JAERI-Data/Code
95-017

ROCKING : A COMPUTER PROGRAM FOR SEISMIC RESPONSE ANALYSIS OF RADIOACTIVE MATERIALS TRANSPORT AND/OR STORAGE CASKS

November 1995

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　㈱原子力資料サービス

ROCKING : A Computer Program For Seismic Response Analysis of Radioactive Materials Transport AND/OR Storage Casks

Takeshi IKUSHIMA

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

(Received October 5, 1995)

The computer program ROCKING has been developed for seismic response analysis, which includes rocking and sliding behavior, of radioactive materials transport and/or storage casks.

Main features of ROCKING are as follows;

(1) Cask is treated as a rigid body.
(2) Rocking and sliding behavior are considered.
(3) Impact forces are represented by the spring dashpot model located at impact points.
(4) Friction force is calculated at interface between a cask and a floor.
(5) Forces of wire repes against tip-over work only as tensile loads.

In the paper, the calculation model, the calculation equations, validity calculations and user's manual are shown.

Keywords: Computer Program, Seismic Response Analysis, Rocking Analysis, Structural Analysis, Cask, Transport Cask, Storage Cask

ROCKING：放射性物質輸送・貯蔵容器の地震応答解析プログラム

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

（1995年10月5日受理）

放射性物質輸送・貯蔵容器の地震時のロッキングや滑り挙動を解析するために，計算プログラムROCKINGを開発した。

ROCKINGの主要な特徴は次のとおりである。

(1) 輸送容器は剛体として取り扱う。

(2) 輸送容器のロッキングおよび輸送容器と床面との間滑り挙動を考慮する。

(3) 衝突力はばねとダッシュポットによって取り扱う。

(4) 輸送容器と床面との間の摩擦力を考慮する。

(5) 転倒防止用のワイヤーロープに加わる力は引張力のみを考慮する。

本報告書では，計算モデル，計算式，検証計算，計算プログラムの使用マニュアルについて記述されている。

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

In the stand point of seismic safety of transport and/or storage casks for spent fuels, it is necessary to evaluate integrity of casks against seismic loads. Therefore, it is important to clarify the tip-over behavior of casks during seismic loads using experimental and analytical methods.

Central Research Institute of Electric Power Industry(CRIEPI)[1],[2] has studied seismic response behavior of casks experimentally and analytically. In this analysis the Distinct Element Method(DEM) was used for tip-over behavior of casks. However, the considerable cost and calculation time are necessitated to perform such analysis by the DEM.

Therefore, simplified computer programs which are capable of reducing cost and calculation time are needed to perform parametric calculation in designing casks and conducting safety analysis. To meet the above requirements, the computer program ROCKING[3] has been developed by Japan Atomic Energy Research Institute as shown in Fig.1.1.

In the program, a cask with (or without) tie-down wire ropes is modeled by a vertical cylindrical rigid body tied down with four non-linear springs having a certain tensile strength. An explicit or a semi-implicit numerical integration methods are adopted to predict whether a cask will tip-over or not, and tie-down ropes will break or not. Moreover, in the ROCKING program, sliding between a cask and a floor and vertical seismic load are treated.

The results of the ROCKING program are compared with experimental ones conducted by CRIEPI. Both analytical and experimental results fairly agree each others. The ROCKING program can accurately estimate tip-over behavior of casks during seismic loads.

In the paper, Chapter 2 presents an illustration of calculation model. Chapter 3 presents comparison between ROCKING and experimental results. Chapter 4 provides an user's guide for ROCKING.

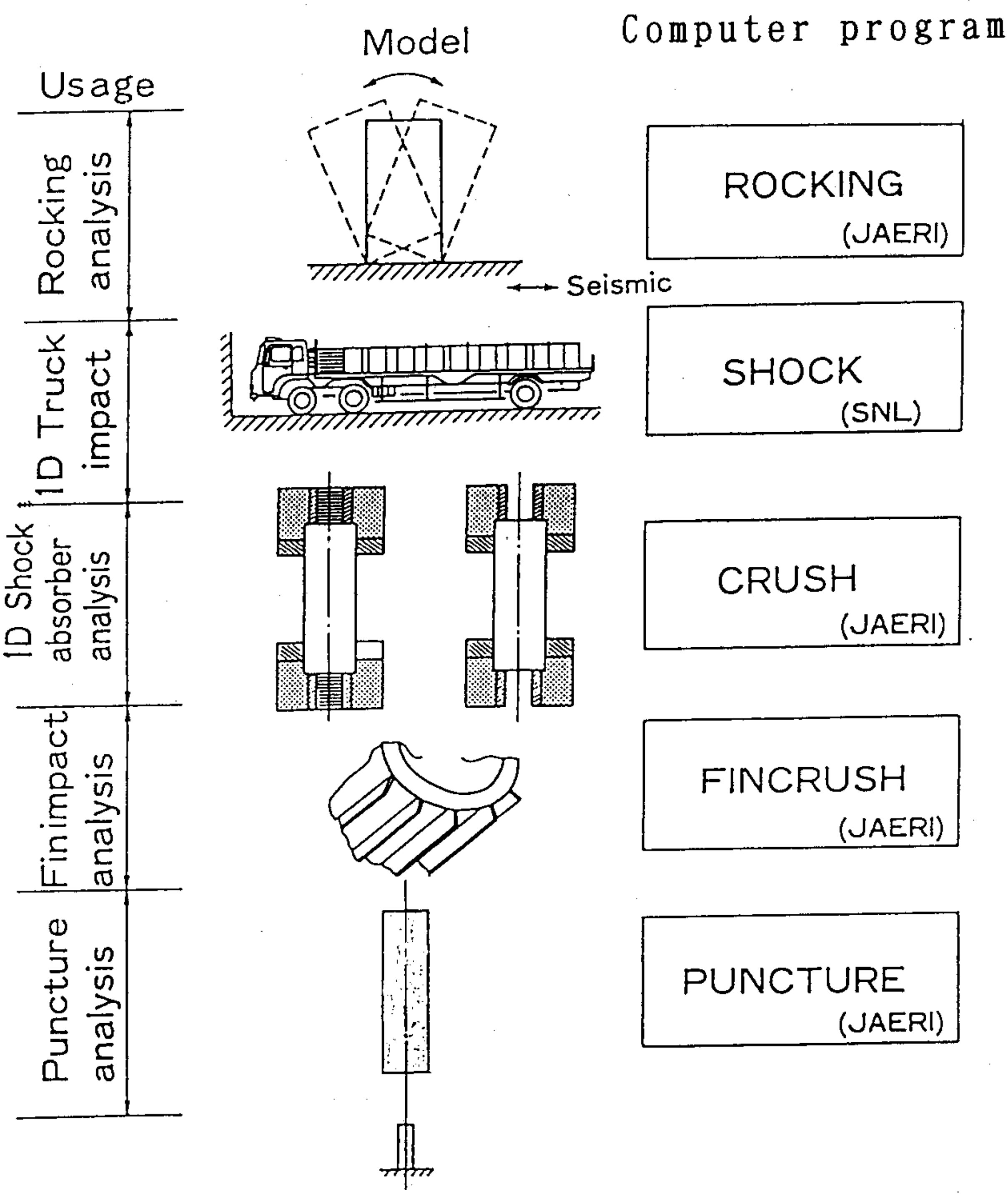


Fig. 1.1 Computer code

2. Calculation equation

2.1 Calculation model

In the computer program ROCKING, the following two-dimensional model is considered:

(1) a cask is modeled as a rigid body,

(2) the cask has three degrees-of-freedom, two translational displacements and one rotation around the cask center of gravity,

(3) impact forces are represented by a spring and dashpot model located at impact points,

(4) friction force due to surface sliding between the cask and a floor is represented by a nonlinear Coulomb element, and

(5) forces of wire ropes work against only as tensile loads.

2.2 Equation of governing motion

Let the coordinate system be chosen as shown in Fig. 2.1. The cask has two translational coorinates, ξ and η, and one rotational coordinate θ. Figure 2.2 shows the forces and moments that act upon the cask. The equations of motion may be written as:

$$m\ddot{\xi} = -m\alpha_{\xi} + F_{\xi A} + F_{\xi B} + F_{\xi C} + F_{\xi D} + R_{\xi} + F_{F} \tag{2.1}$$

$$F_{\eta A} + F_{\eta B} + F_{\eta C} + F_{\eta D} = 0 \tag{2.2}$$

$$m\ddot{z} = -m(\alpha_{z} + g) + F_{zA} + F_{zB} + F_{zC} + F_{zD} + R_{z} \tag{2.3}$$

$$M_{\xi A} + M_{\xi B} + M_{\xi C} + M_{\xi D} = 0 \tag{2.4}$$

$$I\ddot{\theta} = M_{\eta A} + M_{\eta B} + M_{\eta C} + M_{\eta D} + M_{R\xi} + M_{Rz} + M_{F} \tag{2.5}$$

$$M_{zA} + M_{zB} + M_{zC} + M_{zD} = 0 \tag{2.6}$$

where

a : cask rocking spring half width,

b : radius of cask bottom,

d : cask outer radius,

F_F : friction force between cask and floor,

F_z : z -component of force,

F_{zA} : z -component of force generated by wire rope A,

F_{zB} : z -component of force generated by wire rope B,

F_{zC} : z -component of force generated by wire rope C,

F_{zD} : z -component of force generated by wire rope D,

F_η : η -component of force,

$F_{\eta A}$: η -component of foece generated by wire rope A,

$F_{\eta B}$: η -component of force generated by wire rope B,

$F_{\eta C}$: η -component of force generated by wire rope C,

$F_{\eta D}$: η -component of force generated by wire rope D,

F_ξ : ξ -component of force,

$F_{\xi A}$: ξ -component of force generated by wire rope A,

$F_{\xi B}$: ξ -component of force generated by wire rope B,

$F_{\xi C}$: ξ -component of force generated by wire rope C,

$F_{\xi D}$: ξ -component of force generated by wire rope D,

g : gravity constant,

h : half height of cask,

I : mass moment of inertia,

m : mass of cask,

M_F : moment generated by force F_F,

M_{Rz} : moment generated by force R_z.

$M_{R\xi}$: moment generated by force R_ξ,

M_z : moment generated by force F_z,

M_{zA} : moment generated by force F_{zA}.

M_{zB} : moment generated by force F_{zB},

M_{zC} : moment generated by force F_{zC},

M_{zD} : moment generated by force F_{zD},

M_η : moment generated by force F_η,

$M_{\eta A}$: moment generated by force $F_{\eta A}$,

$M_{\eta B}$: moment generated by force $F_{\eta B}$,

$M_{\eta C}$: moment generated by force $F_{\eta C}$,

$M_{\eta D}$: moment generated by force $F_{\eta D}$,

M_{ξ} : moment generated by force F_{ξ},

$M_{\xi A}$: moment generated by force $F_{\xi A}$,

$M_{\xi B}$: moment gerarated by force $F_{\xi B}$,

$M_{\xi C}$: moment generated by force $F_{\xi C}$,

$M_{\xi D}$: moment generated by force $F_{\xi D}$,

R_z : impact force of z-direction,

R_{ξ} : impact force of ξ-direction,

z : z-coordinate or z-direction displacement,

$\dot{z}$: z-component of velocity,

$\ddot{z}$: z-component of acceleration,

α_z : vertical floor acceleration,

α_{ξ} : horizontal floor acceleration,

η : η-coordinate or η-direction displacement,

θ : rotational angle,

$\dot{\theta}$: angular velocity,

$\ddot{\theta}$: angular acceleration,

ν : velocity,

ξ : ξ-coordinate or ξ-direction displacement,

$\dot{\xi}$: ξ-component of velocity.

$\ddot{\xi}$: ξ-component of acceleration.

2.3 Friction force between cask and floor and its associated moment

The friction force due to surface sliding is represented by a nonlinear Coulomb element. The equations for the friction force F_F and its associated moment M_F acting on the cask are as follows.

$$F_F = -\mathrm{sign}(\nu)F(\nu) , \qquad (2.7)$$

$$M_F = F_F(-h \cdot \cos\theta - b \cdot \sin\theta) , \qquad (2.8)$$

where

$$\nu = \dot{\xi} - (h \cdot \cos\theta + b \cdot \sin\theta)\,\dot{\theta} - \dot{\xi}_o. \qquad (2.9)$$

Where $\dot{\xi}$ and $\dot{\xi}_o$ are horizontal velocity of the center of gravity and support floor, respectively. $F(\nu)$ is a prescribed function for the friction characteristics which are related to the vertical contact force and the coefficients of both statical and dynamical friction.

$$F(\nu) = m \cdot g\,(\mu_d + \mu_s), \qquad (2.10)$$

Where μ_d and μ_s are the coefficients of dynamical amd statical frictions, respectively.

2.4 Vertical impact force and its associated moment

The force acting on the interface between the cask and floor is derived in the term of deformation of a spring dashpot unit. When the gap is closing, the spring deformation γ and its time rate $\dot{\gamma}$ are

$$\gamma = 0.5(z_o - \{z + h\,(1 - \cos\theta) - b \cdot \sin\theta\}\,] \qquad (2.11)$$

$$\dot{\gamma} = 0.5\,[\dot{z}_o - \{\dot{z} + h \cdot \sin\theta - b \cdot \cos\theta\}\,\dot{\theta}\,] \qquad (2.12)$$

where b is the cask rocking spring half width. z_o and $\dot{z}_o$ are the vertical displacement and velocity of the floor, respectively. The vertical impact force R_z and its associated moment M_{Rz} acting on the cask are as follows. If $\gamma > 0$

$$R_z = -\,(K_z \cdot \gamma + C_z \cdot \gamma)\,, \qquad (2.13)$$

$$M_{Rz} = -R_z\,(h \cdot \cos\theta - a \cdot \sin\theta)\,. \qquad (2.14)$$

where K_z and C_z are the vertical boundary spring and damping coefficients, respectively. If $\tau \leqq 0$

If $\gamma \leqq 0$

$$R_z = 0, \qquad (2.15)$$

$$M_{Rz} = 0. \qquad (2.16)$$

2.5 Lower corner impact force and its associated moment

The force acting on the cask lower corner as a results of the impact is derived by deformation of a spring dashpot unit which is located on the lower corner of the cask. During impact against the lower corner, spring deformation τ and its time rate $\dot{\tau}$ of the cask are follows.

$$\tau = \xi + h \cdot \sin\theta - b\ (1 - \cos\theta) - \xi_o - \delta, \qquad (2.17)$$

$$\dot{\tau} = \dot{\xi} + (h \cdot \cos\theta - b \cdot \sin\theta)\ \dot{\theta} - \dot{\xi}_o, \qquad (2.18)$$

where δ is the gap between the cask and its lower boundary wall. The boundary wall impact force R_ℓ and its associated moment $M_{R\ell}$ acting on the cask are as follows. If $\tau > 0$

$$R_\ell = -(K_\ell \cdot \tau + C_\ell \cdot \tau), \qquad (2.19)$$

$$M_{R\ell} = R_\ell\ (h \cdot \cos\theta - b \cdot \sin\theta), \qquad (2.20)$$

where K_ℓ and C_ℓ are the horizontal boundary spring and damping coefficients, respectively. If $\tau \leqq 0$

$$R_\ell = 0, \qquad (2.21)$$

$$M_{R\ell} = 0. \qquad (2.22)$$

2.6 Force due to wire rope

A tensile force due to a wire rope is as follows:

$$F = \sigma \cdot A, \qquad (2.23)$$

where

F : tensile force of wire rope,

σ : tensile stress of wire rope,

$$\sigma = E \cdot \Delta L / L, \qquad (2.24)$$

A : sectional area of wire rope,

E : Young's modulus,

ΔL : elongation of wire rope,

L : wire rope length.

