The DC_N-S_N Hybrid Method for Two-Dimensional Neutron Transport Computation in R-Z Geometry

Yoichi Watanabe and Charles W. Maynard

October 1984

UWFDM-595

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

The DC$_N$-S$_N$ Hybrid Method for Two-Dimensional Neutron Transport Computation in R-Z Geometry

Yoichi Watanabe and Charles W. Maynard

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

October 1984

UWFDM-595

THE DC_N-S_N HYBRID METHOD FOR TWO-DIMENSIONAL NEUTRON TRANSPORT COMPUTATION IN R-Z GEOMETRY

Yoichi Watanabe and Charles W. Maynard

Fusion Technology Institute
Nuclear Engineering Department
University of Wisconsin-Madison
Madison, Wisconsin 53706

October 1984

UWFDM-595

TABLE OF CONTENTS

1. INTRODUCTION

In a previous paper,[1] we developed the theory of the discrete cones method (the DC_N method) for two-dimensional neutron transport calculations. The DC_N method was applied to a solution in a void and the S_N method was used for a solution in non-void regions. This hybrid method was formulated only for X-Y geometry. Numerical experiments demonstrate the DC_N-S_N hybrid method significantly mitigates the anomalous oscillations in a scalar flux distribution. Since an extension of the method to a solution in curved geometry is a necessary step from a practical point of view, in the present paper we formulate the hybrid method in R-Z geometry.

Although in the previous report we formulated two schemes of the DC_N method in a void, we choose one of those, in which a void is partitioned into mesh cells, and outgoing cone fluxes, incoming cone fluxes, and a source of a mesh cell are related by means of transfer and escape matrices. The present method is similar to the streaming matrix hybrid method (SMHM) by Clark.[2,3] However, there are the following major differences between the two methods:

1. In the SMHM, the incoming and outgoing fluxes of subsurfaces of boundary surfaces of a void are related through streaming matrices; meanwhile, in the DC_N-S_N hybrid method the relations are obtained for a mesh cell.
2. In the SMHM, the matrix elements are calculated numerically; meanwhile, in the DC_N-S_N hybrid method analytical formulas of the elements are derived. In these formulas single integrals that cannot be analytically integrated are evaluated by using low order Gaussian quadratures.

Consequently, the DC_N-S_N hybrid method has the following advantages over the SMHM:

A. For the same problem, the DC_N-S_N hybrid method requires less computer memory.

B. Computation of the matrix elements in the DC_N-S_N method is much faster than in the SMHM.

C. The fluxes in void mesh cells are calculated by the DC_N-S_N method.

After this introduction, in Sections 2, 3 and 4 the DC_N method will be formulated. In Section 5, a computer program utilizing the DC_N-S_N hybrid method will be described. In Section 6, sample calculations will be demonstrated. In Section 7, the difficulty of duct streaming problems will be discussed as well as the limited capability of the present hybrid method for such problems. Section 8 will conclude this report.

2. CONSTRUCTION OF THE DISCRETE CONES METHOD

In this section we shall solve the neutron transport equation of a void in R-Z geometry by the discrete cones method. By applying the standard discrete ordinates method to non-void domains, a hybrid method will be constructed.

The transport equation is given by Eq. (2.22) in Ref. 1:

$$\Psi(\underline{r},\underline{\Omega}) = \Psi(\underline{r}-s_o\underline{\Omega},\underline{\Omega}) + \int_0^{s_o} Q(\underline{r}-s\underline{\Omega},\underline{\Omega})\, ds \qquad (1)$$

where s is the distance between two spatial points, and Q is a particle source. To simplify the method, we ignore the source term for awhile. Then Eq. (1) becomes

$$\Psi(\underline{r},\underline{\Omega}) = \Psi(\underline{r}',\underline{\Omega}) \qquad (2)$$

where $$\underline{r}' = \underline{r} - s\underline{\Omega} .$$

As we observed in Section 3 of Ref. 1, the discrete cones method for a void cell, which is created by partitioning a large void, is more efficient than the discrete cones method for the large void itself in terms of the computer memory and the computing time. Hence we shall construct the discrete cones method for a void cell.

Defining $\underline{n}$ as a unit vector normal to a cell boundary surface, we multiply Eq. (2) by $\underline{\Omega}\cdot\underline{n}$ and integrate it over both a cone and an outgoing surface to find relations between the outgoing and incoming cone fluxes. The same procedure was applied in Ref. 1; however, there is a fundamental difference between the discrete cones method in X-Y geometry and one in R-Z geometry. The difference is that as a particle streams over mesh cells, the cone in which the particle lies varies in the R-Z coordinate system. Consequently, a particle on an outgoing surface of a mesh cell may cross the incoming surface of the mesh cell in a different cone. Hence we must consider the following recursive relation:

$$\psi_{km} = \sum_{m'} \sum_{k'} T_{kk'}^{mm'} \psi_{k'm'} \tag{3}$$

where $T_{kk'}^{mm'}$ is an element of the transfer matrix. In Eq. (3), k and k' denote the outgoing surface k and the incoming surface k', respectively; m and m' denote the cone m on the surface k and the cone m' on the surface k', respectively.

To find explicit formulas of the transfer matrix elements, we first look into a cylindrical annular mesh cell in the coordinate system illustrated in

Figs. 1 and 2. In Fig. 2(b) we refer to the top, bottom, outer, and inner boundary surfaces of the cell as TOP, BOTTOM, OUTER, and INNER, respectively. Since the system is symmetrical about the plane made of the unit vectors $\hat{e}_r$ and $\hat{e}_z$, and the X-Y plane, we need to consider two of eight octants of the unit sphere of directions. One of the octants satisfies $\mu > 0$, $\eta > 0$, and $\xi > 0$; the other satisfies $\mu < 0$, $\eta > 0$, and $\xi > 0$. Here μ and ξ are defined as follows: $\mu = \sqrt{1 - \eta^2} \cos \omega$ and $\xi = \sqrt{1 - \eta^2} \sin \omega$. Consequently, the transfer matrices of the following ten cases are sufficient for a numerical solution by the present method:

1. The transfer matrix for a particle streaming from INNER to OUTER: $\mu > 0$.
2. The transfer matrix for a particle streaming from OUTER to OUTER: $\mu > 0$.
3. The transfer matrix for a particle streaming from BOTTOM to OUTER: $\mu > 0$.
4. The transfer matrix for a particle streaming from INNER to TOP: $\mu > 0$.
5. The transfer matrix for a particle streaming from OUTER to TOP: $\mu > 0$.
6. The transfer matrix for a particle streaming from BOTTOM to TOP: $\mu > 0$.
7. The transfer matrix for a particle streaming from OUTER to INNER: $\mu < 0$.
8. The transfer matrix for a particle streaming from BOTTOM to INNER: $\mu < 0$.
9. The transfer matrix for a particle streaming from BOTTOM to TOP: $\mu < 0$.
10. The transfer matrix for a particle streaming from OUTER to TOP: $\mu < 0$.

A difference between cases 5 and 10, or cases 6 and 9 is that for cases 5 and 6 the polar angle ω of direction on the outgoing surface varies between 0 and $\pi/2$; on the other hand, for cases 9 and 10 it varies between $\pi/2$ and π. These separations are made because in an inner iteration of a numerical computation the spatial sweep over mesh cells is made separately for the directions of positive μ and negative μ; i.e., the directions

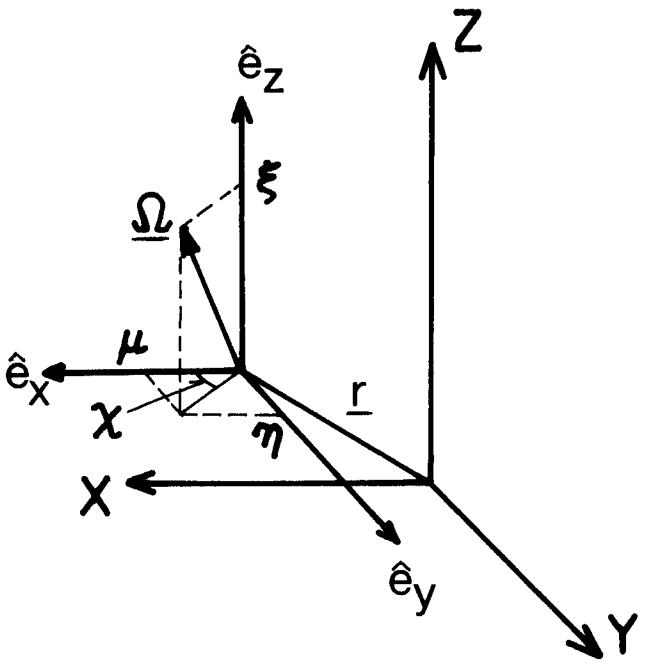


Fig. 1 The coordinate system for R-Z geometry.

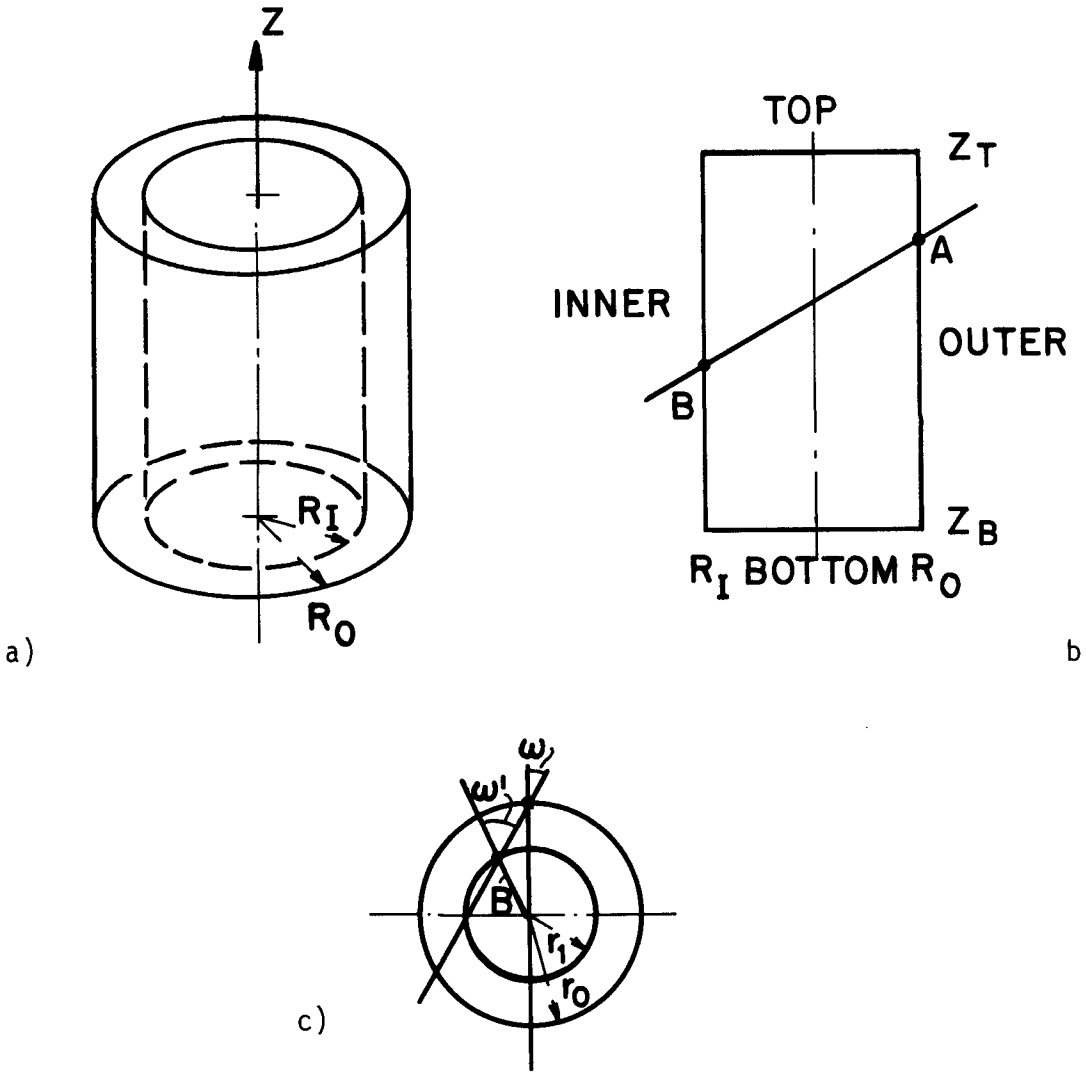


Fig. 2 (a) A cylindrical annular mesh cell.
(b) A cross sectional view of a mesh cell.
(c) A top view of a mesh cell.

with the angle ω varying between 0 and $\pi/2$ are treated separately from the directions with the angle ω varying between $\pi/2$ and π.

3. CALCULATIONS OF ELEMENTS OF THE TRANSFER MATRICES

In this section we shall describe a method to find the formulas of the transfer matrix elements for the ten cases given in Section 2. After a general description of the method, it will be explained in detail by using a specific example. Then a more systematic way suitable to programming will be discussed.

According to the definition of the transfer matrices given by Eq. (3), an element can be represented as follows:

$$T_{kk'}^{mm'} = \frac{\int_{A_{kk'}} dA_{k'} \iint_{\Delta\Omega_{mm'}} d\Omega\ \underline{\Omega}\cdot\underline{n}_{k'}}{\int_{A_k} dA_k \iint_{\Delta\Omega_m} d\Omega\ \underline{\Omega}\cdot\underline{n}_k} \tag{4}$$

where $A_{kk'}$ indicates a subdomain of the surface k' which a characteristic line subtended from the surface k crosses and $\Delta\Omega_{mm'}$ is a domain common to both cone m on the surface k and cone m' on the surface k'.

To perform the integration of the numerator in Eq. (4), we must find the integral limits, $A_{kk'}$ and $\Delta\Omega_{mm'}$. First, remember we are considering a streaming particle which leaves a cell through surface k in the cone m at a point $\underline{r}_A = (r,z)$ in the direction $\underline{\Omega} = (\eta,\omega)$. At the same time the particle enters the cell through surface k' in the cone m' at a point $\underline{r}_B = (r',z')$ in the direction $\underline{\Omega}' = (\eta',\omega')$. The two coordinates (r,z,η,ω) and (r',z',η',ω') in the four-dimensional phase space are related to each other by the following equations:

$$\eta = \eta' \tag{5a}$$

$$r \sin \omega = r' \sin \omega' \tag{5b}$$

$$\gamma(z - z') = r \cos \omega - r' \cos \omega' \tag{5c}$$

where $\gamma = \sqrt{1 - \eta^2}/\eta$. These equations are derived by geometrical considerations as well as more rigorous methods as discussed in Appendix A.

There is only one free spatial variable to specify a surface of a cylindrical annular cell, and we indicate the variable as p for the point A and q for the point B. Consequently, the problem we must solve is to find a domain where q, γ, and ω' vary when p, γ, and ω vary in a domain: $[p_-, p_+] \times [\gamma_-,\gamma_+] \times [\omega_-,\omega_+]$. Here $p_+ = r_O$ or z_T, $p_- = r_I$ or z_B, $\gamma_\pm = \sqrt{1 - \eta^2_{m\pm1/2}}/\eta_{m\pm1/2}$, and $\omega_\pm = \omega_{m\pm1/2}$. Since Eq. (5a) holds, from now on we do not distinguish γ' from γ and use γ.

Before going on, let us represent the numerator of Eq. (4) as follows:

$$N = \int_{q_1}^{q_2} \int_{\gamma_1}^{\gamma_2} f(q)\, dq \frac{d\gamma}{(1 + \gamma^2)^2} \int_{\omega'_-}^{\omega'_+} d\omega'\, \underline{\Omega} \cdot \underline{n}_k , \tag{6}$$

where q_1 and q_2 are functions of γ and ω', γ_1 and γ_2 are functions of q and ω', and

$$f(q)\, dq = \begin{cases} 2\pi r'\, dr' & \text{for TOP or BOTTOM} \\ 2\pi r_O\, dz' & \text{for OUTER} \\ 2\pi r_I\, dz' & \text{for INNER .} \end{cases}$$

The procedure for the integrations in Eq. (6) consists of the following four steps.

Step I: By substituting Eq. (5b) into Eq. (5c) to eliminate ω and solving for γ, we have

$$\gamma = \frac{\sqrt{r^2 - r'^2 \sin^2\omega'} - r' \cos \omega'}{z - z'} . \tag{7}$$

Also Eq. (5b) gives

$$r' = r \frac{\sin \omega}{\sin \omega'} . \tag{8}$$

Since either r or z and either r' or z' are fixed on a surface, γ may be represented as $\gamma = g(p,q,\omega')$. Hence varying p between p_- and p_+ gives the maximum and minimum values of γ as functions of q and ω'. If r is fixed, varying ω in Eq. (8) yields the range of r'. If r varies, varying both ω and r yields the range of r'.

Finally, taking account of $\gamma_- \leq \gamma \leq \gamma_+$ and $q_- \leq q \leq q_+$ where $q_+ = r_O$ or z_T and $q_- = r_I$ or z_B, we find integral limits in the q-γ plane.

All the integral limits are illustrated in Appendix B.1, where $\gamma = \gamma_\pm$ lines are not shown. Appendix B.2 shows the functions that represent the integral limits.

Step II: By using Eqs. (5b) and (5c), ω' can be represented as a function of p, q, γ, and ω. Hence the maximum and minimum values of ω' can be obtained for given ranges of these variables. We define the range of ω' by $[\omega'_{min}, \omega'_{max}]$. Since the range may cover the ω-range of several cones, it can be partitioned as follows:

$$[\omega'_{min}, \omega'_{max}] = \sum_{\ell=1}^{L} [\omega_{\ell-1}, \omega_\ell] \tag{9}$$

where $\omega_0 = \omega'_{min}$, $\omega_L = \omega'_{max}$, and ω_ℓ is either $\omega_{m+1/2}$ or $\omega_{m-1/2}$ for $\ell \neq$ either 0 or L.

Step III: Let us represent the result of an analytical integration with respect to q and γ in Eq. (6) by a function $G(\omega')$. Then

$$N = \int_{\omega'_-}^{\omega'_+} G(\omega')\, W(\omega')\, d\omega' \tag{10}$$

where $W(\omega')$ is either cos ω' or 1, and ω'_- and ω'_+ are one of ω_ℓs given in Eq. (9).

To perform the analytical integration with respect to ω' in Eq. (10), it is convenient to divide the range $[\omega'_-,\omega'_+]$ into subdomains so that the analytical expression of $G(\omega')$ is identical in each subdomain. Then Eq. (10) is represented as

$$N = \sum_{k=1}^{K} \int_{\omega_{k-1}}^{\omega_k} G_k(\omega')\, W(\omega')\, d\omega' = \sum_{k=1}^{K} \left[F_k(\omega')\right]_{\omega_{k-1}}^{\omega_k} \tag{11}$$

where $\omega_0 = \omega'_-$, $\omega_K = \omega'_+$, $G_k(\omega')$ is a function corresponding to a range $[\omega_{k-1},\omega_k]$, and $F_k(\omega')$ is a primitive function of the integration.

Step IV: Integrations in Eq. (11) are carried out analytically or numerically. Finally, an element of the transfer matrix is obtained by Eq. (4).

The above procedure will be discussed in more detail by using Case 1: streaming from INNER to OUTER. Since $r = r_o$ and $r' = r_I$, Eqs. (7) and (8) become

$$\gamma = \frac{\sqrt{r_o^2 - r_I^2 \sin^2\omega'} - r_I \cos \omega'}{z - z'} \tag{12}$$

and $$r_I = r_o \frac{\sin \omega}{\sin \omega'} . \tag{13}$$

Since $z' \leq z$ and $z_B \leq z \leq z_T$, by Eq. (12) the range of γ is

$$\frac{\sqrt{r_o^2 - r_I^2 \sin^2\omega'} - r_I \cos \omega'}{z_T - z'} \leq \gamma < \infty . \tag{14}$$

The range of z' is $$z_B \leq z' \leq z_T . \tag{15}$$

The domain made by these inequalities is illustrated in Fig. 3.

From Eq. (13) the range of ω' is given for $\omega_- \leq \omega \leq \omega_+$ as follows:

$$\arcsin \left(\frac{r_o}{r_I} \sin \omega_-\right) \leq \omega' \leq \arcsin\left(\frac{r_o}{r_I} \sin \omega_+\right) . \tag{16}$$

Here it is noted that if $\omega_+ > \arcsin (r_I/r_o)$, the right-most term is replaced by $\pi/2$. Similarly, if $\omega_- > \arcsin (r_I/r_o)$, the range cannot exist; in other words, the transfer matrix element for this case is zero. Hence, we assume $\omega_- < \arcsin (r_I/r_o)$.

After partitioning the range of ω' given by inequalities (16) into ω-ranges of cones as given in Eq. (9), we have Eq. (10), where $W(\omega') = \cos \omega'$. To find Eq. (11), first see Fig. 3. Let us represent an intersection of a surface $\gamma = f(z',\omega')$ and a plane $z' = z_B$ as $\gamma_1(\omega')$. Then a domain made by limiting the domain shown in Fig. 3 by two planes $\gamma = \gamma_+$ and $\gamma = \gamma_-$ is categorized into three types, two of which are illustrated in Figs. 4(a). For type 1, $\gamma_+ < \gamma_1(\omega') < \gamma_-$. For type 2, $\gamma_1(\omega') < \gamma_+ < \gamma_-$. For type 3, the area is zero.

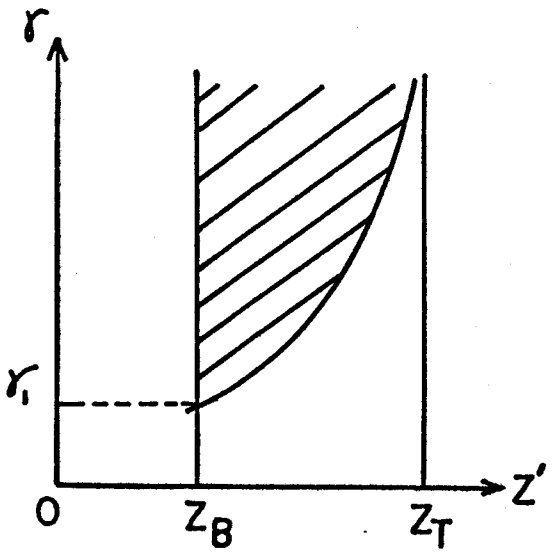
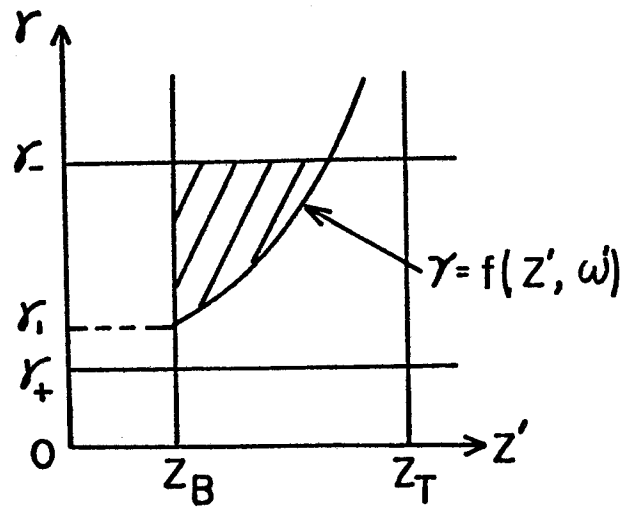


Fig. 3 The integral limit in the z' - γ plane for Case 1: INNER to OUTER. $\gamma = \gamma_{\pm}$ lines are not shown.

Fig. 4 (a) z' - γ plane area of integration.

(i) Type 1: $\gamma_+ < \gamma_1(\omega') < \gamma_-$



(ii) Type 2: $\gamma_1(\omega') < \gamma_+ < \gamma_-$

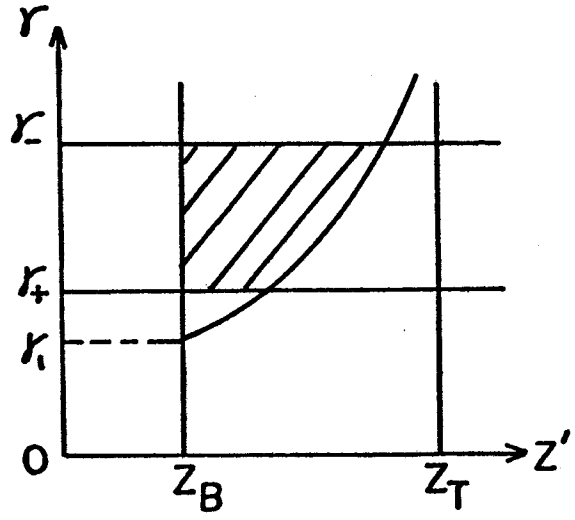
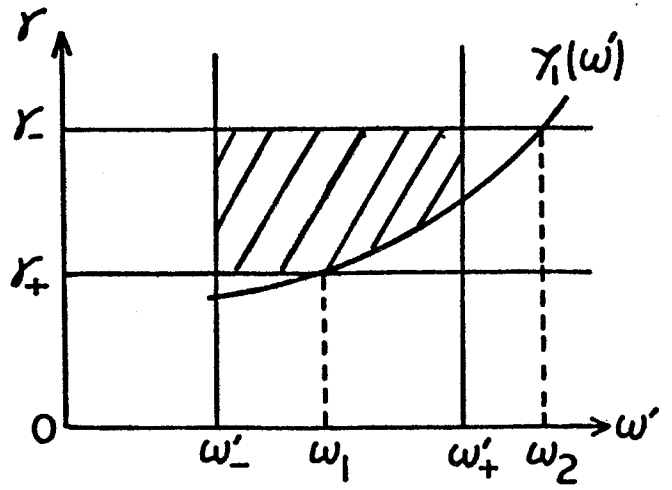


Fig. 4 (b) ω' - γ plane area of integration; $z' = z_B$.



(i) $$\gamma_1(\omega') = \frac{\sqrt{r_o^2 - (r_1 \sin\omega')^2} - r_I \cos\omega'}{\Delta z}$$

(ii) $$f(z', \omega') = \frac{\sqrt{r_o^2 - (r_I \sin\omega')^2} - r_I \cos\omega'}{z_T - z'}$$

By taking into account these three types of integral limits, Eq. (11) is obtained separately for the following six cases:

1. If $\gamma_- < \gamma_1(\omega'_-)$, $N = 0$.
2. If $\gamma_+ < \gamma_1(\omega'_-) < \gamma_- < \gamma_1(\omega'_+)$, z'-γ plane area of integration is given by Type 1 for $\omega'_- < \omega' < \omega_2$. Here ω_2 is a root of the equation $\gamma_- = \gamma(\omega')$. The area is zero for $\omega_2 < \omega' < \omega'_+$.
3. If $\gamma_+ < \gamma(\omega'_-) < \gamma(\omega'_+) < \gamma_-$, the area is Type 1 for $\omega'_- < \omega' < \omega'_+$.
4. If $\gamma_1(\omega'_-) < \gamma_+ < \gamma_- < \gamma_1(\omega'_+)$, the area is Type 2 for $\omega'_- < \omega' < \omega_2$, Type 1 for $\omega_2 < \omega' < \omega_1$, and zero for $\omega_1 < \omega' < \omega'_+$. Here ω_1 is a root of the equation $\gamma_+ = \gamma_1(\omega')$.
5. If $\gamma_1(\omega'_-) < \gamma_+ < \gamma_1(\omega'_+) < \gamma_-$, the area is Type 2 for $\omega'_- < \omega' < \omega_1$ and Type 1 for $\omega_1 < \omega' < \omega'_+$.
6. If $\gamma_1(\omega'_+) < \gamma_+$, the area is Type 2 for $\omega'_- < \omega' < \omega'_+$.

For instance, for Case 4, Eq. (11) becomes

$$N = \int_{\omega'_-}^{\omega_2} G_1(\omega')\cos\omega'\, d\omega' + \int_{\omega_2}^{\omega_1} G_2(\omega')\cos\omega'\, d\omega' + \int_{\omega_1}^{\omega'_+} G_3(\omega')\cos\omega'\, d\omega' \quad (17)$$

where

$$G_1(\omega') = 2\pi r_I \int_{\gamma_+}^{\gamma_-}\int_{z_B}^{z_1} \frac{\gamma^2}{(1+\gamma^2)^2}\, d\gamma\, dz' ,$$

$$G_2(\omega') = 2\pi r_I \int_{\gamma_1}^{\gamma_-}\int_{z_B}^{z_1} \frac{\gamma^2}{(1+\gamma^2)^2}\, d\gamma\, dz' ,$$

and

$$G_3(\omega') = 0 .$$

In the above definitions of G_1 and G_2, z_1 is given by

$$z_1(\gamma,\omega') = z_T - (\sqrt{r_o^2 - r_I^2 \sin^2\omega'} - r_I \cos\omega')/\Delta z \ .$$

After elaborate calculations, we find the exact analytical formula:

$$N = 2\pi r_I [F_1(\omega')]_{\omega_-'}^{\omega_2} + 2\pi r_I [F_2(\omega')]_{\omega_2}^{\omega_1} \ . \tag{18}$$

$F_1(\omega')$ and $F_2(\omega')$ are given as follows:

$$F_1(\omega) = \frac{1}{2} r_I \Delta z \ [\eta\sqrt{1-\eta^2} + \arcsin\eta]_{\eta_-}^{\eta_+} \sin\omega + \frac{1}{4} r_I^2(\eta_+^2 - \eta_-^2)$$

$$\times \{\omega' + \frac{1}{2}\sin 2\omega - \sin\omega \sqrt{(\frac{r_o}{r_I})^2 - \sin^2\omega} - (\frac{r_o}{r_I})^2 \tag{19}$$

$$\times \arcsin(\frac{r_I}{r_o}\sin\omega)\}$$

$$F_2(\omega) = \frac{1}{2} r_I \Delta z[(\arccos\eta_- - \eta_-\sqrt{1-\eta_-^2})\sin\omega - \sin\omega \arctan(y_1(\omega))$$

$$- \frac{1}{4r_I\Delta z}\{q^2 \arcsin((BB + \frac{2*CC}{y_1^2(\omega)}) * DD) \tag{20}$$

$$- \Delta z^2 \arcsin((2*AA*y_1^2(\omega) + BB) * DD) - \sqrt{CK} \arcsin((BB -$$

$$2*AA - \frac{2*CK}{y_1^2(\omega) + 1}) * DD)\}] - \frac{1}{4} r_I^2 \eta_-^2 \{\omega + \frac{1}{2}\sin 2\omega -$$

$$\sin\omega \sqrt{(\frac{r_o}{r_I})^2 - \sin^2\omega} - (\frac{r_o}{r_I})^2 \arcsin(\frac{r_I}{r_o}\sin\omega)\}$$

where $$y_1(\omega) = (\sqrt{r_o^2 - r_I^2 \sin^2\omega} - r_I \cos \omega)/\Delta z$$

$$q^2 = r_o^2 - r_I^2$$

$$AA = -\Delta z^4$$

$$BB = 2(r_o^2 + r_I^2)\ \Delta z^2$$

$$CC = -(r_o^2 + r_I^2)^2$$

$$DD = 1/(BB^2 - 4*AA*CC)$$

$$CK = BB - AA - CC\ .$$

In the above example, to obtain the explicit formula of Eq. (11) we must consider six cases, which are found by carefully examining Figs. 3. Since a computer cannot do such a visual examination, a more mechanical method to do the same work must be devised to write a workable computer program.

To accomplish this goal, we shall introduce graphs associated with tables and functions. In the remainder of this section, we shall explain the graphs and tables, and describe how to read these graphs and tables to find the final expression of Eq. (11) using Case 1 of streaming (INNER to OUTER) and Case 3 of streaming (BOTTOM to OUTER).

A graph is made up of vertices and arcs. As an example, see Graph 2.2 in

Table 1(b). Here the numbers are called vertex, and the arrows between the vertices are called arc. (See Ref. 4 about the terminology.) Each vertex is associated with a primitive function of integration, $F_k(\omega')$ in Eq. (11). The function corresponding to a vertex is found by looking into a table associated with the graph. Each arc is associated with a value of ω', which is ω_k in Eq. (11).

The integration in Eq. (11) is carried out from ω'_- to ω'_+ with respect to ω' in the following manner:

1. Find the vertex number corresponding to $\omega' = \omega'_-$, using an associated table. The vertex is called the starting point.
2. Do the same thing for $\omega' = \omega'_+$. The vertex is called the end point.
3. Find a path between the starting point and the end point by tracing the arcs in the direction of the arrow. Note that if there are more than two arcs leaving a vertex, we choose an arc whose ω is the smallest. At a vertex the lower limit of integration, ω_{k-1} in Eq. (11), is set to ω of the arc reaching the vertex, and the upper limit of integration, ω_k in Eq. (11), is set to ω of the arc leaving the vertex. A primitive function of integration, $F_k(\omega')$, is also chosen.
4. At the end point, we have an expression for Eq. (11).

To be more specific, let us explain Case 1 of streaming (INNER to OUTER). First, look at Graph 1.2 in Table 1(a). Here the first column shows the graph number, the second column shows an identification number that indicates the direction of arrows, and the fourth column shows the table numbers associated with this graph. For the present example, we must examine Table 1.1. (All the graphs and tables are given in Appendix C.1 and C.2, respectively.)

Table 1(a)

(1.1)

CASE	y_1		FUNCTION
1	+-		0
2	+	-	2
3		+-	1

Graph 1.2

GRAPH ID.	IDCRT	GRAPH		TABLE NUMBER THAT CORRESPONDS TO THIS GRAPH	
1-2	2	← 1	2	1.1	6A.2
			↑	3A.1	7.1
			|	3A.2	8.2
			3	4.1	9.4
				4.2	10.4

Table 1(b)

(3A.1)

CASE	y_1		FUNCTION
1	+-		1
2	+	-	2
3		+-	3

(3A.4)

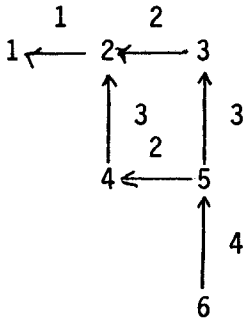
CASE	y_3	y_1		FUNCTION
1	+-			0
2	+	+-		6
3	+		-	11
4		+-		8
5		+	-	12
6			+-	13

1) $y_1 = r_o \sin(\omega' - \omega_-)/\sin\omega'$

$y_3 = r_o \sin(\omega' - \omega_+)/\sin\omega'$

2) $\pm = \gamma_\pm$

Graph 2.2



Now see Table 1(a) (1.1). In this table, y_1 is a function given by $y_1(\omega') = (\sqrt{r_o^2 - r_I^2 \sin^2\omega'} - r_I \cos\omega')/\Delta z$. + and - denote γ_+ and γ_-, respectively. There are three cases depending on the order of values γ_+, γ_-, and $y_1(\omega')$. Each case is accompanied by a function, $F_i(\omega')$, which is a primitive function of integration. The function number, i, is given in the right-most column of the table. The case number shown in the left-most column of the table corresponds to the vertex number in Graph 1.2.

Let us represent the values of ω' as ω_1 for an arc between the vertices 1 and 2, and ω_2 for an arc between the vertices 2 and 3. For the present example, ω_1 is a root of an equation $\gamma_+ = y_1(\omega)$ and ω_2 is a root of an equation $\gamma_- = y_1(\omega)$.

