MASTER

DOE/CS/30254 -3

THE CONTINUING DEVELOPMENT
OF THE DEROB SYSTEM

Quarterly Report

for Period January 1, 1981-March 31, 1981

Francisco Arumi-Noe, Ph.D.
Professor
School of Architecture

University of Texas
Austin, Texas 78712

March 1981

Prepared for

THE U. S. DEPARTMENT OF ENERGY
AGREEMENT NO. DE-AC02-80CS30254

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed or represents that its use would not infringe privately-owned rights.

ABSTRACT

More accurate and efficient algorithms to calculate geometric factors and obstruction logic was developed, programmed and tested.

These new algorithms were needed in order to facilitate the incorporation of lighting, both natural and artificial, into the DEROB System.

With this new set of algorithms it will be possible to hard-wire, with minimum user involvement, the highly desirable capability of calculating light propagation via light shafts, courtyards and other multispace radiation coupling problems. This capability will include as a special case, the simple one room illumination analysis.

The report contains the basic theory for these algorithms, and the listing of the DRBGF module that has them incorporated.

The completed incorporation of diffuse lighting into DEROB is expected to be completed by September 1981.

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TABLE OF CONTENTS

OVERALL OBJECTIVES.	.1
COMPLETED TASKS	.2
SECTION I: Geometric Factors.	.I-1
SECTION II: Clipping Algorithm	II-1
SECTION III: Space Geometry	.III-1
SECTION IV: Transformation of Geometric Factors.	IV-1
SECTION V: Listing of DRBGF	.V-1

Overall Objectives

The overall objective of the project is to develop additional capabilities of the public domain version of the DEROB (Dynamic Energy Response of Buildings) program. This program is to be used as a research and analysis tool for the investigation of the energy response of buildings.

The objectives of this phase of the project are:

- a) to modify the load calculation module (DRBTL) so the solution convergence tolerance is more easily specified by the user; and also build the logic into this module so that it can self-diagnose the tolerance required for a given analysis. This task has been completed and was reported in the September 1980, and December 1980 Technical Reports.
- b) to incorporate daylighting calculations with the existing thermal load calculations. The work toward the completion of this objective is underway now. This technical report contains the details of the task accomplished so far toward this end. The additional tasks to be carried out toward achieving daylighting objectives are expected to be completed over the next two reporting periods. Thus by September 1981, there should be a version of DEROB with daylighting capabilities available for technical evaluation by September..
- c) to incorporate specular reflection algorithms into the DRBSOL module. The tasks toward achieving this objective will be undertaken during the September-December 1981 quarter.
- d) to incorporate a multispace cross ventilation algorithm. The tasks will be undertaken during the course of the 1982 calendar year.

The execution of any given task may have to be postponed whenever a correction needs to be implemented in any of the modules.

The schedule of completion of the tasks may be protracted if the level of funding falls below the grant level.

Tasks Completed During the January 1 - March 31, 1981 Quarter

TASK 1 To re-write the Geometric Factors calculations using the circulation integral approach.

This task was needed to be carried out in order to improve the accuracy of the calculation of the geometric factors. Improved accuracy is required for the calculation of daylighting; and also to overcome some inadequacies in the previous calculations which resulted in incorrect values whenever extreme proportions between two surfaces existed.

The circulation integral approach allows also to derive an exact general algorithm for the calculation of the view factor of a surface from a point whose location and orientation of the line of sight are defined.

The mathematical details are enclosed in Section I of the report.

TASK 2 The development of a clipping algorithm.

This task was needed because the circulation integral assigns negative values to all the contributions to the geometric factor from elements of a surface that rest behind the test surface. Physically these contributions should be zero. Thus the clipping algorithm is needed to truncate the partially hidden surface to only that fraction that is visible from the test surface.

The mathematical details are enclosed in Section II of this report.

TASK 3 The development of a set of general space geometry algorithms.

This task was needed in order to complete the clipping algorithm.

TASK 4 To develop an obstruction algorithm.

Previously DEROB contained an algorithm to account for surfaces interfering with the view factor among other surfaces (shading of angle factors.) The development of the circulation integral approach gives us sufficient accuracy in the values of the geometric factors that these results can be used to expedite the illumination of non-shaded walls. In this manner the programs now have the intelligence to know whether a given surface is shaded or not. The result is much faster execution. The mathematical details of obstruction algorithm are summarized in section IV.

TASK 5 Cleaning the resultant DRBGF module to increase the computer memory available.

This task is desirable as a general course to achieve better pro-

gramming, but also to free up space in the module to carry out the calculation in the future of the geometric contributions to the cross ventilation algorithms.

Section V of this report contains the computer listing of the DRBGF module as it appears with these completed algorithms.

Geometric Factors

Definition

Consider an element of surface $dA^{(j)}$ radiating toward another element of surface $dA^{(i)}$. These elements of area are located respectively at points $\underline{r}^{(j)}$ and $\underline{r}^{(i)}$. Let $dE^{(j)}$ stand for the intensity of the radiation at point $\underline{r}^{(i)}$ due to the source at point $\underline{r}^{(j)}$

$$dE^{(j)} = \xi \sigma T_j^4 \underline{R} \frac{\underline{R} \cdot dA^{(j)}}{|R|^4} * \quad (1)$$

Integrating over the surface j one obtains the intensity (and direction) of the total radiation emitted by j and arriving at $\underline{r}^{(i)}$

$$\underline{E}^{(j)} = \xi \sigma T_j^4 \int \underline{R} \frac{\underline{R} \cdot dA^{(j)}}{|R|^3} \quad (2)$$

where we have assumed that the intensity of the radiation source (σT_j^4) is constant throughout the extent of surface j.

The radiation emitted by surface j and intercepted by surface i is

$$\phi_{ji} = \int \underline{E}^{(j)} \cdot dA^{(i)} \quad (3)$$

$$\phi_{ji} = \xi \sigma T_j^4 \int \frac{\underline{R} \cdot dA^{(i)} \underline{R} \cdot dA^{(j)}}{|R|^4} \quad (4)$$

Define the "area" quantity

$$K_{ij} = \xi \int \frac{\underline{R} \cdot dA^{(i)} \underline{R} \cdot dA^{(j)}}{|R|^4} \quad (5)$$

If surface j is a member of a set of n surfaces that completely enclose a space, the sum of the radiation originated at surface j and intercepted by each of the n surfaces must equal the net radiation emitted by surface

*see the end of this section for the calculation of the normalization factor ξ .

$$\text{i.e. } \phi_{j1} + \phi_{j2} + \phi_{j3} + \dots + \phi_{jn} = \sigma T_j^4 A_j \quad (6)$$

$$\therefore \sum_{i=1}^n K_{ij} = A_j \rightarrow \sum_{i=1}^n \frac{K_{ij}}{A_j} = 1 \quad (7)$$

Define the geometric factor G_{ij} as

$$G_{ij} \equiv K_{ij}/A_j \quad (8)$$

It follows that the geometric factors obey two fundamental properties

$$\text{a) } \sum_{i=1}^n G_{ij} = 1$$

$$\text{and since } K_{ij} = K_{ji} \quad (9)$$

$$\text{b) } G_{ij} A_j = G_{ji} A_i$$

Therefore, the total radiation intercepted by surface i due to surface j is

$$\phi_{ji} = \sigma T_j^4 G_{ij} A_j \quad (10)$$

while the radiation intercepted per unit of area is

$$\phi_{ji}/A_i = \sigma T_j^4 (G_{ij} A_j/A_i) \quad (11)$$

* Calculation of the normalization factor ξ

$$\begin{aligned} dA^{(i)} &= R^2 \sin \beta d\alpha d\beta \frac{R}{R} \\ \int dE^{(j)} \cdot dA^{(i)} &= \sigma T_j^4 dA_j \\ dE^{(j)} &= \xi \sigma T_j^4 R \cdot dA^{(j)} R/R^4 \\ R \cdot dA^{(j)} &= R \sin \beta \sin \alpha dA_j \end{aligned}$$

$$\therefore \int d\mathbf{E}^{(j)} \cdot d\mathbf{A}^{(i)} = \xi \sigma T_j^4 \int_0^\pi \int_0^\pi \frac{R^2 \sin \beta \sin \alpha d\mathbf{A}_j R^2 \sin \alpha}{R^4}$$

$$\sigma T_j^4 d\mathbf{A}_j = \xi \sigma T_j^4 \int_0^\pi \sin^2 \beta d\beta \int_0^\pi \sin \alpha d\alpha$$

