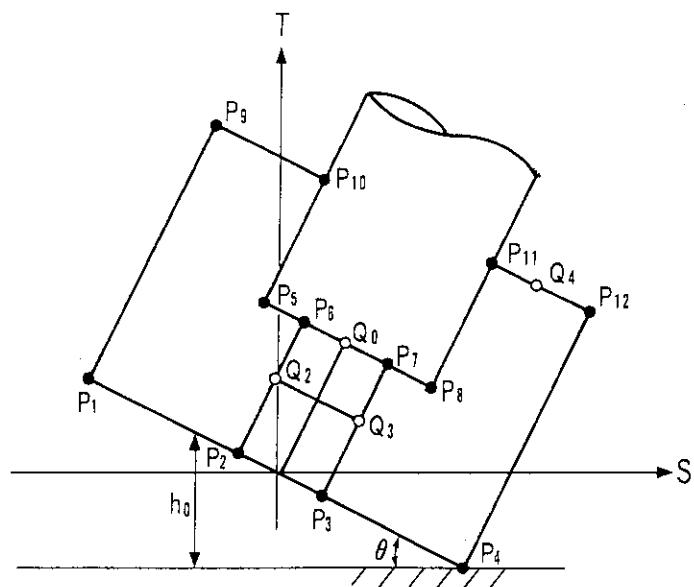


Fig. 2.14 Geometry of shock absorber in the case of oblique drop and S-T coordinate

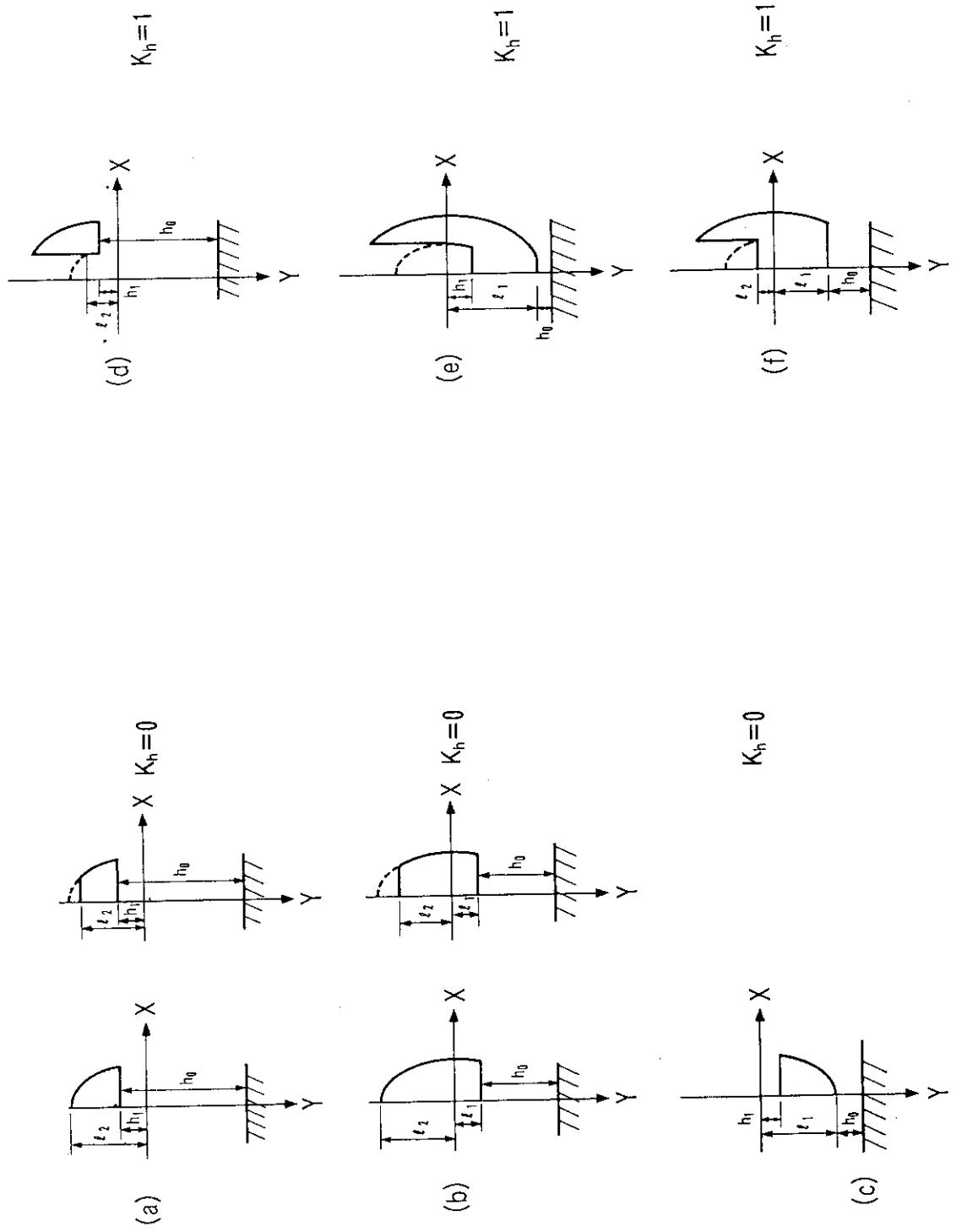


Fig. 2.15 Various sectional figure of shock absorber in the case of oblique drop ( I )

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

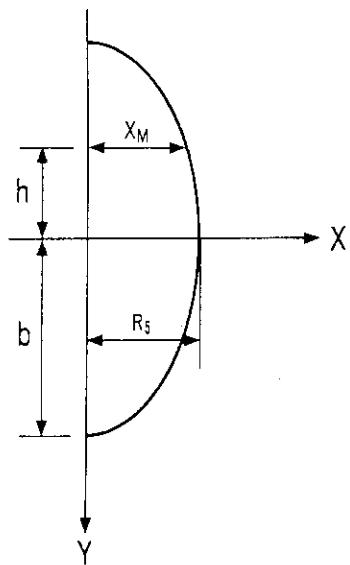


Fig. 2.17 Ellipsoid

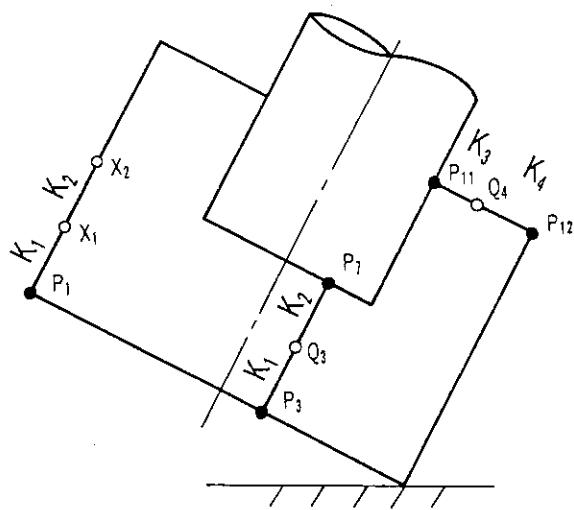


Fig. 2.18 Boundary condition constant in the case of oblique drop

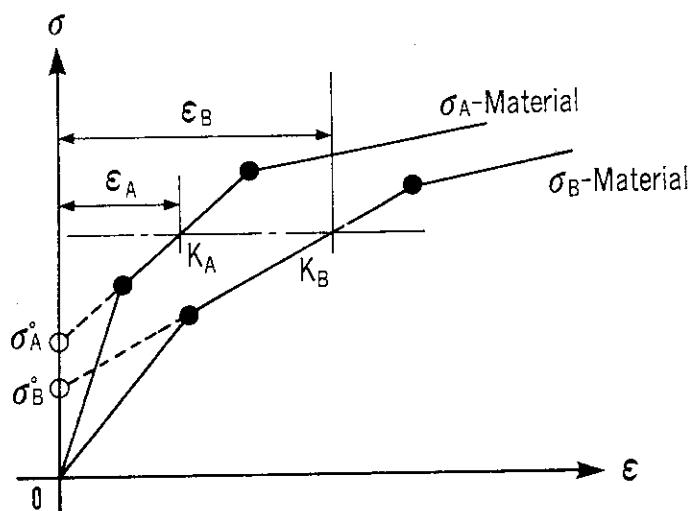


Fig. 2.19 Stress-strain curves of shock absorber materials

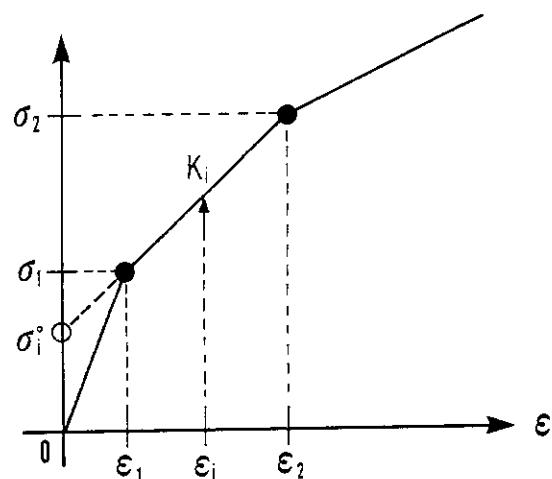


Fig. 2.20 Stress-strain curve of shock absorber material

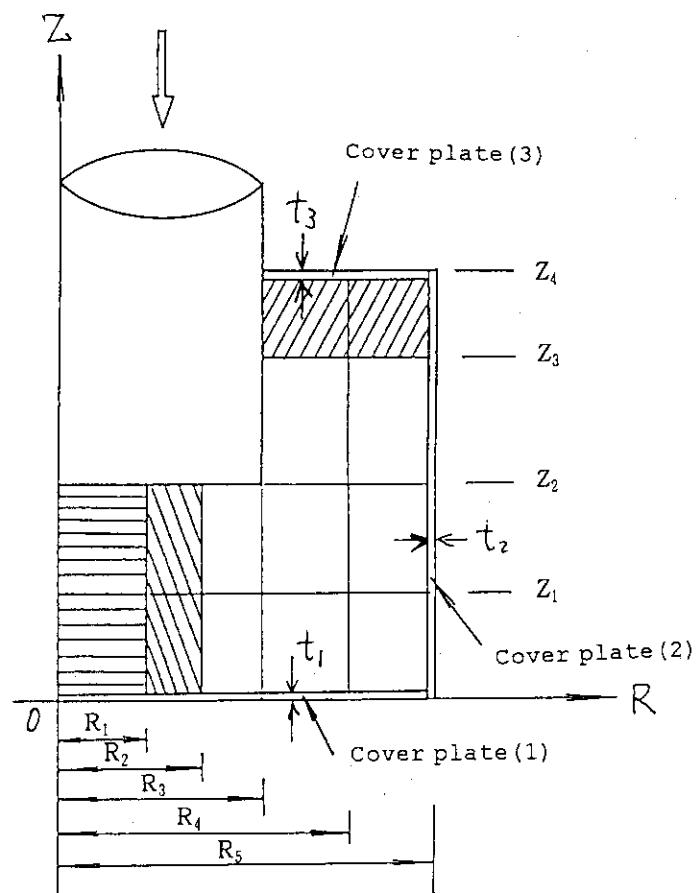


Fig. 2.21 Cover plate in the case of vertical impact model

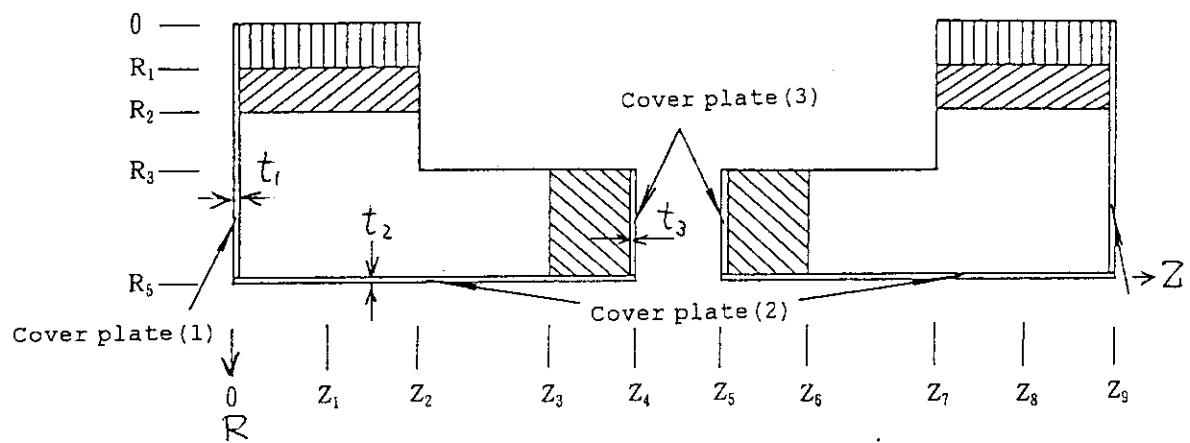


Fig. 2.22 Cover plate in the case of vertical impact model

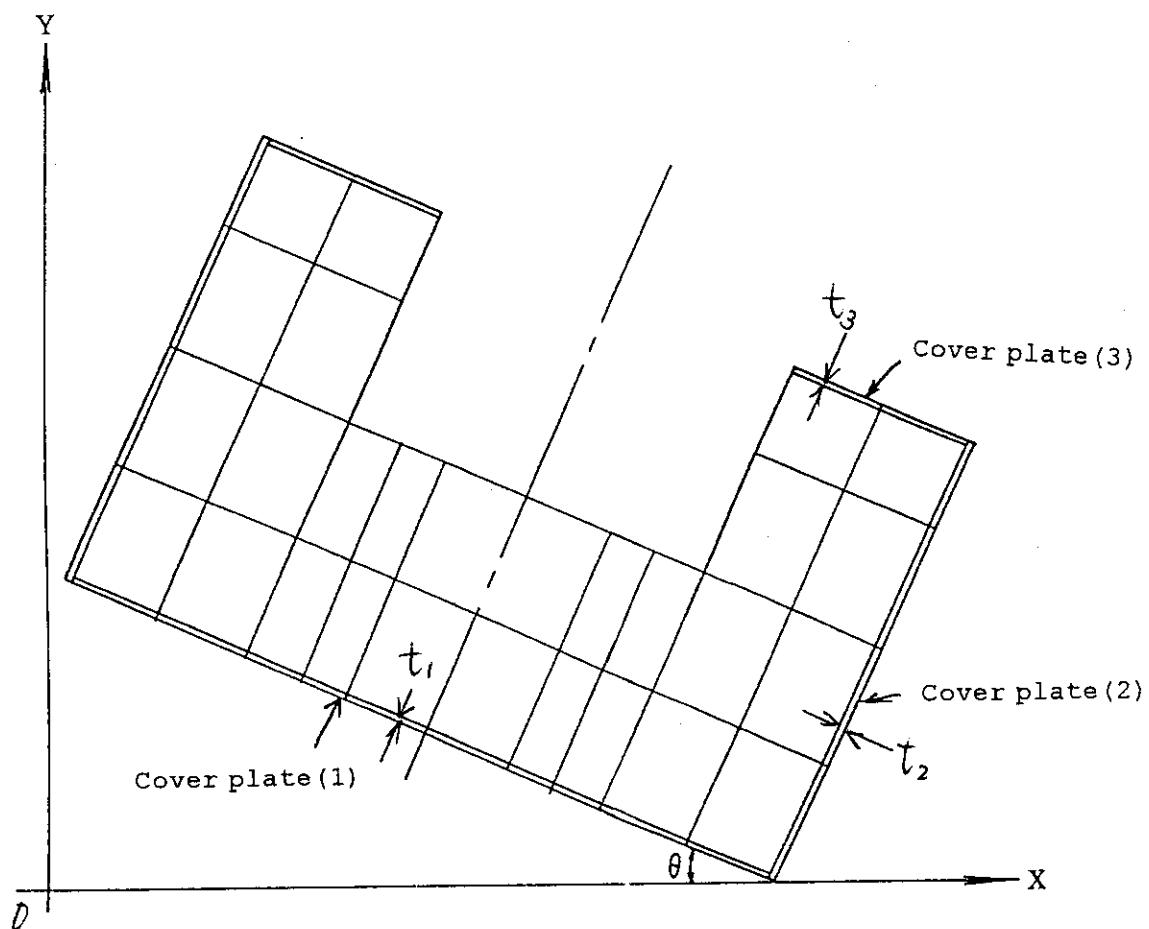


Fig. 2.23 Cover plate in the case of vertical impact model

### 3. Benchmark Calculation

In order to demonstrate the adequacy of the simplified computer program CRUSH2, 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		Detailed analysis DYNA3D	Experiment	Simplified analysis		Detailed analysis DYNA3D
		CRUSH1	CRUSH2			CRUSH1	CRUSH2	
Vertical	200	179	208	271 * (200) **	51	52	50	50
Corner	106	125	136	145 ( 97)	127	151	146	118
Horizontal	180	183	219	240 (182)	61	63	74	57

\* Value of low pass filter is 600 Hz.

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

(NUPAC 125B cask 1/4 scale model).

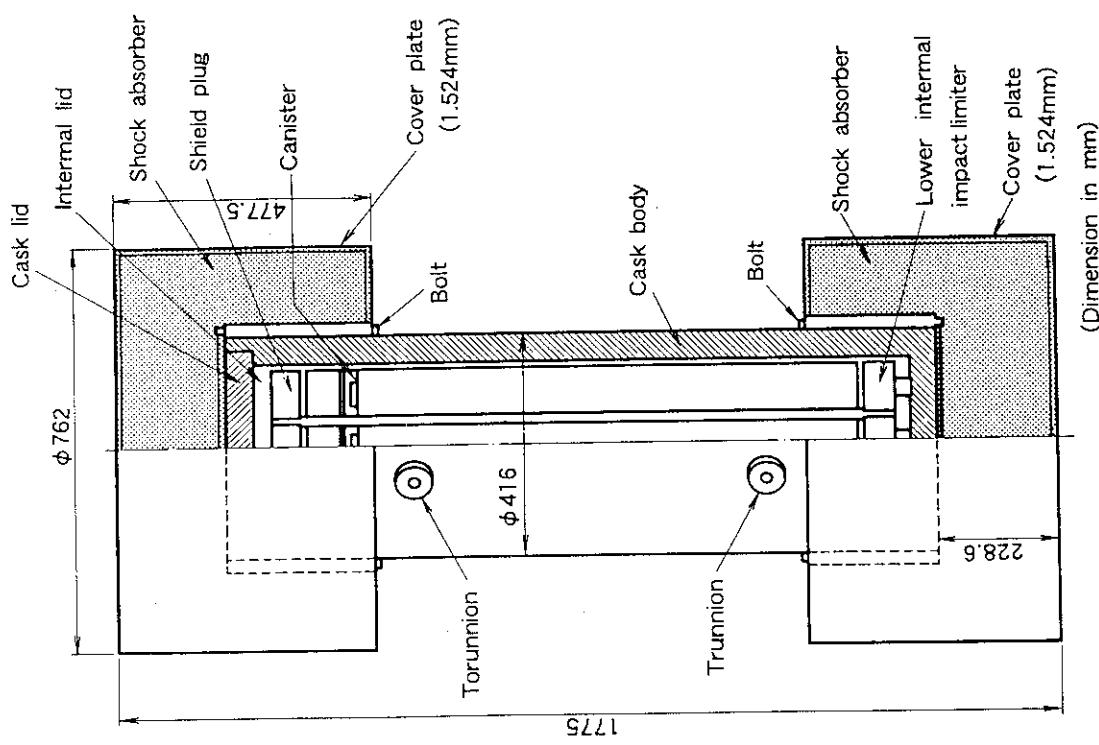


Fig. 3.1 Shipping cask NUPAC 125B (1/4 scale model)

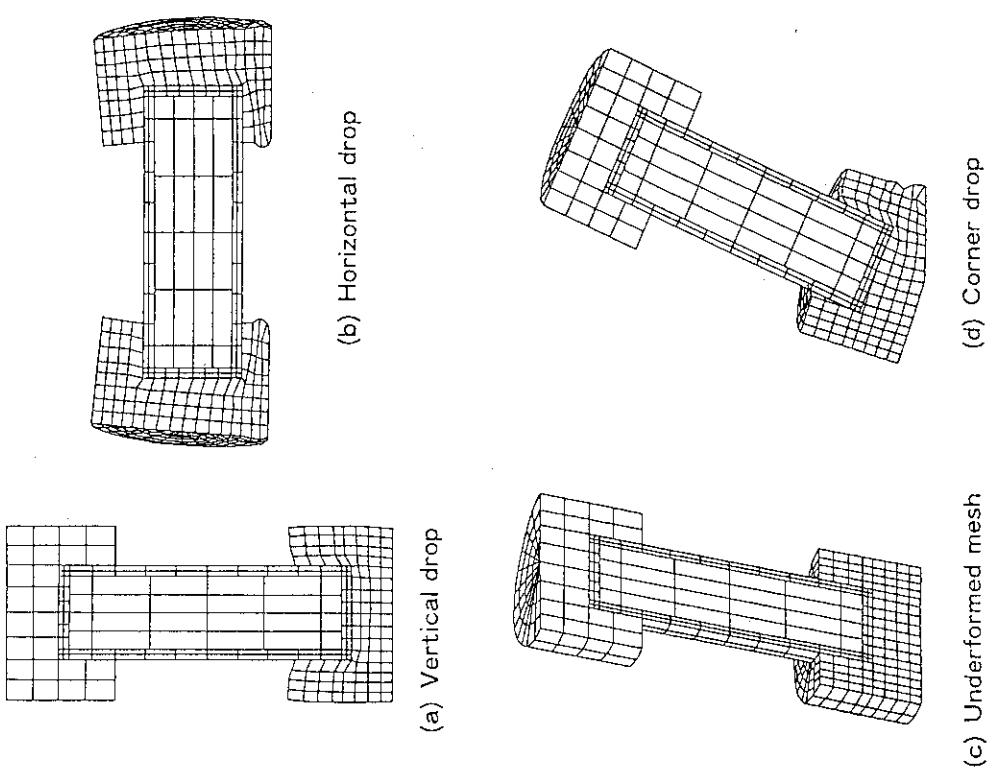


Fig. 3.2 Deformed shape after 9m drop impact  
(NUPAC 125B cask 1/4 scale model)