Suppose that $y_1(\omega'_-) \leq \gamma_+ \leq \gamma_- \leq y_1(\omega'_+)$. Then, we see by Table 1(a) (1.1) that the starting point is the vertex 3 and the end point is the vertex 1. According to the rules described above, we find the following expression of Eq. (11):

$$N = [F_1(\omega')]_{\omega'_-}^{\omega_2} + [F_2(\omega')]_{\omega_2}^{\omega_1} + [0]_{\omega_1}^{\omega'_+} .$$

This is identical to Eq. (18) except a factor $2\pi r_I$ which is included in the functions F_1 and F_2.

Next we consider Case 3 of streaming (BOTTOM to OUTER). In this case, ω' can be larger than $\pi/2$. Hence we consider two cases, $0 \leq \omega' \leq \pi/2$ (Case 3A) and $\pi/2 \leq \omega' \leq \pi$ (Case 3B), separately. For the present example, we assume $0 \leq \omega' \leq \pi/2$.

Case 3A (BOTTOM to OUTER, $0 \leq \omega' \leq \pi/2$) is made up of four tables and four graphs instead of only one table and one graph for Case 1 (INNER to

OUTER). Hence, at first we must find which tables and graphs should be looked into. This is done by examining Table 2. In the table the number, n, over each subsection on the ω' axis indicates a corresponding table number, 3A.n. The definitions of $\hat{\omega}_1$ and $\hat{\omega}_2$ are as follows:

$$\hat{\omega}_1 = \begin{cases} \arcsin\left(\frac{r_o}{r_I}\sin\omega_-\right) & \text{if } \sin\omega_- < r_I/r_o \\ \pi/2 & \text{otherwise} \end{cases}$$

$$\hat{\omega}_2 = \begin{cases} \arcsin\left(\frac{r_o}{r_I}\sin\omega_+\right) & \text{if } \sin\omega_+ < r_I/r_o \\ \pi/2 & \text{otherwise.} \end{cases}$$

Now suppose that $\omega_- < \omega'_- < \omega_+ < \omega'_+ < \hat{\omega}_1$. Then, examining Table 2.3A.ii, we find that we must read Table 3A.1 for $\omega'_- < \omega' < \omega_+$ and Table 3A.4 for $\omega_+ < \omega' < \omega'_+$. Further assume $\gamma_- > y_1(\omega'_+) > \gamma_+$ and $\gamma_+ > y_1(\omega'_-)$. Here $y_1(\omega) = r_o \sin(\omega - \omega_-)/\sin\omega$. Then referring to both Table 3A.1 (in Table 1(b)) and Graph 1.2 gives

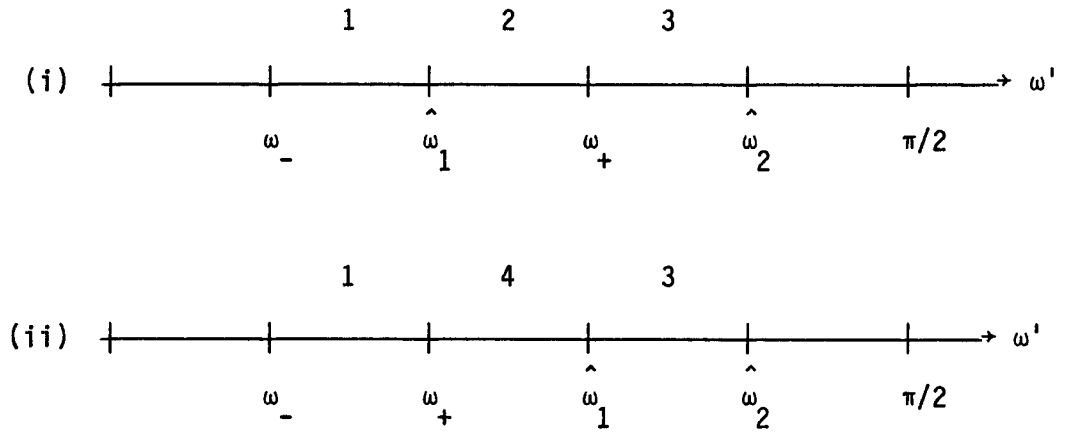
$$N_1 = [F_3(\omega')]_{\omega'_-}^{\omega_2} + [F_2(\omega')]_{\omega_2}^{\omega_+} \tag{21}$$

where ω_2 is a root of an equation $\gamma_- = y_1(\omega)$.

Next assume that $y_3(\omega'_+) < \gamma_+ < y_1(\omega'_+) < \gamma_-$ and $\gamma_+ < y_3(\omega'_+) < \gamma_- < y_1(\omega'_+)$. Here $y_3(\omega) = r_o \sin(\omega - \omega_+)/\sin\omega$. In Table 3A.4 (in Table 1(b)), $\omega' = \omega'_+$ is Case 5 and $\omega' = \omega'_+$ is Case 2. As seen in Graph 2.2, there are two possible paths between vertices 5 and 2. Which path should be chosen is

Table 2

(3A.)



1) $\hat{\omega}_1 = \arcsin\left(\frac{r_o}{r_I} \sin \omega_-\right)$

$\hat{\omega}_2 = \arcsin\left(\frac{r_o}{r_I} \sin \omega_+\right)$

2) The number, i, over each subsection on ω' axis indicates the table number in Table 4.1(b), 3A.i.

determined by comparing ω_2 of the arc 2 and ω_3 of the arc 3. Here ω_2 and ω_3 are roots of $\gamma_+ = y_1(\omega)$ and $\gamma_- = y_3(\omega)$, respectively. The next path is an arc with a smaller ω. Hence, if $\omega_2 < \omega_3$, the next vertex is 4, which is followed by the vertex 2. Consequently, using Table 3A.4 in Table 1(b), we find the following formula:

$$N_2 = [F_{12}(\omega')]_{\omega_+}^{\omega_2} + [F_8(\omega')]_{\omega_2}^{\omega_3} + [F_6(\omega')]_{\omega_3}^{\omega'_+} . \tag{22}$$

The final formula of N is a summation of Eqs. (21) and (22).

In the above derivation of N we made several assumptions; however, in actual calculations of the transfer matrix elements there are no such assumptions. That is, all parameters necessary to compute the transfer matrix elements are uniquely determined once an outgoing surface and a cone are specified.

All other tables and graphs similar to Tables 1 and Graphs 1.2 and 2.2 are given in Appendix C. In the appendix, associated tables such as Table 2 and functions such as a function y_1 in Table 1(a) are also given.

The primitive functions of integrals are not given in this paper because they are too complicated to list in a limited space. The reader can find the explicit formulas of the functions in function programs in a program TWODCRZ, which will be explained in the following section. For convenience, the names of function programs are given in Table 4.

4. DERIVATION OF THE ESCAPE MATRICES FOR A SOURCE

In the preceding sections, the transfer matrices were derived when no particle source existed in a void. In this section, we shall derive the escape matrices for a source. The transport equation considered is

$$\Psi(\underline{r},\underline{\Omega}) = \Psi(\underline{r}-s_o\underline{\Omega},\underline{\Omega}) + \int_o^{s_o} Q(\underline{r}-s\underline{\Omega},\underline{\Omega})\, ds \ . \tag{1}$$

The transfer matrices for the streaming term, which is the first term of the right-hand side of Eq. (1), are the same as those obtained in Section 3. Hence we consider Eq. (1), neglecting the streaming term:

$$\Psi(\underline{r},\underline{\Omega}) = \int_o^{s_o} Q(\underline{r}-s\underline{\Omega},\underline{\Omega})\, ds \ . \tag{23}$$

We multiply both sides of Eq. (21) by $\underline{\Omega}\cdot\underline{n}_k$ and integrate it over the boundary surface k and the m'th cone. Then we have

$$\iint_{A_k}\iint_{\Delta\Omega_m} \Psi(\underline{r},\underline{\Omega})\ \underline{\Omega}\cdot\underline{n}_k\ dA d\underline{\Omega} = \iint_{A_k}\iint_{\Delta\Omega_m}\int_o^{s_o} Q(\underline{r}-s\underline{\Omega},\underline{\Omega})\ \underline{\Omega}\cdot\underline{n}_k\ ds dA d\underline{\Omega} \ . \tag{24}$$

We apply the discrete cones approximation to the left-hand side of Eq. (24); that is, the angular flux is assumed to be constant over the surface k and the cone m. Then, it becomes

$$\text{L.H.S.} = A_k \left(\iint_{\Delta\Omega_m} \underline{\Omega}\cdot\underline{n}_k\ d\underline{\Omega}\right) \Psi_{km} \ . \tag{25}$$

On the other hand, we apply the standard discrete ordinates approximation to the right-hand side of Eq. (24). Also, we assume that the source is distributed uniformly over the mesh cell and is isotropic. As a result, the right-hand side becomes

$$\text{R.H.S.} = \left(\iint_{A_k} s_o(\underline{r})\ dA\right)(\underline{\Omega}_m\cdot\underline{n}_k)\ w_m Q_m \tag{26}$$

where s_o is the distance between a point, $\underline{r}$, on the surface k and the point where the characteristic in the direction $\underline{\Omega}_m$ extended from the point $\underline{r}$ intersects another boundary surface of the mesh cell, $\underline{\Omega}_m$ is a direction vector of the discrete ordinates, and w_m and Q_m are the same as those of the discrete ordinates method. It is worth noting that there is no contribution of a source of different cones to the outgoing flux because an isotropic source is assumed. Now, by Eqs. (25) and (26), an element of the escape matrix, P_k^m, is represented as follows:

$$P_k^m = \frac{(\underline{\Omega}_m \cdot \underline{n}_k)\, w_m}{\iint_{\Delta\Omega_m} \underline{\Omega} \cdot \underline{n}_k \, d\underline{\Omega}} \frac{\iint_{A_k} s_o(\underline{r})\, dA}{A_k} .$$

Recalling the definition of the adjustment factors described Eq. (2.31) in Section 2 of Ref. 1, we have

$$P_k^m = \frac{1}{\alpha_{km}} \frac{\iint_{A_k} s_o(\underline{r})\, dA}{A_k} \tag{27}$$

where α_{km} is the adjustment factor of the surface k. By using both the transfer matrices and the escape matrices, the m'th outgoing cone flux on the surface k is represented by

$$\Psi_{km} = \sum_{k'} \sum_{m'} T_{kk'}^{mm'} \Psi_{k'm'} + P_k^m Q_m . \tag{28}$$

There are four cases of the escape matrices:

Case 1 is of the surface OUTER,

Case 2 is of the surface TOP for $0 \leq \omega \leq \pi/2$,

Case 3 is of the surface INNER, and

Case 4 is of the surface TOP for $\pi/2 \leq \omega \leq \pi$.

In Eq. (27), A_ks of the Cases 1, 2, 3, and 4 is given by $2\pi r_o \Delta z$, $\pi(r_o^2 - r_I^2)\Delta z$, $2\pi r_I \Delta z$, and $\pi(r_o^2 - r_I^2)\Delta z$, respectively. By taking account of the geometry of a mesh cell as shown in Fig. 2(a), the function $s_o(\underline{r})$ of the integrand in Eq. (27) is obtained. For example, for Case 1, $s_o(\underline{r}) = (z_1 - z_B)/\eta_m$ if $\omega_m < \omega_o$ and $z_1 \leq z \leq z_T$; $s_o(\underline{r}) = (z - z_B)/\eta_m$ if $\omega_m < \omega_o$ and $z_B \leq z < z_1$. Here ω_m and η_m are the polar angle and the direction cosine of the m'th direction of the discrete ordinates, $\omega_o = \arcsin(r_I/r_o)$, and z_1 is given by $z_1 = z_B + \frac{1}{\gamma} + \frac{1}{\gamma}(r_o \cos\omega - \sqrt{r_I^2 - r_o^2 \sin^2\omega})$. By using this result, Eq. (27) becomes

$$P_0^m = \frac{1}{\alpha_{om}} \frac{2\pi r_o}{2\pi r_o \Delta z} \left\{ \int_{z_1}^{z_T} \frac{z_1 - z_B}{\eta_m} dz + \int_{z_B}^{z_1} \frac{z - z_B}{\eta_m} dz \right\} \quad \text{for } z_1 < z_T ,$$

or

$$P_0^m = \frac{1}{\alpha_{om}} \frac{2\pi r_o}{2\pi r_o \Delta z} \int_{z_B}^{z_T} \frac{z - z_B}{\eta_m} dz \quad \text{for } z_1 > z_B .$$

These integrations are easily performed to obtain the final formula of the element. All the elements are obtained in the same way and those are listed in Appendix D.

As we see in the derivation of the escape matrices, the matrix is not obtained so that particles produced by the source in a mesh cell balance the particles leaving the cell. As a result, using these escape matrices reduces the convergence rate of the inner iteration in a numerical calculation. To avoid such a deficiency, we enforce the particle balance by introducing a balance factor.

The balance factor, f, is defined by the following equation:

$$f[2(\sum_{m=1}^{M} 2\pi\mu_m w_m P_o^m Q_m + \sum_{m=1}^{M} 2\pi\mu_m w_m P_I^m Q_m) + 2(\sum_{m=1}^{M} 2\pi\eta_m w_m P_{T1}^m Q_m + \sum_{m=1}^{M} 2\pi\eta_m w_m P_{T2}^m Q_m)] = \frac{1}{2} 4\pi Q_m \pi(r_o^2 - r_I^2)\,\Delta z \ . \tag{29}$$

Here M is the total number of discrete ordinates in an octant of the direction space, and P_o^m, P_{T1}^m, P_I^m, and P_{T2}^m are the elements of the escape matrices for Cases 1, 2, 3, and 4, respectively.

Equation (29) is solved for each void cell to find f, and all the elements P_k^m of the mesh cell are multiplied by f. The new elements are applied to an actual numerical solution of the transport equation. Our experiences with numerical calculations indicates that using the balance factor does indeed accelerate the inner iteration up to the same level as that of the discrete ordinates solution.

5. PROGRAM TWODCRZ

The TWOTRAN-II program was modified so that the new program solves the neutron transport equation in a void by the discrete cones method described in the preceding sections. The new program is named TWODCRZ, whose structure is shown in Table 3. To compute the transfer and escape matrix elements and store them in a common array, the TEMS module is inserted as the fourth overlay after the first overlay, which is an input module.

The TEMS module creates two output files, OUTTM and TESUM. The file OUTTM contains lists of the transfer matrix elements, the adjustment factors, the quadrature set, and the discrete cones boundaries. The file TESUM contains brief lists of the transfer and escape matrix elements and tells whether

Table 3. Structure of the TWODCRZ Program

OVERLAY(0,0)	OVERLAY(1,0)	OVERLAY(4,0)	OVERLAY(2,0)	OVERLAY(3,0)
TWODCRZ[2]	INPUT1	TEMS[3]	GRIND2	OUTPUT3
1 MONITOR	1 LOAD	1 PRECAL	1 REBAL	1 OUTPT31
2 ERROR	2 INPT11[2]	a EQNGEN	2 GRIND21	a FINAL
3 CLEAR	a DUMPRD	b QUADSET	a INITAL	2 OUTPT32
4 MPLY	3 INPT12	c CONSTANT	b INITQ	a EDCALL
5 WRITE	a CSPREP	d ADJUST	c FISCAL	b GENFLO
6 ECHECK	b IFINXS	e OMEGRAN	3 GRIND22	c EDITOR
7 DUMPER	4 INPT13	f TRANS1	a OUTER	d EDMAP
8 PCMBAL	a READQF	g TRANS2	b INNER[2]	3 IFOUT
9 REED	b IFINQF	h TRANS3	c IN[2]	a IFRITE
10 RITE	5 INPT14[2]	i TRANS4	d OUT[2]	
	a EQNCON[1]	j TRANS5	e FIX-UP	
	b IFINSN	k TRANS6	f SETBC	
	c PNGEN	l TRANS7	g STORAF	
	6 INPT15	m TRANS8	h SAVEAF	
	a CSMESH	n TRANS9	i GSUMS	
	b MAPPER	o TRANS10	j VACIN[1]	
		p TRANSQ	k VACOUT[1]	
		q TMQREBL	4 GRIND23	
			a TESTS	
			b NEWPAR	

1) These subprograms are newly added to the TWOTRAN-II program.
2) These subprograms are modified.
3) This module is new.

a sum of the transfer matrix elements of an outgoing surface and a cone added over all possible incoming surfaces and cones becomes unity. If the sum is larger than 1.005 or smaller than 0.995, a warning error message is issued.

In addition to the TEMS module, a subprogram SNCON, which contains the built-in quadrature set, is replaced by a subprogram EQNCON, which provides the equal weight quadrature set EQ_N. Furthermore, subprograms IN and OUT are modified so that in a void IN and OUT call subprograms VACIN and VACOUT, respectively. The subprogram VACIN calculates the discrete ordinates fluxes leaving the void by using the incoming discrete ordinates fluxes, the transfer and escape matrix elements, and the adjustment factors for the inward sweep of space, or in the direction of decreasing r. On the other hand, the subprogram VACOUT does the same for the outward sweep of space, or in the direction of increasing r.

VACOUT uses the transfer matrix elements of Cases 1 through 6 described in Section 2. Since calculations of the outgoing fluxes of cases 2, 3, 5, and 6 require the cone fluxes that are in the inward direction, the fluxes calculated by VACIN in a previous inward sweep must be stored. In contrast, VACIN uses the transfer matrix elements of cases 7 through 10. No fluxes by VACOUT are required in the subprogram VACIN.

Since the transfer matrix depends on incoming cones and surfaces, outgoing cones and surfaces, and the size of spatial mesh cells, it can be represented as a four-dimensional array, TME(n,m,k,i). This denotes the element of the transfer matrix of particles streaming into the m'th cone from the n'th cone in the k'th of ten cases of streaming in the i'th void cell. Here, m varies between 1 and MM. n varies between 1 and NM(= N/2) for Cases 1, 2, 4, 5, 7, 8 ,9, and 10; meanwhile it varies between 1 and 2 x NM for Cases 3 and

and 6. MM is given by MM = N(N + 2)/8 as usual. If the number of void cells that have different spatial intervals is IVT, the total number of the transfer matrix elements, ITME, is given by 12 x NM x MM x IVT.

On the other hand, the escape matrix depends on the surface and the cone in which a particle leaves a void cell. As described in Section 4, there are four types of escape matrices. Therefore, the escape matrix can be represented as a three-dimensional array, TMEQ(m,k,i), where m denotes the cone number, k denotes one of four types, and i is a void cell. As a result, the number of the escape matrix elements stored, ITMEQ, is given by 4 x MM x IVT.

For example, consider a system with a void. Then, suppose that the boundaries of the void are divided into 50 intervals in the r-direction and 10 different intervals in the z-direction. Moreover, suppose the DC_8 approximation is used in the void. Then, ITME = 240,000 and ITMEQ = 20,000. Although this number might be reduced by choosing the mesh intervals carefully, the number is typical for most practically interesting systems. The point is that 242 K memory is not small even for a large computer.

Fortunately, particles crossing an outgoing surface in a cone cannot cross the incoming surface in all the NM or 2*NM cones. In other words, the elements of the transfer matrix are zero for some cones. Hence, we can ease the memory requirement by storing only non-zero elements. This strategy, however, prohibits us from using the transfer matrix in the matrix form because the first dimension of the matrix TME varies cone by cone. Consequently, the computational algorithm becomes complex. In spite of its complexity, this algorithm is applied to the present program to ease the memory requirement. In this algorithm the computer must call the transfer matrix elements, corresponding to each cone and each cell. For this calling, first the place of

desired transfer matrix elements in a storage array must be calculated. This is done by using data provided by a subprogram CHOP as described later in this section.

Due to complexities, the algorithm is expected to be inefficient, especially when the order of the DC_N approximation is high or a large number of spatial mesh cells are used.

The program structure of the TEMS module is illustrated in Appendix E. Roles of subprograms will be explained in the remainder of this section.

OMEGRAN computes the range of the polar angle ω' for outgoing and incoming surfaces and an outgoing cone.

CONSTAT calculates almost all the constants used in the subprograms in the TEMS module, especially the machine epsilon of the computer is computed for floating point arithmetics. Furthermore, it computes the denominators of the transfer matrix elements.

TRANS1,2,3,...,10, BOTOUT, and BOTOP compute the transfer matrix elements for ten cases described in Section 1.

CHOP finds numbers of incoming cones, which correspond to an outgoing cone, and it stores them for use in VACIN and VACOUT.

SEARCH searches a starting vertex and an end vertex on a graph with the help of tables. It finds a possible path between the two vertices, knowing roots associated with the arcs. Finally, it returns the vertex and arc numbers to the program that called it.

SORT is a sorting program.

MULLER solves a nonlinear equation to find real roots in a given range by the Muller method.[5]

SECANT also solves a nonlinear equation by the SECANT method.[6] It is called if we know in advance that only one root exists in a given range.

QUAD numerically integrates single integrals, which cannot be integrated analytically. Gaussian quadrature is used for the integrations. The order is specified as an input variable, IQUAD. The examples described in the next sections are solved by the 8th order Gaussian quadrature because this results in sufficient accuracy for the present computations.

QUADSET computes the cone boundaries.

ADJUST computes the adjustment factors.

RINTGA and RINTGB assign roots to arcs of graphs and computes $y_i(\omega_0)$. Here ω_0 is the minimum or maximum value of ω'. $i = 1, 2, \ldots, 7$ for TRANS6 and $i = 1, 2, \ldots, 8$ for BOTOP. Having these values, it calls the subprogram SEARCH to find primitive functions.

TRANSQ computes the escape matrix elements.

TMQREBL calculates the balance factors to find the adjusted escape matrix elements.

The functions used in this module are included in subprograms shown in Table 4.

6. NUMERICAL RESULTS

PROBLEM 1. Streaming Problem

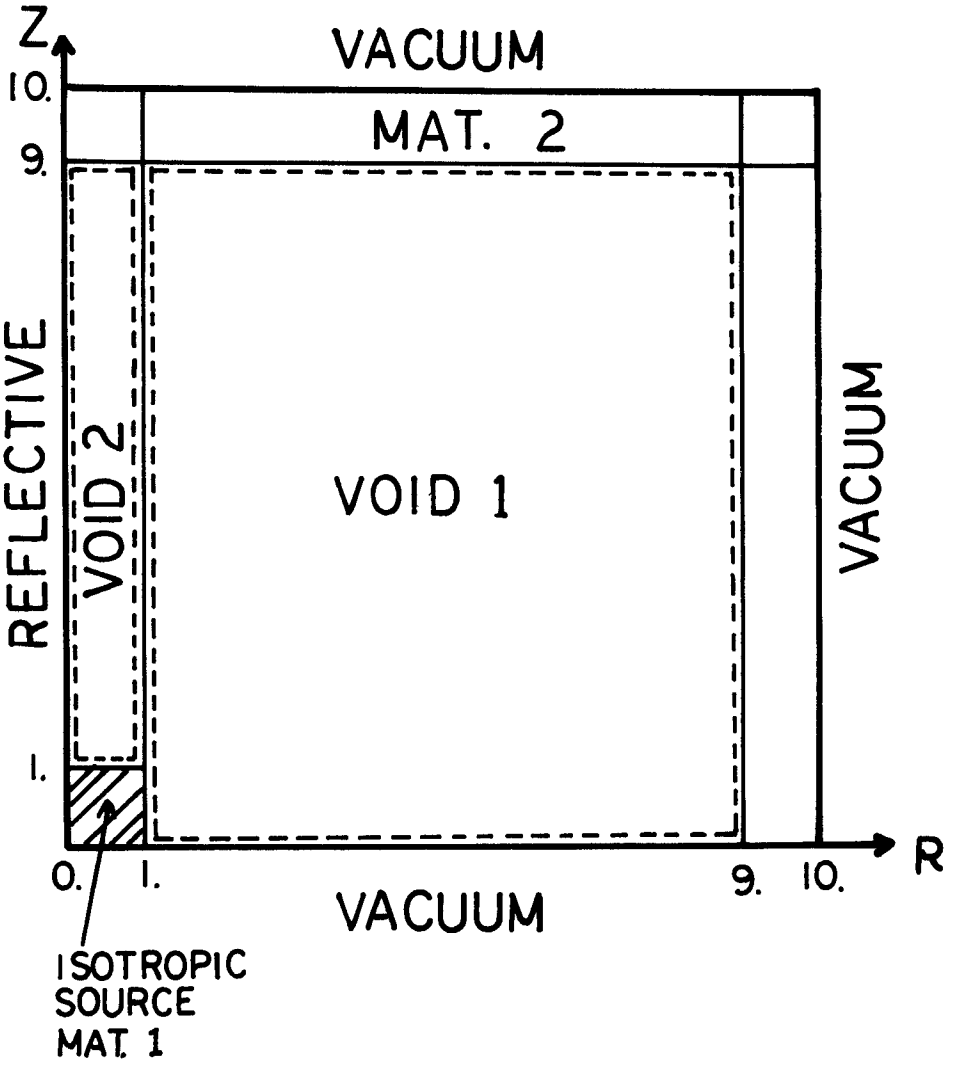
First, we solve a streaming problem for one energy group. As illustrated in Fig. 5, a void is surrounded by a highly absorbing material. An isotropic source is located at the center and the boundary conditions are shown in the figure. In this problem the source region is not void, but consists of a material with very small cross sections; that is, the mean free path is much longer than the dimensions of the source region. To calculate the transfer

Table 4. Function Subprograms that Contain the Functions Necessary to Evaluate the Transfer Matrix Elements

[1]	I → O:		FU11, FU12
[2]	O → O:		FU21, FU22
[3]	B → O:	(A)	FUBOA, QI301
		(B)	FUBOB, GGBOB, QI311
[4]	I → T		FUIT
[5]	O → T		FUOT1
[6]	B → T	(A)	FUBT1A, FBT1A, GBT1A, GGBT1A, QINTBTA, QI601, GAMZBTA
		(B)	FUBT1B, FBT1B, GBT1B, GGBT1B, QINBTB, QI611, GAMZBT
[7]	O → I:		FU71, FU72
[8]	B → I:		FUBI, QI801
[9]	B → T:		FUBT2, FBT2, GBT2, GGBT2, QINBT2, QI901, GAMZBT2
[10]	O → T:		FUOT2

Function subprograms that are called by any of the above subprograms:

SQRT1, TUN, ATUN, SIN1, ASIN1, F1AB, F2AB, F3AB, F4AB



CROSS SECTION	σ_a	σ_f	σ_t	σ_s
MAT. 1	.0	.0	.01	.01
MAT. 2	.8	.0	1.0	.2

Fig. 5 Geometry and cross sections for Problem 1.

matrix elements (TMEs), the TWODCRZ program must know the maximum and minimum Z coordinates, the maximum and minimum R coordinates, and the number of partitions in the R direction for a cylinder or an annular cylinder. These data are provided as inputs to the program.

For the present problem, the void is divided into one annular cylinder, VOID1 and one cylinder, VOID2. In Fig. 5, these regions are surrounded by dashed lines. The entire system is uniformly partitioned into a 10 times 10 mesh. This means that the Z direction of VOID1 is uniformly divided into 9 subsections and the R direction is divided into 8 subsections so that the subtraction of the inner radius from the outer radius results in equal length for all the 8 annular cylinders. Since the TMEs depend on the outer and inner radii themselves rather than the ratio, the TMEs must be individually calculated for the 8 annular cylinders with width 1 in the R direction. In different Z levels, the same TMEs may be used. The Z direction of VOID2 is divided into 8 subsections; meanwhile, the R direction consists of one subsection. Hence, in this case the TMEs for only one set of the inner and outer radii must be calculated.

The problem was solved by TWOTRAN, TWODCRZ, and MCNP. The scalar flux distributions by the S_8, S_{16}, DSC_8 approximations and an MCNP calculation using 1 million particle histories are plotted for the top edge at Z = 9.5 and the right edge at R = 9.5 in Fig. 6(a) and Fig. 6(b), respectively. Here the DSC_N approximation implies that the S_N method in non-voids and DC_N method in voids are used.

Comparing the flux distributions by the DSC_8 solution with that by the S_8 approximation clearly shows a strong mitigation of the streaming ray effects. Qualitatively speaking, the DSC_8 solution is as accurate as the S_{16} solution.

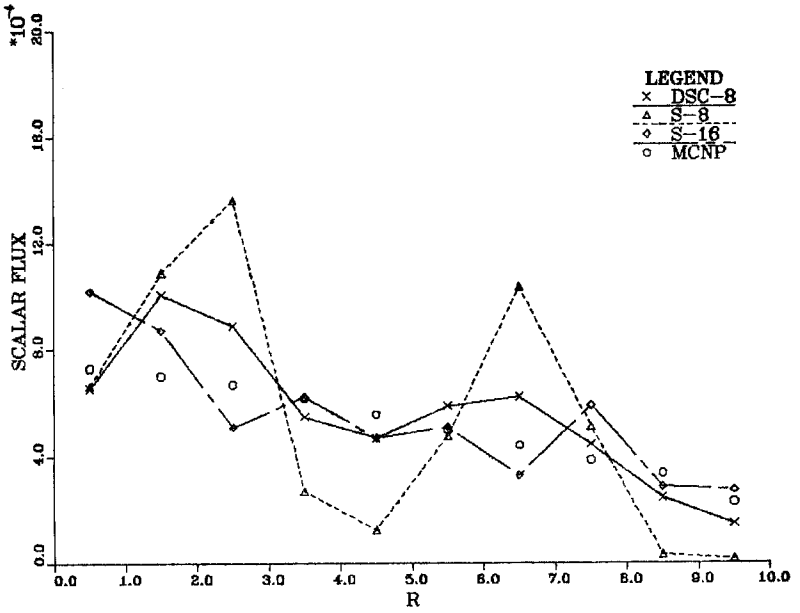


Fig. 6 (a) Scalar flux distribution along the top boundary.

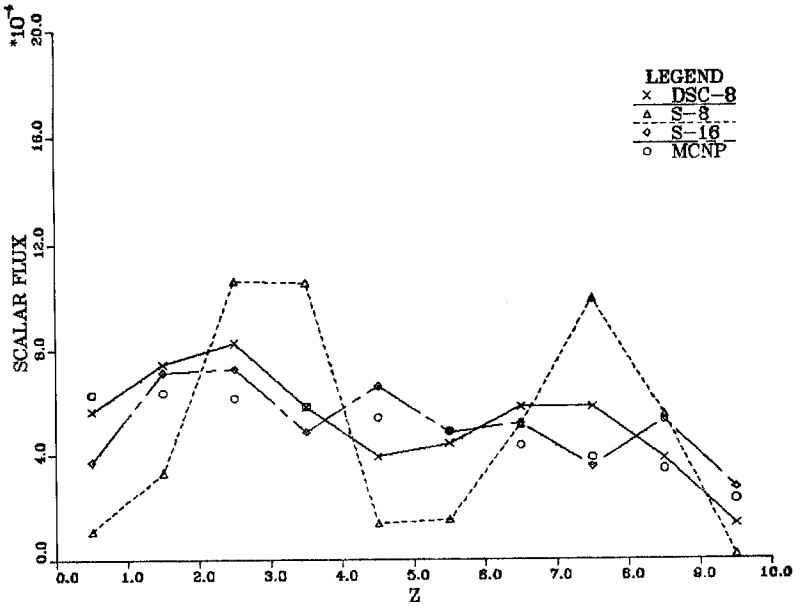


Fig. 6 (b) Scalar flux distribution along the right boundary.

Table 5 represents the computing time of these calculations. The CPU time of the DSC_8 solution is not only much longer than that of the S_8 solution, but also it is more than twice that of the CPU time for the S_{16} solution. For the DSC_8 solution, the CPU time to calculate the transfer matrix elements is 2.53 seconds. Subtracting this from the total CPU time, we find the CPU time that is actually consumed to solve the transport equation. In this case, it is 1.5 seconds, which is almost the same as the total CPU time of the S_{16} solution. Therefore, the current version of TWODCRZ is inefficient compared with TWOTRAN.

However, we can justify this inefficiency from the following point of view. Now, suppose that we solve a multigroup problem, which has a small void compared with a non-void domain. Furthermore, suppose the void contains a localized particle source whose volume is much smaller than the void, and we are interested in the scalar flux distribution near the void as well as in the non-void domain far from the void.

For this problem, we must employ a method which is free from the streaming ray effect. Let us assume, for example, we use TWOTRAN with the S_{16} approximation and TWODCRZ with the DSC_8 approximation. As mentioned in the paragraph second above, the CPU time of these two calculations may be comparable in a void if we exclude the computing time for the transfer matrix elements. Since the S_{16} calculation consumes three to four times more CPU time than an S_8 calculation in the non-void domain, the TWODCRZ calculation is much faster than the TWOTRAN calculation. Consequently, beyond a certain volume ratio of void to non-void, the TWODCRZ calculation with the DSC_8 approximation results in a sufficiently accurate solution in a shorter CPU time than the TWOTRAN solution.

Table 5. Computing Time of Problem 1

	S_8	S_{16}	DSC_6	DSC_8
CPU time at TEMS	---	---	1.515	2.532
CPU time by whole	0.636	1.541	2.647	4.001
CPU (whole)-CPU (TEMS)	0.636	1.541	1.132	1.469
No. of inner iterations	5	5	6	5
CPU time per inner iteration	0.1272	0.3082	0.1887	0.2939

(1) IQUAD = 8 for TEMS

(2) convergence criteria ε = 1.0E-4

(3) units = seconds

If we take account of the computing time of the transfer matrix elements, the ratio may become larger, but such a critical ratio still exists. We can reduce the computing time by using meshes as equal as possible. Also we remind you that the transfer matrix is independent of energy.

To make the TWODCRZ solution faster, we must invent a more efficient algorithm for sweeping spatial meshes and cones. This task is left for the future.

Before we pay much attention to the computing time, however, we should look into the accuracy of the new method in more detail.

PROBLEM 2. Streaming Problem

In this problem, we shall investigate effects of spatial mesh size on the accuracy of the DSC_N solutions. The geometry of the problem is illustrated in Fig. 7. Two voids are specified as shown in the figure. The voids are surrounded by a material with very small cross sections. In fact, the total and scattering cross sections are 1.0×10^{-10}. An isotropic source is located on the Z axis in the same material. These domains with the low cross section material are solved by the S_8 approximation.

The whole system is partitioned in three different ways:

a) $\Delta R = 1.0$, $\Delta Z = 2.0$

b) $\Delta R = 0.5$, $\Delta Z = 1.0$

c) $\Delta R = 0.25$, $\Delta Z = 0.5$,

and these cases are solved by the S_8, S_{16}, DSC_8 approximations and the MCNP code.