$\downarrow$
 $\pi/2$

$\downarrow$
 2

$$\therefore 1 = \xi \pi$$

$$\therefore \xi = 1/\pi$$

The General Form of the Geometric Factor

From the defining equation 8 it follows that

$$G_{ij} = \frac{1}{\pi A_j} \int \frac{(d\mathbf{A}_j \cdot \mathbf{R})(d\mathbf{A}_i \cdot \mathbf{R})}{R^4} \quad (\mathbf{R} = \mathbf{r}^{(j)} - \mathbf{r}^{(i)}) \quad (12)$$

which is a double surface integral. This form of the expression has universal validity, but it is untractable to integrate in general. A more tractable form of 12 can be obtained by specializing to plane surfaces and transforming the surface integrals to line integrals through the use of Stoke's theorem:

$$\oint d\mathbf{r} * \mathbf{f} = \iint (d\mathbf{A} \times \nabla) * \mathbf{f} \quad (13)$$

The star (*) can denote dot, cross or ordinary multiplication. In the last case the vector $\mathbf{f}$ becomes a scalar quantity.

The Line Integral Form of the Geometric Factor

The integral

$$I \equiv \pi G_{ij} A_j = \int \frac{d\mathbf{A}_j \cdot \mathbf{R} d\mathbf{A}_i \cdot \mathbf{R}}{R^4} \quad (14)$$

can be re-written in the form

$$I = \int d\mathbf{A}_j \cdot \mathbf{J} \quad (15)$$

where

$$\mathbf{J} = \int \mathbf{R} \frac{\mathbf{R} \cdot d\mathbf{A}_i}{R^4}$$

Noting the vector identity

$$\underline{R} \cdot \frac{\underline{R} \cdot d\underline{A}_i}{R^4} = -\frac{1}{2} (d\underline{A}_i \times \nabla_i) \times \left(\frac{\underline{R}}{R^2} \right) \quad (16)$$

the expression for $\underline{J}$ can be cast in the form of the righthand side of equation 13 (Stoke's theorem)

$$\therefore \underline{J} = -\frac{1}{2} \int (d\underline{A}_i \times \nabla_i) \times \left(\frac{\underline{R}}{R^2} \right) = -\frac{1}{2} \oint \frac{d\underline{r}^{(i)} \times \underline{R}}{R^2} \quad (17)$$

Thus the integral $\underline{I}$ becomes

$$\underline{I} = \int d\underline{A}^{(j)} \cdot \underline{J} = -\frac{1}{2} \int d\underline{A}^{(i)} \cdot \oint \frac{d\underline{r}^{(i)} \times \underline{R}}{R^2} \quad (18)$$

$$\underline{I} = -\frac{1}{2} \oint d\underline{r}^{(i)} \cdot \left(\int \frac{\underline{R} \times d\underline{A}^{(j)}}{R^2} \right)$$

where the order of integration was cyclically permuted to obtain the last form.

Noting the vector identity

$$\frac{\underline{R} \times d\underline{A}^{(i)}}{R^2} = -(d\underline{A}^{(j)} \times \nabla^{(j)}) \text{lu} (R/R_0) \quad (19)$$

(where R_0 is an arbitrary reference length) we can once again apply Stoke's theorem to obtain the line integral for $\underline{I}$

$$\underline{I} = \frac{1}{2} \oint d\underline{r}^{(i)} \cdot \oint d\underline{r}^{(j)} \text{lu} (R/R_0)^2 \quad (2)$$

The Line Integral Statement

The plane surfaces, therefore, the geometric factor takes the general form

$$G_{ij} = \frac{1}{4\pi A_j} \oint d\underline{r}^{(i)} \cdot \oint d\underline{r}^{(j)} \text{lu} (R/R_0)^2 \quad (21)$$

$$R = |\underline{R}|$$

Surfaces Enclosed by Straight Edges

The circulation integrals can be simplified to a discrete sum by specializing the calculation to surface enclosed by straight edges

$$G_{ij} = \frac{1}{4\pi A_j} \sum_l \sum_m P_{ij}^{(l,m)} \quad (22)$$

where the sums are carried out over the edges enclosing each respective surface. The contribution to the integral due to the l th edge of the i th surface and the m th edge of the j th surface is

$$P_{ij}^{(l,m)} = \int dr^{(i)} \cdot \int dr^{(j)} \ln (R/R_0)^2 \quad (23)$$

For simplicity of notation we suppress the indices l and m on the right hand side of equation 23. Remember, therefore, that this last expression represents the contribution of one edge-pair from surface i and j . One edge from surface i and one edge from surface j .

In the sections that follow, equation 23 is integrated explicitly. Simpler notation however is used: the l th edge of the i th surface is represented by prime quantities, while the m th edge of the j th surface by the unprimed quantities.

An Edge Pair

Consider two straight edges in space. The starting and ending points of these space segments are represented respectively by

$$\underline{r}_1 \text{ and } \underline{r}_2$$

and $\underline{r}'_1$ and $\underline{r}'_2$

The orientation of the segments are defined through the unit vectors

$$\begin{aligned}\hat{n} &= (\underline{r}_2 - \underline{r}_1) / \ell \\ \hat{n}' &= (\underline{r}'_2 - \underline{r}'_1) / \ell'\end{aligned}\quad (24)$$

where ℓ and ℓ' are the segment lengths

$$\begin{aligned}\ell &= |\underline{r}_2 - \underline{r}_1| \\ \ell' &= |\underline{r}'_2 - \underline{r}'_1|\end{aligned}\quad (25)$$

The position of an arbitrary point along each segment is given by

$$\begin{aligned}\underline{r} &= \underline{r}_1 + p\hat{n} \quad (0 \leq p \leq \ell) \\ \underline{r}' &= \underline{r}'_1 + p'\hat{n}' \quad (0 \leq p' \leq \ell')\end{aligned}\quad (26)$$

The position of the point in one segment relative to the point in the other segment is

$$\begin{aligned}\underline{R} &= \underline{r}' - \underline{r} \\ &= \underline{\delta} + p'\hat{n}' - p\hat{n} \\ \text{where } \underline{\delta} &= \underline{r}'_1 - \underline{r}_1\end{aligned}\quad (27)$$

and the vector differential $d\underline{r}$ and $d\underline{r}'$ are obtained from equation 26 after noting that the starting points and direction vectors are constant

$$\begin{aligned}d\underline{r} &= \hat{n} dp \\ d\underline{r}' &= \hat{n}' dp'\end{aligned}\quad (28)$$

Thus equation 23 takes the form

$$P = (\hat{n} \cdot \hat{n}') \int_0^{\ell'} dp' \int_0^{\ell} dp \, 1u R^2 \quad (29)$$

where the integrals are scalar quantities. Note that the reference length, R_0 , has been set equal to unity.

Perpendicular Edges

Wherever the edges are perpendicular to each other in space the dot product of their direction vectors is zero, thus the value of P (eq'n 29) is zero.

Parallel Edges

If the edges are parallel (or anti-parallel), then

$$\hat{n} \cdot \hat{n} = + 1$$

equation 29 takes on a simple explicit form.