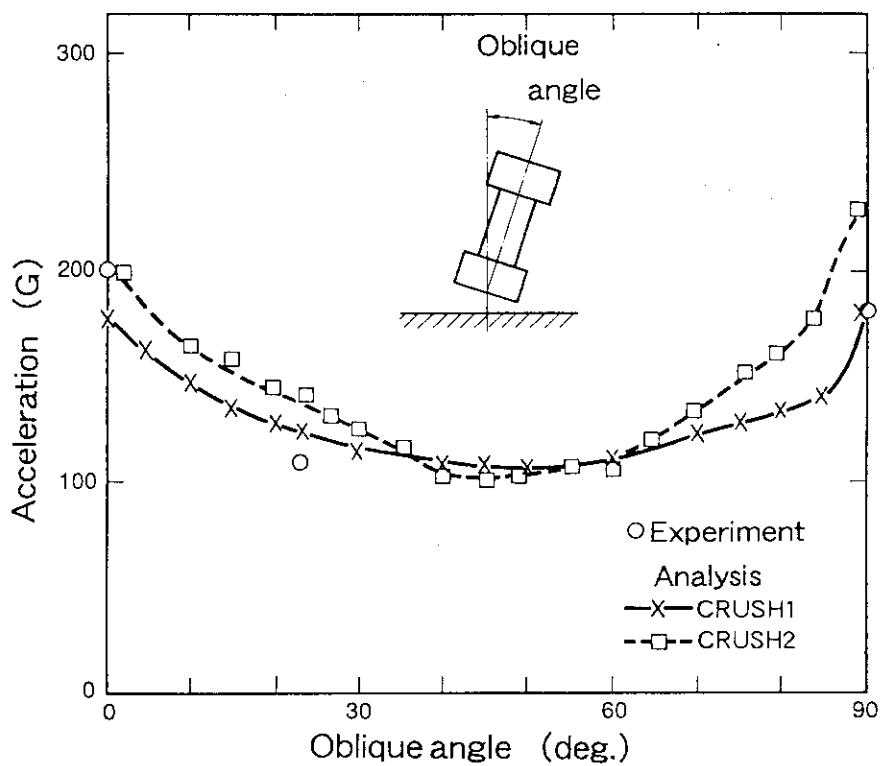


Fig. 3.3 Comparison between simplified analysis and experiment on impact acceleration

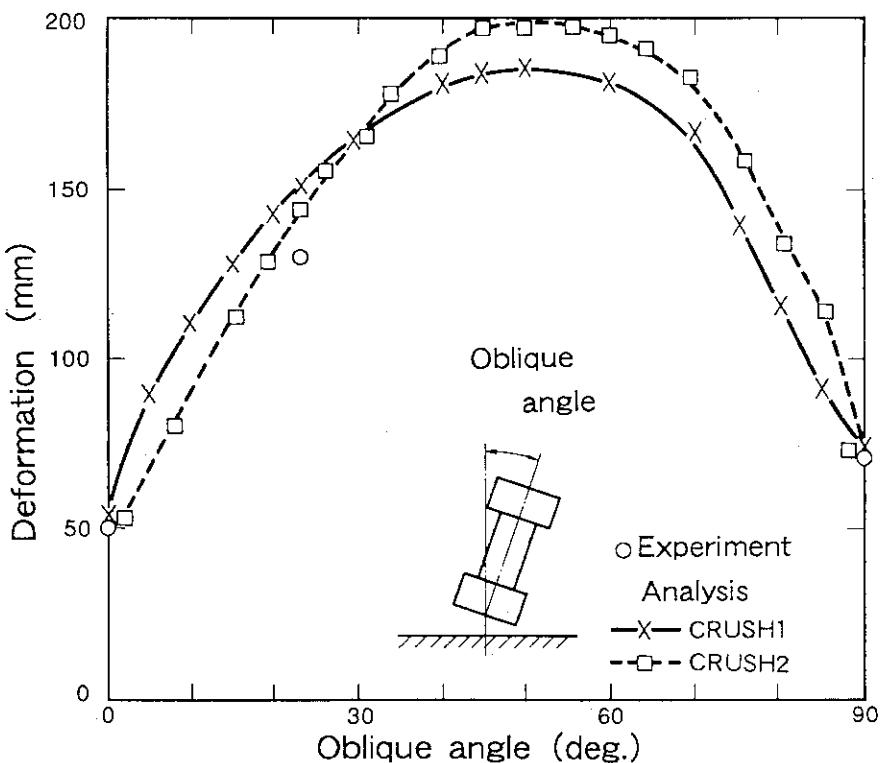


Fig. 3.4 Comparison between simplified analysis and experiment on shock absorber deformation

## 4. Computer Program

### 4.1 Program Description

The computer program CRUSH2 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 CRUSH2 consists of a main routine and thirteen subroutines that are MAIN, CARDIN, VERT, HORIZ, CORNER, MULTI, GETMAT, EQSOLV, GETSIG, FITSTP, SIMPLL, MPLOT, DPLOT and UPLOT. The eleven basic subroutines for plotting are GOSCAL, XYSCL, PLTBGN, PLPEN, PLINEs, PLMARK, PSYM, PNUMBB, PLTOR and PLREND. Overall structure of CRUSH2 is shown in Fig. 4.1. Functions of subroutines are as follows:

```

MAIN   : initializes start of run,
CARDIN : reads input data,
VERT   : computes acceleration and deformation in the case
          of vertical (head-on or bottom-on) drop impact,
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,
SIMPLL : computes cover plate length of shock absorber using
          Simpson interpolation method,
MPLOT  : geometry plot,
DPLOT  : X-Y curve plot,
UPLOT  : deformation plot.

```

The plot basic subroutines are as follows:

The plot basic subroutines are as follows:

GOSCAL : scaling of geometry plot,  
XYSCAL : scaling of X-Y plot  
PLTBGN : initialization of plotter,  
PLPEN : change plotter pen size,  
PLINES : draw line,  
PLMARK : plot round mark,  
PSYM : write letter,  
PNUMB : write number,  
PNUMBB : write number,  
PLTEOR : plot new figure,  
PLTEND : plot end.

A macroscopic flow chart of CRUSH2 is shown in Fig. 4.2.

#### 4.2 Description of Input Data

This section describes the input data required by CRUSH2. 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 geo metry unit selection and options for output plotting. The input instruction is simple and easy 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 CRUSH2. 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

CRUSH2 provides users with graphical output of the deformation, relationship acceleration and deformation, dissipation energy and deformation, acceleration and dissipation energy and so forth.

Table 4.1 Input data for CRUSH2

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). 'HORIZONTAL':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 $1.0 \times 10^{-6}$ .
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, $0.0 < EPMAX < 1.0$ . If EPMAX is blank or zero, EPMAX equal to 1.0,

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
<b>Data set No. 3:Geometry data.</b>			
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 do exist. 'NONE':material σ do not exist.
41 - 50	F10.0	T1	Thickness of cover plate of shock absorber.
51 - 60	F10.0	T2	Thickness of cover plate of shock absorber.
61 - 70	F10.0	T3	Thickness of cover plate of shock absorber.
<b>Data set No. 4:Options for input data check and plotting.</b>			
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.
<b>Data set No. 5:Geometry of shock absorber(I).</b>			
1 - 8	2A4	NAME	Flag for coordinate of radial direction. 'RCOOR' .
9 - 10	2X	—	Blank.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
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.
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 - 8	2A4	NAME	Flag for data type. 'MESH'

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
9 - 10	2X	-	Blank.
11 - 15	I5	NPART	Number of partitions in the X-Y plain of horizontal or oblique drop case. 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: Stress-strain data of shock absorber.

1 - 8	2A4	NAME	Flag for material data identification. 'SIGA' : $\sigma_A - \varepsilon_A$ stress-strain data of shock absorber(see Fig. 2.5, 2.6 and 2.9). 'SIGB' : $\sigma_B - \varepsilon_B$ stress-strain data of shock absorber. 'SIGC' : $\sigma_C - \varepsilon_C$ stress-strain data of shock absorber. 'SIGD' : $\sigma_D - \varepsilon_D$ stress-strain data of shock absorber. 'SIGX' : $\sigma_X - \varepsilon_X$ stress-strain data of shock absorber. 'SIG1' : $\sigma_1 - \varepsilon_1$ stress-strain data of cover plate(see Fig. 2.21, 2.22 and 2.23). 'SIG2' : $\sigma_2 - \varepsilon_2$ stress-strain data of
-------	-----	------	---

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
			cover plate. 'SIG3' : $\sigma_3 - \epsilon_3$ stress-strain data of cover plate.
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 of materials for shock absorber or cover plate of shock absorber.			
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.
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.
			Repeat 8B data set for NN data.
Repeat 8A and 8B data set for SIGA, SIGB, SIGD, SIGX, SIG1, SIG2 and SIG3 stress-strain data.			
Data set No. 9A: Force-displacement data for shock absorber structure.			
1 - 8	2A4	NAME	Identification of shock absorber model. 'FORCE' : Using force and deformation data of shock absorber.
			' ' : Using stress-strain data of shock absorber.
9 - 10	2X	-	Blank.
11 - 15	F10.0	NN	Number of force-displacement data.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
Data set No. 9B: force-displacement data of shock absorber.			
1 - 10	F10. 0	DISPL(1)	Displacement.
11 - 20	F10. 0	FORCE(1)	Force.
21 - 30	F10. 0	DISPL(2)	Displacement.
31 - 40	F10. 0	FORCE(2)	Force.
41 - 50	F10. 0	DISPL(3)	displacement.
51 - 60	F10. 0	FORCE(3)	Force.
61 - 70	F10. 0	DISPL(4)	Displacement.
71 - 80	F10. 0	FORCE(4)	Force.
Repeat 9B data set for NN data.			

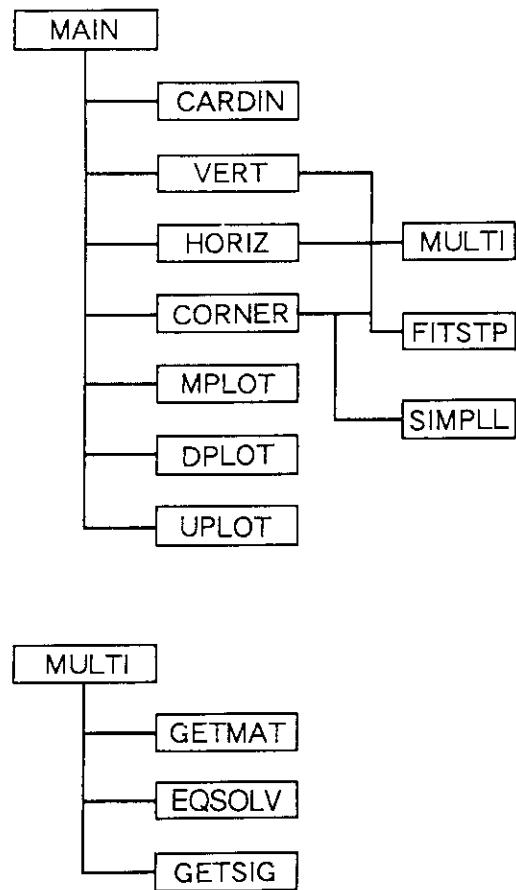


Fig. 4.1 Structure of computer program CRUSH2

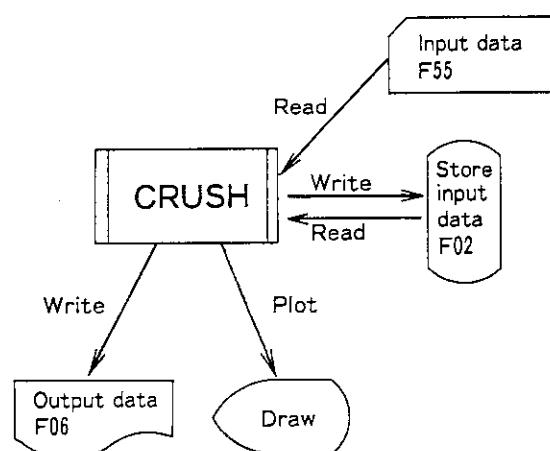


Fig. 4.2 Program flow

## 5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask body and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH2 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. CRUSH2 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

## Acknowledgements

The author is indebted Dr. Kazuo Asada of Mitsubishi 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).
- (3) Ikushima, T., "CRUSH1: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-Data/code 96-025 (1996).

## 5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask body and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH2 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. CRUSH2 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

## Acknowledgements

The author is indebted Dr. Kazuo Asada of Mitsubishi 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).
- (3) Ikushima, T., "CRUSH1: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-Data/code 96-025 (1996).

## 5. Conclusions

In regard to the evaluation of the maximum acceleration of the cask body and the maximum deformation of the shock absorber on the drop impact, a simplified computer program CRUSH2 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. CRUSH2 is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

## Acknowledgements

The author is indebted Dr. Kazuo Asada of Mitsubishi 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).
- (3) Ikushima, T., "CRUSH1: A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-Data/code 96-025 (1996).

## Appendix A Sample Problem Input

## INPUT DATA ECHO

DATA SEQ. NO.	1	2	3	4	5	6	7	8
1	TITLE	NUPAC-12SB	MODEL	DROP	(VERTICAL)			1
2	VERTICAL	1283.6	9000.0	0.00	2.0	0.0		2
3	LENGTH	1775.0	208.0	SIGD	1.524	1.524		3
4	RUN	PLOT						4
5	RCOOR	100.0	220.0	247.0	350.0	380.0		5
6	ZCOOR	100.0	224.0	350.0	474.0			6
7	MESH		1.0	1.0	1.0	0.75	0.75	7
8	SIGA	24	0.75					8
9	0.0	0.0	0.02	0.2	0.03	0.27	0.04	9
10	0.05	0.3	0.25	0.3	0.33	0.33	0.4	10
11	0.45	0.4	0.5	0.44	0.55	0.52	0.58	11
12	0.6	0.61	0.62	0.68	0.64	0.74	0.66	12
13	0.68	0.91	0.7	1.06	0.72	1.20	0.74	13
14	0.76	1.75	0.78	2.18	0.8	2.78	0.82	14
15	SIGB	24	0.75					15
16	0.0	0.0	0.02	0.2	0.03	0.27	0.04	16
17	0.05	0.3	0.25	0.3	0.33	0.33	0.4	17
18	0.45	0.4	0.5	0.44	0.55	0.52	0.58	18
19	0.6	0.61	0.62	0.68	0.64	0.74	0.66	19
20	0.68	0.91	0.7	1.06	0.72	1.20	0.74	20
21	0.76	1.75	0.78	2.18	0.8	2.78	0.82	21
22	SIGC	24	0.75					22
23	0.0	0.0	0.02	0.2	0.03	0.27	0.04	23
24	0.05	0.3	0.25	0.3	0.33	0.33	0.4	24
25	0.45	0.4	0.5	0.44	0.55	0.52	0.58	25
26	0.6	0.61	0.62	0.68	0.64	0.74	0.66	26
27	0.68	0.91	0.7	1.06	0.72	1.20	0.74	27
28	0.76	1.75	0.78	2.18	0.8	2.78	0.82	28
29	SIGD	24	0.75					29
30	0.0	0.0	0.02	0.2	0.03	0.27	0.04	30
31	0.05	0.3	0.25	0.3	0.33	0.33	0.4	31
32	0.45	0.4	0.5	0.44	0.55	0.52	0.58	32
33	0.6	0.61	0.62	0.68	0.64	0.74	0.66	33
34	0.68	0.91	0.7	1.06	0.72	1.20	0.74	34
35	0.76	1.75	0.78	2.18	0.8	2.78	0.82	35
36	SIG1	3	1.0					36
37	0.000	0.0	0.002095	44.0	0.103095	65.0		37
38	SIG2	3	0.75					38
39	0.000	0.0	0.002095	44.0	0.103095	65.0		39
40	SIG3	3	1.0					40
41	0.000	0.0	0.002095	44.0	0.103095	65.0		41
42								42

\* \* \* INPUT DATA END \* \* \*

## Appendix B Sample Problem Output

## CRUSH2 MODEL DATA

```

1   2   3   4   5   6   7   8
.....0.....0.....0.....0.....0.....0.....0.....0.....0
1- 'TITLE NUPAC-125B MODEL DROP (VERTICAL)
2- VERTICAL 1283.6 9000.0 0.00 2.0 0.0 MM
3- LENGTH 1775.0 208.0 SIGD 1.524 1.524 1.524
4- RUN PLOT
5- RCGDR 100.0 220.0 247.0 350.0 380.0
6- ZCGDR 100.0 224.0 350.0 474.0
7-
8- MESH      1.0   1.0   1.0   0.75   0.75
9- SIGA     24    0.75
10- 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    0.75
17- 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    0.75
24- 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    0.75
31- 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
37- SIG1     3     1.0
38- 0.000   0.0    0.002095 44.0   0.103095 65.0
39- SIG2     3     0.75
40- 0.000   0.0    0.002095 44.0   0.103095 65.0
41- SIG3     3     1.0
42- 0.000   0.0    0.002095 44.0   0.103095 65.0

```

..... MODEL WEIGHT = 1283.60 (KG)  
 ..... MODEL HEIGHT = 9000.00 (MM)  
 ..... CORNER ANGLE = 0.00 (DEG.)  
 ..... INCREMENT DISP= 2.00 (MM)

..... K-FACTOR = 1.0000 1.0000 1.0000 0.7500 0.7500

## Appendix B (Continued)

NUPAC-125B MODEL DROP		(VERTICAL)		
MODEL TYPE = VERTICAL		TOTAL ENERGY = 11552.40 (KG-M)		
STEP	DEPTH (MM)	FORCE (KG)	ENERGY (KG-M)	ACCELERATION (G)
1	2.00	140335.74	280.67	109.33
2	4.00	161782.53	604.24	126.04
3	6.00	180303.02	964.84	140.47
4	8.00	193792.33	1352.43	150.98
5	10.00	203818.99	1760.07	158.79
6	12.00	211335.79	2182.74	164.64
7	14.00	218082.50	2618.90	169.90
8	16.00	222062.31	3063.03	173.00
9	18.00	225700.08	3514.43	175.83
10	20.00	229014.53	3972.46	178.42
11	22.00	232030.51	4436.52	180.77
12	24.00	234953.24	4906.42	183.04
13	26.00	237347.43	5381.12	184.91
14	28.00	239741.63	5860.60	186.77
15	30.00	242135.82	6344.87	188.64
16	32.00	244530.01	6833.93	190.50
17	34.00	246924.21	7327.78	192.37
18	36.00	249318.40	7826.42	194.23
19	38.00	251712.59	8329.84	196.10
20	40.00	254106.79	8838.06	197.96
21	42.00	256500.98	9351.06	199.83
22	44.00	258895.18	9868.85	201.69
23	46.00	261289.37	10391.43	203.56
24	48.00	263683.56	10918.79	205.43
25	50.00	266077.76	11450.95	207.29
26	50.20	266317.18	11504.21	207.48
27	50.38	266533.58	11552.40	207.65

Appendix C Graphical Output

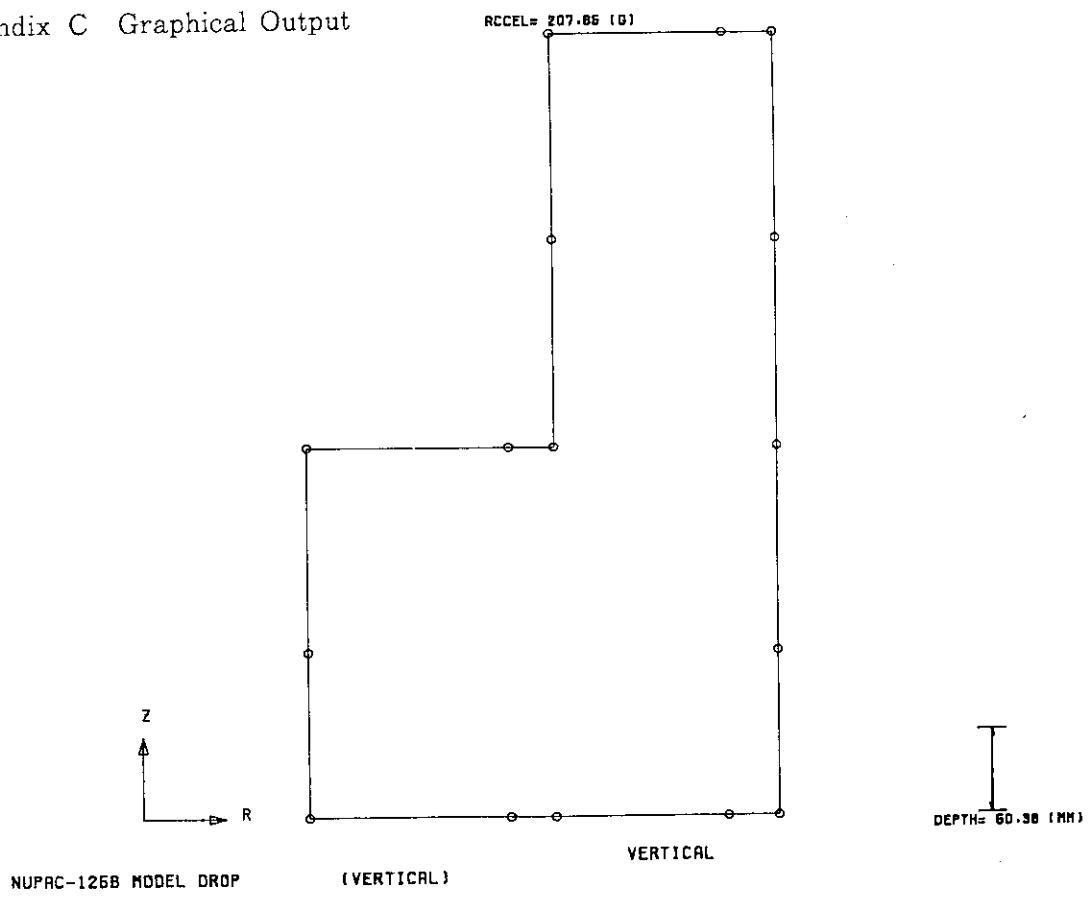


Fig. C.1 Graphical Output of CRUSH2 (1)

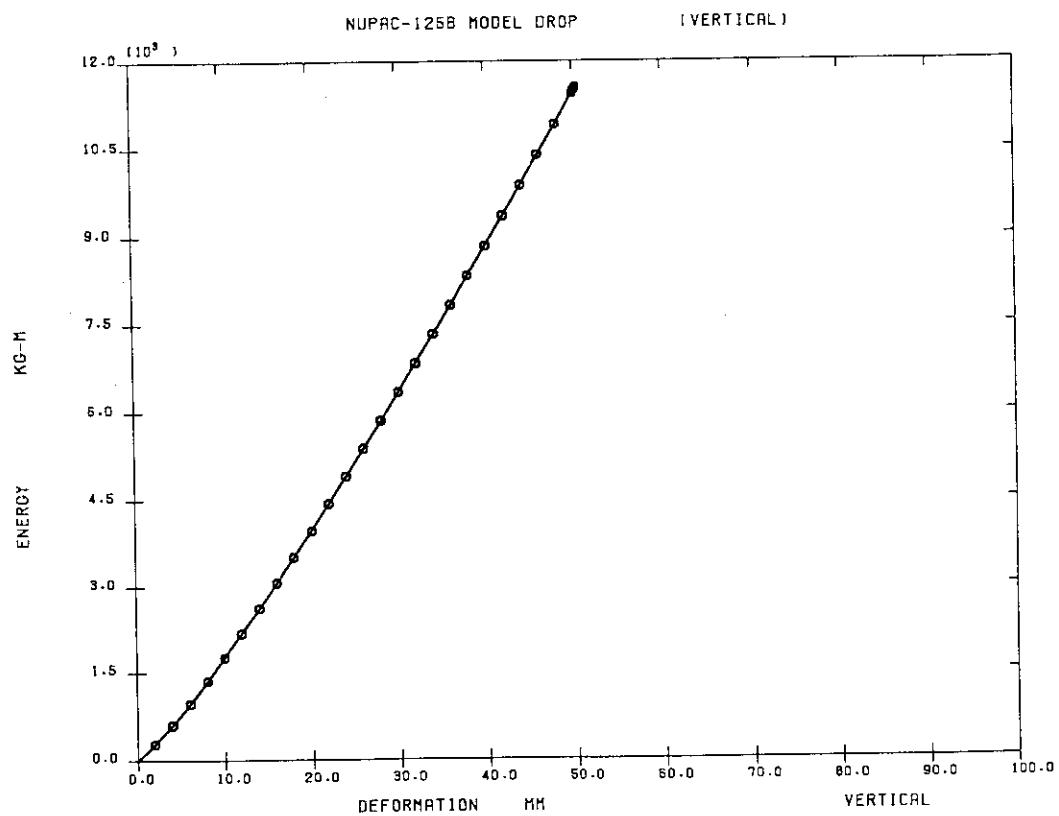


Fig. C.2 Graphical Output of CRUSH2 (2)

