JAERI-Data/Code
97-036

# PUNCTURE:
# A COMPUTER PROGRAM FOR PUNCTURE ANALYSIS OF RADIOACTIVE MATERIAL TRANSPORT CASKS

September 1997

Takeshi IKUSHIMA

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

編集兼発行　日本原子力研究所
印　　　刷　いばらき印刷㈱

# PUNCTURE: A Computer Program for Puncture 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 July 31, 1997)

In the drop puncture analyses for radioactive transport casks, it has become possible to perform them in detail by using interaction evaluation computer programs such as DYNA3D. However, the considerable cost and the computer time are necessitated to perform analyses by these programs. To decrease the computer cost and time, a simplified computer program PUNCTURE has been developed. The PUNCTURE is a static calculation computer program based on the Onat's thory and Asada's research. The PUNCTURE is capable of evaluating the acceleration of cask bodies, the deformation of punctured plates and, the stress and the deformation of puncture bars.

Main features of the computer program are as follows;

(1) three analysis models for punctured plate are used that are the fixed supported bending plate model, the simply supported bending plate model and the fixed supported membrane plate model,

(2) it is capable of graphical representations for calculation results and

(3) not only main frame computers (OS MSP) but also work stations (OS: UNIX) and personal computers (OS: Windows) are available for use of PUNCTURE.

In the paper, brief illustration of calculation method using the Onat's thory and Asada's research is presented. The second section presents comparisons between calculation and experimental results. The third section provides a user's guide for PUNCTURE.

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

JAERI-Data/Code 97-036

# PUNCTURE: 放射性物質輸送容器の貫通計算プログラム

日本原子力研究所東海研究所燃料サイクル安全工学部
幾島　毅

（1997年７月31日受理）

放射性物質輸送容器の落下貫通解析では、DYNA3Dのような詳細計算プログラムを使用しているが、多くの計算費用と計算時間が必要である。費用と時間を少なくするために簡易計算プログラムPUNCTUREを開発した。PUNCTUREはOnatの理論と浅田の研究に基づく円板の静的弾塑性解析法によっている。PUNCTUREでは容器の加速度、貫通板の変形および貫通棒の応力と変形を計算できる。

PUNCTUREの主要な特徴は次の通りである。

⑴　貫通板の構造解析では、完全固定円板曲げモデル、単純支持円板曲げモデルおよび完全固定円板膜モデルの３種類のを取り扱うことができる。

⑵　計算結果は図形表示することができる。

⑶　大型計算機（OS:MSP）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）によっても使用できる。

本報告書はPUNCTUREの計算手法、PUNCTUREによる計算結果と実験結果との比較およびPUNCTURE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記述した。

---

東海研究所：〒319－11　茨城県那珂郡東海村白方白根２－４

# Contents

# 目　　次

# 1 . Introduction

Radioactive material transport casks (IAEA regulatory standard type B pckages, type A fissile packages and so on) are required to maintain integrity against the puncture test where it is dropped onto a 150 mm diameter mild steel bar from a height of one meter.

In the drop puncture analyses for radioactive transport casks, it has become possible to perform them in detail calculation bu using interaction evaluation computer programs, DYNA2D(1), DYNA3D(2), NIKE2D(3), NIKE3D(4), PISCES(5) and HONDO(6). However, the considerable cost and the computer time are necessitated to perform analyse by these programs. To decrease the cost and the time, a simplified computer program PUNCTURE shown in Fig. 1.1 has been developed(7),(8). The PUNCTURE is a static calculation computer program based on the Onat's theory(9) and Asada's research(10). The PUNCTURE is capable of evaluating the acceleration of cask bodies, the deformation of puncture plates and the stress and the deformation of puncture bars.

Main features of the computer program PUNCTURE are as follows:

(1) three analysis models for punctured plates are used. Those are the fixed supported bending plate model, the simply supported bending plate model and the fixed supported membrane plate model,

(2) it is capable of graphical representations for calculation results and

(3) not only main frame computers (OS MSP) but also work stations (OS UNIX) and personal computers (OS Windows) are available for use of PUNCTURE.

In the first section of the paper, a brief illustration of calculation method using Onat's theory and Asada's research is presented. The second section presents comparison between calculation and experimental results. The third section provides a user's guide for PUNCTURE.

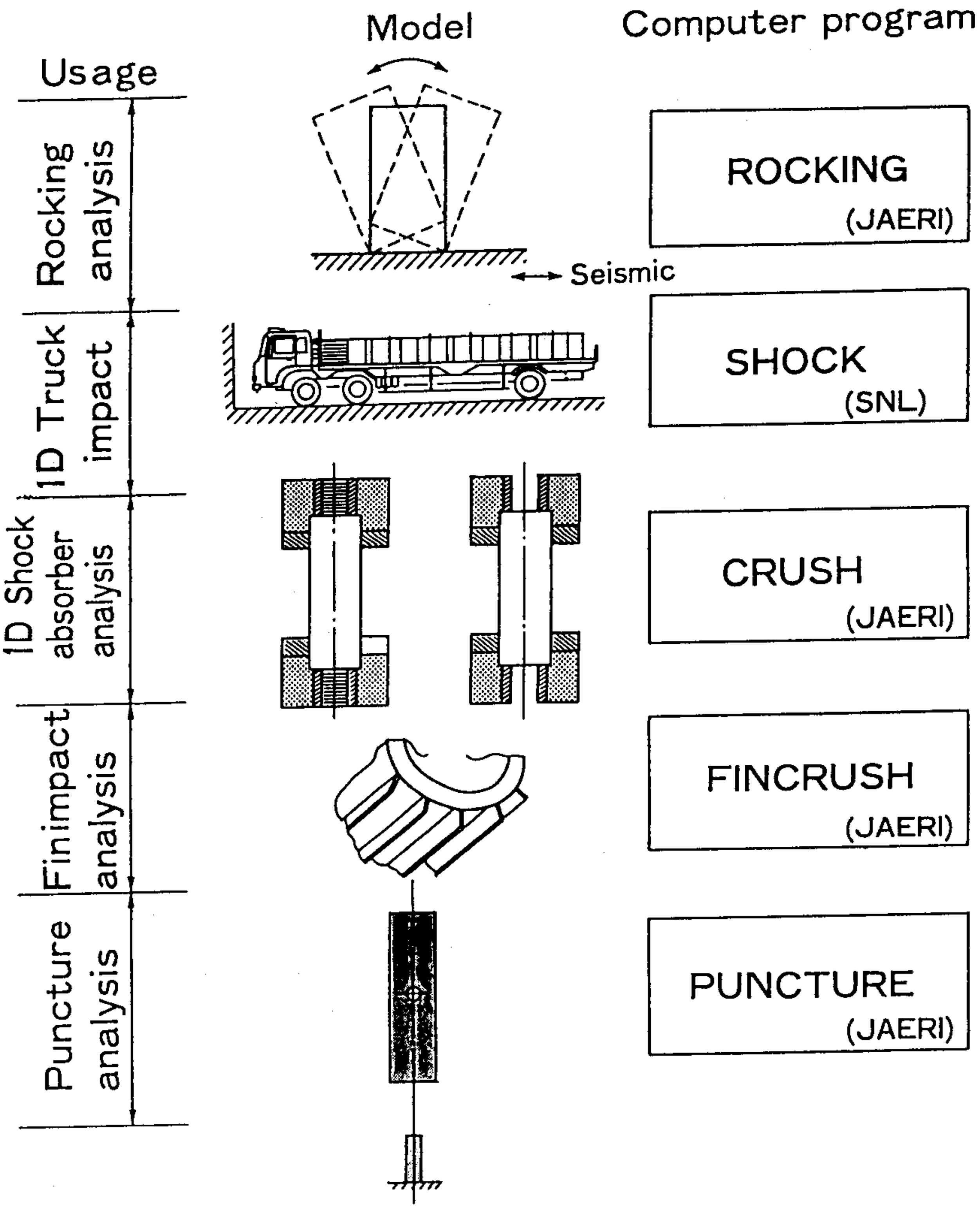


Fig. 1.1 Simplified analysis computer programs

# 2.Calculation Equation

## 2.1 Calculation Model

In the case of the puncture does not occur when the cask is dropped onto a mild steel bar in a puncture test. In the modeling of a puncture analysis as shown in Fig. 2.1, it is assumed that the kinematic energy of the cask is absorbed into deformations of both the cask body and the mild steel bar, provided the puncture does not occur. The deformation generated on the cask body can be evaluated based on the fact that loads generated on the cask and the mild steel bar are equal to each other. In this evaluation, the plastic theory of bending of a circular plate having a multi-layer construction(e.g. steel-lead-steel three layers) is used, which has been developed by extending Onat's theory and Asada's research.

## 2.2 Energy Balance

In the modeling of a puncture analysis program PUNCTURE, it is assumed plastic enegy of a cask and a pucture bar is balanced with impact energy. That is

$$E_v = W H_0 , \qquad (2.1)$$

where

$E_v$ : impact energy,

$W$ : weight of impacting body,

$H_0$ : drop height of impacting body.

In the theory, elasticity of the material usually is neglected and the load-carrying capacity is estimated as the load at which a model composed of an ideal rigid-plastic material would begin to deform. It can be shown that, if the material is perfectly plastic(i.e.,nonstrain hardening) and if the accompanying change in geometry is disregarded, plastic flow continues under constant load.

The load-deflection relationship of the mild steel bar is shown in the following equation.

$$P_p = f(\delta_p),\qquad (2.2)$$

where

$P_p$ : load act on steel bar,

$\delta_p$ : plastic deformation of steel bar.

On the other hand, the load-deflection relationship of the cask body is shown in the following equation:

$$P_c = h(\delta_c),\qquad (2.3)$$

where

$P_c$ : load act on cask body,

$\delta_c$ : plastic deformation of cask body.

In the case of the cask whose weight is W is dropped from a height H onto the mild steel bar, the following equations are given from the conservation law of energy:

$$W H_0 = \int_0^{\delta_p} f(\delta_p)d\delta_p + \int_0^{\delta_c} h(\delta_c)d\delta_c\ ,\qquad (2.4)$$

$$f(\delta_p) = h(\delta_c)\ .\qquad (2.5)$$

The acceleration of the cask body is determined by the following equation:

$$\alpha = h(\delta_c)/(W\ /g),\qquad (2.6)$$

where

$\alpha$ : acceleration of cask body,

$g$ : gravity constsnt.

### 2.3 Load Deflection Relationship

(1) Fixed supported bending plate model

When a distributed load is applied to a three-layer circular rigidly clamped at its edges as shown in Figure 2.2, the relationship between the displacement and three load can be given by

$$P^* = 1 + \alpha_1 U + \alpha_2 U^2 \ ; \quad (U \leqq U^*) \ , \tag{2.7}$$

$$P^* = \beta_1 + \beta_2 U + \beta_3 / U \ ; \quad (U \geqq U^*) \ , \tag{2.8}$$

where

$P^* = P \,/\, P_L$ ,

$P^*$ : nondimensional load,

P : load,

$P_L$ : limit load,

$U = \delta \,/\, t^*$ ,

U : nondimensional deflection,

$\delta$ : center deflection,

$t^*$ : equivalent plate thickness,

$$U^* = (\,1 + \ln\,(R \,/\, \rho)\,)\,/\,2 \ , \tag{2.9}$$

R : radius of plate,

$\rho$ : discontinuity radius of velocity curvature,

a : radius of loaded area,

and the factors $\alpha_1$, $\alpha_2$, $\beta_1$, $\beta_2$ and $\beta_3$ are given by

$$\alpha_1 = \frac{\left(1 + 2\ \ln\dfrac{R}{\rho}\right)}{\left(2 + \ln\dfrac{R}{\rho}\right)\left(1 + \ln\dfrac{R}{\rho}\right)} \ , \tag{2.10a}$$

$$\alpha_2 = \frac{2\left(1 + 3\ \ln\dfrac{R}{\rho}\right)}{3\left(2 + \ln\dfrac{R}{\rho}\right)\left(1 + \ln\dfrac{R}{\rho}\right)^2} \ , \tag{2.10b}$$

$$\beta_1 = \frac{\left(3 + \ln\frac{R}{\rho}\right)}{2\left(2 + \ln\frac{R}{\rho}\right)} , \tag{2.10c}$$

$$\beta_2 = \frac{2\left(1 + 2\ln\frac{R}{\rho}\right)}{\left(2 + \ln\frac{R}{\rho}\right)\left(1 + \ln\frac{R}{\rho}\right)} , \tag{2.10d}$$

$$\beta_3 = \frac{\left(1 + \ln\frac{R}{\rho}\right)}{12\left(2 + \ln\frac{R}{\rho}\right)} . \tag{2.10e}$$

The discontinuity radius of the velocity $\rho$ curveture is determined by the following equations:

$$1 - \frac{2}{3}\frac{a}{\rho}\left(1 + \ln\frac{R}{\rho}\right) = 0 \ ; \qquad ( a / R \leqq 0.606 ), \tag{2.11a}$$

$$1 - \left(\frac{2}{\rho}\right)^2\left(1 + 2\ln\frac{R}{\rho}\right) + \frac{2}{3}\left(1 + \ln\frac{R}{\rho}\right) = 0 \ ; \ (a/R \geqq 0.606), \tag{2.11b}$$

The limit load $P_L$ is given by

$$P_L = 2\pi M_p \frac{A}{B} , \tag{2.12}$$

where

$$A = 2 + \ln\frac{R}{\rho} , \tag{2.13a}$$

$$B = 1 + \ln\frac{R}{\rho} - \frac{2\,a}{3\,\rho} \quad ; \ (\,\rho \geqq a\,)\ , \tag{2.13b}$$

$$B = \frac{1}{2} + \ln\frac{R}{a} - \frac{\rho^2}{6\,a^2} \quad ; \ (\,\rho \leqq a\,), \tag{2.13c}$$

$$M_p = \frac{(t*)^2}{\rho}\sigma* \ , \tag{2.13d}$$

where

$M_p$ : sectional yield moment,

$\sigma*$ : equivalent stress.

(2) Simply supported plate bending model

In the case of the simply supported plate bending model as shown in Fig. 2.3, the equivalent load are given by

$$P* = 1 + \frac{4}{3}U^2 \ ;(\,U \leqq 0.5\,)\ , \tag{2.14a}$$

$$P* = 2\,U + \frac{1}{6\,U} \ ; \ (\,U \geqq 0.5\,). \tag{2.14b}$$

The limit load $P_L$ is given by

$$P_L = \frac{2\,\pi\,Mp}{\left(1 - \dfrac{2\,a}{3}\right)} \quad . \tag{2.15}$$

(3) Fixed supported membrane plate model

In the case of the fixed supported membrane plate model as shown in

Fig. 2.4, the load acting on the plate are given by the following:

$$P = \frac{2\pi N_p \delta}{\frac{1}{2} + \ln\frac{R}{a}}, \tag{2.16}$$

where Np is the sectional yielding load of the plate,

$$N_p = \sigma^* t^*, \tag{2.17}$$

and $t^*$ is a equvalent thickness of the plate.

## 2.4 Equivalent Thichness and Equivalent Stress

In converting a three-layer plate into an equivalent single plate, the quivalent plate thicknss $t^*$ and the equivalent stress $\sigma^*$ are relateed by the following equations:

$$t^* = \frac{4 M_p}{N_p}, \tag{2.18}$$

$$\sigma^* = \frac{N_p}{t^*}, \tag{2.19}$$

where the sectional yield load $N_p$ is given by the following equations:

$$N_p = (\sigma_1)_T\ t_1 + (\sigma_2)_T\ t_2 + (\sigma_3)_T\ t_3, \tag{2.20}$$

where

$t_1$ : thickness of outer plate,
$t_2$ : thickness of intermediate plate,
$t_3$ : thickness of inner plate,
$\sigma_1$ : stress of outer plate,
$\sigma_2$ : stress of intermediate plate,
$\sigma_3$ : stress of inner plate,

$(\ )_C$ : denotes compression,

$(\ )_T$ : denotes tension.