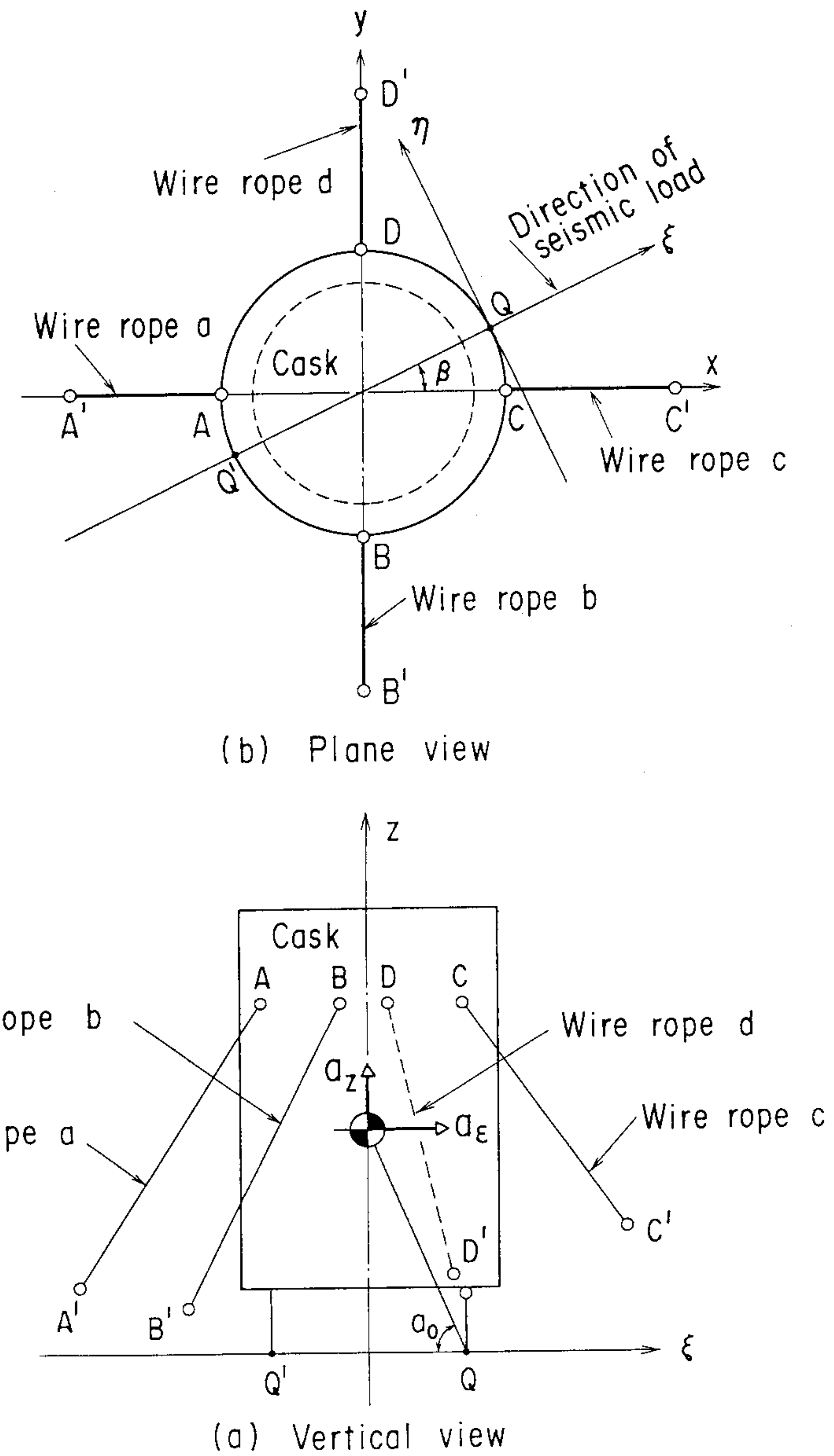
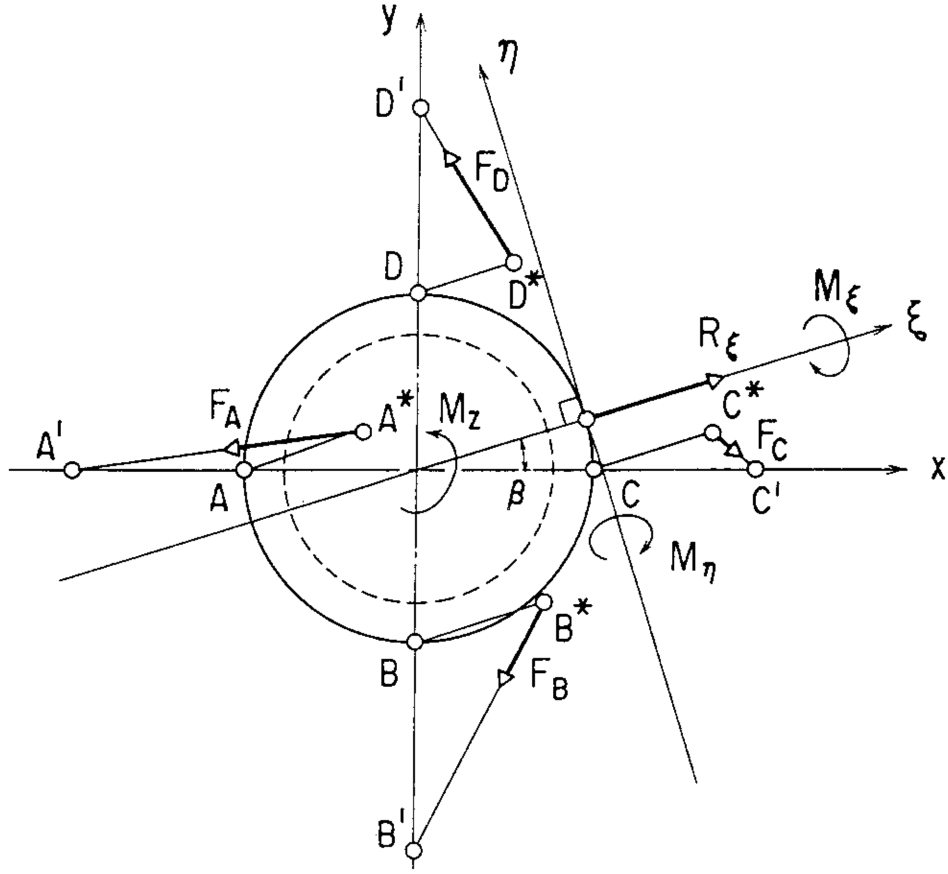
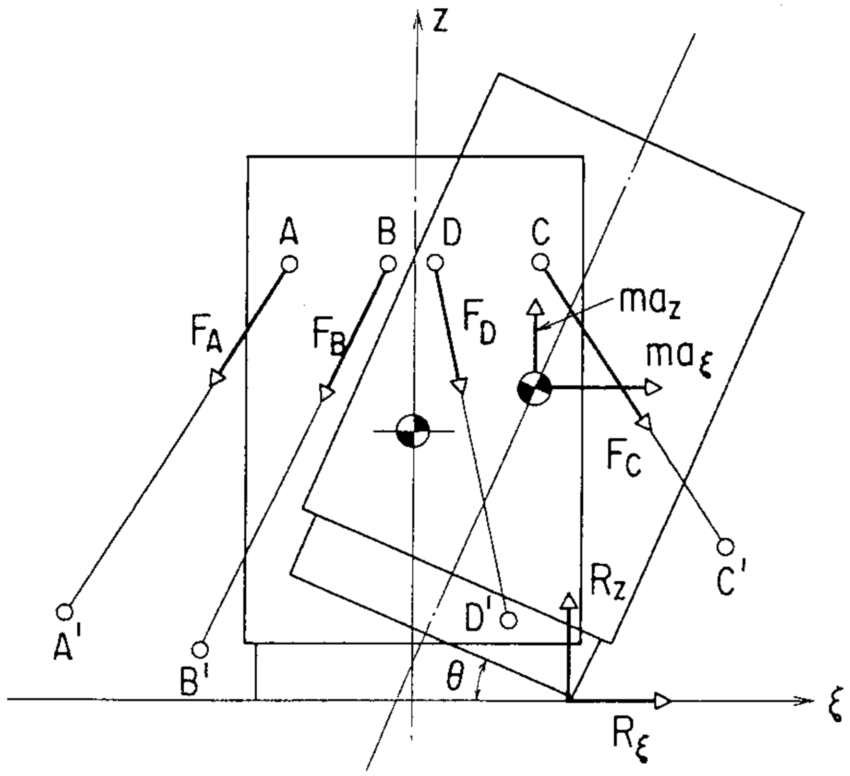


Fig. 2.1 Coordinate system



(b) Plane view



(a) Vertical view

Fig. 2.2 Forces and moments acting on cask

3. Benchmark caluculation

In order to demonstrate the adequacy of ROCKING, benchmark calculations using the CRIEPI experimental results of the 1/3 scale model of spent fuel cask as shown in Fig. 3.1 have been performed.

The rocking analysis of the model cask was performed under constant sinusoidal excitations. Figure 3.2 shows the maximum rotational angles in comparison with the experimental results. According to Fig.3.2 analytical results of ROCKING agree with the experimental ones.

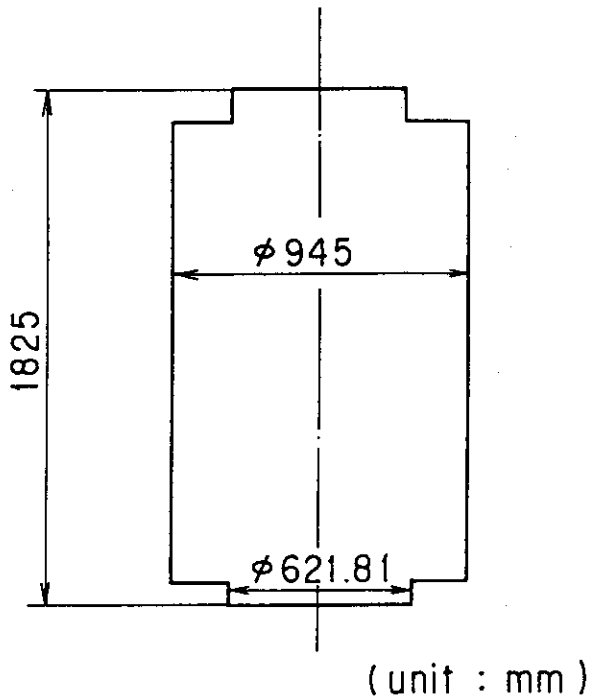


Fig. 3.1 Analysis model

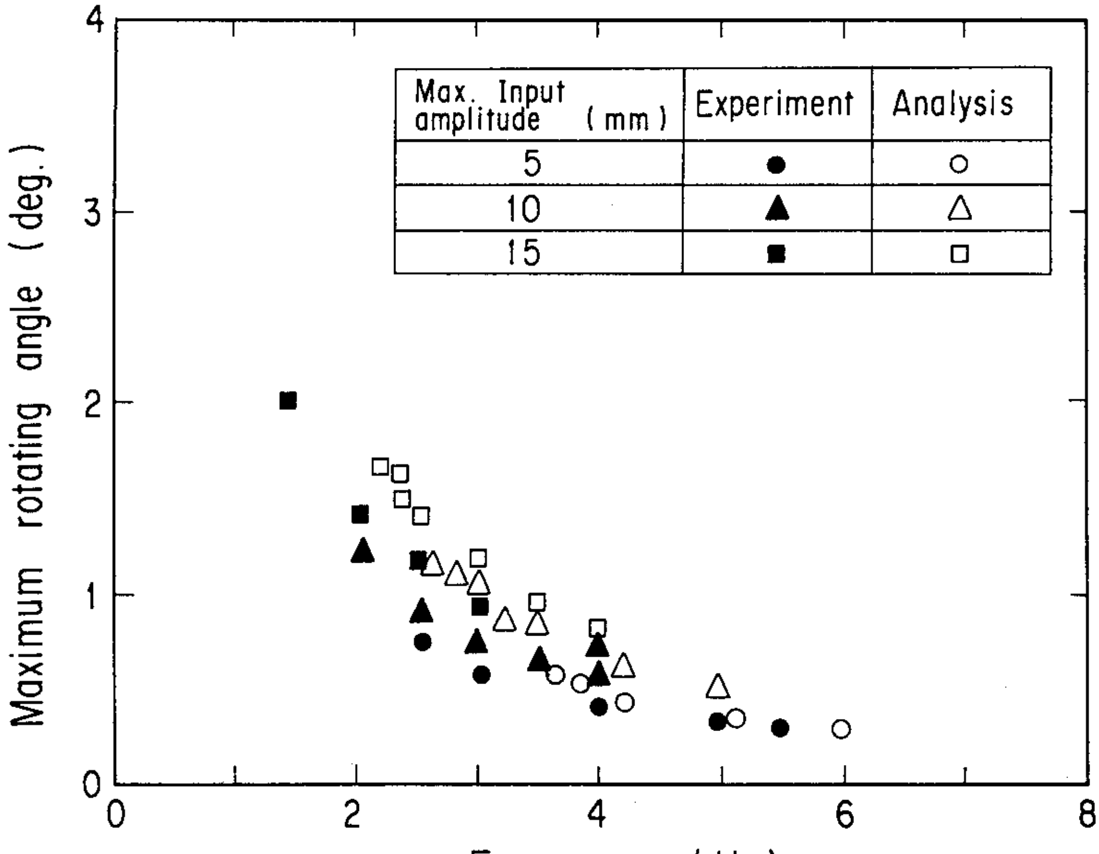


Fig. 3.2 Comparison between experiment and analysis

4. Computer program

4.1 Program description

The computer program ROCKING consists of a main routine and 26 subroutines. Overall structure of ROCKING is shown in Fig. 4.1. Functions of the main routine and subroutines are as follows:

MAIN : initializes the start of run.

JULY31 : numerical integration using the Runge-Kutta method.

FUN : formation of calculation equations.

MOMENT : calculation of moments.

WIRDSP : calculation of wire rope lengths.

WIRINI : calculation of initial wire rope lengths.

GREATT : search of maximum values.

BIGLK : printout of maximum values.

DTVWST : store data for plotting.

DTPLOT : plotting of cask displacements.

XYPLOT : plotting of response values.

UWPLOT : plotting of input wave.

BLKPLT : plotting of response values.

FRIC : calculation of friction force,

FUNU0 : set of horizontal seismic wave,

FUNW0 : set of vertical seismic wave,

FIT : calculation of amplitude of seismic wave,

SSIN : calculation of sine function,

GCOS : calculation of cosine function,

ZEROCL : clear memory,

MEMMAX : search of maximum values,

MEMMIN : search of minimum values,

UMMAX : search of maximum values,

UMMIN : search of minimum values,

DLOG : calculation of log-scale data,

RNUM : plot number,

DTLIST : printout of input data echo.

A macroscopic flowsheet of ROCKING is shown in Fig. 4.2.

4.2 Descrption of input data

This section describes the input data required by ROCKING. The input data consists of the job description, analysis type such as with or without wire ropes for tip-over turn, geometries, the cask weight, initial conditions, boundary condition constants, the integral time step, strength data of wire ropes, seismic data and options for plottings. The input instructions are simple and easy to follow. The input data forms are presented in Table 4.1.

4.3 Descrption of output data

This section describes the output data required by ROCKING. The contents of these various output quantities are described in the followings.

(1) Input data

Input data are printed in two formats. The first printout format is exactly the same as they are read. Second, the code listes the data as interpreted by the code.

(2) Calculation data

The calculation values; the maximum displacement, velocity, acceleration, forces and moments are printed.

(3) Graphical output

The computer code ROCKING provides users with graphical output of cask deflection and time history curves such as displacement, velocity, acceleration, forces and moments.

Table 4.1 Input data for ROCKING

Columns	Format	Variables	Descriptions
Data set No.1:Job description.			
1 - 5	5A1	—	'TITLE'.
6 - 10	5X	—	Blank.
11 - 72	62A1	ITI	Job description.
Data set No.2:Option data.			
1 - 7	7A1	—	'CONTROL'.
8 - 10	3X	—	Blank.
11 - 15	I5	IWIR	Option data for wire rope. =0:Without wire rope. =1:With wire rope.
Data set No.3:Geometry data.			
1 - 8	8A1	—	'GEOMETRY'.
9 - 10	2X	—	Blank.
11 - 20	F10.0	HH	Height of gravity center.
21 - 30	F10.0	BB	Upper radius.
31 - 40	F10.0	RCSK(2)	Lower radius.
41 - 50	F10.0	EI	Moment of inertia.
51 - 60	F10.0	EM	Mass.
61 - 70	F10.0	WI	Weight.
Data set No.4:Data for vertical impact.			
1 - 10	10A1	—	'V-SPRING-L'.
11 - 20	F10.0	XB	Radius of vertical impact point.
Data set No.5:Spring constant and damping coefficient of vertical impact.			
1 - 8	8A1	—	'VERTICAL'.
9 - 10	2X	—	Blank.
11 - 20	F10.0	V1K	Spring constant(1).
21 - 30	F10.0	V2K	Spring constant(2).

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
31 - 40	F10.0	V3K	Spring constant(3).
41 - 50	F10.0	V1C	Damping coefficient(1).
51 - 60	F10.0	V2C	Damping coefficient(2).
61 - 70	F10.0	V3C	Damping coefficient(3).
Resultant spring constant K_z; $K_z = V1K + V2K \cdot \gamma + V3K \cdot \gamma^2$. Resultant damping coefficient C_z; $C_z = V1C + V2C \cdot \dot{\gamma} + V3C \cdot \dot{\gamma}^2$, where, γ : Overlap vector of vertical spring, $\dot{\gamma}$: Relative velocity between cask and floor.			
Data set No.6:Spring constant and damping coefficient of lower boundary horizontal impact.			
1 - 8	8A1	—	'HORIZONTAL'.
9 - 10	2X	—	Blank.
11 - 20	F10.0	B1K	Spring constant(1).
21 - 30	F10.0	B2K	Spring constant(2).
31 - 40	F10.0	B3K	Spring constant(3).
41 - 50	F10.0	B1C	Damping coefficient(1).
51 - 60	F10.0	B2C	Damping coefficient(2).
61 - 70	F10.0	B3C	Damping coefficient(3).
Resultant spring constant K_ℓ; $K_\ell = B1K + B2K \cdot \tau + B3K \cdot \tau^2$. Resultant damping coefficient C_ℓ; $C_\ell = B1C + B2C \cdot \dot{\tau} + B3C \cdot \dot{\tau}^2$, where, τ : Overlap vector of horizontal spring, $\dot{\tau}$: Relative velocity between cask and floor.			
Data set No.7;Friction coefficient.			
1 - 8	8A1	—	'FRICTION'.
9 - 10	2X	—	Blank.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
11 - 20	F10.0	F1	Coefficient of static friction.
21 - 30	F10.0	F2	Coefficient of kinetic friction.
Data set No.8;Initial values.			
1 - 7	7A1	–	'INITIAL'.
8 - 10	3X	–	Blank.
11 - 20	F10.0	X(1)	Initial angle.
21 - 30	F10.0	X(2)	Initial angular velocity.
31 - 40	F10.0	X(3)	Initial horizontal displacement.
41 - 50	F10.0	X(4)	Initial horizontal velocity.
51 - 60	F10.0	X(5)	Initial vertical displacement.
61 - 70	F10.0	X(6)	Initial vertical velocity.
Data set No.9;Timestep data.			
1 - 8	8A1	–	'TIMESTEP'.
9 - 10	2X	–	Blank.
11 - 20	F10.0	TINT	Initial problem time.
21 - 30	F10.0	TEND	Final problem time.
31 - 40	F10.0	DT	Integral time step size.
41 - 50	F10.0	TSTS	Time step size of plot data output for time history curve.
51 - 60	F10.0	TSTP	Time step size of plot data output for cask rocking.
Data set No.10A;Wire rope A data.			
1 - 10	10A1	–	'WIREROPE-A'.
11 - 20	F10.0	RWI(1)	Radius of attached wire rope of lower end.
21 - 30	F10.0	ZWI(1)	Height of attached wire rope of lower end.
31 - 40	F10.0	ZWCI(1)	Height of attached wire rope of upper end.
41 - 50	F10.0	WGEO(1,1)	Young's modulus of wire rope.
51 - 60	F10.0	WGEO(2,1)	Sectional area of wire rope.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
61 - 70	F10.0	WGEO(3,1)	Ultimate strength of wire rope(default value is 10^{20}).
Data set No.10B;Wire rope B data.			
1 - 10	10A1	–	'WIREROPE-B'.
11 - 20	F10.0	RWI(2)	Radius of attached wire rope of lower end.
21 - 30	F10.0	ZWI(2)	Height of attached wire rope of lower end.
31 - 40	F10.0	ZWCI(2)	Height of attached wire rope of upper end.
41 - 50	F10.0	WGEO(1,2)	Young's modulus of wire rope.
51 - 60	F10.0	WGEO(2,2)	Sectional area of wire rope.
61 - 70	F10.0	WGEO(3,2)	Ultimate strength of wire rope(default value is 10^{20}).
Data set No.10C;Wire rope C data.			
1 - 10	10A1	–	'WIREROPE-C'.
11 - 20	F10.0	RWI(3)	Radius of attached wire rope of lower end.
21 - 30	F10.0	ZWI(3)	Height of attached wire rope of lower end.
31 - 40	F10.0	ZWCI(3)	Height of attached wire rope of upper end.
41 - 50	F10.0	WGEO(1,3)	Young's modulus of wire rope.
51 - 60	F10.0	WGEO(2,3)	Sectional area of wire rope.
61 - 70	F10.0	WGEO(3,3)	Ultimate strength of wire rope(default value is 10^{20}).
Data set No.10D;Wire rope D data.			
1 - 10	10A1	–	'WIREROPE-D'.
11 - 20	F10.0	RWI(4)	Radius of attached wire rope of lower end.
21 - 30	F10.0	ZWI(4)	Height of attached wire rope of lower end.
31 - 40	F10.0	ZWCI(4)	Height of attached wire rope of upper end.
41 - 50	F10.0	WGEO(1,4)	Young's modulus of wire rope.
51 - 60	F10.0	WGEO(2,4)	Sectional area of wire rope.
61 - 70	F10.0	WGEO(3,4)	Ultimate strength of wire rope(default value

