JAERI-M
86-140

MUTUAL: A COMPUTER CODE FOR ANALYSING NUCLEAR CRITICALITY SAFETY ON ARRAY SYSTEM

September 1986

Yoshitaka NAITO, Toshiyuki KANEKO* and Hiroshi OKUNO

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

JAERI-M レポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319－11 茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　山田軽印刷所

JAERI-M 86-140

MUTUAL : A Computer Code for Analysing Nuclear Criticality Safety on Array System

Yoshitaka NAITO, Toshiyuki KANEKO* and Hiroshi OKUNO

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

(Received September 9, 1986)

For evaluating critical safety on array system, the solid angle method is often used as a simple method. There are, however, some demerits in the method : the effective multiplication factor of the system cannot be known and it is excessively overestimated under certain circumstances. Therefore a new method has been developed to obtain the effective mutiplication factor of array system simply. In this method, neutron transportation being expressed in terms of solid angles between units arranged in the system, and effects of "shadow" caused by other units and effects of neutron reflection by the wall being taken into account, the eigen equation is solved and the effective mutiplication factor is obtained. A computer code MUTUAL has been programmed according to this method. This report is a user's manual for the MUTUAL code.

Keywords: Criticality Safety, Array, Effective Multiplication Factor, Solid Angle Method, Shadow, Reflection, Computer Code, MUTUAL

---

* Japan Information Service, Ltd.

配列体系の臨界安全解析コードMUTUAL

日本原子力研究所東海研究所燃料安全工学部
内藤　俶孝・金子 俊幸*・奥野　浩

（1986年9月9日受理）

配列体系の臨界安全性を評価するのに，簡便な方法として立体角法がしばしば用いられる。しかし，この方法には2つの欠点がある。それは，体系の実効増倍率が求められないこと，及びある状況の下では安全裕度が大きすぎるということである。このため，配列体系の実効増倍率を簡便に得る新しい方法を開発した。この方法では中性子の輸送は体系中に置かれたユニット間の立体角によって表現され，さらに，他のユニットによる「影」の効果及び壁による中性子の反射効果が考慮された上で，固有値方程式を解くことにより実効増倍率が得られる。計算コードMUTUALはこの方法に従ってプログラムされた。この報告書はMUTUALコードの使用手引書にもなっている。

東海研究所：〒319－11　茨城県那珂郡東海村白方字白根2－4
＊日本情報サービス（株）

CONTENTS

目　　次

1. Introduction

There are mainly two methods currently used to evaluate criticality safety on neutron interacting arrays of nuclear fissile materials, namely the solid angle method and the Monte Carlo method. The solid angle method developed by ORNL, as is well-known, has a form[1]:

$$\Omega_{tot} \leqq 9\text{-}10k_s$$

with the limitations of $k_s \leqq 0.80$ and $\Omega_{tot} \leqq 6.03$; the interaction among the fissile materials is limited according to the total solid angle Ω_{tot} subtended at the most reactive or the central unit and its single-unit neutron multiplication factor k_s. This method is semi-empirical : the safety curve was derived from the experimental data on highly enriched uranium units and from the simple assumption of half-reflection. The method was proved to have a considerable safety margin for well moderated fissile materials, whereas it is not generally applicable to under-moderated or unmoderated fissile materials.

In the Monte Carlo method (e.g. employed in the computer code KENO-IV[2]), neutron birth by nuclear fission as well as scattering and absorption by nuclides are traced immitating their physical processes. The results become as precise at least in principle as increase of the number of neutrons (the neutron history) considered in calculation. For practical usage, however, a lot of computer time is required for obtaining reliable results especially for a system of many units.

To avoid the overestimation of safety margins by solid angle method and too much time consuming by Monte Carlo method, we propose in this paper a method of solving a neutron balance equation among fissile units simply. The assumptions employed in the equation for simplicity are : (1) the neutron energy is only one-grouped, (2) the probability of neutron transfer from one unit to another is propotional to the solid angle subtended at the unit by the other, (3) neutron spatial distribution in each unit is not disturbed by the other units.

In Chapter 2, we show the basic equation expressing the neutron balance among fissile units. For practical usage the numerical procedure to calculate the solid angles and to solve the eigenvalue equation are made into a calculation program named MUTUAL. The logical flow of the program is summerized in Chapter 3. Limitation of the program as well as input and output data information are explained in Chapter 4. In the last Chapter we show sample inputs and outputs. In Appendix, a new solid angle method is proposed.

2. Theory

2.1 Derivation of Eigen Equation

The neutron balance equation of the array system which consists of nuclear fuel units is represented as follows. The neutron balance for unit i is expressed as,

$$\int_{vi} D_i \nabla^2 \phi_i dV - \int_{vi} \Sigma_{ai} \phi_i dV + 1/\lambda \int_{vi} \nu \Sigma_{fi} \phi_i dV + t_{1i} + \cdots + t_{ni} = 0 \qquad (1)$$

where $t_{ji} = -\int_{vj} D_j \nabla^2 \phi_j dV \cdot \int (e_i \cdot e_r) dA_i / 4\pi r^2$,

$\int_{vi} D_i \nabla^2 \phi_i dV$: neutron leakage from unit i,
$\int_{vi} \Sigma_{ai} \phi_i dV$: neutron absorption in unit i,
$\int_{vi} \nu \Sigma_{fi} \phi_i dV$: neutron production in unit i,
λ : the eigen-values (its maximum value corresponds to the effective multiplication factor of the system),
r : the distance between the center of unit j and the dA_i surface,
e_r : the unit vector directing from the center of unit j to the dA_i surface,
e_i : the unit vector perpendicular to the dA_i surface,
dA_i : a differential area on the surface of the unit i.

We derive a simple representation to get the maximum eigen value, that is the effective multiplication factor of the system, using following three approximations.

(i) Even if there were neutron interactions among units, the neutron flux distribution in each unit is assumed to be as the same as that of without interaction. Namely, neutron leakage from the unit in array system is approximated as follows with the buckling B_i^2 of an isolated unit i:

$$-\int_{vi} D_i \nabla^2 \phi_i dV_i = -\int_{Ai} (D_i \nabla \phi_i \cdot e_i) dA_i \simeq \int_{vi} D_i B_i^2 \phi_i dV_i \qquad (2)$$

(ii) Neutrons are assumed to be emitted isotropically from the center of the unit; the transportation probability t_{ji} from unit j to unit i is approximated as,

$$t_{ji} \simeq \int_{vj} D_j B_j^2 dV_j \Omega_{ji}/4\pi \tag{3}$$

where Ω_{ji} is the solid angle subtended at the center of unit j by unit i

(iii) All neutrons emitted from other units and reached a specific unit are assumed to be absorbed in it.

In the above three assumptions, the effective multiplication factor of the system is underestimated by the assumptions (i) and (ii), and is overestimated by the assumption (iii).

Under the above approximations, the neutron balance equation (1) can be rewritten as,

$$(-D+T+\frac{1}{\lambda}P)\phi=0 \tag{4}$$

where

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_n \end{bmatrix}, \quad T = \begin{bmatrix} T_{11} & \cdots \\ & T_{nn} \end{bmatrix}, \quad P = \begin{bmatrix} P_1 & 0 \\ 0 & P_n \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_n \end{bmatrix},$$

$d_i = \{(DB^2)_i + \Sigma_{ai}\}V_i$: neutron destruction operator for unit i,

$T_{ij} = (DB^2)_j V_j \Omega_{ji}/4\pi$: neutron transportation operator from unit j to unit i,

$P_i = \nu\Sigma_{fi}V_i$: neutron production operator for unit i,

ϕ_i : averaged neutron flux for unit i,

Ω_{ji} : solid angle subtended at the center of unit j by unit i,

λ : eigenvalues of the system

(The effective multiplication factor is its maximum value).

And matrices K and I are respectively defined as follows,

$$PD^{-1} \equiv K \equiv \begin{bmatrix} k_1 & & 0 \\ & \ddots & \\ 0 & & k_n \end{bmatrix}, \quad I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}, \tag{5}$$

where k_i is the effective multiplication factor of an isolated unit i.

The following equation is derived from Eq.(4).

$$(I-TD^{-1})(D\phi) = \frac{1}{\lambda}(PD^{-1})(D\phi) . \tag{6}$$

The above equation can be rewritten with the matrix K as,

$$K^{-1}(I-TD^{-1})(D\phi) = \frac{1}{\lambda}(D\phi) . \tag{7}$$

Equation (7) is the eigen equation to be solved. And element a_{ij} of matrix $A \equiv K^{-1}(I-TD^{-1})$ can be represented as follows using the relation $M^2=D/\Sigma_a$,

$$a_{ii} = 1/k_i(1-T_{ii}/d_i) = 1/k_i\{1-((M^2B^2)_i\Omega_{ii}/4\pi)/(1+(M^2B^2)_i)\}, \tag{8}$$

$$a_{ij} = -(1/k_i)(T_{ij}/d_j) = -(1/k_i)\{((M^2B^2)_j\Omega_{ji}/4\pi)/(1+(M^2B^2)_j\} \quad \text{for } j\neq i. \tag{9}$$

The multiplication factor of the array system is obtained by solving Eq.(7) with effective multiplication factor k_i of a single unit system i and neutron leakage $(M^2B^2)_j/(1+(M^2B^2)_j)$ from a unit j.