According to the position of the neutral axis of stress, there are three kinds of stress states as shown in Fig. 2.5. Those are named stress states A, B and C. In the case of the stress state A, the neutral axis exists in the outer layer, the stress state B in the intermediate and the stress state C in the inner.

(1) Stress state A

The equilibrium of stress in the three layer plates leads the following equations (see Fig.2.5a):

$$(\sigma_1)_C\, t = (\sigma_1)_T (t_1 - t) + (\sigma_2)_T t_2 + (\sigma_3)_T t_3 \ , \tag{2.21}$$

where $t$ is a distance from outer surface to neutral axis.

$$t = \frac{(\sigma_1)_T\, t_1 + (\sigma_2)_T\, t_2 + (\sigma_3)_T\, t_3}{(\sigma_1)_C + (\sigma_1)_T} \ . \tag{2.22}$$

The sectional yield moment is given by the following equation:

$$M_p = \frac{(t)^2}{2}(\sigma_1)_C + \frac{1}{2}(t_1 - t)^2(\sigma_1)_T + \left(t_2\left(t_1 - t + \frac{t_2}{2}\right)\right)(\sigma_2)_T$$

$$+ \left(t_3\left(t_1 - t + t_2 + \frac{t_3}{2}\right)\right)(t_3)_T \ . \tag{2.23}$$

The stress state A is established by the following condition:

$$(\sigma_1)_C\, t_1 \geqq (\sigma_2)_T\, t_2 + (\sigma_3)_T\, t_3 \ . \tag{2.24}$$

(2) Stress state B

The equilibrium of stress in the three layer plates leads the following equations (see Fig.2.5b):

$$(\sigma_1)_c\ t_1 + (\sigma_2)_c\ (t - t_1) = (\sigma_2)_T\ (t_1 + t_2 - t) + (\sigma_3)\ t_3\ , \tag{2.25}$$

$$t = \frac{-\ (\sigma_1)_c\ t_1 + (\sigma_2)_c\ t_1 + (\sigma_2)_T\ (t_1 - t_2) + (\sigma_3)_T\ t_3}{(\sigma_2)_c + (\sigma_2)_T}\ , \tag{2.26}$$

$$\begin{aligned} M_p = {} & t_1\left(t - \frac{t_1}{2}\right)(\sigma_1)_c + \frac{1}{2}(t - t_1)^2\,(\sigma_2)_c \\ & + \frac{1}{2}(t_1 + t_2 - t)^2\,(\sigma_2)_T + t_3\left(t_1 + t_2 - t + \frac{t_3}{2}\right)(\sigma_3)_T\ , \end{aligned} \tag{2.27}$$

$$(\sigma_1)_c\ t_1 + (\sigma_2)_c\ t_2 \geqq (\sigma_3)_T\ t_3\ . \tag{2.28}$$

(3) Stress state C

The equilibrium of stress in the three layer plates leads the following equations (see Fig.2.5c):

$$(\sigma_1)_c\ t_1 + (\sigma_2)_c\ t_2 + (\sigma_3)_c\ (t - t_1 - t_2) = (\sigma_3)_T\ (t_1 + t_2 + t_3 - t)\ , \tag{2.29}$$

$$t = \frac{-\ (\sigma_1)_c\ t_1 - (\sigma_2)_c\ t_2 + (\sigma_3)_c\ (t_1 + t_2) + (\sigma_3)_T\ (t_1 + t_2 + t_3)}{(\sigma_3)_c + (\sigma_3)_T}\ , \tag{2.30}$$

$$\begin{aligned} M_p = {} & t_1\left(t - \frac{t_1}{2}\right)(\sigma_1)_c + t_2\left(t - t_1 - \frac{t_2}{2}\right)(\sigma_2)_c \\ & + \frac{1}{2}(t - t_1 - t_2)^2(\sigma_3)_c + \frac{1}{2}(t_1 + t_2 + t_3 - t)^2(\sigma_3)_c\ , \end{aligned} \tag{2.31}$$

$$(\sigma_1)_c\ t_1 + (\sigma_2)_c\ t_2 \leqq (\sigma_3)_T\ t_3\ . \tag{2.32}$$

## 2.5 Condition of Plate Puncture

The condition of the three layer plate puncture is given by the following equation:

$$P \geqq \pi\, d\ (\tau_1\ t_1 + \tau_2\ t_2 + \tau_3\ t_3)\ , \tag{2.33}$$

where

P : load,

$\tau_1$ : failure shear stress of outer plate,

$\tau_2$ : failure shear stress of intermediate plate,

$\tau_3$ : failure shear stress of inner plate,

$$\tau_1 = 0.6(\sigma_1)_u \ , \tag{2.34a}$$

$$\tau_2 = 0.6(\sigma_2)_u \ , \tag{2.34b}$$

$$\tau_3 = 0.6(\sigma_3)_u \ , \tag{2.34c}$$

$(\sigma_1)_u$ : ultimate tensile strenght of outer plate,

$(\sigma_2)_u$ : ultimate tensile strenght of intermediate plate,

$(\sigma_3)_u$ : ultimate tensile strength of inner plate.

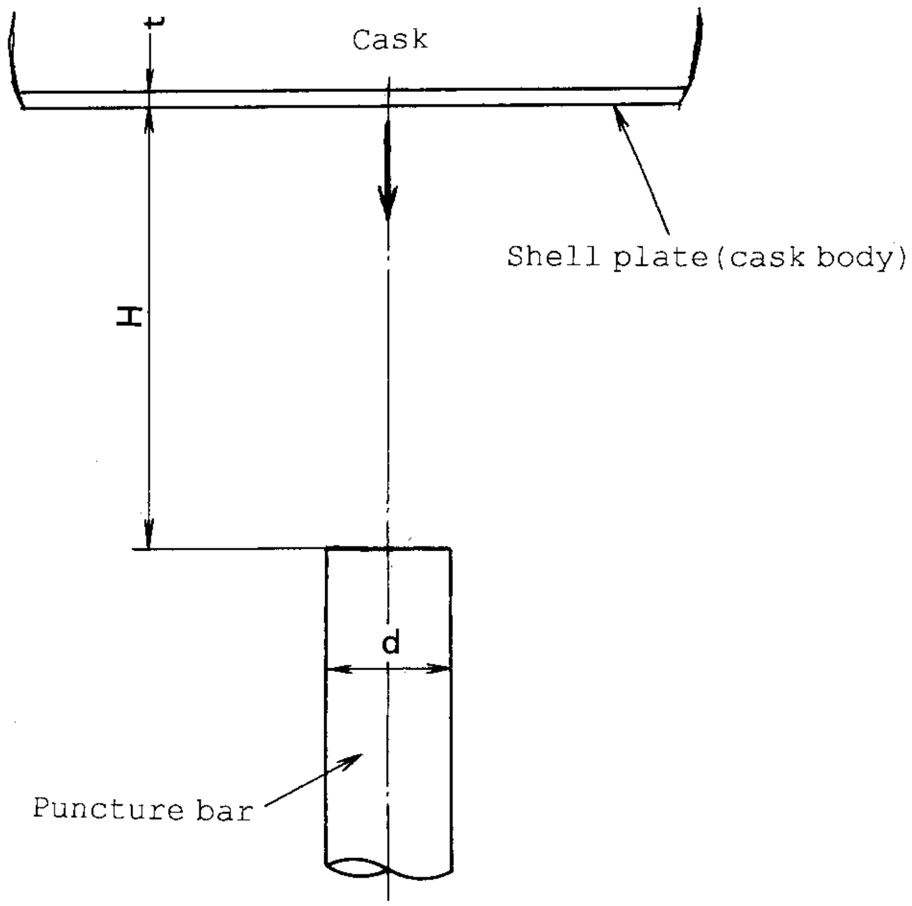
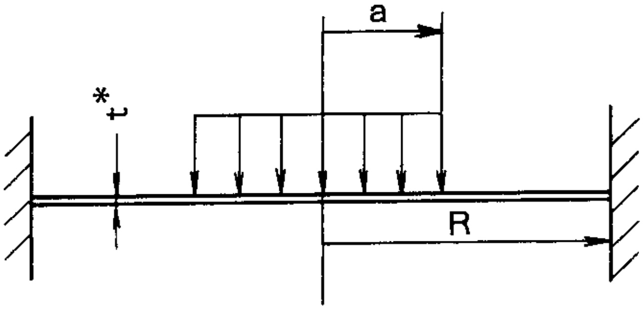
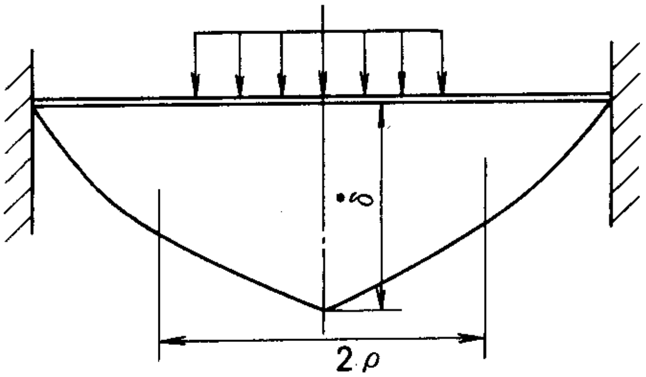


Fig. 2.1 Puncture analysis model

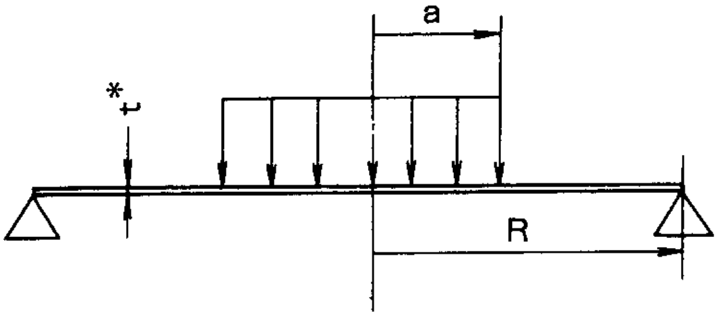


(a) Load distribution

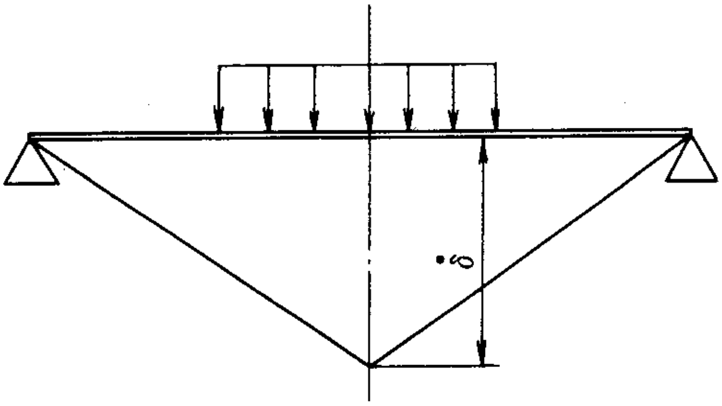


(b) Velocity distribution

Fig. 2.2 Fixed supported bending plate with distributed load

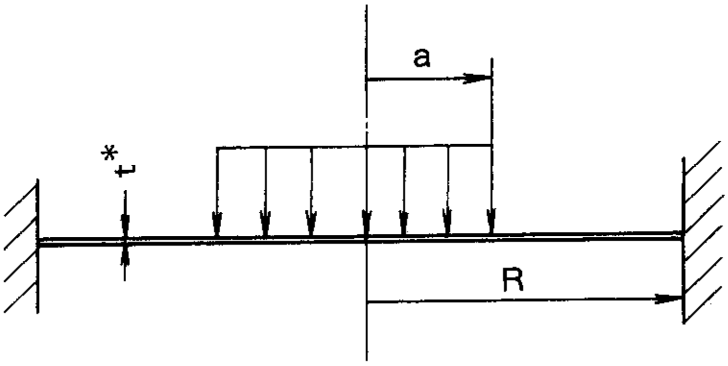


(a) Load distribution

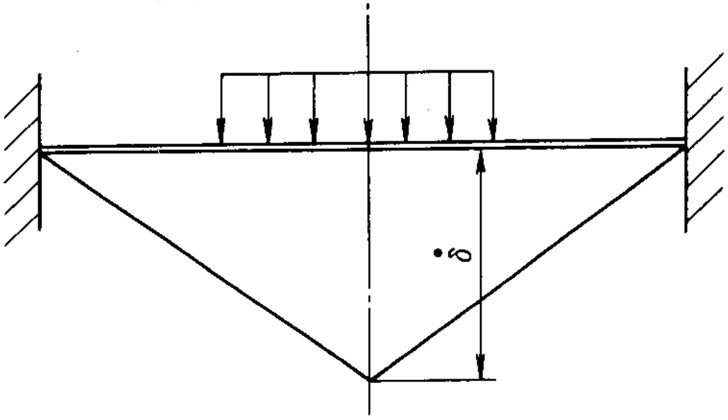


(b) Velocity distribution

Fig. 2.3 Simply supported bending plate with distributed load



(a) Load distribution



(b) Velocity distribution

Fig. 2.4 Fixed supported membrane plate with distributed load

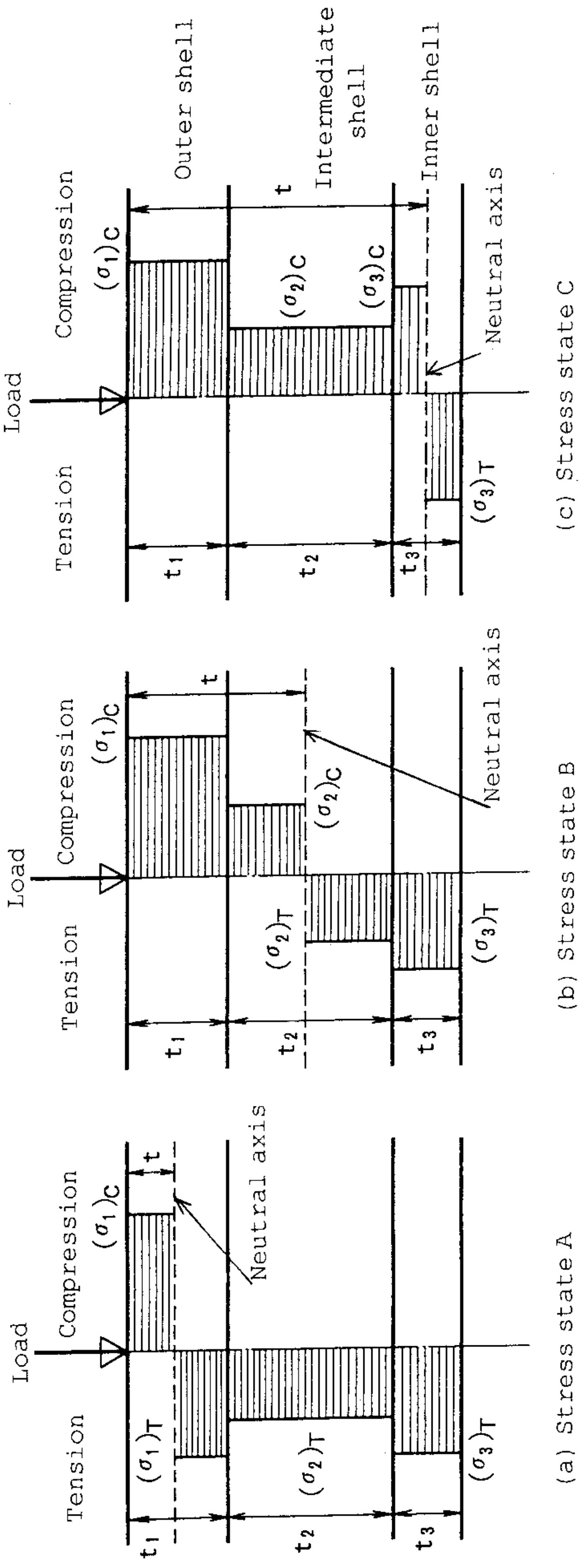


Fig. 2.5 Three kinds of stress states in three layer shells

## 3. Benchmark Calculation

In order to demonstrate the adequacy of the simplified computer program PUNCTURE, the benchmark calculation using experimental results of the LLNL has been performed. Table 3.1 shows the LLNL experimental data.