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
			is 10^{20}).
Data set No.10E;Angle of seismic load.			
1 - 10	10A1	−	'INPUTANGLE'.
11 - 20	F10.0	BETA	Angle of seismic load.
Skip data set 10A −10E if there are not wire ropes.			
Data set No.11A;Seismic data(I).			
1 - 10	10A1	−	'SINUSOIDAL'.
11 - 20	F10.0	G	Amplitude of input wave. KIK=0:Displacement amplitude. KIK=2:Acceleration amplitude.
21 - 30	F10.0	P	Circular frequency of input wave. KIK=0:Circular frequency(rad/s). KIK=2:Freauency(Hz).
Data set No.11B;Seismic data(II).			
1 - 9	9A1	−	'SEISMIC-1'.
10	1X	−	Blank.
11 - 15	I5	KIK	Option for seismic wave. KIK=0:Sinusoidal wave. KIK=2:Sinusoidal wave. KIK=3:Seismic wave.
Data set No.11C;Seismic data(III).			
1 - 9	9A1	−	'SEISMIC-2'.
10	1X	−	Blank.
11 - 15	I5	KXK(1)	Option for horizontal seismic wave. KXK(1)=0:No seismic wave. KIK(1)=1:Seismic wave.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
16 - 20	5X	—	Blank.
21 - 25	I5	KXK(3)	Option for vertical seismic wave. KXK(3)=0:No seismic wave. KXK(3)=1:Seismic wave.
26 - 30	5X	—	Blank.
31 - 40	F10.0	ALP	Scale factor of displacement of horizontal seismic wave.
41 - 50	F10.0	BET	Scale factor of velocity of horizontal seismic wave.
Data set No.11D1:Seismic data(IVA:Option for displacement of horizontal seismic wave).			
1 - 10	10X	—	Blank.
11 - 14	I4	KXN(1,1)	Number of data.
15 - 30	16X	—	Blank.
31 - 40	F10.0	DDT(1,1)	Time step size of data.
Data set No.11D2:Seismic data(IVB:Displacement of horizontal seismic wave).			
1 - 10	10X	—	Blank.
11 - 20	F10.0	VX(1,1,1)	Horizontal displacement.
21 - 30	F10.0	VX(1,1,2)	....
31 - 40	F10.0	VX(1,1,3)	....
41 - 50	F10.0	VX(1,1,4)	....
51 - 60	F10.0	VX(1,1,5)	Horizontal displacement.
Repeat data set No.11D2.			
Data set No.11E1:Seismic data(VA:Option for velocity of horizontal seismic wave).			
1 - 10	10X	—	Blank.
11 - 14	I4	KXN(1,2)	Number of data.
15 - 30	16X	—	Blank.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
31 - 40	F10.0	DDT(1,2)	Time step size of data.
Data set No.11E2:Seismic data(VB:Velocity of horizontal seismic wave).			
1 - 10	10X	—	Blank.
11 - 20	F10.0	VX(1,2,1)	Horizontal velocity.
21 - 30	F10.0	VX(1,2,2)	....
31 - 40	F10.0	VX(1,2,3)	....
41 - 50	F10.0	VX(1,2,4)	....
51 - 60	F10.0	VX(1,2,5)	Horizontal velocity.
Repeat data set No.11E2.			
Data set No.11F1:Seismic data(VIA:Option for displacement of vertical seismic wave).			
1 - 10	10X	—	Blank.
11 - 14	I4	KXN(3,1)	Number of data.
15 - 30	16X	—	Blank.
31 - 40	F10.0	DDT(3,1)	Time step size of data.
Data set No.11F2:Seismic data(VIB:Displacement of vertical seismic wave).			
1 - 10	10X	—	Blank.
11 - 20	F10.0	VX(3,1,1)	Vertical displacement.
21 - 30	F10.0	VX(3,1,2)	....
31 - 40	F10.0	VX(3,1,3)	....
41 - 50	F10.0	VX(3,1,4)	....
51 - 60	F10.0	VX(3,1,5)	Vertical displacement.
Repeat data set No.11F2.			
Data set No.11G1:Seismic data(VIIA:Option for velocity of vertical seismic wave).			
1 - 10	10X	—	Blank.
11 - 14	I4	KXN(3,2)	Number of data.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
15 - 30	16X	—	Blank.
31 - 40	F10.0	DDT(3,2)	Time step size of data.
Data set No.11G2:Seismic data(VIIB:Velocity of vertical seismic wave).			
1 - 10	10X	—	Blank.
11 - 20	F10.0	VX(3,2,1)	Vertical velocity.
21 - 30	F10.0	VX(3,2,2)	....
31 - 40	F10.0	VX(3,2,3)	....
41 - 50	F10.0	VX(3,2,4)	....
51 - 60	F10.0	VX(3,2,5)	Vertical velocity.
Repeat data set No.11G2.			
Data set No.12:Plot data			
1 - 4	4A1	—	'PLOT'.
5 - 10	6X	—	Blank.
(11-70)	20I3	IPLOT(I)	Option for time history plot.
		(I=1,20)	IPLOT(I)=0:No plot.
			IPLOT(I)=1:Plotting.
11 - 13	I3	IPLOT(1)	Angle(degree).
14 - 16	I3	IPLOT(2)	Anglular velocity(rad/s).
17 - 19	I3	IPLOT(3)	Anglular acceleration(rad/s^2)
20 - 22	I3	IPLOT(4)	Horizontal displacement.
23 - 25	I3	IPLOT(5)	Horizontal velocity.
26 - 28	I3	IPLOT(6)	Horizontal acceleration.
29 - 31	I3	IPLOT(7)	Vertical displacement.
32 - 34	I3	IPLOT(8)	Vertical velocity.
35 - 37	I3	IPLOT(9)	Vertical acceleration.
38 - 40	I3	IPLOT(10)	Moment due to gravitational acceleration.
41 - 43	I3	IPLOT(11)	Frictional force.
44 - 46	I3	IPLOT(12)	Moment due to frictional force.
47 - 49	I3	IPLOT(13)	Vertical impact force.

Table 4.1 (Continued)

Columns	Format	Variables	Descriptions
50 - 52	I3	IPLOT(14)	Moment due to vertical impact force.
53 - 55	I3	IPLOT(15)	Horizontal impact force.
56 - 58	I3	IPLOT(16)	Moment due to horizontal impact force.
59 - 61	I3	IPLOT(17)	Tensile force of wire rope A.
62 - 64	I3	IPLOT(18)	Tensile force of wire rope B.
65 - 67	I3	IPLOT(19)	Tensile force of wire rope C.
68 - 70	I3	IPLOT(20)	Tensile force of wire rope D.

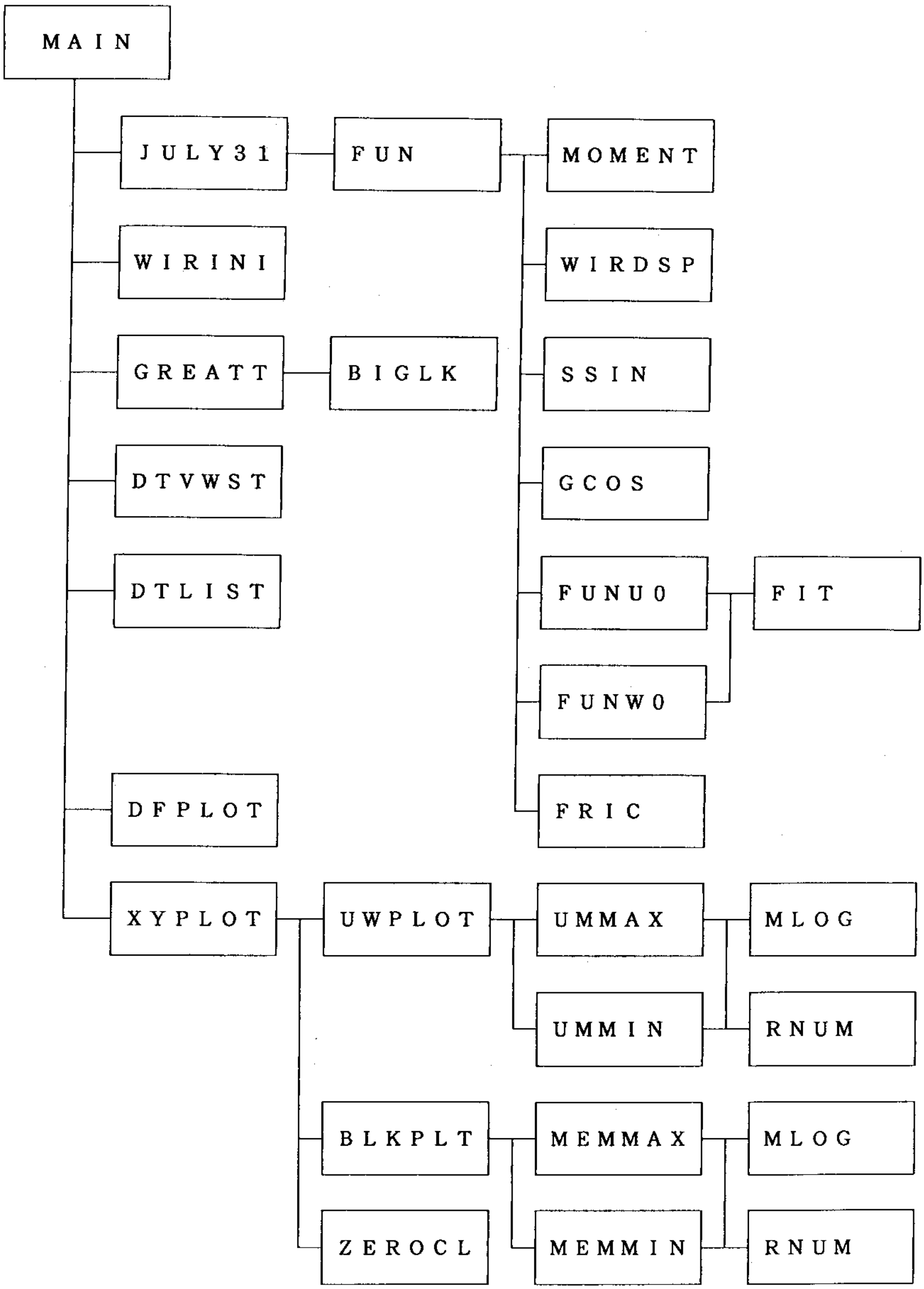


Fig. 4.1 Structure of ROCKING

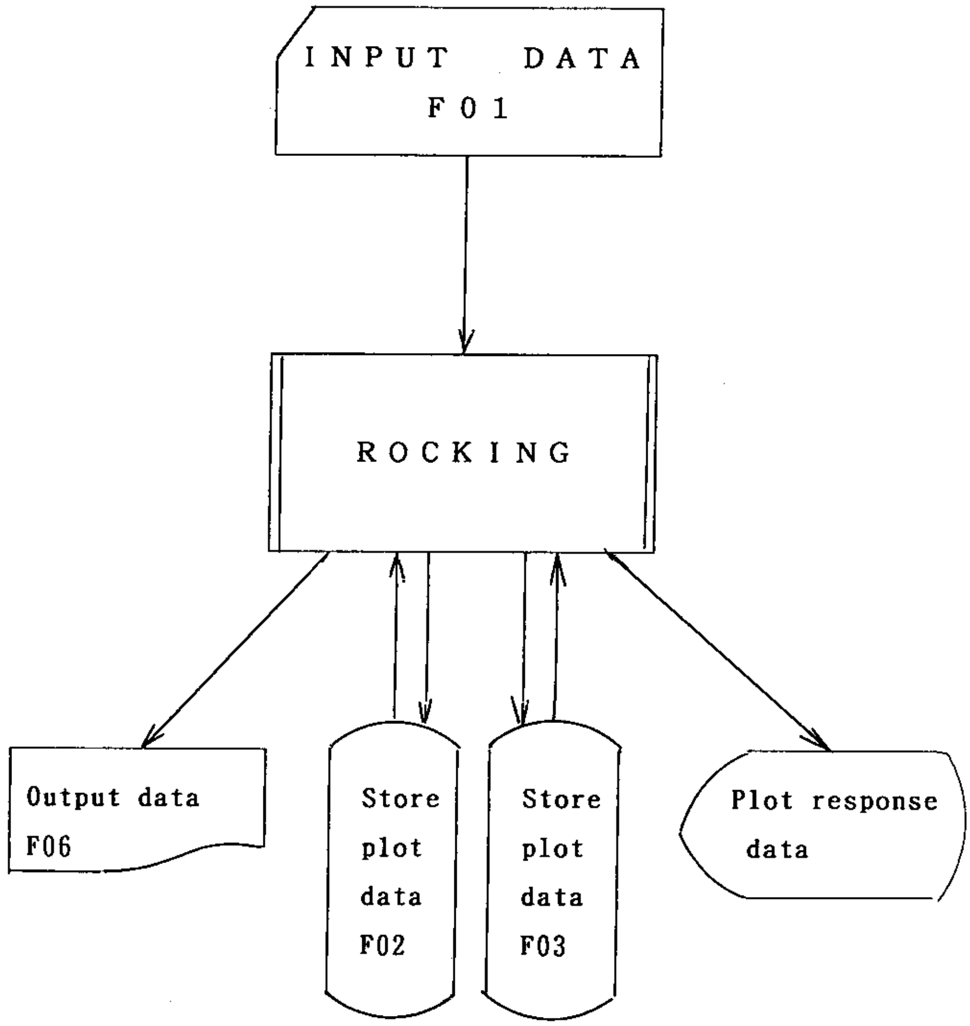


Fig. 4.2 Flowsheet of ROCKING

5. Conclusions

A computer program ROCKING for analysis of casks tip-over behavior has been developed. Validity tests for ROCKING have been performed using CRIEPI experimental results. Good agreement was obtained between analytical and experimental results.

In regard to evaluation of tip-over behavior of casks during sinusoidal and seismic excitations, the computer program ROCKING makes analysis economically shortening CPU time.

Acknowledgments

The author is indebted to Dr. C. Ito and his colleagues of Central Research Institute of Electric Power Industries for providing the experimental data. The author also indebted to Mr. J. Oshika of Century Research Center Ltd., for assistance of making the computer program.

References

(1) T. Saegusa, C. Ito, H. Imaeda, et al.,"Cask Storage of Spent Fuel at Reactors", CRIEPI U27(1993).

(2) K. Sirai, C. Ito, H. Ryu,"Development of Seismic Response Analysis Code of Cask",PATRAM 91(1991-Japan).

(3) T. Ikushima,"CASKETSS-2:A Computer Code System for Thermal and Structural Analysis of Nuclear Fuel Shipping Casks(Version 2)", JAERI-M 91-129(1991).

5. Conclusions

A computer program ROCKING for analysis of casks tip-over behavior has been developed. Validity tests for ROCKING have been performed using CRIEPI experimental results. Good agreement was obtained between analytical and experimental results.

In regard to evaluation of tip-over behavior of casks during sinusoidal and seismic excitations, the computer program ROCKING makes analysis economically shortening CPU time.

Acknowledgments

The author is indebted to Dr. C. Ito and his colleagues of Central Research Institute of Electric Power Industries for providing the experimental data. The author also indebted to Mr. J. Oshika of Century Research Center Ltd., for assistance of making the computer program.

References

(1) T. Saegusa, C. Ito, H. Imaeda, et al.,"Cask Storage of Spent Fuel at Reactors", CRIEPI U27(1993).

(2) K. Sirai, C. Ito, H. Ryu,"Development of Seismic Response Analysis Code of Cask",PATRAM 91(1991-Japan).

(3) T. Ikushima,"CASKETSS-2:A Computer Code System for Thermal and Structural Analysis of Nuclear Fuel Shipping Casks(Version 2)", JAERI-M 91-129(1991).

5. Conclusions

A computer program ROCKING for analysis of casks tip-over behavior has been developed. Validity tests for ROCKING have been performed using CRIEPI experimental results. Good agreement was obtained between analytical and experimental results.

In regard to evaluation of tip-over behavior of casks during sinusoidal and seismic excitations, the computer program ROCKING makes analysis economically shortening CPU time.

Acknowledgments

The author is indebted to Dr. C. Ito and his colleagues of Central Research Institute of Electric Power Industries for providing the experimental data. The author also indebted to Mr. J. Oshika of Century Research Center Ltd., for assistance of making the computer program.

References

(1) T. Saegusa, C. Ito, H. Imaeda, et al.,"Cask Storage of Spent Fuel at Reactors", CRIEPI U27(1993).

(2) K. Sirai, C. Ito, H. Ryu,"Development of Seismic Response Analysis Code of Cask",PATRAM 91(1991-Japan).

(3) T. Ikushima,"CASKETSS-2:A Computer Code System for Thermal and Structural Analysis of Nuclear Fuel Shipping Casks(Version 2)", JAERI-M 91-129(1991).