Scalar fluxes of subdomains represented by numbers are obtained by averaging several scalar fluxes in the subdomain. The average fluxes are shown in Table 6 for the solutions mentioned above. In this table, the lower

Table 6. The Average Scalar Fluxes of Problem 2

	1	2	3	4	5	6
DSC_8	3.39244E-3	1.44016E-3	1.09128E-3	7.11857E-4	1.32837E-3	3.66402E-4
a	+29.4%	-17.1%	+28.3%	-1.40%	+60.1%	-58.6%
b	3.05325E-3	1.78735E-3	1.18373E-3	4.01107E-4	1.63951E-3	2.59560E-4
	+16.4%	+2.94%	+39.2%	-4.44%	+97.6%	-70.7%
c	2.76701E-3	1.65779E-3	1.28793E-3	2.64093E-4	1.83408E-3	1.82279E-4
	+5.52%	-4.52%	+51.5%	-63.4%	+121%	-79.4%
S_8	1.98575E-3	1.48616E-3	1.38737E-3	2.82503E-4	1.93335E-3	5.93910E-4
a	-24.3%	-14.4%	+63.2%	-60.9%	+133%	-32.9%
b	2.10470E-3	3.74208E-3	1.67281E-3	2.33920E-6	2.17149E-3	2.49834E-4
	-19.7%	+116%	+197%	3.23×10^{-3}	2.616	-71.8%
c	2.44322E-3	2.13046E-3	1.60984E-3	1.54575E-5	2.18724E-3	1.03753E-4
	-6.82%	+22.7%	+89.3%	0.0214	2.636	-88.3%
S_{16}	2.72815E-3	1.70638E-3	7.36456E-4	6.57655E-4	7.18571E-4	1.13015E-3
c	+4.04%	-1.73%	-13.4%	-8.91%	-13.4%	+27.6%
MCNP*	2.62217E-3	1.73634E-3	8.50251E-4	7.21993E-4	8.29854E-4	8.85576E-4
	(±0.53%)	(±0.22%)	(±0.23%)	(±0.29%)	(±0.33%)	(±0.71%)

* 1,000,000 histories, 10 minutes on CDC-7600.

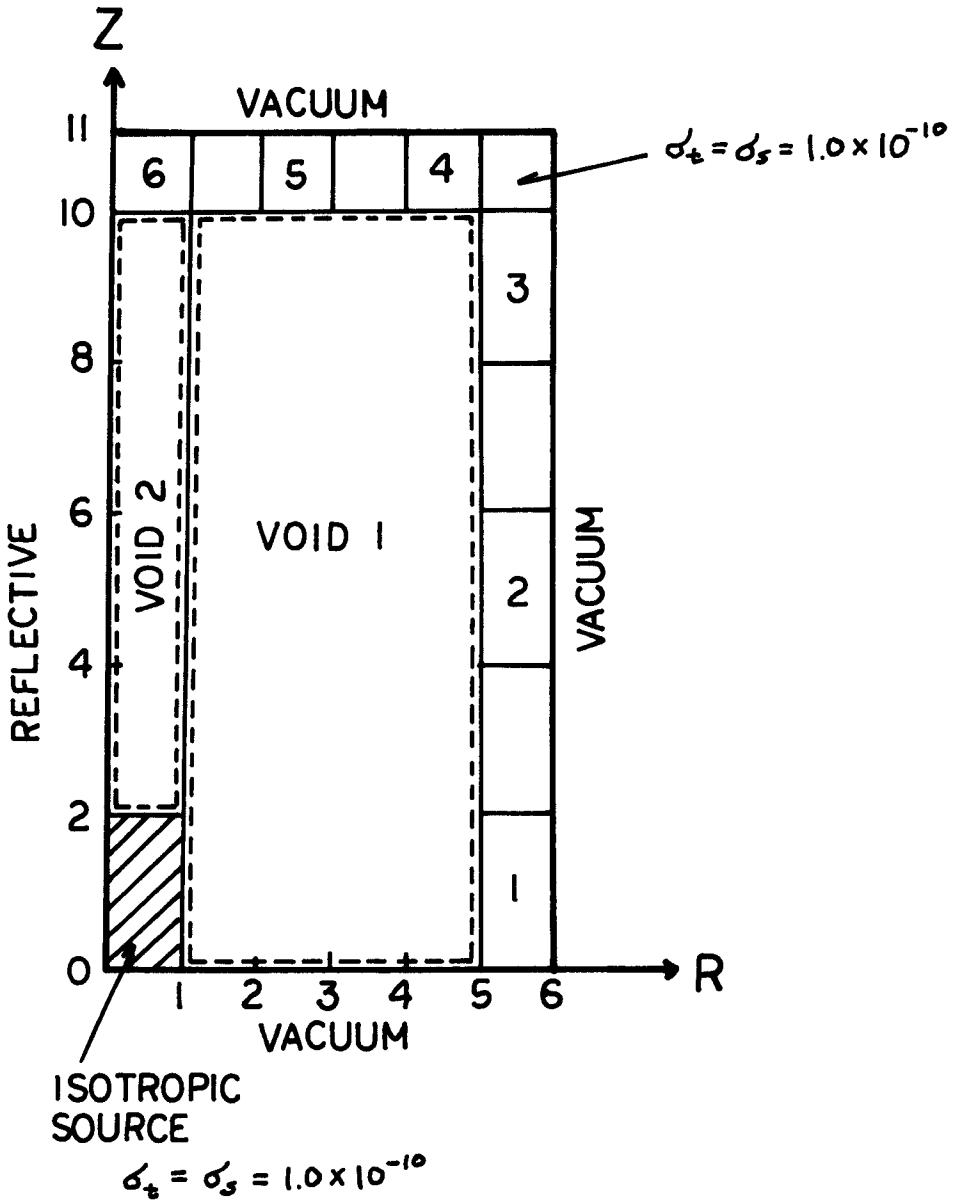


Fig. 7 Geometry and cross sections for Problem 2.

number in a box is a percent error compared with the MCNP solution. If the number does not have a symbol %, then it indicates the ratio of the solution to the MCNP solution.

First, we realize that refining the spatial mesh does not result in a better accuracy of the DSC_8 solutions. On the contrary, the error increases as the mesh size decreases, except solutions for the DSC_8 approximation. Unexpectedly, the same thing is true for the S_8 solutions. As a whole, we see much better accuracy of the DSC_8 solutions than the S_8 solutions. However, comparing the average fluxes at subdomains 5 and 6 at the top of the system by the DSC_8 approximation with the S_{16} solutions we find that the DSC_8 does not give sufficient accuracy.

Next, we compare more global quantities. Table 7 represents the right and top leakages. The right leakage is a leakage from a cylindrical surface with R = 5 and Z = 0 to 10. The top leakage is a leakage from a circle with a radius 5 at Z = 10. Differing from the average flux in a region, these quantities become more accurate as the mesh size decreases. The top leakage by DSC_8 for case (a) is less accurate than S_8 for case (a), but for the other two cases, the DSC_8 solutions are much more accurate than the S_8 solutions. In conclusion, about the accuracy of integrated quantities such as leakages, the DSC_8 and S_8 solutions are comparable, and both solutions are acceptable.

PROBLEM 3. Duct Streaming Problem

Because of the nature of the discrete ordinates approximation, it is difficult for the S_N method to solve a narrow duct streaming problem accurately. Here, a narrow duct is one whose height (L) is much larger than its diameter (D), i.e., $L/D \gg 1$. The following problem is solved to show the usefulness of the DSC_N approximation for such a narrow duct streaming problem.

Table 7. Right and Top Leakages of Problem 2

		Right Leakage	Top Leakage
DSC_8	a	0.54972 (+2.90%)	0.0721901 (+14.5%)
	b	0.54569 (+2.15%)	0.0642282 (+1.85%)
	c	0.53838 (+0.777%)	0.0613779 (-2.67%)
S_8	a	0.51025 (-4.49%)	0.0612613 (-2.86%)
	b	0.52823 (-1.14%)	0.0501096 (-20.5%)
	c	0.54082 (+1.21%)	0.0514688 (-18.4%)
S_{16}	c	0.52423 (-0.018%)	0.0638051 (+1.18%)
MCNP		0.53433 (± 0.20%)	0.0630628 (±0.14%)

* The numbers inside parentheses are percent errors from MCNP solutions.

The schematic diagram of the problem is illustrated in Fig. 8. Its L/D is 8. The S_8, S_{16}, and DSC_8 approximations are applied, and an analytical solution is obtained. For the numerical solutions, the system is partitioned into 6 times 81 mesh cells.

The scalar flux distribution along the Z-axis in the outer layer, i.e., R = 5.5, is plotted in Fig. 10. An analytical solution, which is obtained by integrating a point source kernel over the region of the surface source, is represented by

$$\phi(z) = \frac{q_o}{4} \ln \left| \frac{z^2 + a^2 - b^2 + \sqrt{z^4 + 2(a^2 + b^2)z^2 + (a^2 - b^2)}}{2z^2} \right| \qquad (30)$$

where q_o is the number of particles produced per unit area per unit time, a = 5.0, and b = 5.5 cm. This analytical solution is also plotted in Fig. 10. Comparing the DSC_8 and S_8 solutions, we observe oscillations in the S_8 solution. These are due to the ray streaming effect. The DSC_8 approximation eliminates the anomaly. Despite this success, the DSC_8 solution decreases sharply around Z = 40. At Z = 70, it is an order smaller than the exact solution. Although the deviation is smaller than that of the S_8 solution, the error is unacceptable as a solution.

The sharp decline of the S_8 solution is explained by considering the discrete directions used in S_N solutions. The current S_N solutions apply the quadrature set EQ_N. The maximum η for EQ_8 and EQ_{16} are 0.9603506 and 0.9889102, respectively. In R-Z geometry, η is defined by $\eta = \cos\theta = \underline{e}_z \cdot \underline{\Omega}$ as illustrated in Fig. 1. Hence, the maximum η gives the minimum θ: θ_{min}. θ_{min} is 0.2825393 for EQ_8 and 0.1490661 for EQ_{16}. Now, consider a cylinder having a radius a and a surface source on the bottom boundary as illustrated in Fig.

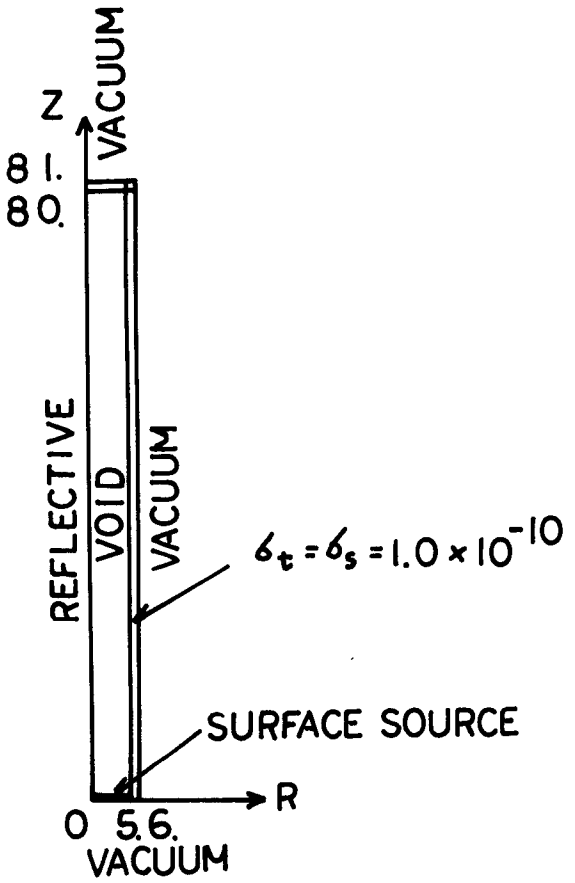


Fig. 8 Geometry and cross sections for Problem 3.

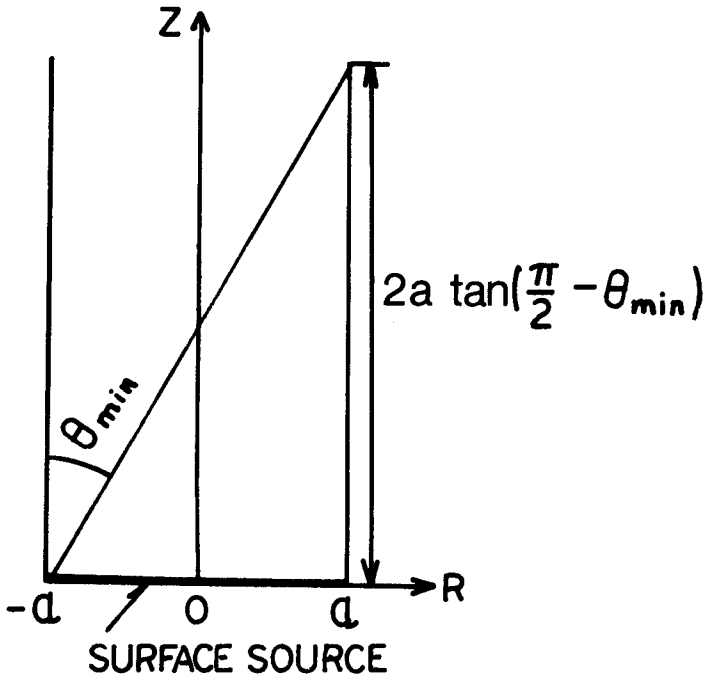


Fig. 9 Schematic diagram of definition of the critical length.

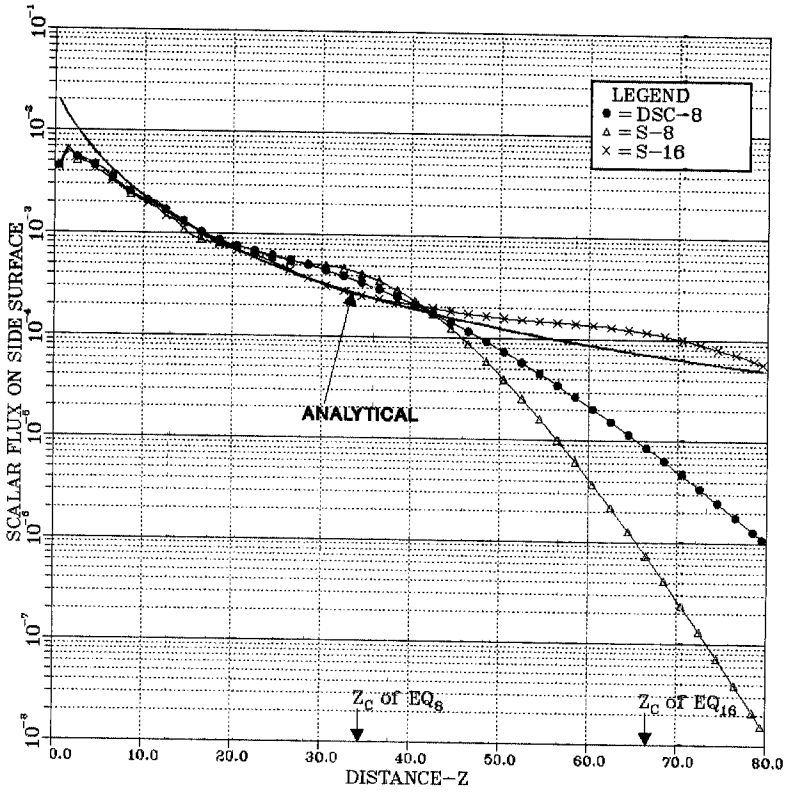


Fig. 10 Scalar flux vs. distance in the z-direction for Problem 3.

9. Since no particles are emitted in a direction with θ smaller than θ_{min}, no particles can reach the outer boundary surface beyond a critical Z: $Z_c = 2a \tan(\pi/2 - \theta_{min})$. In the current problem, $Z_c = 34.44646$ for EQ_6 and $Z_c = 66.58671$ for EQ_{16}. These critical points are shown on the Z axis in Fig. 10. The figure indicates a strong correspondence between the critical points and points where the solutions start falling sharply. Although the DC_N approximation does not have the minimum θ, the DSC_8 solution also sharply falls around Z_c. To some extent, the reason will be discussed in Section 7.

In Fig. 10, we also observe that the S_8, S_{16}, and DSC_8 solutions are somewhat larger than the exact solution around the critical Z. This is because any approximate solution has a point beyond which the solution is much smaller than the exact solution. Since both S_N and DC_N solutions solve the transport equation so that particles in a system are conserved, the flux must be larger than the exact value somewhere to compensate the lack of particles. In conclusion, the DSC_N method as well as the S_N method results in inaccurate solutions at the end of a duct, whose $L/D >> \tan(\pi/2 - \theta_{min})$.

PROBLEM 4. Multigroup Problem (A Tokamak Reactor Model)

The problem of a fusion reactor with a bean shape plasma discussed in Ref. 1 is solved in R-Z geometry by using the TWODCRZ program. The geometry and the source distribution are the same as those illustrated in Ref. 1. However, the X axis and the Y axis in the figure are replaced by the R axis and the Z axis, respectively. In other words, the present solution can take the toroidal effect into account. The left and bottom boundaries are reflective, and others are vacuum.

The problem is solved by the S_8 method (TWOTRAN-II) and the DSC_8 method (TWODCRZ) using the 30 neutron + 12 gamma energy group cross section library described in Ref. 1.

Since the neutron source concentrates around a point (R = 155 cm and Z = 0 cm), the streaming ray effect may significantly affect the flux distribution on the surrounding walls. Since the source is only in the second energy group, the streaming ray effect in other energy groups may be negligible. Therefore, we apply the DSC_8 method only to the second energy group and solve other energy groups by the standard S_8 approximation.

The results are illustrated in Fig. 11(a) and (b). These figures show the scalar flux of the second energy group for the left and top edges of the domain. As seen in Fig. 11(a), the streaming ray effect is strongly mitigated by the DSC_8 method.

In the DSC_8 solution, the CPU time for the calculation of the transfer and escape matrices is 5.3 seconds, and the total CPU time is 232 seconds. Hence, the cost of the matrix calculations is negligible compared with that needed by other parts of the computation. Meanwhile, the total CPU time of the S_8 solution is 184 seconds.

In conclusion, the DSC_8 method can result in a much better solution without a significant increase of computing cost if the discrete cones method is applied only to energy groups in which the streaming effect is important.

7. DIFFICULTY OF DUCT STREAMING PROBLEMS

A goal of this section is to understand why duct streaming problems are very difficult even for the discrete cones (DC_N) approximation. To achieve this goal, we must know how well an angular flux occurring in the duct streaming problem is reproduced by the DC_N approximation.

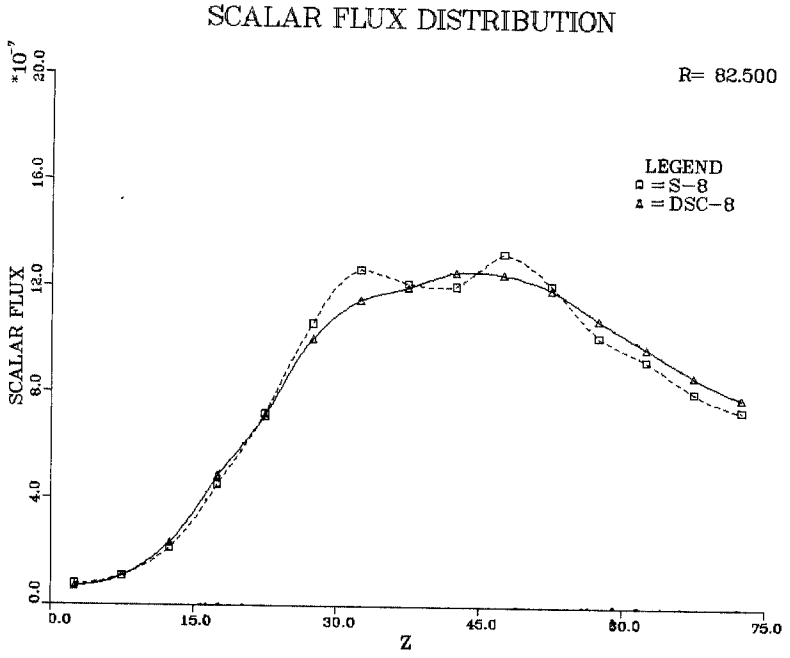


Fig. 11 (a) Scalar flux distribution at R=82.5cm.

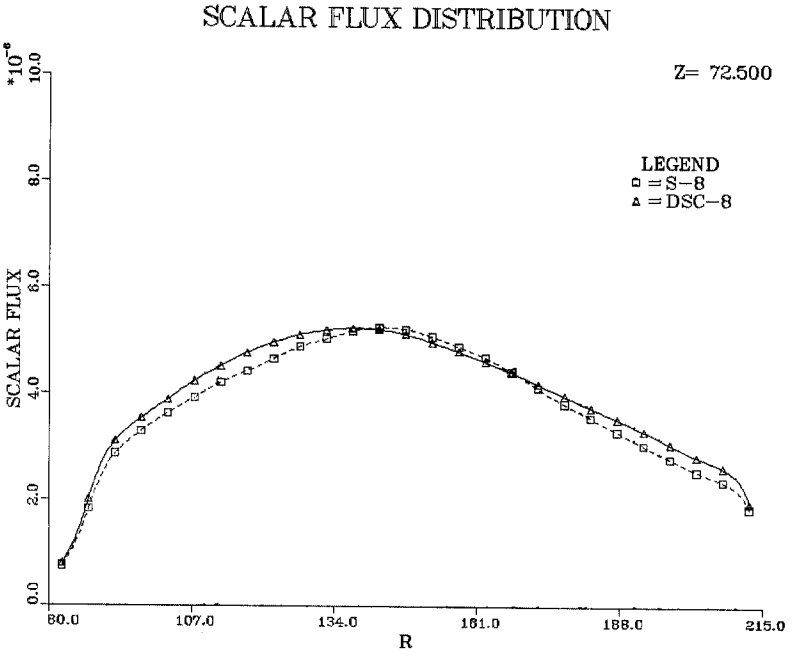


Fig. 11 (b) Scalar flux distribution at Z=72.5 cm.

First, we discuss a problem in X-Y geometry. The system illustrated in Fig. 12 is considered. In the system an isotropic particle source is uniformly distributed on the bottom boundary. We can easily find the angular flux on the top boundary. If the surface source density is Ψ_0, the angular flux at a point A in a direction $\underline{\Omega} = (\omega,\xi)$ is given by

$$\Psi(x,L,\omega) = \begin{cases} \Psi_0 & \text{for } -\arctan\left(\frac{a-x}{L}\right) \leq \omega \leq \arctan\left(\frac{a+x}{L}\right) \\ 0 & \text{otherwise} \end{cases} . \qquad (31)$$

Since the flux is independent of the variable ξ, it is represented as a three-dimensional domain as shown in Fig. 13. Here $\xi = \underline{e}_z \cdot \underline{\Omega}$.

The current at A is exact in the DC_N method because the particles are conserved. Consequently, the total current on the top boundary is also exact. However, a problem arises when we try to find the current on the top surface of a mesh cell above the present mesh cell. For the new cell, the angular distribution of the bottom source is not isotropic, but it is given by Eq. (31) as a function of the coordinate x. Furthermore, in the DC_N approximation this is not directly used, but a distribution obtained by averaging the true distribution over both the boundary surface and a cone is used to find the angular flux on the top boundary.

In order to explain it more rigorously, suppose the DC_4 approximation for the solution. The cone boundaries of this approximation are listed in Table 8.

For convenience, assume that $-\pi/4 \leq -\arctan(a - x/L) \leq \pi/2 \leq \arctan(a + x/L)$. Then, the angular flux at a point of the bottom boundary is repre-

Table 8. The Cone Boundaries of the DC_4 Approximation

Cone m	$\omega_{m-1/2}$	$\omega_{m+1/2}$	$\xi_{m-1/2}$	$\xi_{m+1/2}$
1	0	$\pi/2$	2/3	1
2	0	$\pi/4$	0	2/3
3	$\pi/4$	$\pi/2$	0	2/3
4	$-\pi/2$	0	2/3	1
5	$-\pi/4$	0	0	2/3
6	$-\pi/2$	$-\pi/4$	0	2/3

sented by a domain, whose cross sections by a constant ξ-plane are shown as dashed lines in Figs. 14(a) and 14(b). Here, Fig. 14(a) illustrates the flux of the 1st ξ-level (i.e., $2/3 \leq \xi \leq 1$); Fig. 14(b) does that of the 2nd ξ-level (i.e., $0 \leq \xi \leq 2/3$). This angle distribution, however, is not used to obtain the angular flux on the top boundary. Indeed, the angle distribution of the flux is approximated by six poles, each of which corresponds to a cone flux at the spatial point. The cross sections of these poles are illustrated by solid lines in Figs. 14(a) and 14(b). These are obtained by considering the definition of the cone flux Ψ_m^C:

$$\Psi_m^C = \int_{\Delta\Omega_m} \Psi(\Omega)\, d\underline{\Omega} \;/\; \int_{\Delta\Omega_m} d\underline{\Omega} \,. \tag{32}$$

In Fig. 14(a), the flux of the 1st cone is reduced for $0 \leq \omega \leq \arctan(a + x/L)$; meanwhile, it is produced for $\arctan(a + x/L) \leq \omega \leq \pi/2$, where the flux should not exist. Similar reductions and productions of the flux occur in

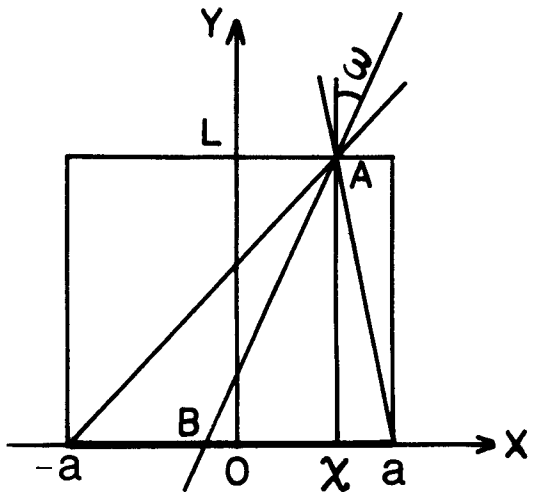


Fig. 12 Geometry for a duct streaming problem in X-Y geometry.

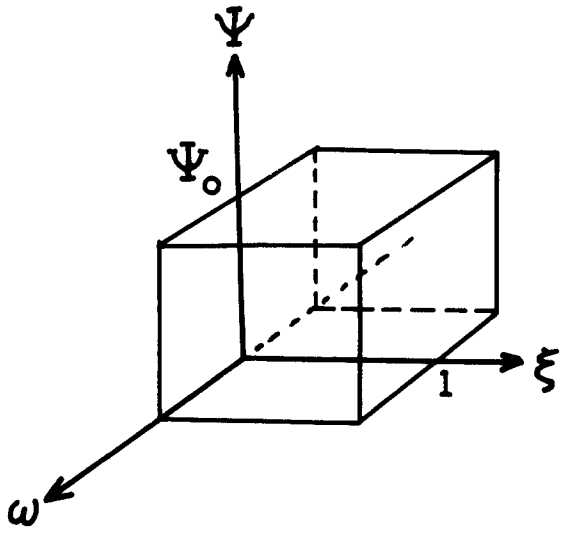
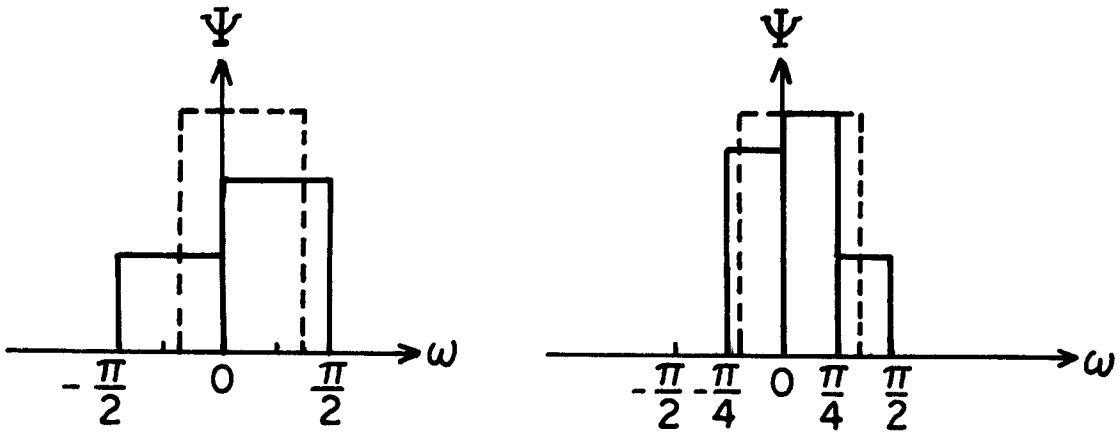


Fig. 13 The angular flux at A in Fig. 12.



a) b)

Fig. 14 (a) The cone flux for the first ξ-level of the DC_4 approximation.
(b) The cone flux for the second ξ-level of the DC_4 approximtion.

other cones as shown in Fig. 14(a) and (b). As a result, using this angle distribution of the flux as the bottom boundary source produces a smaller flux on the top boundary and larger fluxes on the right and left boundaries.

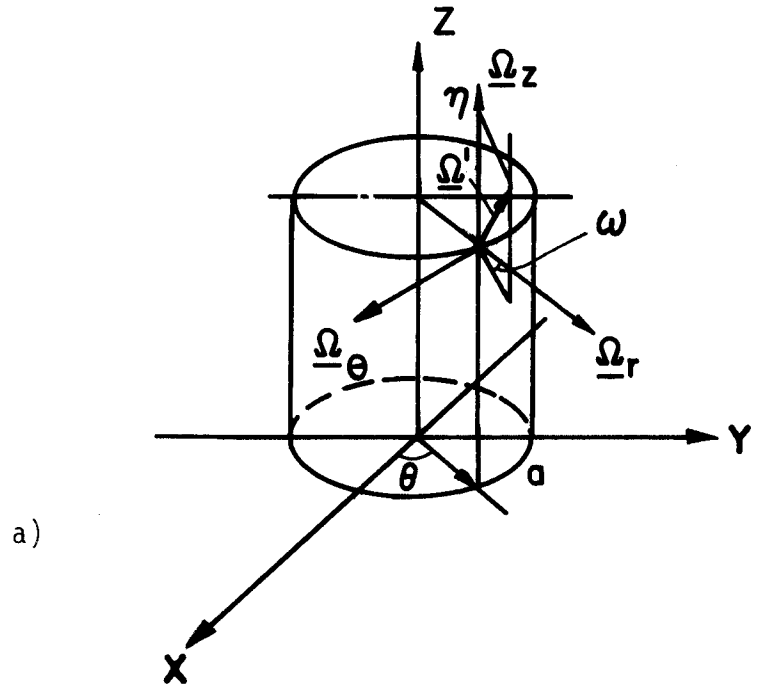
As for a long and narrow duct, we may partition it into many mesh cells in the y-direction. Then the ratio of the flux on the top and side surfaces deviates more and more from the correct value as the calculation proceeds to cells farther and farther from the source. Therefore, the DC_N approximation cannot provide a sufficiently accurate solution for the duct streaming problem.

As the above arguments imply, a way to overcome the difficulty is to use as few as possible mesh cells in the axial direction (the y direction) of the duct. As an extreme case, using only one mesh cell may result in the best solution on the top surface. However, the spatial resolution of the flux on the side surfaces is very poor because a constant flux on a surface is assumed by the DC_N approximation. To overcome the latter defect, we can formulate a method in which boundary surfaces of a duct are partitioned into subsurfaces, and the transfer matrices among these subsurfaces are used. This method is adopted in Ref. 1.

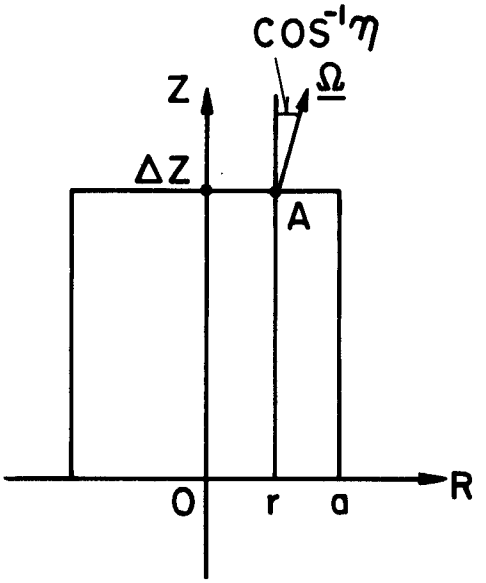
Next, we discuss a problem in R-Z geometry. The system is illustrated in Figs. 15(a), (b), and (c). We want to find the angular flux at A on the top boundary, provided that an isotropic source is uniformly distributed on the bottom boundary.

If a particle generated at B of the bottom boundary travels to A of the top boundary, the following equation holds:

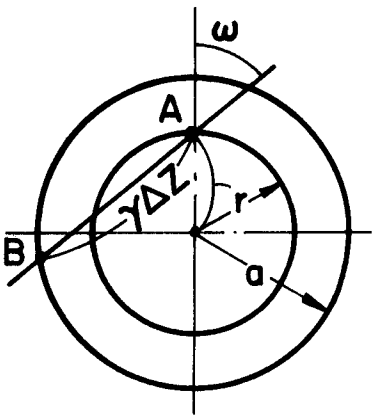
$$\gamma\Delta z = \sqrt{r'^2 - r^2 \sin^2\omega} + r \cos \omega \qquad (33)$$



a)



b)



c)

Fig. 15 a) Geometry of a cylindrical duct.
b) A cross sectional view of a cylindrical duct.
c) A top view of a cylindrical duct.

where $\gamma = \sqrt{1 - \eta^2}/\eta$, ω and η are elements of a direction vector at A and r' is the R coordinate of point B. $\gamma\Delta Z$ must be smaller than $\gamma_{max}\Delta z$ $(= \sqrt{a^2 - r^2 \sin^2\omega} + r \cos \omega)$ so that the point B lies on the bottom boundary surface. In other words, $\eta > \eta_{min}$ $\left(= 1/\sqrt{1 + \gamma^2_{max}}\right)$. Hence, the angular flux at A is represented by the following function:

$$\Psi(r,\Delta z,\omega,\eta) = \begin{cases} \Psi_o & \text{if } \omega \text{ and } \eta \text{ are in a hatched domain in Fig. 4.16(a) or 4.16(b)} \\ 0 & \text{otherwise} \end{cases} . \qquad (34)$$

Figure 16(b) also shows the cone boundaries of the DC_6 approximation. As seen in this figure, the angular flux exists only in the cones 1 and 2. The DC_N approximation replaces the flux Ψ_o by smaller fluxes in these cones. Hence, using these cone fluxes as a bottom surface source for the next mesh cell, which is above the present mesh cell, produces an error of the flux on the top and outer boundary surfaces. Moreover, the approximate solution gets worse as the streaming goes farther.

A difficulty specific to R-Z geometry is that the angular flux is not approximated as well as is done in X-Y geometry even if the order of the DC_N approximation increases. The reason is that increasing N provides more η-levels, but it does not increase the number of cones lying in the first η-level, i.e. the largest η. As shown in Fig. 16(b), the angular flux concentrates near $\eta = 1$. The variable ω of the cone varies between 0 and $\pi/2$. Consequently, the variation of the flux in the ω-direction cannot be approximated well by the cones.

To verify the above arguments, we solve a problem whose spatial scale is one-tenth of that of Problem 3 in Section 5 (Fig. 8); however, the axial length is 10.1 cm instead of 8.1 cm and all other conditions are the same.

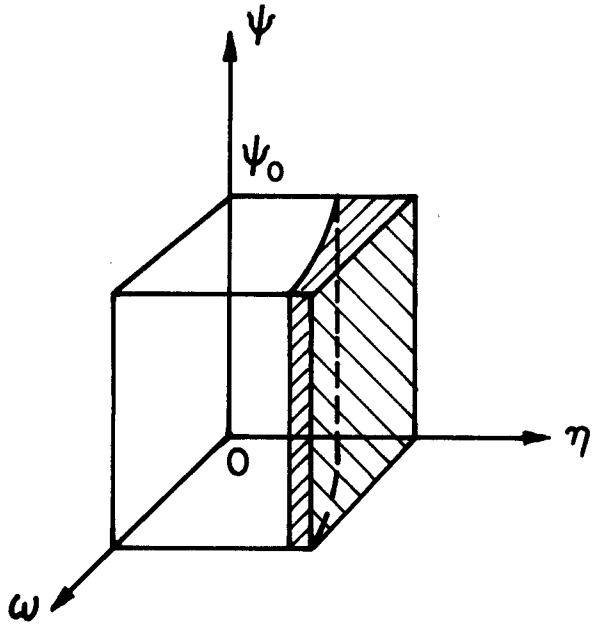


Fig. 16 (a) Schematic diagram of the angular flux at point A.