Table 3.2 and Fig 3.1 shows comparisons between **the LLNL**[11] experiments and the PUNCTURE results. According to Fig. 3.1, results by the computer program PUNCTURE agree with the experimental results performed on multilayer plates. The PUNCTURE program are practicably estimate the actual data to be obtained by the puncture test.

Table 3.1 LLNL experimental data

| Item | Diameter (mm) | Thickness (mm) | Length (mm) | Material |
|---|---|---|---|---|
| Outer circular plate | 203.2 | 5.08 | | **SUS304** |
| Inner circular plate | | 15.24 | | Lead |
| Puncture bar | 15.24 | | 20.32 | Steel |

Table 3.2 Comparison between LLNL experiments and PUNCTURE

| Item | | Limit load ($10^4$ kgf) |
|---|---|---|
| LLNL experinents | Static test | 1.15 |
| | Dynamic test | 1.21 |
| **PUNCTURE** | | 10.4 |

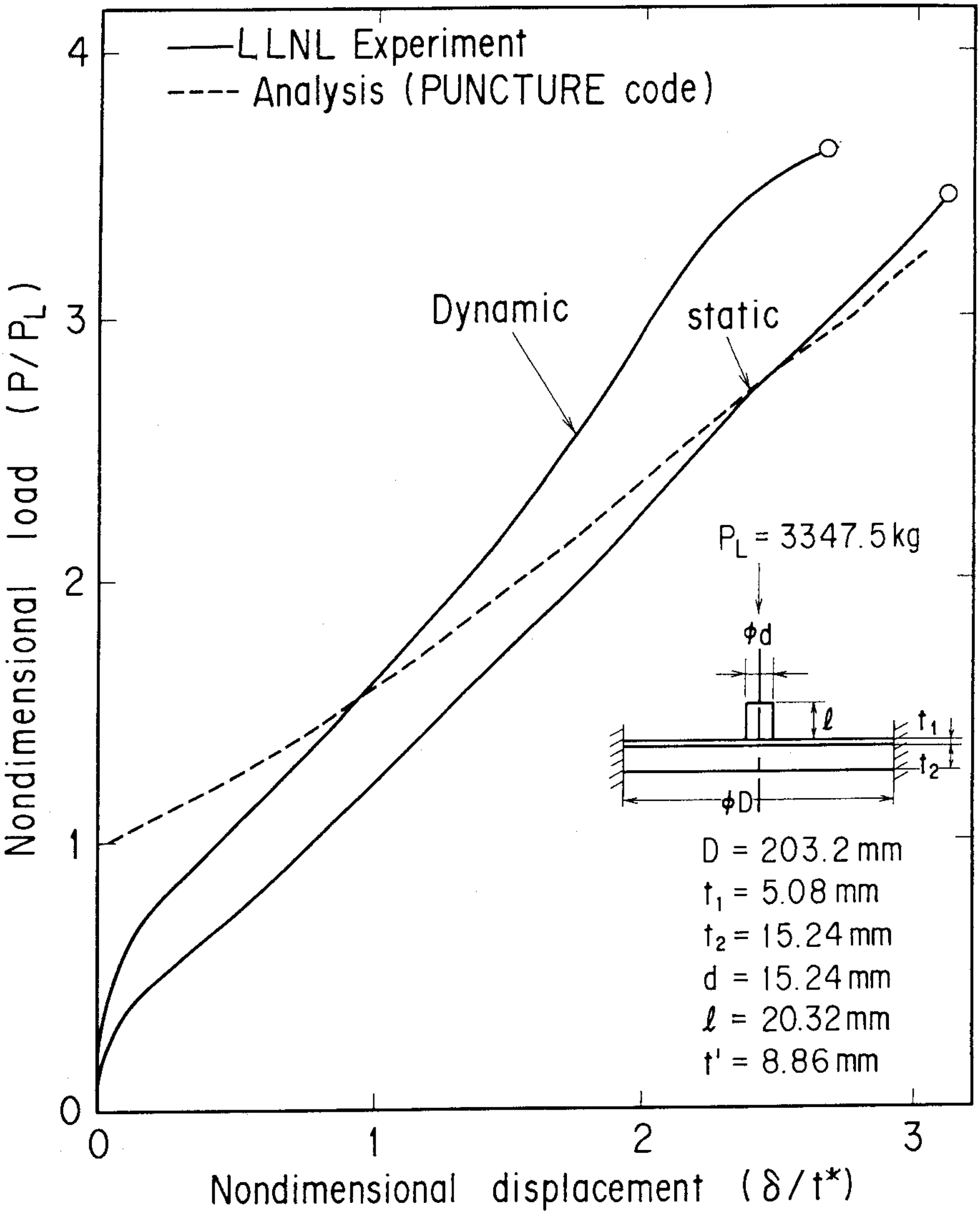


Fig. 3.1 Load vs. displacement

# 4. Computer Program

## 4.1 Program Description

The computer program PUNCTURE is static calculation program capable of evaluating the maximum acceleration、the maximum load and the maximum deformation of cask using Onat's theory and Asada's research.

The computer program PUNCTURE consists of a main routine and nine subroutines that are MAIN, CARDIN, SOLV1, SOLV2, SOLV3, GETEQV, FITSTP, NEWTON, TRGET and DPLOT. The eleven basic subroutines for plotting are GOSCAL, XYSCAL, PLTBGN, PLPEN, PLINES, PLMARK, PSYM, PNUMB, PNUMBR, PLTOR and PLREND. Overall structure of PUNCTURE is shown in Fig. 4.1. Functions of subroutines are as follows:

MAIN : initiarizes the start of run,

CARDIN : reads input data,

SOLV1 : computes load, deformation and stress of punctured plates and puncture bar using fixed supported bending plate model,

SOLV2 : computes load, deformation and stress of punctured plates and puncture bar using simply supported bending plate model,

SOLV3 : computes load, deformation and stress of punctured plates and puncture bar using fixed supported membrane plate model,

GETEQV : computes equivalent thickness of punctured plates,

FITSTP : computes relationship of force and deformation of last step,

NEWTON : computes discontinuity radius of velocity curveture using the Newton Raphson method,

TRGET : computes stress and strain of puncture bar,

DPLOT : X-Y curve plot,

The plot basic subroutines are as follows:

GOSCAL : scaling of geometry plot,

XYSCAL : scaling of X-Y plot,

PLTBGN : initiatization of plotter,

PLPEN : chage plotter pen size,
PLINES : draw line,
PLMARK : plot round mark,
PSYM : write letter,
PNUMB : write number,
PNUMBR : write number,
PLTEOR : plot new figure,
PLTEND : plot end.

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

### 4.2 Description of Input Data

This section describes the input data required by PUNCTURE. The input data consists of the job description, the analysis type such as drop attitude, geometry, the cask weight, the initial condition, the boundary condition constants, the geometry unit selection and options for output plotting. The input instruction is simple and easy 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 PUNCTURE. 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 PUNCTURE.

(2) Calculation data

The input data, the deformations, the forces, the energys and the accelerations are printed at every interval steps.

(3) Graphical output

PUNCTURE provides users with graphical output of the deformations, the relationship between the accelerations and the deformations, the dissipation energys and the deformations, the accelerations and the dissipation energys and so forth.

## Table 4.1 Input data for PUNCTURE

| Columns | Format | Variables | Descriptions |
|---|---|---|---|
| Data set No. 1:Job description. | | | |
| 1 - 5 | A5 | NAME | Flag for job description.<br>'TITLE'. |
| 6 - 10 | 5X | — | Blank, |
| 11 - 80 | 70A1 | NTITLE | Job description. |
| Data set No. 2:Calculation model data. | | | |
| 1 - 8 | A8 | NAME | Flag for calculation model.<br>'FIXED-B ':fixed supported bending plate model.<br>'FIXED-M ':fixed supported membrance plate model.<br>'SIMPLE-B':simply supported bending plate model. |
| 9 - 10 | 2X | — | Blank. |
| 11 - 20 | F10.0 | ROUT | Radius of circular plate(mm). |
| 21 - 30 | F10.0 | AA | Radius of puncture bar(mm). |
| 31 - 40 | F10.0 | TLENG | Length of puncture bar(mm). |
| 41 - 50 | F10.0 | THICK(1) | Outer layer thickness of circular plate(mm). |
| 51 - 60 | F10.0 | THICK(2) | Intermediate layer thickness of circular plate(mm). |
| 61 - 70 | F10.0 | THICK(3) | Inner layer thickness of circular plate(mm). |
| Data set No. 3:Geometry data. | | | |
| 1 - 8 | 2A4 | NAME | Flag for calculation.<br>'ANALYSIS':calculation start. |
| 9 - 10 | 2X | — | Blank. |
| 11 - 20 | F10.0 | WEIGHT | Weight of cask(kg). |
| 21 - 30 | F10.0 | HEGHT | Height of cask(mm). |

## Table 4.1 (Continued)

| Columns | Format | Variables | Descriptions |
|---|---|---|---|
| 31 - 40 | F10.0 | DISP | Deformation increment for calculation(mm). If DISP is blank or 0.0, DISP is equal to 1.0mm. |
| 41 - 45 | 5X | - | Blank. |
| 46 - 49 | A4 | NAME1 | Flag for plotting of calculation results.<br>'PLOT':plotting of calculation results.<br>'    ':no plotting. |
| 50 | 1X | — | Blank. |
| 51 - 60 | F10.0 | TAU(1) | Failure shear stress of outer layer plate (kg/mm$^2$). |
| 61 - 70 | F10.0 | TAU(2) | Failure shear stress of intermediate layer plate(kg/mm$^2$). |
| 71 - 80 | F10.0 | TAU(3) | Failure shear stress of inner layer plate (kg/mm$^2$). |
| Data set No.4:Yield stress data for plates. | | | |
| 1 - 5 | A5 | NAME | Flag for yield stress of plates.<br>'PLATE': |
| 6 - 10 | 5X | — | Blank. |
| 11 - 20 | F10.0 | SIGM1 | Yield stress of outer layer plate(kg/mm$^2$). |
| 21 - 30 | F10.0 | SIGM2 | Yield stress of intermediate layer plate (kg/mm$^2$). |
| 31 - 40 | F10.0 | SIGM3 | Yield stress of inner layer plate(kg/mm$^2$). |
| Data set No.5:Number of stress-strain data of plates and puncture bar. | | | |
| 1 - 4 | A4 | NAME1 | Flag for plate and puncture bar data.<br>'SIG1':flag for outer layer plate.<br>'SIG2':flag for intermediate layer plate.<br>'SIG3':flag for inner layer plate.<br>'SIGT':flag for puncture bar. |
| 5 - 10 | 6X | — | Blank. |

Table 4.1 (Continued)

| Columns | Format | Variables | Descriptions |
|---|---|---|---|
| 11 - 15 | I5 | NSET | Number of stress-strain data of plates. |
| Data set No. 6:Stress-strain data for plates and puncture bar. | | | |
| 1 - 10 | F10. 0 | VALTAB(1, 1) | Strain(mm/mm). |
| 11 - 20 | F10. 0 | VALTAB(2, 1) | Stress(kg/mm$^2$). |
| 21 - 30 | F10. 0 | VALTAB(1, 2) | ........ |
| 31 - 40 | F10. 0 | VALTAB(2, 2) | ........ |
| 41 - 50 | F10. 0 | VALTAB(1, 3) | ........ |
| 51 - 60 | F10. 0 | VALTAB(2, 3) | ........ |
| 61 - 70 | F10. 0 | VALTAB(1, 4) | Strain(mm/mm). |
| 71 - 80 | F10. 0 | VALTAB(2, 4) | Stress(kg/mm$^2$). |
| | | | Rpeat No. 6 data set for number of necessary data NSET of data set No. 5. |