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: SAMPLE PROBLEM SINE 3.0 HZ 1.0CM WITH BASE FRICTION WITH WIRE
CONTROL      1
GEOMETRY       91.25      32.90     29.00  3.000E+5      12.86  1.260E+4
V-SPRING-L     25.00     -25.00
VERTICAL    5.600E+5        0.0       0.0  4.500E-4        0.0       0.0
HORIZONTAL  5.600E+5        0.0       0.0  4.500E-4        0.0       0.0
FRICTION         0.3        0.0
INITIAL          0.0        0.0       0.0       0.0        0.0       0.0
TIMESTEP         0.0        2.0    0.0001     0.002        0.2
WIREROPE-A     250.0        0.0     384.5  2.100E+6       2.19  3.000E+4
WIREROPE-B     250.0        0.0     384.5  2.100E+6       2.19  3.000E+4
WIREROPE-C     250.0        0.0     384.5  2.100E+6       2.19  3.000E+4
WIREROPE-D     250.0        0.0     384.5  2.100E+6       2.19  3.000E+4
INPUTANGLE       0.0
SINUSOIDAL       1.0        3.0       0.0
SEISMIC-1    3
SEISMIC-2    0    0    0
PLOT           1  1  1  1  1  1  0  0  0  0  1  0  1  0  1  0  1  1  1  1  0

----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

ROCKING TITLE: SAMPLE PROBLEM SINE 3.0 HZ 1.0CM WITH BASE FRICTION WITH WIRE

1. CALCULATION SYSTEM

NO. OF MASS	1
START TIME	0.000000
END TIME	2.000000
TIME STEP	0.000100
TIME STEP FOR TIME HISTORY DATA STORE	0.002000
TIME STEP FOR DISPLACEMENT DATA STORE	0.200000

2. DIMENSION

NO.	BLOCK HALH HEIGHT	BLOCK HALH WIDTH	VERTICAL SPRING DISTANCE LEFT	VERTICAL SPRING DISTANCE RIGHT
1	91.2500	32.9000	-25.0000	25.0000

3. MASS

NO. OF MASS	MASS(KG.SEC**2/CM)	WEIGHT(KG)	ROTATION OF INERTIA
1	0.128600E+02	0.126000E+05	0.300000E+06

4. STIFFNESS

4.1 VERTICAL

NO.	KV(1)	KV(2)	KV(3)
1	0.560000E+06	0.000000E+00	0.000000E+00

4.2 HORIZONTAL

NO.	KB(1)	KB(2)	KB(3)
1	0.560000E+06	0.000000E+00	0.000000E+00

5. DAMPING

5.1 VERTICAL

NO.	CV(1)	CV(2)	CV(3)
1	0.450000E-03	0.000000E+00	0.000000E+00

5.2 HORIZONTAL

NO.	CB(1)	CB(2)	CB(3)
1	0.450000E-03	0.000000E+00	0.000000E+00

6. FRICTION

INTERFACE	F(1)	F(2)
1	0.378000E+04	0.000000E+00

7. INITIAL STATE

NO.	THETA	D-THETA/D-T	U	D-U/D-T	W	D-W/D-T
1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00

Appendix B (Continued)

8. PLOTTING DATA

THETA - (DEGREE)

THETA - VELOCITY

THETA - ACCELERATION

U - DISPLACEMENT

U - VELOCITY

U - ACCELERATION

FORCE BY FRICTION

FORCE BY V.SPRING

FORCE BY B.WALL

FORCE BY WIRE-A

FORCE BY WIRE-B

FORCE BY WIRE-C

FORCE BY WIRE-D

9. CALCULATION RESULTS (MAXIMUM VALUES)

9.1 DISPLACEMENT

THETA(DEGREE)	U	W
6.60696E+00	1.03609E+01	3.85642E+00

9.2 VELOCITY

D-THETA/D-T	D-U/D-T	D-W/D-T
8.99792E-01	7.88946E+01	9.65527E+01

9.3 ACCELERATION

D2THETA/DT2	D2U/DT2	D2W/DT2
2.36095E+01	4.84425E+03	1.37748E+04

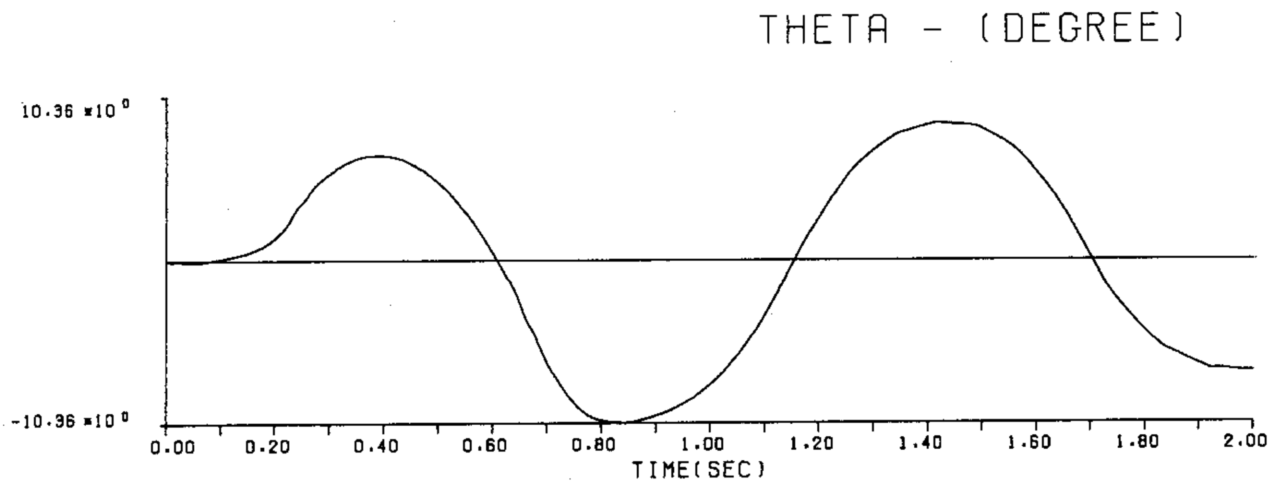
9.4 FORCE AND MOMENT

MOMENT BY GRAVITY	FORCE BY FRICTION	MOMENT BY FRICTION	FORCE BY V.SPRING	MOMENT BY V.SPRING	FORCE BY H.SPRING	MOMENT BY H.SPRING
4.14540E+05	3.78000E+03	3.56938E+05	2.02397E+05	3.99026E+06	5.79090E+04	5.35245E+06

9.5 WIRE FORCE

FORCE BY WIRE-A	FORCE BY WIRE-B	FORCE BY WIRE-C	FORCE BY WIRE-D
6.56596E+04	3.97098E+04	6.56506E+04	6.56944E+04

Appendix C Graphical output



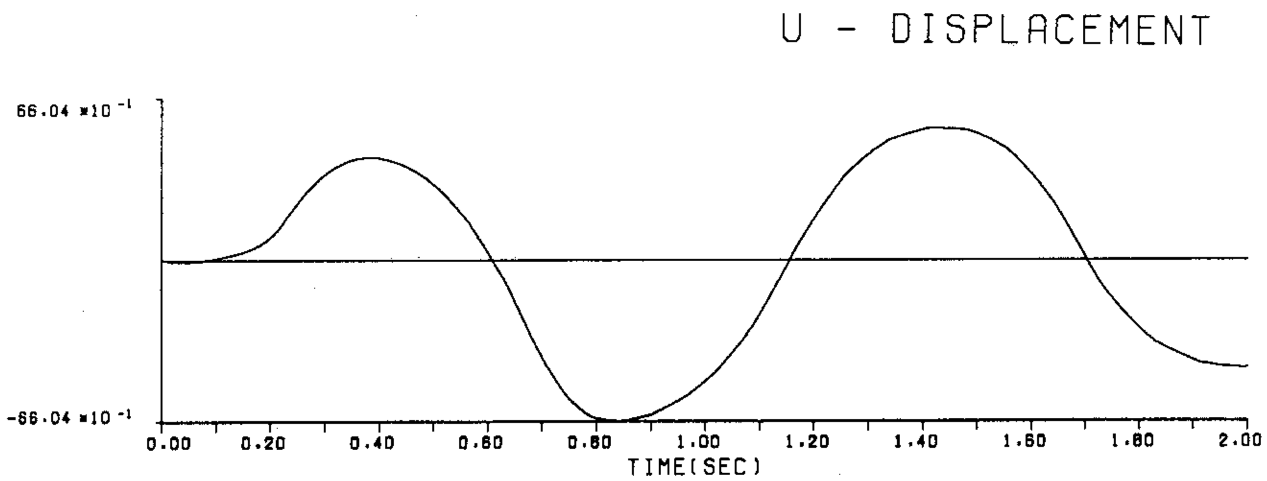


Fig. C.1 Graphical output of ROCKING(1)

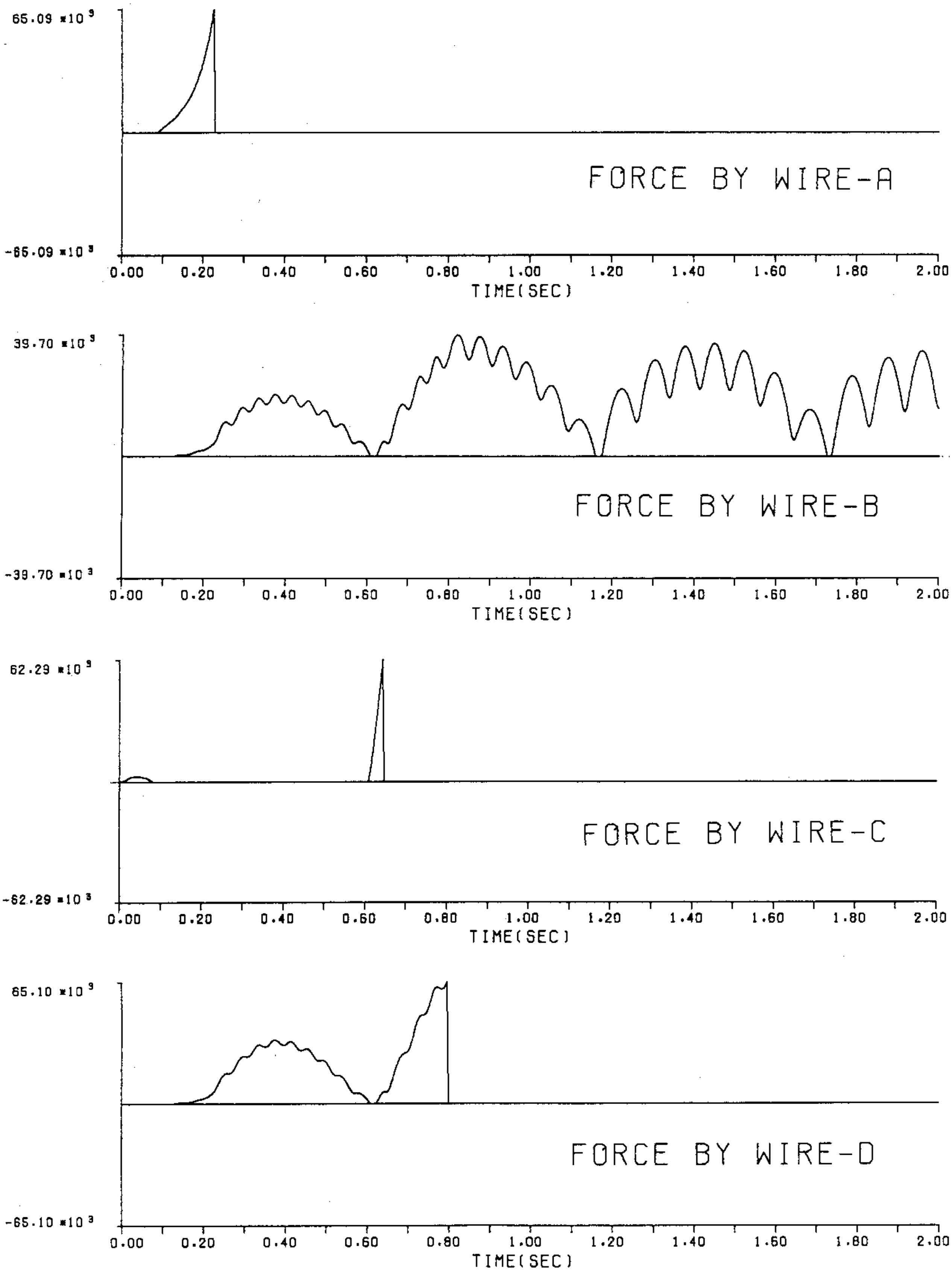


Fig. C.2 Graphical output of ROCKING(2)

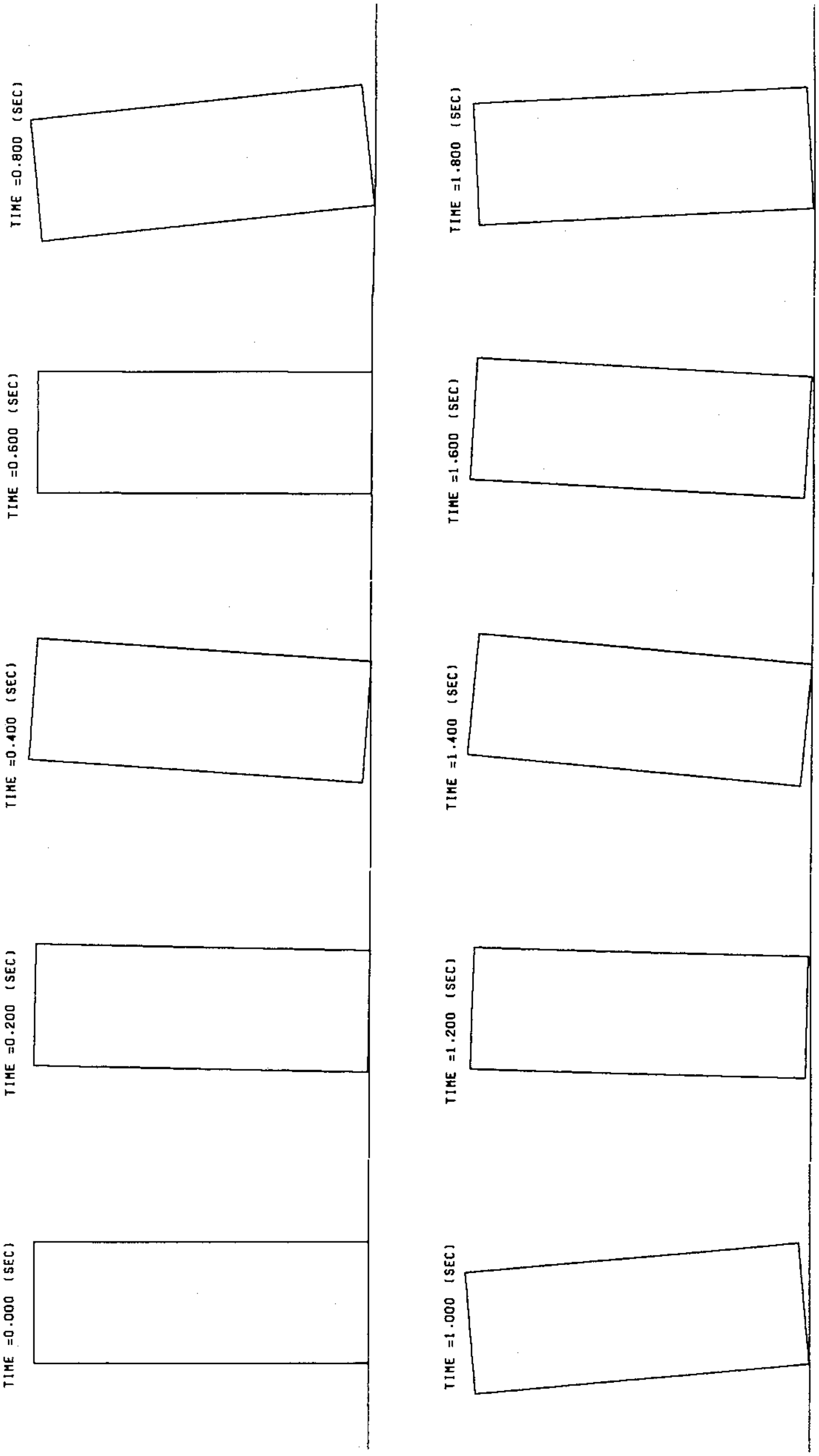


Fig. C.3 Graphical output of ROCKING(3)

Appendix D Job control data

The job control data for ROCKING 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,ROCKING
   T.03 C.02 W.01 I.02 CLS GRP
   OPTP MSGCLASS=A,MSGLEVEL=(2,0,1),CLASS=B,NOTIFY=JXXXX
   OPTP PASSWORD=XXXXXXXX
//RUN EXEC LMGOEX,LM=J2322.LMROCKIN,PNM=ROCKING
// EXPAND GRNLP
//FT01F001 DD DSN=JXXXX.DTROCKIN.DATA,DISP=SHR
//FT02F001 DD SPACE=(TRK,(5,5)),UNIT=TSSWK
//FT03F001 DD SPACE=(TRK,(5,5)),UNIT=TSSWK
++
//
```

Appendix E Program abstract

1. Name :

   ROCKING

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

   FACOM M-780, SUN4, IBM-PC

3. Nature of physical problem solved:

   Drop impact analysis of radioactive material transport casks.

4. Method of solutions:

   One-dimensional dynamic analysis.

5. Restrictions on the complexity of the problem:

   None.

6. Typical running time:

   FACOM M-780 : 1 minute.
   SUN4 : 2 minutes.
   IBM PC : 5 minutes.

7. Unusual features of the program:

   None.

8. Related and auxiliary program:

   None.

9. Status

   -

10. References:

   "CASKETSS-2:Computer Code System for Thermal and Structural Analysis of Nuclear Fuel Shipping Casks(Version2)", JAERI-M 91-129(1991).

11. Machine requirement:

    1100 k bytes of core memory.

12. Program language used:

    FORTRAN-77

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

    FACOM M-780 : MSP

    SUN4 : Solaris 2.1

    IBM PC : Windows 3.1

14. Any other program or operating information or restrictions:

    The program is approximately 1700 source steps. The plotting programs are used as follows:

    FACOM M-780 : CALCOMP plotter 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
C             ROCKING ---  MAIN PROGRAM                                C
C                                                                      C
C***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 HHHH, BBBB
      COMMON/IRIR/ XB(10),VK,VC,DK,DC,BK,BC,HH,BB,EI,EM,WI,NB,ITI(18)
      COMMON/COEF/ F1,  F2,
     1             V1K, V2K, V3K, B1K, B2K, B3K,
     2             V1C, V2C, V3C, B1C, B2C, B3C, UD
      COMMON/FIG/ TIM,STEP,TSTP,GHOST(1024)
      COMMON/AAWWPP/ G, W, P
      COMMON/XXD/ A(6)
      COMMON/FMFMFM/ FGM , FRI , FRM , VFZ , VYM , BWF , BWM ,
     1               FWF(4) , FWM(4)
      COMMON/GRETTT/ GFGMG, GFRIG, GFRMG, GVFZG, GVYMG, GBWFG, GBWMG,
     1               GMXX(6), GMAA(6), GFWF(4)
      COMMON/XDSDSX/ XDIS(6), UB
      COMMON/FRIFRI/ FPAI
      COMMON/CXXCXX/ CXX
      COMMON/KIKKIK/ Z1, Z2, Z3, Z4, Z5, Z6, KIK
      COMMON/KXKXK/ DDT(3,2),VX(3,2,1100), KXK(3), KXN(3,2)
      COMMON/PDPXPD/ PX,PD,PD2,PXX,PDD,PDW2
      COMMON/ALPBET/ ALP,BET
      COMMON/XX/ X(6),Q(6),T1(6),XN(6),N6,NMAX
      COMMON/BC/ EE,TC,NCONTC
      COMMON/UUIDID/ UUIDO,UUID1
      COMMON /WIRE/ RWI(4), ZWI(4), ZWCI(4), WGEO(3,4), WDISP(3,2,4),
     1              WCORDO(3,2,4), WCORD(3,2,4), WLENGO(4), WLENG(4),
     2              WFORC(4),WFCMP(3,4), WMCMP(3,4), IWSTAT(4),NWIRE
      COMMON /CASK/ RCSK(2), ZCSK, WCSK, CSKI, BETA
      COMMON /LUNITL/ LUCD, LUHT, LUDP, LUPR
C
      DIMENSION XWRIT(21),IPLOT(21),E(6),JPLOT(5,21)
      CHARACTER*26  PARM(3)
      DATA ((JPLOT(I,J),I=1,5),J=1,9)
     1         /'THET','A - ','(DEG','REE)','    '
     2          ,'THET','A - ','VELO','CITY','    '
     3          ,'THET','A - ','ACCE','LERA','TION'
     4          ,'    ','U - ','DISP','LACE','MENT'
     6          ,'    ','U - ','VELO','CITY','    '
     7          ,'    ','U - ','ACCE','LERA','TION'
     8          ,'    ','W - ','DISP','LACE','MENT'
     9          ,'    ','W - ','VELO','CITY','    '
     +          ,'    ','W - ','ACCE','LERA','TION'/
      DATA ((JPLOT(I,J),I=1,5),J=10,21)
     1          /'MOME','NT B','Y GR','AVIT','Y   '
     2          ,'FORC','E BY',' FRI','CTIO','N   '
     3          ,'MOME','NT B','Y FR','ICTI','ON  '
     4          ,'FORC','E BY',' V.S','PRIN','G   '
     5          ,'MOME','NT B','Y V.','SPRI','NG  '
     6          ,'FORC','E BY',' B.W','ALL ','    '
     7          ,'MOME','NT B','Y B.','WALL','    '
     8          ,'FORC','E BY',' WIR','E-A ','    '
     9          ,'FORC','E BY',' WIR','E-B ','    '
     +          ,'FORC','E BY',' WIR','E-C ','    '
     1          ,'FORC','E BY',' WIR','E-D ','    '
```