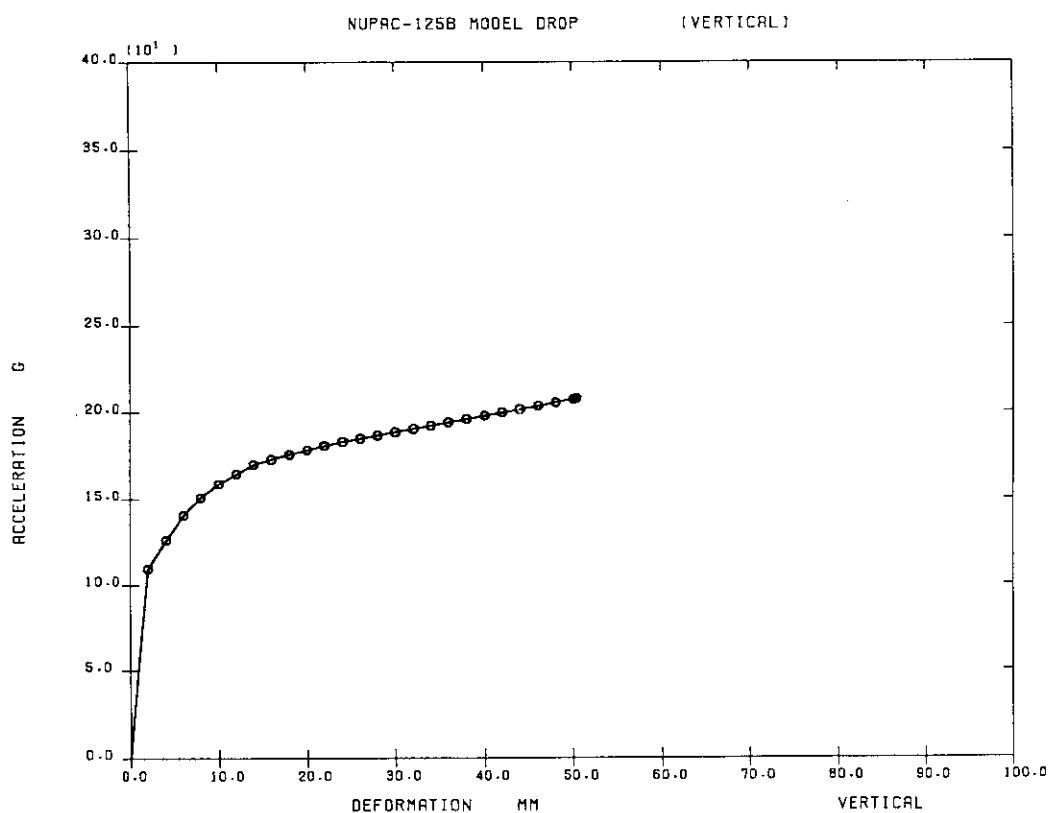


Fig. C.3 Graphical Output of CRUSH2 (3)

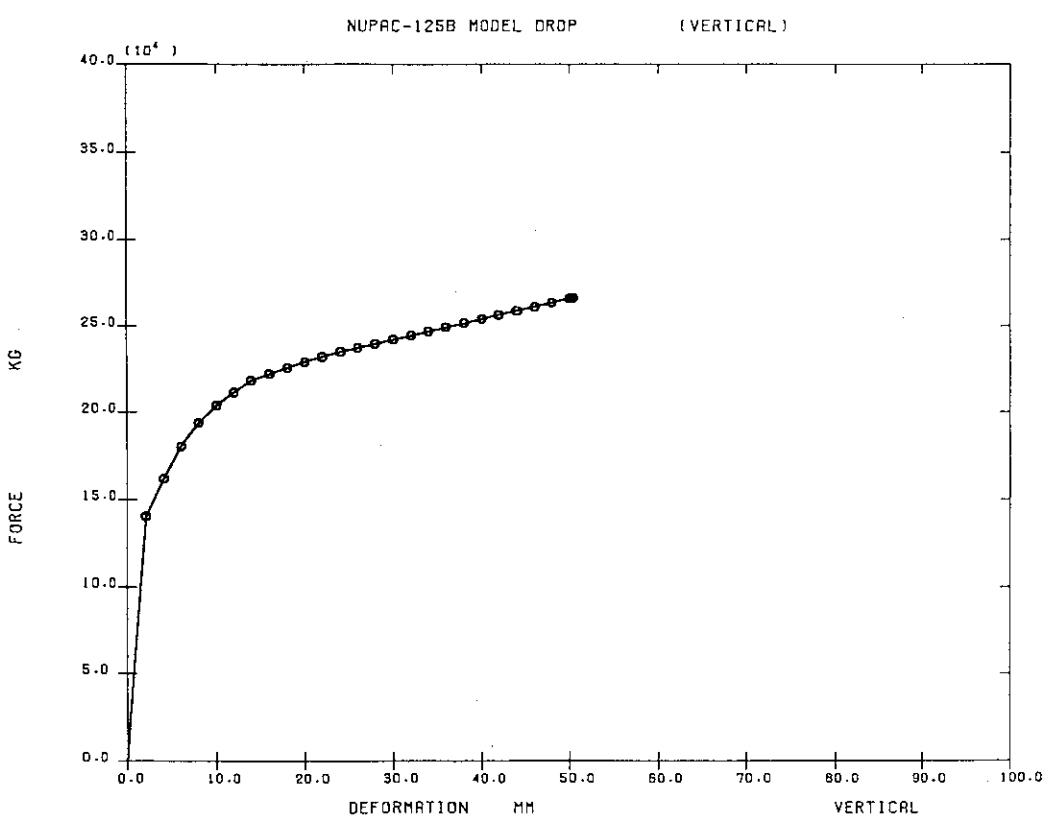


Fig. C.4 Graphical Output of CRUSH2 (4)

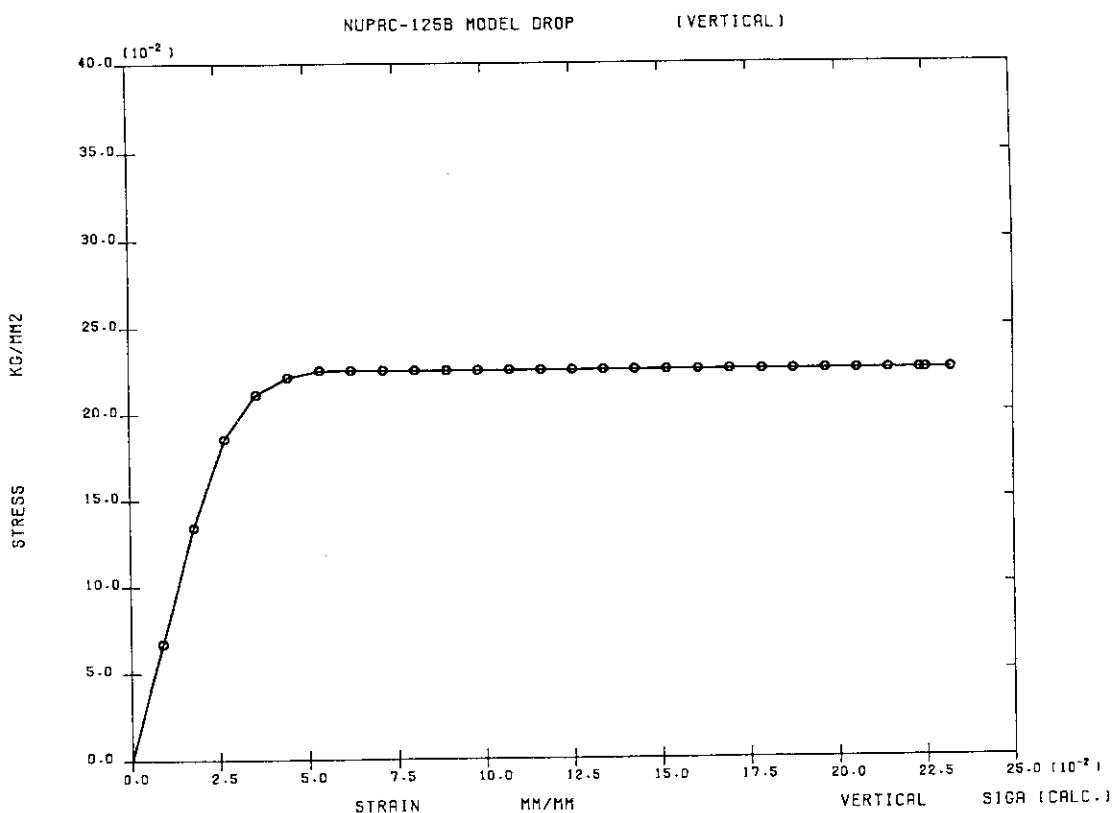


Fig. C.5 Graphical Output of CRUSH2 (5)

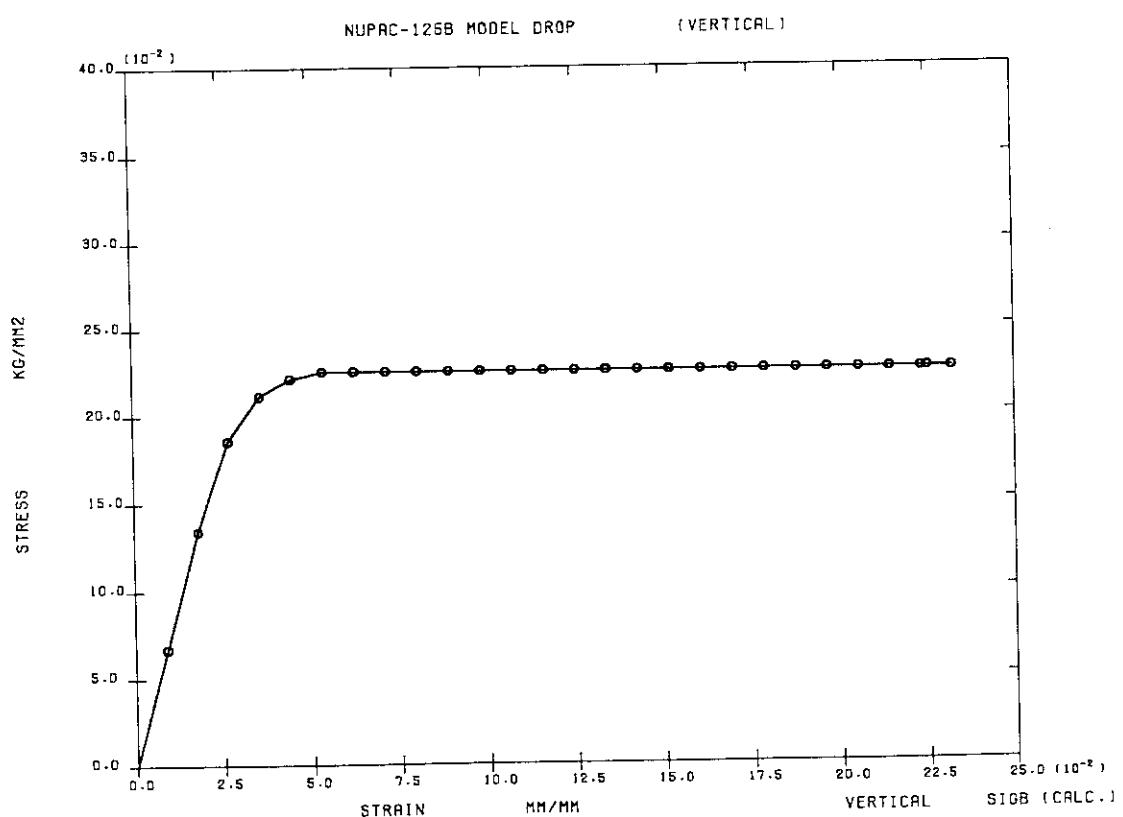


Fig. C.6 Graphical Output of CRUSH2 (6)

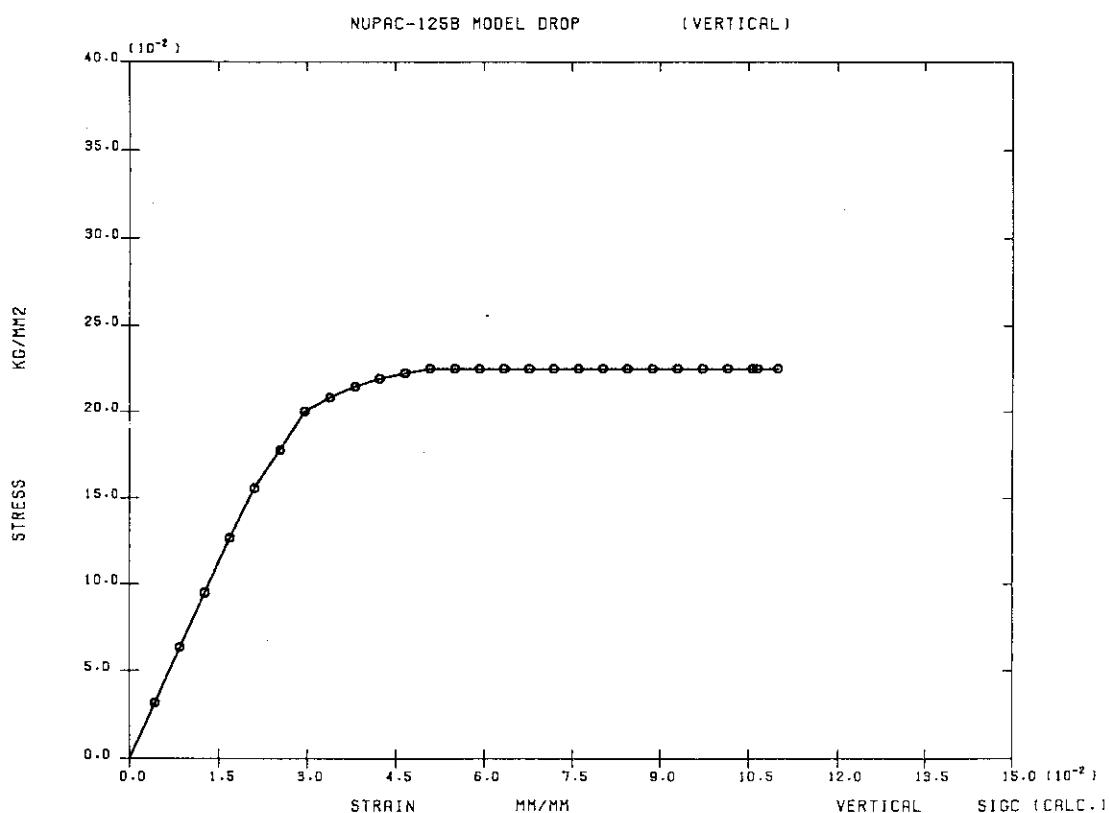


Fig. C.7 Graphical Output of CRUSH2 (7)

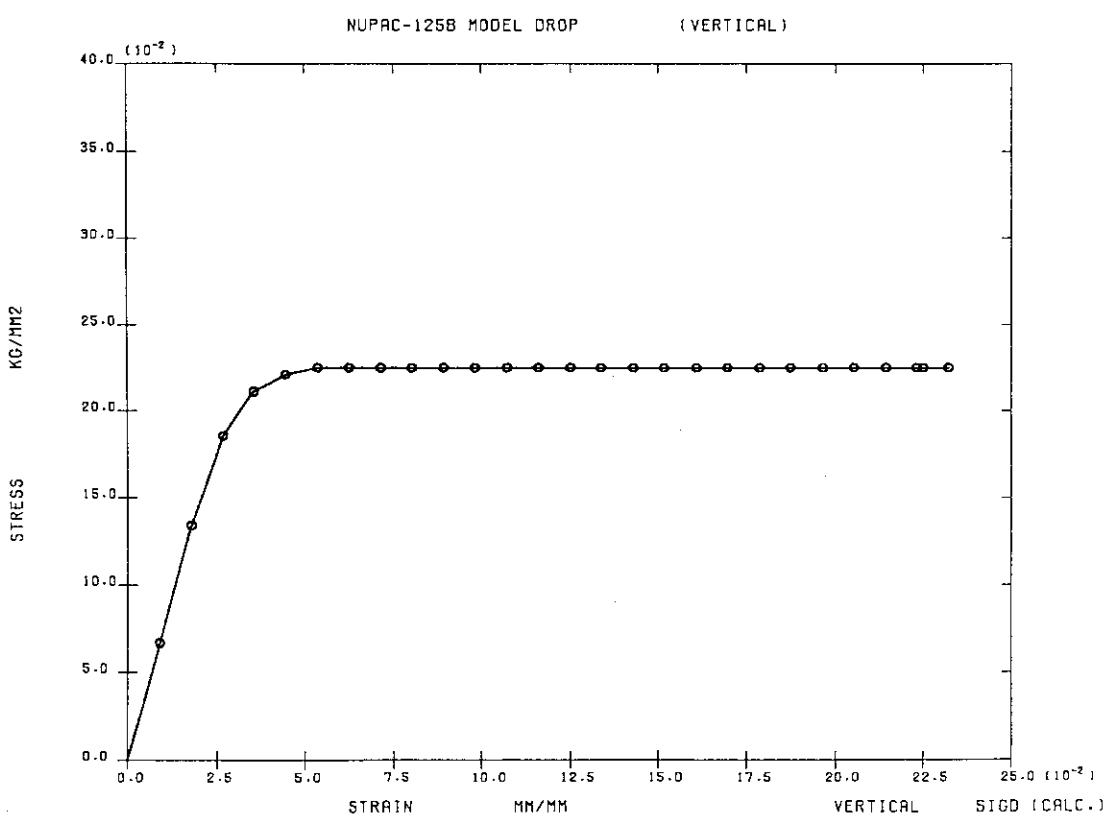


Fig. C.8 Graphical Output of CRUSH2 (8)

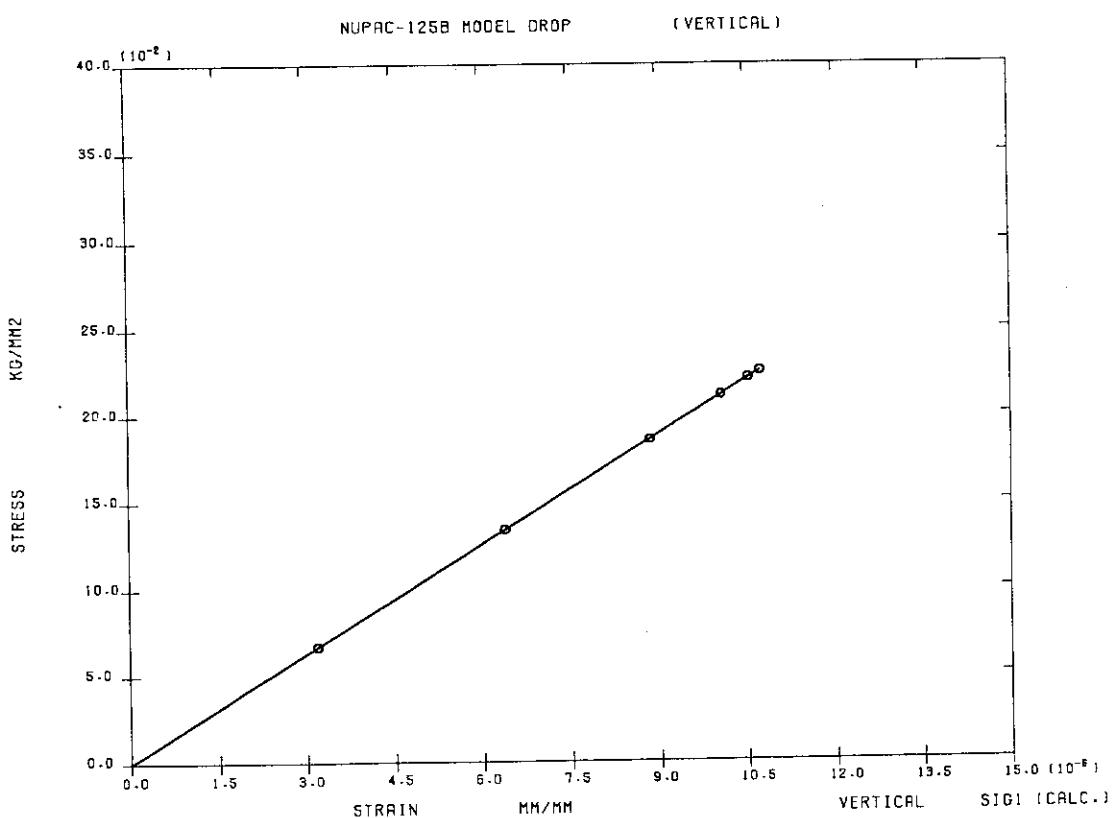


Fig. C.9 Graphical Output of CRUSH2 (9)