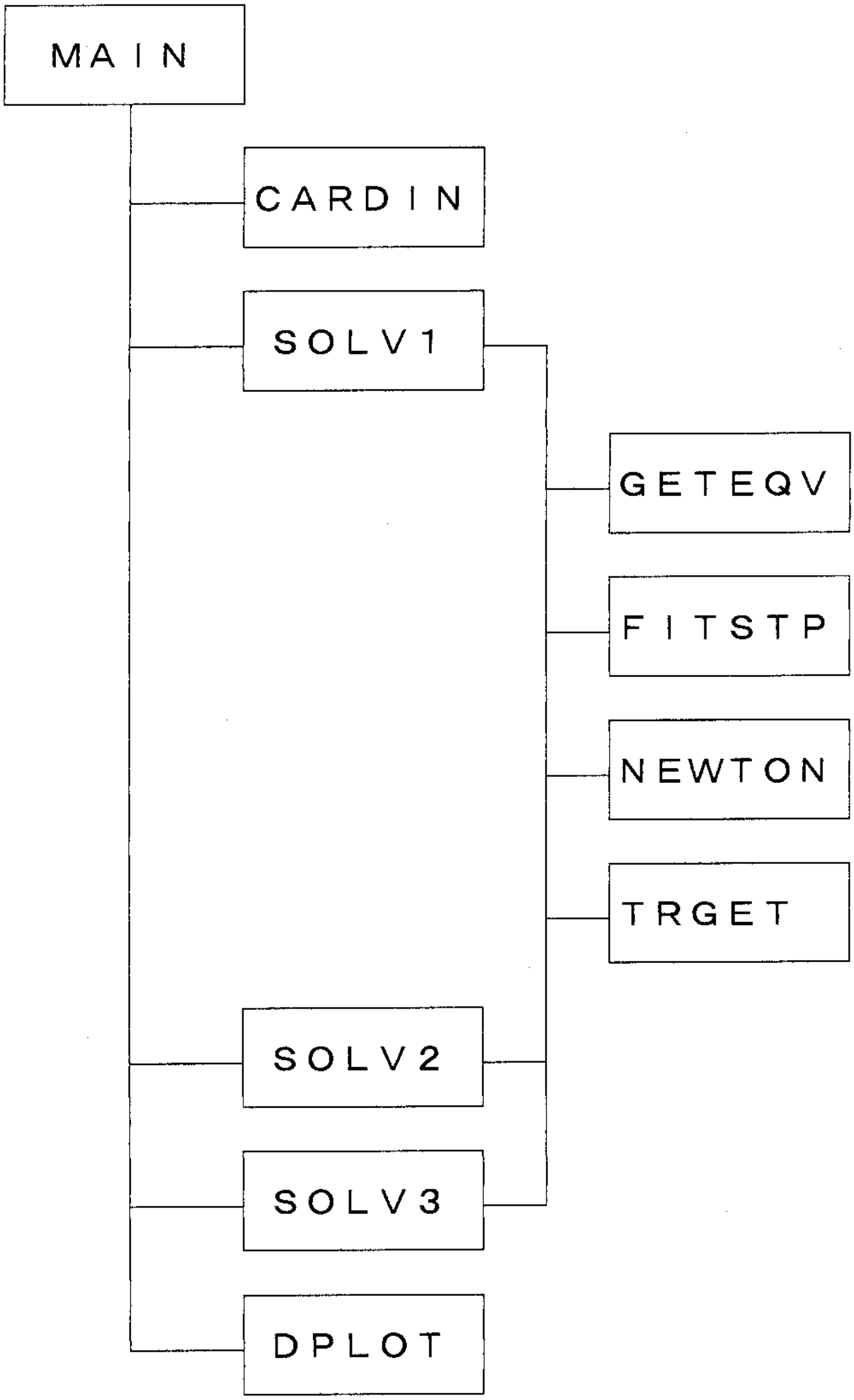


Fig. 4.1 Structure of computer program PUNCTURE

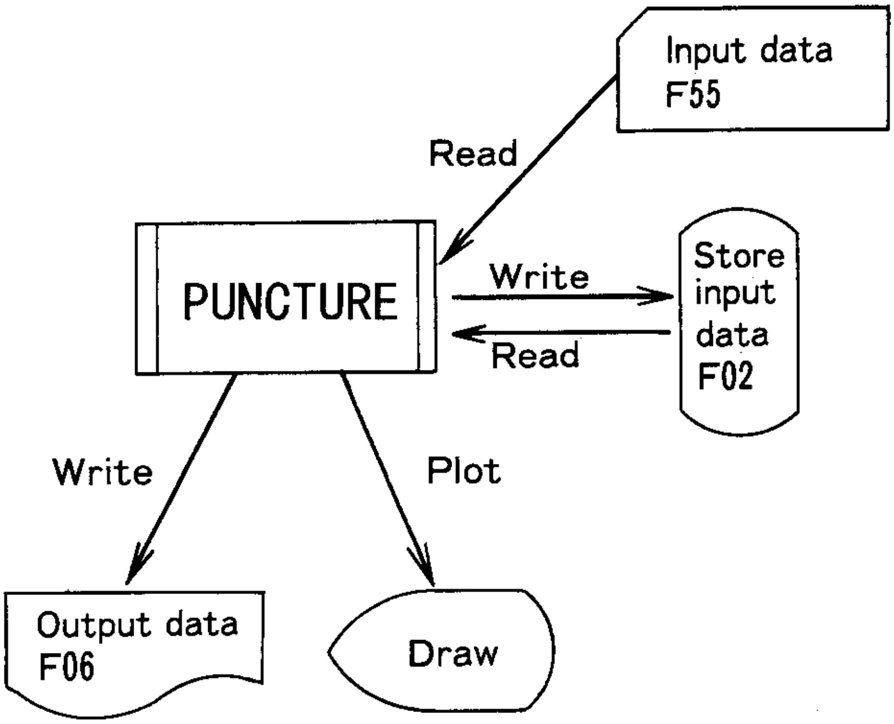


Fig. 4.2 Program flow

## 5. Conclusions

With regard to the evaluation of the maximum acceleration and deformation of the cask bodys in the case of the drop impact onto the puncture bar, a simplified computer program PUNCTURE will analyze it economically and by save computer time as compared with the other detailed computer programs with a analysis method of dynamic interactions. The PUNCTURE is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

## Ackowledgements

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

## 5. Conclusions

With regard to the evaluation of the maximum acceleration and deformation of the cask bodys in the case of the drop impact onto the puncture bar, a simplified computer program PUNCTURE will analyze it economically and by save computer time as compared with the other detailed computer programs with a analysis method of dynamic interactions. The PUNCTURE is further being utilized satisfactory in safety analysis and designing not only spent fuel transport casks but also those for various radioactive transport casks.

## Ackowledgements

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

## References

(1)Halliquist,J.O.,"User's Manual for DYNA2D-An Explicit Two-dimensional Hydro-dynamic Finite Element Code with Interactive Rezoning and Graphical Display", UCID-18756,Rev.3,(1988).

(2)Halliquist,J.O.,"User's Manual for DYNA3D(Nonlinear Dynamic Analysis of Structures in Three Dimensions)", UCID-19592,Rev.5,(1989).

(3)Halliquist,J.O.,"NIKE2D:An Implicit, Finite-Deformation, Finite Element Code with Analyzing the Static and Dynamic Response of Two-Dimensional Solids",UCID-52678,(1978).

(4)Halliquist,J.O.,"NIKE3D:An Implicit, Finite-Deformation, Finite Element Code with Analyzing the Static and Dynamic Response of Three-Dimensional Solids",UCID-18822,(1981).

(5)Trigg,M.,et al.,"PISCES User's Manual",Physics International(1980).

(6)Key,S.W.,Beisinger,Z.E. and Krieg.,D.,"HONDO II-A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids",SAND 78-0422(1978).

(7)Ikushima, T. et al.,"Simplified Computer Codes for Cask Impact Analysis" , 10th Int. symposium on the Packaging and Transportation of Radioactive Materials, pp.1419-1426, Japan(Yokohama), (1992).

(8)Ikushima, T., Ohshika, J. and Ishiwata, T.,"Computer Codes System for Structural Analysis of Radioactive Materials Transport" , 11th Int. symposium on the Packaging and Transportation of Radioactive Materials, pp.1174-1181, U.S.A(Las Vegas), (1995).

(9)Onat,E.T. and Haythornthwaite,R.M.,"The Load-Carrying Capacity of Circular Plates at Large Deflection",Trans. of ASME, Journal of Applied Mechanics, Vol.23,pp49-55(1956).

(10)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).

(11)Larder,R.A. and Arthur,D.,"Puncture of Shielded Radioactive Material Shipping Containers - Analysis and Results",NUREG/CR-0930 UCRL/52638 Vol.1 RT(1980).

# Appendix A Sample Problem Input

```
                             INPUT DATA ECHO

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

TITLE     TEST MODEL RUN  ( (D/T1)=3.0 MODEL )
SIMPLE-B  101.6      7.62      20.32     5.08      15.24      0.0
ANALYSIS  9000.0    1000.0      1.0          PLOT  61.80      3.164
PLATE       27.70     1.120
SIGT          1
0.0015      31.64
FIN

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

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

# Appendix B Sample Problem Output

```
                                   P U N C T U R E   M O D E L   D A T A

                1         2         3         4         5         6         7         8
        ....:....0....:....0....:....0....:....0....:....0....:....0....:....0....:....0
  1-    TITLE     TEST MODEL RUN  ( (D/T1)=3.0 MODEL )
  2-    SIMPLE-B  101.6     7.62      20.32     5.08      15.24     0.0
  3-    ANALYSIS  9000.0    1000.0    1.0           PLOT  61.80     3.164
  4-    PLATE       27.70     1.120
  5-    SIGT          1
  6-    0.0015      31.64
```

```
 ...... ANALYSIS MODEL=SIMPLE-BENDING
 ...... MODEL WEIGHT  =   9000.00  (KG)
 ......       HEIGHT  =   1000.00  (MM)
 ......       RADIAUS =    101.60  (MM)
 ......TARGET RADIAUS =      7.62  (MM)
 ......TARGET LENGTH  =     20.32  (MM)
 ...... INCREMENT DISP=      1.00  (MM)
 ...... TOTAL ENERGY  =   9000000.0(KG-MM)
 ......  PUNCTURE LOAD=  10403.77  (KG)
```

# Appendix B (Continued)

```
TEST MODEL RUN  ( (D/T1)=3.0 MODEL )

     STEP        DEPTH           FORCE          ENERGY  ACCELERATION  BAR-DISP.  BAR-STRESS
        1       1.0000         2350.80         2379.99       2559.76       0.01       12.89
        2       2.0000         2468.59         4880.76       2688.02       0.03       13.53
        3       3.0000         2664.89         7583.15       2901.77       0.04       14.61
        4       4.0000         2939.72        10568.51       3201.02       0.06       16.12
        5       5.0000         3291.61        13917.34       3584.20       0.07       18.04
        6       6.0000         3699.61        17689.23       4028.47       0.09       20.28
        7       7.0000         4140.13        21919.88       4508.14       0.11       22.70
        8       8.0000         4600.96        26632.63       5009.93       0.14       25.22
        9       9.0000         5075.33        31844.00       5526.47       0.16       27.82
       10      10.0000         5559.19        37566.40       6053.34       0.19       30.48
       11      11.0000         6049.95        43809.64       6587.72       0.23       33.17
       12      12.0000         6545.87        50581.79       7127.73       0.26       35.88
       13      13.0000         7045.77        57889.73       7672.07       0.30       38.63
       14      14.0000         7548.80        65739.47       8219.81       0.34       41.38
       15      15.0000         8054.33        74136.40       8770.28       0.38       44.15
       16      16.0000         8561.90        83085.43       9322.95       0.43       46.94
       17      17.0000         9071.13        92591.11       9877.45       0.47       49.73
       18      18.0000         9581.76       102657.73      10433.47       0.52       52.53
       19      19.0000        10093.57       113289.32      10990.77       0.58       55.33
       20      19.1000        10144.80       114847.31      11046.56       0.69       55.61
       21      19.2000        10196.05       116415.93      11102.37       0.74       55.89
       22      19.3000        10247.31       117995.21      11158.18       0.80       56.18
       23      19.4000        10298.58       119585.18      11214.00       0.85       56.46
       24      19.5000        10349.85       121185.86      11269.84       0.90       56.74
       25      19.6000        10401.14       122797.30      11325.68       0.96       57.02
       26      19.6051        10403.77       122880.46      11328.55       1.01       57.03