Appendix F (Continued)

```
     2            ,'      ','      ','      ','      ','      '/
C
CJR    CALL  OSPARM( PARM )
CJR    OPEN(UNIT=LUCD,FILE=PARM(1),FORM='FORMATTED',STATUS='OLD')
CJR    OPEN(UNIT=LUHT,FILE=PARM(2),FORM='UNFORMATTED',STATUS='UNKNOWN')
CJR    OPEN(UNIT=LUDP,FILE=PARM(3),FORM='UNFORMATTED',STATUS='UNKNOWN')
C
       PAIPAI = 3.14159265359D0
       MAXD = 21
       LUCD = 1
       LUHT = 2
       LUDP = 3
       LUPR = 6
       GFGMG = 0.
       GFRIG = 0.
       GFRMG = 0.
       GVFZG = 0.
       GVYMG = 0.
       GBWFG = 0.
       GBWMG = 0.
       DO 50 N=1,4
    50 GFWF(N) = 0.
       DO 60 N=1,6
       GMXX(N) = 0.
    60 GMAA(N) = 0.
       DO 70 I=1,6
       A(I) = 0.E0
       X(I) = 0.0
       Q(I) = 0.0
       T1(I)= 0.0
    70 XN(I)= 0.0
       DO 80 I=1,3
       DO 80 J=1,4
    80 WFCMP(I,J) = 0.0
       DO 90 N=1,21
       IPLOT(N) = 0
    90 XWRIT(N) = 0.0
       UD     = 0.
C
       CALL DTLIST( LUCD, LUPR)
C
C      INPUT DATA
C
       READ(LUCD,6000,END=300)  ITI
       READ(LUCD,6010)   NWIRE
       NN = 1
       N6 = 6
       NMAX = N6
       NB = 2
       NBND = 0
C
C      CALCULATION DATA
C
       READ(LUCD,6020)  HH,  BB, RCSK(2), EI, EM, WI
       READ(LUCD,6020)  XB(1)
       XB(2) = - XB(1)
       READ(LUCD,6020)  V1K ,V2K ,V3K ,V1C ,V2C ,V3C
       READ(LUCD,6020)  B1K ,B2K ,B3K ,B1C ,B2C ,B3C
```

Appendix F (Continued)

```
      READ(LUCD,6020)  F1  , F2 , CXX
      READ(LUCD,6020)  (X(I),I=1,N6)
      READ(LUCD,6020)  TINT, TEND, DT, TSTS, TSTP
C
C    PLOTTING DATA INTERVAL
C
      IF ( TSTS .LT. DT ) GO TO 100
      ITHTMX = ( TEND - TINT ) / TSTS
      IF ( ITHTMX .GT. 2048 ) ITHTMX = 2048
      TSTS = ( TEND - TINT ) / DFLOAT( ITHTMX )
  100 CONTINUE
      IF ( TSTP .LT. DT ) GO TO 105
      ITDPMX = ( TEND - TINT ) / TSTP
      IF ( ITDPMX .GT. 2048 ) ITDPMX = 2048
      TSTP = ( TEND - TINT ) / DFLOAT( ITDPMX )
  105 CONTINUE
C
C-----  READ WIRE PROPERTY AND GEOMETORY
C
      IF( NWIRE .EQ. 0 ) GO TO 115
      DO  110  I=1,4
      READ(LUCD,6020)  RWI(I), ZWI(I), ZWCI(I), (WGEO(J,I),J=1,3)
      IF(WGEO(3,I) .EQ. 0.0 )   WGEO(3,I) = 1.0D20
  110 CONTINUE
      RCSK(1) = BB
      WCSK    = WI
      CSKI    = EI
      ZCSK    = HH
      READ(LUCD,6020)  BETA
C
C------ WIRE GEOM INITIAL SET
C
      CALL  WIRINI( 4, WCORDO, WLENGO, RWI, ZWI, ZWCI, RCSK(2), BETA)
C
  115 CONTINUE
C
C    SEISMIC DATA
C
      READ(LUCD,6020)  G, W, P, DIFDF, CHECK
      READ(LUCD,6025)  KIK, Z1, Z2, Z3, Z4, Z5, Z6
      READ(LUCD,6230) (KXK(I),I=1,3),ALP,BET
      IF ( DIFDF .EQ. 0.0 )   DIFDF = 1.E-5
      IF ( CHECK .EQ. 0.0 )   CHECK = 0.1
      IF (ALP.EQ.0.0) ALP=1.0
      IF (BET.EQ.0.0) BET=1.0
      DO 120 I=1,3
      IF(KXK(I).EQ.0) GO TO 120
      DO 125 J=1,2
      READ(LUCD,6240) KXN(I,J),IWAVE,DDT(I,J)
      JXN = KXN(I,J)
      READ(LUCD,6250) (VX(I,J,MM),MM=1,JXN)
  125 CONTINUE
  120 CONTINUE
C
C    PLOTTING DATA
C
      IF ( TSTS .LT. DT ) GO TO 130
      READ(LUCD,6005)  IPLOT
```

Appendix F (Continued)

```
  130 CONTINUE
C
C     FRICTION FORCE
C
      F1  = WI * F1
C
C     PRINT INPUT DATA
C
      WRITE(LUPR,6030) ITI,NN,TINT,TEND,DT,TSTS,TSTP
      WRITE(LUPR,6040)
      WRITE(LUPR,6310) NN, HH , BB, XB(2), XB(1)
      WRITE(LUPR,6060)
      WRITE(LUPR,6070) NN, EM  ,WI    ,EI
      WRITE(LUPR,6080)
      WRITE(LUPR,6090) NN, V1K ,V2K   ,V3K
      WRITE(LUPR,6110)
      WRITE(LUPR,6090) NN, B1K ,B2K   ,B3K
      WRITE(LUPR,6120)
      WRITE(LUPR,6090) NN, V1C ,V2C   ,V3C
      WRITE(LUPR,6140)
      WRITE(LUPR,6090) NN, B1C ,B2C   ,B3C
      WRITE(LUPR,6150) NN, F1 ,F2
      WRITE(LUPR,6160)
      WRITE(LUPR,6170) NN,(X(J),J=1,6)
      WRITE(LUPR,6320)
C
      IF ( TSTS .LT. DT ) GO TO 145
      NDATA = 0
      DO 140 I=1,21
      IF ( IPLOT(I) .EQ. 0 )              GO TO  140
      WRITE(LUPR,6330) (JPLOT(J,I),J=1,5)
      NDATA = NDATA + 1
  140 CONTINUE
  145 CONTINUE
      TIM  = TINT
      TPLT = TSTS
      TPTP = TSTP
C
C     BLOCK DISPLACEMENT PLOT DATA
C
      HHHH = HH
      BBBB = BB
      IF ( TSTP .GE. DT )  WRITE( LUDP ) HHHH, BBBB
C
      E(1) = FUNU0(TIM)
      E(2) = FUNU1(TIM)
      E(1) = PXX
      E(2) = PDD
      E2D  = FUNU1(TIM+0.01)
      E2D  = PDD
      E(3) = 100.*(E2D-E(2))
      E(4) = FUNW0(TIM)
      E(5) = FUNW1(TIM)
      E(4) = UUID0
      E(5) = UUID1
      E5D  = FUNW1(TIM+0.01)
      E5D  = UUID1
      E(6) = 100.*(E5D-E(5))
```

Appendix F (Continued)

```
      UB = PXX
C
C     TIME HISTORY CURVE PLOT DATA
C
      IF( TSTS .LT. DT ) GO TO 150
      WRITE(LUHT)  NN,TINT,TEND
      WRITE(LUHT)  IPLOT
      WRITE(LUHT)  TIM,E,(XWRIT(I),I=1,NDATA)
  150 CONTINUE
C
C
C     DISPLACEMENT PLOT DATA
C
      IF( TSTP .LT. DT ) GO TO 160
      CALL  DTVWST ( TIM, UB, XN, HH, BB )
  160 CONTINUE
C
  170 NZR = 0
      E(1) = FUNU0(TIM)
      E(2) = FUNU1(TIM)
      E(1) = PXX
      E(2) = PDD
      E2D  = FUNU1(TIM+0.01)
      E2D  = PDD
      E(3) = 100.*(E2D-E(2))
      E(4) = FUNW0(TIM)
      E(5) = FUNW1(TIM)
      E(4) = UUID0
      E(5) = UUID1
      E5D  = FUNW1(TIM+0.01)
      E5D  = UUID1
      E(6) = 100.*(E5D-E(5))
      UB = PXX
C
      DT1 = DT
  180 CALL JULY31(TIM,DT1,X,XN,N6,Q,T1)
      XDIS(3) = XN(3) - UB
      XDIS(1) = XN(1)
      XDIS(2) = XN(2)
      XDIS(4) = XN(4)
      XDIS(5) = XN(5)
      XDIS(6) = XN(6)
C
      N1 = 0
      IF(NZR.EQ.0.AND.X(1)*XN(1).LT.0.) THEN
              DT2 = DABS(X(1))/(DABS(X(1))+DABS(XN(1)))
              IF(DT2.LT.CHECK)                  GO TO  190
              DTXXDT=DT
              DT2 = DTXXDT *DT2
              IF(DT2.GE.DT1) GO TO 190
              DT1 = DT2
              N1  = 1
      ENDIF
  190 CONTINUE
C
      IF(NZR .EQ. 1)               GO TO  200
      IF(N1  .EQ. 0)               GO TO  200
      N2=N1
```

Appendix F (Continued)

```
      NZR = 1
      GO TO 180
C
  200 CONTINUE
      TIM = TIM + DT1
      IF(NZR.EQ.0)                        GO TO  205
      IF(N2.EQ.1)    XN(6*N2-5) = 0.
C
  205 NDATA = 0
C
C     MAXIMUM VALUE SEARCH
C
      CALL  GREATT ( XDIS, A, WFORC )
C
      IF( TSTS .LT. DT ) GO TO 210
      IF( TIM .LT. TPLT ) GO TO 210
      TPLT = TPLT + TSTS
      DO 550 I=1,21
      IF(IPLOT(I).EQ.0)                   GO TO  550
      NDATA = NDATA + 1
      IF ( NDATA .GT. MAXD )              GO TO  560
      IF ( I .LE. 9 )                     GO TO  500
      IMS9 = I-9
      GO TO (601,602,603,604,605,606,607,608,609,610,611),IMS9
  601 CONTINUE
      XWRIT(NDATA) = FGM
                                          GO TO  550
  602 CONTINUE
      XWRIT(NDATA) = FRI
                                          GO TO  550
  603 CONTINUE
      XWRIT(NDATA) = FRM
                                          GO TO  550
  604 CONTINUE
      XWRIT(NDATA) = VFZ
                                          GO TO  550
  605 CONTINUE
      XWRIT(NDATA) = VYM
                                          GO TO  550
  606 CONTINUE
      XWRIT(NDATA) = BWF
                                          GO TO  550
  607 CONTINUE
      XWRIT(NDATA) = BWM
                                          GO TO  550
  608 CONTINUE
      XWRIT(NDATA) = -WFORC(1)
                                          GO TO  550
  609 CONTINUE
      XWRIT(NDATA) = -WFORC(2)
                                          GO TO  550
  610 CONTINUE
      XWRIT(NDATA) = -WFORC(3)
                                          GO TO  550
  611 CONTINUE
      XWRIT(NDATA) = -WFORC(4)
                                          GO TO  550
C
```

Appendix F (Continued)

```
  500 CONTINUE
      THANGF = 1.0
      GO TO (501,502,521,540,503,522,504,505,523),I
  501 I1 =  1
      THANGF = 180.0 / PAIPAI
                                                       GO TO  510
  502 I1 =  2
                                                       GO TO  510
  503 I1 =  4
                                                       GO TO  510
  504 I1 =  5
                                                       GO TO  510
  505 I1 =  6
  510 CONTINUE
  520 XWRIT(NDATA) = XN(I1) * THANGF
                                                       GO TO  550
  521 I1 =  2
                                                       GO TO  530
  522 I1 =  4
                                                       GO TO  530
  523 I1 =  6
  530 CONTINUE
  535 XWRIT(NDATA) = A(I1)
                                                       GO TO  550
  540 CONTINUE
  545 XWRIT(NDATA) = XDIS(3)
  550 CONTINUE
                                                       GO TO  570
  560 NDATA = MAXD
  570 CONTINUE
C
C     TIME HISTORY CURVE PLOT DATA
C
      WRITE(LUHT)  TIM,E,(XWRIT(I),I=1,NDATA)
C
CJR   WRITE(LUPR,6400) TIM
 6400 FORMAT(1H , 'TIME HISTORY DATA PLOT TIME= ', F12.5)
C
  210 CONTINUE
C
C     DISPLACEMENT PLOT DATA
C
      IF( TSTP .LT. DT ) GO TO 250
      IF( TIM .LT. TPTP ) GO TO 250
CJR   WRITE(LUPR,6410) TIM
 6410 FORMAT(1H , 'BLOCK DEFORMATION PLOT TIME= ', F12.5)
      CALL  DTVWST ( TIM, UB, XN, HH, BB )
      TPTP = TPTP + TSTP
  250 CONTINUE
C
      DO 260 I=1,N6
  260 X(I) = XN(I)
      UD    = X(4)
C
      IF(TIM.GT.TEND)    GO TO  300
      GO TO 170
  300 CONTINUE
C
```

Appendix F (Continued)

```
C     TIME HISTORY CURVE PLOT DATA
C
      IF ( TSTS .LT. DT ) GO TO 310
      WRITE(LUHT)  TIM,E,(XWRIT(I),I=1,NDATA)
  310 CONTINUE
C
C     DISPLACEMENT PLOT DATA
C
      IF( TSTP .LT. DT ) GO TO 320
      CALL  DTVWST ( TIM, UB, XN, HH, BB )
  320 CONTINUE
C
C     PLOTTER ROUTINE
C
      CALL  BIGLK ( XDIS, A, WFORC )
      IF ( TSTS .LT. DT ) GO TO 330
      IF ( TSTP .LT. DT ) GO TO 330
      CALL  PLOTS(GHOST,1024)
      CALL  FACTOR(1.0)
C     CALL  FACTOR(0.8)
C     CALL  FACTOR(0.4)
      CALL  SYMBOL(0.0, 200.0, 4.0, ITI, 0.0, 72 )
      CALL  PLOT( 300.0, 0.0, -3 )
      CALL  PLOT( 0.0, 0.0, 666 )
      CALL  PLOT( 0.0, 0.0, 777 )
      CALL  PLOT( 0.0, 0.0, 888 )
      IF ( TSTS .GE. DT ) CALL  XYPLOT
      IF ( TSTP .GE. DT ) CALL  DFPLOT
      CALL  PLOT( 0.0, 0.0, 999 )
  330 CONTINUE
C
      WRITE(LUPR,6340)
      STOP
C
 6000 FORMAT(18A4)
 6005 FORMAT(10X,21I3)
 6010 FORMAT(10X,10I5)
 6020 FORMAT(10X,7E10.5)
 6025 FORMAT(10X,I5,5X,6F10.0 )
C
 6030 FORMAT(1H1,10X,'ROCKING   ',5X,18A4//5X,
     1 21H1. CALCULATION SYSTEM  //10X,
     2 37HNO. OF MASS                          ,3X,I12/   10X,
     3 37HSTART TIME                           ,3X,F12.6/ 10X,
     4 37HEND TIME                             ,3X,F12.6/ 10X,
     5 37HTIME STEP                            ,3X,F12.6/ 10X,
     6 37HTIME STEP FOR TIME HISTORY DATA STORE,3X,F12.6/ 10X,
     7 37HTIME STEP FOR DISPLACEMENT DATA STORE,3X,F12.6 )
 6040 FORMAT(1H0,///5X,12H2. DIMENSION//20X, 32HB L O C K       VERTICAL
     1 SPRING / 7X, 3HNO., 10X, 7HH A L H, 10X, 15HD I S T A N C E, /
     2 11X, 11HH E I G H T, 2X, 9HW I D T H, 3X, 7HL E F T, 2X,
     3 9HR I G H T )
 6060 FORMAT(////5X,7H3. MASS//10X,11HNO. OF MASS,5X,18HMASS(KG.SEC**2/C
     1M),13X,10HWEIGHT(KG),4X,19HROTATION OF INERTIA  )
 6070 FORMAT(I20,3E23.6)
 6080 FORMAT(1H0,///,4X,12H4. STIFFNESS//7X,12H4.1 VERTICAL/10X,3HNO.,
     1 11X,5HKV(1),11X,5HKV(2),11X,5HKV(3))
 6090 FORMAT(I13,3E16.6)
```