2.2 Calculation of Solid Angle

The solid angle by unit i subtended at the center of unit j is obtained by the following way. At first, imagine a unit sphere arround the center of unit j. Then unit i projects a shadow on the surface of the unit sphere. As an example, the calculation method for

cylindrical units is being explained. The diameter and height of cylindrical unit i is expressed as D and H, respectively. And a distance from the center of unit j to the surface of unit i is defined as L. Using a sphere coordinate whose origin is the center of unit j, the solid angle is obtained by integrating $\sin\theta d\theta d\phi$ over the area surrounded by a curve J in Fig.1.

$$J : \tan\theta = \{(2(L+D/2))/H\}\{\sin\phi \pm \sqrt{((D/2)/(L+D/2))^2-\cos^2\phi}\} \qquad (10)$$

Calculating a solid angle with this curve is disadvantageous with respect to computing time, because numerical integration must be performed. To reduce computing time, and to obtain an evaluation in safety-side, solid angle is assumed to be obtained by integrating $\sin\theta d\theta d\phi$ over the rectangular ABCD in Fig.1.

2.3 Calculation of Solid Angle in Case of Shadowed Area

The effect of "shadow" caused by other units must be considered in calculating a solid angle as shown in Fig.2. That is, in calculating the solid angle at unit j subtended by unit i if the unit k is between the two units, the unit k projects a "shadow" on the unit j, the corresponding solid angle is subtracted. Therefore, the solid angle corresponding to the area S≡EFGH shown in Fig.1 is subtracted as,

$$\Omega'_{ji} = \Omega_{ji} - \iint_S \sin\theta d\theta d\phi$$

2.4 Consideration of Neutron Reflection by Wall

If the specific unit is surrounded by the wall, the contribution of neutron reflection from the wall must be considered. In this case, imaginary units are added to the system in order to take into account the effect of neutron reflection. The transportation operator t'_{ji} from the imaginary unit j' of the unit j to the unit i is given as

$$t'_{ji} = (DB^2)_j V_j \phi_j \beta' \Omega'_{ji}/4\pi$$

where β' is albedo of the wall. If there are walls in many sides, the operator are added with respect to every direction. The total transportation operator t'_{ji} can be written as

$$t'_{ji} = t_{ji} + \underset{j}{\text{Sum}}\, t_{j'i}$$

$$= (DB^2)_j V_j \phi_j (\Omega_{ji} + \beta' \Omega_{j'i} + \beta'' \Omega_{j''i} + \cdots)/4\pi$$

Therefore in calculating t_{ji} in Eq.(7), the solid angle

$$\Omega'_{ji} = \Omega_{ji} + \beta' \Omega_{j'i} + \beta'' \Omega_{j''i} + \cdots$$

may be substituted for Ω_{ji} as shown in Fig. 3.

2.5 Reaction Rate

Various quantity of reaction rates which is, an optional output, is explained here after. As shown in Eq.(7), eigen vector has been defined as a quantity of neutron destruction, that is, leakage plus absorption. Therefore, reaction rates of unit i can be written as follows with element i of the eigen vector,

$$X_i = D_i \Phi_i = \{(DB^2)_i + \Sigma_{ai}\} V_i \phi_i \ . \tag{11}$$

$$A_i = \Sigma_{ai} V_i \phi_i = X_i / \{1 + (M^2 B^2)_i\} \ , \tag{12}$$

$$L_{out,i} = (DB^2)_i V_i \phi_i = A_i (M^2 B^2)_i \ , \tag{13}$$

$$L_{in,i} = \underset{j \neq i}{\text{Sum}} (\Omega_{ji} L_{out,j}) \ , \tag{14}$$

$$L_{net,i} = L_{out,i} - L_{in,i} \ , \tag{15}$$

$$F_i = k_{\infty,i} A_i / k_{eff} \ , \tag{16}$$

where A_i : absorption in unit i,

$L_{out,i}$: leakage from unit i,
$L_{in,i}$: leakage into unit i from all other units,
$L_{net,i}$: net leakage for unit i,
F_i : fission in unit i,
$k_{\infty i}$: infinite-medium multiplication factor of unit i,
k_{eff} : effective multiplication factor of the array system.

Absolute value of each quantity is normalized as,

$$\sum_i (A_i + L_{net,i}) = 1 \ .$$

3. Structure of Program

3.1 Logical Program Flow

3.1.1 Read Input Data

Input data is read by subroutine CRDIN. However the nuclear constant data and the array of BOXes[2] and SUPER BOXes[3] are read by CARDIN2 and CARDIN3 respectively. Subroutine GTYPE1 deals with geometrical data. Arrays of BOXes and SUPER BOXes are stored by subroutine UNITN1. All input data are echoed back.

3.1.2 Check, Edit and Print Input Data

Input data are edited and printed out by subroutine CRDPRT. The geometrical data of a unit are checked by subroutine BOXCHK. Then the data of cubic units and cylindrical units are respectively checked by subroutine CUBCHK and CYLCHK. The arrays of BOXes and SUPER BOXes are respectively checked by subroutine SBCHK and SBCHK2.

3.1.3 Calculation of Solid Angle

At first, direction vector is calculated from the center of a specific unit to other units. Subroutine LOCBX(Y)(Z) is used for two units in the same "SUPER BOX". The subroutine for two units not in the same SUPER BOX is LOCSBX(Y)(Z). The last letter in the name of subroutines, that is, X,Y OR Z means the element of the vector to be calculated. These two subroutines are controlled by subroutine LOCATX(Y)(Z). The last letter means as above mentioned. The center of the specific unit is assumed to be the origin of the sphere coordinate system to describe the location. The coordinate of a shadow on the unit sphere projected by the specific unit is calculated. Then subroutine CUBOID and CYLNDR are respectively used for a cuboid and a cylinder. The subroutine which calculates the solid angle from the coordinate as mentioned above is SOLID2 for a cylinder. That for a cuboid is SOLID3. If peripheral units are surrounded by a reflector, the location of the imaginary unit is calculated by subroutine LOCAT2. Subroutine SOLIDA controlles these subroutines and prints out the clalculated solid angle.

3.1.4 Calculation of Eigen Value and Eigen Vector

Subroutine EIGEN calculates the coefficient matrix for the eigen equation with the solid angle calculated in SOLIDA and the nuclear constants read in CRDIN2. Subroutine DBIEIG solves the eigen equation. In fact, its coefficient matrix is 2x2 matrix. Subroutine EJRBES is used for more than 3x3 matrix. It is one of subroutines included in EISPAC-J[4], a subprogram package for solving eigen value problem.

3.1.5 Edit Neutron Balance for Each Unit

Subroutine EDIT2 calculates reaction rate for each unit based on the eigen vector and prints it out.

3.2 Tree Structure of Program

Tree structure of MUTUAL is shown in Fig.4 .

3.3 Alphabetical Subroutine Summary

The symbol (F) after the subroutine name means function routine.

ANORM (F)

ANORM calculates a distance between two points.

BOXCHK

This subroutine judges whether a geometrical description of each unit is valid.

CRDIN

This subroutine reads the input data except for the nuclear constant data and the arrangement of units.

CRDIN2

CRDIN2 reads the nuclear constant data, that is, infinite multiplication factors (k_∞) and/or effective multiplication

factors (k_{eff}) of units.

CRDIN3

This subroutine reads the arrangement of "BOX"es and "SUPER BOX"es.

CRDPRT

This subroutine edits and prints input data, and controlles subroutines which check the data.

CUBCHK

CUBCHK checks geometrical data for cuboids.

CYLCHK (entry name of subroutine CUBOID)

This procedure checks geometrical data for cylinders.

CUBOID

This subroutine calculates coordinate prints of a shadow on unit sphere projected by a cuboid.

CYLNDR

This subroutine calculates coordinate points of a shadow on unit sphere projected by a cylinder.

DAYTIM

DAYTIM gets date and time with library routine DATE and TIME.

DBIEIG

This subroutine calculates eigen value and eigen vector whose coefficient matrix is 2x2 .

DOMEG1

This subroutine gets the overlapped part with respect to ϕ coordinate in two integral regions in calculating "shadow" between two units.

DOMEG2 (entry name DOMEG1)

This subroutine gets the overlapped part with respect to θ coordinate as subroutine DOMEG1.

DOMEG3

This subroutine calculates a "shadow" for a cuboid.

DTLIST

This subroutine echoes back the input data.

EDIT1

This subroutine evaluates criticality safety of the system with the method described in CEA-R-3114[5], and prints out the result.

EDIT2

This subroutine calculates and prints out a reaction rate with the eigen vector for each unit.

EIGEN

This subroutine calculates the coefficient matrix with solid angle between units and nuclear constant data, and calls subroutine DBIEIG or EJRBES to solve eigen equation.

GTYPE1

This subroutine reads geometrical data.

GTYPE2 (entry name of GTYPE1)

This procedure prints out the geometrical data read by subroutine GTYPE1.

HEADER

This subroutine prints out the problem title with date and time.