   PLATE IS PUNCTURED.
```

# Appendix C Graphical Output

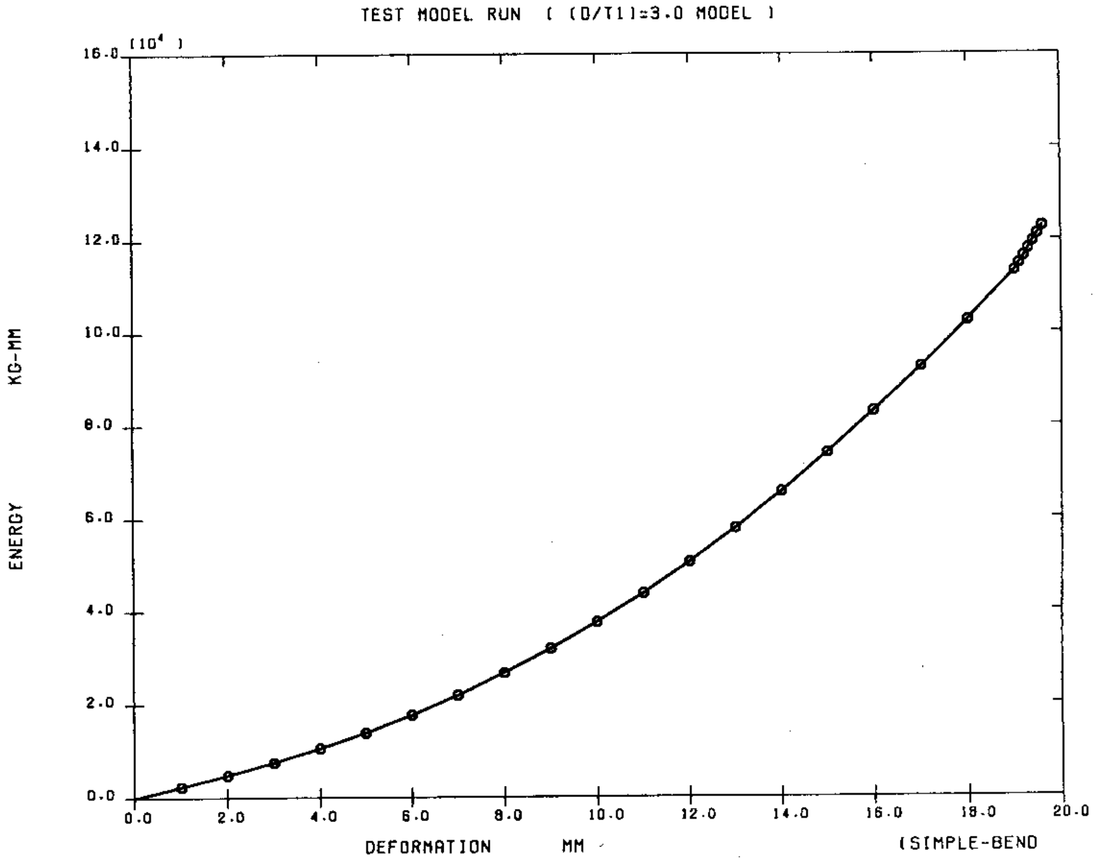


Fig. C.1 Graphical Output of PUNCTURE(1)

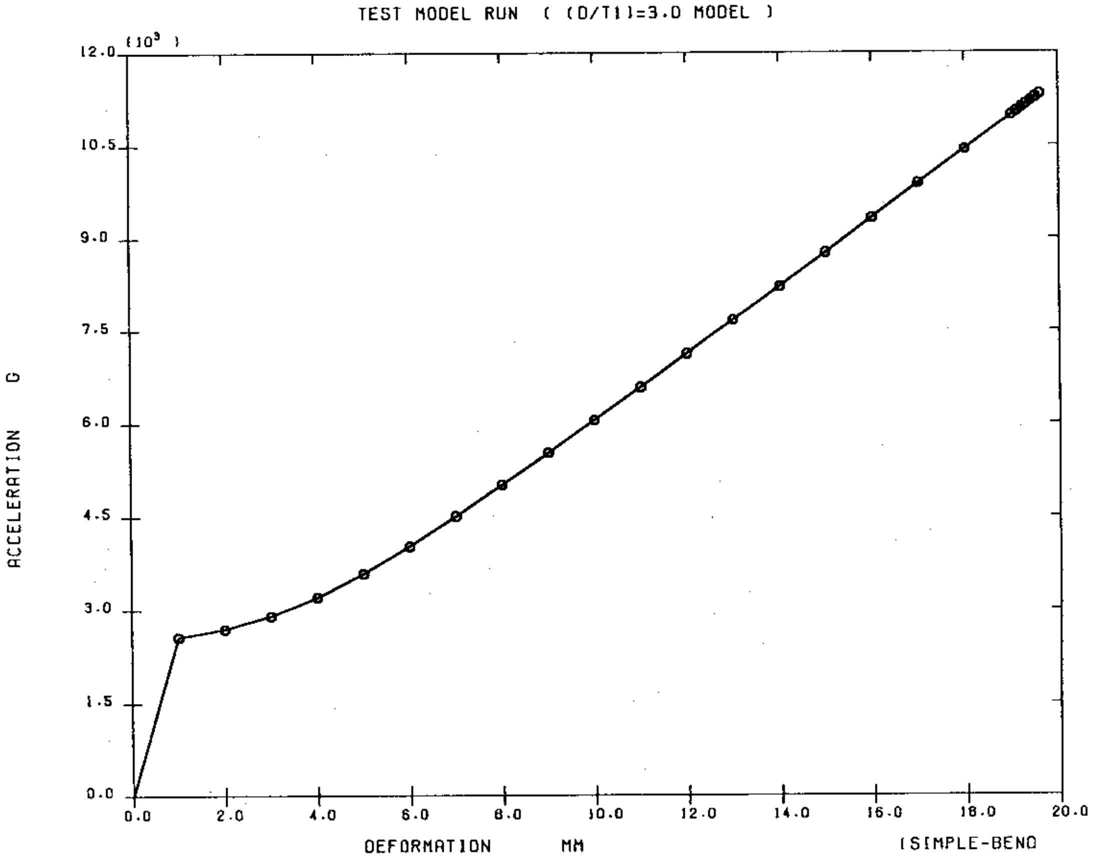


Fig. C.2 Graphical Output of PUNCTURE(2)

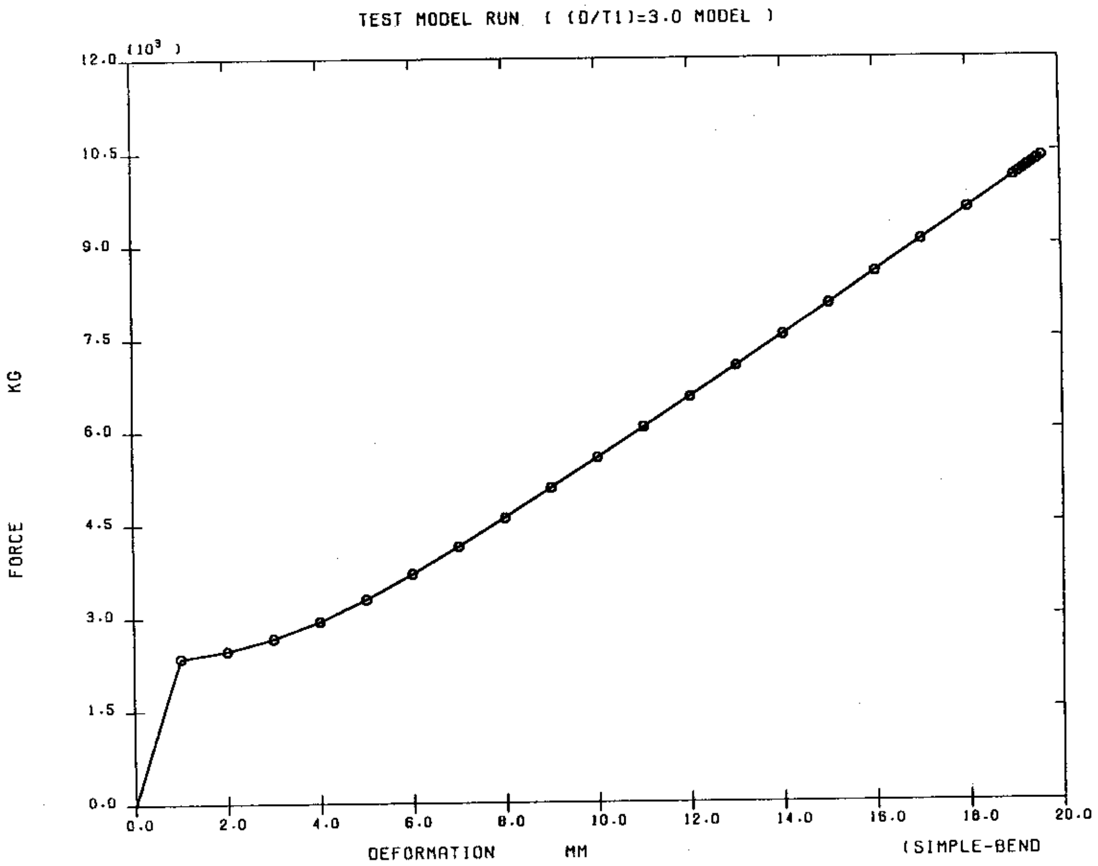


Fig. C.3 Graphical Output of PUNCTURE(3)

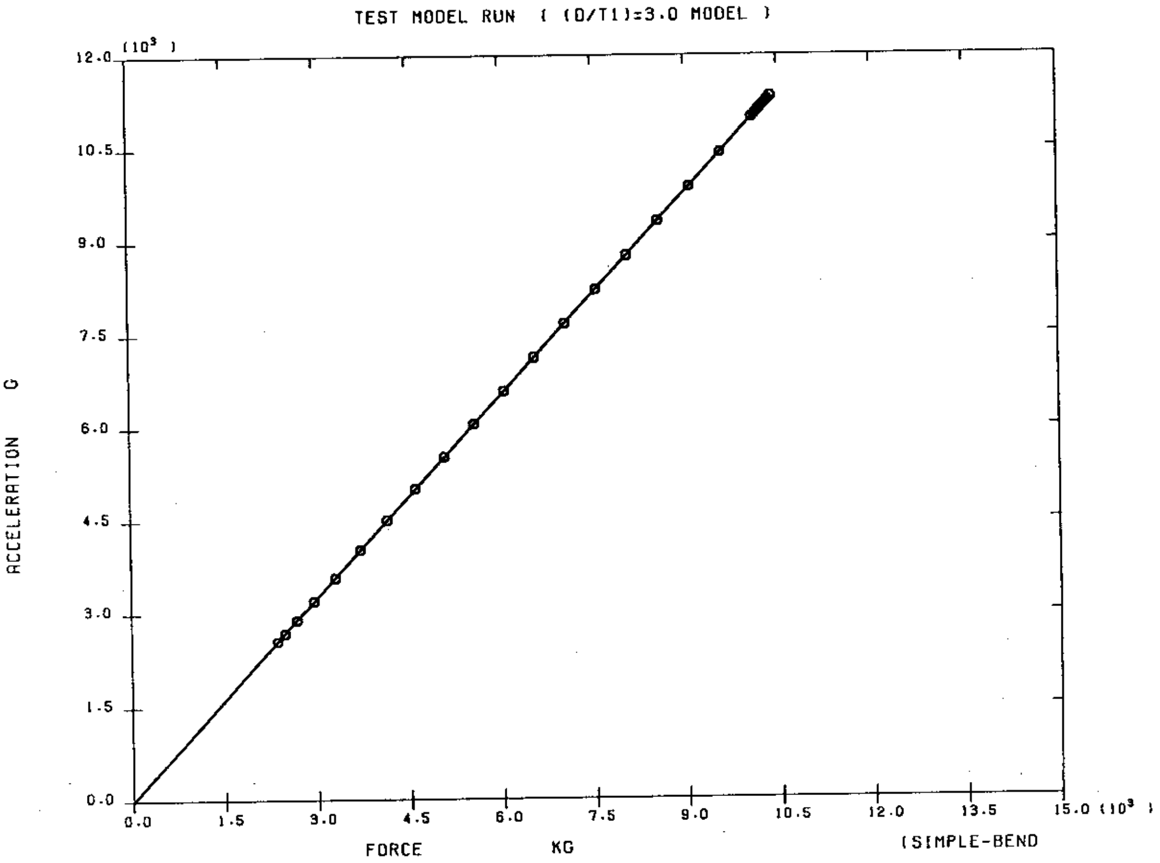


Fig. C.4 Graphical Output of PUNCTURE(4)

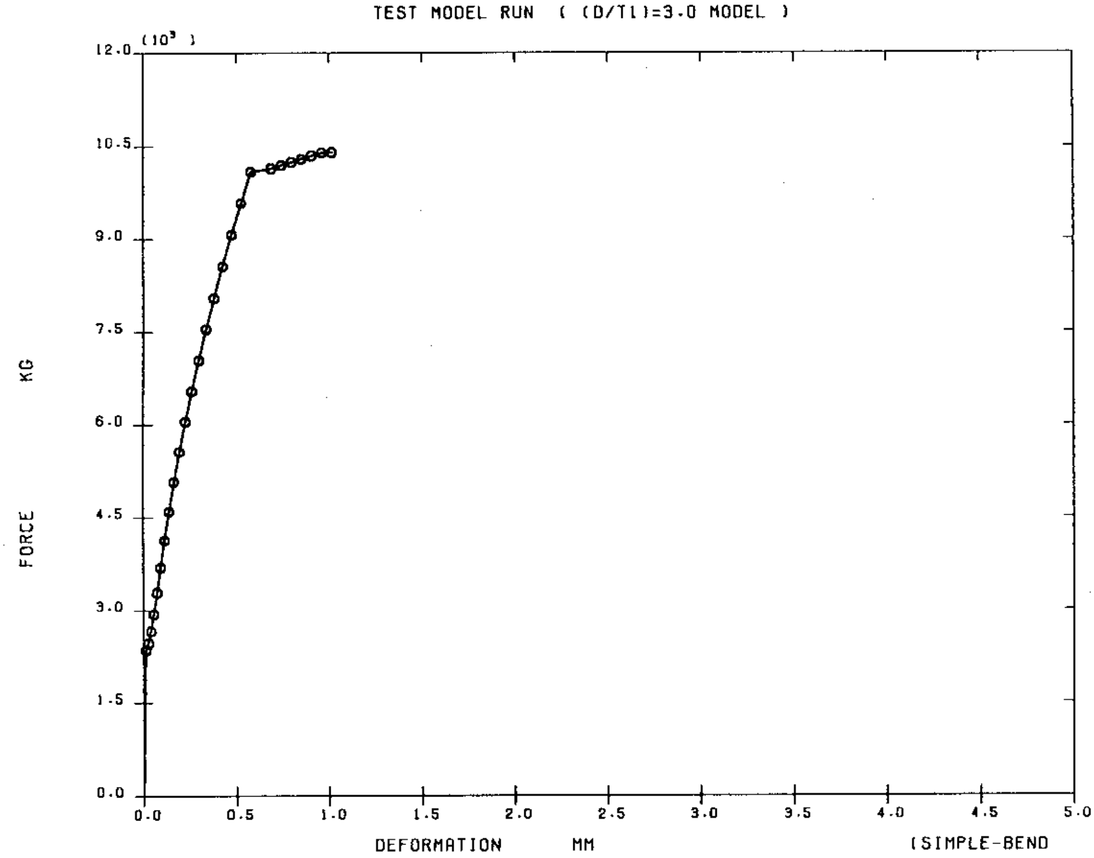


Fig. C.5 Graphical Output of PUNCTURE(5)

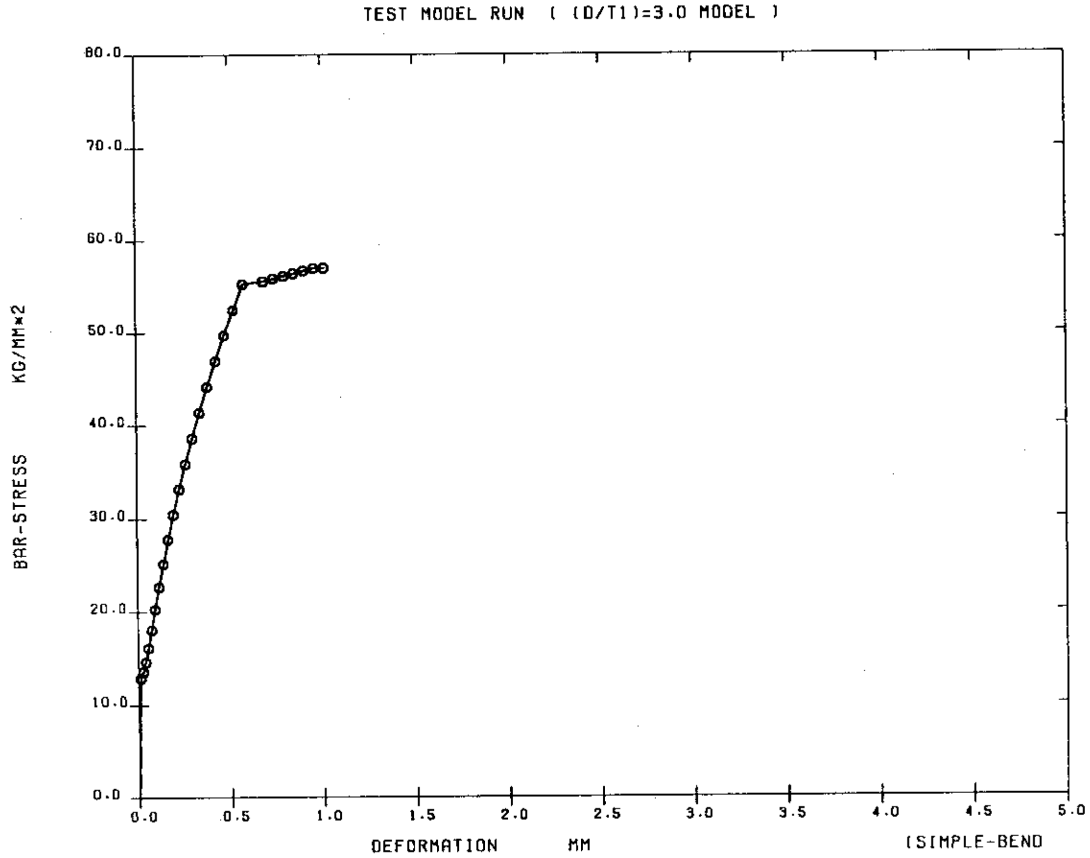


Fig. C.6 Graphical Output of PUNCTURE(6)

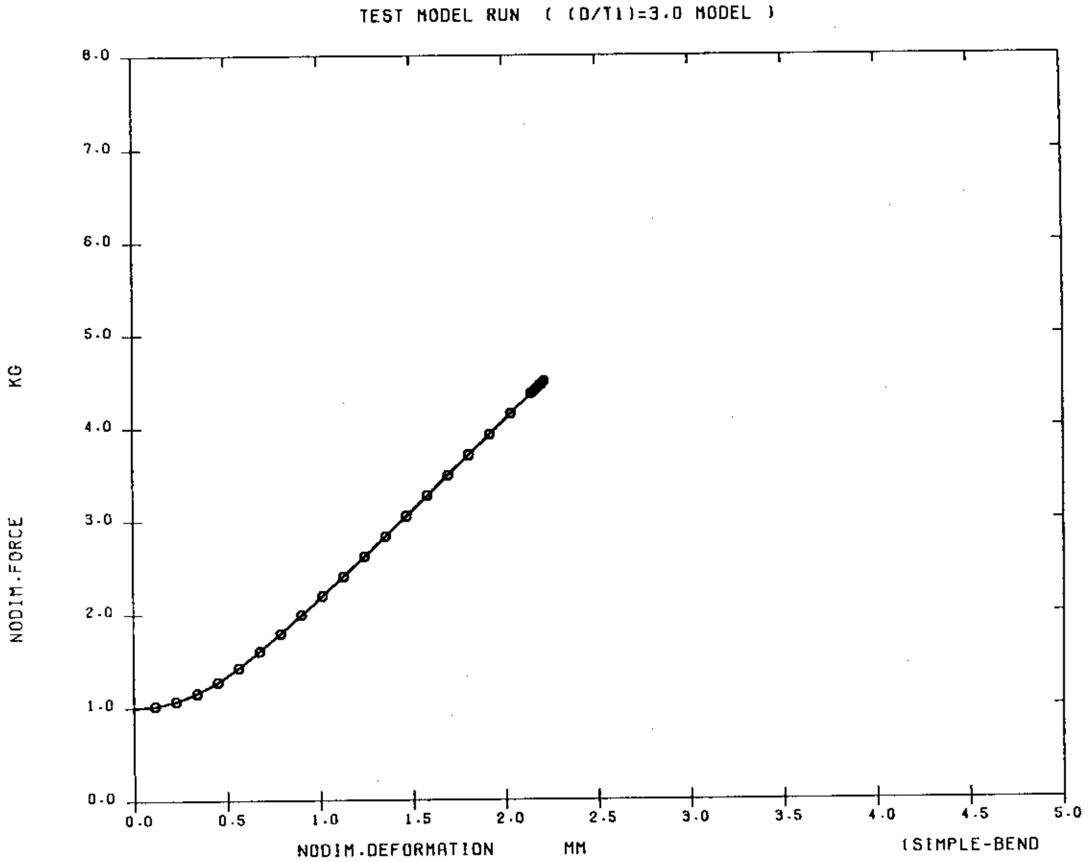


Fig. C.7 Graphical Output of PUNCTURE(7)

# Appendix D Job Control Data

The job control data for PUNCTURE 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,PUNCTURE
// T.01 C.02 W.01 I.02 CLS GRP
// OPTP MSGCLASS=A,MSGLEVEL=(2,0,1),CLASS=B,NOTIFY=JXXXX
// OPTP PASSWORD=XXXXXXXX
// EXEC LMGOEX,LM=J2322.LMPUNCT1,PNM=FINLIB
// EXPAND GRNLP
//FT55F001 DD DSN=JXXXX.DTPUNCT1.DATA,DISP=SHR
//FT02F001 DD DSN=SPACE=(TRK,(5,5)),UNIT=TSSWK
++
//
```

# Appendix E Program Abstract

1. Name :

**PUNCTURE.**

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:

Puncture 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 : 1 second.
SUN4 : 2 seconds.
IBM-PC : 3 seconds.

7. Unusual features of the program:

None.

8. Related and auxiliary program:

None.