Noting that

$$\hat{n}' = s\hat{n} \quad (s = \hat{n} \cdot \hat{n}')$$

$$\underline{R} = (\delta_{||} + p's - p) \hat{n} + \underline{\delta}_{\perp} \quad (30)$$

$$\delta_{||} = \underline{\delta} \cdot \hat{n} \quad \& \quad \underline{\delta}_{\perp} = \underline{\delta} - \delta_{||} \hat{n}$$

$$(\underline{\delta}_{\perp} \cdot \hat{n} = 0)$$

equation 29 takes the form

$$P = -s \int_0^{\ell'} dp' \int_{p=0}^{p=\ell} dx \ln(x^2 + \underline{\delta}_{\perp}^2) \quad (31)$$

$$x = \delta_{||} + p's - p \quad (31a)$$

Equation 31 can be integrated directly (use the integrals listed at the bottom of the page) to yield

$$P = -\left\{ F(p, p') \right\}_{\substack{p=0 \\ p'=0}}^{\substack{p=\ell \\ p'=\ell'}} \quad (32)$$

$$F(p, p') = \frac{1}{2} (x^2 + \delta_{\perp}^2) [\ln(x^2 + \delta_{\perp}^2) - 1] - x^2 \\ + 2 \delta_{\perp}^2 \left[\frac{x}{\delta_{\perp}} \tan^{-1} \left(\frac{x}{\delta_{\perp}} \right) - \frac{1}{2} \ln \left(1 + \frac{x^2}{\delta_{\perp}^2} \right) \right] \quad (33)$$

Note the limits

$$F(p, p') \xrightarrow{\delta_{\perp} \rightarrow 0} \frac{1}{2} x^2 [\ln(x^2) - 1] - x^2 \xrightarrow{x \rightarrow 0} 0$$

Note, of course

$$P = F(l, 0) - F(l, l') + F(0, l') - F(0, 0) \quad (34)$$

This equation is calculated in subroutine PRALEL (PaRAILEL)

Integrals Used in these sections:

$$1. \int (luY) dx + (x + \frac{b}{2c}) (luY) - 2x + \frac{\sqrt{4ac-b^2}}{c} \tan^{-1} \left(\frac{2cx+b}{\sqrt{4ac-b^2}} \right)$$

$$Y = a + bx + cx^2 \quad (4ac-b^2 > 0)$$

$$2. \int x lu(x^2 + a^2) dx = \frac{1}{2} (x^2 + a^2) [lu(x^2 + a^2) - 1]$$

$$3. \int \tan^{-1} x dx = x \tan^{-1} x - \frac{1}{2} lu(1+x^2)$$

$$4. \int x \tan^{-1} \left(\frac{a}{x} \right) dx = \frac{1}{2} [(a^2 + x^2) \tan^{-1} \left(\frac{a}{x} \right) + ax]$$

Example

Consider two apposing faces of a cube of unit length.

The corresponding geometric factor as obtained by direct application to equation 31a, 32, and 33 is

$$G_{21} = \frac{1}{\pi} \left[lu \left(\frac{4}{3} \right) + 4 \sqrt{2} \tan^{-1} \left(\frac{1}{\sqrt{2}} \right) \right] - 1 \quad (35)$$

$$\approx .199824896$$

The geometric factor of two adjacent faces of a unit cube is

$$G_{31} = \frac{1}{2} + \frac{1}{\pi} \left[\frac{1}{4} lu \left(\frac{3}{4} \right) - \sqrt{2} \tan^{-1} \left(\frac{1}{\sqrt{2}} \right) \right] \quad (36)$$

$$\approx .200043776$$

Note that summing the geometric factors of the 4 faces adjacent to face 1 and the value of its opposing face we obtain unity

Edge Pairs at Arbitrary Angles

For edge pairs whose direction vectors are neither perpendicular nor parallel to each other, the two integration variables (p and p') of equation 29 can not be combined linearly such as in the case of parallel edges (equations 31 and 31a.) The inside integral can be executed by direct quadrature; the resultant function is then integrated over the outside variable.

Equation 29 can be written in the form

$$P = (\hat{n} \cdot \hat{n}') \int_0^{\ell} dp J(p) \quad (37)$$

$$\text{where } J(p) = \int_0^{\ell'} dp' \, lu R^2 \quad (38)$$

can be integrated exactly using integral #1 listed in the previous page. Since

$$R^2 = V^2 - 2(\underline{V} \cdot \underline{n}') p' + p'^2 \quad (39)$$

$$\text{and } \underline{V} = \underline{\delta} + p\hat{n} \quad \text{and } \underline{\delta} = \underline{r}_1 - \underline{r}_1'$$

$$\begin{aligned} \int dp' lu R^2 = & (p' - \underline{V} \cdot \hat{n}') \, lu [V^2 - 2(\underline{V} \cdot \hat{n}') p' + p'^2] - 2p' \\ & + 2 \sqrt{V^2 - (\underline{V} \cdot \hat{n}')^2} \tan^{-1} \left[\frac{p' - \underline{V} \cdot \hat{n}'}{V^2 - (\underline{V} \cdot \hat{n}')^2} \right] \end{aligned} \quad (40)$$

Grouping the two transcendental terms together, equation 38 becomes

$$J(p) = G(\ell^1, p) - G(0, p) - 2\ell' \quad (41)$$

and equation 37

$$P = (\hat{n} \cdot \hat{n}') \{-2\ell\ell' + \int_0^{\ell} dp [G(\ell', p) - G(0, p)]\} \quad (42)$$

It is convenient for programming to re-state this last expression in the form

$$P = (\hat{n} \cdot \hat{n}') \{-2\ell\ell' - \sum_{i=1}^2 (-1)^i \int_0^{\ell} dp G(p', p)\} \quad (43)$$

$$p' = \frac{1}{2}[1 - (-1)^i]\ell'$$

$$\begin{aligned} \therefore \quad & \text{for } i = 1 \quad p' = \ell' \\ & i = 2 \quad p' = 0 \end{aligned}$$

Except for the special cases already mentioned ($n \cdot n' = \pm 1$) the integral

$$Q = \int_0^{\ell} dp \, G(p', p) \quad (44)$$

is carried out numerically in DEROB

It is convenient to re-express $G(p', p)$ (eq'n 42) in the more compact form

$$G(p', p) = (p' - V_{\parallel}) \ln [V^2 + (p' - V_{\parallel})^2] + 2V \tan^{-1} \left(\frac{p' - V_{\parallel}}{V_{\perp}} \right) \quad (45)$$

$$\begin{aligned} \text{where} \quad & V_{\parallel} = \underline{V} \cdot \hat{n}' \\ \text{and} \quad & \underline{V}_{\perp} = \underline{V} - V_{\parallel} \hat{n}' \quad \cdot \cdot \quad V_{\perp}^2 = V^2 - V_{\parallel}^2 \end{aligned} \quad (46)$$

In this form it is a simple matter to regain the special case for edges that are parallel ($\hat{n} \times \hat{n}' = 0$) since $V_{\perp} = \underline{\delta}$. In this special case the functional dependance of $V_{\perp}$ on the integration variable p disappears. In the general case, however, V retains its dependance on p

The form for $G(p', p)$ in equation 45 is also useful because it allows us to treat explicitly for singularities. If $V_{\perp}$ is not zero then there are no singularities in either of the two terms. If $V_{\perp}$ is zero, then the argument of the $\tan^{-1}$ is singular; however,

$$V_{\perp} \tan^{-1} \left(\frac{p' - V_{\parallel}}{V_{\perp}} \right) \xrightarrow{V_{\perp} \rightarrow 0} 0 \quad (47)$$

The other potential singularity occurs when the argument of the logarithmic function goes to zero. This can happen, however, only if $V = 0$ and $p' = V$; but note that

$$(p' - V_{\parallel}) \ln (p' - V_{\parallel})^2 \xrightarrow{p' \rightarrow V_{\parallel}} 0 \quad (48)$$

In general, therefore, we can summarize the value of $G(p', p)$ as follows:

$$G(p, p') = \begin{cases} = 0 & p' - V_{||} = 0 \\ = (p' - V_{||}) \ln (p' - V_{||})^2 & V_{\perp} = 0 \\ = (p' - V_{||}) \ln [V_{\perp}^2 + (p' - V_{||})^2] + 2V_{\perp} \tan^{-1} \left(\frac{p' - V_{||}}{V_{\perp}} \right) & V_{\perp} \neq 0 \end{cases} \quad \left. \begin{matrix} \\ \\ \\ \end{matrix} \right\} p' - V_{||} \neq 0 \quad (49)$$

Numerical Integration

The integration of $G(p, p')$ (equation 50) over the variable p is carried out by dividing the length ℓ of the un-primed edge in M equal segments of length

$$\Delta p = \ell/M$$

$$p = m\Delta p \quad 0 \leq m \leq M \quad (50)$$

Adopt the convention that

$$G(p, p') = G_m(p') \quad (51)$$

$$\text{or simply} \quad = G_m$$

Therefore, equation 45 becomes

$$Q = \left\{ \frac{1}{2}(G_0 + G_n) + \sum_{m=1}^{M-1} G_m \right\} \Delta p \quad (52)$$

Consequently, equation 43 becomes

$$p = (\hat{n} \cdot \hat{n}') \{-2\ell\ell^1 + Q\} \quad (53)$$

This integration is carried out in subroutine ANYDCT (ANY DirecTion)

Computer Solution

The geometric factor between two surfaces is carried out in DEROB through the use of 4 subroutines:

1. GFDRVR (Geometric Factor DRiVeR) executes equation 22

$$G_{ij} A_i = \frac{1}{4\pi} \sum_{\ell} \sum_m p_{ij}^{(\ell, m)}$$

Given the coordinates of the corners of each surface, this

subroutine returns the value (SUM) $G_{ij}A_j$. The term $P_{ij}^{(\ell,m)}$ for each segment pair is calculated by calling the second subroutine of this sequence SGPAIR.