LOCATX

This subroutine controlles subroutines which calculate x-element of the location vector between two units.

LOCATY and LOCATY (entry name of LOCATX)

These procedures are the same as LOCATX ,however, calculates respectively y-element and z-element.

LOCAT2

This subroutine calculates a location vector of an imaginary unit for the specific unit, if the system is surrounded by wall.

LOCBX

This subroutine calculates x-element of location vector between two units in the same "SUPER BOX".

LOCBY and LOCBZ (entry name of LOCBX)

These procedures are the same as LOCBX ,however, calculate respectively y- and z-element.

LOCSBX

This subroutine calculates x-elements of location vectors between the center of the specific unit and those of the other units.

LOCSBY and LOCSBZ (entry name of LOCSBX)

These procedure are the same as LOCSBX ,however, calculate respectively y- and z-elements.

LSET

This is an assistant subroutine used in LOCATX(Y)(Z), LOCAT2, and LOCBX(Y)(Z).

NOTEQ (F)

This function is equivalent to the relational operator ".NE." (in FORTRAN) for single precision real type variables.

EQUAL (entry name of NOTEQ)

This function is equivalent to the relational operator ".EQ." (in FORTRAN) for single precision real type variables.

DNOTEQ (entry name of NOTEQ)

This function is equivalent to the relational operator ".EQ." (in FORTRAN) for double precision real type variables.

OMCYL1 (F)

This function calculates a solid angle with θ and ϕ coordinates.

OMCYL2 (F)

This function calculates the solid angle of overlapped part in two integral regions.

RCLEAR

This subroutine sets single precision real type array equal to a constant value.

DCLEAR (entry name of RCLEAR)

This procedure sets double precision real type array equal to a constant value.

ICLEAR (entry name of RCLEAR)

This procedure sets a integer type array equal to a constant value.

CCLEAR (entry name of RCLEAR)

This procedure sets character type array equal to a constant value.

LCLEAR (entry name of RCLEAR)

This procedure sets logical type array equal to a constant value.

SBCHK

This subroutine checks arrangement of "BOX"es in the specific "SUPER BOX"es.

SBCHK2

This subroutine checks arrangement of "SUPER BOX"es.

SOLID

This subroutine controls subroutines which are used to calculate solid angles.

SOLID2

This subroutine calculates a solid angle with coordinate of a shadow on unit sphere projected by a cylinder.

SOLID3

This subroutine calculates a solid angle with coordinate of a shadow on unit sphere projected by a cuboid.

SORT

This subroutine performs quick sorting.

UNITN1

This subroutine stores arrangement of "BOX"es and "SUPER BOX"es in memory.

UNITN2 (entry name of UNITN1)

This procedure calculates the unit type and its location with a serial number defined in the code.

Above subroutines are all user's routines. The following subroutines built in the system library are used in addition to user's routines.

DATE

This subroutine gets the running date.

TIME

This subroutine gets the running time.

EJRBES

This subroutine solves eigen equation ,however, the coefficient matrix must be greater than 2x2 matrix. This is one of subroutines

included in EISPACK-J[4], a subprogram package for solving eigenvalue problems, which has been developped in JAERI.

4. Program Usage

4.1 Limitation

4.1.1 Limitaiton of Available Geometrical Shape

Cylinder and cuboid are available as a geometrical shape. The array system which can be calculated is a system consisted of only cylinders, only cuboids and their combination.

4.1.2 Dimension of Array

Dimension of arrays in the code is determined by PARAMETER statement in MAIN routine. Therefore, the values in the statement must be changed, if the array size is short of capacity. The meaning of each parameter and its value given in the code are shown in Table 1.

4.2 Input Data Instruction

The input form for geometrical data of MUTUAL is similar to that of MULTI-KENO[3],an improved version of KENO-IV[2]. However, "free format input" is not adopted in MUTUAL. In geometrical definition of a system, KENO-IV is characterized by existence of geometry key words and box type desriptions. The geometry input for KENO-IV consists of geometry key words representing simple types of three-dimensional configurations. A geometry shape defines a region. These regions can be nested one outside another to construct a desired object. Each region must be completely enclosed by the next larger region. This procedure is used to describe box types, each of which may contain a different geometry configuration. The box types can then be stacked to form a three-dimensional array of units. When stacking box types, the adjacent faces of adjacent box types must be the same size. SUPER BOX type is an extension of BOX TYPE. A system is divided into any number of SUPER BOXes. A SUPER BOX is divided into many BOX TYPEs similarly. A geometry defined by BOX TYPEs is the same as that of KENO-IV. By introducing the SUPER BOX concept, the adjacent faces of adjacent box types are not necessary to be the same size, if they are cotained by

different SUPER BOX each other. Availability of the code is increased by this extension. Each SUPER BOX is enclosed by CELL BOUNDARY similar to the CORE BOUNDARY of KENO-IV. An array of the SUPER BOXes is enclosed by CORE BOUNDARY.

Card 1 Title Card (A80)

TITLE Contains title only. If first column is blank, editting reaction rate is not performed.

Card 2 Super Box Array Card (4I5)

1 SBOX Number of SUPER BOX types.

2 SBXMAX Number of units in the x direction of the super box array.

3 SBYMAX Number of units in the y direction of the super box array.

4 SBZMAX Number of units in the z direction of the super box array.

Card 3 Super Box Array Card (A9,6X,5I5)

1 CT Must be "SUPER BOX".

2 NSB Super box number (between 1 and SBOX)

3 NSBARY(1,NSB) Number of box types contained by SUPER BOX NSB.

4 NSBARY(2,NSB) Number of units in the x direction of the array made by box types.

5 NSBARY(3,NSB) Number of units in the y direction of the array made by box types.

6 NSBARY(4,NSB) Number of units in the z direction of the array made by box types.

Card 4 Box Type Card(1) (A8,7X,I5)

1 CT Must be "BOX TYPE".

2 NB BOX number

Card 5 Box Type Card(2) (A8,12X,6F10.0)

1 IGEOM(NB) IGEOM(NB) must be one of the following :
CUBE,CUBOID,CYLINDER,CORE BDY,CELL BDY.
CUBE has +X=+Y=+Z and -X=-Y=-Z. Note that the +X

dimension need not equal the -X dimension of the cube; i.e., the origin need not be at the center of the cube. CUBOID is a rectangular parallel piped and may be described anywhere relative to the orign. CYLINDER has its length described along the Z axis and its center line must lie on the Z axis.

2 GPARAM(1) Radius for cylinder, +x dimension for cube, cuboid

GPARAM(2) -x dimension for cube, cuboid, +z for cylinder, omit GPARM(2) for a sphere.

GPARAM(3) +y dimension for cuboid, -z for cylinder, omit GPARAM(3).

GPARAM(4) -y dimension for cuboid, omit GPARAM(4) for all other geometry types except CORE BDY.

GPARAM(5) +z dimension for cuboid, omit GPARAM(5) for all other geometry types except CORE BDY.

GPARAM(6) -z dimension for cuboid, omit GPARAM(6) for all other geometry types except CORE BDY and CELL BDY.

Card 6 Nuclear Constant Data (20X,3F10.0)

1 SQMK(1) M^2B^2 for the unit.

2 SQMK(2) Effective multiplication factor k_{eff} for the unit.

3 SQMK(3) Infinite multiplication factor k_∞ for the unit

Only two of SQMK must be inputted. If the box has no unit, this card is not necessary.

Card 7 (A8,12X,6F10.0)

1 IGEOM(NB) Must be "CUBOID"

2 GPARAM(1) +x dimension for a cuboid or cube.

3 GPARAM(2) -x dimension for a cuboid or cube.

4 GPARAM(3) +y dimension for a cuboid, zero for a cube.

5 GPARAM(4) -y dimension for a cuboid, zero for a cube.

6 GPARAM(5) +z dimension for a cuboid, zero for a cube.

7 GPARAM(6) -z dimension for a cuboid, zero for a cube.

If the box is a void, this card is not necessary.

NOTE Card 4 ∿ Card 7 must be repeated NSBARY(1,NSB) (Card3) times.

Card 8 (A8,12X,6F10.0)

1 IGEOM(NSB) Must be "CELL BDY".

2 GPARAM(1)

} same as Card 7

7 GPARAM(6)

Card 9 (A8,12X,6F10.0)

1 IGEOM(NSB) Must be "CUBOID".

2 GPARAM(1)

} same as Card 7

7 GPARAM(6)

NOTE Card 3 ∿ Card 9 must be repeated SBOX (Card2) times.

Card 10 (A8,12X,6F10.0)

1 IGEOM Must be "CORE BDY".

2 GPARAM(1)

} same as Card 7

7 GPARAM(6)

Card 11 (A8,12X,6F10.0)

1 IGEOM Must be "CUBOID".

2 GPARAM(1)

} same as Card 7

7 GPARAM(6)

Card 12 Mixed Box Orientation Card. (11I5)

Enter only if NSBARY(1,NSB) is greater than 1.

N The box type. N must be greater than zero and less than or equal to NSBARY(1,NSB) (parameter 3, card 3).

IX1 The starting point in the x direction. IX1 must be at least 1 and less than or equal to NSBARY(3,NSB) (parameter 4, card 3).