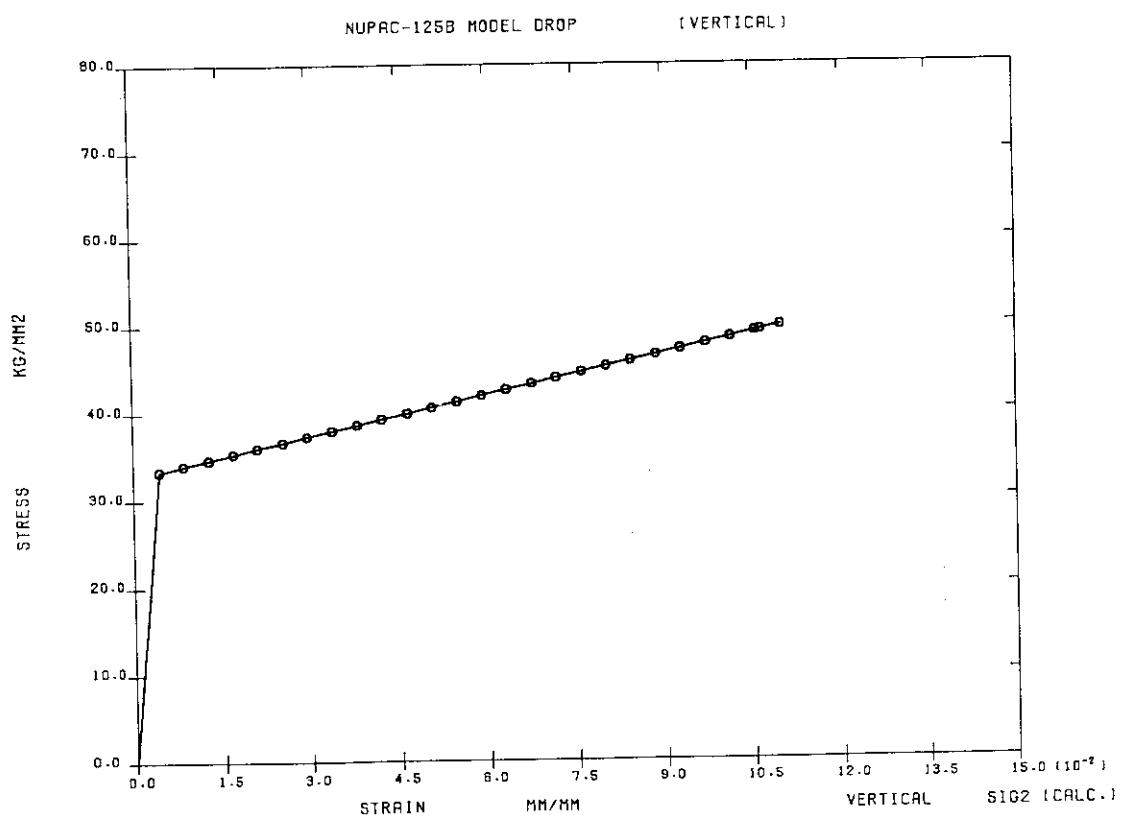


Fig. C.10 Graphical Output of CRUSH2 (10)

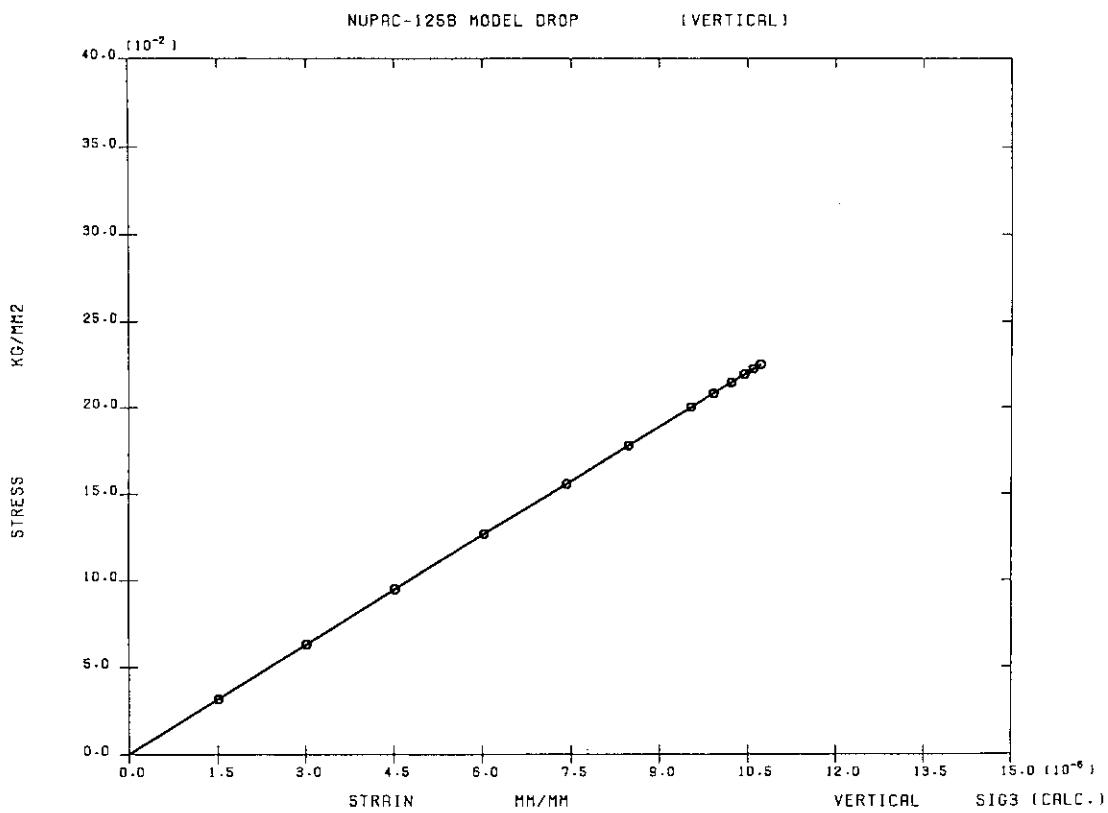


Fig. C.11 Graphical Output of CRUSH2 (11)

#### Appendix D Job Control Data

The job control data for CRUSH2 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  
// OOPTP MSGCLASS=A,MSGLEVEL=(2,0,1),CLASS=B,NOTIFY=JXXXX  
// OOPTP 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 :

CRUSH2.

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:

- (1) Ikushima, T., "Simplified Analysis Computer Program and Their Adequacy for Radioactive Materials Shipping casks", PATRAM' 89 (Washington) DC, USA June 11-16, 1989).
- (2) Ikushima, T., "CRUSH:A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-M 90-004(1990).
- (3) Ikushima, T., "CRUSH1:A Simplified Computer Program for Impact Analysis of Radioactive Material Transport Casks", JAERI-Data/Code 96-025.

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.

14. Any other programming or operating information or restrictions:

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

FACOM M-780 : CALCOMP plitter or the compatible ones.

SUN4 : X-windows.

IBM PC : windows 3.1.

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

### Appendix F (Continued)

```

C ..... C .....
```

```

C CRUSH2 CODE <1994-08-01>
C
C *** /PROGRAMMED BY JAERI
C *** MODIFIED 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 1990-05-15
C *** MODIFIED BY JAERI 1990-09-14
C
C IMPLICIT REAL*8 (A-H,O-Z)
COMMON /MODELS/
RCOR(S), ZCOR(S), WEIGHT(S), HEIGHT,
ANGLE, GRAVITY,
FACT(S), ENERGY,
EPSMAX,
MTOUT, MODEL,
IFLAG, KSOLV,
NITITLE(18),
KSIGOD,
NITITLE(9), MLEN(S),
ELIMIT(9), MLEN(S),
TABLE(100-2,9),
C
COMMON /MATDAT/
COMMON /RESLT/
DEPTH(500),
FORCE(500),
ACCEL(500),
TODISP,
MXSTEP,
ENERGY,
SIG(500,10),
EPS(500-10),
MSUSE(10)
COMMON /SWITCH/
KEOF,
KKTTEST
C
C DIMENSION LABEL(3,23), NDIMM(2)
C DATA LABEL/ 4HVERT, 4HICAL, 4H /
+ 4HHORI, 4HZONT, 4HAL /
C DATA NDIMM/ 4HMM / 4HCM /
DATA NDIM / 4H /
C
C.. GET MODEL
C KSWPLT = 0
KEOF = 0
C
C 100 CONTINUE
CALL CARDIN(MEOF )
IF ( MEOF .NE. 0 ) GO TO 8000
C
C... PLOT OR ANALYSIS
C IF ( KSOLV.EQ.0 ) GO TO 300
C IF ( KSPLT.EQ.0 ) GO TO 8000
KSPLT = KSWPLT + 1
IF ( KSPLT.EQ.1 ) CALL PLTBGN
CALL MPLOT ( LABEL(1,MODEL) )
DO 200 J= 1, 8
IF ( MLEN(J).EQ.0 ) GO TO 200
CALL DPLOT ( LABEL(1,MODEL), NITITLE, TABLE(1,1-J), TABLE(1,2-J),
+ MLEN(J), 3, IFLAG, J )
GO TO 8000
C
C 200 CONTINUE
IF ( MLEN(9).NE.0 )
+ CALL DPLOT ( LABEL(1,MODEL), NITITLE, TABLE(1,1-J), TABLE(1,2-J),
+ MLEN(J), 5, IFLAG, 9 )
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... END OF ANALYSIS
C
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 = 0.001
ENERGO = ENERGO * FMETER
WRITE(MTOUT,9010) NITITLE
9010 FORMAT(1H1.5X 18A4/)
WRITE(MTOUT,9020) (LABEL(1,MODEL),I=1,3), ENERGO
9020 FORMAT(5X,13HMODEL TYPE = ,3A4,10X,14H TOTAL ENERGY = , F12.2,
+ 1X, 6H(KG-M) // )
C
C
C 9030 FORMAT(10X,4HSTEP,12H DEPTH ,4X,11H
+ 3X,12HACCELERATION ) FMETER
C 9035 FORMAT(9035), NDIM
C 9040 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)
9040 FORMAT(10X,14, F12.2,3X,F12.2,3X,F12.2,3X,F12.2)
IF ( MOD( J , 50 ) .NE. 0 ) GO TO 5200
WRITE(MTOUT,9050)
9050 FORMAT(1H1.5X 18A4/
+ 10X,4HSTEP,12H DEPTH ,4X,11H
+ 3X,12HACCELERATION ) FMETER
C 5200 CONTINUE
C
C... PLOT RESULT
C
```

## Appendix F (Continued)

## Appendix F (Continued)

JAERI-Data/Code 97-001

```

C 5500 CONTINUE
  IF( KOUT.EQ.0 )      GO TO 100
  KSWPLT = KSWPLT + 1
  IF( KSWPLT.EQ. 1 )   CALL PLTBGN
C
C   CALL UPLT  ( LABEL(1,MODEL), DEPTH(MXSTEP), ACCEL(MXSTEP) )
C
C   CALL DPLOT ( LABEL(1,MODEL), NTITLE, DEPTH, ENRG, MXSTEP, 1 ,
C                 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
C   DO 5700 J= 1, 8
C     IF( MSUSE(J).EQ.0 )      GO TO 5700
C     CALL DPLOT ( LABEL(1,MODEL), NTITLE, EPSU(1,J), SIGU(1,J),
+                 MXSTEP, 4 , IFLAG, J )
+  CONTINUE
C   IF( MSUSE(9).NE.0 )
+  CALL DPLOT ( LABEL(1,MODEL), NTITLE, EPSU(1,9), SIGU(1,9),
+                 MXSTEP, 5 , IFLAG, 9 )
C
C   IF( KTEST.EQ.0 )
C     GO TO 100
C     WRITE(MTOUT,9220)
C   9220 FORMAT(1H1/10.,20H USED EPS/SIGM TABLE)
C   DO 5800 J= 1, 8
C     IF( MSUSE(J).EQ.0 )      GO TO 5800
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C   9220 FORMAT(//5X,1H MATERIAL TYPE NO.=,14,/)
C   9250 FORMAT(8H EPS =,1E10E12.4)
C   9260 FORMAT(8H SIGNM=,10F12.5)
C   5800 CONTINUE
C     IF( MSUSE(9).EQ.0 )
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C     WRITE(MTOUT,9220)
C   9270 FORMAT(//5X,1H USED FORCE TABLE. )
C   9280 FORMAT(8H DISP =,10F12.3)
C   9290 FORMAT(8H FORCE=,10F12.3)
C   GO TO 100
C
C   8000 CONTINUE
  IF( KSWPLT .GT. 0 ) CALL PLTEND
  STOP
END
C
C   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
C               MODIFIED BY JAERI
C               MODIFIED BY JAERI
C               MODIFIED BY JAERI
C
C   **** IMPLICIT REAL*8 (A-H,O-Z)
C
C   COMMON /MODELS/
C     RCOORD(5), ZCOORD(9), WEIGHT,
C     DISP, FACTK(5), ENERGY,
C     MTINN, MTOU1, GRAVIT,
C     KOUT, NPART, IFLAG,
C     KSIGDD, NTITLE(18),
C     COMMON /MODEL2/
C       CLENG, RADIUS,
C     COMMON /MATERIAL/
C       TABLE(100-2,9),
C     COMMON /COVER/
C       THICK1, THICK2,
C     COMMON /RESULT/
C       DEPTH(500),
C       ENERGY,
C       ENERG(500),
C       EPSU(500-10),
C     COMMON /SIGCHV/
C       MSUSE(10),
C     COMMON /CONV/
C       CONV,
C     COMMON /MODEL1/
C       S2,
C       C2,
C       KKTEST
C
C   DIMENSION LABEL(2-7),
C             NAME(2),
C             KEY(5),
C             NCARD(2),
C             KTYPE(2-9),
C             NVAL(2),
C             NDIMM(2),
C             KOPT(2-3),
C             NCARD(2),
C             KTYPE(2-9),
C             NVAL(2),
C             NDIMM(2),
C             DATA NCARD/ 20*4H /
C             DATA DATA LABEL/ 4H1T1L /,
C             DATA DATA LCOVM/ 4H*** /,
C             DATA DATA MFIN/ 4HF1N /,
C             DATA DATA RAD/ 0.01745329 /,
C             DATA DATA EPSS/ 1.0E-8 /,
C             DATA DATA NDIMM/ 4HMM /,
C             DATA DATA NDIM/ 4H /,
C             DATA DATA KSIGNM/ 4HNONE /
C
C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C...      FUNCTION
        ABS(Z) = DABS(Z)
        SIN(Z) = DSIN(Z)
        COS(Z) = DCOS(Z)
C...      INITIAL VALUES
C
        MTINN = 2
        MTOUT = 6
        MXITER = 20
        DO 80 J= 1, 5
        MSUSE(1) = 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  SIGUR(I,J)= 0.0
C...      CARO IMAGE
C
        LINE = 0
        REWIND MTINN
C
        IF( KEOF.EQ.0 )          GO TO 110
        MEOF = 1
        RETURN
C
        110  CONTINUE
        READ(5,120-END=150)      NCARD
        120  FORMAT(20A4)          NCARD
        IF( NCARD(1).EQ.MFIN )    GO TO 160
C
        LINE = LINE+1
        IF( MOD(LINE,50).NE.1 )   GO TO 130
        WRITE(MTOUT,100) <1,1=1,8>
        100  FORMAT(1H1 //134X,34HC R U S H 2, 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,14,SH-     >20A4)
C
        IF( NCARD(1).EQ.LCOMM )  GO TO 110
        WRITE(MTINN,120)          NCARD
        GO TO 110
C
        C      READ(MTINN,211)      NAME, NTITLE
C...      GET A COMMAND.
C
        150  KEOF = 1
        160  IF( LINE.GT.0 )      MEOF = 1
        RETURN
C
        C. (1)
        170  REWIND MTINN
        READ(MTINN,211)
C
        C. (2)
        READ(MTINN,221)          NAME, WEIGHT, HEIGHT, ANGLE, DISP, CONV,
                                + NDAT, EPMAX
        221  FORMAT(2A4,2X,5F10.0,A4,6X,F10.0)
        LINE = LINE+1
        DO 222 J= 1, 3
        IF( NAME(1).NE.LABEL(1,J) )  GO TO 222
        IF( NAME(2).NE.LABEL(2,J) )  GO TO 224
        222  CONTINUE
        GO TO 999
C
        C. (2)
        MODEL = J
        IF( NDAT.EQ.KEY(1) )      IFLAG = 0
        IF( NDAT.EQ.KEY(2) )      IFLAG = 1
        IF( NDAT.EQ.KEY(4) )      IFLAG = 0
C
        C. (3)
        IF( DISP.EQ.0.0 )        DISP = 1.0
        IF( CONV.EQ.0.0 )        CONV = 1.0D-5
        ENERO = WEIGHT*HEIGHT
        IF( IFLAG.EQ.0 )          GRAVIT = 9800.0
        IF( IFLAG.EQ.1 )          GRAVIT = 980.0
        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. (3)
        READ(MTINN,225)          CLENG, RADIUS, KTYPE,
                                + THICK1, THICK2, THICK3
        LINE = LINE+1
        IF( CLENG.LT.0.0 )        GO TO 992
        IF( RADIUS.LT.0.0 )      GO TO 994
        KSIGDD = 0
        IF( KTYPE.EQ.KTYPE(1,4) )  KSIGDD = 0
        IF( KTYPE.EQ.KSIGN )     KSIGDD = 1
        IF( THICK1.LT.0.0 )      GO TO 990
        IF( THICK2.LT.0.0 )      GO TO 990
        IF( THICK3.LT.0.0 )      GO TO 990
C
        C. (4)
        READ(MTINN,227)          NAME, NAME1, NDATA
        LINE = LINE+1
        DO 228 J= 5, 6
        IF( NAME(1).NE.LABEL(1,J) )  GO TO 228
        IF( NAME(2).NE.LABEL(2,J) )  GO TO 229
        228  CONTINUE
        GO TO 999
C
        C. (2)
        KSOLV = J-5
        KOUT = 0
        IF( NAME1.EQ.KEY(3) )    KOUT = 1

```

## Appendix F (Continued)

## Appendix F (Continued)

```

      KKTEST = 0
      IF( NDATA.EQ.5 ) KKTEST = 1
      C. (5) READ(MTINN,231)
      NAME, (RCOOR(I),I=1,5)
      231 FORMAT(2A4,2X,F10.0/10X,7F10.0)
      LINE = LINE+2
      IF( NAME(1).NE.NAME(2) ) GO TO 999
      IF( NAME(1).NE.NAME(2) ) GO TO 999
      IF( NAME(2).NE.NAME(2) ) GO TO 999

      C. (6) READ(MTINN,231)
      NAME, (ZCOOR(I),I=1,9)
      LINE = LINE+2
      IF( NAME(1).NE.NAME(1,3) ) GO TO 999
      IF( NAME(2).NE.NAME(2,3) ) GO TO 999
      DO 233 J= 2, 4
      IF( RCOOR(I).LT.RCOOR(J-1) ) GO TO 999
      IF( ZCOOR(I).LT.ZCOOR(J-1) ) GO TO 999
      233 CONTINUE
      IF( RCOOR(5).LT.RCOOR(4) ) GO TO 999

      C. (7) READ(MTINN,251)
      NAME, NVAL, FACTK
      251 FORMAT(2A4,2X,2I5,5F10.0)
      LINE = LINE+4
      IF( NAME(1).NE.NAME(1,7) ) GO TO 999
      IF( NAME(2).NE.NAME(2,7) ) GO TO 999

      C. (8) IF( MODEL.NE.1 )
      DO 252 J= 1, 5
      IF( FACTK(I).LT.0.0 ) GO TO 997
      IF( FACTK(I).GT.1.0 ) GO TO 997
      FACTK(I) = 1.0
      252 CONTINUE
      MXFCRK = 5
      GO TO 260

      C. 253 IF( MODEL.NE.2 )
      NPART = NVAL(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
      FACTK(I) = 1.0
      254 CONTINUE
      MXFCRK = 4
      GO TO 260

      C. 255 NPART = NVAL(1)
      KPART = NVAL(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).GT.1.0 ) GO TO 997

      IF( FACTK(I).EQ.0.0 ) FACTK(I) = 1.0
      FACTK5 = 1.0
      MXFCRK = 4
      GO TO 274

      C. (8) CONTINUE
      READ(MTINN,271,END=280) NAME, NN, FACT, ZLIMIT
      271 FORMAT(2A4,2X,15,2F10.0)
      LINE = LINE+1
      DO 272 J= 1, 9
      IF( NAME(I).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 ) FACT = 1.0
      IF( FACT.EQ.0.0 ) ZLIMIT = 10000.0
      IF( ID.EQ.9 ) ZLIMIT = 1.0
      IF( ZLIMIT.EQ.0.0 ) ZLIMIT = 1.0
      ELIMIT(ID) = ZLIMIT
      MLENG(10) = NN

      C. READ(MTINN,275)
      (TABLE(I,1,1D),TABLE(I,1,2,1D), I=1,NN)
      275 FORMAT(8I10.0)
      LINE = LINE+1+(NN-1)/4
      DO 276 I= 1, NN
      TABLE(I,2,1D) = FACT*TABLE(I,2,1D)
      276 CONTINUE
      C. DATA OF TABLE
      DO 277 I= 2, NN
      IF( TABLE(I,1,1D).LE.TABLE(I-1,1,1D) ) GO TO 998
      277 CONTINUE
      DO 278 I= 2, NN
      DS = ABS(TABLE(I,2,1D)-TABLE(I-1,2,1D))
      IF( DS.GT.EPSS ) GO TO 278
      TABLE(I,2,1D) = TABLE(I,2,1D)+TABLE(I,1,2,1D)*EPSS
      278 CONTINUE
      C. 280 IF( ID.NE.4 )
      IF( KSIGD.EQ.0 ) MLENG(ID) = 0
      GO TO 260
      C. 281 IF( MODEL.EQ.3 )
      IF( MLENG(1).EQ.0 ) GO TO 999
      IF( MLENG(2).EQ.0 ) GO TO 999
      IF( MLENG(3).EQ.0 ) GO TO 999
      IF( MLENG(4).GT.0 ) KSIGD = 0
      GO TO 900
      C. 281 CONTINUE
      S = SIN(RAD*ANGLE)
      C = COS(RAD*ANGLE)
      S2 = S*S
      C2 = C*C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

NEWVVV = 0          GO TO 900
IF C MLENG(5) .GT. 0 )          WRITE(6,2)
NEWVVV = 1          WRITE(6,2)
NEWVVV = MLENG(1)          WRITE(6,2)
N1 = MLENG(2)          WRITE(6,2)
N2 = MLENG(2)          WRITE(6,2)
TABLE(N1+1,1,1) = TABLE(N2,1,2)+1.0          WRITE(6,2)
TABLE(N2+1,1,2) = TABLE(N1,1,1)+1.0          WRITE(6,2)
NN = 0              WRITE(6,2)
N1 = 1              WRITE(6,2)
N2 = 1              WRITE(6,2)

C.** 285 IF( N1.GT.MLENG(1) .AND. N2.GT.MLENG(2) )          GO TO 290
NN = NN+1          WRITE(6,2)
EPS1 = TABLE(N1,1,1)          WRITE(6,2)
SIG1 = TABLE(N1,2,1)          WRITE(6,2)
EPS2 = TABLE(N2,1,1)          WRITE(6,2)
SIG2 = TABLE(N2,2,2)          WRITE(6,2)
IF( ABS(EPS1-EPS2).GT.EPSS )          WRITE(6,2)
TABLE(NN,1,5) = EPS1          WRITE(6,2)
TABLE(NN,2,5) = EPS2          WRITE(6,2)
N1 = N1+1          WRITE(6,2)
N2 = N2+1          WRITE(6,2)
GO TO 285          WRITE(6,2)
287 IF( EPP2.GT.EPS1 )          GO TO 288
K1 = N1          K1 = MLENG(1)
IF( N1.GT.MLENG(1) )          K1 = MLENG(1)
K0 = K1-1          K1 = MLENG(1)
DX = TABLE(K1,1,1)-TABLE(K0,1,1)          K1 = MLENG(1)
DY = TABLE(K1,2,1)-TABLE(K0,2,1)          K1 = MLENG(1)
SIG1 = TABLE(K0,2,1) + DY/DXX*(EPS1-TABLE(K0,1,1))          K1 = MLENG(1)
TABLE(NN,1,5) = EPS1          K1 = MLENG(1)
TABLE(NN,2,5) = EPS2          K1 = MLENG(1)
N1 = N1+1          K1 = MLENG(1)
N2 = N2+1          K1 = MLENG(1)
GO TO 285          K1 = MLENG(1)
288 K1 = N2          K1 = MLENG(2)
IF( N2.GT.MLENG(2) )          K1 = MLENG(2)
K0 = K1-1          K1 = MLENG(2)
DX = TABLE(K1,1,2)-TABLE(K0,1,2)          K1 = MLENG(2)
DY = TABLE(K1,2,2)-TABLE(K0,2,2)          K1 = MLENG(2)
SIG2 = TABLE(K0,2,2) + DY/DXX*(EPS1-TABLE(K0,1,2))          K1 = MLENG(2)
TABLE(NN,1,5) = EPS1          K1 = MLENG(2)
TABLE(NN,2,5) = EPS2          K1 = MLENG(2)
N1 = N1+1          K1 = MLENG(2)
N2 = N2+1          K1 = MLENG(2)
GO TO 285          K1 = MLENG(2)

C.** 290 MLENG(5) = NN          K1 = MLENG(2)
C... END OF INPUT CARD.
C.** 900 CONTINUE          K1 = MLENG(2)
WRITE(MTOUT,910)          K1 = MLENG(2)
910 FORMAT(1//1X,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) / )
          +          CCC WRITE(MTOUT,920)          (FACTK(J),J=1,MFACTK)
920 FORMAT(1//1X,22H-----, K-FACTOR =>SF10.4)
C... 998 WRITE(MTOUT,9981)          KTYPE(1,J), I
C... 9981 FORMAT(4OH *** ERROR STRAIN TABLE.          LABEL=A4,5X,6HCOLUMN=,
          + I4)          CALL EXIT
          RETURN
C... 999 WRITE(MTOUT,9991)          LINE
          +          9991 FORMAT(4OH *** ERROR CRAD FOUND.          CARD NO. =15)
          +          CALL EXIT
          RETURN
END