9. Status:

-

10. References:

(1)Ikushima,T. and Hode S.,"Simplified Analysis Computer Program and Their Adequacy for Radioactive Materials Shipping casks", PATRAM'89, pp.1202-1209, Washington DC, USA ,June 11-16,(1989).

(2)Ikushima,T. et al.,"Simplified Computer Codes for Cask Impact Analysis", PATRAM'92, pp.1419-1426, Yokohama, Japan, September 13-18,(1992).

(3)Ikushima,T., Ohshika,J. and Ishiwata,T.,"Computer Code System for Structural Anaalysis of Radioactive Materials Transport", PATRAM'95, pp.1174-1181, Las Vegas, USA, December 3-8,(1995).

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 1300 source steps(include comment lines). The graphical programs are as follows:

FACOM M-780 : CALCOMP pltter 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

```
C..................................................
C
C       P U N C T U R E   C O D E   (1994-08-01)
C
C..................................................
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +           WEIGHT,   HEIGHT,   ROUT,     TLENG,    AA,
     +           DISP,     ENERGY,   THICK(3), TAU(3),   PCR,
     +           MODEL,    KPLOT,    MTINN,    MTOUT,
     +           NTITLE(18)
      COMMON /MATDAT/      VALTAB(2,20,4),     LNGTAB(4),
     +           SIGM1,    SIGM2,    SIGM3
      COMMON /RESLT/
     +           DEPTH(500),         DEPTHB(500),        FORCE(500),
     +           ACCEL(500),         ENERG(500),         TODISP,
     +           MXSTEP,   MPUNCT
      COMMON /EQUIVL/      EQIVT,    PL
      COMMON /SIGSIG/      BARSIG(500)
      COMMON /DATEOF/      KEOF,     KSWPLT
C...
C
      DATA PAI /  3.141592654D0 /
C
C...    GET MODEL
C
      KSWPLT = 0
      KEOF   = 0
C
      MXSTEP = 0
      MPUNCT = 0
C
  100 CONTINUE
      CALL CARDIN( MEOF )
      IF( MEOF .NE. 0 )              GO TO 8000
C
C...    CALCULATION
C
      IF( MODEL.EQ.0 )               GO TO 8000
      GO TO ( 310, 320, 330 )     , MODEL
C. (FIXED-BENDING)
  310 CALL SOLVE1
      GO TO 1000
C. (SIMPLE-BENDING)
  320 CALL SOLVE2
      GO TO 1000
C. (FIXED-MEMBRANE)
  330 CALL SOLVE3
C
C
C...      END OF ANALYSIS
C
 1000 CONTINUE
      WRITE(MTOUT,9010)        NTITLE
 9010 FORMAT(1H1/5X,18A4/)
C
```

## Appendix F (Continued)

```
      WRITE(MTOUT,9030)
 9030 FORMAT(10X,4HSTEP,12H       DEPTH ,9X,6H FORCE,7X,6HENERGY,
     +        2X,35HACCELERATION  BAR-DISP.  BAR-STRESS    )
      BARARE = PAI*AA*AA
      DO 5200 J= 1, MXSTEP
      BARSIG(J) = FORCE(J)/BARARE
      WRITE(MTOUT,9040)        J, DEPTH(J), FORCE(J), ENERG(J),
     +                            ACCEL(J), DEPTHB(J), BARSIG(J)
 9040 FORMAT(10X,I4,F12.4, 2(3X,F12.2), 3(3X,F8.2) )
      IF( MOD( J ,50) .NE. 0 )     GO TO 5200
      WRITE(MTOUT,9050)        NTITLE
 9050 FORMAT(1H1/5X,18A4//
     +        10X,4HSTEP,12H       DEPTH ,9X,6H FORCE,7X,6HENERGY,
     +        2X,35HACCELERATION  BAR-DISP.  BAR-STRESS    )
 5200 CONTINUE
      IF( MPUNCT.NE.0 )            WRITE(MTOUT,9060)
 9060 FORMAT(/10X,20HPLATE IS PUNCTURED. /)
C
C...   PLOT RESULT
C
      IF( KPLOT.EQ.0 )             GO TO 6000
      KSWPLT = KSWPLT + 1
      IF( KSWPLT .EQ. 1 )          CALL PLTBGN
C
      CALL DPLOT  ( MODEL,NTITLE, DEPTH, ENERG, MXSTEP, 1 )
      CALL DPLOT  ( MODEL,NTITLE, DEPTH, ACCEL, MXSTEP, 2 )
      CALL DPLOT  ( MODEL,NTITLE, DEPTH, FORCE, MXSTEP, 3 )
      CALL DPLOT  ( MODEL,NTITLE, FORCE, ACCEL, MXSTEP, 4 )
      CALL DPLOT  ( MODEL,NTITLE,DEPTHB, FORCE, MXSTEP, 6 )
      CALL DPLOT  ( MODEL,NTITLE,DEPTHB,BARSIG, MXSTEP, 7 )
C
C
C
 6000 WRITE(MTOUT,9010)        NTITLE
      WRITE(MTOUT,9080)
 9080 FORMAT(     10X,4HSTEP,12H   NON.DEPTH ,7X,9HNON.FORCE)
C
      DO 6100 J= 1, MXSTEP
      DEPTH(J) = DEPTH(J)/EQIVT
      FORCE(J) = FORCE(J)/PL
      WRITE(MTOUT,9040)        J, DEPTH(J), FORCE(J)
      IF( MOD( J ,50) .NE. 0 )     GO TO 6100
      WRITE(MTOUT,9090)
 9090 FORMAT(1H1//10X,4HSTEP,12H   NON.DEPTH ,7X,9HNON.FORCE)
 6100 CONTINUE
C
      IF( KPLOT.EQ.0 )             GO TO  99
      CALL DPLOT  ( MODEL, NTITLE, DEPTH, FORCE, MXSTEP, 5 )
   99 CONTINUE
C
 8000 CONTINUE
      IF( KSWPLT .GT. 0 ) CALL PLTEND
 8100 CONTINUE
      STOP
      END
```

## Appendix F (Continued)

```
      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***
C***
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,   HEIGHT,   ROUT,     TLENG,     AA,
     +          DISP,     ENERGY,   THICK(3), TAU(3),    PCR,
     +          MODEL,    KPLOT,    MTINN,    MTOUT,
     +          NTITLE(18)
      COMMON /MATDAT/     VALTAB(2,20,4),     LNGTAB(4),
     +          SIGM1,    SIGM2,    SIGM3
      COMMON /RESLT/
     +          DEPTH(500),         DEPTHB(500),         FORCE(500),
     +          ACCEL(500),         ENERG(500),          TODISP,
     +          MXSTEP,   MPUNCT
      COMMON /DATEOF/     KEOF,     KSWPLT
C
C
      DIMENSION LABEL(2,6),         KSIG(4),  NAME(2),  KEY(2),
     +          NCARD(20),          KHEAD(2), KEYMDL(4,3)
      CHARACTER CARD*80
C
      DATA NCARD/ 20*4H    /
      DATA LABEL/ 4HTITL, 4HE   ,  4HANAL, 4HYSIS,
     +            4HFIXE, 4HD-B ,  4HSIMP, 4HLE-B,
     +            4HFIXE, 4HD-M ,  4HPLAT, 4HE   /
      DATA KEYMDL/4HFIXE, 4HD-BE, 4HNDIN, 4HG   ,
     +            4HSIMP, 4HLE-B, 4HENDI, 4HNG  ,
     +            4HFIXE, 4HD-ME, 4HMBRA, 4HNE  /
      DATA KSIG/  4HSIG1, 4HSIG2, 4HSIG3, 4HSIGT/
      DATA MFIN/  4HFIN /
      DATA NBLANK/4H    /
      DATA KEY/   4H    , 4HPLOT/
      DATA LCOMM/ 4H**  /
      DATA PAI2/  6.283185307D0 /
      DATA PAI /  3.141592654D0 /
C
C
C...    INITIAL VALUES
C
      MTINN  = 2
      MTOUT  = 6
      MXSTEP = 0
      MXPHI  = 0
      DO 30 JJ=1,4
   30 LNGTAB(JJ) = 0
C
C...    CARD IMAGE
```

## Appendix F (Continued)

```
C
      LINE   = 0
      IF( KEOF.NE.0 )               GO TO 160
C
      REWIND MTINN
C
  110 CONTINUE
  120 FORMAT(A80)
      READ(  55,130)            NCARD
  130 FORMAT(20A4)
      IF( NCARD(1).EQ.MFIN )        GO TO 160
C
      LINE   = LINE+1
      IF( MOD(LINE,50).NE.1 )       GO TO 140
      WRITE(MTOUT,100)          (I,I=1,8)
  100 FORMAT(1H1 ///34X,38HP U N C T U R E   M O D E L   D A T A //
     +              10X,8I10/10X,10H....:....0,7(10H....:....0) )
C
  140 WRITE(MTOUT,145)          LINE, NCARD
  145 FORMAT(1X,I4,5H-    ,20A4)
C
      IF( NCARD(1).EQ.LCOMM )       GO TO 110
      WRITE(MTINN,130)          NCARD
      GO TO 110
C
C...    GET A COMMAND.
C
  150 KEOF   = 1
  160 IF( LINE.GT.0 )               GO TO 170
      MEOF   = 1
      RETURN
C
C
C. (1)
  170 DO 171 I=1,20
  171 NCARD(I) = NBLANK
      NCARD(1) = MFIN
      WRITE(MTINN,130)          NCARD
C
      REWIND MTINN
      READ(MTINN,211)           NAME, NTITLE
  211 FORMAT(2A4,2X,17A4,A2)
      LINE   = 1
      IF( NAME(1).NE.LABEL(1,1) )         GO TO 999
      IF( NAME(2).NE.LABEL(2,1) )         GO TO 999
C
C. (2)
      READ(MTINN,221)           NAME, ROUT, AA, TLENG, THICK
  221 FORMAT(2A4,2X,6F10.0)
      LINE   = LINE+1
C
      DO 222 MODEL=1,3
      IF( NAME(1).NE.LABEL(1,MODEL+2) ) GO TO 222
      IF( NAME(2).EQ.LABEL(2,MODEL+2) ) GO TO 225
  222 CONTINUE
      GO TO 999
  225 CONTINUE
C
```

```
C. (3)
      READ(MTINN,231)          NAME, WEIGHT, HEIGHT, DISP,  NAME1, TAU
  231 FORMAT(2A4,2X,3F10.0,5X,A4,1X,3F10.0)
      LINE    = LINE+1
      IF( NAME(1).NE.LABEL(1,2) )       GO TO 999
      IF( NAME(2).NE.LABEL(2,2) )       GO TO 999
C
      KPLOT   = 0
      IF( NAME1.EQ.KEY(1) )             KPLOT   = 0
      IF( NAME1.EQ.KEY(2) )             KPLOT   = 1
C
      IF( DISP.EQ.0.0 )                 DISP    = 1.0
C
      DO 232 I=1,3
  232 TAU(I) = 0.6D0*TAU(I)
      D3      = 0.0
      DO 233 I=1,3
  233 D3      = D3 + TAU(I)*THICK(I)
      PCR     = PAI2*AA*D3
C
C. (4)
      READ(MTINN,241)          NAME, SIGM1, SIGM2, SIGM3
  241 FORMAT(2A4,2X,3F10.0)
      LINE    = LINE+1
      IF( NAME(1).NE.LABEL(1,6) )       GO TO 999
      IF( NAME(2).NE.LABEL(2,6) )       GO TO 999
C
      IF( SIGM1.LT.0.0D0 )              GO TO 999
      IF( SIGM2.LT.0.0D0 )              GO TO 999
      IF( SIGM3.LT.0.0D0 )              GO TO 999
C
C. (5)
  300 CONTINUE
      READ(MTINN,251)          NAME1, NSET
  251 FORMAT(A4,6X,I5)
      LINE    = LINE+1
      NAME(1) = NAME1
      NAME(2) = NBLANK
      IF( NAME1.EQ.MFIN )               GO TO 400
      DO 252 JJ=1,4
      IF( NAME1.EQ.KSIG(JJ) )           GO TO 255
  252 CONTINUE
      GO TO 999
  255 CONTINUE
      IF( NSET.LE.0 )                   GO TO 999
      IF( NSET.GT.20 )                  GO TO 999
      LNGTAB(JJ) = NSET
C
C. (6)
      READ(MTINN,261)          (VALTAB(1,I,JJ),VALTAB(2,I,JJ),I=1,NSET)
  261 FORMAT(8F10.0)
      LINE    = LINE+(NSET-1)/4+1
C
      S       = 0.0D0
      DO 265 I=1,NSET
      IF( VALTAB(2,I,JJ).LE.S )         GO TO 999
      S       = VALTAB(2,I,JJ)
  265 CONTINUE
```

```
      GO TO 300
C
C...
  400 CONTINUE
      IF( LNGTAB(4).EQ.0 )              GO TO 984
      ENERGY = WEIGHT*HEIGHT
C
C
C
      WRITE(MTOUT,640)         (KEYMDL(J,MODEL),J=1,4), WEIGHT, HEIGHT,
     +                          ROUT, AA, TLENG, DISP, ENERGY, PCR
  640 FORMAT(/////11X,22H...... ANALYSIS MODEL=,4A4 /
     +            11X,22H...... MODEL WEIGHT  =,F10.2,2X,4H(KG)      /
     +            11X,22H......       HEIGHT  =,F10.2,2X,4H(MM)      /
     +            11X,22H......       RADIAUS =,F10.2,2X,4H(MM)      /
     +            11X,22H......TARGET RADIAUS =,F10.2,2X,4H(MM)      /
     +            11X,22H......TARGET LENGTH  =,F10.2,2X,4H(MM)      /
     +            11X,22H...... INCREMENT DISP=,F10.2,2X,4H(MM)      /
     +            11X,22H...... TOTAL ENERGY  =,F12.1    ,8H(KG-MM)  /
     +            11X,22H......  PUNCTURE LOAD=,F10.2,2X,4H(KG)      /)
C
C
C
      MEOF    = 0
      RETURN
C
C
C
C...    ERROR FORMAT CARD.
  999 MEOF    = 1
      WRITE(MTOUT,9991)        LINE, NAME
 9991 FORMAT(40H **** ERROR CRAD FOUND.       CARD NO. =,I5,5X,
     + 13HCOLUMN (1-8)=,2A4)
      RETURN
C
  981 MEOF    = 1
      WRITE(MTOUT,9981)
 9981 FORMAT(40H **** EMPTY MATERIAL DATA LABEL =(SIG1) )
      RETURN
  982 MEOF    = 1
      WRITE(MTOUT,9982)
 9982 FORMAT(40H **** EMPTY MATERIAL DATA LABEL =(SIG2) )
      RETURN
  983 MEOF    = 1
      WRITE(MTOUT,9983)
 9983 FORMAT(40H **** EMPTY MATERIAL DATA LABEL =(SIG3) )
      RETURN
  984 MEOF    = 1
      WRITE(MTOUT,9984)
 9984 FORMAT(40H **** EMPTY MATERIAL DATA LABEL =(SIGT) )
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    SOLVE1
C
C****   /PURPOSE/
C         CALCULATION ENERGRY OF PUNCTURE
C           (FIXED-BENDING ANALYSIS)
C
C***     PROGRAMMED BY JAERI        1989.10.06
C***    /MODIFIED   BY JAERI        1990.12.06
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,   HEIGHT,   ROUT,     TLENG,    AA,
     +          DISP,     ENERGY,   THICK(3), TAU(3),   PCR,
     +          MODEL,    KPLOT,    MTINN,    MTOUT,
     +          NTITLE(18)
      COMMON /MATDAT/     VALTAB(2,20,4),     LNGTAB(4),
     +          SIGM1,    SIGM2,    SIGM3
      COMMON /RESLT/
     +          DEPTH(500),         DEPTHB(500),        FORCE(500),
     +          ACCEL(500),         ENERG(500),         TODISP,
     +          MXSTEP,   MPUNCT
      COMMON /EQUIVL/     EQIVT,    PL
      DATA PAI/ 3.141592654D0 /
C
C...    INITIAL VALUE GET
C
      CALL   NEWTON (AA,ROUT,PP)
      XXPP    = DLOG(ROUT/PP)
      D1      = 1.0D0+2.0D0*XXPP
      D2      = 2.0D0+XXPP
      D3      = 1.0D0+XXPP
      D4      = 1.0D0+3.0D0*XXPP
      UDISP   = 0.5D0+0.5D0*XXPP
      A1      = D1/(D2*D3)
      A2      = 2.0D0*D4/(3.0D0*D2*D3*D3)
      B1      = (3.0D0+XXPP)/(2.0D0*D2)
      B2      = 2.0D0*D1/(D2*D3)
      B3      = D3/(12.0D0*D2)
      A       = D2
      B       = D3-2.0D0*AA/(3.0D0*PP)
      IF( PP.LT.AA )          B       = 0.5D0+DLOG(ROUT/AA)-PP*PP/
     +                                  (6.0D0*AA*AA)
C
C...    INITIAL STATUS
C
      XWEIGH =WEIGHT/9800.0
      MXSTEP = 0
      TODISP = 0.0
      BBDISP = 0.0
      SUM    = 0.0
      MINI   = 0
      DISPX  = DISP
      AREAB  = PAI*AA*AA
      CALL GETEQV (EQIVMP,EQIVNP,EQIVT)
      PL     = 2.0D0*PAI*EQIVMP*A/B
      UDELT  = 1.0D0/EQIVT
C
C...    GET CURRENT FORCE
```