IX2 The ending point in the x direction. IX2 must be at least 1 and less than or equal to NSBARY(2,NSB).

INCX The number of boxes by which increments are made in the

positive x directin. INCX must be greater than zero and less than or equal to NSBARY(2,NSB).

IY1 The starting point in the y direction. IY1 must be at least 1 and less than or equal NSBARY(3,NSB) (parameter 5, card 3).

IY2 The ending point in the y direction. IY2 must be at least 1 and less than or equal to NSBARY(3,NSB).

INCY The number of boxes by which increments are made in the positive y direction. INCY must be greater than zero and less than or equal to NSBARY(3,NSB).

IZ1 The starting point in the z direction. IZ1 must be at least 1 and less than or equal to NSBARY(4,NSB) (parameter 6, card 3).

IZ2 The ending point in the z direction. IZ2 must be at least 1 and less than or equal to NSBARY(4,NSB).

INCZ The number of boxes by which increments are made in the positive z direction. INCZ must be greater than zero and less than or equal to NSBARY(4,NSB).

ISTP Indicates whether to read another set of mixed box orientation data.
=0, read another set of data.
≠0, do not read any more mixed box orientation data.

NOTE Repeat this card until box orientation of all super box have been described. This card must be entered in super box type sequence.

Card(s) 13 Mixed Super Box Orientation Card. Enter only if SBOX is greater than 1 (parameter 1, card 2).
This card must be entered in the same manner as card 12.

N The Super Box Type. N must be greater than zero and less than or equal to SBOX (parameter 1, card 2).

IX1 The starting point in the x direction. IX1 must be at least 1 and less than or equal to SBXMAX (parameter 2, card 2).

IX2 The ending point in the x direction. IX2 must be at least 1 and less than or equal to SBXMAX.

INCX The number of super box by which increments are made in the positive x direction. INCX must be greater than zero and less than or equal to SBXMAX.

IY1 The starting point in the y direction. IY1 must be at least 1 and less than or equal to SBYMAX (parameter 3, card 2).

IY2 The ending point in the y direction. IY2 must be at least 1 and less than or equal to SBYMAX.

INCY The number of super boxes by which increments are made in the positive y direction. INCY must be greater than zero and less than or equal to SBYMAX.

IZ1 The starting point in the z direction. IZ1 must be at least 1 and less than or equal to SBZMAX (parameter 4, card 2).

IZ2 The ending point in the z direction. IZ2 must be at least 1 and less than or equal to SBZMAX.

INCZ The number of super boxes by which increments are made in the positive z direction. INCZ must be greater than zero and less than or equal to SBZMAX.

ISTP Indicate whether to read another set of mixed super box orientation data.

= 0, read another set of data.

$\neq$ 0, do not read any more mixed super box orientation data.

NOTE The data of this card have the same feature as card 12. The last set of mixed super box orientation data must have a nonzero entry in the last field.

Card 14 Reflector Constants (6F10.0)

The value of the reflector constant is the specular albedo, that is, the fractional return for that face.

1 REFCST(1) Reflector constant for +x face of the array.

2 REFCST(2) Reflector constant for -x face of the array.

3 REFCST(3) Reflector constant for +y face of the array.
4 REFCST(4) Reflector constant for -y face of the array.
5 REFCST(5) Reflector consatnt for +z face of the array.
6 REFCST(6) Reflector constant for -z face of the array.

Card 15 END CASE. (A8)
This card is optional. It enable MUTUAL to read to the end of a case that contains an error and to start on a new case.

Card 16 END MUTUAL. (A10)
This card is optional and comes after the last card of the last case. No more data will be read after this card has been encountered.

4.3 Output Data Information

A unique number is defined for each unit in the system. The number of BOX and SUPER BOX in which the unit is located, and printed out. Calculated fractional solid angle of units subtended by each other is printed out using above number. Where, the fractional solid angle occupied by the solid angle for all space, i.e., 4π. The final result obtained is the effective multiplication factor (k_{eff}) of the system. If necessary, neutron balance for each unit, that is, absorption, fission, out-leakage,in-leakage and net leakage can be printed out. "Out-leakage" means a quantity of neutrons escaped from the specific unit. "In-leakage" means that of neutrons which reached the specific unit from the others.

5. Sample Input and Output

Input data for a sample problem is shown in Fig 5. This problem is a 3x3x2 array of identical cylinders. The result is shown in Fig. 6, and this JCL is shown in Fig. 7.

Acknowledgement

The authors wish to express thier thanks to Dr. Y. Nomura of JAERI for his valuable suggestions.

References

1) TID-7016 Rev.2, "Nuclear Safety Guide", USAEC (1978).

2) L.M.Petrie and N.F.Cross, "KENO-IV An Improved Monte Carlo Criticality Program", ORNL-4938, Oak Ridge National Laboratory (1975).

3) Yoshitaka NAITO, Masahiko YOKOTA and Koh NAKANO, "MULTI-KENO : A Monte Carlo Code for Criticality Safety Analysis", JAERI-M 83-049 (1983).

4) Toichiro FUJIMURA and Tsuneo TSUTSUI, "EISPACK-J:Subprogram Package for Solving Eigenvalue Problems", JAERI-M 8253 (1979).

5) CEA-R-3114, "Guide de Criticites", Centre d'Etudes Nucleaires de Saclay (1967).

5. Sample Input and Output

Input data for a sample problem is shown in Fig 5. This problem is a 3x3x2 array of identical cylinders. The result is shown in Fig. 6, and this JCL is shown in Fig. 7.

Acknowledgement

The authors wish to express thier thanks to Dr. Y. Nomura of JAERI for his valuable suggestions.

References

1) TID-7016 Rev.2, "Nuclear Safety Guide", USAEC (1978).

2) L.M.Petrie and N.F.Cross, "KENO-IV An Improved Monte Carlo Criticality Program", ORNL-4938, Oak Ridge National Laboratory (1975).

3) Yoshitaka NAITO, Masahiko YOKOTA and Koh NAKANO, "MULTI-KENO : A Monte Carlo Code for Criticality Safety Analysis", JAERI-M 83-049 (1983).

4) Toichiro FUJIMURA and Tsuneo TSUTSUI, "EISPACK-J:Subprogram Package for Solving Eigenvalue Problems", JAERI-M 8253 (1979).

5) CEA-R-3114, "Guide de Criticites", Centre d'Etudes Nucleaires de Saclay (1967).

5. Sample Input and Output

Input data for a sample problem is shown in Fig 5. This problem is a 3x3x2 array of identical cylinders. The result is shown in Fig. 6, and this JCL is shown in Fig. 7.

Acknowledgement

The authors wish to express thier thanks to Dr. Y. Nomura of JAERI for his valuable suggestions.

References

1) TID-7016 Rev.2, "Nuclear Safety Guide", USAEC (1978).

2) L.M.Petrie and N.F.Cross, "KENO-IV An Improved Monte Carlo Criticality Program", ORNL-4938, Oak Ridge National Laboratory (1975).

3) Yoshitaka NAITO, Masahiko YOKOTA and Koh NAKANO, "MULTI-KENO : A Monte Carlo Code for Criticality Safety Analysis", JAERI-M 83-049 (1983).

4) Toichiro FUJIMURA and Tsuneo TSUTSUI, "EISPACK-J:Subprogram Package for Solving Eigenvalue Problems", JAERI-M 8253 (1979).

5) CEA-R-3114, "Guide de Criticites", Centre d'Etudes Nucleaires de Saclay (1967).

Appendix A New Solid Angle Method

A solid angle method has been newly derived with the fundamental equation of MUTUAL for evaluating neutron interaction between array units. Derivation of the equation will be shown as follows.

As shown in Chap. 2 of this report, the eigen equation for array units is,

$$K^{-1}(I-TD^{-1})(D\phi) = 1/\lambda(D\phi) \tag{A.1}$$

The element a_{ij} of the matrix $A \equiv K^{-1}(I-TD^{-1})$ is represented as follows,

$$a_{ii} = 1/k_i\{1-((M^2B^2)_i\Omega_{ii})/4\pi)/(1+(M^2B^2)_j)\} , \tag{A.2}$$

$$a_{ij} = -(1/k_i)\{((M^2B^2)_j\Omega_{ji}/4\pi)/(1+(M^2B^2)_j)\} \tag{A.3}$$

The above Eqs. (A.1), (A.2) and (A.3) are the same as Eqs.(7),(8) and (9) of Chap.2, respectively.

Applying the theorem of Gerschgorin[1] to Eq.(A.1), following relation is obtained (see note).

$$\underset{j=1,n}{\text{Min}} \left(a_{ii} - \sum_{j=1,j\neq i}^{n} |a_{ij}|\right) \leq 1/\lambda \tag{A.4}$$

With Eqs.(A.2) and (A.3),the above relation is expressed as follows,

$$\underset{i=1,n}{\text{Min}} \{(1-\nu_i)/k_i\} \leq 1/\lambda , \tag{A.5}$$

where $\nu_i = \sum_{j=1}^{n} \{(M^2B^2)_j\}/\{1+(M^2B^2)_j\}\cdot\Omega_{ji} = \sum_{j=1}^{n} F_j\Omega_{ji} = \sum_{j=1}^{n} \nu_{ji}$,

$F_j = \{(M^2B^2)_j\}/\{1+(M^2B^2)_j\}$: neutron leakage probability from a unit j,

$\nu_{ji} = F_j \Omega_{ji}$: probability of neutron transfer from unit j to i,

$\nu_i = \sum_{j=1}^{n} \nu_{ji}$: probability of neutron transfer from all units to unit i.