```

## Appendix F (Continued)

## Appendix F (Continued)

JAERI - Data/Code 97-001

```

SUBROUTINE VERT
C****/PURPOSE/
C*** CALCULATION ENERGY OF VERTICAL SHPAE
C*** PROGRAMMED BY JAERI
C*** MODIFIED BY JAERI
C*** MODIFIED BY JAERI
C*** MODIFIED BY JAERI
C*** MODIFIED BY JAERI

C IMPLICIT REAL*8 (A-H-O-Z)
COMMON /MODELS/ RCOORD(5), ZCOORD(9), WEIGHT,
DISP, FACTK(5), ENRG0,
MTINN, MTOUT,
NPART, KPART,
COMMON /MATDATA/ TABLE(100,2-9),
COMMON /COVER/ THICK1,
COMMON /SWITCH/ KEOF,
COMMON /RESULT/ DEPTH(500),
ENRG(500),
EPSU(500,10),
MSUSE(10),
+
DIMENSION P(6),
DL(4),
IDB(2),
DATA PAI/ 3.14159265 /
C... INITIAL STATUS
C MXSTEP = 0
TODISP = 0.0
MINI = 0
DISPX = DISP
C A(1) = PAI*RCOORD(1)**2
A(2) = PAI*(RCODR(2)*ZCOORD(1)**2-RCOORD(1)*ZCOORD(2)**2)
A(3) = PAI*(RCODR(3)*ZCOORD(2)**2-RCOORD(2)*ZCOORD(3)**2)
A(4) = PAI*(RCODR(4)*ZCOORD(3)**2-RCOORD(3)*ZCOORD(4)**2)
A(5) = PAI*(RCODR(5)*ZCOORD(4)**2-RCOORD(4)*ZCOORD(5)**2)
A(6) = PAI*RCOORD(5)*THICK2**2.0D0
IF (KKTEST.NE.0 ) WRITE(6,1)
1 FORMAT(GH AREA=,6F12.2)
C DL(1) = ZCOORD(3)
DL(2) = ZCOORD(4)-ZCOORD(3)
ID(1) = 2
ID(2) = 3
KS3 = 2
C DLX(1) = THICK1
DLX(2) = ZCOORD(2)
IDA(1) = 6
IDA(2) = 4
IDB(1) = 6
IDB(2) = 1
IDC(1) = 6
GO TO 50
IF (MLENG(6).EQ.0 )
KS3 = KS3+1
DL(KS3)= THICK1
ID(KS3)= 8
GO TO 100
IF (MLENG(8).EQ.0 )
KS3 = KS3+1
DL(KS3)= THICK3
ID(KS3)= 8
GO TO 1000
IF (MLENG(9).GT.500 )
TODISP = TODISP + DISPX
IF (MLENG(6).GT.0 )
GO TO 300
C C... GET CURRENT FORCE
C 100 CONTINUE
MXSTEP = MXSTEP+1
IF (MXSTEP.GT.500 )
GO TO 1000
TODISP = TODISP + DISPX
IF (MLENG(6).GT.0 )
GO TO 300
C C. (R1-R2)
EPS = TODISP/ZCOORD(2)
CALL GETSIG (EPS, 4, SIGM )
P(1) = A(1)*SIGM
C C. (R1-R2)
EPS = TODISP/ZCOORD(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
GO TO 500
C C. (O-R1)
300 CONTINUE
CALL MULTI (TODISP, DLX, IDA, 2, SIGM , IERR )
IF (IERR.NE.0 )
GO TO 3100
P(1) = A(1)*SIGM
C C. (R1-R2)
CALL MULTI (TODISP, DLX, IDB, 2, SIGM , IERR )
IF (IERR.NE.0 )
GO TO 3200
P(2) = A(2)*SIGM
C C. (R2-R3)
500 CONTINUE
CALL MULTI (TODISP, DL, ID, KS3, SIGM , IERR )
IF (IERR.NE.0 )
GO TO 3300
P(3) = A(3)*SIGM
P(4) = A(4)*SIGM
C C. (R4-R5)

```

## Appendix F (Continued)

## Appendix F (Continued)

```

P(5) = A(5)*SIGN
C. (COVER.2)
EPS = TODISP/ZCOORD(4)
CALL GETSIG (EPS, 7, SIGN)
P(6) = A(6)*SIGN
C...
SUM OF EACH FORCES
C
SUM = 0.0
DO 700 J=1, 5
700 SUM = SUM + P(JJ)*FACTK(J)
CALL GETSIG (TODISP, 9, PO)
SUM = SUM + P(6)+PO
C
IFC KKTTEST NE.0
3 FORMAT(1X,15,4X,7F12.3)
C
ENERGY = SUM*DISPX
DEPTH(MXSTEP) = TODISP
FORCE(MXSTEP) = SUM
ACCEL(MXSTEP) = SUM/WEIGHT
ENERG(MXSTEP) = ENERGY
IFC MXSTEP.NE.1 ) ENERGY = ENERG(MXSTEP-1)+ENERGY
C
IFC ENERGY(MXSTEP).LT.ENERGO GO TO 100
IFC MINI.NE.0 ) GO TO 800
MINI = 1
TODISP = TODISP - DISPX
MXSTEP = MXSTEP-1
DISPX = 0.1*DISP
GO TO 100
C
INTERPOLATION OF LAST STEP
C
800 CONTINUE
CALL FITSTP
C...
END OF VERTICAL
C
1000 CONTINUE
IFC 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 =,15,
+ 5X,7H DEPTH =,F10.4,5X,11HNO. OF MAT=,I3,5X,5I3)
CALL EXIT
RETURN
C
3100 WRITE(MTOUT,3110) MXSTEP, TODISP, IDA
3110 FORMAT(// 29H *** MULTI-MATERIAL ERROR. 5X,7H STEP =,15,
+ 5X,7H DEPTH =,F10.4,5X,11HNO. OF MAT=,I3,5I3)
CALL EXIT
RETURN
3200 WRITE(MTOUT,3110) MXSTEP, TODISP, IDB

```

```

CALL EXIT
RETURN
3300 WRITE(MTOUT,3110) MXSTEP, TODISP, IDC
CALL EXIT
RETURN
END

SUBROUTINE HORIZ
C*** /PURPOSE/
C*** CALCULATION ENERGY OF HORIZONTAL SHAPE
C
PROGRAMMED BY JAERI
MODIFIED BY JAERI
MODIFIED BY JAERI
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /MODELS/
RCOOR(S), ZCOOR(9), WEIGHT,
ANGLE, EPMAX,
FACT(5), ENERGO,
MTOUT, MTLINN,
DISP, IFLAG,
KSOLV,
KSIGDD, NTITLE(18)
C
C*** COMMON /PART/
NPART, KPART,
COMMON /MATERIAL/
TABLE(100,2,9),
ELIMIT(9), MLENG(9)
THICK2,
THICK3
COMMON /COVER/
THICK1,
KTEST
COMMON /SWITCH/
KEOF,
COMMON /RESULT/
DEPTH(500),
FORCE(500),
ENERGY,
EPSU(500,1:0),
SIGU(500,10),
MSUSE(10)
C
DIMENSION P(6),
E(6),
DZ(6),
R(5),
R0(3),
ID(4),
MESH1(3),
MESH2(2),
XLENG(6,300,6)
C
FUNCTION
SQRT(Z) = DSQRT(Z)
C...
INITIAL STATUS
C
MXSTEP = 0
TODISP = 0.0
MINI = 0
DISPX = DISP
C...
GET (Z) LENGTH
DZ(1) = ZCOORD(1)
DZ(2) = ZCOORD(2)-ZCOORD(1)
DZ(3) = ZCOORD(3)-ZCOORD(2)
DZ(4) = ZCOORD(4)-ZCOORD(3)
DZ(5) = 2.0*THICK1
DZ(6) = 2.0*THICK3
IF ( ZCOORD(5).GT.ZCOORD(4) ) GO TO 30
DO 20 I= 1, 4
DZ(I) = 2.0*DZ(I)
20 DO 10 50
GO TO 50

```

## Appendix F (Continued)

## Appendix F (Continued)

```

30 DZ(1) = DZ(1) + ZCOORD(9)-ZCOORD(8)
DZ(2) = DZ(2) + ZCOORD(8)-ZCOORD(7)
DZ(3) = DZ(3) + ZCOORD(7)-ZCOORD(6)
DZ(4) = DZ(4) + ZCOORD(6)-ZCOORD(5)

C.. GET (R) LENGTH
50 R(1) = RCOORD(1)
R(2) = RCOORD(2)-RCOOR(1)
R(3) = RCOORD(3)-RCOOR(2)
R(4) = RCOORD(4)-RCOOR(3)
R(5) = RCOORD(5)-RCOOR(3)
NPART = NPART
DPART = RCOORD(4)/DPART
DX = RCOORD(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 C KTEST NE 0 ) WRITE(6,1)
1 FORMAT(6H MESH=,3I5,5X,215,5X,6F10.3)

C.. (C-C) MESH DATA
MLLOOP = 0
R(1) = 0.0
R(2) = RCOORD(1)
R(3) = RCOORD(2)

DO 150 J= 1, 3
LOOP = MESH1(J)
DPART = LOOP
DX = R(J)/DPART
XX = R(J) + 0.5*DX
DO 100 I= 1, LOOP
XR5 = SQRT( RCOORD(S)**2-XX**2 )
H = RCOORD(S)-XR5
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP,1) = DX
XLENG(2,MLLOOP,1) = H
XLENG(3,MLLOOP,1) = DX
XLENG(4,MLLOOP,1) = H
XLENG(5,MLLOOP,1) = DX
XLENG(6,MLLOOP,1) = H

GO TO ( 91, 92, 93 ), J
91 XR1 = SQRT( RCOORD(1)**2-XX**2 )
XR2 = SQRT( RCOORD(2)**2-XX**2 )
IF( MLENG(4).EQ.0 ) GO TO 99
XLENG(3,MLLOOP,1) = 2.0*(XRS-XR2)
XLENG(4,MLLOOP,1) = 2.0*(XRS-XR1)
XLENG(5,MLLOOP,1) = 2.0*(XRS-XR1)
XLENG(6,MLLOOP,1) = 2.0*(XRS-XR1)
XLENG(5,MLLOOP,2) = THICK2
XLENG(6,MLLOOP,2) = 2.0*XRS-XR2
XLENG(4,MLLOOP,2) = 2.0*XRS-XR1
XLENG(3,MLLOOP,2) = 2.0*XRS-XR1
XLENG(2,MLLOOP,2) = DX
XLENG(1,MLLOOP,2) = H

DO 100 XX = XX + DX
150 CONTINUE
C.. (D-D) MESH
MLLOOP = 0
R(2) = RCOORD(3)

C.. DO 250 J= 1, 2
LOOP = MESH2(J)
DPART = LOOP
DX = R(J+3)/DPART
XX = R(J) + 0.5*DX
DO 200 I= 1, LOOP
XR5 = SQRT( RCOORD(S)**2-XX**2 )
H = RCOORD(S)-XR5

```

## Appendix F (Continued)

## Appendix F (Continued)

```

MLoop = MLoop+1
XLEN(1,MLoop,-3) = DX
XLEN(2,MLoop,-3) = H
XLEN(3,MLoop,-4) = DX
XLEN(4,MLoop,-4) = H
XLEN(5,MLoop,-6) = DX
XLEN(6,MLoop,-6) = H
IFC J,NE,1)          GO TO 190
X3 = SQRT(RCQR(3)**2-XX**2)
XLEN(3,MLoop,-3) = XRS-XR3
XLEN(4,MLoop,-3) = THICK2
XLEN(5,MLoop,-3) = 0.0
XLEN(6,MLoop,-3) = 0.0
XLEN(3,MLoop,-4) = XRS-XR3
XLEN(4,MLoop,-4) = THICK2
XLEN(5,MLoop,-4) = 0.0
XLEN(6,MLoop,-4) = 0.0
XLEN(3,MLoop,-6) = XRS-XR3
XLEN(4,MLoop,-6) = 0.0
XLEN(5,MLoop,-6) = 0.0
XLEN(6,MLoop,-6) = 0.0
GO TO 200
190 XLEN(3,MLoop,-3) = 2.0*XRS
XLEN(4,MLoop,-3) = THICK2
XLEN(5,MLoop,-3) = 0.0
XLEN(6,MLoop,-3) = 0.0
XLEN(3,MLoop,-4) = 2.0*XRS
XLEN(4,MLoop,-4) = THICK2
XLEN(5,MLoop,-4) = 0.0
XLEN(6,MLoop,-4) = 0.0
XLEN(3,MLoop,-6) = 2.0*XRS
XLEN(4,MLoop,-6) = 0.0
XLEN(5,MLoop,-6) = 0.0
XLEN(6,MLoop,-6) = 0.0
GO TO 200
200 XX = XX + DX
250 CONTINUE
C ...
C ... GET CURRENT FORCE
C ...
1000 CONTINUE
MXSTEP = MXSTEP+1
IFC MXSTEP .GT. 500 )      GO TO 2000
TODISP = TODISP + DISPX
C
C .. Z = 0.0 - 21
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) = 3
KS3 = KS3-1
IFC MLEN(7) .EQ. 0 )      GO TO 572
DO 573 I= 1, LOOP
MLoop = MLoop+1
U = TODISP-XLEN(2,MLoop,1)
IFC IERR.NE.0 )           GO TO 573
CALL MULTI( U, XLEN(3,MLoop,1), ID, KS3, SIGM, IERR )
UO = DISPX
IFC U.LE.0.0 )             CALL MULTI( U, XLEN(1,MLoop,1), ID, KS3, SIGM, IERR )
UO = DISPX
IFC U.LT.DISPX )           P(1) = P(1) + SIGM*XLEN(1,MLoop,1)
E(1) = E(1) + SIGM*XLEN(1,MLoop,1)*UO
573 CONTINUE
GO TO 600
581 ID(1) = 2
ID(2) = 1
ID(3) = 3
KS3 = KS3-1
IFC MLEN(7) .EQ. 0 )      GO TO 582
DO 582 I= 1, LOOP
MLoop = MLoop+1
U = TODISP-XLEN(2,MLoop,1)
IFC U.LE.0.0 )             CALL MULTI( U, XLEN(3,MLoop,1), ID, KS3, SIGM, IERR )
UO = DISPX
IFC U.LT.DISPX )           P(1) = P(1) + SIGM*XLEN(1,MLoop,1)
E(1) = E(1) + SIGM*XLEN(1,MLoop,1)*UO
582 CONTINUE
GO TO 600
591 IF C MLEN(7) .GT. 0 )      GO TO 593
DO 592 I= 1, LOOP
MLoop = MLoop+1
U = TODISP-XLEN(2,MLoop,1)
IFC U.LE.0.0 )             CALL GETSIG( EPS, 2, SIGM )
UO = DISPX
IFC U.LT.DISPX )           P(1) = P(1) + SIGM*XLEN(1,MLoop,1)
E(1) = E(1) + SIGM*XLEN(1,MLoop,1)*UO
592 CONTINUE
GO TO 600
593 ID(1) = 2
ID(2) = 3
KS3 = KS3-1
DO 594 I= 1, LOOP
MLoop = MLoop+1
U = TODISP-XLEN(2,MLoop,1)
IFC U.LE.0.0 )             CALL MULTI( U, XLEN(3,MLoop,1), ID, KS3, SIGM, IERR )
IFC IERR.NE.0 )           GO TO 3000

```

## Appendix F (Continued)

## Appendix F (Continued)

```

U0 = DISPX
IF( U.LE.0.0 ) U0 = 0.0
P(1) = P(1) + SIGM*XLENG(1,MLLOOP,1)
E(1) = E(1) + SIGM*XLENG(1,MLLOOP,1)*U0
594 CONTINUE
C 600 CONTINUE
C
C.. Z = 21- 72
ID(1) = 2
ID(2) = 0.0
ID(3) = 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
ID(4) = 7
KS3 = 4
IF( MLEN(4).GT.0 ) GO TO 672
ID(3) = 7
KS3 = 3
IF( MLEN(7).EQ.0 ) KS3 = KS3-1
DO 673 I= 1, L0OP
  MLloop = MLloop+1
  U = TODISP-XLENG(2,MLLOOP,2)
  IF( U.LE.0.0 ) GO TO 673
  CALL MULTI( U, XLENG(3,MLLOOP,2), ID, KS3, SIGM, IERR )
  IF( IERR.NE.0 ) GO TO 3000
  U0 = DISPX
  IF( U.LT.DISPX ) U0 = 0
  P(2) = P(2) + SIGM*XLENG(1,MLLOOP,2)
  E(2) = E(2) + SIGM*XLENG(1,MLLOOP,2)*U0
  673 CONTINUE
  GO TO 700
  681 ID(1) = 2
  ID(2) = 1
  ID(3) = 7
  KS3 = 3
  IF( MLEN(7).EQ.0 ) KS3 = 2
  DO 682 I= 1, L0OP
    MLloop = MLloop+1
    U = TODISP-XLENG(2,MLLOOP,2)
    IF( U.LE.0.0 ) GO TO 682
    CALL MULTI( U, XLENG(3,MLLOOP,2), ID, KS3, SIGM, IERR )
    IF( IERR.NE.0 ) GO TO 3000
    U0 = DISPX
    IF( U.LT.DISPX ) U0 = 0
    P(2) = P(2) + SIGM*XLENG(1,MLLOOP,2)
    E(2) = E(2) + SIGM*XLENG(1,MLLOOP,2)*U0
  682 CONTINUE
  GO TO 700
  691 IF( MLEN(7).GT.0 ) GO TO 693
  DO 692 I= 1, L0OP
    MLloop = MLloop+1
    U = TODISP-XLENG(2,MLLOOP,2)
    IF( U.LE.0.0 ) GO TO 692
    EPS = U/XLENG(3,MLLOOP,2)
    CALL GETSIG( EPS, 2, SIGM )
    U0 = DISPX
    IF( U.LT.DISPX ) U0 = 0
    P(2) = P(2) + SIGM*XLENG(1,MLLOOP,2)
    E(2) = E(2) + SIGM*XLENG(1,MLLOOP,2)*U0
  692 CONTINUE
  GO TO 700
  693 ID(1) = 2
  ID(2) = 7
  KS3 = 2
  DO 694 I= 1, L0OP
    MLloop = MLloop+1
    U = TODISP-XLENG(2,MLLOOP,2)
    IF( U.LE.0.0 ) GO TO 694
    CALL MULTI( U, XLENG(3,MLLOOP,2), ID, KS3, SIGM, IERR )
    IF( IERR.NE.0 ) GO TO 3000
    U0 = DISPX
    IF( U.LT.DISPX ) U0 = 0
    P(2) = P(2) + SIGM*XLENG(1,MLLOOP,2)
    E(2) = E(2) + SIGM*XLENG(1,MLLOOP,2)*U0
  694 CONTINUE
  GO TO 700
  C.. Z = 22- 23
  P(3) = 0.0
  E(3) = 0.0
  MLloop = 0
  DO 700 J= 1, 2
    LOOP = MESH2(J)
    IF( MLEN(7).GT.0 ) GO TO 703
    DO 702 I= 1, L0OP
      MLloop = MLloop+1
      U = TODISP-XLENG(2,MLLOOP,3)
      IF( U.LE.0.0 ) EPS = U/XLENG(3,MLLOOP,3)
      CALL GETSIG( EPS, 2, SIGM )
      U0 = DISPX
      IF( U.LT.DISPX ) U0 = 0
      P(3) = P(3) + SIGM*XLENG(1,MLLOOP,3)
      E(3) = E(3) + SIGM*XLENG(1,MLLOOP,3)*U0
    702 CONTINUE
    DO 704 I= 1, L0OP
      MLloop = MLloop+1
      U = TODISP-XLENG(2,MLLOOP,3)
      IF( U.LE.0.0 ) CALL MULTI( U, XLENG(3,MLLOOP,3), ID, KS3, SIGM, IERR )
      IF( IERR.NE.0 ) GO TO 3000
      U0 = DISPX
    704 CONTINUE
    GO TO 700
  700 CONTINUE