Appendix F (Continued)

```
6110 FORMAT(//7X,14H4.2 HORIZONTAL/10X,3HNO. ,11X,5HKB(1),
    1  11X,5HKB(2),11X,5HKB(3))
6120 FORMAT(1H0,///,4X,10H5. DAMPING//7X,12H5.1 VERTICAL/10X,3HNO.,
    1 11X,5HCV(1),11X,5HCV(2),11X,5HCV(3))
6140 FORMAT(//7X,14H5.2 HORIZONTAL/10X,3HNO.,11X,5HCB(1),
    1 11X,5HCB(2),11X,5HCB(3))
6150 FORMAT(1H1,4X,11H6. FRICTION//10X,9HINTERFACE,12X,4HF(1),12X,
    1 4HF(2)/(I15,4X,2E16.6))
6160 FORMAT(1H0,///,4X,17H7. INITIAL STATE   //10X,3HNO.,11X,5HTHETA,
    1 5X,11HD-THETA/D-T,15X,1HU,9X,7HD-U/D-T,15X,1HW,9X,7HD-W/D-T)
6170 FORMAT(I13,7E16.6)
6220 FORMAT( I5, 2X, 5( 3X, 1PE11.4 ) , OP )
6230 FORMAT(10X,3I5,5X,3F10.5)
6240 FORMAT(10X,I4,6X,I1,9X,F10.0)
6250 FORMAT(10X,5F10.3)
6310 FORMAT( I10, 4(1X,F10.4) )
6320 FORMAT(1H0,///,4X,17H8. PLOTTING DATA )
6330 FORMAT(/10X,10A4)
6340 FORMAT(1H0,//////, 4X,38H*** CALCULATION NORMALLY FINISHED. ***)
     END


     SUBROUTINE JULY31(T,DT,XO,XN,NMAX,Q,T1)
     IMPLICIT  REAL*8(A-H,O-Z)
     DIMENSION XO(1),XN(1),Q(1),T1(1)
     COMMON/RK/ IGO,IAG,IVSG(10),IMG980(2)
C
     IGO = 0
     IAG = 0
     DO 100 I=1,10
 100 IVSG(I) = 0
     DO 102 I=1,2
 102 IMG980(I) = 0
C
     DO 110  N=1,NMAX
     T1(N) = DT*FUN(N,T,XO)
     XN(N) = XO(N) + 0.5*T1(N)
 110 Q (N) = T1(N)
     IGO = 1
     T9 = T + DT/2.0
     DO 120  N=1,NMAX
 120 T1(N) = DT*FUN(N,T9,XN)
     DO 130  N=1,NMAX
     XN(N) = XO(N) + 0.5*T1(N)
 130 Q (N) = Q (N) + 2.0*T1(N)
     DO 140  N=1,NMAX
 140 T1(N) = DT*FUN(N,T9,XN)
     DO 150  N=1,NMAX
     XN(N) = XO(N) + T1(N)
 150 Q (N) = Q (N) + 2.0*T1(N)
     T9 = T + DT
     DO 160  N=1,NMAX
 160 T1(N) = DT*FUN(N,T9,XN)
     DO 170  N=1,NMAX
 170 XN(N) = XO(N) + ( T1(N) + Q(N) )/6.0
     RETURN
     END
```

Appendix F (Continued)

```
      FUNCTION FUN(N,T,X)
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON/IRIR/ XB(10),VK,VC,DK,DC,BK,BC,HH,BB,EI,EM,WI,NB,ITI(18)
      COMMON/COEF/ F1,  F2,
     1             V1K, V2K, V3K, B1K, B2K, B3K,
     2             V1C, V2C, V3C, B1C, B2C, B3C, UV
      COMMON /FMFMFM/ FGM , FRI , FRM , VFZ , VYM , BWF , BWM ,
     1                FWF(4) , FWM(4)
      COMMON/RK/ IGO,IAG,IVSG(10),IMG980(2)
      COMMON/XXD/A(6)
      COMMON/XDSDSX/ XDIS(6), UB
      COMMON/CXXCXX/ CXX
      COMMON /WIRE/ RWI(4), ZWI(4), ZWCI(4), WGEO(3,4), WDISP(3,2,4),
     1              WCORDO(3,2,4), WCORD(3,2,4), WLENGO(4), WLENG(4),
     2              WFORC(4),WFCMP(3,4), WMCMP(3,4), IWSTAT(4),NWIRE
      COMMON /CASK/ RCSK(2), ZCSK, WCSK, CSKI, BETA
      DIMENSION  X(1)
      CBA = 1.0E0
      IF( N .NE. 1 ) GO TO 400
      BWF    = 0.0
      BWM    = 0.0
      FGM    = 0.0
      FRI    = 0.0
      FRM    = 0.0
      VFZ    = 0.0
      VYM    = 0.0
C
      DO 100 I=1,4
      DO 100 J=1,3
      WFCMP(J,I) = 0.0
      WMCMP(J,I) = 0.0
  100 CONTINUE
C
C     FGM MOMENT BY G
C     FRI FRICTION FORCE , VFZ V-SPRING FORCE ,
C     FRM FRICTION MOMENT, VYM V-SPRING MOMENT,
C***  FRICTION FORCE
C
      ALPHA = X(1)
      SN1 =SSIN( X(1) )
      CS1 =GCOS( X(1) )
      IF(IGO.EQ.0.AND.ALPHA.LT.0.) IAG    = 1
      IF(IAG.EQ.1)           GO TO  110
      V = X(4) - ( HH*CS1 + BB*SN1 ) * X(2) - FUNU1(T)
      FF =-DSIGN(CBA,V)*FRIC(F1, F2, EM, UV)
      FM =-FF * ( HH*CS1 + BB*SN1 )
                                GO TO 120
  110 CONTINUE
      V = X(4) - ( HH*CS1 - BB*SN1 )*X(2) - FUNU1(T)
      FF =-DSIGN(CBA,V)*FRIC(F1, F2, EM, UV)
      FM =-FF * ( HH*CS1 - BB*SN1 )
  120 CONTINUE
      FRI    = FRI    + FF
      FRM    = FRM    + FM
C
C***  VERTICAL SPRING
C
      DO  130  IV=1,NB
```

Appendix F (Continued)

```
      ETA = 0.5*( FUNWO(T) - ( X(5) - HH*( 1. - CS1 ) - XB(IV)*SN1))
      IF(IGO.EQ.0.AND.ETA.LE.0.)    IVSG(IV) = 1
      IF(IVSG(IV).EQ.1)      GO TO  130
      ETD = 0.5*( FUNW1(T) - ( X(6) + ( HH*SN1 - XB(IV)*CS1 )*X(2)))
      VC  = V1C     + ETD*( V2C      + ETD*V3C      )
      VK  = V1K     + ETA*( V2K      + ETA*V3K      )
      FN  = -VK*ETA - VC*ETD
      IF(FN.GE.0.0)        FN = 0.0
      FM  = FN * ( -HH*SN1 + XB(IV)*CS1 )
      VFZ     = VFZ      + FN
      VYM     = VYM      + FM
  130 CONTINUE
C
C***  MOMENT
C
      CALL MOMNT(FGM,X)
C
C     UB  = FUNBO(T)
      UB  = FUNUO(T)
C     UD  = FUNB1(T)
      UD  = FUNU1(T)
      CONTINUE
      NFB  = 0
      NFR  = 0
      NFL  = 0
      VJB  = 0.0
      VJR  = 0.0
      VJL  = 0.0
      SN1  = SSIN( X(1) )
      CS1  = GCOS( X(1) )
C
C *** LOWER RIGHT
C
      XI  = X(3) + BB*( CS1 - 1.0 ) - HH*SN1 - UB
      IF(XI.LE.0.0)                           GO TO  200
      XT  = X(4) - ( BB*SN1 + HH*CS1 )*X(2) - UD
      BK  = B1K     + XI*( B2K      + XI*B3K      )
      BC  = B1C     + XT*( B2C      + XT*B3C      )
      FB  = BK*XI + BC*XT
      IF(FB.LE.0.)      FB = 0.
      BWF     = BWF      - FB
      BWM     = BWM      + FB*( HH*CS1 + BB*SN1 )
C
C *** LOWER LEFT
C
  200 XI  = -( X(3) - BB*( CS1 - 1.0 ) - HH*SN1 ) + UB
      IF(XI.LE.0.)                            GO TO  300
      XT  = -X(4) - ( BB*SN1 - HH*CS1 )*X(2) + UD
      BK  = B1K     + XI*( B2K      + XI*B3K      )
      BC  = B1C     + XT*( B2C      + XT*B3C      )
      FB  = BK*XI + BC*XT
      IF(FB.LE.0.)      FB = 0.
      BWF     = BWF      + FB
      BWM     = BWM      - FB*( HH*CS1 - BB*SN1 )
  300 CONTINUE
C
C***  WIRE   DISP , FORCE
C
```

Appendix F (Continued)

```
      IF( NWIRE .EQ. 0 ) GO TO 350
      CALL  WIRDSP( 4, T, X, ZWCI, RCSK, ZCSK, BETA, WGEO, WDISP, WCORD.
     1                 WCORDO, WLENG, WLENGO, IWSTAT, WFORC, WFCMP, WMCMP,
     2                 IGO)
  350 CONTINUE
  400 CONTINUE
C
      FCX   =   CXX*X(4)
C
C***  DERIVATIVE
C
      IF(N .EQ.1)    FUN = X(2)
      IF(N .EQ.2)    THEN
                     FUN = (FGM    +FRM    +VYM    +BWM    )/EI
                     FUN1 = 0.0
                     DO  500 I=1,4
                     FUN1= FUN1+WMCMP(2,I)/EI
  500                CONTINUE
                     FUN = FUN + FUN1
      ENDIF
      IF(N .EQ.3)    FUN = X(4)
      IF(N .EQ.4)    THEN
                     FUN = (FRI +BWF   )/EM - FCX/EM
                     FUN1 = 0.0
                     DO  600 I=1,4
                     FUN1= FUN1+WFCMP(1,I)/EM
  600                CONTINUE
                     FUN = FUN + FUN1
      ENDIF
      IF(N .EQ.5)    FUN = X(N+1)
      IF(N .EQ.6)    THEN
                     FUN = -(VFZ +WI )/EM
                     FUN1 = 0.0
                     DO  700 I=1,4
                     FUN1= FUN1+WFCMP(3,I)/EM
  700                CONTINUE
                     FUN = FUN + FUN1
      ENDIF
      A(N) = FUN
      RETURN
      END
      SUBROUTINE MOMNT(FXS,X)
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON/IRIR/ XB(10),VK(6),HH,BB,EI,EM,WI,NB,ITI(18)
      DIMENSION  X(1)
      COMMON/RK/ IGO,IAG,IVSG(10),IMG980(2)
      REAL IMG980
      ALPHA = X(1)
      IF(IGO.EQ.0)        GO TO 180
  170 IMG980(2) = ALPHA
  180 IF(IMG980(2))         190,   200,   210
  190 FXS     = FXS     + WI * ( BB * DCOS(X(1)) + HH * DSIN(X(1)))
                                       GO TO 220
  200 FXS     = FXS     + WI * HH * DSIN(X(1))
                                       GO TO 220
  210 FXS     = FXS     - WI * ( BB * DCOS(X(1)) - HH * DSIN(X(1)))
  220 CONTINUE
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE  WIRDSP
     1            ( NW, T, X, ZWCI, RCSK, ZCSK, BETA, WGEO, WDISP, WCORD
     2              WCORDO, WLENG, WLENGO, IWSTAT, WFORC, WFCMP, WMCMP,
     3              IGO)
      IMPLICIT  REAL*8(A-H,O-Z)
C
      DIMENSION  X(1), ZWCI(1), WDISP(3,2,1), WCORDO(3,2,1), WLENGO(1),
     1           WCORD(3,2,1), WLENG(1), WFORC(1), WFCMP(3,1),
     2           WMCMP(3,1), WGEO(3,1), DCMP(3),IWSTAT(1),
     3           SW(4),SANG(4), ALCMP(3)
C
      DATA  SW  /        1.0,           1.0,-1.0,        -1.0 /
      DATA  SANG/        0.0,  1.57079632, 0.0, 1.57079632 /
      DATA  EPSL/        0.0/
C
      SN   = DSIN(X(1))
      DO  10  I=1,NW
      IF( IWSTAT(I) .EQ. 3 )    GO TO 10
      YM   = WGEO(1,I)
      AR   = WGEO(2,I)
      SIGB = WGEO(3,I)
      ZZ   = ZWCI(I) - ZCSK
      BANG = DABS(SANG(I) - BETA)
      R2BT = RCSK*DCOS( BANG )
      CS   = SW(I) * (1.0-DCOS(X(1)))
      DCMP(1)       = ZZ*SN + R2BT*CS
      DCMP(2)       = 0.0
      DCMP(3)       = - ZZ*CS + R2BT*SN*SW(I)
      WDISP(1,1,I) = X(3) + DCMP(1)
      WDISP(2,1,I) = 0.0
      WDISP(3,1,I) = X(5) + DCMP(3)
      WDISP(1,2,I) = FUNUO(T)
      WDISP(2,2,I) = 0.0
      WDISP(3,2,I) = FUNWO(T)
C
      AL   =  0.0
      DO  20  J=1,3
      WCORD(J,1,I)  =  WCORDO(J,1,I)+WDISP(J,1,I)
      WCORD(J,2,I)  =  WCORDO(J,2,I)+WDISP(J,2,I)
      ALCMP(J) = WCORD(J,1,I) - WCORD(J,2,I)
      AL   = AL +  ALCMP(J)**2
   20 CONTINUE
C
      WLENG(I) = DSQRT( AL )
      DLENG    = WLENG(I) - WLENGO(I)
      IF( IGO .EQ. 1 .AND. IWSTAT(I) .EQ. 2 ) THEN
                WFORC(I) = 0.0
                IWSTAT(I) = 2
                GO TO 30
      ENDIF
      IF( DLENG .LE. EPSL )    THEN
                WFORC(I) = 0.0
                IWSTAT(I) = 2
                GO TO 30
      ELSE
                WFORC(I) = DLENG/WLENGO(I) * YM
                IF( WFORC(I) .GT. SIGB )   THEN
                             WFORC(I) = 0.0
```

Appendix F (Continued)

```
                  IWSTAT(I) = 3
                  GO TO 30
            ENDIF
            WFORC(I) = WFORC(I) * AR *(-1.0)
            IWSTAT(I) = 1
      ENDIF
   30 CONTINUE
      DO  40  J=1,3
        ALCMP(J) = ALCMP(J)/WLENG(I)
        WFCMP(J,I) = WFORC(I)*ALCMP(J)
   40 CONTINUE
        WMCMP(1,I) = 0.0
        WMCMP(2,I) = -(WFCMP(1,I)*(WCORD(3,1,I)-(ZCSK+X(5)))
     1             - WFCMP(3,I)*(WCORD(1,1,I)-X(3)))
        WMCMP(3,I) = 0.0
   10 CONTINUE
      RETURN
      END


      SUBROUTINE  WIRINI
     1            ( NW, WCORDO, WLENGO, RWI, ZWI, ZWCI, RCU, BETA)
      IMPLICIT  REAL*8(A-H,O-Z)
C
      DIMENSION  WCORDO(3,2,1), WLENGO(1), RWI(1), ZWI(1), ZWCI(1),
     1           SFANG(4)
C
      DATA  SFANG/ 3.14159265, -1.57079632, 0.0, 1.57079632 /
C
      DO  10  I=1,NW
      TH   = SFANG(I) - BETA
      CS   = DCOS( TH )
      SN   = DSIN( TH )
      WCORDO(1,2,I) = RWI(I)* CS
      WCORDO(2,2,I) = RWI(I)* SN
      WCORDO(3,2,I) = ZWI(I)
      WCORDO(1,1,I) = RCU   * CS
      WCORDO(2,1,I) = RCU   * SN
      WCORDO(3,1,I) = ZWCI(I)
      AL   =  0.0
      DO  20  J=1,3
      AL   =  AL + ( WCORDO(J,1,I) - WCORDO(J,2,I) )**2
   20 CONTINUE
      WLENGO(I) = DSQRT(AL)
   10 CONTINUE
      RETURN
      END


      FUNCTION FRIC(F1, F2, EM, UV)
      IMPLICIT  REAL*8(A-H,O-Z)
      T  = UV * UV
      FRIC = F1 + 0.5 * F2 * EM * T
      RETURN
      END
```

Appendix F (Continued)

```
      FUNCTION FUNUO(T)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /AAWWPP/ A, W, P
      COMMON /KIKKIK/ Z1, Z2, Z3, Z4, Z5, Z6, KIK
      COMMON /FRIFRI/  FPAI
      COMMON /KXKXK/ DDT(3,2),VX(3,2,1100), KXK(3), KXN(3,2)
      COMMON /PDPXPD/ PX,PD,PD2,PXX,PDD,PDW2
      COMMON /ALPBET/ ALP,BET
C
      PAI=3.14159265359D0
      IF(KXK(1).NE.0) GO TO 15
      IF ( KIK .EQ. 1 )  GO TO 10
      IF (KIK.EQ.2) GOTO 12
      IF(KIK.EQ.3) GO TO 13
      FUNUO = A *  DSIN( W * T - P )
      PXX=FUNUO
      RETURN
   10 CONTINUE
      CALL  FIT ( FPAI,  T )
      PF = 2.0 * PAI * FPAI
      PF2 = PF * PF
      PXX =  A / PF2 *   DSIN( PF   * T - P )
      FUNUO = PXX
      RETURN
   12 CONTINUE
      PF = 2.0*PAI*W
      PF2=PF*PF
      PXX=A/PF2*DSIN(PF*T-P)
      FUNUO=PXX
      RETURN
   13 CONTINUE
      CALL FIT(FPAI,T)
      PF=2.0*PAI*FPAI
      PXX=A*DSIN(PF*T-P)
      FUNUO=PXX
      RETURN
   15 CONTINUE
      N = T/DDT(1,1)
      FUNUO = (T-DDT(1,1)*N)*(VX(1,1,N+2)-VX(1,1,N+1))
     1       /DDT(1,1)+VX(1,1,N+1)
      FUNUO=ALP*FUNUO
      PXX=FUNUO
      RETURN
C
      ENTRY FUNU1(T)
      PAI=3.14159265359D0
      IF(KXK(1).NE.0) GO TO 25
      IF ( KIK .EQ. 1 )  GO TO 20
      IF(KIK.EQ.2) GO TO 22
      IF(KIK.EQ.3) GO TO 23
      FUNU1 = A * W * DCOS(W*T-P)
      PDD=FUNU1
      RETURN
   20 CONTINUE
      CALL  FIT ( FPAI,  T )
      PF = 2.0 * PAI * FPAI
      PDD =  A / PF *   DCOS( PF   * T - P )
      FUNU1 = PDD
```