With the maximum eigen value of λ, keff, the subciriticality condition of the array system is expressed as folows,

$$k_{eff} \leqq \underset{i=1,n}{\text{Max}} \{ k_i/(1-\nu_i) \} < k_{limit} , \tag{A.6}$$

where, k_{limit} is the subcritical limit of neutron multiplication factor and is chosen lower than 1. Relation (A.6) requires the following relation for any unit i,

$$k_i/(1-f_i\Omega_i) < k_{limit} , \tag{A.7}$$

where $\Omega_i = \sum_{j=1}^{n} \Omega_{ji}$,

$$f_i = \nu_i/\Omega_i = \left(\sum_{j=1, j\neq i}^{n} F_j \Omega_{ji} \right) \Big/ \left(\sum_{j=1, j\neq i}^{n} \Omega_{ji} \right)$$

$$= \left\{ 1/\left(\sum_{j=1, j\neq i}^{n} \Omega_{ji} \right) \right\} \sum_{j=1, j\neq i}^{n} (1-k_i/k_{\infty j})\Omega_{ji} .$$

Relation (A.7) is rewritten as,

$$\Omega_i \quad 1/f_i(1-k_i/k_{limit}) . \tag{A.8}$$

This is the relation we recommend here to evaluate nuclear criticality safety, and we name it JAERI's solid angle method.

In relation (A.8), if we set f_i and k_{limit} to be equal to 1, the relation is the same as that reported in CEA-R-3114, and if we set f_i and k_{limit} to be equal 0.66 and 0.95 respectively, the relation becomes the same as that shown in TID-7016 except the direction of solid angle. In TID-7016, Ω_i is obtained by summalizing the solid angles subtended at the center of unit i by other units, whereas the

solid angle Ω_i of (A.8) is obtained by summalizing the solid angles subtended by unit i at the centers of other units.

The limited solid angles Ω_i obtained by JAERI's, ORNL's and CEA's methods are compared in Fig. A.1 for the system of two identical units. In this Figure, k_{limit} of relation (A.8) is set equal to 0.94, and f is expressed by k_i and $k_{\infty i}$ with changing $k_{\infty i}$ parametrically. From this Figure, we conclude that JAERI's method is less over-conservative than the methods of TID-7016 and CEA-R-3114 .

〈 Note 〉 Gerschgorin's Theorem [1)]

Let $A = (a_{ij})$ be an arbitrary nxn complex matrix, and let

$$\Lambda_i \equiv \sum_{j\neq i}^{n} |a_{ij}|, \quad 1 \leqq i \leqq n$$

Then, all the eigenvalues λ of A lie in the union of the disks

$$| Z-a_{ii} | \leqq \Lambda_i , \quad 1 \leqq i \leqq n .$$

Proof : Let λ be any eigenvalue of the matrix A, and let x be an eigenvector of A corresponding to λ. We normalize the vector x so that its largest component in modulus is unity. By definition,

$$(\lambda-a_{ii})x_i = \sum_{j=1,j\neq i}^{n} a_{ij}x_j, \quad 1 \leqq i \leqq n.$$

In particular, if $||x_r|| = 1$, then

$$|\lambda-a_{rr}| \leqq \sum_{j=1,j\neq i}^{n} |a_{rj}| \cdot |x_j| \leqq \sum_{j=1,j\neq i}^{n} |a_{rj}| = \Lambda_r$$

Thus, the eigenvalue λ lies in the disk $|z-a_{rr}| \leqq \Lambda_r$. But since λ was an arbitrary eignvalue of A, it follows that all the eigenvalue of the matrix A lie in the *union* of the disks $|z-a_{ii}| \leqq \Lambda_i$, $1 \leqq i \leqq n$, completing the proof.

1) Gerschgorin, S. "Uber die Abrenz und der Eigenwerte einer Matrix", Izv. Aked. Nauk. SSSR Ser. Mat. 7, 749-754;16,22,25 (1931).

Table 1 Parameter in MAIN Routine

parameter	preset value	meaning
NBOXMX	10	Summation of box types for each super box.
NBXMX	10	Number of boxes in the x direction of the array made by box types.
NBYMX	10	Number of boxes in the y direction of the array made by box types.
NBZMX	10	Number of boxes in the z direction of the array made by box types.
NSBMX	5	Number of super box types.
NSBXMX	5	Number of super boxes in the x direction.
NSBYMX	5	Number of super boxes in the y direction.
NSBZMX	5	Number of super boxes in the z direction.
MAXU	100	Number of units in the system.
MAXU2	200	MAXU * 2
NSPBOX	25	Number of super boxes.

Using above value of the parameters, the required core strage size is about 780 K bytes.

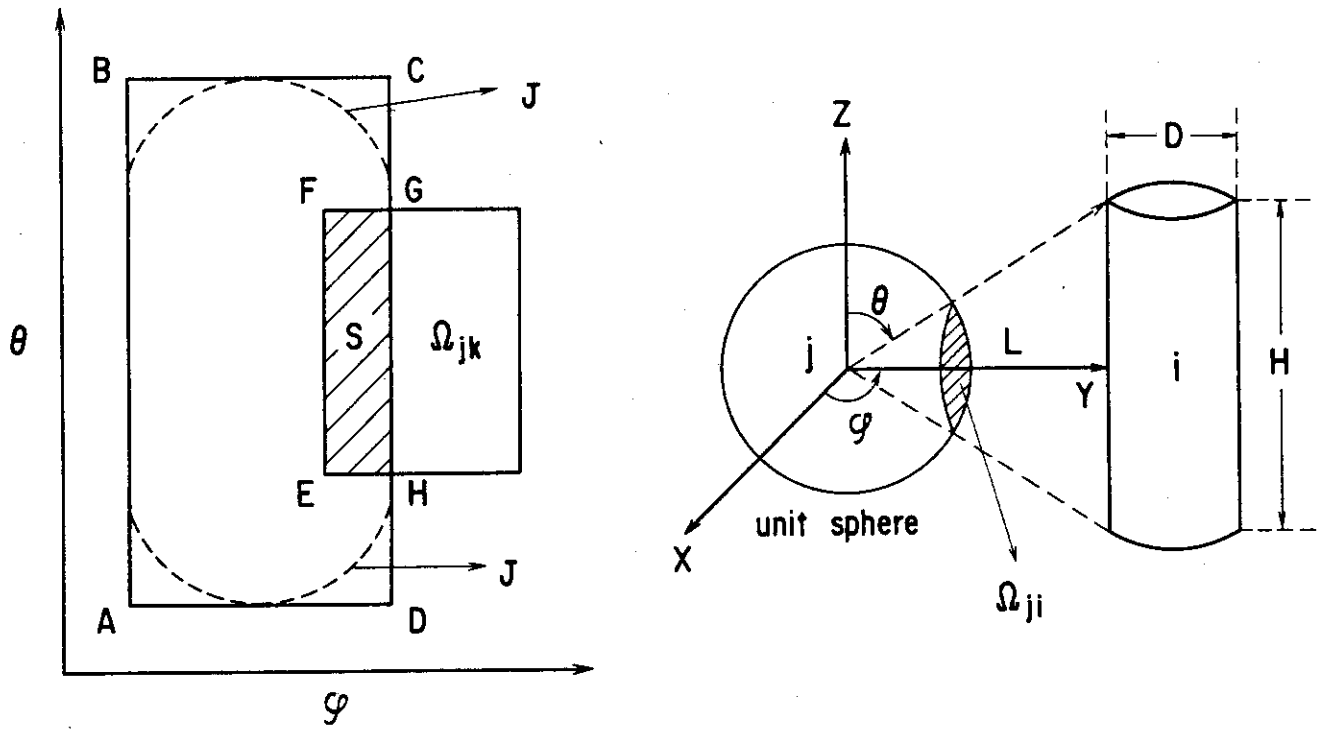


Fig. 1 Calculation Method of Solid Angle

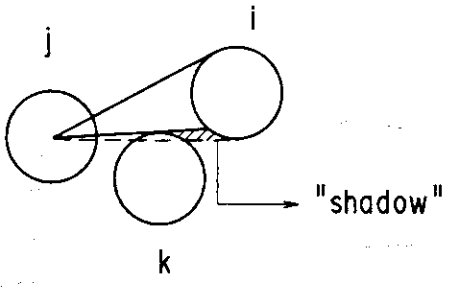


Fig. 2 Effect of "shadow"