2. SGPAIR (SeGment PAIR) executes equation 29

$$P_{ij}^{(\ell,m)} = (\hat{n} \cdot \hat{n}') \int_0^{\ell'} dp' \int_0^{\ell} dp \, lu \, R^2$$

Given the coordinates of the beginning and end of the prime and unprimed segments this subroutine returns (TERM)

$P_{ij}^{(\ell,m)}$ to GFDRVR. The path that SGPAIR follows is determined by the value of $S = \hat{n} \cdot \hat{n}'$:

If $S=0$ TERM is set to zero.

If $S=+1$ SGPAIR calls subroutine PRALEL to calculate TERM

If $0 < |S| < 1$ SGPAIR calls ANYDCT to calculate TERM

3. PRALEL (PaRAlleL) executes equation 34

$$P = F(\ell, 0) - F(\ell, \ell') + F(0, \ell') - F(0, 0)$$

which yields the value of TERM for the case where the two segments are parallel.

4. ANYDCT (ANY DireCTion) executes equation 54

$$P = (\hat{n} \cdot \hat{n}') \{-2\ell\ell' + Q\}$$

which yields the value of TERM for the general case when the segments make an arbitrary angle with each other.

A service subroutine to calculate the dot product of vectors is required for this sequence.

The Additive Property of Geometric Factors

The geometric factor between surfaces i and j is given by

$$I = G_{ij} A_j \pi = \int \frac{(\underline{R} \cdot d\underline{A}_i)(\underline{R} \cdot d\underline{A}_j)}{|\underline{R}|^4} \quad (54)$$

where the integrals are carried out over the entire surface area of each surface.

If the contribution to these integrals due to the holes shown in the figure is represented by the expression

$$\Delta I = \int_{\text{over "holes"}} \frac{(\underline{R} \cdot d\underline{A}_i)(\underline{R} \cdot d\underline{A}_j)}{|\underline{R}|^4} \quad (55)$$

and the remaining contribution to the integral with the symbol I' , then

$$I = I' + \Delta I \quad (56)$$

Geometric Factor Between Two Partially Obstructed Surfaces

If two surfaces are partially obstructed in such a way that the "holes" shown in the figure above represent the obstructed line of sight between the two surfaces, the geometric factor of these surfaces must exclude that contribution arising from the shaded areas (holes)

$$I' = I - \Delta I \quad (57)$$

Since both I and ΔI can be calculated directly then I' can be determined from the information known. Therefore, the geometric factor of two surfaces partially obstructed is given by

$$G'_{ij} = G_{ij} - \frac{1}{\pi A} \Delta I \quad (58)$$

where G_{ij} is the geometric factor of the two surfaces when they are treated as un-obstructed and ΔI is the integral contribution to view factors of the mutually shaded areas.

The Geometric Factor Between a Point and an Extended Surface

Under some conditions, such as for the calculation of the sky factor at a point, it is useful to consider the geometric factor between a point and an extended surface.

The calculation is an abbreviated version of the one for the Geometric Factors. Start with the integral of equation 2

$$g = \frac{1}{\pi} \hat{s} \cdot \int \frac{\underline{R} \cdot d\underline{A}}{R^4} \quad (57)$$

where $\hat{s}$ is the vector normal of the surface on which the measuring point is located

Using the vector identity in equation 16 and Stoke's theorem equation 59 can be expressed as a circulation integral

$$g = -\frac{1}{2\pi} \hat{s} \cdot \oint \frac{d\underline{r}' \times \underline{R}}{R^2} \quad (60)$$

Assuming that the circulation takes place about a series of straight edges, equation 60 becomes a discrete sum of the integrals of each edge

$$g = -\frac{1}{2\pi} \hat{s} \cdot \sum_k \int \frac{d\underline{r}^{(k)} \times \underline{R}}{R^2} \quad (61)$$

The integral for a given edge can be solved by direct quadrature. Note the

$$\underline{r}^{(k)} = \underline{r}_1^{(k)} + p \hat{n}^{(k)} \quad (62)$$

where $\underline{r}_1^{(k)}$ is the starting position of the k^{th} edge and $\hat{n}^{(k)}$ the orientation unit vector of the edge. The scalar quantity p is the position along the edge.

$$\therefore \quad d\underline{r}^{(\kappa)} = \hat{n}^{(\kappa)} dp; \quad \underline{R} = (\underline{r} - \underline{r}_1^{(\kappa)}) - p\hat{n}^{(\kappa)}$$

and

$$\int \frac{d\underline{r}^{(\kappa)} \times \underline{R}}{R^2} = \hat{n}^{(\kappa)} \times \underline{\delta}^{(\kappa)} \int \frac{dp}{[\delta^{(\kappa)^2} - 2\underline{\delta}^{(\kappa)} \cdot \hat{n}^{(\kappa)} p + p^2]} \quad (63)$$

which can be integrated directly

$$\int \frac{d\underline{r}^{(\kappa)} \times \underline{R}}{R^2} = \hat{n}^{(\kappa)} \times \underline{\delta}^{(\kappa)} [Q^{(\kappa)}(\ell_k) - Q^{(\kappa)}(0)] \quad (64)$$

where

$$Q(p) = \frac{1}{q} \tan^{-1} \left[\frac{p + \underline{\delta}^{(\kappa)} \cdot \hat{n}^{(\kappa)}}{q} \right]$$

$$\underline{\delta}^{(\kappa)} = \underline{r} - \underline{r}_1^{(\kappa)}$$

$$q = \sqrt{(\delta^{(\kappa)})^2 - (\underline{\delta}^{(\kappa)} \cdot \hat{n}^{(\kappa)})^2}$$

ℓ_k = length of kth edge

$$\therefore \quad \underline{g} = \frac{1}{2\pi} \hat{s} \cdot \sum_k (\hat{n}^{(\kappa)} \times \underline{\delta}^{(\kappa)}) [Q(\ell_k) - Q(0)]$$

The Clip Subroutine:

The Problem of One Surface Partially Hidden from the Other

The Direction of a Surface is determined by the unit vector normal to the surface. The local coordinate system of the surface has a z-axis defined by the vector normal to the surface.

Whether a Point can be "Seen" from a Surface is determined by the position of the point along the z-axis of the surface local coordinate system

i.e. the triad (x, y, z) defines the local coordinate system of the surface defined by the vector normal $\hat{n}$. Point 1 has a positive z-coordinate; therefore it can be seen from the surface. Point 2 has a negative z-coordinate, therefore it can not be seen from the surface

Subroutine SIVEO ("yes, I see" in Spanish) carried out this calculation:

Given the point and the surface, SIVEO calculates z (RH).

If $RH > 0$ ISEE=1

$RH \leq 0$ ISEE=0

SIVEO returns the index ISEE

A Surface in Partial View of Another Surface

For the surfaces shown in the adjacent figure, the prime surface can see all the corners of the unprime surface. The unprime surface, however, can only see corners 1 and 2 of the prime surface; and it can not see points 3 and 4.

5. Calculate the total number of corners that can be seen. If it is zero, none can be seen; thus $KSEE=0$ and the geometric factor is zero. If all the corners can be seen, then none need to be replaced.
6. Take one segment at a time
i.e. segment defined by corners 1 and 2
since $J1 = JSEEN(1)=1$
and $J2 = JSEEN(2)=1$ then $J1=J2$ therefore the segment defined by points 1 and 2 does not intersect the line L (defined by the intersection of the two planes). Consider then the next segment.

Segment 2 is defined by corners 2 and 3

since $J1 = JSEEN(2)=1$

and $J2 = JSEEN(3)=0$ then $J1 \neq J2$, therefore this segment intersects line L. Find the coordinates of the intersection by calling subroutine TWOLIN.