## Appendix F (Continued)

```
C
  100 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )           GO TO 1000
      TODISP = TODISP + DISPX
C
C. (DISP .VS. UDISP)
C
      U       = TODISP*UDELT
      IF( U.GT.UDISP )              GO TO 200
      P       = PL*( 1.0D0+A1*U+A2*U*U )
      GO TO 300
  200 P       = PL*( B1+B2*U+B3/U )
C
C. (TARGET DISP)
C
  300 CONTINUE
      SB      = P/AREAB
      CALL TRGET (SB,EPSBB)
      DISPB  = TLENG*EPSBB
      BBDISP = BBDISP + DISPB
C
C...    SUM OF EACH ENERGY
C
      SUM     = SUM + P*DISPX + P*DISPB
      DEPTH(MXSTEP) = TODISP
      DEPTHB(MXSTEP)= BBDISP
      FORCE(MXSTEP) = P
      ACCEL(MXSTEP) = P/XWEIGH
      ENERG(MXSTEP) = SUM
      IF( P.GT.PCR )                GO TO 900
      IF( SUM.LT.ENERGY )           GO TO 100
      IF( MINI.NE.0 )               GO TO 800
  700 MINI    = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      SUM    = ENERG(MXSTEP)
      GO TO 100
C
C...    INTERPOLATION OF LAST STEP
C
  800 CONTINUE
      CALL FITSTP
      GO TO 1000
C
C...    PLATE IS PUNCTURED.
C
  900 CONTINUE
      MPUNCT = 1
      IF( MINI.EQ.0 )               GO TO 700
      CALL FITSTP
C...    END OF CALCULATION
 1000 CONTINUE
      IF( MXSTEP.GT.500 )           MXSTEP  = 500
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    SOLVE2
C
C****   /PURPOSE/
C         CALCULATION ENERGRY OF PUNCTURE
C           (SIMPLE-BENDING ANALYSIS)
C
C***    /PROGRAMMED BY JAERI        1990.12.06
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +           WEIGHT,    HEIGHT,    ROUT,      TLENG,     AA,
     +           DISP,      ENERGY,    THICK(3),  TAU(3),    PCR,
     +           MODEL,     KPLOT,     MTINN,     MTOUT,
     +           NTITLE(18)
      COMMON /MATDAT/      VALTAB(2,20,4),       LNGTAB(4),
     +           SIGM1,     SIGM2,     SIGM3
      COMMON /RESLT/
     +           DEPTH(500),           DEPTHB(500),          FORCE(500),
     +           ACCEL(500),           ENERG(500),           TODISP,
     +           MXSTEP,    MPUNCT
      COMMON /EQUIVL/      EQIVT,     PL
C
      DATA PAI/ 3.14159265400 /
C
C
C...    INITIAL STATUS
C
      XWEIGH = WEIGHT/9800.0
      MXSTEP = 0
      TODISP = 0.0
      BBDISP = 0.0
      SUM    = 0.0
      MINI   = 0
      DISPX  = DISP
      AREAB  = PAI*AA*AA
      UDISP  = 0.5D0
      B      = 1.0D0-2.0D0*AA/(3.0D0*ROUT)
      A1     = 4.0D0/3.0D0
      A2     = 1.0D0/6.0D0
C
      CALL GETEQV (EQIVMP,EQIVNP,EQIVT)
      PL     = 2.0D0*PAI*EQIVMP/B
      UDELT  = 1.0D0/EQIVT
C
C
C...    GET CURRENT FORCE
C
  100 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )          GO TO 1000
      TODISP = TODISP + DISPX
C
C
C. (DISP .VS. UDISP)
      U      = TODISP*UDELT
      IF( U.GT.UDISP )             GO TO 200
      P      = PL*( 1.0D0+A1*U*U )
```

## Appendix F (Continued)

```
      GO TO 300
C
  200 P      = PL*( 2.0D0*U+A2/U )
C
C
C. (TARGET DISP)
  300 CONTINUE
      SB     = P/AREAB
      CALL TRGET (SB,EPSBB)
C
      DISPB  = TLENG*EPSBB
      BBDISP = BBDISP + DISPB
C
C
C
C...    SUM OF EACH ENERGY
      SUM    = SUM + P*DISPX + P*DISPB
C
      DEPTH(MXSTEP) = TODISP
      DEPTHB(MXSTEP)= BBDISP
      FORCE(MXSTEP) = P
      ACCEL(MXSTEP) = P/XWEIGH
      ENERG(MXSTEP) = SUM
C
      IF( P.GT.PCR )                GO TO 900
      IF( SUM.LT.ENERGY )           GO TO 100
      IF( MINI.NE.0 )               GO TO 800
  700 MINI   = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      SUM    = ENERG(MXSTEP)
      GO TO 100
C
C...    INTERPOLATION OF LAST STEP
C
  800 CONTINUE
      CALL FITSTP
      GO TO 1000
C
C
C...    PLATE IS PUNCTURE.
C
  900 CONTINUE
      MPUNCT = 1
      IF( MINI.EQ.0 )               GO TO 700
      CALL FITSTP
C
C
C...    END OF CALCULATION
C
 1000 CONTINUE
      IF( MXSTEP.GT.500 )           MXSTEP  = 500
      RETURN
      END
```

```
      SUBROUTINE    SOLVE3
C
C****   /PURPOSE/
C         CALCULATION ENERGRY OF PUNCTURE
C           (FIXED-MEMBRANE ANALYSIS)
C
C***    /PROGRAMMED BY JAERI          1990.12.06
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,   HEIGHT,   ROUT,     TLENG,    AA,
     +          DISP,     ENERGY,   THICK(3), TAU(3),   PCR,
     +          MODEL,    KPLOT,    MTINN,    MTOUT,
     +          NTITLE(18)
      COMMON /MATDAT/     VALTAB(2,20,4),     LNGTAB(4),
     +          SIGM1,    SIGM2,    SIGM3
      COMMON /RESLT/
     +          DEPTH(500),         DEPTHB(500),        FORCE(500),
     +          ACCEL(500),         ENERG(500),         TODISP,
     +          MXSTEP,   MPUNCT
      COMMON /EQUIVL/     EQIVT,    PL
      DATA PAI/ 3.141592654D0 /
C
C...    INITIAL VALUE GET
C
      CALL  NEWTON (AA,ROUT,PP)
C
      XXPP    = DLOG(ROUT/PP)
      D1      = 1.0D0+2.0D0*XXPP
      D2      = 2.0D0+XXPP
      D3      = 1.0D0+XXPP
      D4      = 1.0D0+3.0D0*XXPP
C
      UDISP   = 0.5D0+0.5D0*XXPP
      A1      = D1/(D2*D3)
      A2      = 2.0D0*D4/(3.0D0*D2*D3*D3)
      B1      = (3.0D0+XXPP)/(2.0D0*D2)
      B2      = 2.0D0*D1/(D2*D3)
      B3      = D3/(12.0D0*D2)
C
      A       = D2
      B       = D3-2.0D0*AA/(3.0D0*PP)
      IF( PP.LT.AA )          B      = 0.5D0+DLOG(ROUT/AA)-PP*PP/
     +                                  (6.0D0*AA*AA)
C
C...    INITIAL STATUS
C
      XWEIGH = WEIGHT/9800.0
      MXSTEP = 0
      TODISP = 0.0
      BBDISP = 0.0
      SUM    = 0.0
      MINI   = 0
      DISPX  = DISP
      AREAB  = PAI*AA*AA
      A1     = 2.0D0*PAI
      A2     = 0.5D0+DLOG(ROUT/AA)
      BB     = A1/A2
```

```
      CALL GETEQV (EQIVMP,EQIVNP,EQIVT)
      PL     = 2.0D0*PAI*EQIVMP*A/B
C
C...    GET CURRENT FORCE
C
  100 CONTINUE
      MXSTEP = MXSTEP+1
      IF( MXSTEP.GT.500 )           GO TO 1000
      TODISP = TODISP + DISPX
C
C. (DISP .VS. UDISP)
C
      P      = BB*EQIVNP*TODISP
C
C. (TARGET DISP)
C
  300 CONTINUE
      SB     = P/AREAB
      CALL TRGET (SB,EPSBB)
C
      DISPB  = TLENG*EPSBB
      BBDISP = BBDISP + DISPB
C
C...     SUM OF EACH ENERGY
C
      SUM    = SUM + P*DISPX + P*DISPB
C
      DEPTH(MXSTEP) = TODISP
      DEPTHB(MXSTEP)= BBDISP
      FORCE(MXSTEP) = P
      ACCEL(MXSTEP) = P/XWEIGH
      ENERG(MXSTEP) = SUM
C
      IF( P.GT.PCR )                GO TO 900
      IF( SUM.LT.ENERGY )           GO TO 100
      IF( MINI.NE.0 )               GO TO 800
  700 MINI   = 1
      TODISP = TODISP - DISPX
      MXSTEP = MXSTEP-1
      DISPX  = 0.1*DISP
      SUM    = ENERG(MXSTEP)
      GO TO 100
C
C...     INTERPOLATION OF LAST STEP
C
  800 CONTINUE
      CALL FITSTP
      GO TO 1000
C...     PLATE IS PUNCTURE.
  900 CONTINUE
      MPUNCT = 1
      IF( MINI.EQ.0 )               GO TO 700
      CALL FITSTP
C...     END OF CALCULATION
 1000 CONTINUE
      IF( MXSTEP.GT.500 )           MXSTEP   = 500
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    GETEQV (EQIVMP,EQIVNP,EQIVT)
C
C****    /PURPOSE/
C          CALCULATION EQUIVALENT-(MP,NP,T)
C
C***     /OUTPUT/
C          EQIVMP    = EQUIVALENT (MP) VALUE
C          EQIVNP    = EQUIVALENT (NP) VALUE
C          EQIVT     = EQUIVALENT (T)  VALUE
C
C***      PROGRAMMED BY JAERI        1990.12.06
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,   HEIGHT,   ROUT,     TLENG,    AA,
     +          DISP,     ENERGY,   THICK(3), TAU(3),   PCR,
     +          MODEL,    KPLOT,    MTINN,    MTOUT,
     +          NTITLE(18)
      COMMON /MATDAT/     VALTAB(2,20,4),     LNGTAB(4),
     +          SIGM1,    SIGM2,    SIGM3
C
C...     EQUIVALENT-(NP)
C
      S1     = SIGM1*THICK(1)
      S2     = SIGM2*THICK(2)
      S3     = SIGM3*THICK(3)
      EQIVNP = S1+S2+S3
C
C...     (A)-REGION
C
      T1     = THICK(1)
      T2     = THICK(2)
      T3     = THICK(3)
      H      = 0.5D0
      IF( S1.LE.(S2+S3) )      GO TO 100
C
      TT     = (S1+S2+S3)/(2.0D0*SIGM1)
      EQIVMP = H*TT*TT*SIGM1 + H*(T1-TT)*(T1-TT)*SIGM1 +
     +         T2*(T1-TT+H*T2)*SIGM2 + T3*(T2+T1-TT+H*T3)*SIGM3
      ISW    = 1
      GO TO 500
C
C...     (B)-REGION
C
  100 IF( (S1+S2).LE.S3 )      GO TO 200
C
      TT     = (-S1+SIGM2*T1+SIGM2*(T1+T2)+SIGM3*T3)/(2.0D0*SIGM2)
      EQIVMP = T1*(TT-H*T1)*SIGM1 + H*(TT-T1)*(TT-T1)*SIGM2 +
     +         H*(T2+T1-TT)*(T2+T1-TT)*SIGM2 + T3*(T2+T1-TT+H*T3)*SIGM3
      ISW    = 2
      GO TO 500
C
C...     (C)-REGION
C
  200 CONTINUE
      TT     = (-S1-S2+SIGM3*(T1+T2)+SIGM3*(T1+T2+T3))/(2.0D0*SIGM3)
      EQIVMP = T1*(TT-H*T1)*SIGM1 + T2*(TT-T1-H*T2)*SIGM2 +
     +         H*(TT-T1-T2)*(TT-T1-T2)*SIGM3 +
```

## Appendix F (Continued)

```
     +        H*(T3+T2+T1-TT)*(T3+T2+T1-TT)*SIGM3
      ISW    = 3
  500 CONTINUE
      EQIVT  = 4.0D0*EQIVMP/EQIVNP
      RETURN
      END



      SUBROUTINE    FITSTP
C
C****     /PURPOSE/
C           RESET CALCULATED ENERGRY OF LAST STEP
C
C***      PROGRAMMED BY JAERI        1989.10.09
C
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,   HEIGHT,   ROUT,     TLENG,    AA,
     +          DISP,     ENERGY,   THICK(3), TAU(3),   PCR,
     +          MODEL,    KPLOT,    MTINN,    MTOUT,
     +          NTITLE(18)
      COMMON /RESLT/
     +          DEPTH(500),         DEPTHB(500),        FORCE(500),
     +          ACCEL(500),         ENERG(500),         TODISP,
     +          MXSTEP,   MPUNCT
C
      DATA EPS/ 0.05 /
C
C...       FUNCTION
      ABS(Z) = DABS(Z)
C
C...
C
      GO TO 200
C
  100 MXSTEP = MXSTEP-1
C
  200 CONTINUE
      NXSTEP = MXSTEP-1
      DL1    = FORCE(MXSTEP)-FORCE(NXSTEP)
      DL2    = PCR          -FORCE(NXSTEP)
      IF( ABS(DL2).LE.EPS )      GO TO 100
      DLT    = DL2/DL1
C
      DEPTH(MXSTEP) = DLT*(DEPTH(MXSTEP)-DEPTH(NXSTEP)) + DEPTH(NXSTEP)
      ACCEL(MXSTEP) = DLT*(ACCEL(MXSTEP)-ACCEL(NXSTEP)) + ACCEL(NXSTEP)
      ENERG(MXSTEP) = DLT*(ENERG(MXSTEP)-ENERG(NXSTEP)) + ENERG(NXSTEP)
      FORCE(MXSTEP) = PCR
C
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    NEWTON ( AA,ROUT,PP )
C
C****    /PURPOSE/
C          GET PP VALUE WITH NEWTON-RAPSON
C
C***     /OUTPUT/
C          PP         = DISCONTINUITY RADIUS OF THE VELOCITY CURVATURE
C
C***     /INPUT/
C          AA         = RADIUS OF TARGET
C          ROUT       = RADIUS OF CASK
C
C***     /PROGRAMMED BY JAERI       1989.10.06
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      DATA EPS/ 1.0D-10 /
      DATA D606/ 0.60653066D0 /
C
C
C...     INITIAL SET
C
      NITER   = 100
      LOOP    = 0
      A1      = 1.0D0
      A2      = 2.0D0
      A3      = 3.0D0
      CONST   = A1+A2*DLOG(ROUT/AA)
      PP      = 0.1D0
      FACT    = AA/ROUT
      IF( FACT.GT.D606 )            GO TO 1000
C
C. CASE (1)
  100 LOOP    = LOOP+1
      F1      = DLOG(ROUT/PP)
      F2      = A2*AA/(A3*PP)
      F3      = A2*AA/(A3*PP*PP)
      FUNC1   = A1-F2*(A1+F1)
      FUNC2   = F3*(A2+F1)
C
      PP2     = PP - FUNC1/FUNC2
      IF( DABS(PP-PP2).LE.EPS )     GO TO 2000
      IF( LOOP.GT.NITER )           GO TO 2000
      PP      = PP2
      GO TO 100
C
C. CASE (2)
 1000 LOOP    = LOOP+1
      F1      = A2/A3
      F2      = A2*AA*AA/(PP*PP*PP)
      F3      = AA*AA/(PP*PP)
      F4      = DLOG(ROUT/PP)
      FUNC1   = A1-F3*CONST+F1*(A1+F4)
      FUNC2   = F2*CONST-F1/PP
C
      PP2     = PP - FUNC1/FUNC2
      IF( DABS(PP-PP2).LE.EPS )     GO TO 2000
      IF( LOOP.GT.NITER )           GO TO 2000
```