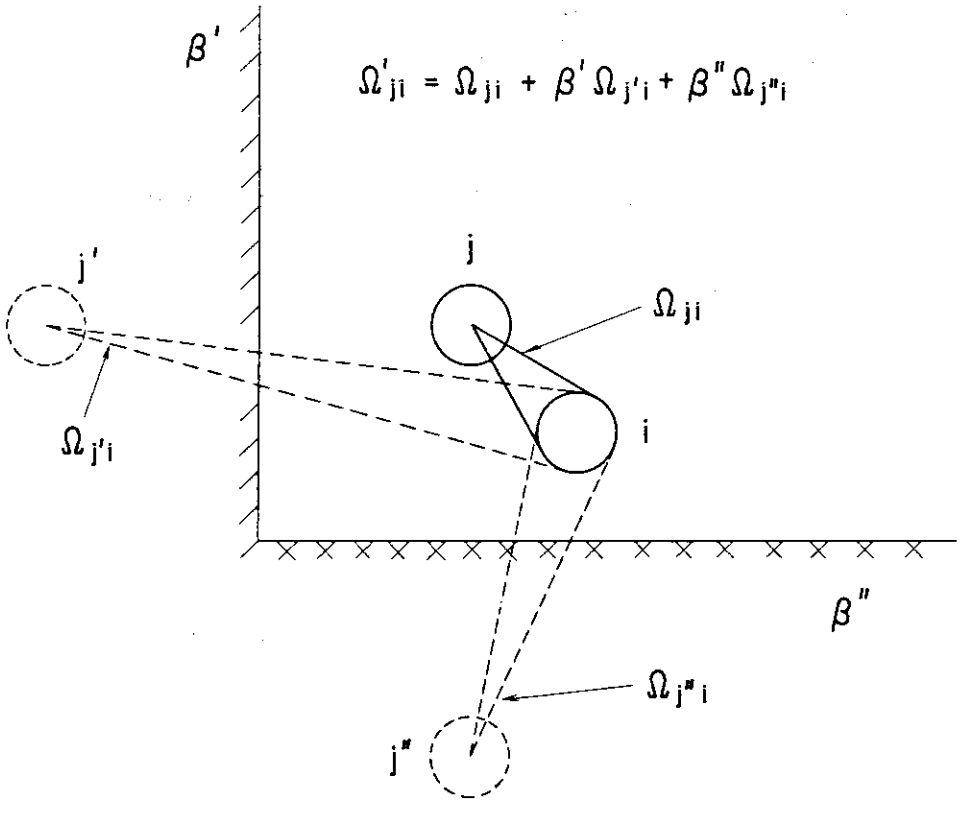
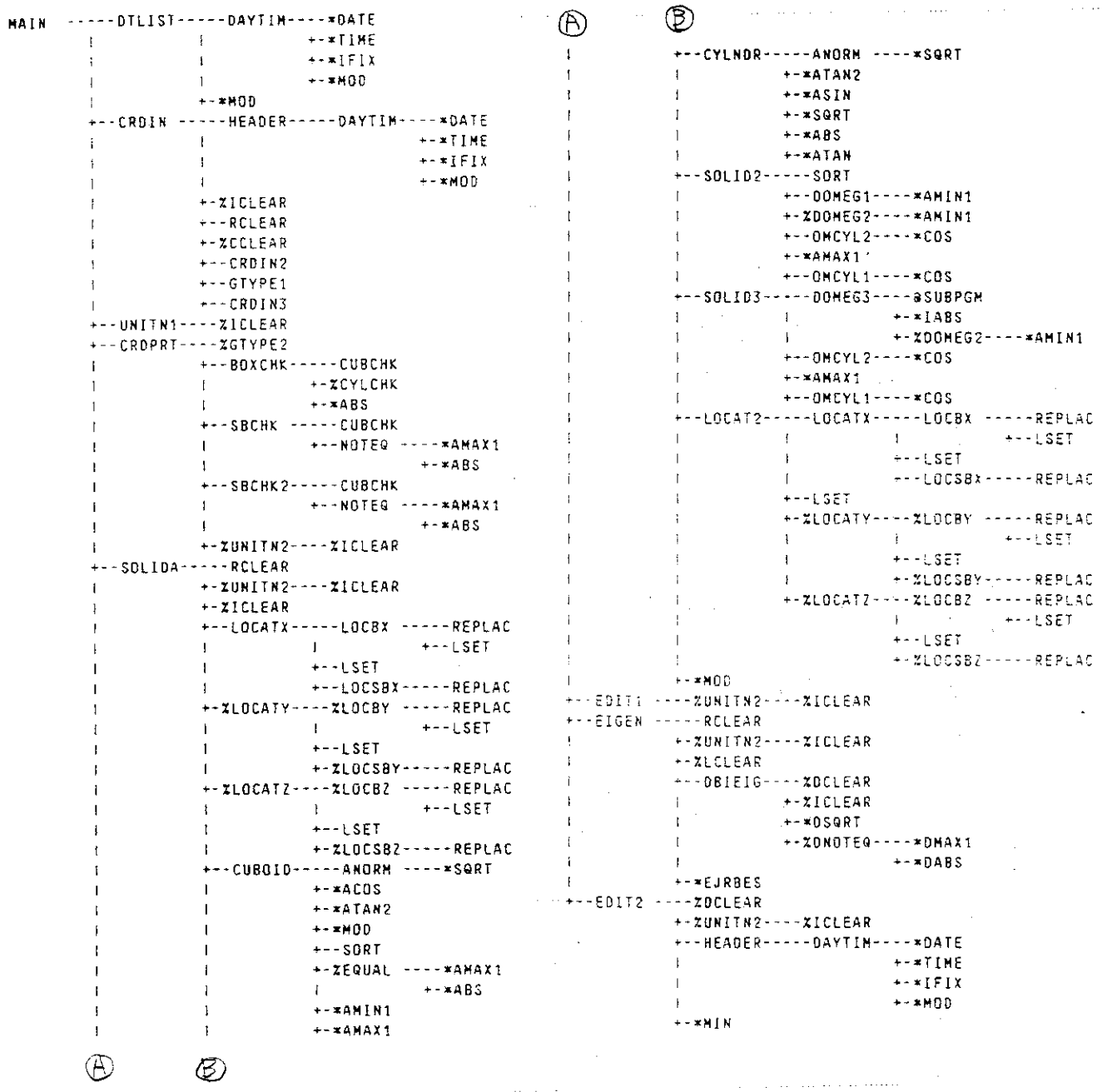


Fig. 3 Effect of Wall



Note * : library routine

% : entry name

@ : external procedure name

Fig. 4 Tree Structure of Program

```
                    **********************
                    *                    *     DSN = J9305.MUTUALV2.DATA(SAMPLE)
                    *   INPUT DATA LIST  *
                    *                    *     DATE  86/02/06  TIME  09:40:32
                    **********************

     . ....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8 .
   1 . **  CUBOID  3X3X2 (  50 CM )  2.0GU/L H/U=10  21.8 X 21.8 X 100  **              .     1
   2 .     1    1     1    1                                                            .     2
   3 . SUPER BOX           1    1    3    3    2                                       .     3
   4 . BOX TYPE            1                                                           .     4
   5 . CUBOID                 10.9      -10.9       10.9      -10.9       50.0       -50.0    .     5
   6 .                                  0.70878    1.469                                .     6
   7 . CUBOID                 35.9      -35.9       35.9      -35.9       75.0       -75.0    .     7
   8 . CELL BDY               107.7     -107.7      107.7     -107.7      150.0      -150.0   .     8
   9 . CUBOID                 107.7     -107.7      107.7     -107.7      150.0      -150.0   .     9
  10 . CORE BDY               107.7     -107.7      107.7     -107.7      150.0      -150.0   .    10
  11 . CUBOID                 107.7     -107.7      107.7     -107.7      150.0      -150.0   .    11
  12 .  0.0        0.0       0.0        0.0        0.0        0.0                          .    12
  13 . END MUTUAL                                                                        .    13
     . ....*....1....*....2....*....3....*....4....*....5....*....6....*....7....*....8 .

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

Fig. 5 Sample Input Data

```
   **  CUBOID  3X3X2 (  50 CM )  2.0GU/L H/U=10  21.8 X 21.8 X 100  **              DATE  86/02/06  TIME  09:40:33


NO. OF SUPER BOX TYPE           =  1
NO. OF SUPER BOX IN X-DIRECTION =  1
NO. OF SUPER BOX IN Y-DIRECTION =  1
NO. OF SUPER BOX IN Z-DIRECTION =  1

  SUPER BOX =  1

 BOX     GEOMETRICAL                                GEOMETRICAL                                                    UNIT
TYPE       TYPE                                     PARAMETER                                        M2*B2        KEFF        KINF

                          +X          -X          +Y          -Y          +Z          -Z
   1    CUBOID        1.0900E+01 -1.0900E+01  1.0900E+01 -1.0900E+01  5.0000E+01 -5.0000E+01  1.0726E+00  7.0878E-01  1.4690E+00

                          +X          -X          +Y          -Y          +Z          -Z
        CUBOID        3.5900E+01 -3.5900E+01  3.5900E+01 -3.5900E+01  7.5000E+01 -7.5000E+01

                          +X          -X          +Y          -Y          +Z          -Z
        CELL BDY      1.0770E+02 -1.0770E+02  1.0770E+02 -1.0770E+02  1.5000E+02 -1.5000E+02