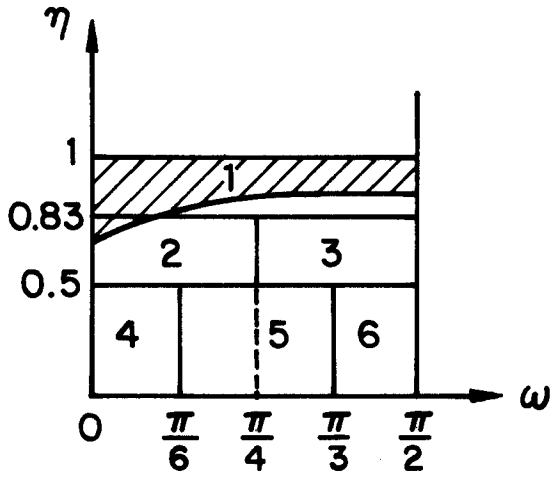


Fig. 16 (b) The angular flux (shaded) and the cone boundaries (numbered) for the DC_6 approximation. A=2, R=1, and Δz=4.

The duct is divided into one interval in the R-direction and JTVAC z-zones in the Z-direction. By varying the parameter JTVAC, a series of the DSC_4, DSC_6, and DSC_8 calculations are performed. The scalar fluxes on the top boundary surface of the duct are plotted in Fig. 17. A MCNP solution (1% error) is also shown in the figure.

The solution for one z-zone is almost identical to the MCNP solution. As the number of z-zones increases, the scalar flux falls sharply. This can be explained by the above arguments. The values, however, start increasing at 10 z-zones and they keep increasing. We can see from Eq. (33) that the angle distribution of the flux becomes uniform over the entire domain of the direction space as Δz approaches 0. This implies that the DC_N approximation leads to a correct value at such a limit.

In conclusion, it is recommended that only one z-zone or z-zones as many as possible should be used to obtain a good solution at the end of a duct. The former technique saves computing time, but it sacrifices the spatial resolution of the flux on the side surface of the duct. Meanwhile, the latter results in much computing time and the solution cannot be as good as one by the former. Consequently, one of the feasible solutions is to partition the boundary surfaces of a void into subsurfaces and obtain the transfer matrices among these subsurfaces. This, indeed, was carried out by Clark.[2]

8. DISCUSSION AND CONCLUSIONS

The hybrid method of the discrete ordinates and discrete cones solutions (the DSC_N method) was extended to R-Z geometry. The new method strongly mitigates the ray streaming effect. Although much work is required for the computation of the transfer matrices and the process marching spatial mesh cells and cones in an inner iteration, it is suitable especially for a problem

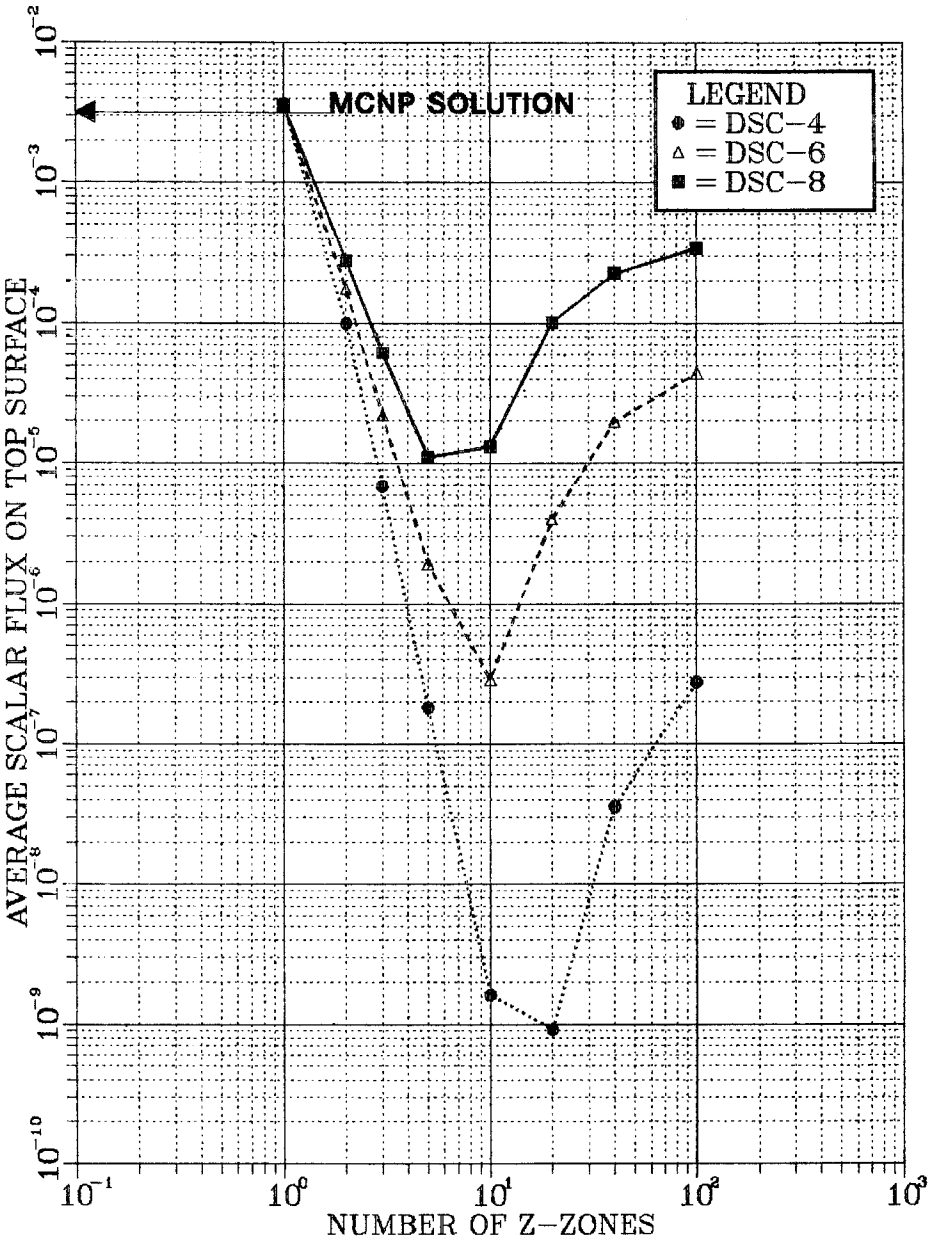


Fig. 17 Scalar flux on the top boundary vs. number of z-zones.

in which the void is much smaller than the non-void and the streaming ray effect significantly affects the solution.

In spite of the success, the method has two deficiencies. One of them is the poor computational efficiency. In fact, a DC_8 solution needs as much CPU time as an S_{16} solution. Such a long computing time is spent by the computation of the transfer matrices and inner iterations. The latter is due to complexities of the solution algorithm. Switching the algorithm to an algorithm where a fixed dimension of the transfer matrices is used indeed shortens the CPU time spent by the inner iteration process, but only 10 to 30% reduction was achieved. Another remedy is to use only two cone fluxes of an incoming surface to find an outgoing cone flux. This is similar to the algorithm of the S_N method, in which only the neighboring discrete ordinates flux is used to find a discrete ordinate flux in curved geometry.[7] Unfortunately, the algorithm destroys the convergence efficiency of the inner iteration because using elements of the transfer matrices partially does not conserve particles.

Another deficiency is that the DC_N method does not work sufficiently well for duct streaming problems. As we described in Section 7, a remedy is to use transfer matrices among subsurfaces of the void boundaries. This was accomplished by Clark.[2] Another remedy is to divide cones so that more cones lie near the z-axis and use them for a solution.

Acknowledgment

Support for this work has been provided by the United States Department of Energy.

References

1. Y. Watanabe and C.W. Maynard, "The Discrete Cones Method in Two-Dimensional Neutron Transport Computation," University of Wisconsin Fusion Technology Institute Report UWFDM-574, May 1984.

2. B.A. Clark, "The Development and Application of the Discrete Ordinates-Transfer Matrix Hybrid Method for Deterministic Streaming Calculations," Los Alamos National Laboratory LA-9357-T, 1982.

3. B.A. Clark, "The Streaming Matrix Hybrid Method for Discrete-Ordinates Calculations," Proc. Top. Meeting ANS Mathematics and Computation Division, Salt Lake City, Utah, March 1983.

4. Claude Berge, Graphs and Hypergraphs, North-Holland Pub., Amsterdam, London (1973).

5. S.D. Conte, and Carl de Boor, Elementary Numerical Analysis, McGraw-Hill Book Company (1972).

6. Kendall E. Atkinson, An Introduction to Numerical Analysis, John Wiley & Sons, New York (1978), p. 48.

7. K.D. Lathrop and F.W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," LA-4848-MS, Los Alamos Scientific Laboratory (1973).

APPENDIX A. DERIVATION OF EQS. (5a), (5b) AND (5c)

In this appendix, we shall derive Eqs. (5a), (5b) and (5c). As already defined, spatial points A = (r,z) and B = (r',z') lie on the same characteristic line whose direction vector is $\underline{\Omega}$. And the coordinates of the vector $\underline{\Omega}$ at A and B are (η,ω) and (η',ω'), respectively.

In the coordinates system utilized in the present paper, the transport equation in R-Z geometry is represented by the following equation:[1]

$$\sqrt{1-\eta^2}\cos\omega\frac{\partial\Psi}{\partial r}-\frac{\sqrt{1-\eta^2}\sin\omega}{r}\frac{\partial\Psi}{\partial\omega}+\eta\frac{\partial\Psi}{\partial z}=q-\sigma_t\Psi\ . \tag{A.1}$$

Since this is a first order partial differential equation, a system of characteristic differential equations is given by[2]

$$\frac{dr}{ds}=\sqrt{1-\eta^2}\cos\omega\ , \tag{A.2}$$

$$\frac{d\omega}{ds}=-\frac{\sqrt{1-\eta^2}\sin\omega}{r}\ , \tag{A.3}$$

$$\frac{dz}{ds}=\eta\ , \tag{A.4}$$

$$\frac{d\eta}{ds}=0\ , \tag{A.5}$$

$$\frac{d\Psi}{ds}=q-\sigma\Psi\ . \tag{A.6}$$

Eliminating the variable s from Eqs. (A.2) and (A.3) and rearranging it, we have

$$\frac{dr}{r} = -\frac{d\omega}{\tan \omega} .$$

Integrating this result in

$$r \sin \omega = \text{constant} \equiv C_1 . \tag{A.7}$$

Dividing Eq. (A.2) by Eq. (A.4) and rearranging it, we have

$$\frac{dr}{\cos \omega} = \gamma \, dz ,$$

where $\gamma = \sqrt{1 - n^2}/n$. We eliminate the variable ω by using Eq. (A.7) to find

$$\frac{r \, dr}{\sqrt{r^2 - C_1^2}} = \gamma \, dz .$$

Integrating this and applying Eq. (A.7) again, we have

$$r \cos \omega - \gamma z = C_2 . \tag{A.8}$$

Note that C_1 and C_2 are constant along the characteristic line. Therefore, Eqs. (A.7) and (A.8) give Eqs. (5b) and (5c), respectively. Since $dn/ds = 0$, n is constant. This leads to Eq. (5a).

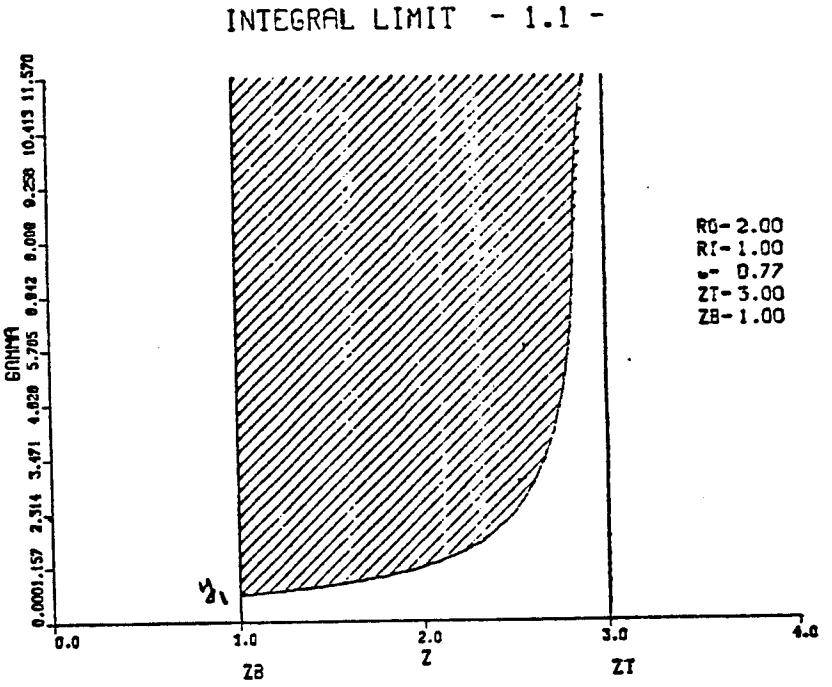
References for Appendix A

1. K.D. Lathrop and F.W. Brinkley, "TWOTRAN-II: An Interface, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory Report LA-4848-MS (1973).

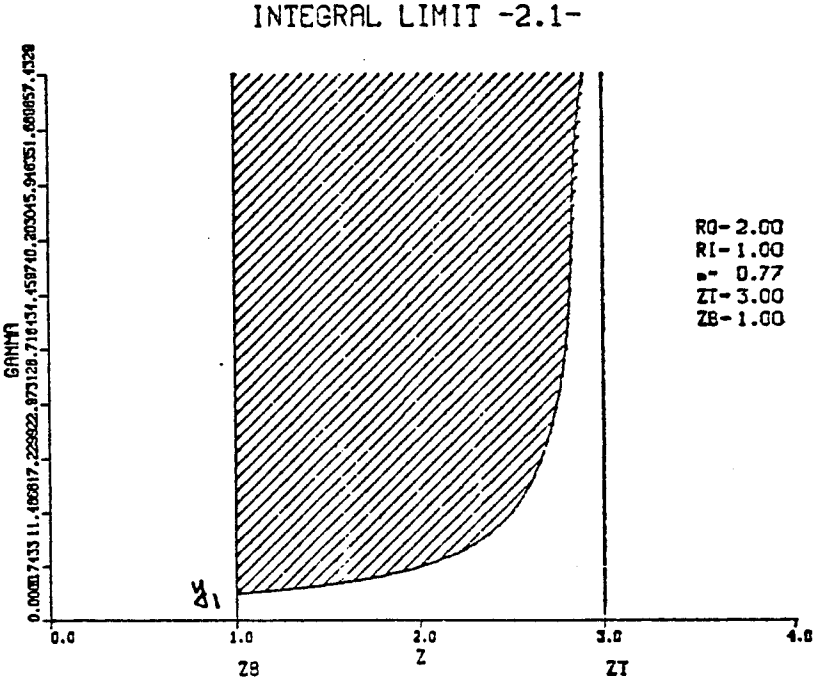
2. R. Courant and D. Hilbert, Method of Mathematical Physics, Interscience Pub. (1962), Vol. II, p. 97.

APPENDIX B.1 $q - \gamma$ Plane of Integrations

(1.1)

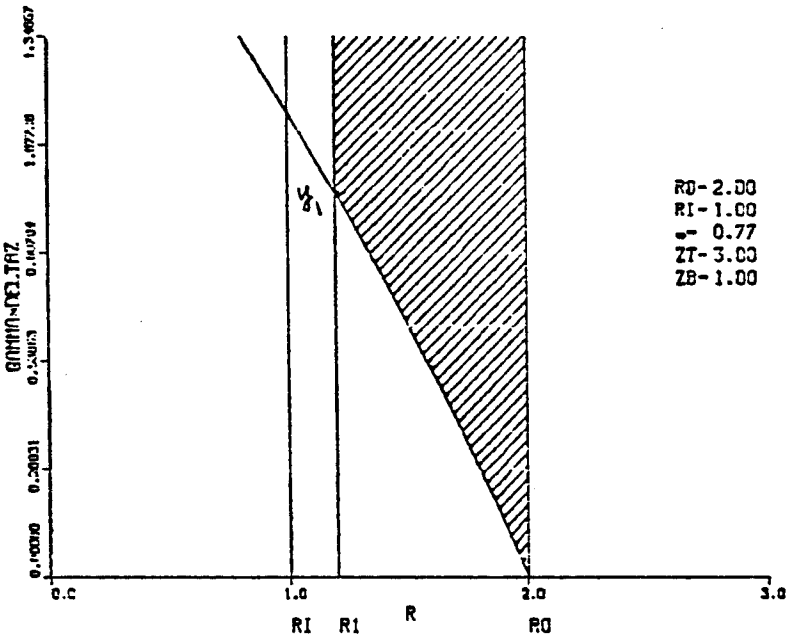

INTEGRAL LIMIT - 1.1 -
GAMMA
0.0001.157 2.314 3.471 4.628 5.785 6.942 8.099 9.256 10.413 11.570
RO-2.00
RI-1.00
e- 0.77
ZT-3.00
ZB-1.00
0.0
1.0
2.0
3.0
4.0
ZB
Z
ZT

(2.1)


INTEGRAL LIMIT -2.1-
GAMMA
RO-2.00
RI-1.00
e- 0.77
ZT-3.00
ZB-1.00
0.0
1.0
2.0
3.0
4.0
ZB
Z
ZT

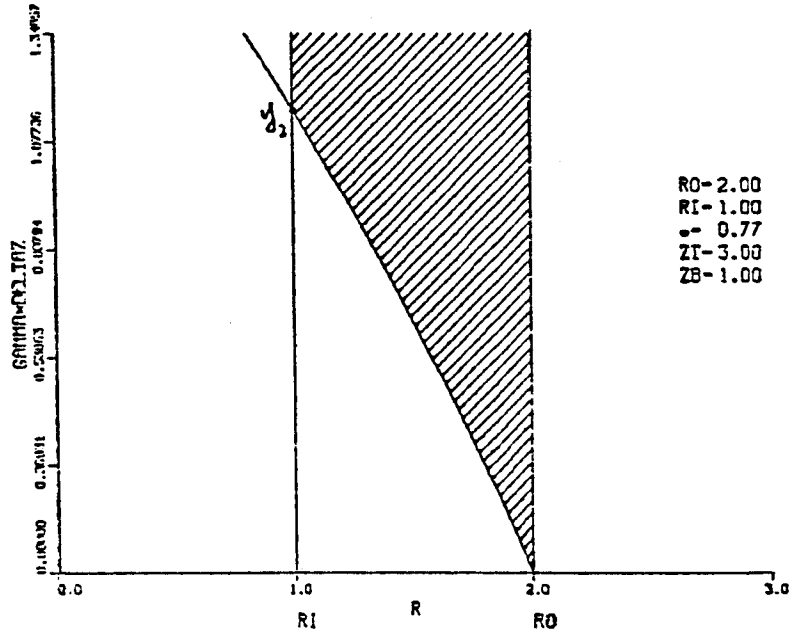
(3A.1)

INTEGRAL LIMIT - 3A.1 -



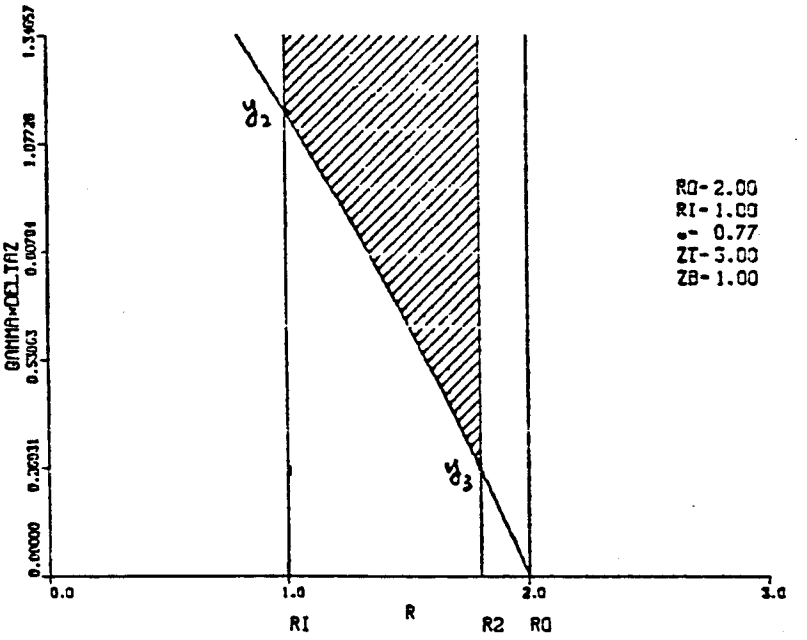
(3A.2)

INTEGRAL LIMIT - 3A.2 -



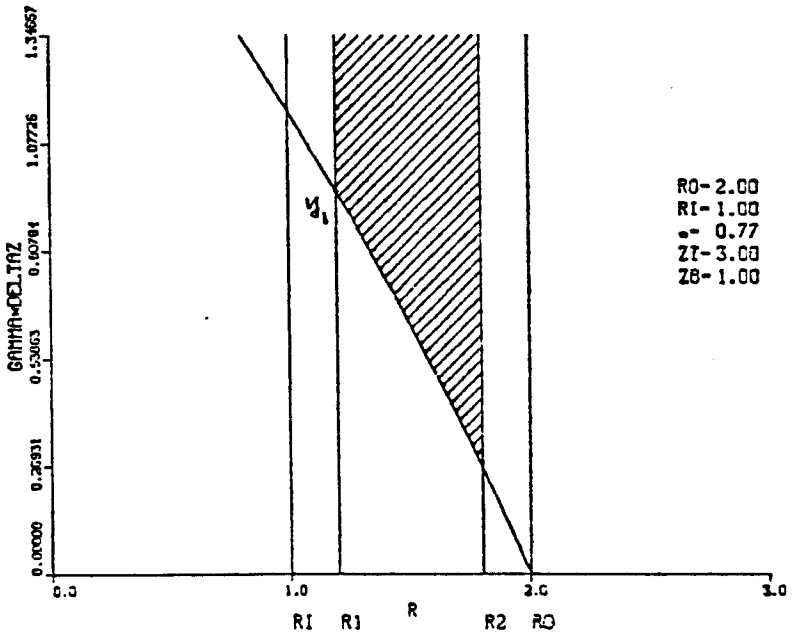
(3A.3)

INTEGRAL LIMIT - 3A.3 -

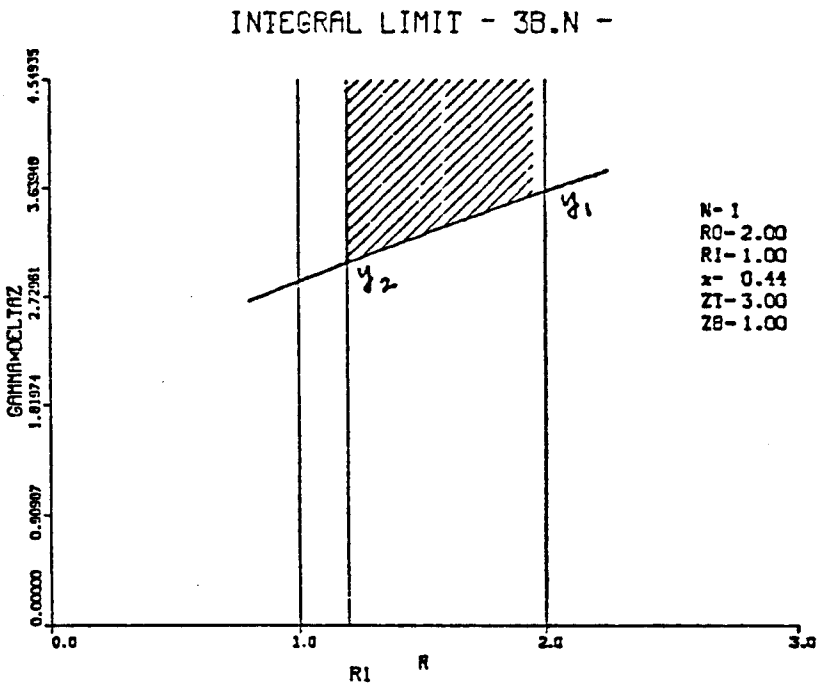


(3A.4)

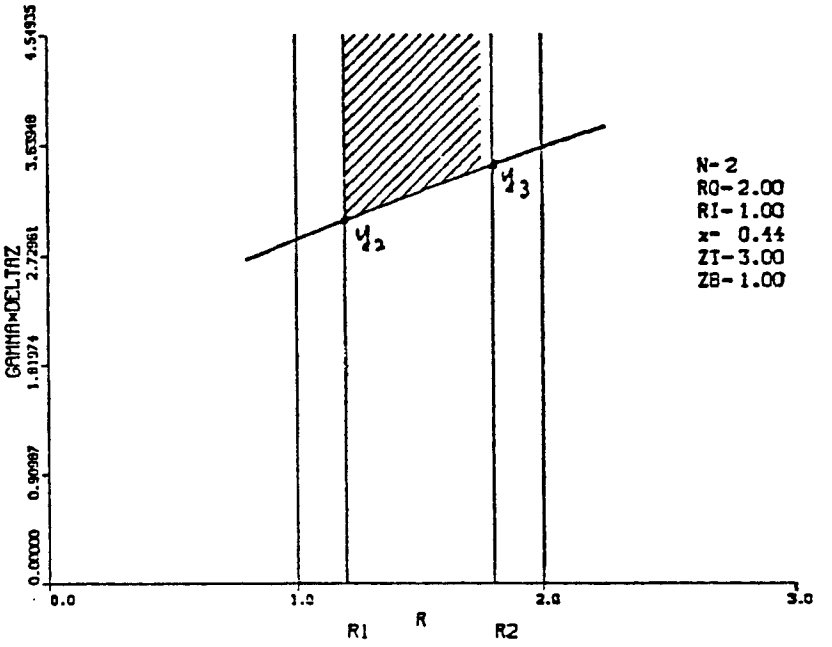
INTEGRAL LIMIT - 3A.4 -



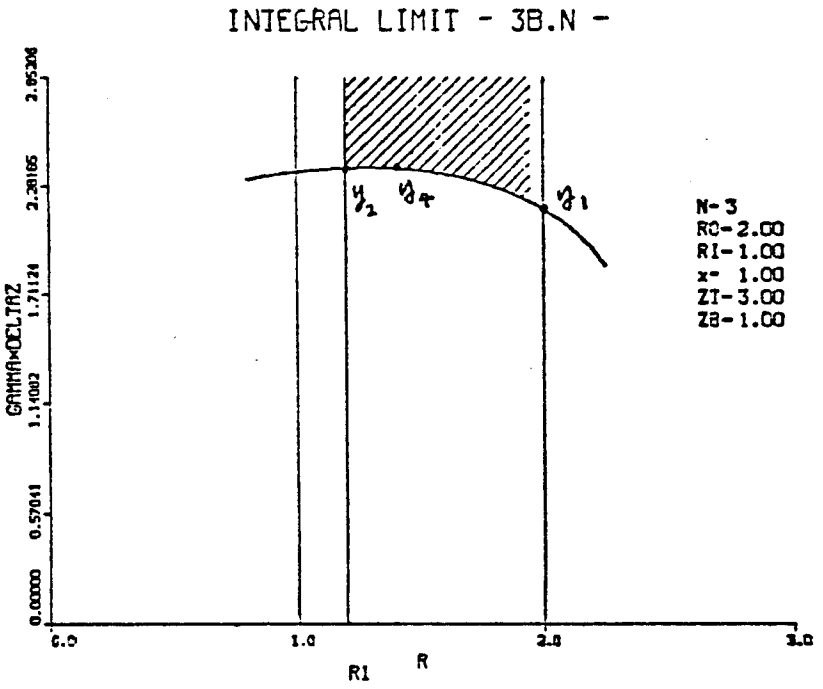
(3B.1)


INTEGRAL LIMIT - 3B.N -
GAMMA*DELTAZ
0.00000
0.90907
1.81974
2.72961
3.63948
4.54935
0.0
1.0
2.0
3.0
R1
R
γ1
γ2
N-1
RO-2.00
RI-1.00
x- 0.44
ZT-3.00
ZB-1.00

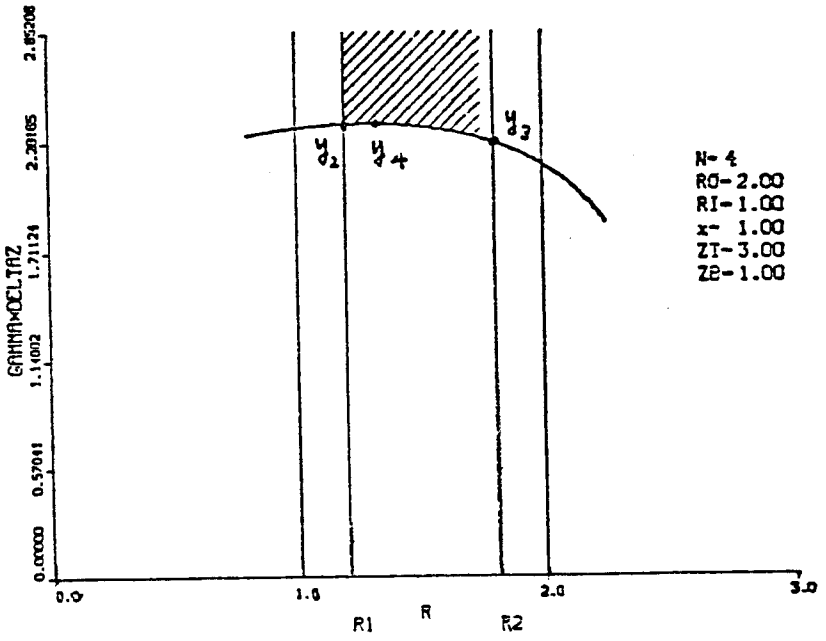
(3B.2)


INTEGRAL LIMIT - 3B.N -
GAMMA*DELTAZ
0.00000
0.90907
1.81974
2.72961
3.63948
4.54935
0.0
1.0
2.0
3.0
R1
R
R2
γ2
γ3
N-2
RO-2.00
RI-1.00
x- 0.44
ZT-3.00
ZB-1.00

(3B.3)

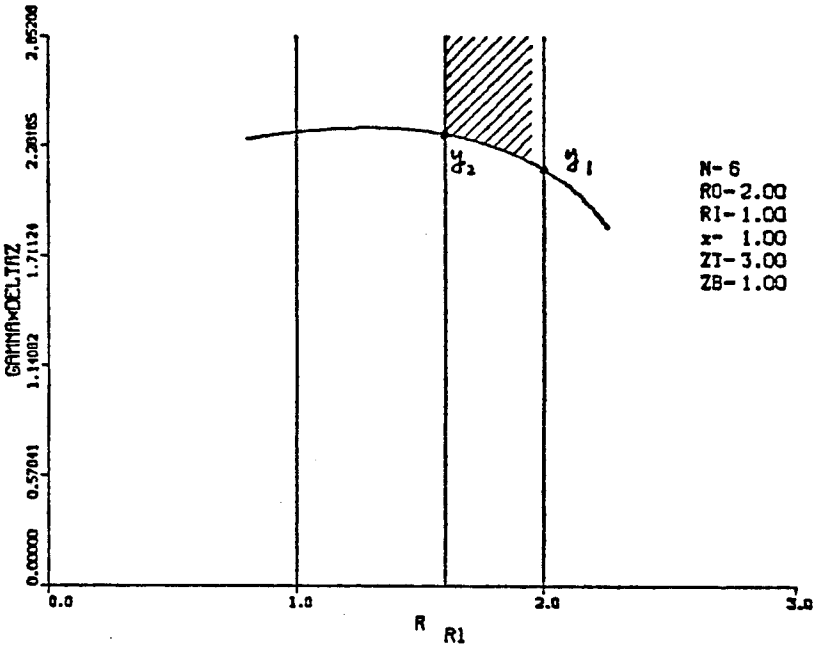

INTEGRAL LIMIT - 3B.N -
GAMMA-DELTAZ
0.00000
0.57041
1.14082
1.71124
2.28165
2.85206
0.0
1.0
2.0
3.0
R1
R
N-3
RO-2.00
RI-1.00
x- 1.00
ZI-3.00
ZB-1.00

(3B.4)


INTEGRAL LIMIT - 3B.N -
GAMMA-DELTAZ
0.00000
0.57041
1.14082
1.71124
2.28165
2.85206
0.0
1.0
2.0
3.0
R1
R
R2
N-4
RO-2.00
RI-1.00
x- 1.00
ZI-3.00
ZB-1.00

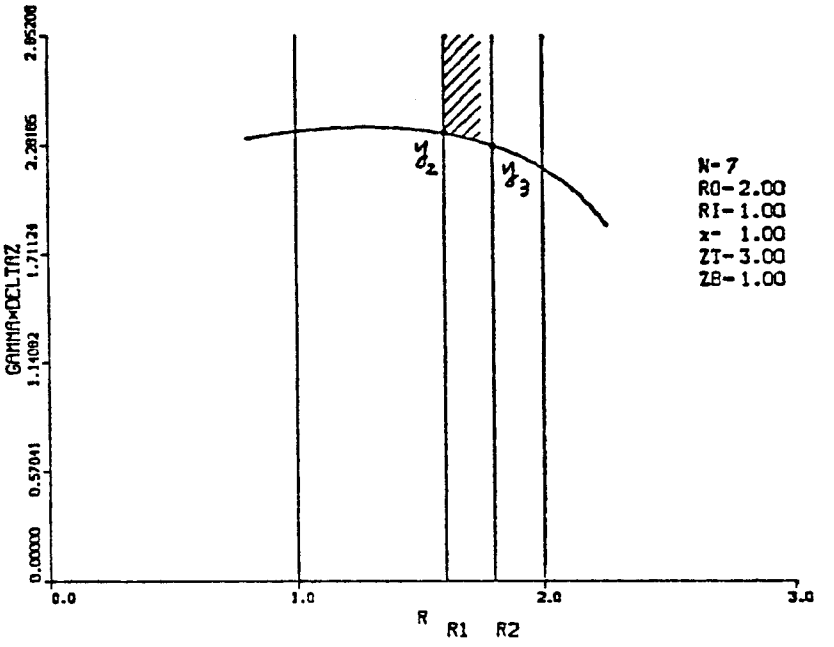
(3B.6)