Appendix F (Continued)

```
      RETURN
   22 CONTINUE
      PF=2.0*PAI*W
      PDD=A/PF*DCOS(PF*T-P)
      FUNU1=PDD
      RETURN
   23 CONTINUE
      CALL FIT(FPAI,T)
      PF=2.0*PAI*FPAI
      PDD=A*DCOS(PF*T-P)*PF
      FUNU1=PDD
      RETURN
   25 CONTINUE
      N = T/DDT(1,2)
      FUNU1 = (T-DDT(1,2)*N)*(VX(1,2,N+2)-VX(1,2,N+1))
     1        /DDT(1,2)+VX(1,2,N+1)
      PDD=FUNU1
      NN=(T-0.01)/DDT(1,2)
      TT=T-0.01
      PDW2=(TT-DDT(1,2)*NN)*(VX(1,2,NN+2)-VX(1,2,NN+1))
     1      /DDT(1,2)+VX(1,2,NN+1)
      FUNU1=BET*FUNU1
      PDD=FUNU1
      RETURN
      END


      FUNCTION FUNWO(T)
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON /KXKXK/ DDT(3,2),VX(3,2,1100), KXK(3), KXN(3,2)
      COMMON /UUIDID/ UUIDO, UUID1
      IF(KXK(3).NE.0) GO TO 15
      FUNWO = 0.0
      UUIDO=FUNWO
      RETURN
   15 CONTINUE
      N = T/DDT(3,1)
      FUNWO = (T-DDT(3,1)*N)*(VX(3,1,N+2)-VX(3,1,N+1))
     1        /DDT(3,1)+VX(3,1,N+1)
      UUIDO=FUNWO
      RETURN
      ENTRY FUNW1(T)
      IF(KXK(3).NE.0) GO TO 25
      FUNW1 = 0.0
      UUID1=FUNW1
      RETURN
   25 CONTINUE
      N = T/DDT(3,2)
      FUNW1 = (T-DDT(3,2)*N)*(VX(3,2,N+2)-VX(3,2,N+1))
     1        /DDT(3,2)+VX(3,2,N+1)
      UUID1=FUNW1
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE  GREATT( X, A, WFORC )
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON / GRETTT/ GFGMG, GFRIG, GFRMG, GVFZG, GVYMG, GBWFG, GBWMG,
     1                  GMXX(6), GMAA(6), GFWF(4)
      COMMON /FMFMFM/ FGM , FRI , FRM , VFZ , VYM , BWF , BWM ,
     1                FWF(4) , FWM(4)
      DIMENSION  X(1), A(1), WFORC(1)
      DO 100 I=1,6
      GMXX(I)=  DMAX1(GMXX(I),DABS( X(I)) )
  100 CONTINUE
      GMAA(2)=  DMAX1(GMAA(2),DABS( A(2)) )
      GMAA(4)=  DMAX1(GMAA(4),DABS( A(4)) )
      GMAA(6)=  DMAX1(GMAA(6),DABS( A(6)) )
      GFGMG  =  DMAX1( GFGMG, DABS( FGM ) )
      GFRIG  =  DMAX1( GFRIG, DABS( FRI ) )
      GFRMG  =  DMAX1( GFRMG, DABS( FRM ) )
      GVFZG  =  DMAX1( GVFZG, DABS( VFZ ) )
      GVYMG  =  DMAX1( GVYMG, DABS( VYM ) )
      GBWFG  =  DMAX1( GBWFG, DABS( BWF ) )
      GBWMG  =  DMAX1( GBWMG, DABS( BWM ) )
      GFWF(1)=  DMAX1(GFWF(1),DABS(WFORC(1)))
      GFWF(2)=  DMAX1(GFWF(2),DABS(WFORC(2)))
      GFWF(3)=  DMAX1(GFWF(3),DABS(WFORC(3)))
      GFWF(4)=  DMAX1(GFWF(4),DABS(WFORC(4)))
      RETURN
      ENTRY  BIGLK ( X, A, WFORC )
      WRITE(6,610)
      GMXX(1) = 180.0 / 3.14  * GMXX(1)
      WRITE(6,660)  GMXX(1), GMXX(3), GMXX(5)
      WRITE(6,620)
      WRITE(6,660)  GMXX(2), GMXX(4), GMXX(6)
      WRITE(6,630)
      WRITE(6,660)  GMAA(2), GMAA(4), GMAA(6)
      WRITE(6,640)
      WRITE(6,670)  GFGMG, GFRIG, GFRMG, GVFZG, GVYMG, GBWFG, GBWMG
      WRITE(6,650)
      WRITE(6,680)  GFWF(1), GFWF(2), GFWF(3), GFWF(4)
      DO 200 N=1,6
      GMXX(N) = 0.DO
  200 GMAA(N) = 0.DO
      DO 300 N=1,4
  300 GFWF(N) = 0.DO
  610 FORMAT(1H1,4X,39H9. CALCULATION RESULTS (MAXIMUM VALUES)//
     1         8X,16H9.1 DISPLACEMENT// 16X,13HTHETA(DEGREE),
     2         13X,1HU,13X,1HW )
  620 FORMAT(1H0,///,8X,12H9.2 VELOCITY,//,18X,11HD-THETA/D-T,
     1         7X, 7HD-U/D-T, 7X, 7HD-W/D-T )
  630 FORMAT(1H0,///,8X,16H9.3 ACCELERATION,//,18X,11HD2THETA/DT2,
     1         7X, 7HD2U/DT2, 7X, 7HD2W/DT2 )
  640 FORMAT(1H0,///,8X,20H9.4 FORCE AND MOMENT //, 19X, 9HMOMENT BY,
     1         6X,8HFORCE BY,5X,9HMOMENT BY,6X,8HFORCE BY,
     2         5X,9HMOMENT BY,6X,8HFORCE BY,5X,9HMOMENT BY,/
     3         21X,7HGRAVITY,6X,8HFRICTION,6X, 8HFRICTION,
     4         2(6X,8HV.SPRING),2(6X,8HH.SPRING),)
  650 FORMAT(1H0,///, 8X,14H9.5 WIRE FORCE,// 20X, 8HFORCE BY,
     1         6X,8HFORCE BY,6X,8HFORCE BY,6X,8HFORCE BY,/ 22X,6HWIRE-A,
     2         8X,6HWIRE-B,8X,6HWIRE-C,8X,6HWIRE-D)
  660 FORMAT(15X, 3( 2X, 1PE12.5 ), 0P )
```

Appendix F (Continued)

```
 670 FORMAT(15X, 7( 2X, 1PE12.5 ), 0P )
 680 FORMAT(15X, 4( 2X, 1PE12.5 ), 0P )
     RETURN
     END

     SUBROUTINE  FIT ( F, T )
     IMPLICIT  REAL*8(A-H,O-Z)
     COMMON /AAWWPP/ A, W, P
     COMMON /KIKKIK/ Z1,Z2,Z3,Z4,Z5,Z6,KIK
     F = W + Z1*T + Z2*T*T + Z3*T**3 + Z4*T**4 + Z5*T**5 +Z6*T**6
     RETURN
     END


     FUNCTION SSIN(X)
     IMPLICIT  REAL*8(A-H,O-Z)
     SSIN = X*(1.0-(1.0-X*X/20.0)*X*X/6.0)
     RETURN
     END


     FUNCTION GCOS(X)
     IMPLICIT  REAL*8(A-H,O-Z)
     GCOS = 1.0-X*X*(1.0-X*X/12.0)*0.5
     RETURN
     END

     SUBROUTINE ZEROCL
     COMMON /B/ BB(58940)
     COMMON /C/ IUWT(3,2),IPLTT(21), CC(63)
     CHARACTER*20 IPLTT
     DO 10 N=1,58940
  10 BB(N)=0.0
     DO 20 N=1,63
  20 CC(N)=0.0
     RETURN
     END


     SUBROUTINE  DTVWST( TIM, UB, XN, HH, BB )
     IMPLICIT  REAL*8 (A-H,O-Z)
     REAL*4 S,T,STEP,TMM,US,X(6)
     COMMON/LUNITL/ LUCD, LUHT, LUDP, LUPR
     DIMENSION  XN(1)
     TMM = TIM
     US  = UB
     DO 100 N=1,6
 100 X(N) = XN (N)
     STEP = 12.0 + 20.0 * BB / HH
     FACT = 80.0 / HH
     S = 4.0 * STEP
     T = FUNWO(TIM) * FACT
C    UO = FUNBO(TIM)
     UO = FUNUO(TIM)
     WRITE(LUDP) TMM, US, S, T, (X(I),I=1,6)
     RETURN
     END
```

Appendix F (Continued)

```
C*********************************************************************C
C                                                                     C
      SUBROUTINE DFPLOT
C                                                                     C
C     PROGRAM NAME  -  ROCKING-DISPLACEMENT-PLOT                      C
C                                                                     C
C*********************************************************************C
      DIMENSION  X(6)
      DATA  XORG, YORG/ 140.0, 20.0 /
      COMMON /LUNITL/ LUCD, LUHT, LUDP, LUPR
      REWIND LUDP
      READ(LUDP,END=200)  HH, BB
      STEP = 12.0 + 20.0 * BB / HH
      FACT = 80.0 / HH
  100 CONTINUE
      READ(LUDP,END=200) TIME, UB, S, T, (X(I),I=1,6)
      XXX  = 100.0
      CALL SYMBOL( XXX , 190.0, 4.0, 6HTIME =, 0.0, 6 )
      XXX = XXX + 24.0
      CALL NUMBER( XXX, 190.0, 4.0, TIME, 0.0, 3 )
      XXX = XXX + 26.0
      CALL SYMBOL( XXX, 190.0, 4.0, 5H(SEC), 0.0, 5 )
      CALL PLOT(XORG-S,YORG+T,3)
      CALL PLOT(XORG+S,YORG+T,2)
C     CX = (X(3)-UB)*10.*FACT  + XORG
C     CY = (HH + 10.*X(5))*FACT + YORG
C     ST = SIN(10.*X(1))
C     CT = COS(10.*X(1))
      CX = (X(3)-UB)*    FACT  + XORG
      CY = (HH +     X(5))*FACT + YORG
      ST = SIN(    X(1))
      CT = COS(    X(1))
      X1 = HH*ST + BB*CT
      X2 = HH*ST - BB*CT
      Y1 = HH*CT - BB*ST
      Y2 = HH*CT + BB*ST
      SO = CX + X1*FACT
      TO = CY + Y1*FACT
      CALL PLOT(SO,TO,3)
      S1 = CX + X2*FACT
      T1 = CY + Y2*FACT
      CALL PLOT(S1,T1,2)
      S1 = CX - X1*FACT
      T1 = CY - Y1*FACT
      CALL PLOT(S1,T1,2)
      S1 = CX - X2*FACT
      T1 = CY - Y2*FACT
      CALL PLOT(S1,T1,2)
      CALL PLOT(SO,TO,2)
      CALL PLOT(  0.0, 0.0, 3 )
      CALL PLOT(  10.0*STEP, 0.0, -3 )
      CALL PLOT(  0.0,0.0, 666 )
      CALL PLOT(  0.0,0.0, 777 )
      CALL PLOT(  0.0,0.0, 888 )
      GO TO 100
  200 CONTINUE
      RETURN
      END
```

Appendix F (Continued)

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE XYPLOT
C                                                                      C
C     ROCKING CODE TIME HISTORY CURVE PLOT                             C
C                                                                      C
C                                                                      C
C     COMMON / A/  DATA RECORD 1.                                     C
C              N       - BLOCK SU.                                    C
C              TO      - TIME (FIRST).                                C
C              T1      - TIME (LAST).                                 C
C              IPL     - PLOT SELECT POINT                            C
C     COMMON / B/  DATA RECORD 2.                                     C
C              T       - TIME                                         C
C              U,U1,U2 - JISHIN-YOKO DISP , VELO , ACCELE .           C
C              W,W1,W2 - JISHIN-TATE  -    ,  -   ,   -    .          C
C              PLT1    - BLOCK PLOT DATA FIRST.                       C
C              PLT2    - BLOCK PLOT DATA SECOND.                      C
C              PLT3    - BLOCK PLOT DATA THIRD.                       C
C     COMMON / C/                                                     C
C              IUWT   - SUB.UWPLOT TITLE AREA.                        C
C              IPLTT  - SUB.BLKPLOT TITLE AREA.                       C
C              IDVA   - SUB.TITLE AREA.                               C
C              P-MAX  - PLT-MAX                                       C
C              P-MIN  - PLT-MIN                                       C
C              P-WIDE PLOT BLOCKS MAX WIDE.                           C
C                                                                      C
C     DIMENSION AREA.                                                 C
C             TEMP   PLT1 READ TEMPORARY AREA.                        C
C             NPL    PLOT SELECT CONTROL AREA.                        C
C                                                                      C
C     SUBROUTINE  UWPLOT                                              C
C             TATE,YOKO JISIN PLOT.                                   C
C     SUBROUTINE  BLKPLOT                                             C
C             BLOCKS MASS POINT YURE PLOT.                            C
C                                                                      C
C      SUBROUTINE  MEMOMIN                                            C
C             BLKPLOT Y-JIKUZAHYO MIN                                 C
C      SUBROUTINE  MEMOMAX                                            C
C             BLKPLOT Y-JIKUZAHYO MAX                                 C
C      SUBROUTINE  UMMIN                                              C
C             UWPLOT Y-JIKUZAHYO MIN                                  C
C      SUBROUTINE  UMMAXIKUZAHYO MIN                                  C
C             UWPLOT Y-JIKUZAHYO MAX                                  C
C      SUBROUTINE  MLOG                                               C
C             SHISU KEISAN                                            C
C      SUBROUTINE  RNUMB                                              C
C             SUJI MIGIZUME PLOT                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON / A/ N,TO,T1,IPL(21)
      COMMON / B/ T(2105),U(2105),U1(2105),U2(2105),W(2105),W1(2105),
     1            W2(2105),PLT1(21,2105)
      COMMON / C/ IUWT(3,2),IPLTT(21), P1MAX(21),
     1            P1MIN(21),P1WIDE(21)
      COMMON / D/ IPLT2,FIPLT2
      COMMON /LUNITL/ LUCD, LUHT, LUDP, LUPR
      CHARACTER*20 IPLTT
      REAL*8  TDPO, TDP1, TDP, UDP, UDP1, UDP2, WDP, WDP1, WDP2, TEMP
```

Appendix F (Continued)

```
      DIMENSION TEMP(21),NPL(21)
      DATA IUWT/'U   ','U1  ','U2  ','W   ','W1  ','W2  '/
      DATA  (IPLTT(J),J=1,9) /
     1       'THETA - (DEGREE)      ',
     2       'THETA - VELOCITY      ',
     3       'THETA - ACCELERATION',
     4       '     U - DISPLACEMENT',
     5       '     U - VELOCITY     ',
     6       '     U - ACCELERATION',
     7       '     W - DISPLACEMENT',
     8       '     W - VELOCITY     ',
     9       '     W - ACCELERATION'/
      DATA  (IPLTT(J),J=10,21) /
     1       'MOMENT BY GRAVITY     ',
     2       'FORCE BY FRICTION     ',
     3       'MOMENT BY FRICTION    ',
     4       'FORCE BY V.SPRING     ',
     5       'MOMENT BY V.SPRING    ',
     6       'FORCE BY B.WALL       ',
     7       'MOMENT BY B.WALL      ',
     8       'FORCE BY WIRE-A       ',
     9       'FORCE BY WIRE-B       ',
     A       'FORCE BY WIRE-C       ',
     B       'FORCE BY WIRE-D       ',
     C       '                      '/
C
      DO 100 I=1,21
  100 TEMP(I)=0.0
      CALL ZEROCL
      FIPLT2=200.0
      REWIND LUHT
      READ(LUHT,END=180)  N, TDPO, TDP1
      READ(LUHT) NPL
C
C        - PLOT DATA SELECT POINT SET (IPL) -
C
      J=1
      DO 110 I=1,21
      IF(NPL(I).EQ.0) GO TO 110
      IPL(J)=I
      NDATA = J
      J=J+1
  110 CONTINUE
      UMAX=0.
      UMIN=0.
      U1MAX=0.
      U1MIN=0.
      U2MAX=0.
      U2MIN=0.
      WMAX=0.
      WMIN=0.
      W1MAX=0.
      W1MIN=0.
      W2MAX=0.
      W2MIN=0.
      P1MAXX=0.
C
C     ... PLOT DATA READ ...
```

Appendix F (Continued)

```
C
      II = 0
  120 CONTINUE
      II = II + 1
      IF(II.EQ.2101) GO TO 180
      READ(LUHT,END=180)  TDP, UDP, UDP1, UDP2, WDP, WDP1, WDP2 ,
     1        (TEMP(I),I=1,NDATA)
      T(II) = TDP
      U(II) = UDP
      U1(II) = UDP1
      U2(II) = UDP2
      W(II) = WDP
      W1(II) = WDP1
      W2(II) = WDP2
C
C       - PLOT DATA MOVE -
C
      DO 140 I=1,NDATA
      PLT1(I,II)=TEMP(I)
  140 CONTINUE
C
C       - MAX . MIN -
C
      IF( U(II).GT.UMAX )      UMAX=U(II)
      IF( U(II).LT.UMIN )      UMIN=U(II)
      IF(U1(II).GT.U1MAX)     U1MAX=U1(II)
      IF(U1(II).LT.U1MIN)     U1MIN=U1(II)
      IF(U2(II).GT.U2MAX)     U2MAX=U2(II)
      IF(U2(II).LT.U2MIN)     U2MIN=U2(II)
      IF( W(II).GT.WMAX )      WMAX=W(II)
      IF( W(II).LT.WMIN )      WMIN=W(II)
      IF(W1(II).GT.W1MAX)     W1MAX=W1(II)
      IF(W1(II).LT.W1MIN)     W1MIN=W1(II)
      IF(W2(II).GT.W2MAX)     W2MAX=W2(II)
      IF(W2(II).LT.W2MIN)     W2MIN=W2(II)
C
      DO 150 I=1,NDATA
      IF(PLT1(I,II).GT.P1MAX(I))          P1MAX(I)=PLT1(I,II)
      IF(PLT1(I,II).LT.P1MIN(I))          P1MIN(I)=PLT1(I,II)
  150 CONTINUE
C
      IF ( T(II) .GE. TDP1 ) GO TO 180
      GO TO 120
  180 CONTINUE
C
C     CHECK WRITE
C
      WRITE(LUPR,6000) UMAX, UMIN
      WRITE(LUPR,6100) U1MAX, U1MIN
      WRITE(LUPR,6200) U2MAX, U2MIN
      WRITE(LUPR,6300) WMAX, WMIN
      WRITE(LUPR,6400) W1MAX, W1MIN
      WRITE(LUPR,6500) W2MAX, W2MIN
 6000 FORMAT(1H1,3X,17H10. PLOTTING DATA //
     1       8X,25HHORIZONTAL DISPLACEMENT  ,11H MAX ,MIN  ,2E15.7,/)
 6100 FORMAT(8X,25HHORIZONTAL VELOCITY      ,11H MAX ,MIN  ,2E15.7,/)
 6200 FORMAT(8X,25HHORIZONTAL ACCELERATION  ,11H MAX ,MIN  ,2E15.7,/)
 6300 FORMAT(8X,25HVERTICAL   DISPLACEMENT  ,11H MAX ,MIN  ,2E15.7,/)
```

Appendix F (Continued)