BOX ARRAY
 Z =  1
    1    1    1
    1    1    1
    1    1    1
 Z =  2
    1    1    1
    1    1    1
    1    1    1

SUPER BOX ARRAY
 Z =  1
    1

                          +X          -X          +Y          -Y          +Z          -Z
        CORE BDY      1.0770E+02 -1.0770E+02  1.0770E+02 -1.0770E+02  1.5000E+02 -1.5000E+02


 UNIT    SUPER BOX    SUPER BOX       BOX        BOX
   NO      TYPE       LOCATION        TYPE     LOCATION
                      ( X, Y, Z)              ( X, Y, Z)

    1        1      ( 1, 1, 1)        1      ( 1, 1, 1)
    2        1      ( 1, 1, 1)        1      ( 2, 1, 1)
    3        1      ( 1, 1, 1)        1      ( 3, 1, 1)
    4        1      ( 1, 1, 1)        1      ( 1, 2, 1)
    5        1      ( 1, 1, 1)        1      ( 2, 2, 1)
    6        1      ( 1, 1, 1)        1      ( 3, 2, 1)
    7        1      ( 1, 1, 1)        1      ( 1, 3, 1)
    8        1      ( 1, 1, 1)        1      ( 2, 3, 1)
    9        1      ( 1, 1, 1)        1      ( 3, 3, 1)
```

Fig. 6 Sample Output Data

```
   10        1      ( 1, 1, 1)      1      ( 1, 1, 2)
   11        1      ( 1, 1, 1)      1      ( 2, 1, 2)
   12        1      ( 1, 1, 1)      1      ( 3, 1, 2)
   13        1      ( 1, 1, 1)      1      ( 1, 2, 2)
   14        1      ( 1, 1, 1)      1      ( 2, 2, 2)
   15        1      ( 1, 1, 1)      1      ( 3, 2, 2)
   16        1      ( 1, 1, 1)      1      ( 1, 3, 2)
   17        1      ( 1, 1, 1)      1      ( 2, 3, 2)
   18        1      ( 1, 1, 1)      1      ( 3, 3, 2)


 UNIT  NO
(BOX TYPE)

SUPER BOX LOCATION    ( 1, 1, 1)
SUPER BOX TYPE =  1

BOX LOCATION Z =  1

   1     2     3
( 1) ( 1) ( 1)

   4     5     6
( 1) ( 1) ( 1)

   7     8     9
( 1) ( 1) ( 1)

BOX LOCATION Z =  2

  10    11    12
( 1) ( 1) ( 1)

  13    14    15
( 1) ( 1) ( 1)

  16    17    18
( 1) ( 1) ( 1)


                    ALBEDO
   +X      -X      +Y      -Y      +Z      -Z
 0.0     0.0     0.0     0.0     0.0     0.0
```

Fig. 6 Sample Output Data (cont'd)

SOLID ANGLE FRACTION

FROM / UNIT	TO UNIT 1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17	18		
1	0.0	3.5435E-02	0.0	3.5435E-02	2.4078E-02	9.3760E-03	0.0	9.3760E-03	0.0	3.2685E-03
	5.2830E-03	1.8176E-03	5.2830E-03	6.4405E-03	4.5154E-03	1.8176E-03	4.5154E-03	1.7820E-03		
2	3.5435E-02	0.0	3.5435E-02	2.4078E-02	3.5435E-02	2.4078E-02	9.3760E-03	0.0	9.3760E-03	5.2830E-03
	3.2685E-03	5.2830E-03	6.4405E-03	5.2830E-03	6.4405E-03	4.5154E-03	1.8176E-03	4.5154E-03		
3	0.0	3.5435E-02	0.0	9.3760E-03	2.4078E-02	3.5435E-02	0.0	9.3760E-03	0.0	1.8176E-03
	5.2830E-03	3.2685E-03	4.5154E-03	6.4405E-03	5.2830E-03	1.7820E-03	4.5154E-03	1.8176E-03		
4	3.5435E-02	2.4078E-02	9.3760E-03	0.0	3.5435E-02	0.0	3.5435E-02	2.4078E-02	9.3760E-03	5.2830E-03
	6.4405E-03	4.5154E-03	3.2685E-03	5.2830E-03	1.8176E-03	5.2830E-03	6.4405E-03	4.5154E-03		
5	2.4078E-02	3.5435E-02	2.4078E-02	3.5435E-02	0.0	3.5435E-02	2.4078E-02	3.5435E-02	2.4078E-02	6.4405E-03
	5.2830E-03	6.4405E-03	5.2830E-03	3.2685E-03	5.2830E-03	6.4405E-03	5.2830E-03	6.4405E-03		
6	9.3760E-03	2.4078E-02	3.5435E-02	0.0	3.5435E-02	0.0	9.3760E-03	2.4078E-02	3.5435E-02	4.5154E-03
	6.4405E-03	5.2830E-03	1.8176E-03	5.2830E-03	3.2685E-03	4.5154E-03	6.4405E-03	5.2830E-03		
7	0.0	9.3760E-03	0.0	3.5435E-02	2.4078E-02	9.3760E-03	0.0	3.5435E-02	0.0	1.8176E-03
	4.5154E-03	1.7820E-03	5.2830E-03	6.4405E-03	4.5154E-03	3.2685E-03	5.2830E-03	1.8176E-03		
8	9.3760E-03	0.0	9.3760E-03	2.4078E-02	3.5435E-02	2.4078E-02	3.5435E-02	0.0	3.5435E-02	4.5154E-03
	1.8176E-03	4.5154E-03	6.4405E-03	5.2830E-03	6.4405E-03	5.2830E-03	3.2685E-03	5.2830E-03		
9	0.0	9.3760E-03	0.0	9.3760E-03	2.4078E-02	3.5435E-02	0.0	3.5435E-02	0.0	1.7820E-03
	4.5154E-03	1.8176E-03	4.5154E-03	6.4405E-03	5.2830E-03	1.8176E-03	5.2830E-03	3.2685E-03		
10	3.2685E-03	5.2830E-03	1.8176E-03	5.2830E-03	6.4405E-03	4.5154E-03	1.8176E-03	4.5154E-03	1.7820E-03	0.0
	3.5435E-02	0.0	3.5435E-02	2.4078E-02	9.3760E-03	0.0	9.3760E-03	0.0		
11	5.2830E-03	3.2685E-03	5.2830E-03	6.4405E-03	5.2830E-03	6.4405E-03	4.5154E-03	1.8176E-03	4.5154E-03	3.5435E-02
	0.0	3.5435E-02	2.4078E-02	3.5435E-02	2.4078E-02	9.3760E-03	0.0	9.3760E-03		
12	1.8176E-03	5.2830E-03	3.2685E-03	4.5154E-03	6.4405E-03	5.2830E-03	1.7820E-03	4.5154E-03	1.8176E-03	0.0
	3.5435E-02	0.0	9.3760E-03	2.4078E-02	3.5435E-02	0.0	9.3760E-03	0.0		
13	5.2830E-03	6.4405E-03	4.5154E-03	3.2685E-03	5.2830E-03	1.8176E-03	5.2830E-03	6.4405E-03	4.5154E-03	3.5435E-02
	2.4078E-02	9.3760E-03	0.0	3.5435E-02	0.0	3.5435E-02	2.4078E-02	9.3760E-03		
14	6.4405E-03	5.2830E-03	6.4405E-03	5.2830E-03	3.2685E-03	5.2830E-03	6.4405E-03	5.2830E-03	6.4405E-03	2.4078E-02
	3.5435E-02	2.4078E-02	3.5435E-02	0.0	3.5435E-02	2.4078E-02	3.5435E-02	2.4078E-02		
15	4.5154E-03	6.4405E-03	5.2830E-03	1.8176E-03	5.2830E-03	3.2685E-03	4.5154E-03	6.4405E-03	5.2830E-03	9.3760E-03
	2.4078E-02	3.5435E-02	0.0	3.5435E-02	0.0	9.3760E-03	2.4078E-02	3.5435E-02		
16	1.8176E-03	4.5154E-03	1.7820E-03	5.2830E-03	6.4405E-03	4.5154E-03	3.2685E-03	5.2830E-03	1.8176E-03	0.0
	9.3760E-03	0.0	3.5435E-02	2.4078E-02	9.3760E-03	0.0	3.5435E-02	0.0		
17	4.5154E-03	1.8176E-03	4.5154E-03	6.4405E-03	5.2830E-03	6.4405E-03	5.2830E-03	3.2685E-03	5.2830E-03	9.3760E-03
	0.0	9.3760E-03	2.4078E-02	3.5435E-02	2.4078E-02	3.5435E-02	0.0	3.5435E-02		
18	1.7820E-03	4.5154E-03	1.8176E-03	4.5154E-03	6.4405E-03	5.2830E-03	1.8176E-03	5.2830E-03	3.2685E-03	0.0
	9.3760E-03	0.0	9.3760E-03	2.4078E-02	3.5435E-02	0.0	3.5435E-02	0.0		
TOTAL	1.4842E-01	2.1606E-01	1.4842E-01	2.1606E-01	2.8821E-01	2.1606E-01	1.4842E-01	2.1606E-01	1.4842E-01	1.4842E-01
	2.1606E-01	1.4842E-01	2.1606E-01	2.8821E-01	2.1606E-01	1.4842E-01	2.1606E-01	1.4842E-01		

FIRST EVALUATION OF CRITICALITY

UNIT	KEFF	OMEGA	1-KEFF	CRITICALITY
1	7.0878E-01	1.4842E-01	2.9122E-01	SUBCRITICAL
2	7.0878E-01	2.1606E-01	2.9122E-01	SUBCRITICAL
3	7.0878E-01	1.4842E-01	2.9122E-01	SUBCRITICAL
4	7.0878E-01	2.1606E-01	2.9122E-01	SUBCRITICAL
5	7.0878E-01	2.8821E-01	2.9122E-01	SUBCRITICAL
6	7.0878E-01	2.1606E-01	2.9122E-01	SUBCRITICAL

Fig. 6 Sample Output Data (cont'd)