INTEGRAL LIMIT - 3B.N -



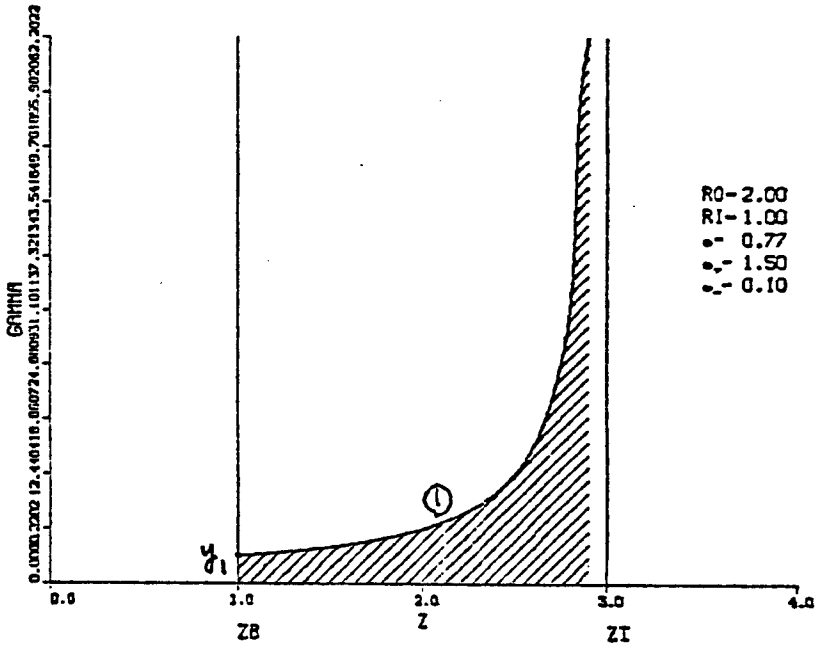
(3B.7)

INTEGRAL LIMIT - 3B.N -



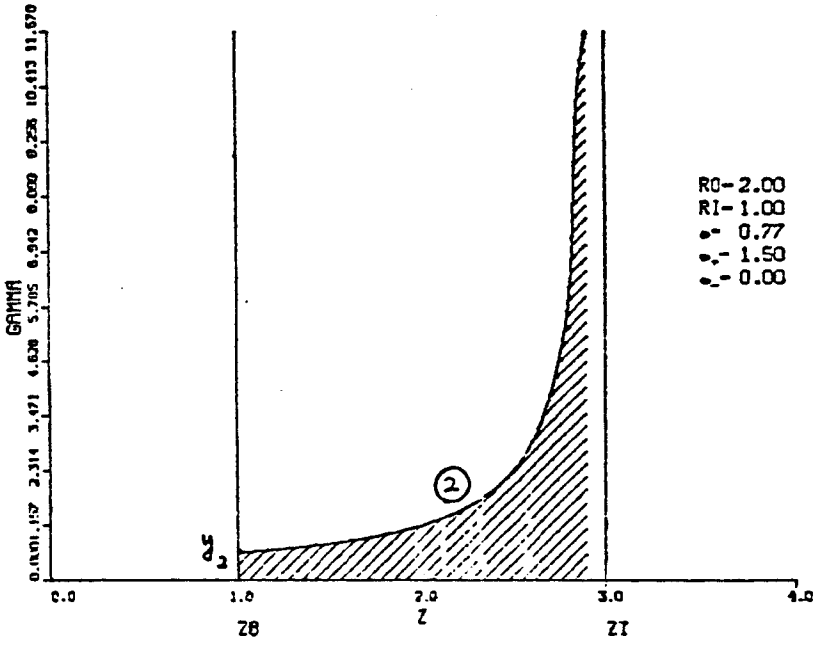
(4.1)

INTEGRAL LIMIT - 4.1 -



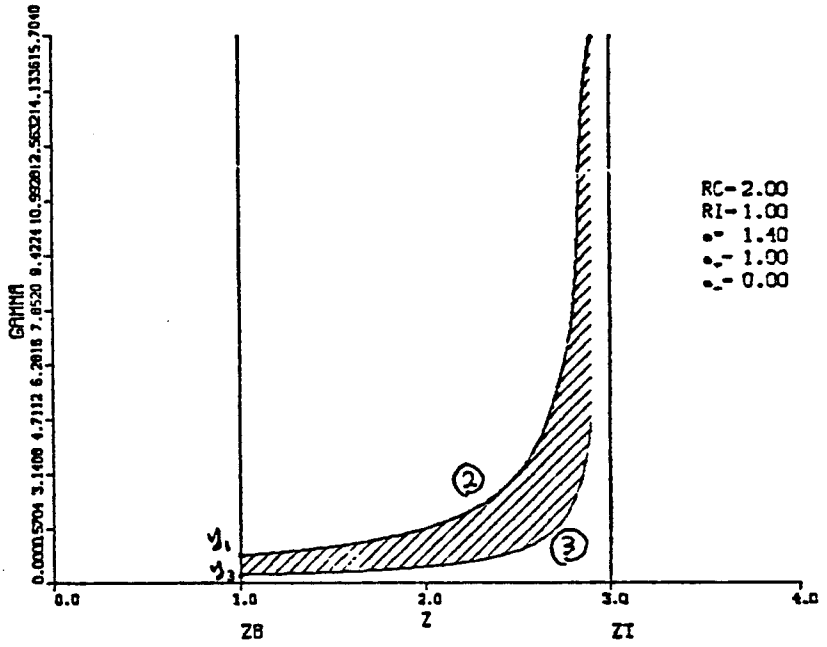
(4.2)

INTEGRAL LIMIT - 4.2 -



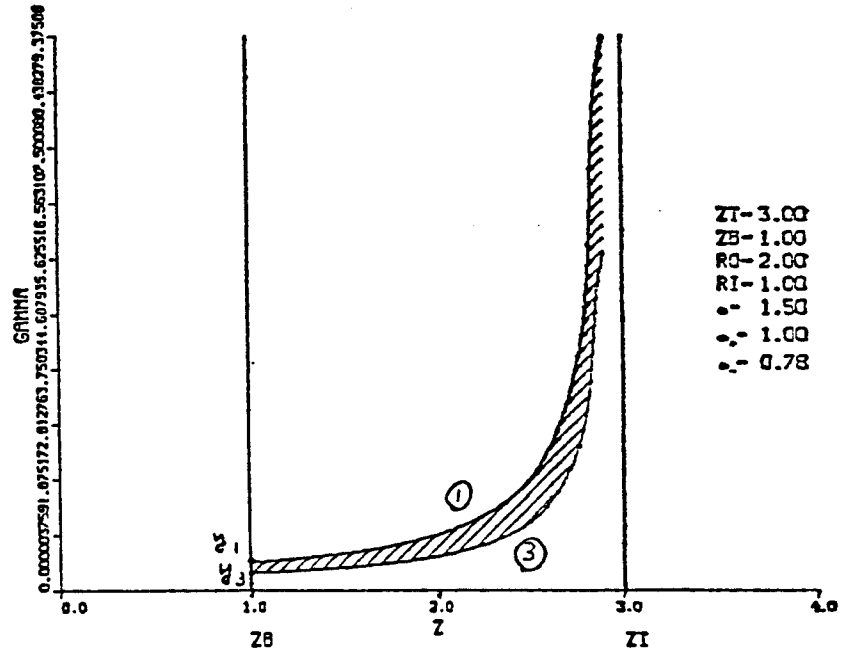
(4.3)

INTEGRAL LIMIT - 4.3 -



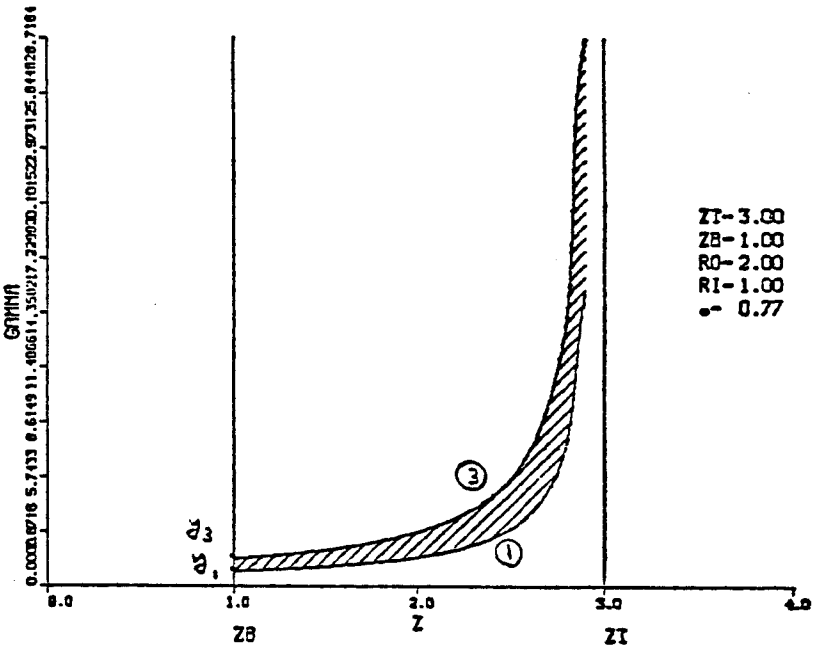
(4.4)

INTEGRAL LIMIT - 4.4 -



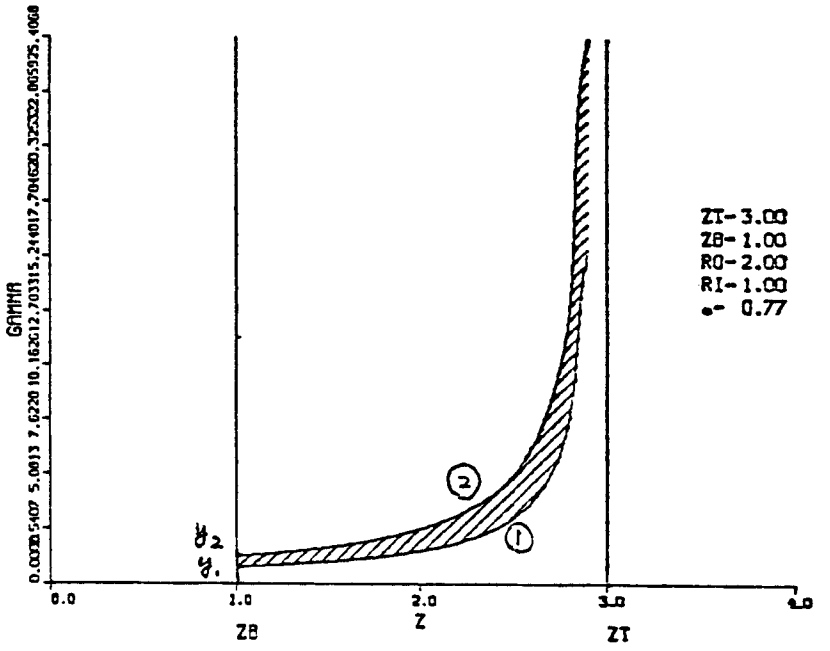
(5.1)

INTEGRAL LIMIT - 5.1 -



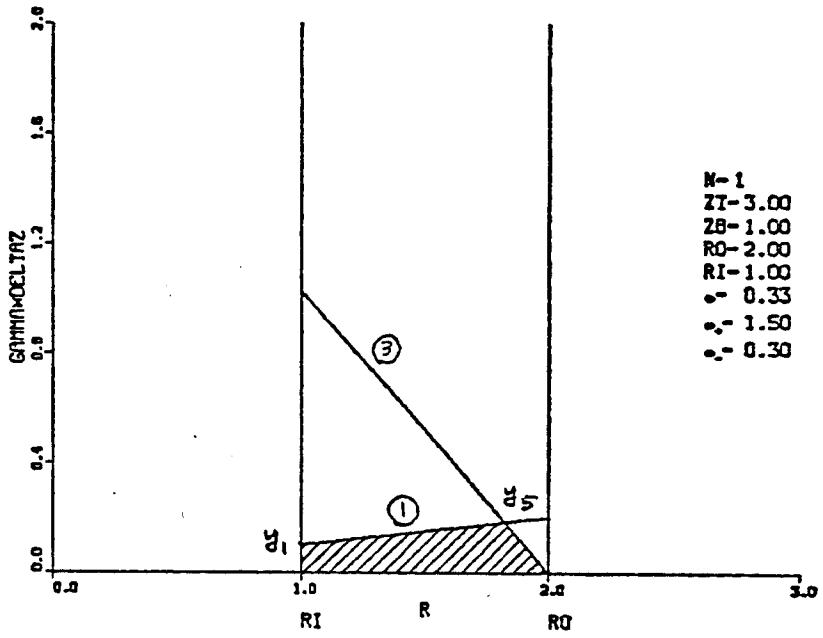
(5.2)

INTEGRAL LIMIT - 5.2 -



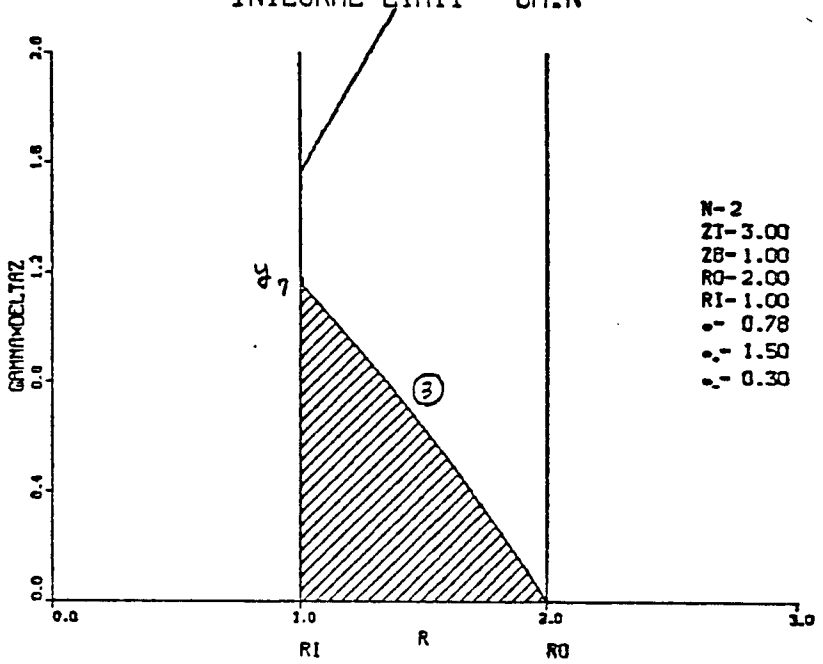
(6A.1)

INTEGRAL LIMIT - 6A.N -



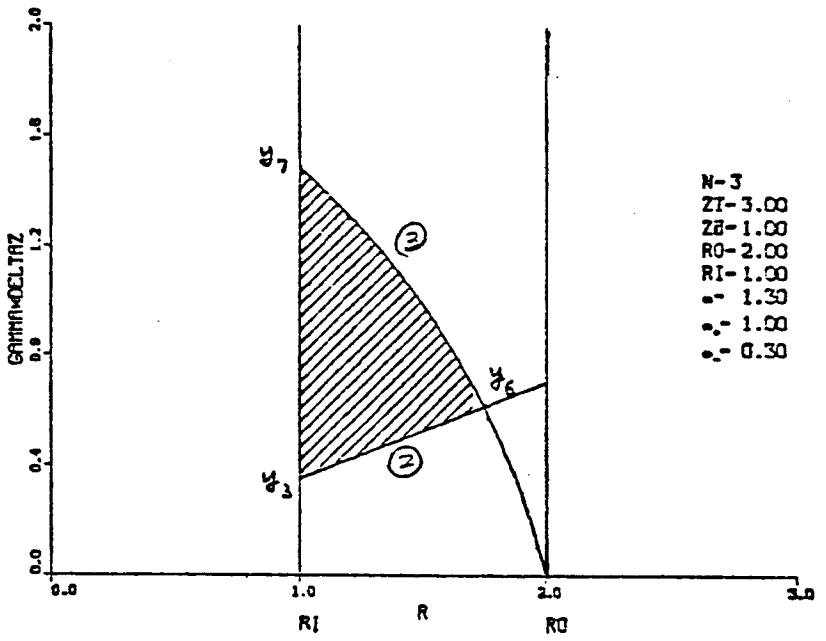
(6A.2)

INTEGRAL LIMIT - 6A.N -



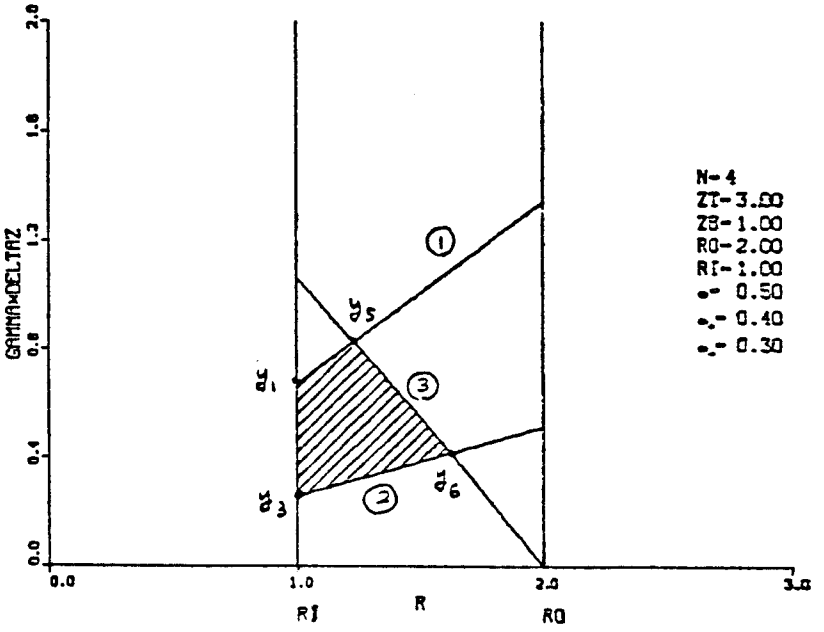
(6A.3)

INTEGRAL LIMIT - 6A.N -



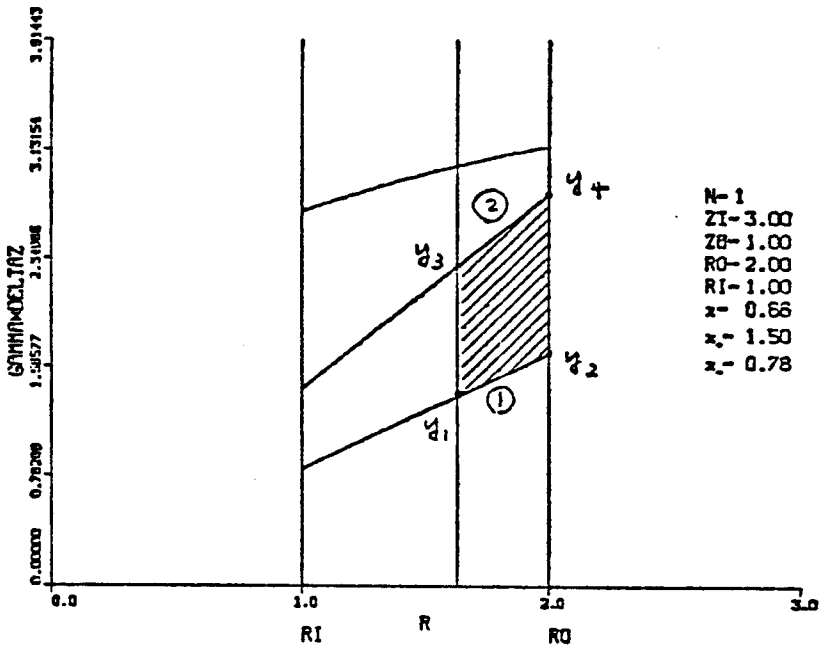
(6A.4)

INTEGRAL LIMIT - 6A.N -



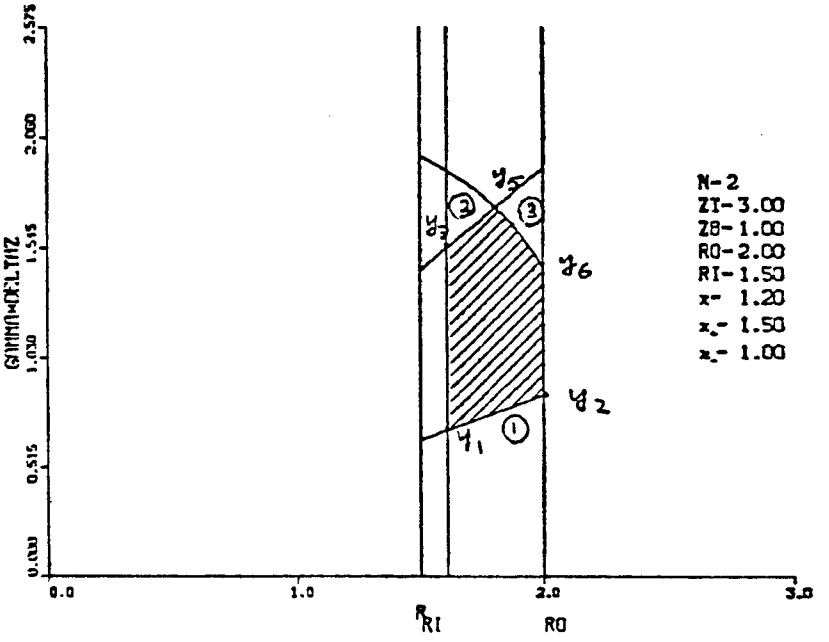
(6B.1)

INTEGRAL LIMIT - 6B.N -

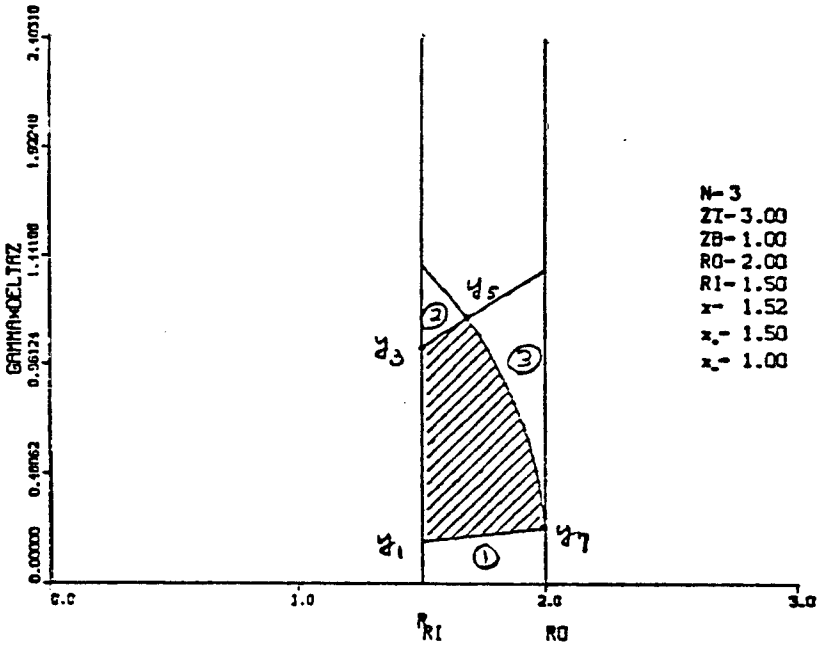


(6B.2)

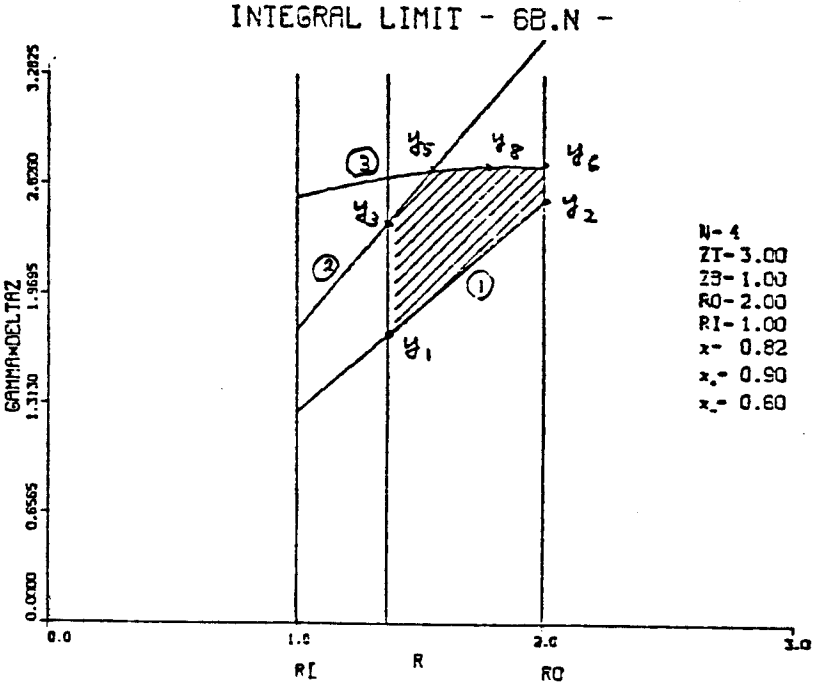
INTEGRAL LIMIT - 6B.N -



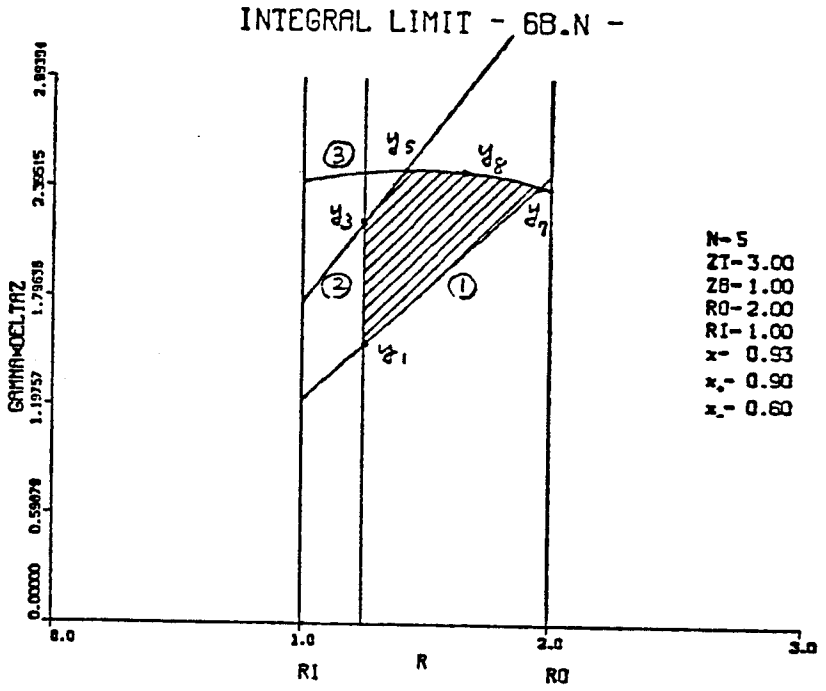
(6B.3)


INTEGRAL LIMIT - 6B.N -
GAMMA*DELTAZ
0.00000
0.46863
0.93724
1.40586
1.87448
2.34310
0.0
1.0
2.0
3.0
RI
R
RO
N-3
ZT-3.00
ZB-1.00
RO-2.00
RI-1.50
x- 1.52
x+ 1.50
x- 1.00

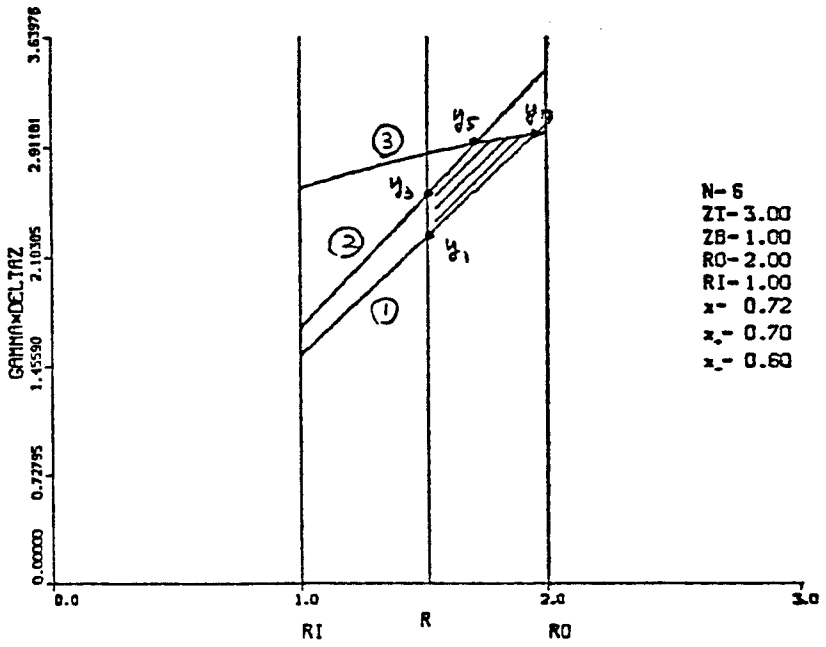
(6B.4)


INTEGRAL LIMIT - 6B.N -
GAMMA*DELTAZ
0.0000
0.6565
1.3130
1.9695
2.6260
3.2825
0.0
1.0
2.0
3.0
RI
R
RO
N-4
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.82
x+ 0.90
x- 0.60

(6B.5)

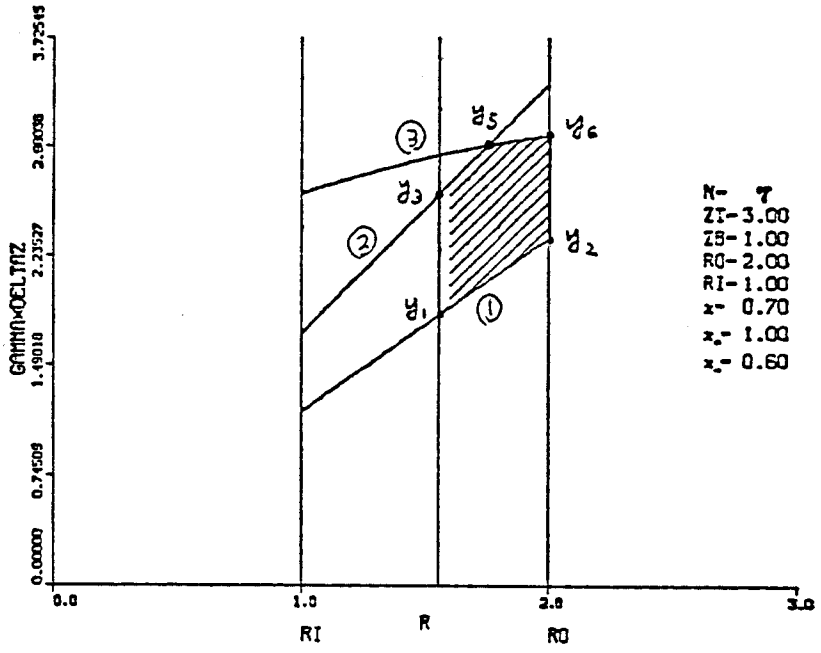

INTEGRAL LIMIT - 6B.N -
GAMMA=DELTAZ
0.00000
0.59079
1.19757
1.79638
2.39515
2.99396
0.0
1.0
2.0
3.0
RI
R
RO
N-5
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.93
x.- 0.90
x_- 0.60

(6B.6)


INTEGRAL LIMIT - 6B.N -
GAMMA=DELTAZ
0.00000
0.72795
1.45590
2.18385
2.91181
3.63976
0.0
1.0
2.0
3.0
RI
R
RO
N-6
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.72
x.- 0.70
x_- 0.60

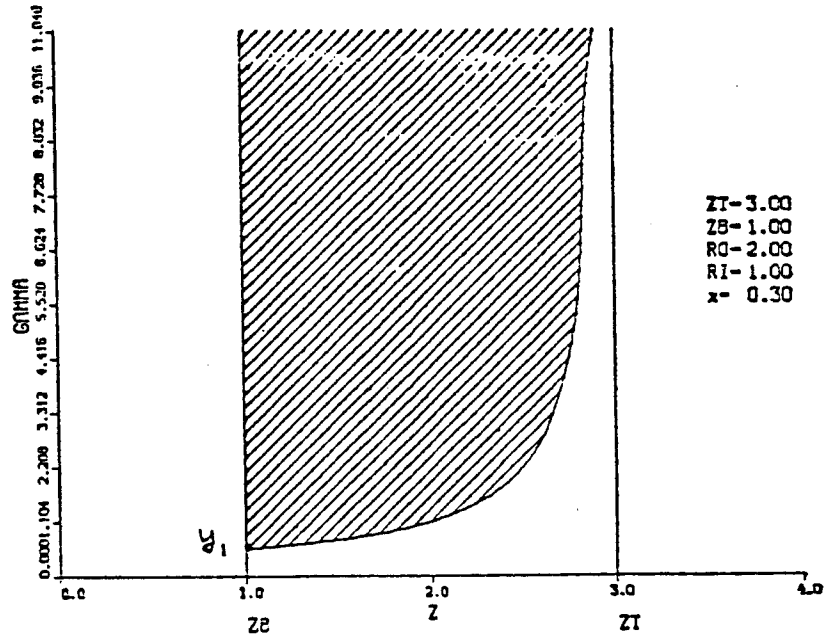
(6B.7)

INTEGRAL LIMIT - 6B.N -

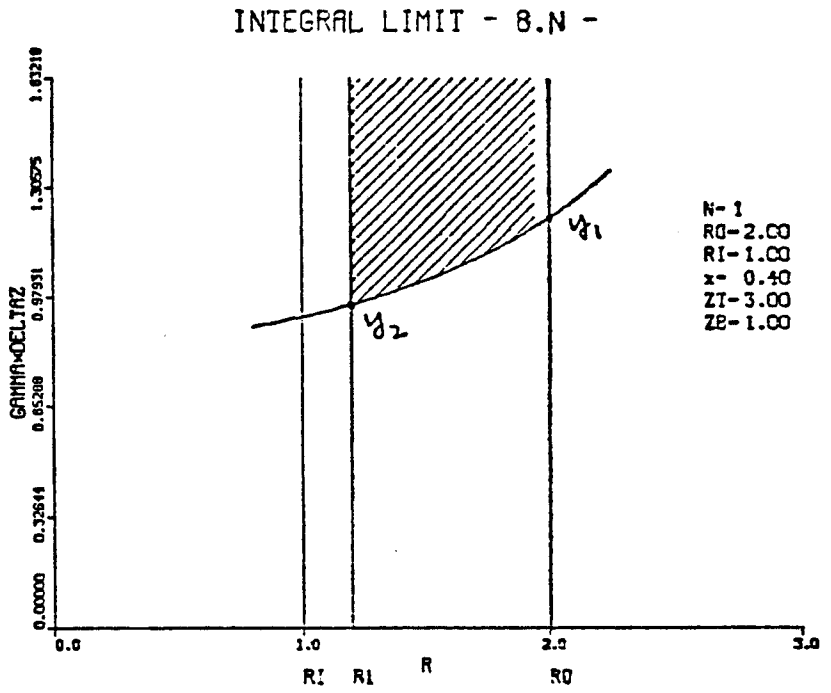


(7.1)