```
 6400 FORMAT(8X,25HVERTICAL   VELOCITY      ,11H MAX ,MIN  ,2E15.7,/)
 6500 FORMAT(8X,25HVERTICAL   ACCELERATION  ,11H MAX ,MIN  ,2E15.7,/)
C
C     CHECK WRITE
C
CJR   WRITE(LUPR,6600) II,T(II)
 6600 FORMAT(1H , 5X, 7HCOUNT =, I4, 10X, 10HLASTTIME =, F10.5)
      DO 190 I=1,NDATA
      WRITE(LUPR,6700) IPLTT(IPL(I)),P1MAX(I),P1MIN(I)
  190 CONTINUE
 6700 FORMAT(11X,A20,2X,11H MAX ,MIN  ,2E15.7,/)
      DO 200 I=1,NDATA
      PPP = ABS(P1MAX(I))
      IF(PPP.LT.ABS(P1MIN(I)))  PPP = ABS(P1MIN(I))
      P1MAX(I) =  PPP
      P1MIN(I) = -PPP
  200 CONTINUE
C
C        - PLOTS -
C
      CALL  PLOT(0.0,0.0,-3)
C
C
C      - DATA SCALING -
C     .. T ..
C
      T(II+2)=T(II+1)
      III=II+2
C
      CALL SCALE(T,FIPLT2,III,1,10.0)
      DO 210 K=1,II
  210 T(K)=T(K)/T(II+4)
C
C ... SUBROUTINE UWPLOT SET AND CALL ...
C
      UWIDE = ABS(UMAX - UMIN)
      U1WIDE = ABS(U1MAX - U1MIN)
      U2WIDE = ABS(U2MAX - U2MIN)
      WWIDE = ABS(WMAX - WMIN)
      W1WIDE = ABS(W1MAX - W1MIN)
      W2WIDE = ABS(W2MAX - W2MIN)
C
      UU=ABS(UMAX)+ABS(UMIN)+ABS(U1MAX)+ABS(U1MIN)+ABS(U2MAX)+ABS(U2MIN)
      IF(UU.EQ.0.0) GO TO 400
C
C      - DATA SCALING -
C     .. U ..
C
      IF(UWIDE.EQ.0.0) GO TO 230
      TKK= 20.0/UWIDE
      DO 240 I=1,II
  240 U(I)= TKK* ABS(U(I)-UMIN)
      GO TO 250
  230 DO 260 I=1,II
  260 U(I)= 10.0
  250 IF(U1WIDE.EQ.0.0) GO TO 280
      TKK= 20.0/U1WIDE
      DO 270 I=1,II
```

Appendix F (Continued)

```
  270 U1(I)= TKK* ABS(U1(I)-U1MIN)
      GO TO 300
  280 DO 290 I=1,II
  290 U1(I)= 10.0
  300 IF(U2WIDE.EQ.0.0) GO TO 310
      TKK= 20.0/U2WIDE
      DO 320 I=1,II
  320 U2(I)= TKK* ABS(U2(I)-U2MIN)
      GO TO 340
  310 DO 330 I=1,II
  330 U2(I)= 10.0
  340 CONTINUE
C
C      - U PLOT -
C
      IC=1
      CALL UWPLOT(UMAX,UMIN,U1MAX,U1MIN,U2MAX,U2MIN,UWIDE,U1WIDE,U2WIDE,
     1                      IC,II)
C
  400 WW=ABS(WMAX)+ABS(WMIN)+ABS(W1MAX)+ABS(W1MIN)+ABS(W2MAX)+ABS(W2MIN)
      IF(WW.EQ.0.0) GO TO 600
C      - DATA SCALING -
C     .. W ..
C
      IF(WWIDE.EQ.0.0) GO TO 420
      TKK= 20.0/WWIDE
      DO 410 I=1,II
  410 W(I)= TKK* ABS(W(I)-WMIN)
      GO TO 440
  420 DO 430 I=1,II
  430 W(I)= 10.0
  440 IF(W1WIDE.EQ.0.0) GO TO 460
      TKK= 20.0/W1WIDE
      DO 450 I=1,II
  450 W1(I)=TKK* ABS(W1(I)-W1MIN)
      GO TO 480
  460 DO 470 I=1,II
  470 W1(I)= 10.0
  480 IF(W2WIDE.EQ.0.0) GO TO 500
      TKK= 20.0/W2WIDE
      DO 490 I=1,II
  490 W2(I)= TKK* ABS(W2(I)-W2MIN)
      GO TO 520
  500 DO 510 I=1,II
  510 W2(I)= 10.0
  520 CONTINUE
C
C      - W PLOT -
C
      IC=2
      CALL UWPLOT(WMAX,WMIN,W1MAX,W1MIN,W2MAX,W2MIN,WWIDE,W1WIDE,W2WIDE,
     1                      IC,II)
C
  600 CONTINUE
C
C ... SUBROUTINE BLKPLOT SET AND CALL ...
C
      DO 650 I=1,NDATA
```

Appendix F (Continued)

```
      IC = IPL(I)
      IF( IC .EQ. 0 )   GO TO 650
      P1MAXX = 0.0
      P1WIDE(I) = ABS(P1MAX(I) - P1MIN(I))
      IF(P1WIDE(I).GT.P1MAXX)        P1MAXX = P1WIDE(I)
      CALL  BLKPLT( I, P1MAXX, IC, II )
  650 CONTINUE
      RETURN
      END


      SUBROUTINE UWPLOT(VMAX,VMIN,V1MAX,V1MIN,V2MAX,V2MIN,
     1                  VWIDE,V1WIDE,V2WIDE,IC,II)
      COMMON / A/ N,TO,T1,IPL(21)
      COMMON / B/ T(2105),U(2105),U1(2105),U2(2105),W(2105),W1(2105),
     1            W2(2105),PLT1(21,2105)
      COMMON / C/ IUWT(3,2),IPLTT(21), P1MAX(21),
     1            P1MIN(21),P1WIDE(21)
      CHARACTER*20 IPLTT
      COMMON / D/ IPLT2,FIPLT2
C
C       - U.W TITLE PLOT -
C
      YY=167.5
      DO 100 I=1,3
      YY=YY-30.0
      CALL SYMBOL(10.0,YY,5.0,IUWT(I,IC),0.0,3)
  100 CONTINUE
C
C       - D.V.A PLOT -
C
      YY=183.5
      DO 200 I=4,6
      YY=YY-30.0
      CALL SYMBOL( 33.0,YY,3.0,IPLTT(I)(9:20),0.0,12)
  200 CONTINUE
C
C     X-AXIS PLOT
C
      CALL AXIS(30.0,70.0,10HTIME(SEC)  ,-10,FIPLT2,0.0,T(II+3),
     1          T(II+4),10.0)
      YY=40.0
      DO 300 I=1,3
      YY=YY+30.0
      CALL PLOT(29.0,YY,3)
      CALL PLOT(30.0,YY,2)
      CALL PLOT(30.0,YY+20.0,2)
      CALL PLOT(29.0,YY+20.0,2)
      CALL PLOT(30.0,YY+10.0,3)
      CALL PLOT(FIPLT2+30.0,YY+10.0,2)
  300 CONTINUE
C
C       .. U2,W2 PLOT ..
C
      IF(V2WIDE.EQ.0.0)  GO TO 400
      CALL UMMIN(V2MIN,70.0)
      CALL UMMAX(V2MAX,87.0)
      GO TO 450
```

Appendix F (Continued)

```
  400 CALL NUMBER(25.0,79.0,2.0,V2WIDE,0.0,-1)
  450 CONTINUE
C
C       .. U1,W1 PLOT ..
C
      IF(V1WIDE.EQ.0.0)  GO TO 500
      CALL UMMIN(V1MIN,100.0)
      CALL UMMAX(V1MAX,117.0)
      GO TO 550
  500 CALL NUMBER(25.0,109.0,2.0,V1WIDE,0.0,-1)
  550 CONTINUE
C
C       .. U,W PLOT ..
C
      IF (VWIDE.EQ.0.0)  GO TO  600
      CALL UMMIN(VMIN,130.0)
      CALL UMMAX(VMAX,147.0)
      GO TO 650
  600 CALL NUMBER(25.0,139.0,2.0,VWIDE,0.0,-1)
  650 CONTINUE
C
C       - U.W PLOT SELECT -
C
      IIKK=II-1
      IF(IC.EQ.2) GO TO 760
C
      CALL PLOT(30.0,80.0,3)
      DO 700 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,70.0+U2(I),2)
  700 CONTINUE
      CALL PLOT(T(I+1)+30.0,70.0+U2(I),3)
      CALL PLOT(30.0,110.0,3)
      DO 720 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,100.0+U1(I),2)
  720 CONTINUE
      CALL PLOT(T(I+1)+30.0,100.0+U1(I),3)
      CALL PLOT(30.0,140.0,3)
      DO 740 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,130.0+U(I),2)
  740 CONTINUE
      CALL PLOT(T(I+1)+30.0,130.0+U(I),3)
      GO TO 900
C
  760 CONTINUE
C
C       .. W2 PLOT ..
C
      CALL PLOT(30.0,80.0,3)
      DO 800 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,70.0+W2(I),2)
  800 CONTINUE
      CALL PLOT(T(I+1)+30.0,70.0+W2(I),3)
      CALL PLOT(30.0,110.0,3)
      DO 820 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,100.0+W1(I),2)
  820 CONTINUE
      CALL PLOT(T(I+1)+30.0,100.0+W1(I),3)
      CALL PLOT(30.0,140.0,3)
```

Appendix F (Continued)

```
      DO 840 I=1,IIKK
      CALL PLOT(T(I+1)+30.0,130.0+W(I),2)
  840 CONTINUE
      CALL PLOT(T(I+1)+30.0,130.0+W(I),3)
C
  900 CONTINUE
C
      CALL PLOT(400.0,0.0,-3)
      CALL PLOT(0.0,0.0,666)
      CALL PLOT(0.0,0.0,777)
      CALL PLOT(0.0,0.0,888)
      RETURN
      END


      SUBROUTINE  BLKPLT(IJUN, PWMAX, IC, II )
      COMMON / A/ N,T0,T1,IPL(21)
      COMMON / B/ T(2105),U(2105),U1(2105),U2(2105),W(2105),W1(2105),
     1            W2(2105),PLT1(21,2105)
      COMMON / C/ IUWT(3,2),IPLTT(21), P1MAX(21),
     1            P1MIN(21),P1WIDE(21)
      CHARACTER*20 IPLTT
      COMMON / D/ IPLT2,FIPLT2
C
C     .. TITLE PLOT ..
C
       XX=30.0
      CALL SYMBOL(XX,200.0,5.0,IPLTT(IC),0.0,20)
C
C     .. X-AXIS PLOT ..
C
      CALL AXIS(30.0,30.0,10HTIME(SEC)  ,-10,FIPLT2,0.0,T(II+3),
     1          T(II+4),10.0)
C
      YY=30.0
      ANN=60.0
      ANH=60.0/2.0
      AN= ANN
      CALL PLOT(29.0,YY,3)
      CALL PLOT(30.0,YY,2)
      CALL PLOT(30.0,YY+AN,2)
      CALL PLOT(29.0,YY+AN,2)
      CALL PLOT(30.0,YY+ANH,3)
      CALL PLOT(FIPLT2+30.0,YY+ANH,2)
      YY=YY+ANN
      CALL PLOT(30.0,YY,3)
C
C     .. Y-AXIS PLOT ..
C
      YY=30.0
      TK=ANN/PWMAX
      IF(P1WIDE(IJUN).EQ.0.0) GO TO 100
      PMAN= (P1MIN(IJUN)+P1WIDE(IJUN)/2.0) - (PWMAX/2.0)
      CALL  MEMMIN( PMAN, YY )
      PMAN= (P1MIN(IJUN)+P1WIDE(IJUN)/2.0) + (PWMAX/2.0)
      CALL  MEMMAX( PMAN, YY, ANN, ANH )
      GO TO 200
  100 CALL NUMBER(25.0,YY+ANH-1.0,2.0,P1WIDE(IJUN),0.0,-1)
  200 CALL PLOT(30.0,YY+ANH,3)
```

Appendix F (Continued)

```
C
C     .. CURVE PLOT ..
C
      IIKK=II-1
      DO 300 J=1,IIKK
      IF(P1WIDE(IJUN).EQ.0.0) GO TO 320
      WNN= ANN * P1WIDE(IJUN) / PWMAX
      IF( WNN.NE.ANN )        GO TO 340
      PN = 0.0
      GO TO 360
  340 PN= (ANN-WNN) / 2.0
      GO TO 360
  320 PN=ANH
  360 PLT1(IJUN,J)=ABS(PLT1(IJUN,J)-P1MIN(IJUN))
      PLY= TK * PLT1(IJUN,J)
      CALL PLOT(T(J+1)+30.0,YY+PN+PLY,2)
  300 CONTINUE
      YY=YY+ANN
      CALL PLOT(T(J+1)+30.0,YY+PN+PLY,3)
C
      CALL PLOT(400.0,0.0,-3)
      CALL PLOT(0.0,0.0,666)
      CALL PLOT(0.0,0.0,777)
      CALL PLOT(0.0,0.0,888)
      RETURN
      END


      SUBROUTINE UMMAX(VMAN,YY)
      CALL MLOG(VMAN,IMN)
      CALL RNUMB(12.0,YY,2.0,VMAN,0.0,2)
      CALL SYMBOL(15.0,YY,2.0,3H*10,0.0,3)
      CALL NUMBER(22.0,YY+1.0,1.5,FLOAT(IMN),0.0,-1)
      RETURN
      END


      SUBROUTINE UMMIN(VMAN,YY)
      CALL MLOG(VMAN,IMN)
      CALL RNUMB(12.0,YY,2.0,VMAN,0.0,2)
      CALL SYMBOL(15.0,YY,2.0,3H*10,0.0,3)
      CALL NUMBER(22.0,YY+1.0,1.5,FLOAT(IMN),0.0,-1)
      RETURN
      END


      SUBROUTINE  MEMMAX( PMAN, YY, ANN, ANH )
      CALL MLOG(PMAN,IMN)
      CALL RNUMB(12.0,YY+ANN-3.0,2.0,PMAN,0.0,2)
      CALL SYMBOL(15.0,YY+ANN-3.0,2.0,3H*10,0.0,3)
      CALL NUMBER(22.0,YY+ANN-1.5,1.5,FLOAT(IMN),0.0,-1)
      CALL PLOT(30.0,YY+ANH,3)
      RETURN
      END
```

Appendix F (Continued)

```
      SUBROUTINE  MEMMIN( PMAN, YY )
      CALL MLOG(PMAN,IMN)
      CALL RNUMB(12.0,YY,2.0,PMAN,0.0,2)
      CALL SYMBOL(15.0,YY,2.0,3H*10,0.0,3)
      CALL NUMBER(22.0,YY+1.0,1.5,FLOAT(IMN),0.0,-1)
      RETURN
      END


      SUBROUTINE RNUMB(X,Y,H,PMAN,ANG,NC)
      APP=ABS(PMAN)
      IF(APP.GE.1) GO TO 100
      IKK=2+NC
      GO TO 200
  100 IKK=INT(ALOG10(APP)+2.00001+NC)
  200 IF(PMAN.LT.0.0)  IKK=IKK+1
      DST=IKK*H*6./7.
      XN= X- DST*COS(ANG*0.01745)
      YN= Y- DST*SIN(ANG*0.01745)
      CALL NUMBER(XN,YN,H,PMAN,ANG,NC)
      RETURN
      END


      SUBROUTINE DTLIST(LU1,LU2)
      DIMENSION IA(20)
      REWIND LU1
      N = 1
      L = 51
   15 IF ( L .LE. 50 ) GO TO 30
   20 WRITE(LU2,2)
      WRITE(LU2,3) ( I, I=1,8 )
      WRITE(LU2,4)
      L = 1
   30 READ(LU1,1,END=50) ( IA(I),I=1,20 )
      WRITE(LU2,5)      N,( IA(I),I=1,20 ),N
      IF ( L .NE. 50 ) GO TO 40
      WRITE(LU2,6)
      WRITE(LU2,10) ( I, I=1,8 )
      WRITE(LU2,7)
   40 L = L + 1
      N = N + 1
      GO TO 15
   50 CONTINUE
      WRITE(LU2,6)
      WRITE(LU2,10) ( I, I=1,8 )
      WRITE(LU2,8)
      WRITE(LU2,9) LU1
      REWIND LU1
    1 FORMAT( 20A4 )
    2 FORMAT( 1H1, //, 50X, '     INPUT DATA ECHO ' / )
    3 FORMAT( 16X, 'DATA ', 4X, 8( 9X, I1 ) )
    4 FORMAT( 14X, 'SEQ. NO.', 3X, 8('----5----0'), / )
    5 FORMAT( 16X, I4, 5X, 20A4, 2X, I4 )
    6 FORMAT( 1H0, 24X, 8('----5----0') )
```

```
 7 FORMAT(1H0, 30X, '* * *  CONTINUE  * * *' )
 8 FORMAT(1H0, 30X, '* * *  INPUT DATA END  * * *' )
 9 FORMAT(1H0,30X,'* * *  INPUT DATA FROM FILE NO. =',I3,' * * *' )
10 FORMAT( 25X, 8(9X,I1) )
11 FORMAT(1H0)
   RETURN
   END
```