```

## Appendix F (Continued)

### (Continued)

```

IF( U.LT.DISPX )          U0
P(3) + SIGM*XLENG(1,MLLOOP,3) = U
E(3) = E(3) + SIGM*XLENG(1,MLLOOP,3)*U0
784 CONTINUE
800 CONTINUE
C
C.. I = 23- 24
P(4) = 0.0
E(4) = 0.0
MLoop = 0
C
DO 900 J= 1, 2
LOOP = MESH2(J)
IF( MLENG(7).GT.0 )      GO TO 883
DO 882 I= 1, LOOP
MLoop = MLLoop+1
U = TODISP-XLENG(2,MLLoop,4)
IF( U.LE.0.0 )            GO TO 882
EPS = U/XLENG(3,MLLoop,4)
CALL GETSIG( EPS, 3, SIGM )
U0 = DISPX
IF( U.LT.DISPX )          U0
P(4) = P(4) + SIGM*XLENG(1,MLLoop,4)
E(4) = E(4) + SIGM*XLENG(1,MLLoop,4)*U0
882 CONTINUE
GO TO 900
883 ID(1) = 3
ID(2) = 7
KS3 = 2
DO 884 I= 1, LDDP
MLoop = MLLoop+1
U = TODISP-XLENG(2,MLLoop,4)
IF( U.LE.0.0 )            GO TO 884
CALL MULTI( U, XLENG(3,MLLoop,4), ID, KS3, SIGM, IERR )
IF( IERR.NE.0 )           GO TO 3000
U0 = DISPX
IF( U.LT.DISPX )          U0
P(4) = P(4) + SIGM*XLENG(1,MLLoop,4)
E(4) = E(4) + SIGM*XLENG(1,MLLoop,4)*U0
884 CONTINUE
900 CONTINUE
C
C.. THICK1
P(5) = 0.0
E(5) = 0.0
MLoop = 0
C
DO 950 J= 1, 3
LOOP = MECH1(J)
DO 940 I= 1, LOOP
MLoop = MLLoop+1
U = TODISP-XLENG(2,MLLoop,5)
IF( U.LE.0.0 )            GO TO 940
EPS = U/XLENG(3,MLLoop,5)
CALL GETSIG( EPS, 6, SIGM )
U0 = DISPX
C

```

## Appendix F (Continued)

### Appendix F (Continued)

```

C 1800 CONTINUE
C CALL FITSTP
C... END OF HORIZONTAL
C
2000 CONTINUE
IF( MXSTEP.GT.500 ) MXSTEP = 500
RETURN

C*** ERROR
3000 CONTINUE
WRITE(*,10)
10 FORMAT(1X, 29H *** MULTI-MATERIAL ERROR. ,5X,7H STEP =,15,
      + 5X,7HDEPTH =,F10.4,5X,11HNO. OF MAT=,13,5X,5I3)
CALL EXIT
RETURN
END

SUBROUTINE CONER
C*** /PURPOSE/
C*** CALCULATION ENERGY OF CORNER SHAPE
C
C PROGRAMMED BY JAERI
1988-02-09
C MODIFIED BY JAERI
1988-09-19
C MODIFIED BY JAERI
1988-01-05
C MODIFIED BY JAERI
1990-05-15
C MODIFIED BY JAERI
1990-09-14
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
COMMON /MODELS/
      RCOORD(5), ZCOORD(9), WEIGHT,
      FACT(5), DISP,
      MTIN, MTOU,
      NPART, KPART,
      COMMON /MATDATA/
      TABLE(100-2,9),
      COMMON /COVER/
      THICK1, THICK2,
      COMMON /MODEL/
      S, C,
      S2, C2
      FORCE(500),
      ACCEL(500),
      DEPTH(500),
      ENERGY,
      SIGU(500,10),
      MSURE(10),
      KTEST
      COMMON /RESLT/
      COMMON /EPSMX /
      COMMON /SIMPDD/
      RSS, RSZ
      DIMENSION ST(2,18),
      COVER(3),
      SLICE(7,400),
      MESH(4,400),
      DIMENSION DL(3),
      ID(3)
      C... FUNCTION
      DS MESH SIZE
      S1 = ST(1,1)
      S2 = ST(1,4)
      S3 = ST(1,12)
      DPART
      DX = (S3-S1)/DPART
      M1 = (S2-S1)/DX
      IF( ANGLE.LE.45.0 )
      1 FORMAT(6H ST =,10F12.4)
      C... INITIAL STATUS
      C
      MXSTEP = 0
      TDISP = 0.0
      MINI = 0
      DISPX = DISP
      C... GET (S,I) COORD
      C
      ST(1, 1) = -RCODR(5)*C
      ST(2, 1) = RCOORD(5)*S
      ST(1, 2) = -RCODR(1)*C
      ST(2, 2) = RCOORD(1)*S
      ST(1, 3) = RCOORD(1)*C
      ST(2, 3) = -RCODR(1)*S
      ST(1, 4) = RCOORD(5)*C
      ST(2, 4) = -RCODR(5)*S
      ST(1, 5) = -RCODR(3)*C+ZCOORD(2)*S
      ST(2, 5) = RCOORD(3)*S+ZCOORD(2)*C
      ST(1, 6) = -RCODR(1)*C+ZCOORD(2)*S
      ST(2, 6) = RCOORD(1)*S+ZCOORD(2)*C
      ST(1, 7) = RCOORD(1)*C+ZCOORD(2)*S
      ST(2, 7) = -RCODR(1)*S+ZCOORD(2)*C
      ST(1, 8) = RCOORD(5)*C+ZCOORD(2)*S
      ST(2, 8) = -RCODR(3)*S+ZCOORD(2)*C
      ST(1, 9) = -RCODR(5)*C+ZCOORD(4)*S
      ST(2, 9) = RCOORD(5)*S+ZCOORD(4)*C
      ST(1, 10) = -RCODR(3)*C+ZCOORD(4)*S
      ST(2, 10) = RCOORD(3)*S+ZCOORD(4)*C
      ST(1, 11) = -RCODR(3)*C+ZCOORD(4)*S
      ST(2, 11) = -RCODR(3)*S+ZCOORD(4)*C
      ST(1, 12) = RCOORD(5)*C+ZCOORD(4)*S
      ST(2, 12) = -RCODR(5)*S+ZCOORD(4)*C
      ST(1, 13) = -RCODR(1)*C+ZCOORD(4)*S
      ST(2, 13) = RCOORD(1)*S+ZCOORD(4)*C
      ST(1, 14) = RCOORD(1)*C+ZCOORD(4)*S
      ST(2, 14) = -RCODR(1)*S+ZCOORD(4)*C
      ST(1, 15) = ZCOORD(2)*S
      ST(2, 15) = RCOORD(4)*C+ZCOORD(4)*C
      ST(1, 16) = -RCODR(4)*S+ZCOORD(4)*C
      ST(2, 16) = RCOORD(4)*C+ZCOORD(4)*C
      ST(1, 17) = -RCODR(5)*C+ZCOORD(1)*S
      ST(2, 17) = RCOORD(5)*S+ZCOORD(1)*C
      ST(1, 18) = -RCODR(5)*C+ZCOORD(2)*S
      ST(2, 18) = RCOORD(5)*S+ZCOORD(2)*C
      1 IF( KTEST.NE.0 ) WRITE(6,1)
      C... DS MESH SIZE
      S1 = ST(1,1)
      S2 = ST(1,4)
      S3 = ST(1,12)
      DPART
      DX = (S3-S1)/DPART
      M1 = (S2-S1)/DX
      M1 = M1+1
      
```

## Appendix F (Continued)

## Appendix F (Continued)

```

      IF( ANGLE.GT.45.0 )      M1 = M1-1
C.   C:S = M1:(KPART-M1)      KFNO = 5
      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.   SIN/COS      KCOVER = 1
      CS = C/S      H1 = -S2X*XX
      SC = S/C      KTYPE = 7
      S2X = 1.0/(S*C)      KFNO = 5
      SX = 1.0/S      KCOVER = 0
      RS = RCOORD(5)**2      H1 = 0.0
      RSMAX = RCOORD(5)*SX      IF( XX.GT.ST(1,3) )    GO TO 53
      B = RSMAX**2      XL1 = S2X*XX
      R3 = RCOORD(3)**2      GO TO 90
      R3MAX = RCOORD(3)*SX      IF( XX.GT.ST(1,4) )    GO TO 54
      G = R3MAX**2      XL1 = S2X*XX
      R1 = RCOORD((1)**2      GO TO 90
      R1MAX = RCOORD((1)*SX      IF( XX.GT.0.0 )      GO TO 57
      C.   DPART = M1      KTYPE = 1
      DX1 = (S2-S1)/DPART      KFNO = 5
      DPART = M2      KCOVER = 1
      DX2 = (S3-S2)/DPART      IF( XX.LE.ST(1,18) )    GO TO 53
      DPART = NPART      IF( XX.LE.ST(1,17) )    GO TO 54
      DX0 = RCOORD(5)/DPART      IF( XX.GT.0.0 )      GO TO 57
      C.   M6M1 = M1      KTYPE = 1
      M6M2 = M1+1      XL1 = 0.0
      C.   WRITE(6,2)      H1 = -S2X*XX
      CCC M1, DX1, M2, DX2, DX0      KFNO = 5
      2 FORMAT(1H0, 8H L-PART=,15,F10.4,10H R-PART=,15,F10.4,10X,
      + 6H DX=,F10.4 /),)      GO TO 58
      C.   COVER(1) = SX*THICK1      H1 = 0.0
      COVER(2) = SX*THICK2      KCOVER = 2
      COVER(3) = SX*THICK3      KTYPE = 2
      C.   MESH DATA      KFNO = 5
      CRC XX = 0      KCOVER = 1
      XX = S1 + 0.5*DX1      IF( XX.GT.ST(1,1) )    GO TO 61
      DO 100 J= 1, M1      KTYPE = 11
      C. (HO)      HO = RCOORD(5)*S - SC*XX      XL1 = 0.0
      C. (KTYPE) AND (XL)      H1 = -S2X*XX
      C. (KTYPE) AND (XL)      KCOVER = 0
      IF( XX.GT.ST(1,5) )      KTYPE = 8
      IF( KSIGDD.EQ.0 )      H1 = 0.0
      C.   XL1 = 0.0      KFNO = 1
      IF( XX.LE.ST(1,9) )      XL2 = R5MAX
      IF( XX.GT.ST(1,9) )      XL2 = S2X*(ZCOORD(4)*S-XX)
      IF( XX.GT.0.0 )      GO TO 90
      KTYPE = 1      KFNO = 2
      IF( XX.GT.ST(1,14) )    KFNO = 2

```

## Appendix F (Continued)

## Appendix F (Continued)

```

IFC( XX.GT.ST(1,3) )      GO TO 63
KTYPE = 8
H1    = 0.0
XL1   = S2X*XX
GO TO 90
63 KTYPE = 8
H1    = 0.0
XL1   = S2X*XX
KCOVER = 1
GO TO 90
64 KTYPE = 9
H1    = 0.0
XL1   = S2X*XX
IFC( XX.LT.ST(1,3) )
IFC( XX.GE.0.0 )
H1    = -S2X*XX
XL1   = 0.0
GO TO 90
C.. 65 KCOVER = 1
IFC( XX.GT.0.0 )
KTYPE = 4
XL1   = 0.0
H1    = -S2X*XX
GO TO 90
67 KTYPE = 6
H1    = 0.0
XL1   = S2X*XX
IFC( XX.LE.ST(1,4) )
XL1   = R5MAX
KCOVER = 0
GO TO 90
C.. 66 KCOVER = 1
IFC( XX.GT.0.0 )
KTYPE = 5
KFN0 = 5
KCOVER = 1
XL2   = 0.0
XL1   = S2X*(XX-ZCOORD(4)*S)
IFC( XX.LT.ST(1,4) )
XL1   = R5MAX
KCOVER = 0
GO TO 90
C.. 67 KTYPE = 3
XL2   = 0.0
H1    = S2X*(XX-ZCOORD(4)*S)
XL1   = S2X*XX
IFC( XX.LT.ST(1,4) )
XL1   = R5MAX
KCOVER = 0
GO TO 90
85 IF( XX.LE.ST(1,12) )
IFC( XX.LE.ST(1,16) )
KFN0 = 4
KFN0 = 3
C.. 90 CONTINUE
MFACE = MFACE+1
NESH(2,MFACE) = KTYPE
NESH(3,MFACE) = KFN0
NESH(4,MFACE) = KCOVER
SLICE(1,MFACE) = H0
SLICE(2,MFACE) = XL1
SLICE(3,MFACE) = XL2
SLICE(4,MFACE) = H1
SLICE(5,MFACE) = DX1
SLICE(6,MFACE) = XX
SLICE(7,MFACE) = XX
100 XX = XX + DX1
C.. 91 CRC = S2 + 0.5*DX2
CRC = S2 + 0.1*DX2
KCOVER = 2
C.. 92 DO 200 J= 1, M2
HO = CS*(XX-RCOORD(5)*C)
C.. 93 IF( XX.GT.ST(1,5) )
IF( KSIGD.EQ.0. )
KTYPE = 7
KFN0 = 1
IFC( XX.GT.ST(1,14) )
H1 = 0.0
XL1 = R5MAX
XL2 = S2X*(ZCOORD(2)*S-XX)
GO TO 190
C.. 94 155 KFN0 = 5
IFC( XX.LE.ST(1,18) )
IFC( XX.LE.ST(1,17) )
KFN0 = 1
KTYPE = 2
H1 = 0.0
XL1 = R5MAX
XL2 = R5MAX
IFC( XX.LE.ST(1,9) )
IFC( XX.GT.ST(1,9) )
XL1 = R5MAX
XL2 = S2X*(ZCOORD(4)*S-XX)
GO TO 190
C.. 95 IF( XX.GT.ST(1,8) )
H1 = S2X*(XX-ZCOORD(2)*S)
GO TO 77
IFC( XX.LT.ST(1,4) )
XL1 = R5MAX
GO TO 90
C.. 96 IF( XX.GT.ST(1,8) )
H1 = S2X*(XX-ZCOORD(2)*S)
GO TO 77
IFC( XX.LT.ST(1,4) )
XL1 = R5MAX
GO TO 90

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C...  C PS(S)<S<QQ(S)   MESH(3,MFACE) = KFNO
      160 IFC XX.GT.ST(1,15)  GO TO 170
      IFC KSIGDD.EQ.0  GO TO 165
C     IFC XX.GT.ST(1,6)  GO TO 161
      KTYPE = 8
      H1 = 0.0
      XL1 = R$MAX
      XL2 = S2X*(ZCOORD(2)*S-XX)
      KFNO = 1
      IF C XX.GT.ST(1,14)  KFNO = 2
      GO TO 190
C 161 KTYPE = 9
      H1 = 0.0
      XL1 = R$MAX
      XL2 = S2X*(ZCOORD(2)*S-XX)
      KFNO = 1
      IF C XX.GT.ST(1,14)  KFNO = 2
      GO TO 190
C 165 KTYPE = 6
      KFNO = 5
      H1 = 0.0
      XL1 = R$MAX
      XL2 = S2X*(ZCOORD(2)*S-XX)
      GO TO 190
C...  C Q0(S)<S<P11(S)  MESH(3,MFACE) = KFNO
      170 IFC XX.GT.ST(1,11)  GO TO 180
      KTYPE = 5
      KFNO = 0
      XL2 = 0.0
      XL1 = R$MAX
      IFC KSIGDD.EQ.0  GO TO 175
      IFC XX.GT.ST(1,7)  GO TO 175
C     KTYPE = 9
      KFNO = 2
      IFC XX.LE.ST(1,14)  KFNO = 1
      H1 = 0.0
      XL2 = S2X*(XX-ZCOORD(2)*S)
      GO TO 190
C 175 H1 = S2X*(XX-ZCOORD(2)*S)
      IFC XX.GT.ST(1,8)  H1 = RCOOR(3)*SX
      GO TO 190
C...  C 180 KTYPE = 3
      XL2 = 0.0
      H1 = S2X*(XX-ZCOORD(4)*S)
      XL1 = R$MAX
      IFC XX.LE.ST(1,12)  KFNO = 4
      IFC XX.LE.ST(1,16)  KFNO = 3
C...  C 190 CONTINUE
      MFACE = MFACE+1
      MESH(2,MFACE) = KTYPE

```

## Appendix F (Continued)

## Appendix F (Continued)

```

315 XM0 = S*SQR(T 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.) THEN
        DPART = ML
        DX = (XM-XM0)/DPART
        XX = XM0 + 0.5*DX
        MESH(1,J) = ML+1
        DO 316 I= 1, ML
          XL = SX*SQR(T RS-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
C. (2)-TYPE
320 CONTINUE
      H1 = -SLICE(2,J)
      H2 = SLICE(3,J)
      CALL SIMPLL (H1,H2,SLICE(7,J))
      ML = RCDR(5)/DX0
      DPART = ML
      DX = RCDR(5)/DPART
      XX = 0.5*DX
      MESH(1,J) = ML
      IF (SLICE(2,J).NE.RSMAX) THEN
        DO 321 I= 1, ML
          XL = SX*SQR(T RS-XX**2)
          MLOOP = MLOOP+1
          XLENG(1,MLOOP) = RSMAX-XL
          XLENG(2,MLOOP) = 2.0*XL
          XLENG(3,MLOOP) = DX
321  XX = XX+DX
      GO TO 500
C
      XM1 = S*SQR(T B-SLICE(3,J)**2)
      DO 323 I= 1, ML
        XL = SX*SQR(T RS-XX**2)
        MLOOP = MLOOP+1
        XLENG(1,MLOOP) = RSMAX-XL
        XLENG(2,MLOOP) = XL-SLICE(3,J)
        XLENG(3,MLOOP) = DX
        IF (XX.GT.XM1) THEN
          XL = RSMAX-XL
        ELSE
          XX = XX+DX
323  XM0 = S*SQR(T B-SLICE(2,J)**2)
      IF (SLICE(3,J).NE.RSMAX) THEN
        XLENG(2,MLOOP) = 2.0*XL
      GO TO 500
C
      325 XM0 = S*SQR(T B-SLICE(2,J)**2)
      IF (SLICE(3,J).NE.RSMAX) THEN
        XLENG(2,MLOOP) = SLICE(2,J)-H
      GO TO 327
DO 326 I= 1, ML
      XL = (XM-XM0)/DX0

```

## Appendix F (Continued)

## Appendix F (Continued)

```

IF( ML.EQ.0 )      ML    = 1
DPART = ML
DX   = (XM-XMM)/DPART
XX   = XM + 0.5*DX
MESH(1,J) = ML+1
DO 336 I= 1, ML
XL   = SX*SQRT( R5-XX**2 )
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = SLICE(2,J)-XL
XLENG(2,MLLOOP) = XL-SLICE(4,J)
XLENG(3,MLLOOP) = 0X
336 XX   = XX+DX
GO TO 500
C. (4)-TYPE
C. 340 CONTINUE
H1   = H
H2   = R5MAX
CALL SIMPL( H1,H2-SLICE(7,J) )
XMM  = S*SQRT( G-SLICE(3,J)**2 )
H    = SLICE(4,J)
XMM = S*SQRT( B-H**2 )
ML   = (XM-XMM)/DXO
IF( ML.EQ.0 )      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( RS-XX**2 )
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = 0.0
XLENG(2,MLLOOP) = XL-H
XLENG(3,MLLOOP) = DX
341 XX   = XX+DX
GO TO 500
C. (5)-TYPE
C. 350 CONTINUE
H1   = -SLICE(2,J)
H2   = R5MAX
CALL SIMPL( H1,H2-SLICE(7,J) )
H    = SLICE(4,J)
XMO  = S*SQRT( B-SLICE(2,J)**2 )
XMM  = S*SQRT( G-H**2 )
XXX  = XM
IF( XMM.LT. XMO )      XXX = XMM
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = 0.0
XLENG(2,MLLOOP) = SLICE(2,J)-H
XLENG(3,MLLOOP) = XXX
C.
ML1  = 0
IF( XM.GE.XMM )      GO TO 355
ML1  = (XMM-XMO)/DXO
IF( ML1.EQ.0 )      ML1  = 1
DPART = ML1
DX   = (XMM-XMO)/DPART

```

```

XX   = XM0 + 0.5*DX
DO 351 I= 1, ML1
XL   = SX*SQRT( R5-XX**2 )
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = SLICE(2,J)-XL
XLENG(2,MLLOOP) = XL-SLICE(4,J)
XLENG(3,MLLOOP) = 0X
351 XX   = XX+DX
XXX  = XMM
C. 355 ML2 = 0
IF( XXX.GE.RC00R(3) )      GO TO 357
ML2 = (RC00R(3)-XXX)/DXO
ML2  = 1
IF( ML2.EQ.0 )
DPART = ML2
DX   = (RC00R(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)
MLLOOP = MLLOOP+1,
XLENG(1,MLLOOP) = SLICE(2,J)-XL
XLENG(2,MLLOOP) = XL-XLM
XLENG(3,MLLOOP) = DX
IF( XX.LE.XMO )
352 XX   = XX+DX
C. 357 ML3 = (RC00R(5)-RC00R(3))/DXO
IF( ML3.EQ.0 )
DPART = ML3
DX   = (RC00R(5)-RC00R(3))/DPART
XX   = RC00R(3) + 0.5*DX
DO 353 I= 1, ML3
XL   = SX*SQRT( RS-XX**2 )
IF( XL.GT.SLICE(2,J) )      XL   = SLICE(2,J)
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = SLICE(2,J)-XL
XLENG(2,MLLOOP) = 2.0*XL
XLENG(3,MLLOOP) = DX
353 XX   = XX+DX
C. MESH(1,J) = ML1+ML2+ML3+1
GO TO 500
C. (6)-TYPE
C. 360 CONTINUE
H1   = -SLICE(2,J)
H2   = R5MAX
CALL SIMPL( H1,H2-SLICE(7,J) )
XMO  = S*SQRT( B-SLICE(2,J)**2 )
XMM  = S*SQRT( G-SLICE(3,J)**2 )
XXX  = XM
XXX = XMM
IF( XMM.LT. XMO )      XXX = XMM
MLLOOP = MLLOOP+1
XLENG(1,MLLOOP) = 0.0
XLENG(2,MLLOOP) = SLICE(2,J)-H
XLENG(3,MLLOOP) = XXX
C.
ML1  = 0
IF( XM.GE.XMM )      GO TO 355
ML1  = (XMM-XMO)/DXO
IF( ML1.EQ.0 )      ML1  = 1
DPART = ML1
DX   = (XMM-XMO)/DPART

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C
      ML = (RCGDR(5)-XXX)/DX0    ML      = SLICE(3,J)
      IF( ML.EQ.0 )               XL
      DPART = ML                  MLOOP = MLOOP+1
      DX = (RCGDR(5)-XXX)/DPART  XLENG(1,MLOOP) = 0.0
      XX = XX + 0.5*DX           XLENG(2,MLOOP) = XL-H
      DO 363 I= 1, ML            XLENG(3,MLOOP) = DX
      XL = SX*SQRT( RS-XX**2 )   372 XX = XX+DX
      IF( XX.GT.XMO )           GO TO 361
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XI+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
      GO TO 500
C. (8)-TYPE
C. 370 CONTINUE
      H1 = SLICE(4,J)
      H2 = SLICE(3,J)
      IF( SLICE(2,J).GT.0.0 )    H1 = -SLICE(2,J)
      CALL SIMPLL( H1,H2,SLICE(7,J) )
      IF( SLICE(2,J).GT.0.0 )    GO TO 373
      H = SLICE(4,J)
      X1 = SSQRT( R1MAX**2-H**2 )
      XM = S*SQRT( B-H**2 )
      ML = X1/DX0
      IF( ML.EQ.0 )             ML = 1
      DPART = ML
      DX = X1/DPART
      XX = 0.5*DX
      MESH(1,J) = ML
      DO 371 I= 1, ML
      XL = SX*SQRT( RS-XX**2 )
      IF( XL.GT.SLICE(3,J) )    XL = SLICE(3,J)
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = XL-H
      XLENG(2,MLOOP) = XLM-XL
      XLENG(3,MLOOP) = DX
      371 XX = XX+DX
      ML = (XM*XI)/DX0
      IF( ML.EQ.0 )             ML = 1
      DPART = ML
      DX = (XM*XI)/DPART
      XX = X1*0.5*DX
      MESH(1,J) = MESH(1,J)+ML
      DO 372 I= 1, ML
      XL = SX*SQRT( RS-XX**2 )
      XL = SLICE(3,J)
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL-x2
      XLENG(3,MLOOP) = DX
      372 XX = XX+DX
      GO TO 500
C. (8)-TYPE