## Appendix F (Continued)

```
      PP      = PP2
      GO TO 1000
C
C.  PP-DATA
 2000 CONTINUE
      PP      = PP2
      RETURN
      END



      SUBROUTINE    TRGET  (SIGM,EPS)
C
C****    /PURPOSE/
C          GET TARGET STRAIN(EPS)
C
C***     /OUTPUT/
C          EPS     :TARGET STRAIN
C
C***     /INPUT/
C          SIGM    :TARGET STRESS
C
C***      PROGRAMMED BY T.ISHIWATA           1990.12.06
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      COMMON /GEOMDT/
     +          WEIGHT,    HEIGHT,    ROUT,      TLENG,     AA,
     +          DISP,      ENERGY,    THICK(3),  TAU(3),    PCR,
     +          MODEL,     KPLOT,     MTINN,     MTOUT,
     +          NTITLE(18)
      COMMON /MATDAT/      VALTAB(2,20,4),       LNGTAB(4),
     +          SIGM1,     SIGM2,     SIGM3
C
C
C...     STRESS RAMGE (J1-J2)
C
      LENGTH = LNGTAB(4)
      DO 100 J2= 1, LENGTH
      IF( SIGM.LE.VALTAB(2,J2,4) )      GO TO 200
  100 CONTINUE
      J2      = LENGTH
C
C...     GET STRAIN
  200 J1      = J2-1
      IF( J1.GT.0 )                     GO TO 300
      S1      = 0.0D0
      E1      = 0.0D0
      GO TO 500
  300 S1      = VALTAB(2,J1,4)
      E1      = VALTAB(1,J1,4)
C
  500 S2      = VALTAB(2,J2,4)
      E2      = VALTAB(1,J2,4)
      EPS     = (E2-E1)/(S2-S1)*(SIGM-S1) + E1
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    DPLOT ( MODEL, NTITLE, XVAL, YVAL, LENGTH, KTYPE )
C
C****    /PURPOSE/
C          XYPLOT PROGRAM
C
C***    /INPUT/
C          MODEL      = ANALYSIS TYPE NUMBER
C          NTITLE     = MODEL TITLE MESSAGE
C          XVAL       = X VARIABLES
C          YVAL       = Y VARIABLES
C          LENGTH     = NO. OF VARIABLES
C          KTYPE      = 1   DEFORMATION .VS. ENERGY PLOT
C                     = 2   DEFORMATION .VS. ACCELERATION PLOT
C                     = 3   DEFORMATION .VS. FORCE PLOT
C                     = 4   FORCE.VS. ACCELERATION PLOT
C                     = 5   DEFORMATION .VS. FORCE PLOT
C                                     (NONDIMENSIONAL VALUE)
C                     = 6   BAR-DEFORMATION .VS. FORCE PLOT
C                     = 7   BAR-DEFORMATION .VS. BAR-STRESS PLOT
C
C***     PROGRAMMED BY JAERI                  1989.03.10
C***    /MODIFIED   BY JAERI                  1990.12.06
C
      IMPLICIT      REAL*8 (A-H,O-Z)
      DIMENSION NTITLE(18)
      DIMENSION XVAL(1),  YVAL(1)
      DIMENSION X(5),      Y(5),      NP(2,7),  KSCL(2),  KHEAD(3,5),
     +          MHEAD(2,5), K2HEAD(2), KEYMDL(4,3)
      DATA NP/        1, 2,   1, 3,   1, 4,   4, 3,   1, 4,
     +                1, 4,   1, 5/
      DATA KHEAD/  4HDEFO, 4HRMAT, 4HION ,
     +             4HENER, 4HGY  , 4H    ,
     +             4HACCE, 4HLERA, 4HTION,
     +             4HFORC, 4HE   , 4H    ,
     +             4HBAR-, 4HSTRE, 4HSS  /
      DATA K2HEAD/ 4HNODI, 4HM.  /
      DATA MHEAD/  4H  MM, 4H    ,              4H KG-, 4HMM  ,
     +             4H G  , 4H    ,              4H KG , 4H    ,
     +             4H KG/, 4HMM*2/
      DATA KSCL/   4H(10 , 4H )  /
      DATA NPARTX,NPARTY/  10, 8 /
      DATA XL,YL/  200.0,  160.0 /
      DATA XO,YO,DH/ 40.0, 30.0, 20.0 /
      DATA HH,H2,H3/  2.5,  2.0,  1.5 /
      DATA KEYMDL/ 4H(FIX, 4HED-B, 4HEND), 4H    ,
     +             4H(SIM, 4HPLE-, 4HBEND, 4H)   ,
     +             4H(FIX, 4HED-M, 4HEMBR, 4HANE)/
C
C...    DATA RANGE
C
      CALL GOSCAL ( LENGTH, XVAL, YVAL, XMN, YMN, XMX, YMX, FACT )
      CALL XYSCAL ( XMN, XMX, NPARTX, DX, NEXPX )
      CALL XYSCAL ( YMN, YMX, NPARTY, DY, NEXPY )
C
C...    GRID MESH
C
      CALL PLPEN( 1 )
      X(1)    = XO
```

## Appendix F (Continued)

```
      X(2)    = XO+XL
      X(3)    = XO+XL
      X(4)    = XO
      X(5)    = XO
      Y(1)    = YO
      Y(2)    = YO
      Y(3)    = YO+YL
      Y(4)    = YO+YL
      Y(5)    = YO
      CALL PLINES ( X, Y, 5 )
C. (X) SCALE
      Y(1)    = YO-2.0
      Y(2)    = YO+2.0
      Y(3)    = YO-5.0
      Y(4)    = YO+YL-2.0
      Y(5)    = YO+YL
      X(1)    = XO
      X(2)    = XO
      DNO     = 0.0
      LPARTX = NPARTX+1
      DO 300 J= 1, LPARTX
      CALL PLINES ( X, Y, 2 )
      CALL PLINES ( X, Y(4), 2)
      X(3)    = X(1)-H2
      CALL PNUMB  ( X(3), Y(3), H2, DNO, 0.0D0, 1 )
      X(1)    = X(1)+DH
      X(2)    = X(2)+DH
      DNO     = DNO+DX
  300 CONTINUE
      IF( NEXPX.EQ.0 )                GO TO 350
      X(3)    = XO+XL+4.0*H2
      CALL PSYM   ( X(3), Y(3), H2, KSCL, 0.0D0, 6 )
      X(3)    = XO+XL+7.0*H2
      Y(3)    = Y(3)+H3
      CALL PNUMBR ( X(3), Y(3), H3, NEXPX, 0.0D0 )
C. (Y) SCALE
  350 X(1)    = XO-2.0
      X(2)    = XO+2.0
      X(3)    = XO-5.0*H2
      X(4)    = XO+XL-2.0
      X(5)    = XO+XL
      Y(1)    = YO
      Y(2)    = YO
      DNO     = 0.0
      LPARTY = NPARTY+1
      DO 400 J= 1, LPARTY
      CALL PLINES ( X, Y, 2 )
      CALL PLINES ( X(4), Y, 2 )
      Y(3)    = Y(1)
      CALL PNUMB  ( X(3), Y(3), H2, DNO, 0.0D0, 1 )
      Y(1)    = Y(1)+DH
      Y(2)    = Y(2)+DH
      DNO     = DNO+DY
  400 CONTINUE
      IF( NEXPY.EQ.0 )                GO TO 500
      X(3)    = XO
      Y(3)    = YO+YL+H2
      CALL PSYM   ( X(3), Y(3), H2, KSCL, 0.0D0, 6 )
```

## Appendix F (Continued)

```
      X(3)    = XO+3.0*H2
      Y(3)    = Y(3)+H3
      CALL PNUMBR ( X(3), Y(3), H3, NEXPY, 0.0D0 )
C
C...    HEADER PLOT
C
  500 CONTINUE
      XX      = XO+20.0*HH
      YY      = YO+YL+3.0*HH
      CALL PSYM    ( XX, YY, HH, NTITLE, 0.0D0, 70 )
C. (X-TITLE)
      IPX     = NP(1,KTYPE)
      XX      = XO+20.0*HH
      YY      = 0.6*YO
      CALL PSYM    ( XX, YY, HH, KHEAD(1,IPX), 0.0D0, 12 )
      XX      = XX+15.0*HH
      CALL PSYM    ( XX, YY, HH, MHEAD(1,IPX), 0.0D0, 8 )
C. ANALYSIS TITLE
      XX      = XO+66.0*HH
      CALL PSYM    ( XX, YY, HH, KEYMDL(1,MODEL), 0.0D0, 12 )
      IF( KTYPE.NE.5 )              GO TO 510
      XX      = XO+14.0*HH
      CALL PSYM    ( XX, YY, HH, K2HEAD,       0.0D0, 6 )
C. (Y-TITLE)
  510 IPY     = NP(2,KTYPE)
      XX      = 0.4*XO
      YY      = YO+20.0*HH
      CALL PSYM    ( XX, YY, HH, KHEAD(1,IPY), 90.0D0, 12 )
      YY      = YY+15.0*HH
      CALL PSYM    ( XX, YY, HH, MHEAD(1,IPY), 90.0D0, 8 )
      IF( KTYPE.NE.5 )              GO TO 600
      YY      = YO+14.0*HH
      CALL PSYM    ( XX, YY, HH, K2HEAD,       90.0D0, 6 )
C
C...    XYPLOT
C
  600 CONTINUE
      CALL PLPEN( 2 )
      PX      = DX* 10.0**NEXPX
      PY      = DY* 10.0**NEXPY
      XFACT   = DH/PX
      YFACT   = DH/PY
C
      X(1)    = XO
      Y(1)    = YO
      IF( KTYPE.EQ.5 )              Y(1)    = Y(1) + YFACT
      R       = 1.0
      DO 700 J= 1, LENGTH
      X(2)    = XFACT*XVAL(J) + XO
      Y(2)    = YFACT*YVAL(J) + YO
      CALL PLINES ( X, Y, 2 )
      CALL PLMARK ( X(2), Y(2), R )
      X(1)    = X(2)
      Y(1)    = Y(2)
  700 CONTINUE
      CALL PLTEOR
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    GOSCAL ( ND, X, Y, XMN, YMN, XMX, YMX, SCALEG )
C
C****   /PURPOSE/
C         GET SCALING FACTOR OF THE GEOMETRY PLOT
C
C
C***    /OUTPUT/
C         XMN         = X MINIMUM
C         YMN         = Y MINIMUM
C         XMX         = X MAXIMUM
C         YMX         = Y MAXIMUM
C         SCALEG      = SCALING FACTOR
C
C***    /INPUT/
C         ND          = NO. OF NODES
C         X           = X COORNINATES
C         Y           = Y COORNINATES
C
C***     PROGRAMMED BY JAERI        1989.03.10
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION X(1),      Y(1)
      DATA PAPER/ 180.0 /
C
C...     FUNCTION
C
      AMIN1 (Z1,Z2) = DMIN1 (Z1,Z2)
      AMAX1 (Z1,Z2) = DMAX1 (Z1,Z2)
C
C...     GEOMETRY RANGE
C
      XMN     =  1.0D20
      YMN     =  1.0D20
      XMX     = -1.0D20
      YMX     = -1.0D20
C
C...  MAX-MIN
C
      DO 100 I= 1, ND
      XI      = X(I)
      XMN     = AMIN1 (XMN,XI)
      XMX     = AMAX1 (XMX,XI)
      YI      = Y(I)
      YMN     = AMIN1 (YMN,YI)
      YMX     = AMAX1 (YMX,YI)
  100 CONTINUE
C
C...  SCALE
C
      XS      = XMX-XMN
      YS      = YMX-YMN
      AS      = AMAX1 (XS,YS)
      SCALEG  = PAPER/AS
      RETURN
      END
```

## Appendix F (Continued)

```
      SUBROUTINE    XYSCAL ( VMIN, VMAX, NPART, DSIZE , NEXP )
C
C****   /PURPOSE/
C         XYPLOT SCALING
C
C***    /OUTPUT/
C         DSIZE      = ADDITIONAL VALUE
C         NEXP       = EXPONENT NUMBER
C
C***    /INPUT/
C         VMIN       = MINIMUM VALUE
C         VMAX       = MAXIMUM VALUE
C         NPART      = NO. OF DIVISION
C
C***     PROGRAMMED BY JAERI       1989.03.10
C
C
      IMPLICIT     REAL*8 (A-H,O-Z)
      DIMENSION DVAL(8)
      DATA EPS/  1.0D-20 /
      DATA DVAL/ 0.5, 1.0, 1.5, 2.0, 2.5, 5.0, 10.0, 15.0 /
C
C...    FUNCTION
      ABS(Z)     = DABS(Z)
      ALOG10(Z) = DLOG10(Z)
C
C...
C
      DPART  = NPART
      XMIN   = 0.0
      DS     = (VMAX-XMIN)/DPART
      IF( DS.LT.EPS )           GO TO 900
      NEXP   = ALOG10 (DS)
      IF( NEXP.LT.0 )           NEXP   = NEXP-1
      GVAL   = 10.0**NEXP
C
      HWIDE  = DS/GVAL
      DO 100 J= 1, 7
      IF( HWIDE.LE.DVAL(J) )   GO TO 200
  100 CONTINUE
      J      = 7
  200 DSIZE  = DVAL(J)
C
      RETURN
C
C
C
  900 CONTINUE
      NEXP   = 0
      DSIZE  = DVAL(2)
      AA     = ABS(XMIN)
      IF( AA.LT.EPS )           GO TO 950
      NEXP   = ALOG10 (AA)
      IF( NEXP.LT.0 )           NEXP   = NEXP-1
C
  950 CONTINUE
      RETURN
      END
```

## Appendix F (Continued)

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


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


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


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

## Appendix F (Continued)

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


      SUBROUTINE     PLTEOR
      IMPLICIT      REAL*8 (A-H,O-Z)
CRC   CALL  PLOT (300.0D0,0.0D0,-3)
      CALL  PLOT ( 0.0D0, 0.0D0, 666 )
      CALL  PLOT ( 0.0D0, 0.0D0, 777 )
      CALL  PLOT ( 0.000, 0.0D0, 888 )
      RETURN
      END


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