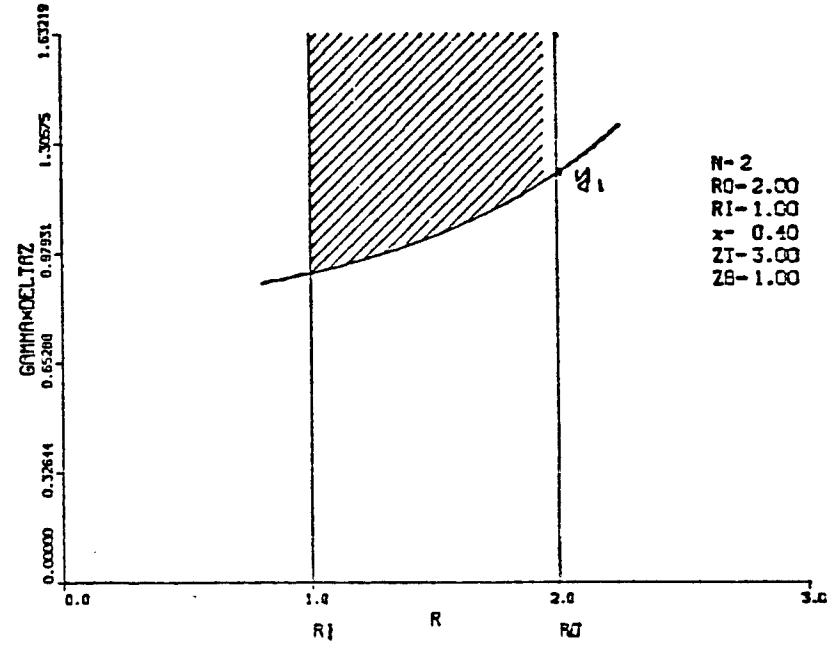
INTEGRAL LIMIT - 7.1 -



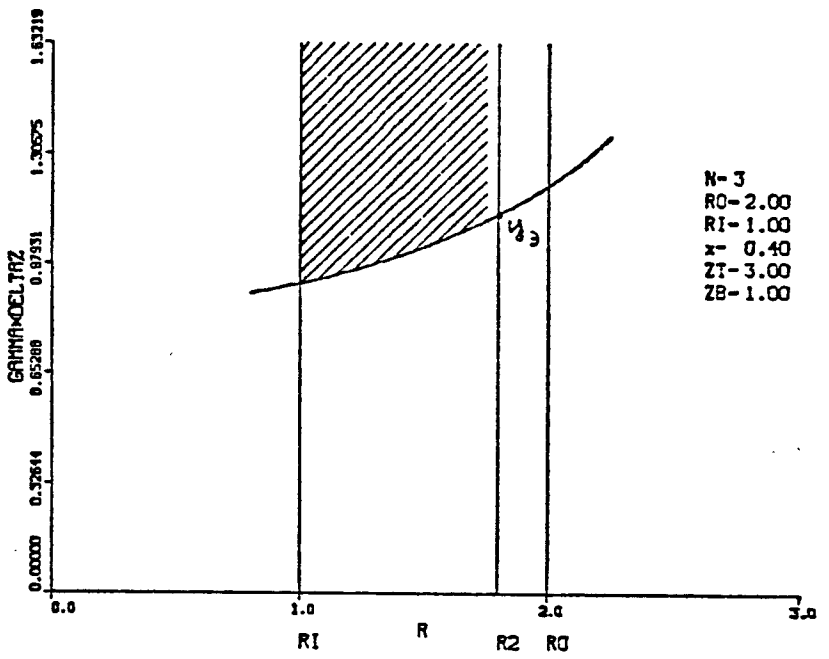
(8.1)


INTEGRAL LIMIT - 8.N -
GAMMA*DELTAZ
0.00000
0.32644
0.65288
0.97931
1.30575
1.63219
0.0
1.0
2.0
3.0
RI
R1
R
R0
N-1
R0-2.00
RI-1.00
x- 0.40
ZI-3.00
Z8-1.00

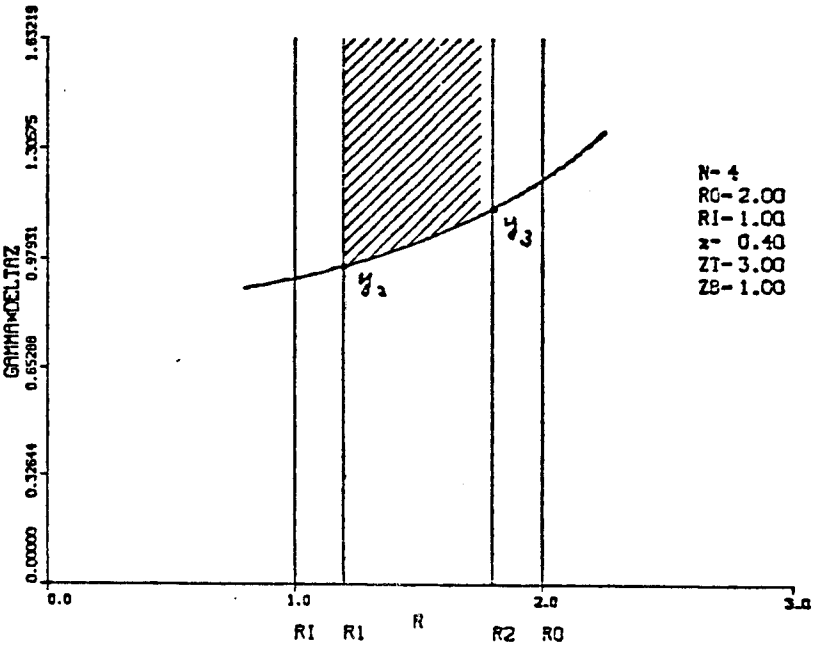
(8.2)


INTEGRAL LIMIT - 8.N -
GAMMA*DELTAZ
0.00000
0.32644
0.65288
0.97931
1.30575
1.63219
0.0
1.0
2.0
3.0
R1
R
R0
N-2
R0-2.00
RI-1.00
x- 0.40
ZI-3.00
Z8-1.00

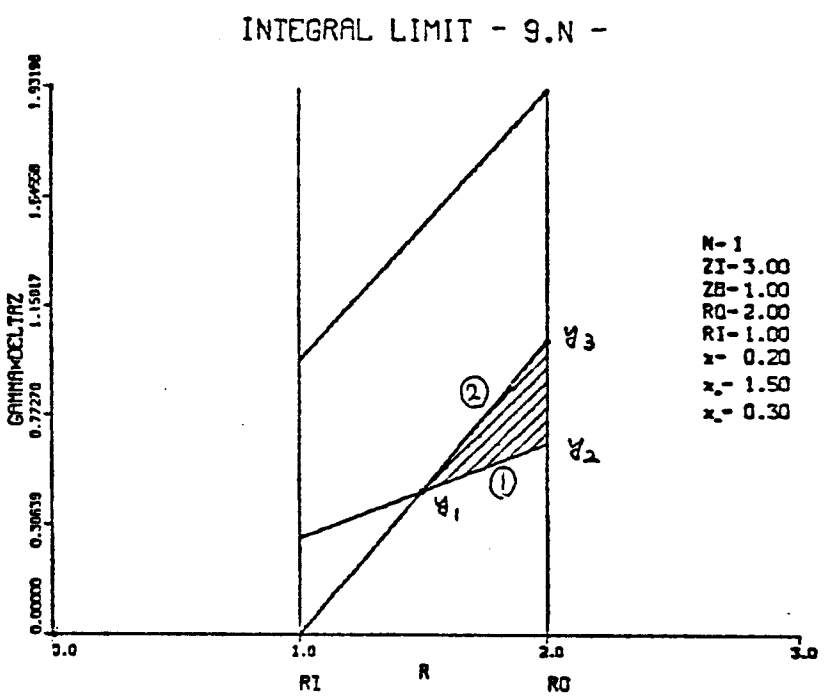
(8.3)


INTEGRAL LIMIT - 8.N -
GAMMA*DELTAZ
0.00000
0.32644
0.65288
0.97931
1.30575
1.63219
0.0
1.0
2.0
3.0
R
RI
R2
RO
N-3
RO-2.00
RI-1.00
x- 0.40
ZT-3.00
ZB-1.00

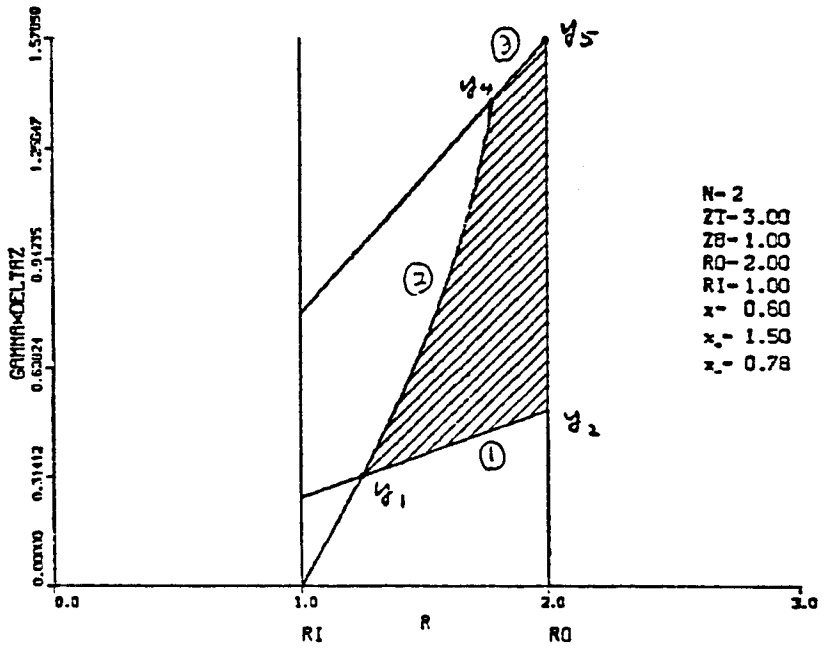
(8.4)


INTEGRAL LIMIT - 8.N -
GAMMA*DELTAZ
0.00000
0.32644
0.65288
0.97931
1.30575
1.63219
0.0
1.0
2.0
3.0
R
RI
R1
R2
RO
N-4
RO-2.00
RI-1.00
x- 0.40
ZT-3.00
ZB-1.00

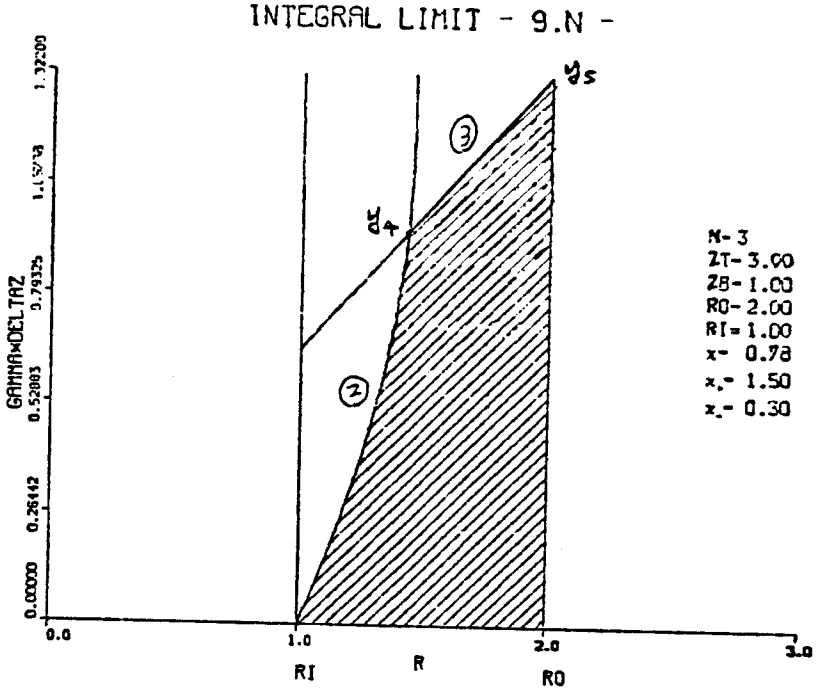
(9.1)


INTEGRAL LIMIT - 9.N -
GAMMA<OCLTAZ
0.00000
0.30639
0.77270
1.15817
1.54528
1.93198
0.0
1.0
2.0
3.0
RI
R
RO
N-1
ZI-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.20
x.- 1.50
x_- 0.30
γ3
γ2
γ1
①
②

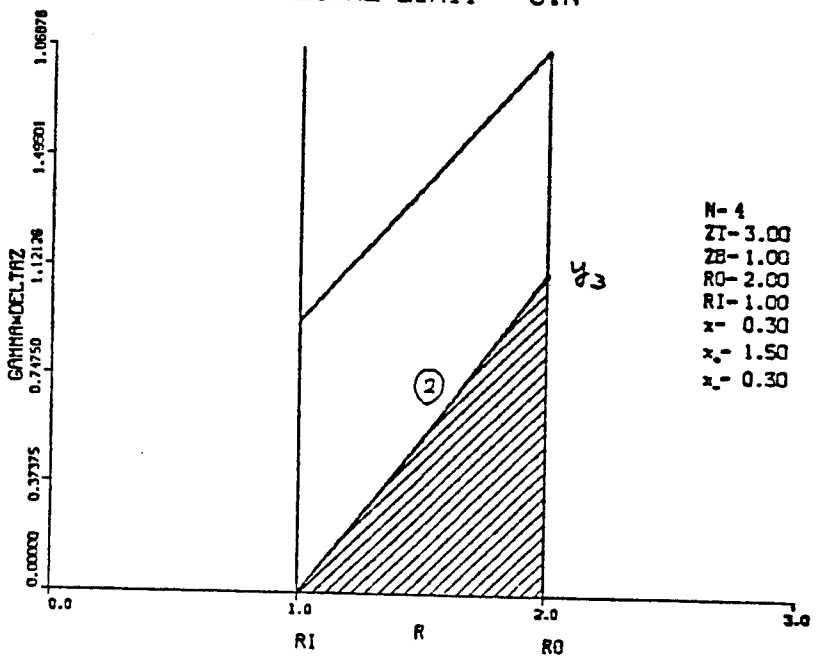
(9.2)


INTEGRAL LIMIT - 9.N -
GAMMA<OCLTAZ
0.00000
0.31412
0.63024
0.91295
1.25047
1.57050
0.0
1.0
2.0
3.0
RI
R
RO
N-2
ZI-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.60
x.- 1.50
x_- 0.78
γ5
γ4
γ2
γ1
①
②
③

(9.3)

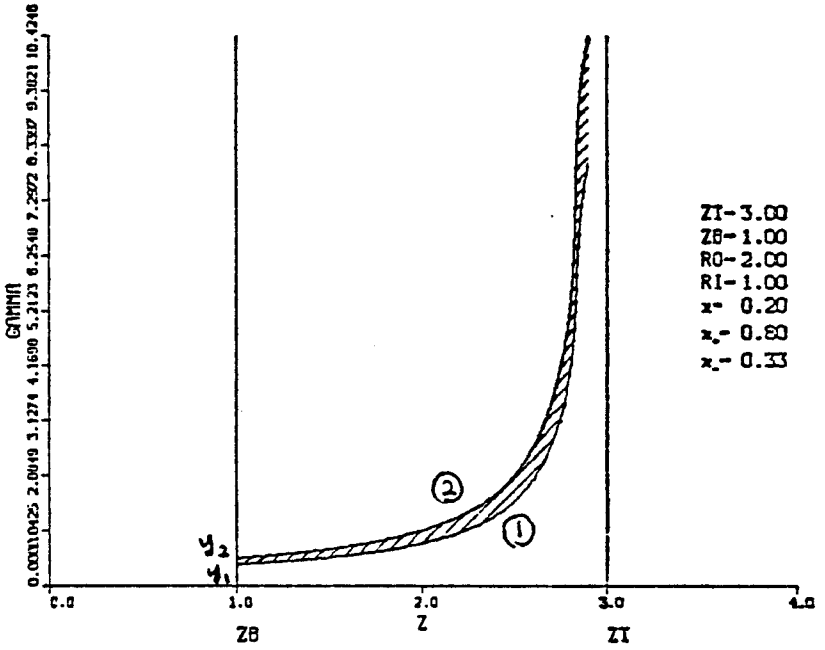

INTEGRAL LIMIT - 9.N -
GAMMA=DELTAZ
0.00000
0.26442
0.52883
0.79325
1.32208
0.0
1.0
2.0
3.0
RI
R
RO
③
②
y_4
y_5
N-3
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.78
x_1- 1.50
x_2- 0.30

(9.4)


INTEGRAL LIMIT - 9.N -
GAMMA=DELTAZ
0.00000
0.37375
0.74750
1.12126
1.49501
1.86878
0.0
1.0
2.0
3.0
RI
R
RO
②
y_3
N-4
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.30
x_1- 1.50
x_2- 0.30

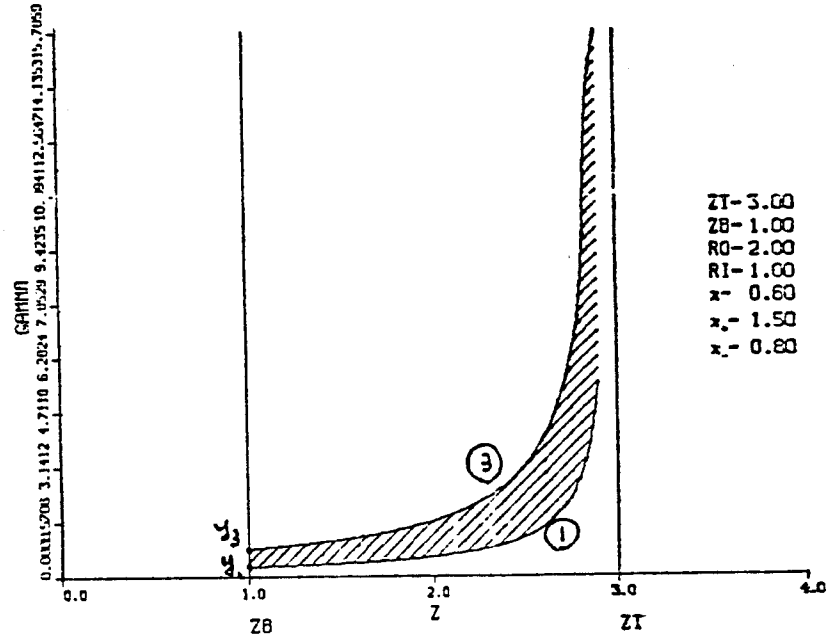
(10.1)

INTEGRAL LIMIT - 10.1

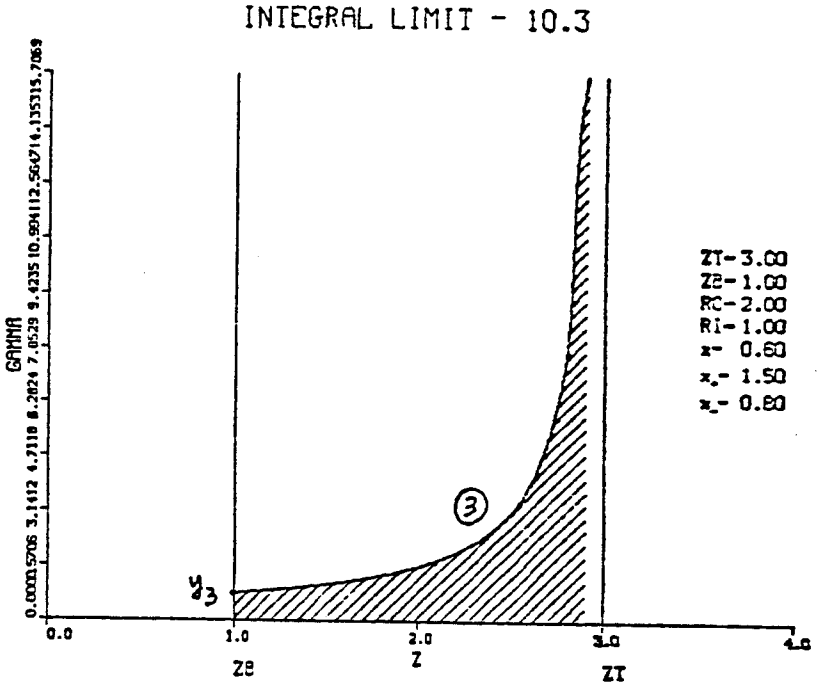


(10.2)

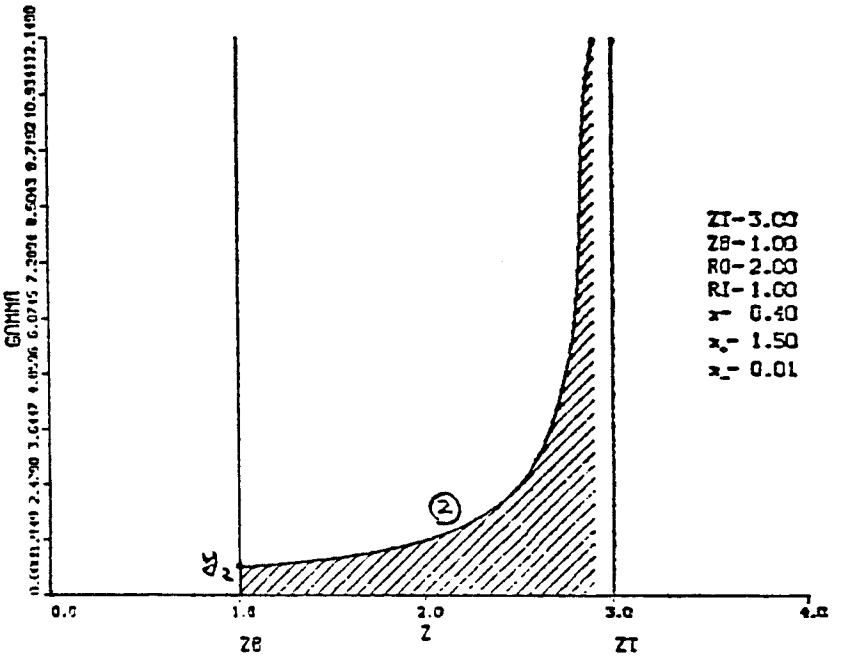
INTEGRAL LIMIT - 10.2



(10.3)


INTEGRAL LIMIT - 10.3
GAMMA
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.60
x+- 1.50
x_- 0.80
③
0.0
1.0
2.0
3.0
4.0
ZB
Z
ZT

(10.4)


INTEGRAL LIMIT - 10.4
GAMMA
ZT-3.00
ZB-1.00
RO-2.00
RI-1.00
x- 0.40
x+- 1.50
x_- 0.01
②
0.0
1.0
2.0
3.0
4.0
ZB
Z
ZT

APPENDIX B.2 Surface Functions of Integral Domains

[1] $$\gamma = \frac{\sqrt{r_o^2 - (r_I \sin\omega')^2} - r_I \cos\omega'}{z_T - z'}$$

[2] $$\gamma = \frac{2r_o \Delta z \cos\chi'}{z_T - z'}$$

[3A] $$\gamma\Delta z = \sqrt{r_o^2 - (r'\sin\omega')^2} - r'\cos\omega'$$

$$r_1 = r_o \frac{\sin\omega_-}{\sin\omega'}$$

$$r_2 = r_o \frac{\sin\omega_+}{\sin\omega'}$$

[3B] $$\gamma\Delta z = \sqrt{r_o^2 - (r'\sin\chi')^2} + r'\cos\chi'$$

$$r_1 = r_o \frac{\sin\chi_-}{\sin\chi'}$$

$$r_2 = r_o \frac{\sin\chi_-}{\sin\chi'}$$

[4] (1) $$\gamma = \frac{r_I \sin(\omega' - \omega_-)}{\sin \omega_- (z_T - z')}$$

(2) $$\gamma = \frac{\sqrt{r_o^2 - (r_I \sin \omega')^2} - r_I \cos \omega'}{z_T - z'}$$

(3) $$\gamma = \frac{r_I \sin(\omega' - \omega_+)}{\sin \omega_+ (z_T - z')}$$

[5] (1) $$\gamma = \frac{r_o \sin(\chi' + \omega_+)}{\sin \omega_+ (z_T - z')}$$

(2) $$\gamma = \frac{r_o \sin(\chi' + \omega_-)}{\sin \omega_- (z_T - z')}$$

(3) $$\gamma = \frac{2 r_o \cos \chi'}{z_T - z'}$$

[6A] (1) $$\gamma \Delta z = r' \frac{\sin(\omega' - \omega_-)}{\sin \omega_-}$$

(2) $$\gamma \Delta z = r' \frac{\sin(\omega' - \omega_+)}{\sin \omega_+}$$

(3) $$\gamma \Delta z = \sqrt{r_o^2 - (r' \sin \omega')^2} - r' \cos \omega'$$

[6B] (1) $\gamma\Delta z = r' \dfrac{\sin(\chi' + \omega_+)}{\sin \omega_+}$

(2) $\gamma\Delta z = r' \dfrac{\sin(\chi' + \omega_-)}{\sin \omega_-}$

(3) $\gamma\Delta z = \sqrt{r_o^2 - (r'\sin \chi')^2} + r'\cos \chi'$

$$r_1 = r_I / \sin \chi'$$

[7] $$\gamma = \frac{r_o \cos \chi' - \sqrt{r_I^2 - (r_o \sin \chi')^2}}{z_T - z'}$$

[8] $$\gamma\Delta z = r'\cos \chi' - \sqrt{r_I^2 - (r'\sin \chi')^2}$$

$$r_1 = r_I \frac{\sin \chi_-}{\sin \chi'}$$

$$r_2 = r_I \frac{\sin \chi_+}{\sin \chi'}$$

[9] (1) $$\gamma\Delta z = \frac{r'\sin(\omega_- - \chi')}{\sin \omega_-}$$

(2) $$\gamma\Delta z = r'\cos \chi' - \sqrt{r_I^2 - (r'\sin \chi')^2}$$

(3) $$\gamma\Delta z = \frac{r'\sin(\omega_+ - \chi')}{\sin \omega_+}$$

[10] (1) $$\gamma = \frac{r_o\sin(\omega_- - \chi')}{\sin \omega_-(z_T - z')}$$

(2) $$\gamma = \frac{r_o\cos \chi' - \sqrt{r_I^2 - (r_o\sin \chi')^2}}{z_T - z'}$$

(3) $$\gamma = \frac{r_o\sin(\omega_+ - \chi')}{\sin \omega_+(z_T - z')}$$

APPENDIX B.3 Functions for Intersections

[1] $$y_1 = \sqrt{r_o^2 - (r_I \sin \omega)^2} - r_I \cos \omega$$

[2] $$y_1 = 2r_o \cos \omega$$

[3A] $$y_1 = r_o \sin(\omega - \omega_-)/\sin \omega$$
$$y_2 = \sqrt{r_o^2 - (r_I \sin \omega)^2} - r_I \cos \omega$$
$$y_3 = r_o \sin(\omega - \omega_+)/\sin \omega$$

[3B] $$y_1 = 2r_o \cos \chi$$
$$y_2 = r_o \sin(\chi + \omega_-)/\sin \chi$$
$$y_3 = r_o \sin(\chi + \omega_+)/\sin \chi$$
$$y_4 = r_o/\sin \chi$$

[4] $$y_1 = r_I \sin(\omega - \omega_-)/\sin \omega_-$$
$$y_2 = \sqrt{r_o^2 - (r_I \sin\omega)^2} - r_I \cos \omega$$
$$y_3 = r_I \sin(\omega - \omega_+)/\sin \omega_+$$

[5] $y_1 = r_o \sin(\chi + \omega_+) / \sin \omega_+$

$y_2 = r_o \sin(\chi + \omega_-) / \sin \omega_-$

$y_3 = 2 r_o \cos \chi$

[6A] $y_1 = r_I \sin(\omega - \omega_-) / \sin \omega_-$

$y_2 = r_o \sin(\omega - \omega_-) / \sin \omega_-$

$y_3 = r_I \sin(\omega - \omega_+) / \sin \omega_+$

$y_4 = r_o \sin(\omega - \omega_+) / \sin \omega_+$

$y_5 = r_o \sin(\omega - \omega_-) / \sin \omega$

$y_6 = r_o \sin(\omega - \omega_+) / \sin \omega$

$y_7 = \sqrt{r_o^2 - (r_I \sin \omega)^2} - r_I \cos \omega$

[6B] $y_1 = r_I \sin(\chi + \omega_+)/\sin \omega_+ \sin \chi$

$y_2 = r_o \sin(\chi + \omega_+)/\sin \omega_+$

$y_3 = r_I \sin(\chi + \hat{\omega}_-)/\sin \hat{\omega}_- \sin \chi$

$y_4 = r_o \sin(\chi + \hat{\omega}_-)/\sin \hat{\omega}_-$

$y_5 = r_o \sin(\chi + \hat{\omega}_-)/\sin \chi$

$y_6 = 2 r_o \cos \chi$

$y_7 = r_o \sin(\chi + \omega_+)/\sin \chi$

$y_8 = r_o/\sin \chi$

[7] $y_1 = r_o \cos \chi - \sqrt{r_I^2 - (r_o \sin \chi)^2}$

[8] $y_1 = r_o \cos \chi - \sqrt{r_I^2 - (r_o \sin \chi)^2}$

$y_2 = r_I \sin(\omega_- - \chi)/\sin \chi$

$y_3 = r_I \sin(\omega_+ - \chi)/\sin \chi$

[9]

$$y_1 = r_I \sin(\omega_- - \chi)/\sin\chi$$

$$y_2 = r_o \sin(\omega_- - \chi)/\sin\omega_-$$

$$y_3 = r_o \cos\chi - \sqrt{r_I^2 - (r_o \sin\chi)^2}$$

$$y_4 = r_I \sin(\omega_+ - \chi)/\sin\chi$$

$$y_5 = r_o \sin(\omega_+ - \chi)/\sin\omega_+$$

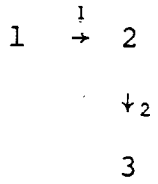
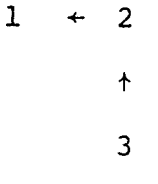
[10]

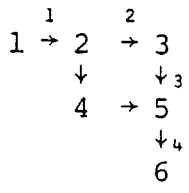
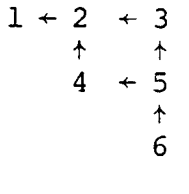
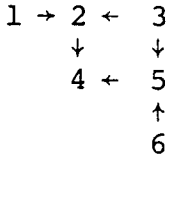
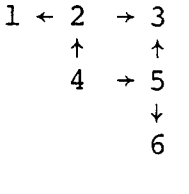
$$y_1 = r_o \sin(\omega_- - \chi)/\sin\omega_-$$

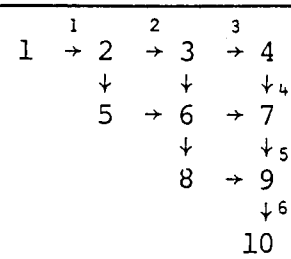
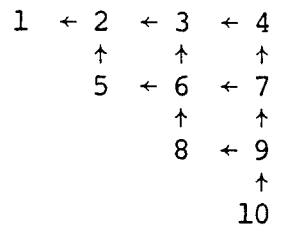
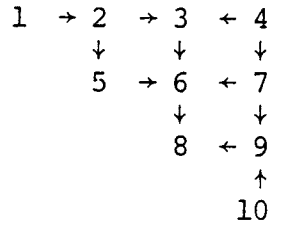
$$y_2 = r_o \cos\chi - \sqrt{r_I^2 - (r_o \sin\chi)^2}$$

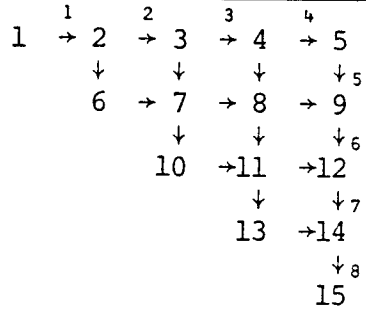
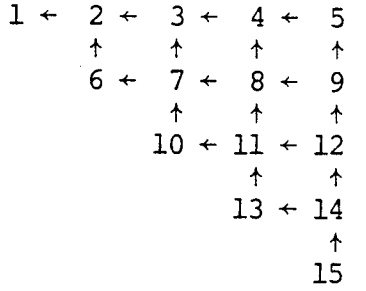
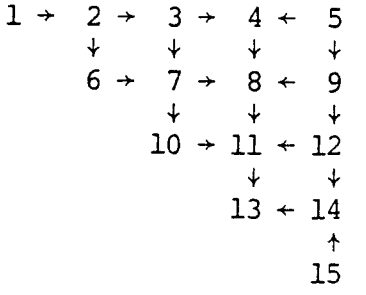
$$y_3 = r_o \sin(\omega_+ - \chi)/\sin\omega_+$$

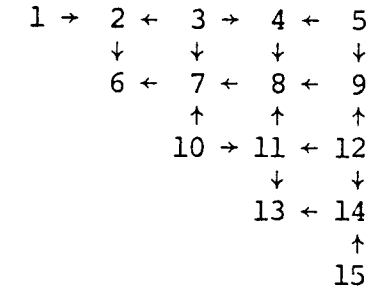
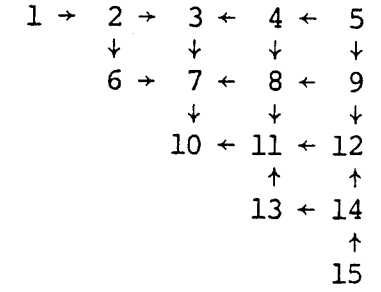
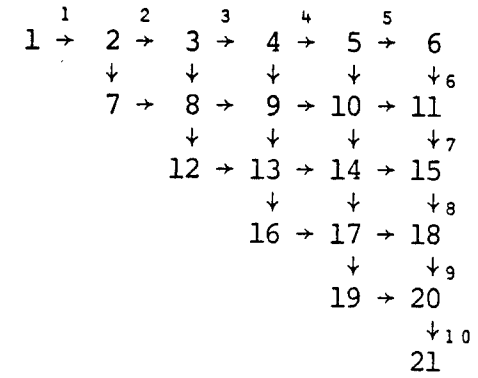
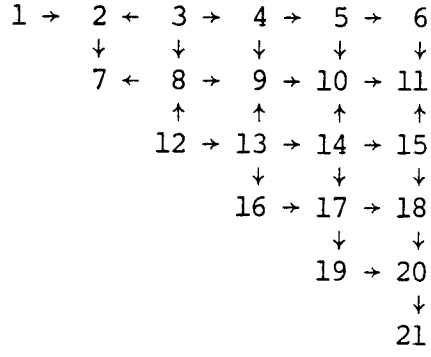
APPENDIX **C.1 Graphs Used to Find the Transfer Matrix**

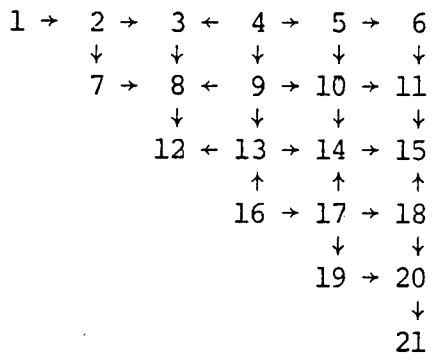
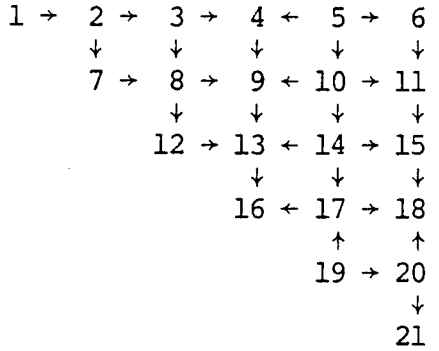
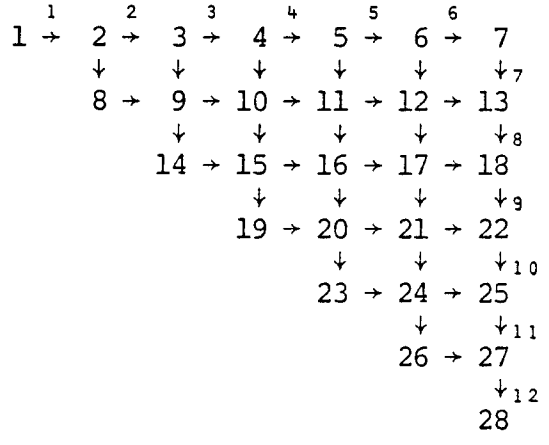
GRAPH ID.	GRAPH	TABLE NUMBER CORRESPONDING TO THIS GRAPH
1.1		2.1 8.3 10.3
1.2		1.1 3A.1 3A.2 4.1 4.2 6A.1 7.1 8.2 9.4 10.4