```

## Appendix F (Continued)

## Appendix F (Continued)

```

380 CONTINUE
      H1 = SLICE(4,J)
      H2 = RMAX
      IF( SLICE(2,J) GT 0.0 ) H1 = -SLICE(2,J)
      CALL SIMPL( H1,H2,SLICE(7,J) )
      Y3 = SLICE(3,J)
      X3 = S*SQR( G-SLICE(3,J)**2 )
      IF( SLICE(2,J).GT.0.0 ) GO TO 383
      H = SLICE(4,J)
      X1 = S*SORT( R1MAX**2-H**2 )
      XM = S*SORT( B-H**2 )
      ML = X1DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = X1/DPART
      XX = 0.5*DX
      MESH(1,J) = ML+1
      DO 381 I= 1, ML
      XL = SX*SQRT( R1-XX**2 )
      MLOOP = MLLOOP+1
      XLENG(1,MLLOOP) = XL-H
      XLENG(2,MLLOOP) = Y3-XL
      XLENG(3,MLLOOP) = DX
      XX = XX+DX
      C
      MLOOP = MLLOOP+1
      XLENG(1,MLLOOP) = 0.0
      XLENG(2,MLLOOP) = Y3-H
      XLENG(3,MLLOOP) = X3-X1
      C
      ML = (XM-X3)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (XM-X3)/DPART
      XX = X3+0.5*DX
      MESH(1,J) = MESH(1,J)+ML
      DO 388 I= 1, ML
      XL = SX*SQRT( R5-XX**2 )
      MLOOP = MLLOOP+1
      XLENG(1,MLLOOP) = 0.0
      XLENG(2,MLLOOP) = XL-X2
      XLENG(3,MLLOOP) = DX
      XX = XX+DX
      GO TO 500
      C. (9)-TYPE
      390 CONTINUE
      H1 = -SLICE(2,J)
      H2 = RSMAX
      CALL SIMPL( H1,H2,SLICE(7,J) )
      Y3 = SLICE(3,J)
      X3 = S*SORT( G-SLICE(3,J)**2 )
      IF( SLICE(2,J).GT.0.0 ) GO TO 393
      H = SLICE(4,J)
      X0 = S*SORT( R1MAX**2-SLICE(3,J)**2 )
      X1 = S*SORT( R1MAX**2-H**2 )
      XM = S*SORT( B-H**2 )
      ML = (X1-X0)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (X1-X0)/DPART
      XX = X0+0.5*DX
      MESH(1,J) = ML+1
      DO 384 I= 1, ML
      XL = SX*SQRT( R1-XX**2 )
      C
      383 X2 = SLICE(2,J)
      X1 = RCDR(1)
      XM = RCDR(5)
      IF( SLICE(2,J).GT.R1MAX ) GO TO 385
      X0 = S*SORT( R1MAX**2-X2**2 )
      ML = (X1-X0)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (X1-X0)/DPART
      XX = X0+0.5*DX
      MESH(1,J) = ML+1
      GO TO 387
      C
      385 ML = X1/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = X1/DPART
      XX = 0.5*DX
      MESH(1,J) = ML+1
      DO 386 I= 1, ML
      XL = SX*SQRT( R1-XX**2 )
      MLOOP = MLLOOP+1
      XLENG(1,MLLOOP) = 0.0
      XLENG(2,MLLOOP) = X3-X2
      XLENG(3,MLLOOP) = X3*X1
      C
      ML = (XM-X3)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (XM-X3)/DPART
      XX = X3+0.5*DX
      MESH(1,J) = MESH(1,J)+ML
      DO 388 I= 1, ML
      XL = SX*SQRT( R5-XX**2 )
      MLOOP = MLLOOP+1
      XLENG(1,MLLOOP) = 0.0
      XLENG(2,MLLOOP) = XL-X2
      XLENG(3,MLLOOP) = DX
      XX = XX+DX
      GO TO 500
      C
      H1 = -SLICE(2,J)
      H2 = RSMAX
      CALL SIMPL( H1,H2,SLICE(7,J) )
      Y3 = SLICE(3,J)
      X3 = S*SORT( G-SLICE(3,J)**2 )
      IF( SLICE(2,J).GT.0.0 ) GO TO 393
      H = SLICE(4,J)
      X0 = S*SORT( R1MAX**2-SLICE(3,J)**2 )
      X1 = S*SORT( R1MAX**2-H**2 )
      XM = S*SORT( B-H**2 )
      ML = (X1-X0)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (X1-X0)/DPART
      XX = X0+0.5*DX
      MESH(1,J) = ML+1
  
```

## Appendix F (Continued)

### Appendix F (Continued)

```

DO 391 I= 1, ML
  XL = SX*SQRT( R1-XX**2 )
  MLOOP = MLOOP+1
  XLENG(1,MLLOOP) = XL-H
  XLENG(2,MLLOOP) = Y3-XL
  XLENG(3,MLLOOP) = DX
  391 XX = XX+DX
C   MLOOP = MLOOP+1
  XLENG(1,MLLOOP) = 0.0
  XLENG(2,MLLOOP) = Y3-H
  XLENG(3,MLLOOP) = X3-X1
C   ML = (XM-X3)/DX0
  IF( ML.EQ.0 ) ML = 1
  DPART = ML
  DX = (XM-X3)/DPART
  XX = X3+0.5*DX
  MESH(1,J) = MESH(1,J)+ML
  DO 392 I= 1, ML
    XL = SQRT( RS-XX**2 )
    MLOOP = MLOOP+1
    XLENG(1,MLLOOP) = 0.0
    XLENG(2,MLLOOP) = XL-H
    XLENG(3,MLLOOP) = DX
  392 XX = XX+DX
  GO TO 500
C   C. (10)-TYPE
  400 CONTINUE
    H1 = -SLICE(2,J)
    H2 = RMAX
    CALL SIMPLI( H1/H2, SLICE(7,J) )
    H = SLICE(4,J)
    X3 = S*SQRT( G-H**2 )
    X2 = SLICE(2,J)
    X1 = S*SQRT( RIMAX**2-H**2 )
    X0 = S*SQRT( RIMAX**2-X2**2 )
    XM = S*SQRT( B-X2**2 )
    IF( ML.EQ.0 ) ML = 1
    DPART = ML
    DX = (X1-X0)/DPART
    XX = X0+0.5*DX
    MESH(1,J) = ML+1
  DO 394 I= 1, ML
    XL = SX*SQRT( R1-XX**2 )
    MLOOP = MLOOP+1
    XLENG(1,MLLOOP) = 0.0
    XLENG(2,MLLOOP) = X2-XL
    XLENG(3,MLLOOP) = DX
  394 XX = XX+DX
  GO TO 397
  395 ML = X1/DX0
  IF( ML.EQ.0 ) ML = 1
  DPART = ML
  DX = X1/DPART
  XX = 0.5*DX
  MESH(1,J) = ML+1
  DO 396 I= 1, ML
    XL = SX*SQRT( R1-XX**2 )
    MLOOP = MLOOP+1
    XLENG(1,MLLOOP) = 0.0
    XLENG(2,MLLOOP) = X3-X1
    XLENG(3,MLLOOP) = DX
  396 XX = X3+0.5*DX
C   MLOOP = MLOOP+1
  XLENG(1,MLLOOP) = 0.0
  XLENG(2,MLLOOP) = X2-H
  XLENG(3,MLLOOP) = DX
  401 XX = XX+DX
C   MLOOP = MLOOP+1
  XLENG(1,MLLOOP) = 0.0
  XLENG(2,MLLOOP) = X3-X1
  XLENG(3,MLLOOP) = ML
  IF( ML.EQ.0 ) ML = 1
  DPART = ML
  DX = (XM-X3)/DPART
  XX = X3+0.5*DX

```

## Appendix F (Continued)

## Appendix F (Continued)

```

      MESH(1,J) = MESH(1,J)+ML
      DO 402 I= 1, ML
      XL = SX*SQRT( RS-XX**2 )
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XI-X2
      XLENG(3,MLOOP) = DX
      402 XX = XX+DX
      GO TO 500
C- <(11)-TYPE
C- 410 CONTINUE
      H1 = H
      H2 = R$MAX
      CALL SIMPLL (H1,H2,SLICE(7,J))
      H = SLICE(4,J)
      Y3 = SLICE(3,J)
      X3 = S*SQR( G-Y3**2 )
      XM = S*SQR( B-H**2 )
      MESH(1,J) = 1
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = Y3-H
      XLENG(3,MLOOP) = X3

      C ML = (XM-X3)/DX0
      IF( ML.EQ.0 ) ML = 1
      DPART = ML
      DX = (XM-X3)/DPART
      XX = X3+0.5*DX
      MESH(1,J) = MESH(1,J)+ML
      DO 411 I= 1, ML
      XL = SX*SQRT( RS-XX**2 )
      MLOOP = MLOOP+1
      XLENG(1,MLOOP) = 0.0
      XLENG(2,MLOOP) = XL-H
      XLENG(3,MLOOP) = DX
      411 XX = XX+DX
      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 XLENG,I5,3F12.5)
      IF( XLENG(2,I).GE.XMIN )
      XMIN = XLENG(2,I)
      HMIN = XLENG(1,I)
      550 IF( XLENG(2,I).LE.XMAX ) GO TO 550
      XMAX = XLENG(2,I)
      HMAX = XLENG(1,I)
      550 CONTINUE
      IF( KTEST.NE.0 )
      + WRITE(6,8) MLOOP1, HMAX, HMAX
      XMIN, HMIN, XMAX, HMAX
      CCC 9 FORMAT(14H *TOTAL MESH */SX18HNO. OF (S-T) FACE=-16,10X,
      + 18HNO. OF (X-Y) FACE=,16)
      C M7LOOP = 0
      IF( MLENG(7).GT.0 ) M7LOOP = 1
      C
      C- GET CURRENT FORCE
      C- 1000 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 ) GO TO 2000
      TDISP = TDISP + DISPX
      C
      C- SUM = 0.0
      TUM = 0.0
      MLOOP = 0
      C
      DO 1500 J= 1, MFACE
      UO = TDISP - SLICE(1,J)
      IF( UO.LE.0.0 ) GO TO 1450
      C
      ML = MESH(1,J)
      KENO = MESH(3,J)
      MULT = MESH(4,J)
      PSUM = 0.0
      ESUM = 0.0
      C
      C- (COVER)
      EPS = UO/SLICE(7,J)
      CALL GETSIG( EPS, 7, SIGM )
      UU = DDISP
      IF( UO.LT.DDISP ) PSUM = PSUM + SIGM*COVER(2)
      UU = UO
      ESUM = ESUM + SIGM*COVER(2)*UU
      C
      MUSE = 0
      IF( J.LT.M6M1 ) GO TO 1100
      YF( J.GT.M6M2 ) GO TO 1100
      MUSE = 1
      1100 CONTINUE
      DO 1400 I= 1, ML
      MLOOP = MLOOP+1
      U = UO-XLENG(1,MLOOP)
      IF( U.LE.0.0 ) GO TO 1400
      IF( MLOOP.GT.0 ) GO TO 1200
      EPS = UXLENG(2,MLOOP)
      CCC 10 IF( MXSTEP.GT.5 ) GO TO 1090
      CCC 11 IF( EPS.LE.0.30 ) GO TO 1090
      CCC 12 WRITE(6,19) MXSTEP, MLOOP, (XLENG(K,MLOOP),K=1,3), U, TDISP, UO
      C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

CCC19 FORMAT(6H OVER=,215,8F12.5)
C1090 CONTINUE
  IF( EPS.LT.EPMAX )          GO TO 1110
  EPS = EPMAX
  U   = EPMAX*XLENG(2,MLLOOP)
  UCONTINUE
  CALL GETSIG( EPS, S, SIGM )
  GO TO 1300
1110 ID(1) = 5
  ID(2) = 7
  DL(1) = XLENG(2,MLLOOP)
  DL(2) = COVER(2)
  CALL MULTI( U, DL, ID, 2, SIGM, IERR )
  IF( IERR.NE.0 )             GO TO 3000
  EPS = EPMSXV
  GO TO 1300
1200 ID(1) = 5
  ID(2) = S*MULT
  DL(1) = XLENG(2,MLLOOP)
  DL(2) = COVER(MULT)
  KS3 = 2
  IF( MLLOOP.EQ.0.0 )         GO TO 1250
  IF( MULT.EQ.2.0 )           GO TO 1210
  ID(3) = 7
  DL(3) = COVER(2)
  KS3 = 3
  GO TO 1250
1210 DL(2) = 2.0*COVER(2)
  CALL MULTI( U, DL, ID, KS3, SIGM, IERR )
  IF( IERR.NE.0 )             GO TO 3000
  EPS = EPMSXV
  C
  1300 UU = DISPX
  IF( U.LT.DISPX )            UU = U
  PSUM = PSUM + SIGN*XLENG(3,MLLOOP)
  ESUM = ESUM + SIGN*XLENG(3,MLLOOP)*UU
  C
  IF( M6USE.EQ.0 )            GO TO 1400
  IF( EPSU(MXSTEP,10).GT.EPS ) GO TO 1400
  MSUSE(10) = MXSTEP
  EPSU(MXSTEP,10) = EPS
  SIGU(MXSTEP,10) = SIGM
  C
  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 MLLOOP)
  1450 ML = MESH(1,J)
  MLLOOP = MLLOOP+ML
  C
  1500 CONTINUE
  C
  CALL GETSIG( TODISP, 9, P0 )
  SUM = SUM + P0
  TUM = TUM + P0*DISPX
  C-- STEP RESULTANT
  C
  ENERGY = TUM

```

- 73 -

```

C DEPTH(MXSTEP) = TODISP
C FORCE(MXSTEP) = SUM
C ACCEL(MXSTEP) = SUM/WEIGHT
C ENERG(MXSTEP) = ENERGY
C IFC( MXSTEP.NE.1 )      ENERG(MXSTEP) = ENERG(MXSTEP-1)+ENERGY
C
C IFC( ENERG(MXSTEP).LT.ENERG ) GO TO 1000
C IFC( MINI.NE.0 )        GO TO 1800
MINI = 1
C
C TODISP = TODISP - DISPX
MXSTEP = MXSTEP-1
DISPX = 0.1*DISP
GO TO 1000
C-- INTERPOLATION OF LAST STEP
C
C 1800 CONTINUE
CALL FITSP
C-- END OF CORNER
C
C 2000 CONTINUE
IF( MXSTEP.GT.500 ) MXSTEP = 500
RETURN
C*** ERROR
3000 CONTINUE
3099 WRITE(MTOUT,3099) MXSTEP, TODISP, 10, J
3099 FORMAT(// 29H *** MULTI-MATERIAL ERROR. 5X,7H STEP =,15,
+ 5X,7HDEPTH =,F10.4,5X,11HNO. OF MAT=,3H 2.5X,2.3,6H FACE=,15)
CALL EXIT
RETURN
C
C 4000 CONTINUE
4000 WRITE(MTOUT,4099)
4099 FORMAT(// 29H *** (XLENG) CAPACITY OVER. )
CALL EXIT
RETURN
END
C
C*** /PURPOSE/
C*** GET STIFFNESS AND INITIAL SIGM
C*** /OUTPUT/
C*** BK = STIFFNESS
C*** SIG0 = INITIAL SIGM
C
C*** /INPUT/
C*** EPS = STRAIN
C*** IO = MATERIAL IDENT
C
C*** /PROGRAMMED BY T-JAERI
C*** MODIFIED BY T-JAERI
C*** MODIFIED BY T-JAERI
C*** MODIFIED BY T-JAERI
C
SUBROUTINE GETMAT( EPS, ID, BK, SIG0 )

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C      IF( EPS.GT.ELIMIT(ID) )    EPX = ELIMIT(ID)
C      DO 1400 J2= 2, LENGTHH
C         IF( EPX.LT.TABLE(J2,1,ID) )  GO TO 1500
C      1400 CONTINUE
C         J2 = LENGTH
C      1500 CONTINUE
C         J1 = J2-1
C         DX = TABLE(J2,1-ID)-TABLE(J1,1-ID)
C         DY = TABLE(J2,2-ID)-TABLE(J1,2-ID)
C
C         SIG0 = TABLE(J1+2,1-ID)+DY/DX*(EPX-TABLE(J1,1-ID))
C
C         CRC WRITE(6,5) ID, EPX, SIG0
C         CRC S FORMAT(8H GETSIG=,I5,6H EPS=F12.4,6H SIG=,F12.4)
C
C         MSUSE(ID) = MXSTEP
C         EPSUMXSTEP>ID) = DMAX1( EPSUMXSTEP-ID), EPX )
C         SIGUMXSTEP>ID) = DMAX1( SIGUMXSTEP-ID), SIG0 )
C
C         RETURN
C
C***   ERROR
C***   1600 CONTINUE
C         SIG0 = 0.0
C
C         RETURN
C
C
C      300 CONTINUE
C         EPX = EPS
C         IF( EPS.LE.0.0 )    EPX = 0.0
C         IF( EPS.GT.ELIMIT(ID) )    EPX = ELIMIT(ID)
C         DO 400 J2= 2, LENGTH
C            IF( EPX.LT.TABLE(J2,1-ID) )  GO TO 500
C            J2 = LENGTH
C      400 CONTINUE
C
C      500 CONTINUE
C         J1 = J2-1
C         DX = TABLE(J2,1-ID)-TABLE(J1,1-ID)
C         DY = TABLE(J2,2-ID)-TABLE(J1,2-ID)
C         IF( ABS(DX).LT.DLT )  GO TO 1000
C
C         BK = DY/DX
C         SIG0 = TABLE(J1,2-ID)-BK*TABLE(J1,1-ID)
C
C         RETURN
C
C***   ERROR DATA
C
C         1000 WRITE(6,1010) ID, J1, J2
C         1010 FORMAT(24H *** ERROR MATERIAL DATA,10X,4H ID=,I5,8H COLUMN=,2I5)
C
C         CALL EXIT
C         RETURN
C
C         ENTRY
C         GETSIG( EPS, ID, S 1 6 0 )
C
C         LENGTH = MLENG(ID)
C         IF( LENGTH.EQ.0 )  GO TO 1600
C
C         EPX = EPS
C
C         IMPLICIT REAL*8 (A-H,O-Z)
C         COMMON /MODELS/
C            RCOORD(5), ZCOORD(5), WEIGHT,
C            FACTR(5), ANGLE,
C            DISP, HEIGHT,
C            MTINN, GRAVIT,
C            KSOVL, EPMAX,
C            NTITLE(18),
C            COMMON /RSRL/
C            MTOUR, IFLAG,
C            KOUT, NPART,
C            KPRT,
C            COMMON /DEPTH/
C            DEPTH(500),
C            ENERGY(500),
C            EPSU(500),
C            SIGU(500,10),
C            DATA EPS/ 0.05 /
C
C***   FUNCTION
C         ABS(Z) = DABS(Z)
C
C         C...
C
C         GO TO 200
C
C         100 MXSTEP = MXSTEP-1
C
C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

200 CONTINUE
  NXSTEP = MXSTEP-1
  DL1 = ENERG(NXSTEP)-ENERG(NXSTEP)
  DL2 = ENERG(NXSTEP)-ENERG(NXSTEP)
  IF( ABS(DL2) .LE. EPS ) GO TO 100
  DL1 = DL2/DL1

C   DEPTH(MXSTEP) = DLT*(DEPTH(MXSTEP)-DEPTH(NXSTEP)) + DEPTH(NXSTEP)
C   FORCE(MXSTEP) = DLT*(FORCE(MXSTEP)-FORCE(NXSTEP)) + FORCE(NXSTEP)
C   ACCEL(MXSTEP) = DLT*(ACCEL(MXSTEP)-ACCEL(NXSTEP)) + ACCEL(NXSTEP)
C   ENERG(MXSTEP) = ENERG(NXSTEP)

C   RETURN
END

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

```

— 75 —

## Appendix F (Continued)

### Appendix F (Continued)

```

C... FIND SAME MATERIAL
C
      DO 270 J= 2, LENGTH
      I = J-1
      SAME(1) = DABS(B(J)-B(I))
      KSAME(I) =0
      IF( SAME(I).LE.0LT ) KSAME(I) =1
      270 CONTINUE

C... CALCULATION OF TRUSS DISP
C
      300 CONTINUE
      IF( LENGTH.NE.2 ) GO TO 400
      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)
      FC( 1) = DISP
      FC( 2) = SO(1)-SO(2)
      CALL EQSOLV( SK, F, 2, DU, IERR )
      GO TO 600

C... (3)-TRUSS
      400 IF( LENGTH.NE.3 ) GO TO 500
      IF( KSAME(1).NE.0 ) GO TO 410
      IF( KSAME(2).NE.0 ) GO TO 420
      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)
      FC( 1) = DISP
      FC( 2) = SO(1)-SO(2)
      FC( 3) = SO(1)-SO(3)
      CALL EQSOLV( SK, F, 3, DU, IERR )
      GO TO 600
      410 IF( KSAME(2).NE.0 ) GO TO 3000
      SK( 1) = 1.0
      SK( 3) = 1.0
      SK( 2) = -B(1)/(DL(1)+DL(2))
      SK( 4) = B(3)/DL(3)
      FC( 1) = DISP
      FC( 2) = SO(1)-SO(3)
      CALL EQSOLV( SK, F, 2, DU, IERR )
      DU( 3) = DU( 2)
      DU( 1) = DU( 2)
      X1 = DL(1)/(DL(1)+DL(2))
      X2 = DL(2)/(DL(1)+DL(2))
      DU( 1) = D1*X1
      DU( 2) = D1*X2
      GO TO 600
      420 SK( 1) = 1.0
      SK( 3) = 1.0