```
  7    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL
  8    7.0878E-01   2.1606E-01   2.9122E-01  SUBCRITICAL
  9    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL
 10    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL
 11    7.0878E-01   2.1606E-01   2.9122E-01  SUBCRITICAL
 12    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL
 13    7.0878E-01   2.1606E-01   2.9122E-01  SUBCRITICAL
 14    7.0878E-01   2.8821E-01   2.9122E-01  SUBCRITICAL
 15    7.0878E-01   2.1606E-01   2.9122E-01  SUBCRITICAL
 16    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL
 17    7.0878E-01   2.1606E-01   2.9122E-01  SUBCRITICAL
 18    7.0878E-01   1.4842E-01   2.9122E-01  SUBCRITICAL


 SECOND EVALUATION OF CRITICALITY

 KEFF OF SYSTEM   CRITICALITY

         0.7917   SUBCRITICAL
```



```
    ##  NEUTRON BALANCE  ##

        (UNIT NO.)
        ABSORPTION
        OUT LEAKAGE
        IN  LEAKAGE
        NET LEAKAGE
        FISSION


    SUPER BOX LOCATION   ( 1, 1, 1)
    SUPER BOX TYPE =  1


    BOX LOCATION Z =  1

BOX      / BOX
LOCATION / LOCATION
   Y     /    X
                1           2           3

             ( 1)        ( 2)        ( 3)
          2.4733E-02  3.2247E-02  2.4733E-02
          2.6528E-02  3.4587E-02  2.6528E-02
      1   5.3679E-03  6.9988E-03  5.3679E-03
          2.1160E-02  2.7588E-02  2.1160E-02
          4.5892E-02  5.9835E-02  4.5892E-02
```

Fig. 6 Sample Output Data (cont'd)

```
                ( 4)          ( 5)          ( 6)
              3.2247E-02    4.1547E-02    3.2247E-02
              3.4587E-02    4.4562E-02    3.4587E-02
        2     6.9988E-03    9.0172E-03    6.9988E-03
              2.7588E-02    3.5545E-02    2.7588E-02
              5.9835E-02    7.7091E-02    5.9835E-02

                ( 7)          ( 8)          ( 9)
              2.4733E-02    3.2247E-02    2.4733E-02
              2.6528E-02    3.4587E-02    2.6528E-02
        3     5.3679E-03    6.9988E-03    5.3679E-03
              2.1160E-02    2.7588E-02    2.1160E-02
              4.5892E-02    5.9835E-02    4.5892E-02


     BOX LOCATION Z =  2

BOX       / BOX
LOCATION  / LOCATION
   Y      /    X
                  1             2             3

                ( 10)         ( 11)         ( 12)
              2.4733E-02    3.2247E-02    2.4733E-02
              2.6528E-02    3.4587E-02    2.6528E-02
        1     5.3679E-03    6.9988E-03    5.3679E-03
              2.1160E-02    2.7588E-02    2.1160E-02
              4.5892E-02    5.9835E-02    4.5892E-02

                ( 13)         ( 14)         ( 15)
              3.2247E-02    4.1547E-02    3.2247E-02


              3.4587E-02    4.4562E-02    3.4587E-02
        2     6.9988E-03    9.0172E-03    6.9988E-03
              2.7588E-02    3.5545E-02    2.7588E-02
              5.9835E-02    7.7091E-02    5.9835E-02

                ( 16)         ( 17)         ( 18)
              2.4733E-02    3.2247E-02    2.4733E-02
              2.6528E-02    3.4587E-02    2.6528E-02
        3     5.3679E-03    6.9988E-03    5.3679E-03
              2.1160E-02    2.7588E-02    2.1160E-02
              4.5892E-02    5.9835E-02    4.5892E-02


     TOTAL ABSORPTION      5.3893D-01
     TOTAL NET LEAKAGE     4.6107D-01
     TOTAL FISSION         1.0000D+00
```

Fig. 6 Sample Output Data (cont'd)

```
***
//KYON1 EXEC FORT77,
// SO='J9305.MUTUALV2',
// A='ELM(*)'
***
//KYON2 EXEC LKED77
***
//KYON3 EXEC GO,
// OBSIZE=137
//FT05F001  DD  DSN=J9305.MUTUALV2.DATA(SAMPLE),DISP=SHR,LABEL=(,,,IN)
***
//
```

Fig. 7 Sample JCL

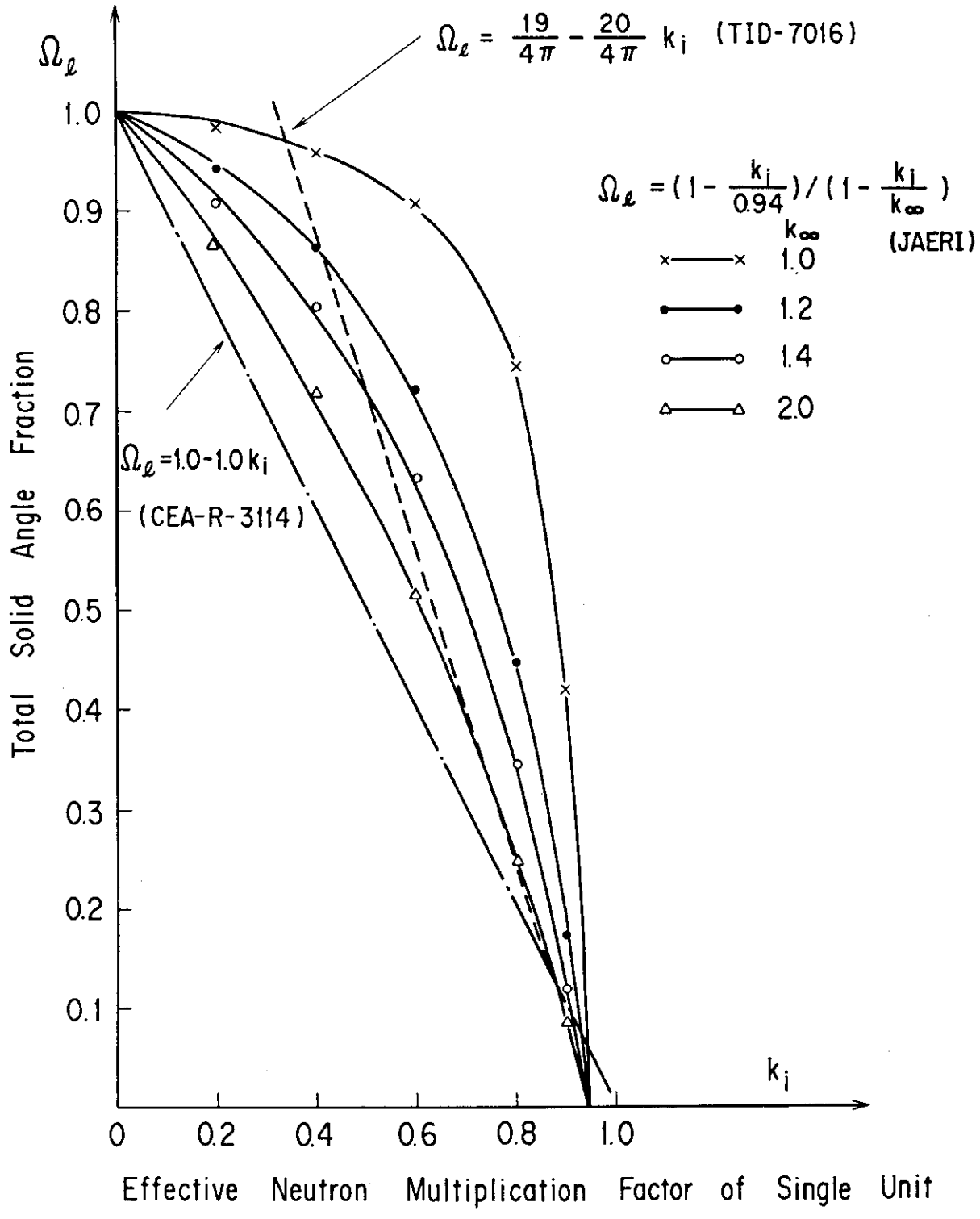


Fig.A.1 Various Limited Solid Angles for the System Consisted of Two Identical Units