Store the coordinates in the scratch array $XRP(3,K)$ and remember that point 3 is to be replaced by setting $IRP(3)=1$.

Then consider the next segment, the one defined by corners 3 and 4.

since $J1=JSEEN(3)=0$

and $J2=JSEEN(4)=0$ $J1=J2$ then this segment does not intersect line L.

We proceed to the next segment,

The next segment is defined by corners 4 and 1.

since $J1 = JSEEN(4)=0$

and $J2 = JSEEN(1)=1$, then $J1 \neq J2$;

therefore, this segment intersects line L. Calculate the coordinates of the intersection using subroutine TWOLIN. Store these coordinates in the array

$XRP(4,K)$

and remember that point 4 is to be replaced by setting $IRP(4)=1$.

7. When all the edges have been thus calculated proceed to replace the corners that need to be replaced.

i.e. since $IRP(1)=0$ and $IRP(2)=0$

corners 1 and 2 are not replaced

since $IRP(3)=1$ then $PNTS(3,K)=XRP(3,K)$

and $IRP(4)=1$ then $PNTS(4,K)=XRP(4,K)$

Space Geometry

The Equation of a Plane

If $\underline{R}$ is a point known to be on the plane, and $\hat{n}$ is the vector normal to the plane, then a point $\underline{r}$ is on the plane provided the vector $(\underline{r} - \underline{R})$ is perpendicular to the vector normal to that plane

$$(\underline{r} - \underline{R}) \cdot \hat{n} = 0$$

The Equation of a Line

If $\underline{r}_0$ is a point known to be on the line, and $\underline{L}$ is the unit vector defining the direction of the line, then a point $\underline{r}$ is on the line provided the vector $(\underline{r} - \underline{r}_0)$ is parallel to the direction vector of the line

$$(\underline{r} - \underline{r}_0) \times \underline{L} = 0$$

The Intersection of Two Planes

The intersection of two planes define a line. Such a line is uniquely defined provided the two planes are not parallel.

Consider the two planes

$$(\underline{r} - \underline{R}) \cdot \hat{n} = 0$$

$$(\underline{r} - \underline{R}') \cdot \hat{n}' = 0$$

Their line of intersection must satisfy both equations simultaneously.

Define the vector $\underline{L} = \hat{n} \times \hat{n}'$

$$\text{Therefore } \underline{r} \times \underline{L} = \hat{n}(\hat{n}' \cdot \underline{r}) - \hat{n}'(\hat{n} \cdot \underline{r})$$

$$\text{But } \hat{n}' \cdot \underline{r} = \hat{n}' \cdot \underline{R}' \quad \text{and} \quad \hat{n} \cdot \underline{r} = \hat{n} \cdot \underline{R}$$

$$\text{Therefore } \underline{r} \times \underline{L} = \underline{F}$$

which is the equation of the line of intersection since the defining property of $\underline{F}$

$$\underline{F} = \hat{n}(\hat{n}' \cdot \underline{R}') - \hat{n}'(\hat{n} \cdot \underline{R})$$

requires that $\underline{F}$ be perpendicular to $\underline{L}$. Formally, therefore, $\underline{F}$ can be written in the form $\underline{F} = \underline{r} \times \underline{L}$ and the equation of the line can be written in the standard form $(\underline{r} - \underline{r}_0) \cdot \underline{L} = 0$.

The solution holds, of course, provided the planes are not parallel, i.e. $\underline{L} = \hat{n} \times \hat{n} \neq 0$.

The Intersection of Two Lines

Two coplanar lines intersect on a point provided the two lines are not parallel.

Consider the two lines

$$\underline{r} \times \underline{L} = \underline{F} \quad (\underline{F} = \underline{r} \times \underline{L})$$

$$\underline{r} \times \underline{L}' = \underline{F}' \quad (\underline{F}' = \underline{r}' \times \underline{L}')$$

These two lines are coplanar provided the vector $(\underline{r} - \underline{r}')$ is perpendicular to both lines

$$(\underline{r} - \underline{r}') \cdot (\underline{L} \times \underline{L}') = 0 \quad (\text{coplanarity condition})$$

provided the trivial solution of the two lines being parallel is not met

$$\underline{L} \times \underline{L}' \neq 0 \quad (\text{requirement that the two lines not be parallel})$$

The coplanarity requirement follows the equations of the lines

$$\underline{r} \times \underline{L} \cdot \underline{L}' = \underline{F} \cdot \underline{L}'$$

$$- \underline{L}' \times \underline{r} \cdot \underline{L} = -\underline{F}' \cdot \underline{L} \rightarrow \underline{F}' \cdot \underline{L} + \underline{F} \cdot \underline{L}' = 0$$

since $\underline{F}' \cdot \underline{L} = \underline{r}' \times \underline{L}' \cdot \underline{L} = \underline{L}' \times \underline{L} \cdot \underline{r}'$

$$\underline{F} \cdot \underline{L}' = \underline{r} \times \underline{L} \cdot \underline{L}' = \underline{L} \times \underline{L}' \cdot \underline{r}$$

$$\therefore \underline{F}' \cdot \underline{L} + \underline{F} \cdot \underline{L}' = (\underline{L}' \times \underline{L}) \cdot (\underline{r}' - \underline{r}) = 0$$

The point of intersection is calculated as follows

$$\begin{aligned} \underline{r} \times \underline{L} &= \underline{F} \rightarrow \underline{F}' \times (\underline{r} \times \underline{L}) = \underline{F}' \times \underline{F} \\ &\rightarrow \underline{r}(\underline{F}' \cdot \underline{L}) - \underline{L}(\underline{F}' \cdot \underline{r}) = \underline{F}' \times \underline{F} \end{aligned}$$

Therefore $\underline{r} = \underline{F} \times \underline{F} / (\underline{F}' \cdot \underline{L}) \quad \underline{F}' \cdot \underline{L} \neq 0$

Therefore the two lines intersect if they are coplanar, provided they are not parallel

$$(\underline{F}' \cdot \underline{L} \neq 0)$$

Note $\underline{F}' \cdot \underline{L} = \underline{r}' \times \underline{L}' = (\underline{L}' \times \underline{L}) \cdot \underline{r}'$

The point where the two lines intersect is given by the expression

$$\underline{r} = \frac{\underline{F}' \times \underline{F}}{\underline{F}' \cdot \underline{L}}$$

However, if either $\underline{F}'$ or $\underline{F}$ is a null vector (zero magnitude) this expression yields an undetermined result. When this happens a well-defined result can be obtained by transforming all the vectors to a different coordinate system. In fact, it is enough to use a translation transformation along a direction perpendicular to both $\underline{L}$ and $\underline{L}'$.

Move all the position vectors by an amount $\underline{s}$

$$\underline{p} = \underline{r} + \underline{s}$$

$$\underline{p}_i = \underline{r}_i + \underline{s} \quad \underline{p}'_i = \underline{r}'_i + \underline{s}$$

$$\begin{aligned} \text{Since } \underline{F} &= \underline{r} \times \underline{L} & \underline{G} &= \underline{p}_i \times \underline{L} \\ & & &= (\underline{r}_i + \underline{s}) \times \underline{L} = \underline{r}_i \times \underline{L} + \underline{s} \times \underline{L} \\ & & &= \underline{F} + \underline{s} \times \underline{L} \end{aligned}$$

$$\text{Likewise} \quad \underline{G}' = \underline{F}' + \underline{s} \times \underline{L}'$$

The equations of the lines are then transformed into the form

$$\underline{p} \times \underline{L} = \underline{G} \quad \text{and} \quad \underline{p} \times \underline{L}' = \underline{G}'$$

$$\therefore \underline{p} = \underline{G}' \times \underline{G} / (\underline{G}' \cdot \underline{L})$$

$$\text{and} \quad \underline{r} = \underline{G}' \times \underline{G} / (\underline{G}' \cdot \underline{L}) - \underline{s}$$

The Intersection of a Line with a Plane

A line intersects with a plane on a point provided the line is not perpendicular to the vector normal to the plane

The equation for the plane is

$$(\underline{r} - \underline{R}) \cdot \hat{n} = 0$$

and for the line

$$(\underline{r} - \underline{r}_1) \times \underline{L} = 0$$

This line and plane have a well-defined intersection provided $\hat{n} \cdot \underline{L} \neq 0$.