GRAPH ID.	GRAPH	TABLE NO.
2.1		3B.1 5.1.1 3B.2 5.2.1 3B.6 8.4 3B.7 9.3 10.2
2.2		3A.3 3A.4 4.3 4.4 5.2.3 6A.1
2.3		5.2.2 8.1 10.1
2.4		5.1.2

GRAPH ID.	GRAPH	TABLE NO.
3.1		3B.3.1 3B.3.2 3B.4
3.2		6A.3
3.3		9.1

GRAPH ID.	GRAPH	TABLE NO.
4.1		6A.4.1 6A.4.2 9.2.1 9.2.2
4.2		6B.1.1.3 6B.1.2.3 6B.3.1 6B.3.2 6B.6.1 9.2.1 9.2.2
4.3		6B.1.1.2 6B.1.2.2

GRAPH ID.	GRAPH	TABLE NO.
4.4		6B.1.1.1
4.5		6B.1.2.1
5.1		6B.2.1 6B.2.2 6B.2.3 6B.5.1 6B.5.2 6B.7.1.1 6B.7.2.1 6B.7.3.1
5.2		6B.7.1.2

GRAPH ID.	GRAPH	TABLE NO.
5.3		6B.7.2.2
5.4		6B.7.3.2
6.1		6B.4.1 6B.4.2 6B.4.3 6B.4.4 6B.4.5 6B.4.6

APPENDIX C.2 Tables Associated with Graphs

(1.1)

CASE	y_1		FUNCTION
1	+ -		0
2	+	-	2
3		+ -	1

(2.1)

CASE	y_1		FUNCTION
1	+ -		0
2	+	-	1
3		+ -	2

(3A.1)

CASE	y_1		FUNCTION
1	+ -		1
2	+	-	2
3		+ -	3

(3A.2)

CASE	y_2		FUNCTION
1	+ -		1
2	+	-	4
3		+ -	5

(3A.3)

CASE	y_3	y_2		FUNCTION
1	+ -			0
2	+	-		6
3	+		-	7
4		+ -		8
5		+	-	9
6			+ -	10

(3A.4)

CASE	y_3	y_1		FUNCTION
1	+ -			0
2	+	-		6
3	+		-	11
4		+ -		8
5		+	-	12
6			+ -	13

(3B.1)

CASE	y_2	y_1		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	2
4		+ -		3
5		+	-	4
6			+ -	5

(3B.2)

CASE	y_2	y_3		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	6
4		+ -		3
5		+	-	7
6			+ -	8

(3B.3.1)

CASE	Y_1	Y_2	Y_4	FUNCTION
1	+ -			0
2	+	-		9
3	+		-	10
4	+			- 11
5		+ -		12
6		+	-	13
7		+		- 14
8			+ -	15
9			+	- 16
10				+ - 5

(3B.3.2)

CASE	Y_2	Y_1	Y_4	FUNCTION
1	+ -			0
2	+	-		1
3	+		-	10
4	+			- 11
5		+ -		3
6		+	-	17
7		+		- 18
8			+ -	15
9			+	- 16
10				+ - 5

(3B.4.1)

CASE	Y_2	Y_3	Y_4	FUNCTION
1	+ -			0
2	+	-		1
3	+		-	19
4	+			- 20
5		+ -		3
6		+	-	21
7		+		- 22
8			+ -	23
9			+	- 24
10				+ - 8

(3B.4.2)

CASE	Y_3	Y_2	Y_4	FUNCTION
1	+ -			0
2	+	-		26
3	+		-	19
4	+			- 20
5		+ -		27
6		+	-	31
7		+		- 32
8			+ -	23
9			+	- 24
10				+ - 8

(3B.6)

CASE	y_1		y_2		FUNCTION
1	+	−			0
2	+		−		9
3	+			−	29
4		+	−		12
5		+		−	25
6			+	−	5

(3B.7)

CASE	y_3		y_2		FUNCTION
1	+	−			0
2	+		−		26
3	+			−	30
4		+	−		27
5		+		−	28
6			+	−	8

(4.1)

CASE	y_1		FUNCTION
1	+ -		1
2	+	-	2
3		+ -	3

(4.2)

CASE	y_2		FUNCTION
1	+ -		1
2	+	-	4
3		+ -	5

(4.3)

CASE	y_3	y_2		FUNCTION
1	+ -			0
2	+	-		6
3	+		-	7
4		+ -		8
5		+	-	9
6			+ -	10

(4.4)

CASE	y_3	y_1		FUNCTION
1	+ -			0
2	+	-		6
3	+		-	11
4		+ -		8
5		+	-	12
6			+ -	13

(5.1)

CASE	y_1	y_3		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	2
4		+ -		3
5		+	-	4
6			+ -	5

(5.2)

CASE	y_1	y_2		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	6
4		+ -		3
5		+	-	7
6			+ -	8

(6A.1)

CASE	y_1	y_5	FUNCTION
1	+ -		1
2	+	-	2
3	+	-	3
4	+ -		4
5	+	-	5
6		+ -	0

(6A.2)

CASE	y_7	FUNCTION
1	+ -	1
2	+	- 6
3	+ -	0

(6A.3)

CASE	y_3	y_6	y_7	FUNCTION
1	+ -			0
2	+	-		7
3	+	-		8
4	+		-	9
5	+ -			10
6	+	-		11
7	+		-	12
8		+ -		1
9		+	-	6
10			+ -	0

(6A.4.1)

CASE	Y_3	Y_6	Y_1	Y_5	FUNCTION
1	+ −				0
2	+	−			7
3	+		−		8
4	+			−	13
5	+				− 14
6		+ −			10
7		+	−		11
8		+		−	15
9		+			− 16
10			+ −		1
11			+	−	2
12			+		− 3
13				+ −	4
14				+	− 5
15					+ − 0

(6A.4.2)

CASE	Y_3	Y_1	Y_6	Y_5	FUNCTION
1	+ −				0
2	+	−			7
3	+		−		17
4	+			−	13
5	+				− 14
6		+ −			10
7		+	−		18
8		+		−	15
9		+			− 16
10			+ −		19
11			+	−	20
12			+		− 21
13				+ −	4
14				+	− 5
15					+ − 0

(6B.1.1)

CASE	Y_1	Y_2	Y_3	Y_4		FUNCTION
1	+ -					1
2	+	-				2
3	+		-			3
4	+			-		4
5	+				-	5
6		+ -				6
7		+	-			7
8		+		-		8
9		+			-	9
10			+ -			10
11			+	-		11
12			+		-	12
13				+ -		13
14				+	-	14
15					+ -	1

(6B.1.2)

CASE	Y_1	Y_3	Y_2	Y_4		FUNCTION
1	+ -					1
2	+	-				2
3	+		-			15
4	+			-		4
5	+				-	5
6		+ -				6
7		+	-			16
8		+		-		8
9		+			-	9
10			+ -			17
11			+	-		18
12			+		-	19
13				+ -		13
14				+	-	14
15					+ -	1

(6B.2.1)

CASE	Y_1	Y_2	Y_3	Y_6	Y_5	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			3
4	+			−		4
5	+				−	44
6	+					− 45
7		+ −				6
8		+	−			7
9		+		−		8
10		+			−	46
11		+				− 47
12			+ −			10
13			+	−		11
14			+		−	48
15			+			− 49
16				+ −		13
17				+	−	50
18				+		− 51
19					+ −	52
20					+	− 53
21						+ − 1

(6B.2.2)

CASE	y_1	y_3	y_2	y_6	y_5	FUNCTION
1	+ -					1
2	+	-				2
3	+		-			15
4	+			-		4
5	+				-	44
6	+					- 45
7		+ -				6
8		+	-			16
9		+		-		8
10		+			-	46
11		+				- 47
12			+ -			17
13			+	-		18
14			+		-	54
15			+			- 55
16				+ -		13
17				+	-	50
18				+		- 51
19					+ -	52
20					+	- 53
21						+ - 1

(6B.2.3)

CASE	Y_1	Y_2	Y_6	Y_3	Y_5	FUNCTION
1	+ -					1
2	+	-				2
3	+		-			3
4	+			-		56
5	+				-	44
6	+					- 45
7		+ -				6
8		+	-			7
9		+		-		57
10		+			-	58
11		+				- 59
12			+ -			10
13			+	-		60
14			+		-	61
15			+			- 62
16				+ -		63
17				+	-	64
18				+		- 65
19					+ -	52
20					+	- 53
21						+ - 1

(6B.3.1)

CASE	Y_1	Y_3	Y_7	Y_5		FUNCTION
1	+ -					1
2	+	-				2
3	+		-			15
4	+			-		69
5	+				-	70
6		+ -				6
7		+	-			16
8		+		-		72
9		+			-	73
10			+ -			17
11			+	-		66
12			+		-	67
13				+ -		52
14				+	-	53
15					+ -	1

(6B.3.2)

CASE	Y_1	Y_7	Y_3	Y_5		FUNCTION
1	+ -					1
2	+	-				2
3	+		-			68
4	+			-		69
5	+				-	70
6		+ -				6
7		+	-			71
8		+		-		72
9		+			-	73
10			+ -			63
11			+	-		64
12			+		-	65
13				+ -		52
14				+	-	53
15					+ -	1

(6B.4.1)

CASE	Y_1	Y_2	Y_6	Y_3	Y_5	Y_8	FUNCTION
1	+ −						1
2	+	−					2
3	+		−				3
4	+			−			56
5	+				−		44
6	+					−	74
7	+						− 75
8		+ −					6
9		+	−				7
10		+		−			57
11		+			−		58
12		+				−	76
13		+					− 77
14			+ −				10
15			+	−			60
16			+		−		61
17			+			−	78
18			+				− 79
19				+ −			63
20				+	−		64
21				+		−	80
22				+			− 81
23					+ −		52
24					+	−	82
25					+		− 83
26						+ −	84
27						+	− 85
28							+ − 1

(6B.4.2)

CASE	Y_1	Y_2	Y_3	Y_6	Y_5	Y_8	FUNCTION
1	+ -						1
2	+	-					2
3	+		-				3
4	+			-			4
5	+				-		44
6	+					-	74
7	+						- 75
8		+ -					6
9		+	-				7
10		+		-			8
11		+			-		46
12		+				-	76
13		+					- 77
14			+ -				10
15			+	-			11
16			+		-		48
17			+			-	78
18			+				- 79
19				+ -			13
20				+	-		50
21				+		-	86
22				+			- 87
23					+ -		52
24					+	-	82
25					+		- 83
26						+ -	84
27						+	- 85
28							+ - 1

(6B.4.3)

CASE	Y_1	Y_2	Y_3	Y_5	Y_6	Y_8	FUNCTION
1	+ -						1
2	+	-					2
3	+		-				3
4	+			-			4
5	+				-		20
6	+					-	74
7	+						- 75
8		+ -					6
9		+	-				7
10		+		-			8
11		+			-		22
12		+				-	76
13		+					- 77
14			+ -				10
15			+	-			11
16			+		-		24
17			+			-	78
18			+				- 79
19				+ -			13
20				+	-		26
21				+		-	86
22				+			- 87
23					+ -		28
24					+	-	88
25					+		- 89
26						+ -	84
27						+	- 85
28							+ - 1

(6B.4.4)

CASE	Y_1	Y_3	Y_2	Y_6	Y_5	Y_8	FUNCTION
1	+ -						1
2	+	-					2
3	+		-				15
4	+			-			4
5	+				-		44
6	+					-	74
7	+						- 75
8		+ -					6
9		+	-				16
10		+		-			8
11		+			-		46
12		+				-	76
13		+					- 77
14			+ -				17
15			+	-			18
16			+		-		54
17			+			-	90
18			+				- 91
19				+ -			13
20				+	-		50
21				+		-	86
22				+			- 87
23					+ -		52
24					+	-	82
25					+		- 83
26						+ -	84
27						+	- 85
28							+ - 1

(6B.4.5)

CASE	Y_1	Y_3	Y_2	Y_5	Y_6	Y_8	FUNCTION
1	+ -						1
2	+	-					2
3	+		-				15
4	+			-			4
5	+				-		20
6	+					-	74
7	+						75
8		+ -					6
9		+	-				16
10		+		-			8
11		+			-		22
12		+				-	76
13		+					77
14			+ -				17
15			+	-			18
16			+		-		30
17			+			-	90
18			+				91
19				+ -			13
20				+	-		26
21				+		-	86
22				+			87
23					+ -		28
24					+	-	88
25					+		89
26						+ -	84
27						+	85
28							1

(6B.4.6)

CASE	Y_1	Y_3	Y_5	Y_2	Y_6	Y_8	FUNCTION
1	+ -						1
2	+	-					2
3	+		-				15
4	+			-			32
5	+				-		20
6	+					-	74
7	+						- 75
8		+ -					6
9		+	-				16
10		+		-			33
11		+			-		22
12		+				-	76
13		+					- 77
14			+ -				17
15			+	-			34
16			+		-		35
17			+			-	90
18			+				- 91
19				+ -			37
20				+	-		38
21				+		-	92
22				+			- 93
23					+ -		28
24					+	-	88
25					+		- 89
26						+ -	84
27						+	- 85
28							+ - 1

(6B.5.1)

CASE	Y_1	Y_7	Y_3	Y_5	Y_8	FUNCTION
1	+ -					1
2	+	-				2
3	+		-			68
4	+			-		69
5	+				-	94
6	+					95
7		+ -				6
8		+	-			71
9		+		-		72
10		+			-	96
11		+				97
12			+ -			63
13			+	-		64
14			+		-	80
15			+			81
16				+ -		52
17				+	-	82
18				+		83
19					+ -	84
20					+	85
21						1

(6B.5.2)

CASE	y_1	y_3	y_7	y_5	y_8	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			15
4	+			−		69
5	+				−	94
6	+					95
7		+ −				6
8		+	−			16
9		+		−		72
10		+			−	96
11		+				97
12			+ −			17
13			+	−		66
14			+		−	98
15			+			99
16				+ −		52
17				+	−	82
18				+		83
19					+ −	84
20					+	85
21						1

(6B.5.3)

CASE	Y_1	Y_3	Y_5	Y_7	Y_8	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			15
4	+			−		32
5	+				−	94
6	+					− 95
7		+ −				6
8		+	−			16
9		+		−		33
10		+			−	96
11		+				− 97
12			+ −			17
13			+	−		34
14			+		−	98
15			+			− 99
16				+ −		37
17				+	−	100
18				+		− 101
19					+ −	84
20					+	− 85
21						+ − 1

(6B.6.1)

CASE	Y_1	Y_3	Y_5	Y_7	FUNCTION
1	+ -				1
2	+	-			2
3	+		-		15
4	+			-	32
5	+				- 40
6		+ -			6
7		+	-		16
8		+		-	33
9		+			- 41
10			+ -		17
11			+	-	34
12			+		- 42
13				+ -	37
14				+	- 43
15					+ - 1

(6B.7.1)

CASE	Y_1	Y_2	Y_3	Y_5	Y_6	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			3
4	+			−		4
5	+				−	20
6	+					− 21
7		+ −				6
8		+	−			7
9		+		−		8
10		+			−	22
11		+				− 23
12			+ −			10
13			+	−		11
14			+		−	24
15			+			− 25
16				+ −		13
17				+	−	26
18				+		− 27
19					+ −	28
20					+	− 29
21						+ − 1

(6B.7.2)

CASE	Y_1	Y_3	Y_2	Y_5	Y_6	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			15
4	+			−		4
5	+				−	20
6	+					21
7		+ −				6
8		+	−			16
9		+		−		8
10		+			−	22
11		+				23
12			+ −			17
13			+	−		18
14			+		−	30
15			+			31
16				+ −		13
17				+	−	26
18				+		27
19					+ −	28
20					+	29
21						1

(6B.7.3)

CASE	y_1	y_3	y_5	y_2	y_6	FUNCTION
1	+ −					1
2	+	−				2
3	+		−			15
4	+			−		32
5	+				−	20
6	+					− 21
7		+ −				6
8		+	−			16
9		+		−		33
10		+			−	22
11		+				− 23
12			+ −			17
13			+	−		34
14			+		−	35
15			+			− 36
16				+ −		37
17				+	−	38
18				+		− 39
19					+ −	28
20					+	− 29
21						+ − 1

(7.1)

CASE	Y_1		FUNCTION
1	+ -		0
2	+	-	1
3		+ -	2

(8.1)

CASE	Y_2	Y_1		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	2
4		+ -		3
5		+	-	4
6			+ -	5

(8.2)

CASE	Y_1		FUNCTION
1	+ -		6
2	+	-	7
3		+ -	8

(8.3)

CASE	Y_3		FUNCTION
1	+ -		6
2	+	-	9
3		+ -	10

(8.4)

CASE	Y_2	Y_3		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	11
4		+ -		3
5		+	-	12
6			+ -	13

(9.1)

CASE	Y_1	Y_2	Y_3	FUNCTION
1	+ -			1
2	+	-		2
3	+		-	3
4	+			-	4
5		+ -			5
6		+	-		6
7		+		-	7
8			+ -		8
9			+	-	9
10				+ -	1

(9.2.1)

CASE	Y_1	Y_4	Y_2	Y_5	FUNCTION
1	+ -				1
2	+	-			2
3	+		-		10
4	+			-	11
5	+				-	12
6		+ -				5
7		+	-			13
8		+		-		14
9		+			-	15
10			+ -			16
11			+	-		17
12			+		-	18
13				+ -		19
14				+	-	20
15					+ -	1

(9.2.2)

CASE	Y_1	Y_2	Y_4	Y_5	FUNCTION
1	+ -				1
2	+	-			2
3	+		-		3
4	+			-	11
5	+				- 12
6		+ -			5
7		+	-		6
8		+		-	14
9		+			- 15
10			+ -		8
11			+	-	21
12			+		- 22
13				+ -	19
14				+	- 20
15					+ - 1

(9.3)

CASE	Y_4	Y_5	FUNCTION
1	+ -		8
2	+	-	21
3	+		- 22
4		+ -	19
5		+	- 20
6			+ - 1

(9.4)

CASE	Y_3		FUNCTION
1	+ -		8
2	+	-	9
3		+ -	1

(10.1)

CASE	y_1	y_2		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	2
4		+ -		3
5		+	-	4
6			+ -	5

(10.2)

CASE	y_1	y_3		FUNCTION
1	+ -			0
2	+	-		1
3	+		-	6
4		+ -		3
5		+	-	7
6			+ -	8

(10.3)

CASE	y_3		FUNCTION
1	+ -		9
2	+	-	10
3		+ -	11

(10.4)

CASE	y_2		FUNCTION
1	+ -		9
2	+	-	12
3		+ -	13

NOTES

(i) All functions y_i are given in Appendix 4.B.3.

(ii) $\omega_o = \arcsin(r_I/r_o)$

$\omega_1 = \arcsin(r_o \sin \omega_-/r_I)$

$\omega_2 = \arcsin(r_o \sin \omega_+/r_I)$

$\chi_1 = \arcsin(r_I \sin \omega_-/r_o)$

$\chi_2 = \arcsin(r_I \sin \omega_+/r_o)$

$\hat{\omega}_- = \max\{ \omega_- , \omega_o \}$

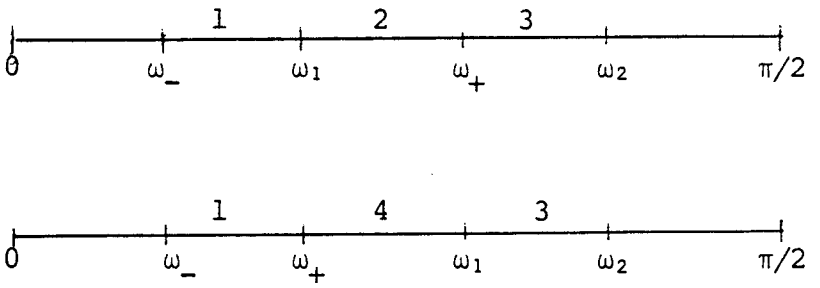
$\omega_\pm = \omega_{m\pm 1/2}$

$\chi = \pi/2 - \omega$

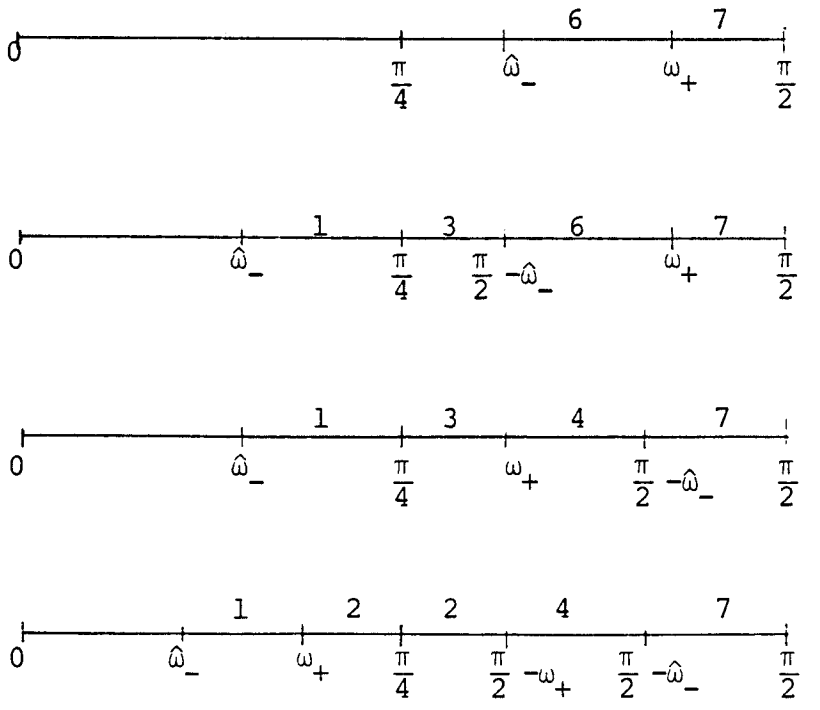
(iii) Since Graphs 5.1.1 and 5.1.2 use the same table, only Table 5.1 is given in Appendix 4.C.2. The same is true of Graphs 5.2.(1,2,3), 6B.1.1.(1,2,3), 6B.1.2.(1,2,3), 6B.7.1.(1,2), 6B.7.2.(1,2), and 6B.7.3.(1,2). That is, tables having indices without numbers inside parentheses are given.

(iv) If there are more than one table under the first index of table numbers, one of them is chosen by examining the following figures where numbers over intervals indicate the second index of the table numbers.

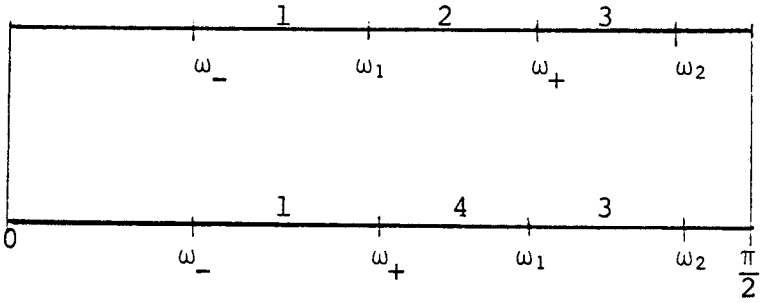
(3A.)



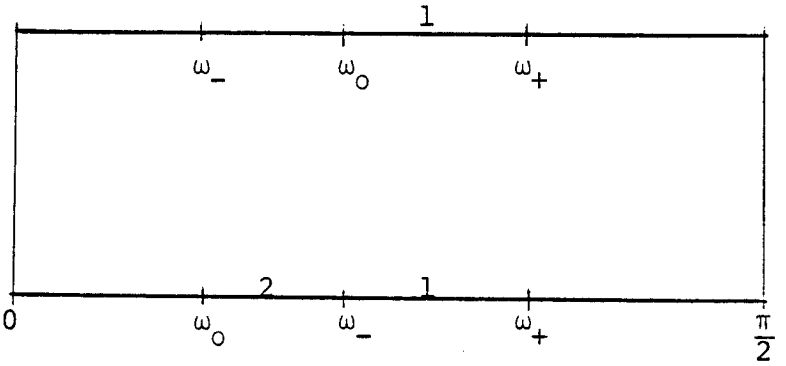
(3B.)



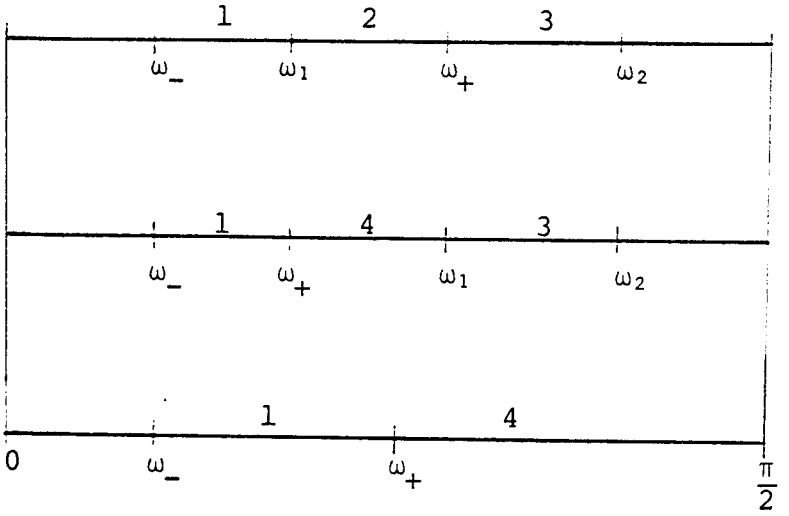
(4.)


1
2
3
ω_-
ω_1
ω_+
ω_2
1
4
3
0
ω_-
ω_+
ω_1
ω_2
$\frac{\pi}{2}$

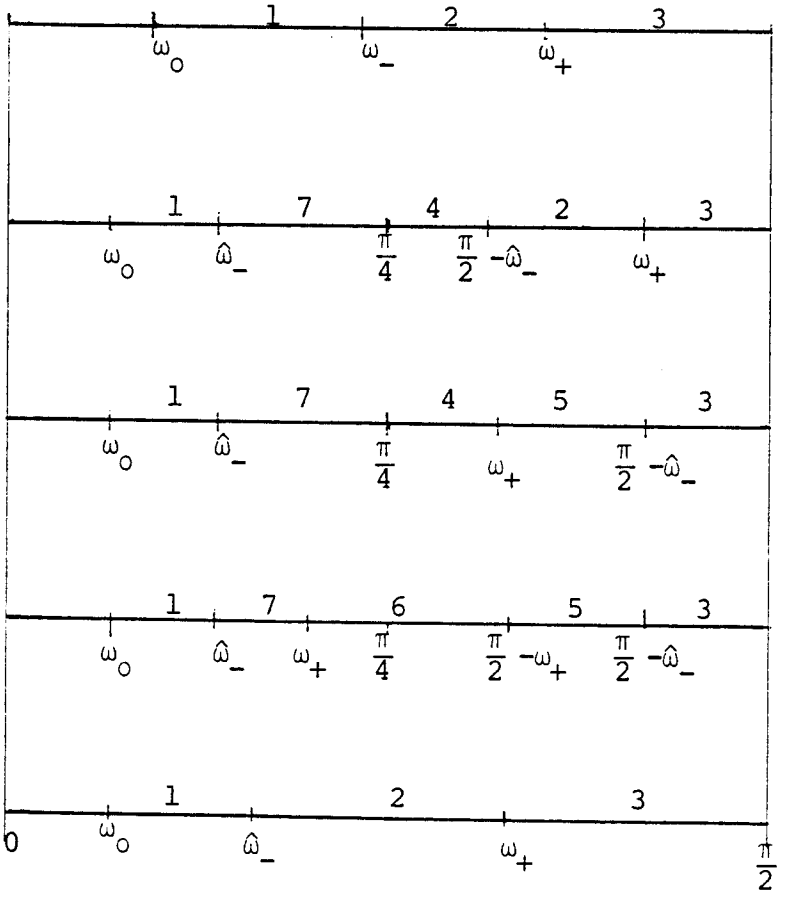
(5.)


1
ω_-
ω_o
ω_+
2
1
0
ω_o
ω_-
ω_+
$\frac{\pi}{2}$

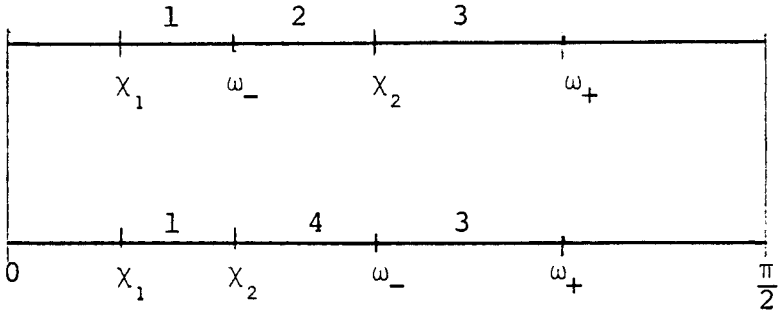
(6A.)


1
2
3
ω_-
ω_1
ω_+
ω_2
1
4
3
ω_-
ω_+
ω_1
ω_2
1
4
0
ω_-
ω_+
$\frac{\pi}{2}$

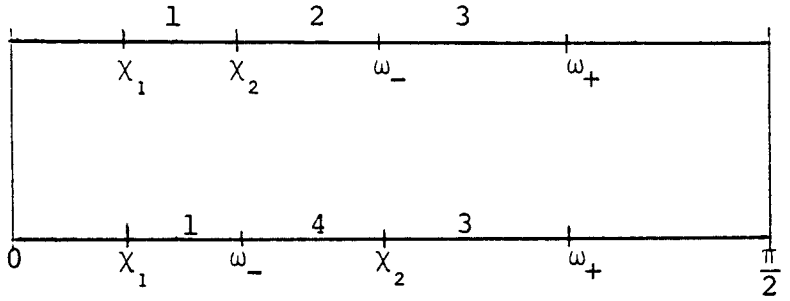
(6B.)


1
2
3
ω_o
ω_-
ω_+
1
7
4
2
3
ω_o
$\hat{\omega}_-$
$\frac{\pi}{4}$
$\frac{\pi}{2}-\hat{\omega}_-$
ω_+
1
7
4
5
3
ω_o
$\hat{\omega}_-$
$\frac{\pi}{4}$
ω_+
$\frac{\pi}{2}-\hat{\omega}_-$
1
7
6
5
3
ω_o
$\hat{\omega}_-$
ω_+
$\frac{\pi}{4}$
$\frac{\pi}{2}-\omega_+$
$\frac{\pi}{2}-\hat{\omega}_-$
1
2
3
0
ω_o
$\hat{\omega}_-$
ω_+
$\frac{\pi}{2}$

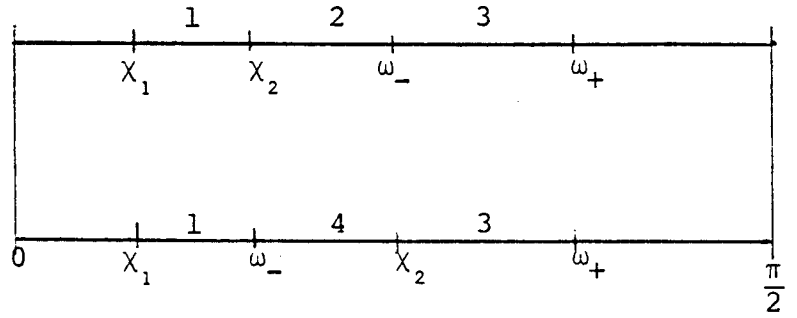
(8.)


1
2
3
χ_1
ω_-
χ_2
ω_+
1
4
3
0
χ_1
χ_2
ω_-
ω_+
$\frac{\pi}{2}$

(9.)


1
2
3
χ_1
χ_2
ω_-
ω_+
1
4
3
0
χ_1
ω_-
χ_2
ω_+
$\frac{\pi}{2}$

(10.)


1
2
3
χ_1
χ_2
ω_-
ω_+
1
4
3
0
χ_1
ω_-
χ_2
ω_+
$\frac{\pi}{2}$

APPENDIX D. ELEMENTS OF THE ESCAPE MATRICES

In the following formulas, the definitions of z_T, z_B, r_o, r_I, ω and η are given (from Figs. 2), with $\Delta z = z_T - z_B$ and $\omega_o = \arcsin(r_I/r_o)$. The definitions of $S_1(x)$, $S_2(x)$ and $S_3(x)$ are given in the end of this appendix.

[1] Streaming to OUTER

(i) $\omega < \omega_o$

$$P_{om} = \begin{cases} 2\pi \dfrac{r_o(z_1 - z_B)(2z_T - z_1 - z_B)}{2\eta} & \text{for } z_1 < z_T \quad (D.1) \\ 2\pi \dfrac{r_o \Delta z^2}{2\eta} & \text{for } z_1 \geq z_T \quad (D.2) \end{cases}$$

(ii) $\omega > \omega_o$

$$P_{om} = \begin{cases} 2\pi \dfrac{r_o(z_2 - z_3)(2z_T - z_2 - z_B)}{2\eta} & \text{for } z_2 < z_T \quad (D.3) \\ 2\pi \dfrac{r_o \Delta z^2}{2\eta} & \text{for } z_2 \geq z_T \quad (D.4) \end{cases}$$

z_1 and z_2 are given by

$$z_1 = z_B + \frac{r_o \cos\omega - \sqrt{r_I^2 - r_o^2 \sin^2\omega}}{\gamma}$$

and

$$z_2 = z_B + \frac{2r_o \cos\omega}{\gamma} .$$

[2] Streaming to TOP ($\omega < \pi/2$)

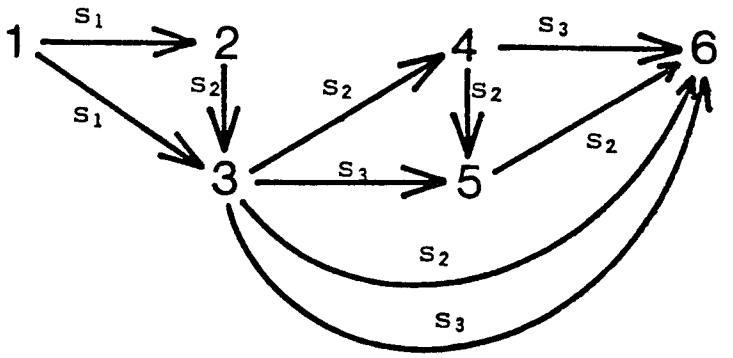
(i) $\omega < \omega_o$

$$P_{T1m} = \begin{cases} 2\pi\{[S_1(x)]_{r_I}^{r_1} + [S_2(x)]_{r_1}^{r_o}\} & \text{for } r_1 < r_o \quad (D.5) \\ 2\pi[S_1(x)]_{r_I}^{r_o} & \text{for } r_1 > r_o \quad (D.6) \end{cases}$$

where $$r_1 = \gamma\Delta z \cos\omega + \sqrt{r_I^2 - \gamma^2\Delta z^2 \sin^2\omega}\ .$$

(ii) $\omega > \omega_o$

The result is presented by using the following graph:



In this graph, vertices 1, 2, 3, 4, 5, and 6 are associated with the values of r, r_I, r_1, r_2, r_3, r_4, and r_o, respectively. Here r_1, r_2, r_3, and r_4 are defined as follows:

$$r_1 = \gamma\Delta z \cos\omega + \sqrt{r_I^2 - \gamma^2\Delta z \sin^2\omega} ,$$

$$r_2 = r_I/\sin\omega ,$$

$$r_3 = \gamma\Delta z \cos\omega + \sqrt{r_o^2 - \gamma^2\Delta z \sin^2\omega} ,$$

$$r_4 = \gamma\Delta z \cos\omega - \sqrt{r_o^2 - \gamma^2\Delta z^2 \sin^2\omega} .$$

s_1, s_2, and s_3 attached to arcs indicate the functions $S_1(x)$, $S_2(x)$, and $S_3(x)$, respectively. The vertex is traced in the direction of arrows from the vertex 1 ($r = r_I$) to the vertex 6 ($r = r_o$) picking up a function. When there are more than two arrows starting at a vertex, it goes to a vertex whose r is the smallest. An exception is from the vertex 3 to the vertex 6. In this case s_2 is chosen if $\gamma\Delta z < r_o \sin(\omega + \omega_0)/\sin\omega$; otherwise, s_3 is chosen.