```

```

      SK( 2) = -B(1)/DL(1)
      SK( 4) = B(2)/DL(2)+DL(3)
      CALL EQSOLV( SK, F, 2, DU, IERR )
      D1 = DU( 2)
      X1 = DL(2)/(DL(2)+DL(3))
      X2 = DL(3)/(DL(2)+DL(3))
      DU( 2) = D1*X1
      DU( 3) = D1*X2
      GO TO 600

C... (4)-TRUSS
      500 CONTINUE
      IF( KSAME(1).NE.0 ) GO TO 510
      IF( KSAME(2).NE.0 ) GO TO 540
      IF( KSAME(3).NE.0 ) GO TO 560
      SK( 1) = 1.0
      SK( 5) = 1.0
      SK( 9) = 1.0
      SK( 13) = 1.0
      SK( 2) = -B(1)/DL(1)
      SK( 6) = B(2)/DL(2)
      SK(10) = 0.0
      SK(14) = 0.0
      SK( 3) = -B(1)/DL(1)
      SK( 7) = 0.0
      SK(11) = B(3)/DL(3)
      SK(15) = 0.0
      SK( 4) = -B(1)/DL(1)
      SK( 8) = 0.0
      SK(12) = 0.0
      SK(16) = B(4)/DL(4)
      FC( 1) = DISP
      FC( 2) = SO(1)-SO(2)
      FC( 3) = SO(1)-SO(3)
      FC( 4) = SO(1)-SO(4)
      CALL EQSOLV( SK, F, 4, DU, IERR )
      GO TO 600
      510 IF( KSAME(2).NE.0 ) GO TO 530
      IF( KSAME(3).NE.0 ) GO TO 520
      SK( 1) = 1.0
      SK( 4) = 1.0
      SK( 7) = 1.0
      SK( 2) = -B(1)/(DL(1)+DL(2))
      SK( 5) = B(3)/DL(3)
      SK( 8) = 0.0
      SK( 3) = SK( 2)
      SK( 9) = B(4)/DL(4)
      FC( 1) = DISP
      FC( 2) = SO(1)-SO(3)
      FC( 4) = SO(1)-SO(4)
      CALL EQSOLV( SK, F, 3, DU, IERR )
      DU( 4) = DU( 3)
      DU( 3) = DU( 2)
      D1 = DU( 1)
      X1 = DL(1)/(DL(1)+DL(2))
      X2 = DL(2)/(DL(1)+DL(2))

```

## Appendix F (Continued)

## Appendix F (Continued)

```

DUC( 1) = D1*X1
DUC( 2) = D1*X2
GO TO 600
520 SK( 1) = 1.0
SK( 3) = 1.0
SK( 2) = -B(1)/(DL(1)+DL(2))
SK( 4) = B(3)/(DL(3)+DL(4))
F( 1) = DISP
F( 2) = SO(1)-SO(3)
CALL EQSOLV( SK, F, 2, DU, IERR )
D1 = DUC( 2)
X1 = DL(5)/(DL(3)+DL(4))
X2 = DL(6)/(DL(3)+DL(4))
DU( 3) = D1*X1
DU( 4) = D1*X2
D1 = DUC( 1)
X1 = DL(1)/(DL(1)+DL(2))
X2 = DL(2)/(DL(1)+DL(2))
DU( 1) = D1*X1
DU( 2) = D1*X2
GO TO 600
530 IF( KSAME(3).NE.0 ) GO TO 3000
SK( 1) = 1.0
SK( 3) = 1.0
SK( 2) = -B(1)/(DL(1)+DL(2)+DL(3))
SK( 4) = B(4)/DL(4)
F( 1) = DISP
F( 2) = SO(1)-SO(4)
CALL EQSOLV( SK, F, 2, DU, IERR )
DU( 4) = DUC( 2)
D1 = DU( 1)
X1 = DL(1)/(DL(1)+DL(2)+DL(3))
X2 = DL(2)/(DL(1)+DL(2)+DL(3))
X3 = DL(3)/(DL(1)+DL(2)+DL(3))
DU( 1) = D1*X1
DU( 2) = D1*X2
DU( 3) = D1*X3
GO TO 600
540 IF( KSAME(3).NE.0 ) GO TO 550
SK( 1) = 1.0
SK( 4) = 1.0
SK( 7) = 1.0
SK( 2) = -B(1)/DL(1)
SK( 5) = B(2)/(DL(2)+DL(3))
SK( 8) = 0.0
SK( 3) = SK( 2)
SK( 6) = 0.0
SK( 9) = B(4)/DL(4)
F( 1) = DISP
F( 2) = SO(1)-SO(2)
F( 3) = SO(1)-SO(3)
CALL EQSOLV( SK, F, 3, DU, IERR )
D1 = DUC( 3)
X1 = DL(3)/(DL(3)+DL(4))
X2 = DL(4)/(DL(3)+DL(4))
DU( 3) = D1*X1
DU( 4) = D1*X2
C
C... CONVERGENCE
C
600 CONTINUE
IF( IERR.NE.0 ) NCV = 0
DO 700 J= 1, LENGTH
EPS1(J) = DU(J)/DL(J)
DCONV = EPS1(J)-EPS1(J)
IF( EPS1(J).NE.0.0 ) NCV = NCV+1
700 EPS(J) = EPS1(J)
IF( NCV.EQ.LENGTH ) GO TO 800
C...
IF( ITER.LE.MXITER ) GO TO 1000
GO TO 1000
C... END OF MULTI
C
800 CONTINUE
C WRITE(6,1) '(ID(I),DUC(I),EPS1(I),I=1,LENGTH)'
C 1 FORMAT(6H MULT=,I4,6H DL =,F12.4,6H DU =,F12.4,6H EPS=F12.4)
C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C          EPSMXV = 0.000
C          DO 900 J= 1, LENGTH
C          CALL GETSIG (EPS1(J)), ID(J), S I G M )
C          IF( EPS1(J).GT.EPSMXV )   EPSMXV = EPS1(J)
C
900      CONTINUE
C
C...    ERROR MULTI
1000    IERR = 1
      SIGM = 0.0
      RETURN
C
C...  SAME MATERIAL OPTION
C
3000  CONTINUE
      BLENGX = 0.0
      DO 3100 J= 1, LENGTH
3100  BLENGX = BLENGX+DL(J)
      EPS1(J) = DISP(BLENGX
C
      DO 3200 J= 1, LENGTH
      CALL GETSIG (EPS1(J)), ID(J), S I G M )
3200  CONTINUE
      EPSMXV = EPS1(1)
      IERR = 0
      RETURN
END

C          SUBROUTINE SIMPLL (H1,H2, SUM )
C
C***  /PURPOSE/
C***  GET COVER LENGTH BY SIMPSON INTEGRATION
C***  /OUTPUT/
C***  SUM = COVER LENGTH
C***  /INPUT/
C***  H1 = START HEIGHT OF Y DIRECTION
C***  H2 = LAST HEIGHT OF Y DIRECTION
C
C***  PROGRAMMED BY JAERI
1990.09.28
C
C          DIMENSION LABEL(3)
C          DIMENSION X(40), Y(40),
C          NP(2,17), DATA NP/
C          +      1, 0, 2, 0, 3, 0, 4, 0, 5, 0,
C          +      5, 3, 5, 4, 4, 4, 3, 3,
C          +      3, 2, 2, 1, 1, 1, 0,
C          DATA NODE/ 17 /
C          DATA LDIR/ 4HR, 4HZ /
C          DATA XL,YL/ 3.8637, 1.0353 /
C          DATA XO,YO/ 40.0, 20.0 /
C          DATA HH/ 2.5 /
C
C          PHI1 = DASIN(H1/RSMAX)
C          PHI2 = DASIN(H2/RSMAX)
C          DPHI = (PHI2-PHI1)/20.0
C          IF( DPHI.GT.0.0 ) GO TO 50

```

## Appendix F (Continued)

## Appendix F (Continued)

```

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

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C   F1    = (DLL-XL)/DLL
C   F2    = (DLL+XL)/DLL
C   X(2)  = XVL(1) + F1*X(1)
C   Y(2)  = YVL(1) + F1*Y(1)
C   XV(6) = XV(1) + F2*X(1)
C   YV(6) = YV(1) + F2*Y(1)
C   X(1)  = X(1)/DLL
C   Y(1)  = Y(1)/DLL
C
C   XV(3) = X(2) - Y(1)*YL
C   XV(4) = X(2) + Y(1)*YL
C   YV(3) = Y(2) + X(1)*YL
C   YV(4) = Y(2) - X(1)*YL
C   CALL PLINES ( XV, YV, 5 )
C   CALL PSYM ( XV(6), YV(6), HH, LDIR(JJ), ANGXX, 1 )
C1200 CONTINUE
C...  HEADER PLOT
C
C   XX   = 10.0
C   YY   = 0.25*YO
C   CALL PSYM ( XX, YY, HH, NTITLE, 0.0D0, 70 )
C
C   XX   = 150.0
C   YY   = 0.50*YO
C   CALL PSYM ( XX, YY, HH, LABEL, 0.0D0, 12 )
C
C...  CALL PLTEOR
C...  RETURN
C...  END
+
SUBROUTINE OPLOT ( LABEL, NTITLE, XVAL, YVAL, LENGTH, KTYPE,
IFLAG, ID )
+
C*** /PURPOSE/
C*** XYPILOT PROGRAM
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 (WITH CALCULATED)
C                 4. STRAIN-STRESS PLOT
C                 5. FORCE PLOT
C
C   IFLAG      = 0, MM = 1, CM
C   ID         = MATERIAL IDENT
C
C*** PROGRAMMED BY JAERI
C*** MODIFIED BY JAERI
C*** MODIFIED BY JAERI
C*** MODIFIED BY JAERI
C
C
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION LABEL(3), NTITLE(18)
C   DIMENSION XVAL(1),
C             YVAL(1)
C   DIMENSION X(5),
C             Y(5),
C             MP(2,5),
C             KSCL(2),
C             KHEAD(3,6),
C             MHEAD(2,6),
C             MCALC(2),
C             MHEAD(2,2),
C             MH2H(2),
C             KKMCM2(2)
C
C   DATA NP/ 1, 2, 1, 3, 4, 5,
C         4, 5, 1, 6 /
C
C   DATA KHEAD / 4HDEF0, 4HRMAT, 4ION,
C                 4HENER, 4HGY, 4H,
C                 4HACCE, 4HLERA, 4HTION,
C                 4HSTRA, 4HIN, 4H,
C                 4HSTRE, 4HSS, 4H,
C                 4HFORC, 4HE, 4H /
C
C   DATA MHEAD / 4H, 4H, 4H KG-, 4HM
C                 4H G, 4H, 4H KG-, 4HM
C                 4H KG/, 4H /
C
C   DATA KSCL / 4H<10 / 4H )
C
C   DATA MLAB/ 4HSIGA, 4HSIGB, 4HSIGC,
C               4HSIGD, 4HSIGX, 4HSIG1,
C               4HSIG2, 4HSIG3, 4HP0, 4HSIG /
C
C   DATA KKMCM/ 4HMM / 4HCM
C
C   DATA MH2H/ 4HMM/M, 4HCM/C/
C
C   DATA MCALC/ 4H(CAL, 4HC.) /
C
C   DATA NPARTX,NPARTY/ 10, 8 /
C
C   DATA XL,YL/ 200,0 / 160,0 /
C
C   DATA X0,Y0,DH/ 40,0 / 30,0 / 20,0 /
C
C   DATA HH,H2,H3/ 2,5 / 2,0 / 1,5 /
C
C...  DATA RANGE
C
C   CALL GOSSAL ( LENGTH, XVAL, YVAL, XMN, YMN, XMX, YMX, FACT )
C
C   CALL XYSCAL ( XMN, XMX, NPARTX, DX, NEXPX )
C   CALL XYSCAL ( YMN, YMX, NPARTY, DY, NEXPY )
C
C...  GRID MESH
C
C   CALL PLPEN( 1 )
C   X(1) = X0
C   X(2) = X0+XL
C   X(3) = X0+XL
C   X(4) = X0
C   X(5) = X0
C   Y(1) = Y0
C   Y(2) = Y0
C   Y(3) = Y0+YL
C   Y(4) = Y0+YL
C   Y(5) = Y0
C
C   CALL PLINES ( X, Y, S )
C
C. (X) SCALE
C   Y(1) = Y0-2.0
C   Y(2) = Y0+2.0
C   Y(3) = Y0-5.0
C   Y(4) = Y0+YL-2.0
C   Y(5) = Y0+YL
C   X(1) = X0
C
C
C

```

## Appendix F (Continued)

## Appendix F (Continued)

```

X(2) = X(0)          MHEAD(1,IPX) = KKMCM1(IFLAG+1)
DNO = 0.0            MHEAD(1,IPX) = MH2H(IFLAG+1)
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, 0.000, 1 )
    X(1) = X(1)+DH
    X(2) = X(2)+DH
    DNO = DNO+DX
300 CONTINUE
    IFC(NEXPX.EQ.0) GO TO 350
    X(3) = X0+XL+4.0*H2
    CALL PSYM ( X(3), Y(3), H2, KSCL, 0.000, 6 )
    X(3) = X0+XL+7.0*H2
    Y(3) = Y(3)+H3
    CALL PNUMB ( X(3), Y(3), H3, NEXPX, 0.000 )
C. (Y) SCALE
    350 X(1) = X0-2.0
    X(2) = X0+2.0
    X(3) = X0-5.0*H2
    X(4) = X0+XL-2.0
    X(5) = X0-XL
    Y(1) = Y0
    Y(2) = Y0
    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, 0.000, 1 )
        Y(1) = Y(1)+DH
        Y(2) = Y(2)+DH
        DNO = DNO+DY
400 CONTINUE
    GO TO 500
    IFC(NEXPY.EQ.0) GO TO 500
    X(3) = X0
    Y(3) = Y0+YL+H2
    CALL PSYM ( X(3), H2, KSCL, 0.000, 6 )
    X(3) = X0+3.0*H2
    Y(3) = Y(3)+H3
    CALL PNUMB ( X(3), Y(3), H3, NEXPY, 0.000 )
C... HEADER PLOT
C
    500 CONTINUE
    XX = X0+20.0*HH
    YY = Y0+YL+3.0*HH
    CALL PSYM ( XX, YY, HH, NTITLE, 0.000, 70 )
C. (X-TITLE)
    IPX = NP(1,KTYPE)
    XX = X0+20.0*HH
    YY = 0.6*Y0
    CALL PSYM ( XX, YY, HH, KHEAD(1,IPX), 0.000, 12 )
    XX = XX+15.0*HH
C. (Y-TITLE)
    IPY = NP(2,KTYPE)
    XX = 0.4*Y0
    YY = Y0+20.0*HH
    CALL PSYM ( XX, YY, HH, KHEAD(1,IPY), 90.000, 12 )
C
    YY = YY+15.0*HH
    IFC(KTYPE.EQ.1) GO TO 510
    IFC(IPY.EQ.2) MHEAD(2,IPY) = KKMCM1(IFLAG+1)
    IFC(IPY.EQ.5) MHEAD(2,IPY) = KKMCM2(IFLAG+1)
510 CONTINUE
    CALL PSYM ( XX, YY, HH, MHEAD(1,IPY), 90.000, 8 )
C. (ANALYSIS LABEL)
    XX = X0+160.0
    YY = 0.6*Y0
    CALL PSYM ( XX, YY, HH, LABEL, 0.000, 12 )
C. (MATERIAL ID)
    XX = XX+13.0*HH
    CALL PSYM ( XX, YY, HH, MLAB(ID), 0.000, 4 )
    IFC(KTYPE.EQ.3) GO TO 600
    XX = X0+5.0*HH
    CALL PSYM ( XX, YY, HH, MCALC, 0.000, 7 )
C
    C... XYPILOT
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) = X0
    Y(1) = Y0
    R = 1.0
    DO 700 J= 1, LENGTH
        X(2) = XFACT*VAL(J) + X0
        Y(2) = YFACT*VAL(J) + Y0
        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
C

```

## Appendix F (Continued)

### Appendix F (Continued)

```

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
C   IMPLICIT REAL*8 (A-H,O-Z)
COMMON /MODEES/
      RCOORD(5), ZCOORD(9), WEIGHT,
      DISP, FACT(5), ENERGY,
      MTOMN, KOUT, NPART,
      COMMON /MODEL1/, S,
      DIMENSION LABEL(3),
      NP(2,17), KKMCM(2),
      KDEPTH(2),
      DATA NP / 1, 0, 2, 0, 3, 0, 4, 0, 5, 0,
      +           5, 2, 5, 3, 5, 4, 4, 4, 3, 3,
      +           3, 2, 2, 1, 2, 1, 1, 1, 0 /
      DATA NODE / 17, 1, 2, 2, 1, 1, 1, 0 /
      DATA LDIR / 4.HR, 4.HZ, /
      DATA KDEPTH/ 4.HDEPTH, 4.HHH= /
      DATA KACCEL/ 4.HACCEL, 4.HL= /
      DATA KKMCM/ 4.H(CMM), 4.H(CM) /
      DATA KKACCG/ 4.H(G), /
      DATA XL,YL/ 3.863/, 1.0353 /
      DATA XD,YO/ 40.0, 20.0 /
      DATA HH/ 2.5 /
      DATA H2/ 2.0 /
C...  FUNCTION
      SQRT(Z) = DSQRT(Z)
C...  CLEAR (X,Y)
      DO 100 J= 1, 35
      X(J) = 0.0
      100 Y(J) = 0.0
C...  GET GEOMETRY
      DO 200 J= 1, NODE
      IX = NP(1,J)
      IY = NP(2,J)
      X(J) = RCOORD(IX)
      IF (IY.EQ.0) GO TO 200
      Y(J) = ZCOORD(IY)
      200 CONTINUE
      NN = NODE
      ANGXX = 0.0
C
      IF ( MODEL(NE,2) ) GO TO 400
      IF ( ZCOORD(5).LT.ZCOORD(4) ) GO TO 310
      NN = 2*NODE
      DO 300 J= 1, NODE
      IX = NP(1,J)
      IY = NP(2,J)
      X(J+NODE) = RCOORD(IY)
      Y(J+NODE) = ZCOORD(9-IY)
      300 CONTINUE
      GO TO 330
      310 ZWIDE = 3.0*ZCOORD(4)
      DO 320 J= 1, NODE
      X(J+NODE) = X(J)
      Y(J+NODE) = ZWIDE-Y(J)
      320 CONTINUE
      C
      MM = NN+4
      X(NN+1) = 0.0
      X(NN+2) = 0.5*RCODR(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*RCODR(1)
      DO 330 J= 1, MM
      R = X(J)
      Z = Y(J)
      X(J) = Z
      Y(J) = -R
      330 CONTINUE
      GO TO 500
      C...  CORNER MODEL
      C
      MM = NN+4
      X(NN+1) = 0.0
      X(NN+2) = 0.5*RCODR(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) = 0.0
      DO 350 J= 1, MM
      R = X(J)
      Z = Y(J)
      X(J) = Z
      Y(J) = -R
      350 CONTINUE
      GO TO 500
      C...  CORNER MODEL
      C
      MM = NN+4
      X(NN+1) = 0.0
      X(NN+2) = 0.5*RCODR(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) = 0.0
      DO 450 J= 1, MM
      R = X(J)
      Z = Y(J)
      X(J) = Z
      Y(J) = -R
      450 CONTINUE
      C...  GEOMETRY SCALING
      C
      500 CONTINUE

```

## Appendix F (Continued)

## Appendix F (Continued)

```

CALL GDSCAL ( MM, X, Y, XMN, YMN, XMX, YMX, FACTOR )
C... PLDT COORDINATES
C
C DO 600 J= 1, MM
X(J) = FACTOR*(X(J)-XMN) + X0
Y(J) = FACTOR*(Y(J)-YMN) + Y0
600 CONTINUE
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... 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 = SQR((X(1)**2+(Y(1)**2))
IF( DLL .LE. 1.0D-2 ) DLL = 1.0D-2
F1 = (OLL-XL)/DLL
F2 = (OLL-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... 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) = Y0
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 = Y(1)-5*I2
CALL PSYM ( XX, YY, H2, KDEPTH, 0.000, 7 )
XX = XX+7.0*H2
CALL PNUMB ( XX, YY, H2, DEPTH, 0.000, 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, KKMC(MFLAG+1), 0.000, 4 )
C
XX = 120.0
YY = YO+H2* FACTOR*(YMX-YMN)
CALL PSYM ( XX, YY, H2, KACCEL, 0.000, 7 )
XX = XX-7.0*H2
CALL PNUMB ( XX, YY, H2, ACCEL, 0.000, 2 )
XX = XX-7.0*H2
CALL PSYM ( XX, YY, H2, KKACCG, 0.000, 3 )
C... HEADER PLOT
C
XX = 10.0
YY = 0.25*YO
CALL PSYM ( XX, YY, HH, NTITLE, 0.000, 70 )
C
XX = 150.0
YY = 0.5*YO
CALL PSYM ( XX, YY, HH, LABEL, 0.000, 12 )
C... CALL PLTEOR
RETURN
END

```

## Appendix F (Continued)

## Appendix F (Continued)

```

C      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 COORDINATES
C          Y    = Y COORDINATES
C
C      *** PROGRAMMED BY JAERI           1988.02.01
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION DVAL(8)
C      DATA DVAL/ 0.5, 1.0, 1.5, 2.0, 2.5, 5.0, 10.0, 15.0 /
C
C      FUNCTION ALOGIO(Z) = DLOG10(Z)
C
C      *** DPART = NPART
C          XMIN = 0.0
C          DS   = (YMAX-YMIN)/DPART
C          NEXP = ALOGIO(DS)
C          IF( NEXP.LT.0 ) NEXP = NEXP-1
C          GVAL = 10.0**NEXP
C          HWDI = DS/GVAL
C          DO 100 J= 1, 7
C              IF( HWIDE.LE.DVAL(J) ) GO TO 200
C 100 CONTINUE
C          J = 7
C 200 DSIZE = DVAL(J)
C
C      RETURN
C      END

C      XS   = XMX-XMN
C      YS   = YMX-YMN
C      AS   = AMAX1 (XS,YS)
C      SCALEG = PAPER/AS
C      RETURN
C      END

C      XS   = XMX-XMN
C      YS   = YMX-YMN
C      AS   = AMAX1 (XS,YS)
C      SCALEG = PAPER/AS
C      RETURN
C      END

C      SUBROUTINE PLPEN (NO)
C          IMPLICIT REAL*8 (A-H,O-Z)
C          CALL PLOTS (0.0D0,0.0D0,10)
C          RETURN
C      END

```

## Appendix F (Continued)

## Appendix F (Continued)

```

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

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

SUBROUTINE PLTEND
IMPLICIT REAL*8 (A-H,O-Z)
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

SUBROUTINE PLMARK ( X0,Y0,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,
+      -0.92388, -0.38268,
+      0.38268, -0.92388,
+      0.92388, 0.38268,
C DO 10 J=1,9
X(J) =R*CR(1,J) + X0
10 Y(J) =R*CR(2,J) + Y0
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 )
IMPLICIT REAL*8 (A-H,O-Z)
CALL NUMBER (X,Y,H,AN,ANG)
RETURN
END

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

```