The point of intersection is obtained as follows

$$\begin{aligned} \underline{r} \cdot \underline{L} &= \underline{r}_1 \times \underline{L} \rightarrow \hat{n} \times (\underline{r} \times \underline{L}) = \hat{n} \times (\underline{r}_1 \times \underline{L}) \\ &\rightarrow \underline{r}(\hat{n} \cdot \underline{L}) - \underline{L}(\underline{r} \cdot \hat{n}) = \hat{n} \times (\underline{r}_1 \times \underline{L}) \end{aligned}$$

but $\underline{r} \cdot \hat{n} = \underline{R} \cdot \hat{n}$ (equation of the plane)

$$\therefore \underline{r}(\hat{n} \cdot \underline{L}) - \underline{L}(\underline{R} \cdot \hat{n}) = \hat{n} \times (\underline{r}_1 \times \underline{L})$$

$$\therefore \underline{r} = \underline{r}_1 + \frac{(\underline{R} - \underline{r}_1) \cdot \hat{n}}{\underline{L} \cdot \hat{n}} \underline{L}$$

Transformation of Geometric Factors

Transformation to the Local Coordinate System

The local coordinate system of a plane has its z-axis coincident with the normal to the plane. Any point on the plane, therefore, can be described in two dimensions in its local coordinate system. Consider three corners of a bounded plane, such as the surface of a wall.

The positions of these corners relative to the building coordinate system are given respectively by $\underline{r}_1$, $\underline{r}_2$, $\underline{r}_3$. The direction of the first edge is defined by the vector

$$\underline{p}_1 = \underline{r}_2 - \underline{r}_1$$

and of the second edge by $\underline{p}_2 = \underline{r}_3 - \underline{r}_2$.
 The vector normal to the surface is given by

$$\hat{k}' \propto \underline{p}_1 \times \underline{p}_2$$

The local x axis can be made to coincide with the first edge

$$\hat{i}' = \underline{p}_1 / |\underline{p}_1|$$

If the local coordinate system is to be orthonormal, then the y axis is defined by

$$\hat{j}' = \hat{k}' \times \hat{i}'$$

The translational transformation is carried out by using any point known to lie on the plane. DEROB uses the first corner of the surface. This is the reference point $\underline{R} \equiv \underline{r}_1$.

Let $\underline{r}$ be a point in space as described in the building coordinate system; $\underline{r}'$ is the same point described from the local coordinate system.

$$\underline{r} = \underline{R} + \underline{r}'$$

if $\underline{r} = x_i \hat{e}_i$, $\underline{R} = R_i \hat{e}_i$, and $\underline{r}' = x'_i \hat{e}'_i$

$$\begin{aligned} \hat{e}_1 &= \hat{i} & \hat{e}'_1 &= \hat{i}' \\ \hat{e}_2 &= \hat{j} & \hat{e}'_2 &= \hat{j}' \\ \hat{e}_3 &= \hat{k} & \hat{e}'_3 &= \hat{k}' \end{aligned}$$

We can convert from the building to the local coordinate system by solving for x'_i

$$x'_i = U_{ij}(x_j - R_j)$$

where $U_{ij} = \hat{e}'_i \cdot \hat{e}_j$

is the transformation matrix. Since it is a unitary matrix, its inverse is equal to its transform; therefore, we can carry out the inverse transformation through

$$x_i = R_i + U_{ji}x'_j$$

The Transformation for the Other Side of the Wall can be carried out in identical fashion. However, since the two sides are intimately related, it is advantageous to define the transformation matrix only for side 1, and modify its values appropriately for the side.

The evaluation of the transformation matrix components is carried out in subroutine TRNSFM (TRaNSForm). The transformation from one coordinate system to another is carried out in subroutine CTRNSF (Coordinate TRaNSFormation).

Subroutine OBSTRT (Obstruct) Logic

Given surfaces I and J of Volume IV

1. identify the surfaces in the overall surface counter (I0, and J0); and identify the number of grid elements of each surface (N1 and N2).
2. sweep over the grid elements of each surface. For each pair of grid elements:
3. determine the coordinates of the center of each grid element in the building coordinate system. Call them POINT1 ($\underline{p}_1$) and POINT2 ($\underline{p}_2$)
4. calculate the direction vector of the line going from $\underline{p}_1$ to $\underline{p}_2$

$$\hat{\underline{L}} = (\underline{p}_2 - \underline{p}_1) / |\underline{p}_2 - \underline{p}_1|$$

(Remember that the equation of the line is given by $(\underline{r} - \underline{p}_1) \times \underline{L} = 0$)

5. test to see if anything (surfaces) obstructs (intersects) the line from $\underline{p}_1$ to $\underline{p}_2$.
6. take all the other surfaces facing into volume IV; one at a time until we find one that intercepts the line from $\underline{p}_1$ to $\underline{p}_2$

- a) find the point of intersection of the line and the plane ($\underline{r}$)
- b) find if the point $\underline{r}$ is between pts $\underline{p}_1$ and $\underline{p}_2$

$$(\underline{r} - \underline{p}_1) \cdot \underline{L} > 0$$

$$(\underline{r} - \underline{p}_2) \cdot \underline{L} < 0$$

- c) find if the point $\underline{r}$ is bound within the edges of the surface

7. If 6b and 6c are answered yes, then the intersection exists, and the geometric factor of surfaces I and J is reduced by the geometric factor between the grid elements that were obstructed.

PROGRAM DRBGF5(OUTPUT,TAPE1,TAPE5)

C..DEVELOPMENT VERSION (FEB,1981).GEOMETRIC FACTORS ARE CALCULATED
 C..WITH CIRCULATION INTEGRALS. ONLY ONE TRANSFORMATION MATRIX PER
 C..WALL IS USED.ALL IN PREPARATION FOR SPECULAR REFLECTION CALCULATIONS
 C..AND MULTISPACE DAYLIGHTING CALCULATIONS.

```

2.      COMMON/BLKVOL/CNTV(9,3),VOL(9)
B      3.      COMMON/BLKA/AR(154),SUMG(154)
B      4.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      5.      COMMON/BLKN/NOVOL,NOWALS,NOPTS(154)
B      6.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      7.      COMMON/BLK1/IVOL(154)
B      8.      INTEGER RNUM,ENUM
B      9.      CALL DAVID
B     10.      CALL SRFINF
B     11.      DO 19 IV=1,NOVOL
B     12.          JM=IWC(IV)
B     13.          IVM=IV
B     14.          PRINT 212,JM,IVM
B     15. 212  FORMAT(1H1,/,2X,*,THERE ARE*,I3,*, SURFACES FACING VOLUME*,I3)
B     16.          ART=0.
B     17.          ARG=0.
B     18.          DO 198 JS=1,JM
B     19.              JW=(IVC(IV,JS)+1)/2
B     20.              JG=2*JW-1
B     21.              JI=JG+1
B     22.              ART=ART+AR(JI)
B     23.              IF(IVOL(JG).EQ.0)ARG=ARG+AR(JG)
B     24.              IF(IVOL(JG).EQ.-1)ARG=ARG+AR(JG)
B     25.              PRINT 453,JS,IVC(IV,JS),JW,AR(JI)
B     26. 453  FORMAT(X,*,SURFACE*,I3,*, IS BLDG COUNTER*,I4,
B     27.          1X,*, WALL*,I3,*, AREA =*,F7.2)
B     28.          CALL POINTR(1,IV,JS,JS,JJ)
B     29.          SUMG(JS)=EOM(JJ)=0.
B     30. 198  CONTINUE
B     31.      PRINT 998,ART,ARG,ARG,VOL(IV)
B     32. 998  FORMAT(/,2X,*,TOTAL AREA ENCLOSING THIS VOLUME*,F7.1,
B     33.          1,/,      2X,*,OF THIS*,F7.1,*, IS CONNECTED TO THE AMBIENT AIR*,
B     34.          2,/,      2X,*, AND      *,F7.1,*, IS CONNECTED TO THE GROUND*,
B     35.          3,/,      2X,*, THE VOLUME ENCLOSED IS*,F9.2)
B     33.      NR=IWC(IV)-1
B     34.      CALL GEM(NW,NR,IV)
B     35.      CALL ITER(NW,IV)
B     36. 19  CONTINUE
B     37.      CALL OUTDTA
B     38.      END