Let us explain the algorithm using a specific example. Suppose that we are at the vertex 1, and $r_2 > r_3$. By the rule, we go to the vertex 3 picking up the function $S_1(x)$. Next suppose that $r_4 > r_6 > r_5$. Now we go to the vertex 5 picking up the function $S_3(x)$. Then we go to the vertex 6 picking up the function $S_2(x)$. As a result, we obtain the following formula for the escape matrix element P_{T1m}:

$$P_{T1m} = 2\pi\{[S_1(x)]_{r_I}^{r_3} + [S_3(x)]_{r_3}^{r_5} + [S_2(x)]_{r_5}^{r_o}\} . \qquad (D.7)$$

[3] Streaming to INNER

$$P_{Im} = \begin{cases} 2\pi \dfrac{r_I(z_1 - z_B)(2z_T - z_1 - z_B)}{2\eta} & \text{for } z_1 < z_T \quad (D.8) \\ 2\pi \dfrac{r_o \Delta z^2}{2\eta} & \text{for } z_1 > z_T \quad (D.9) \end{cases}$$

where $z_1 = z_B + (\sqrt{r_o^2 - r_I^2 \sin^2\chi} - r_I \cos\chi)/\gamma$

and $\chi = \pi - \omega$.

[4] Streaming to TOP ($\pi > \omega > \pi/2$)

$$P_{T2m} = \begin{cases} 2\pi\{[S_2(x)]_{r_I}^{r_3} + [S_3(x)]_{r_3}^{r_o}\} & \text{for } r_3 > r_I \quad (D.10) \\ 2\pi[S_3(x)]_{r_I}^{r_o} & \text{for } r_3 < r_2 \quad (D.11) \end{cases}$$

where $r_3 = \sqrt{r_o^2 - \gamma^2 \Delta z^2 \sin^2\chi} - \gamma \Delta z \cos\chi$ and $\chi = \pi - \omega$.

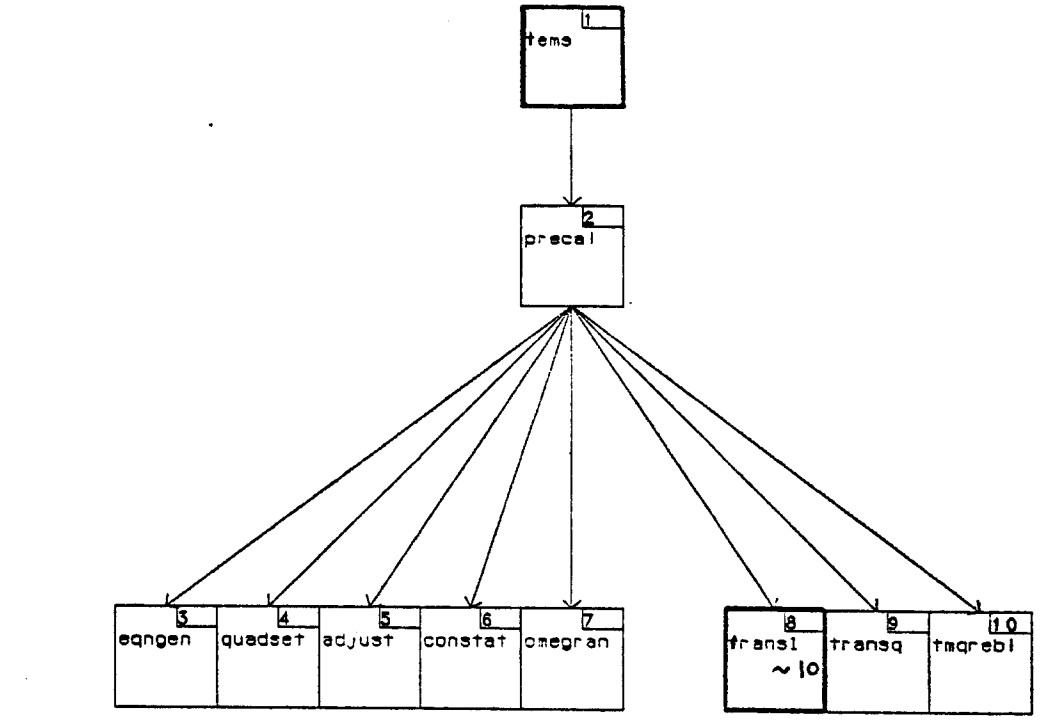
The definitions of S_1, S_2, and S_3 are as follows:

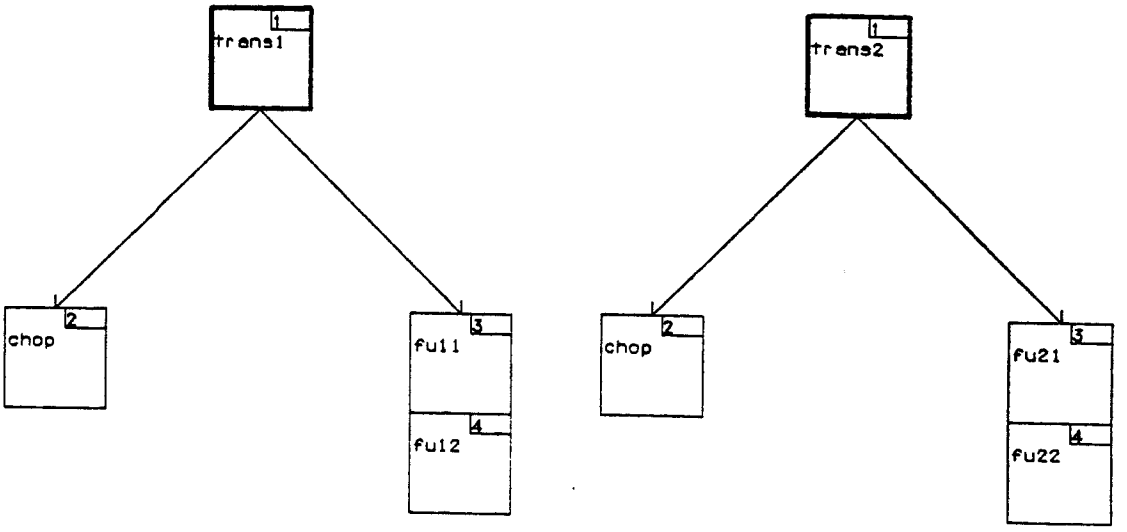
$$S_1(x) = \frac{1}{3\sqrt{1 - \eta^2}} \{x^3 \cos\omega + \frac{(r_I^2 - x^2 \sin^2\omega)^{3/2}}{\sin^2\omega}\} \quad (D.12)$$

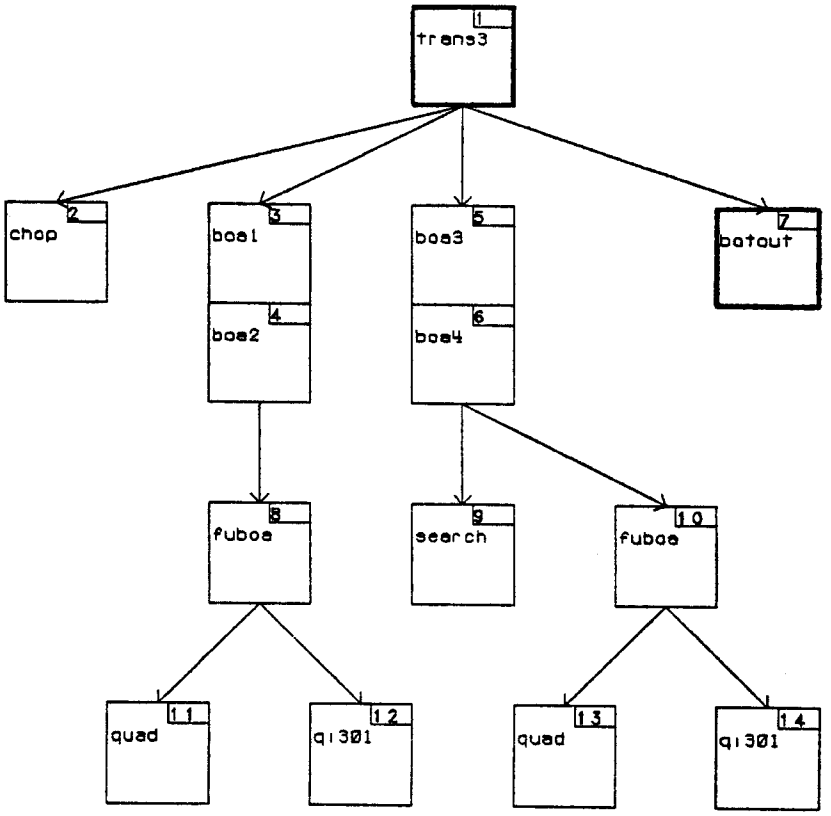
$$S_2(x) = \frac{\Delta z^2}{2\eta} x^2 \quad (D.13)$$

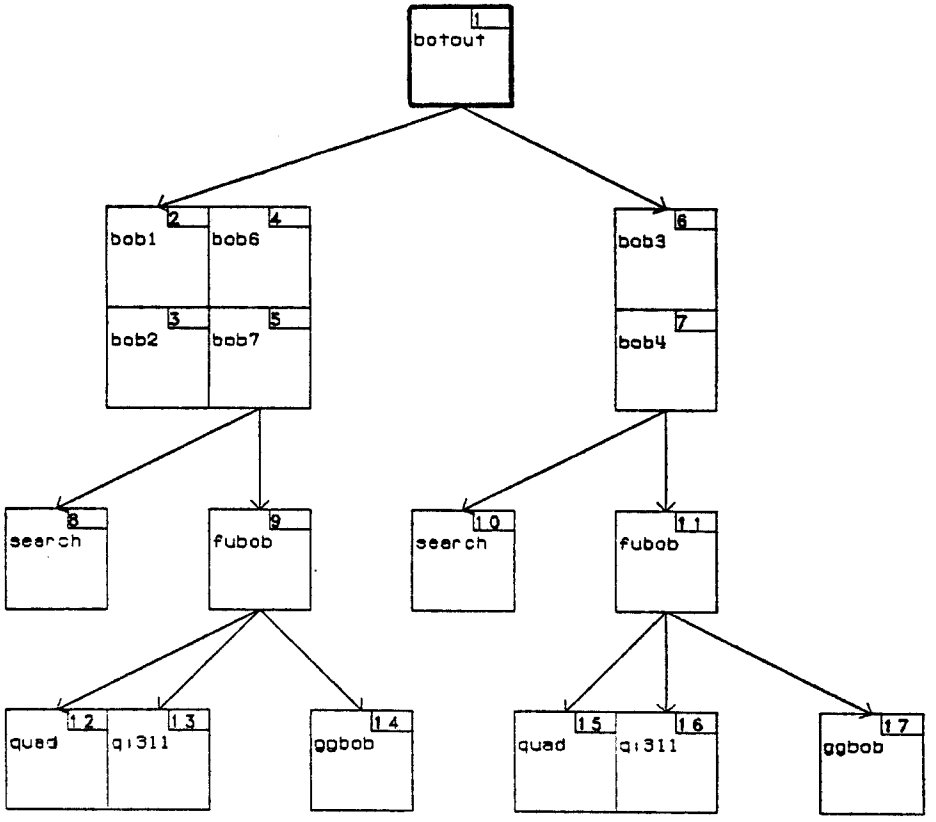
$$S_3(x) = \frac{1}{3\sqrt{1 - \eta^2}} \{x^3 \cos\omega - \frac{(r_o^2 - x^2 \sin^2\omega)^{3/2}}{\sin^2\omega}\} . \quad (D.14)$$

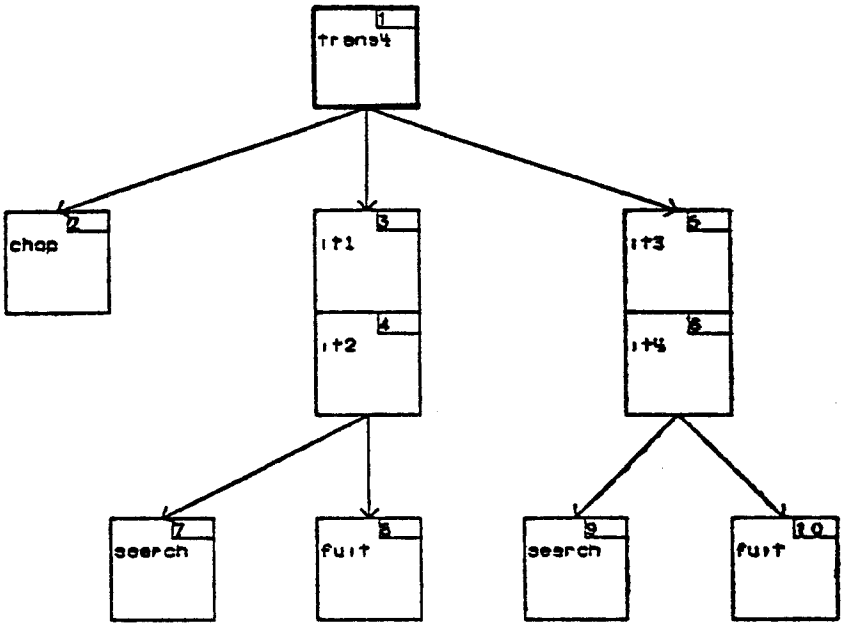
APPENDIX E. STRUCTURE CHARTS OF THE TEMS MODULE

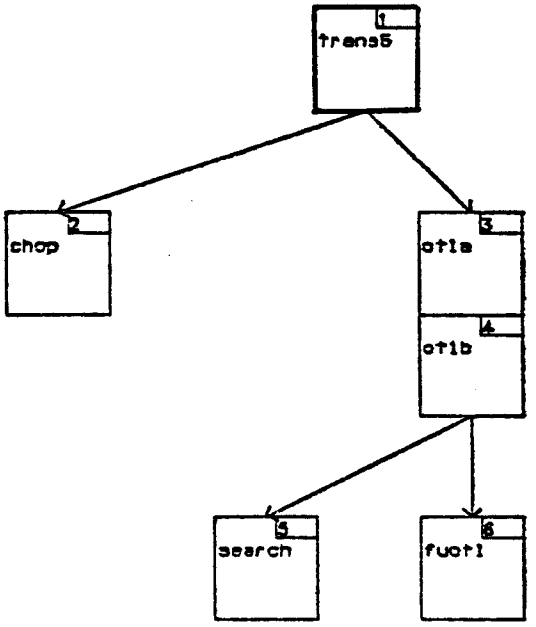

1
tems
2
precal
3
eqngen
4
quadset
5
adjust
6
constat
7
omegran
8
transl
~ 10
9
transq
10
tmqrebl

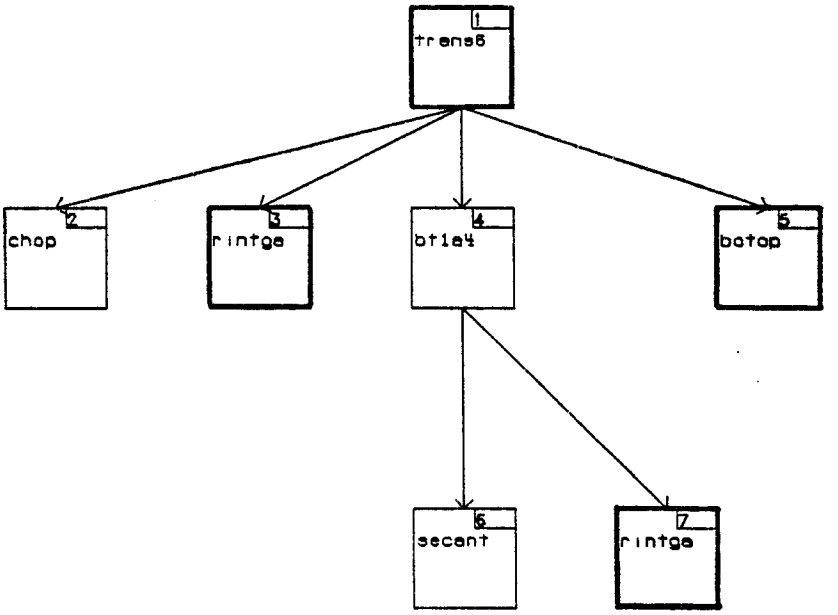

1
trans1
2
chop
3
fu11
4
fu12
1
trans2
2
chop
3
fu21
4
fu22

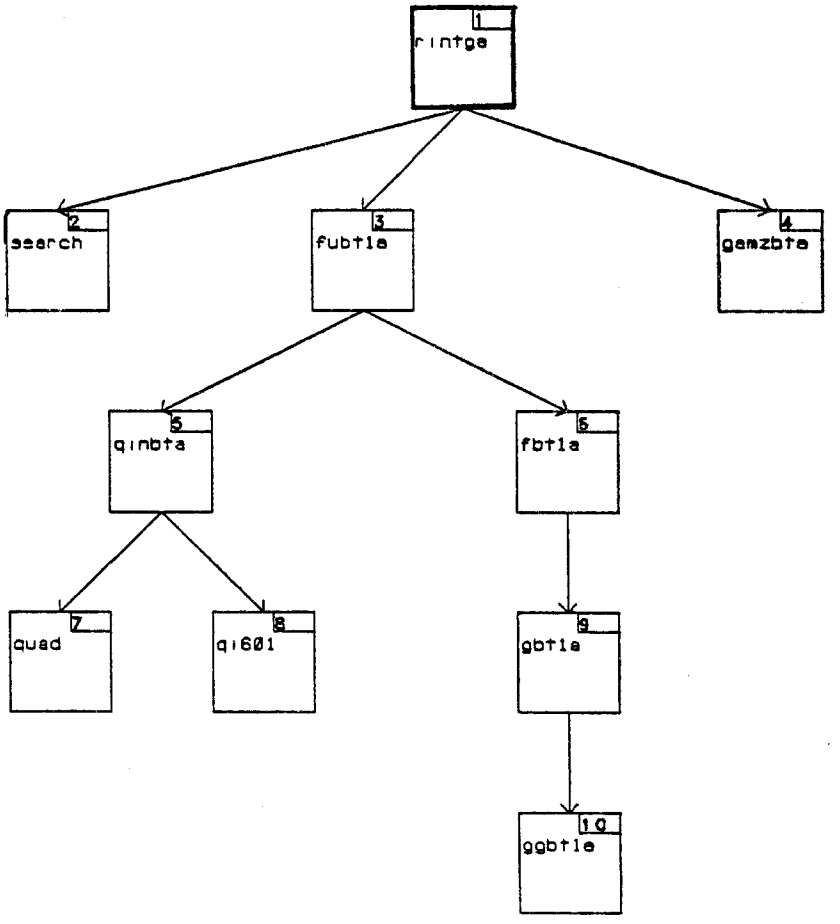

1
trans3
2
chop
3
boa1
4
boa2
5
boa3
6
boa4
7
botout
8
fuboa
9
search
10
fuboa
11
quad
12
q:301
13
quad
14
q:301

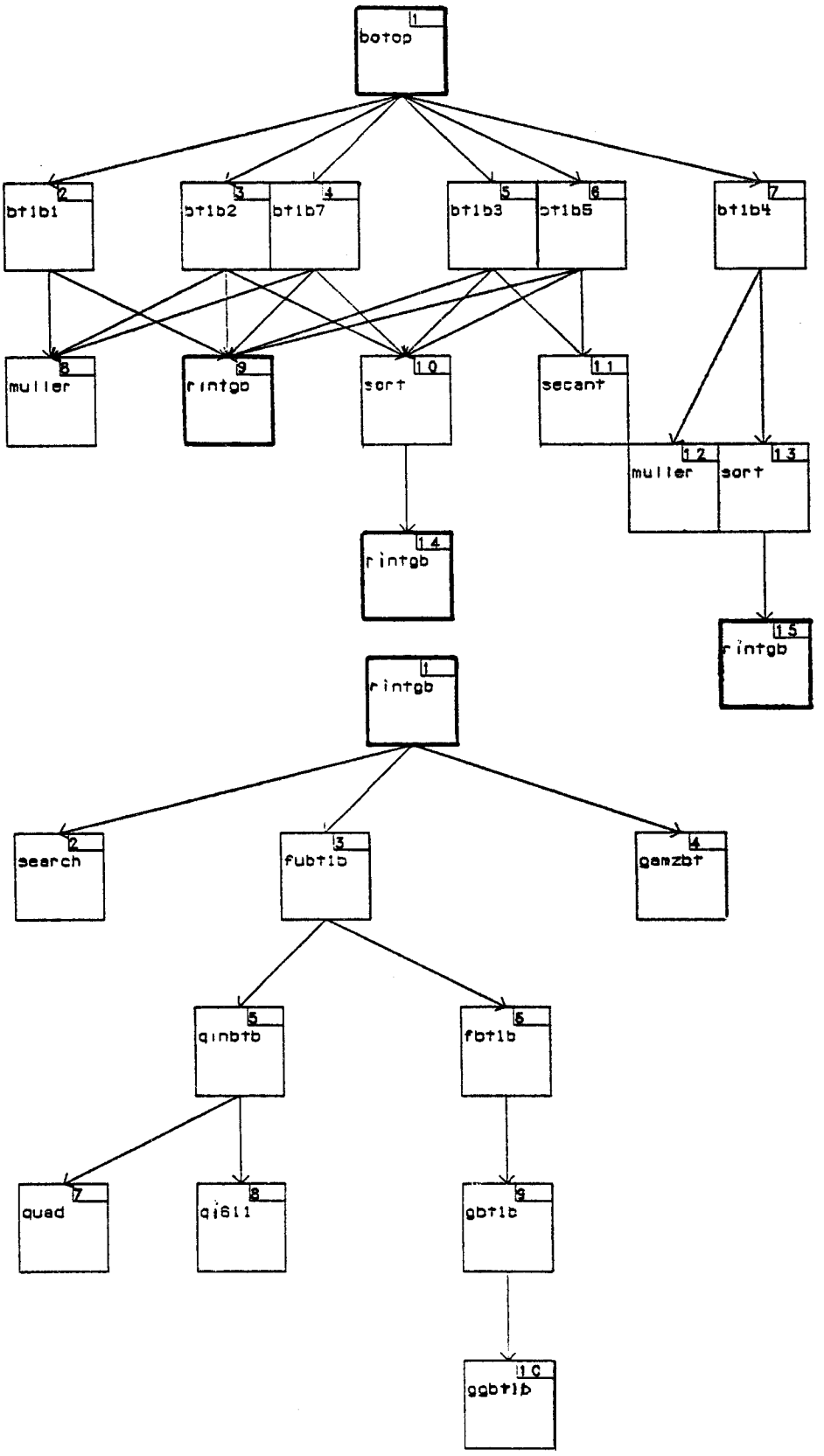

1
botout
2
bob1
4
bob6
3
bob2
5
bob7
6
bob3
7
bob4
8
search
9
fubob
10
search
11
fubob
12
quad
13
q:311
14
ggbob
15
quad
16
q:311
17
ggbob

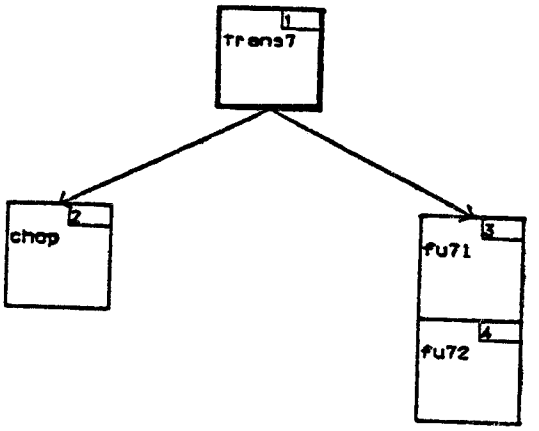

1
trans4
2
chop
3
it1
4
it2
5
it3
6
it4
7
search
8
fuit
9
search
10
fuit

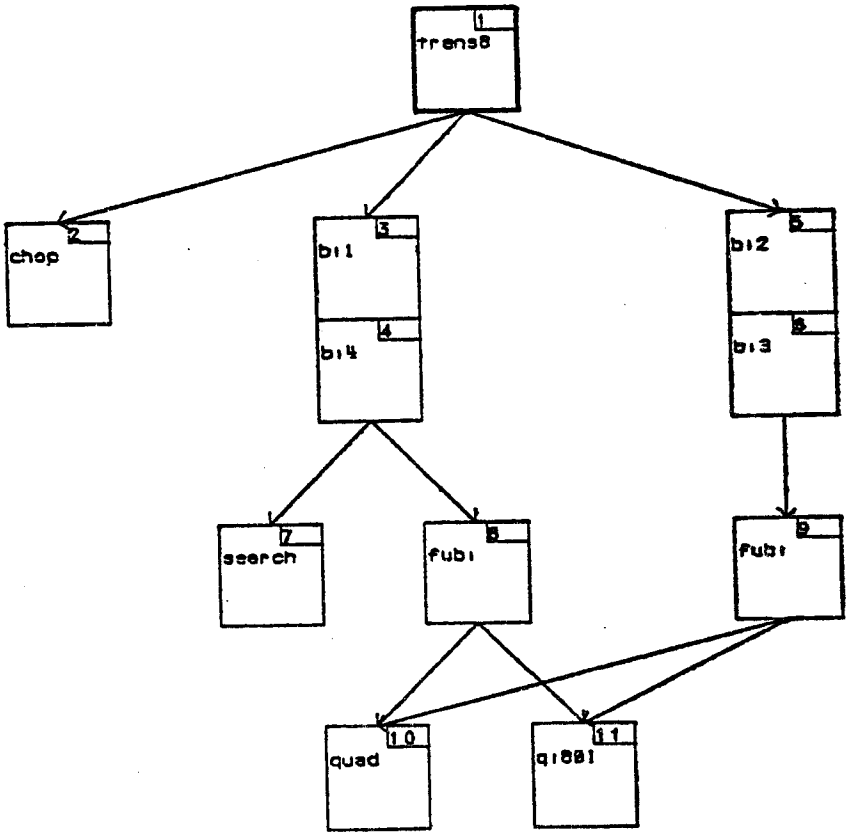

1
trans5
2
chop
3
otla
4
otlb
5
search
6
fuotl

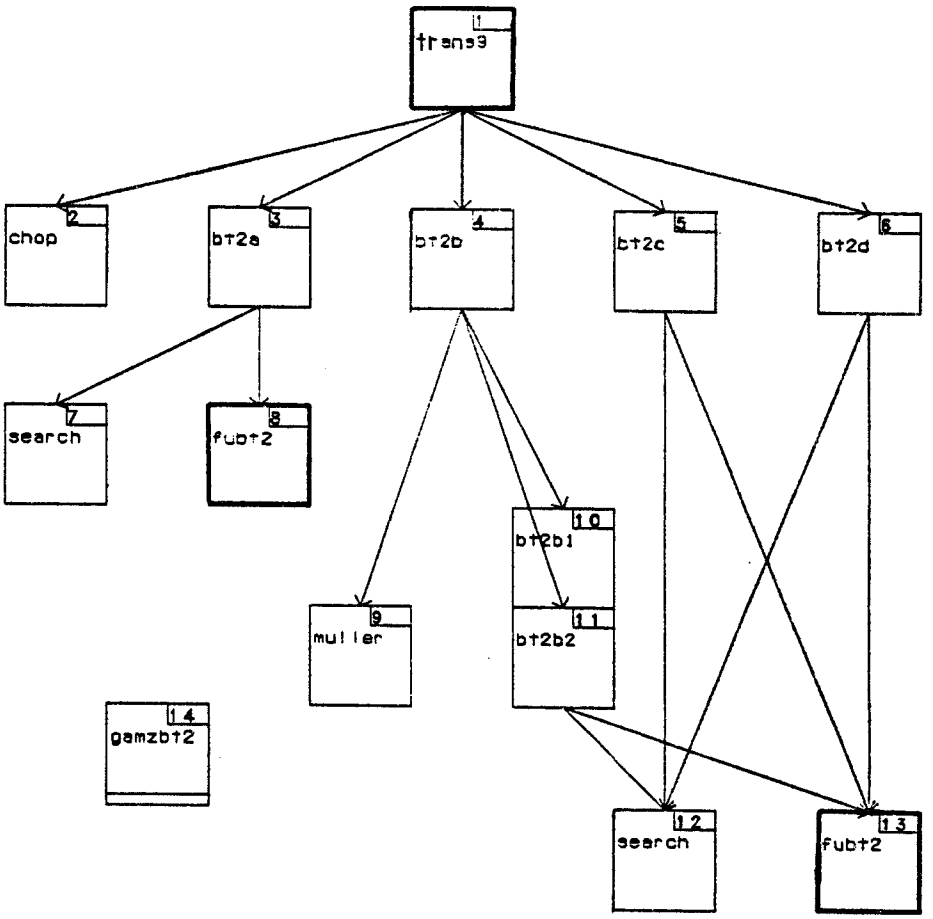

1
trans6
2
chop
3
rintge
4
bt1e4
5
botop
6
secant
7
rintge

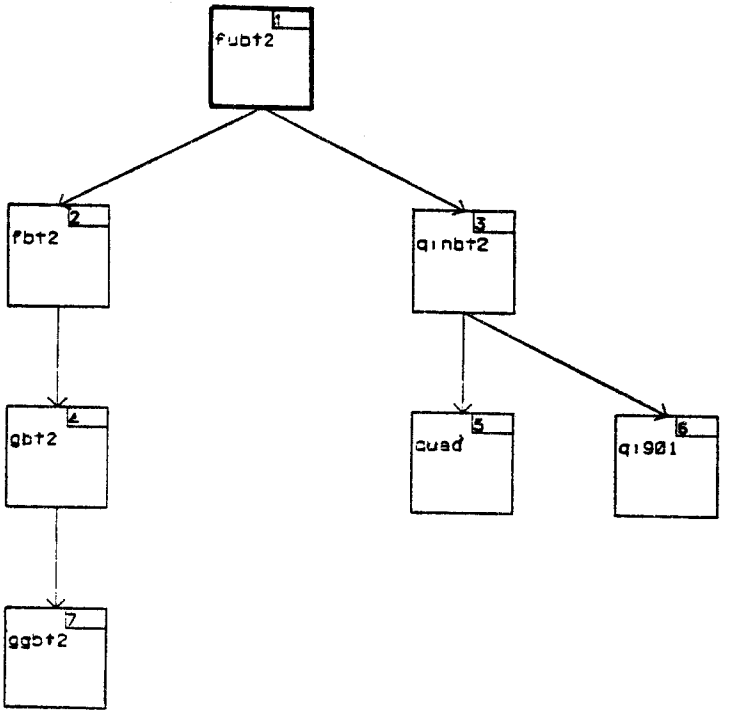

1
rintge
2
search
3
fubt1a
4
gamzbta
5
q1nbta
5
fbt1a
7
quad
8
q1601
9
gbt1a
10
ggbt1a

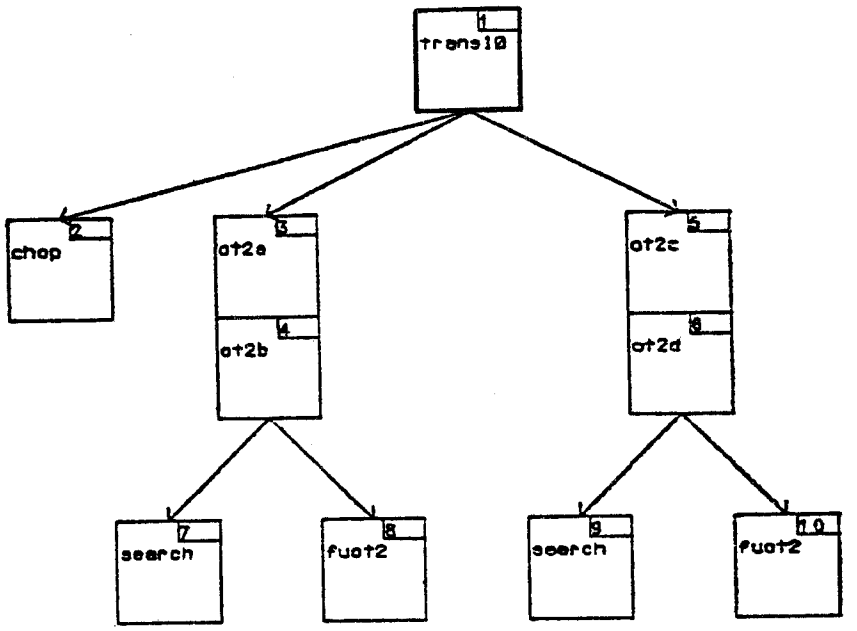

1
botop
2
bt1b1
3
bt1b2
4
bt1b7
5
bt1b3
6
bt1b5
7
bt1b4
8
muller
9
rintgb
10
sort
11
secant
12
muller
13
sort
14
rintgb
15
rintgb
1
rintgb
2
search
3
fubt1b
4
gamzbt
5
qinbtb
6
fbt1b
7
quad
8
qi611
9
gbt1b
10
ggbt1b


1
trans7
2
chop
3
fu71
4
fu72


1
trans8
2
chop
3
bi1
4
bi4
5
bi2
6
bi3
7
search
8
fubi
9
fubi
10
quad
11
qi881


1
trans3
2
chop
3
bt2a
4
bt2b
5
bt2c
6
bt2d
7
search
8
fubt2
9
muller
10
bt2b1
11
bt2b2
12
search
13
fubt2
14
gamzbt2


1
fubt2
2
fbt2
3
qinbt2
4
gbt2
5
cuad
6
qi901
7
ggbt2



* These charts were drawn by using the Structure Chart Interface(SCI) available through NMFECC at the Lawrence Livermore National Laboratory.