```

```

SUBROUTINE SRFINF
B      2.      DIMENSION CNTR(3)
B      3.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      4.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      5.      COMMON/BLKA/AR(154),SUNG(154)
B      6.      COMMON/BLKN/NOVOL,NOVALS,NOPTS(154)
B      7.      COMMON/BLKP/PCN(154,3),PTS(154,10,3)
B      8.      COMMON/BLKT/P(10,3)
B      9.      COMMON/BLKI/IVOL(154)
B     10.      DIMENSION XXL(10,3)
B     11.      DIMENSION POINT(3),TOINT(3)
B     12.      READ(1,100)NOVALS,(NOPTS(I),I=1,NOVALS)
B     13.      DO 1 J=1,9
B     14. 1      IWC(J)=0
B     15.      NOVOL=0
B     16. 100  FORMAT(16I5)
B     17.      DO 6 I=1,NOVALS
B     18.      READ(1,101)GSKY(I)
B     19.      READ(1,100)IVOL(I)
B     20.      IV=IVOL(I)
B     21.      IF(IV.GT.NOVOL)NOVOL=IV
B     22.      IF(IV.LE.0)GO TO 7
B     23.      IK=IWC(IV)=IWC(IV)+1
B     24.      IVC(IV,IK)=I
B     25.      IVCR(I)=IK
B     26.      NUMPT=NOPTS(I)
B     27.      READ(1,101)((PTS(I,J,K),K=1,3),J=1,NUMPT)
B     28. 101  FORMAT(9F7.2)
B     29.      DO 5 J=1,NUMPT
B     30.      DO 5 K=1,3
B     31. 5      P(J,K)=PTS(I,J,K)
B     32. 20  CONTINUE
B     33.      CALL TRNSFM(I)
B     34.      DO 3 J=1,NUMPT
B     35.      DO 2 K=1,3
B     36. 2      POINT(K)=PTS(I,J,K)
B     37.      CALL CTRNSM(1,1,POINT,TOINT)
B     38.      DO 22 K=1,3
B     39.      XXL(J,K)=TOINT(K)
B     40. 22  CONTINUE
B     41. 3  CONTINUE
B     42.      CALL AREAS(CNTR,XXL,NUMPT,ARE)
B     43.      CNTR(3)=0.
B     44.      CALL CTRNSM(2,I,POINT,CNTR)
B     45.      DO 501 K=1,3
B     46. 501  PCN(I,K)=POINT(K)
B     47.      IF(ABS(ARE).LT.1.E-5) GO TO 10
B     48.      IF(ARE.GT.0.) GO TO 4
B     49.      CALL CYCLY(NUMPT)
B     50.      GO TO 20
B     51. 4  CONTINUE
B     52. 6  AR(I)=ARE
B     53.      CALL VOLUME
B     54.      RETURN
B     55. 10  CONTINUE
B     56.      PRINT 209,I
B     57. 209  FORMAT(2X,*AREA OF SURFACE*,15,* IS ZERO CHECK THE
```

1RESPECTIVE DATA FOR THIS SURFACE*)

58.

STOP202

59.

END

```

SUBROUTINE GEM(NW,NR,IV)
B      2.      COMMON/BLKA/AR(154),SUM6(154)
B      3.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      4.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      5.      DIMENSION SUMA(50)
B      6.      PI=3.141592654
B      7.      NW=IWC(IV)
B      8.      NR=NW-1
B      9.      DO 11 IC=1,NR
B
B      10.     I=IVC(IV,IC)
B      11.     NE=IC+1
B      12.     CALL LCLPNT(1,I,1)
B      13.     DO 10 JC=NE,NW
B      14.     J=IVC(IV,JC)
B      15.     CALL POINTR(1,IV,IC,JC,IJ)
B      16.     EOM(IJ)=0.
C      IF(SHADIJ.EQ.0.)GO TO 10
B      17.     CALL LCLPNT(2,J,1)
B      18.     CALL FRMSTR
B      19.     CALL REPLCE(KSEE,I,J)
B      20.     IF(KSEE.EQ.0)GO TO 10
B      21.     CALL SFORVR(SUM,I,J)
B      22.     EOM(IJ)=SUM
B      23.     PRINT 1001,IV,IC,JC,IJ,SUM/AR(J),SUM/AR(I)
B      24. 1001  FORMAT(X,*IV,IC,JC,IJ*,4I3,* GEOM=*,2F9.3)
B      25. 10  CONTINUE
B      26. 11  CONTINUE
C..
C..SUM THE GEOMETRIC FACTORS OF EACH SURFACE
B      27.     DO 13 JS=1,NW
B      28.     SUMA(JS)=0.
B      29.     J=IVC(IV,JS)
B      30.     DO 12 IS=1,NW
B      31.     IF(IS.EQ.JS)GO TO 12
B      32.     CALL POINTR(1,IV,IS,JS,IJ)
B      33.     SUMA(JS)=SUMA(JS)+EOM(IJ)
B      34. 12  CONTINUE
B      35.     SUMA(JS)=SUMA(JS)/AR(J)
B      36.     PRINT 1002,JS,SUMA(JS),(SUMA(JS)-1.)
B      37. 1002  FORMAT(X,*SUMA(*,I2,*)=*,F9.3,F12.7)
B      38. 13  CONTINUE
C..
C..TEST FOR OBSTRUCTIONS IN THE LINE OF SIGHT FOR THOSE SURFACES
C..WHOSE SUMA IS GREATER THAN 1.
B      39.     DO 15 I=1,NR
B      40.     IF(ABS(SUMA(I)-1.).LT.1.E-4)GO TO 15
B      41.     IO=IVC(IV,I)
B      42.     CALL VALGEN(1,IO)
B      43.     IP=I+1
B      44.     DO 14 J=IP,NW
B      45.     IF(ABS(SUMA(J)-1.).LT.1.E-4)GO TO 14
B      46.     CALL POINTR(1,IV,I,J,IJ)
B      47.     IF(EOM(IJ).LT.1.E-5)GO TO 14
B      48.     JO=IVC(IV,J)
B      49.     CALL VALGEN(2,JO)
B      50.     CALL OBSTRT(I,J,IV,NW)
B      51. 14  CONTINUE

```

MNF I=DRBGF5 J E Z

V-6 UT2D 6600 16

B	52. 15	CONTINUE
B	53.	RETURN
B	54.	END


```

SUBROUTINE ITER(NWL,IVL)
B      2.      COMMON/BLKA/AR(154),SUMG(154)
B      3.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      4.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      5.      COMMON/SOMF/SOM(400),VIEWF(400)
B      6.      IM=0
B      7.      NW=NWL
B      8.      IV=IVL
B      9.      CALL CHNGVF(1,NW,IV)
B     10. 401   IM=IM+1
B     11.      DO 601 IC=1,NW
B     12.          I=IVC(IV,IC)
B     13.          SUMG(IC)=0.
B     14.          DO 599 JC=1,NW
B     15.              CALL PONTR1(1,IV,IC,JC,IJ)
B     16.              J=IVC(IV,JC)
B     17.              SUMG(IC)=SUMG(IC)+SOM(IJ)*AR(J)
B     18. 599   CONTINUE
B     19.          IF(SUMG(IC).GT.0.)GO TO 722
B     20.          IVM=IV
B     21.          IS=(IVC(IV,IC)+1)/2
B     22.          PRINT 723,IS,IVM
B     23. 723   FORMAT(X,*ABORT WALL*,I3,* AND OTHER WALLS THAT HAVE*,
1/5X,*A SIDE FACING VOLUME*,I3,* MAY BE IMPROPERLY ORIENTED*)
B     24.          STOP201
B     25. 722   CONTINUE
B     26. 601   CONTINUE
B     27.          DO 13 JC=1,NW
B     28.              DIFT=0.
B     29.              J=IVC(IV,JC)
B     30.              DO 18 IC=1,NW
B     31.                  I=IVC(IV,IC)
B     32.                  CALL PONTR1(1,IV,JC,IC,JI)
B     33.                  CALL PONTR1(1,IV,IC,JC,IJ)
B     34.                  VIEWF(JI)=SOM(IJ)*AR(J)/SUMG(IC)
B     35.                  DIF=AR(I)-SUMG(IC)
B     36.                  ADIF=ABS(DIF/AR(I))
B     37.                  IF(ADIF.GT.DIFT)DIFT=ADIF
B     38. 18     CONTINUE
B     39. 13     CONTINUE
B     40.          DO 701 IC=1,NW
B     41.              I=IVC(IV,IC)
B     42.              SUMG(IC)=0.
B     43.              DO 699 JC=1,NW
B     44.                  J=IVC(IV,JC)
B     45.                  CALL PONTR1(1,IV,IC,JC,IJ)
B     46.                  CALL PONTR1(1,IV,IC,JC,KC)
B     47.                  SUMG(IC)=SUMG(IC)+VIEWF(IJ)*AR(J)
B     48.                  SOM(KC)=VIEWF(IJ)
B     49. 699   CONTINUE
B     50. 701   CONTINUE
B     51.          IF(DIFT.GT.1.E-4)GO TO 401
B     52.          PRINT 3311,(JM,JM=1,NW)
B     53. 3311  FORMAT(/15X,* VIEW OR ANGLE OR GEOMETRIC FACTORS*,/,5X,16I6,/)
B     54.          DO 129 IOM=1,NW
B     55.              CALL PONTR1(1,IV,IOM,1,JM1)
B     56.              CALL PONTR1(1,IV,IOM,NW,JM2)

```

B	57.	PRINT 139,IOM,(SOM(JM),JM=JM1,JM2)
B	58. 139	FORMAT(2X,I3,X,15F6.2)
B	59. 129	CONTINUE
B	60.	CALL CHNGVF(2,NW,IV)
B	61.	RETURN
B	62.	END

SUBROUTINE OUTDTA

```
B      2.      COMMON/BLKVOL/CNTV(9,3),VOL(9)
B      3.      COMMON/BLKN/NOVOL,NOVALS,NOPTS(154)
B      4.      COMMON/BLKP/PCN(154,3),PTS(154,10,3)
B      5.      COMMON/BLKU/U(77,3,3)
B      6.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      7.      COMMON/BLKV/VAL(2,100,2),NVAL(2),NP
B      8.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      9.      COMMON/BLKA/AR(154),SUMG(154)
B     10.      COMMON/BLK1/IVOL(154)
B     11.      WRITE(5)NOVALS,NP,(NOPTS(I),
1          GSKY(I),AR(I),IVOL(I),IVCR(I),I=1,NOVALS),
B     12.      NSLD=NOVALS/2
B     13.      WRITE(5)((U(I,J,K),K=1,3),J=1,3),I=1,NSLD)
B     14.      WRITE(5)NOVOL
B     15.      DO 2 IV=1,NOVOL
B     16.      WRITE(5)(CNTV(IV,LV),LV=1,3),VOL(IV)
B     17.      WRITE(5)NCU(IV),IWC(IV)
B     18.      IW=IWC(IV)
B     19.      WRITE(5)(IVC(IV,IC),IC=1,IW)
B     20.      IWL=IW-1
B     21.      DO 999 I1=1,IWL
B     22.      I1P=I1+1
B     23.      DO 998 I2=I1P,IW
B     24.      CALL POINTR(2,IV,I1,I2,I12)
B     25.      WRITE(5)GEOM(I12)
B     26. 998    CONTINUE
B     27. 999    CONTINUE
B     28. 2      CONTINUE
B     29.      DO 3 I=1,NOVALS
B     30.      JX=NOPTS(I)
B     31.      DO 4 J=1,JX
B     32.      DO 5 K=1,3
B     33.      WRITE(5)PTS(I,J,K)
B     34. 5      CONTINUE
B     35. 4      CONTINUE
B     36. 3      CONTINUE
B     37.      RETURN
B     38.      END
```

SUBROUTINE VALGEN(LCL,IL)

C..FIND THE NUMBER AND COORDINATES OF GRID ELEMENTS OF SURFACE I
 C..THIS INFO IS NEEDED FOR OBSTRUCTION CALC BETWEEN SURFACE PAIRS
 C..THUS, LC IS =1 OR =2 DEPENDING WHICH MEMBER OF THE PAIR IS I.

```

B      2.      COMMON/BLKA/AR(154),SUMG(154)
B      3.      COMMON/BLKN/NOVOL,NOWALS,NOPTS(154)
B      4.      COMMON/BLKP/PCN(154,3),PTS(154,10,3)
B      5.      COMMON/BLKV/VAL(2,100,2),NVAL(2),NP
B      6.      COMMON/BLKV2/DELX(2),DELY(2)
B      7.      DIMENSION POINT(3),TOINT(3)
B      8.      DIMENSION XX(154),YX(154),XLP(10,2)
B      9.      LC=LCL
B     10.      I=IL
B     11.      NPMAX=8
B     12.      NVMIN=9

```

C..LET THE NUMBER OF GRID PARTITIONS GO FROM 3 TO NPMAX IN ORDER
 C..TO HAVE NO LESS THAN NVMIN GRID SAMPLING POINTS.

```

B     13.      NP=2
B     14. 90    NP=NP+1
B     15.      IF(NP.GT.NPMAX)GO TO 95
B     16.      PN=NP
B     17.      NV=0
B     18.      M=NOPTS(I)

```

C..

C..EXPRESS THE CORNERS OF SURFACE I IN LOCAL COORDINATES

```

B     19.      DO 77 J=1,M
B     20.      DO 70 K=1,3
B     21.      POINT(K)=PTS(I,J,K)
B     22. 70    CONTINUE
B     23.      CALL CTRNSM(1,I,POINT,TOINT)
B     24.      XX(J)=TOINT(1)
B     25.      YX(J)=TOINT(2)
B     26.      DO 16 K=1,2
B     27.      XLP(J,K)=TOINT(K)
B     28. 16    CONTINUE
B     29. 77    CONTINUE

```

C..

C..CALCULATE DELX AND DELY

```

B     30.      CALL MINMAX(XX,M,XMA,XMI)
B     31.      DELX(LC)=(XMA-XMI)/PN
B     32.      CALL MINMAX(YX,M,YMA,YMI)
B     33.      DELY(LC)=(YMA-YMI)/PN

```

C..

C..FIND THE COORDINATES (LOCAL SYSTEM) OF THE GRID ELMNTS CENTERS

```

B     34.      DO 14 NX=1,NP
B     35.      XN=NX
B     36.      XV=XMI+(XN-.5)*DELX(LC)
B     37.      DO 13 NY=1,NP
B     38.      YN=NY
B     39.      YV=YMI+(YN-.5)*DELY(LC)
B     40.      CALL INTER(XV,YV,M,XLP,IYES)
B     41.      IF(IYES.EQ.0)GO TO 13
B     42.      NV=NV+1
B     43.      VAL(LC,NV,1)=XV
B     44.      VAL(LC,NV,2)=YV
B     45. 13    CONTINUE
B     46. 14    CONTINUE

```

C..

C..IF THERE ARE NOT AT LEAST NVMIN GRID CENTERS,DIVIDE SURF FURTHER

```
B 47. IF(NV.LT.NVMIN)GO TO 90
B 48. VN=NV
B 49. NVAL(LC)=NV
B 50. RETURN
B 51. 95 PRINT 100,I,NPMAX,NV
B 52. 100 FORMAT(X,*ABORT..ABORT..SURFACE*,I3,* REQUIRES MORE THAN*,I2,/,
1X,* GRID PARTITIONS. ONLY*,I2,* GRID ELEMENTS CAN BE ASSIGNED*,/
2X,* SUGGESTION: CHANGE THE GEOMETRY OF THIS SURFACE; OR EDIT*,/,
3X,* THE PROGRAM, I.E.INCREASE THE VALUE OF NPMAX TAKING CARE*,/
4,X,* THAT ALL THE PERTINENT ARRAYS IN GF LUM AND SOL ARE*,/,
5X,* RE-DIMENSIONED APPROPRIATELY.*)
B 53. STOP
B 54. END
```

```

SUBROUTINE OBSTRT(IFR,JFR,IVFR,NW)
  COMMON/BLKA/AR(154),SUMG(154)
  COMMON/BLKU/U(77,3,3)
  COMMON/BLKN/NOVOL,NOWALS,NOPTS(154)
  COMMON/BLKP/PCN(154,3),PTS(154,10,3)
  COMMON/BLKV/VAL(2,100,2),NVAL(2),NP
  COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
  COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
  REAL L(3)
  DIMENSION POINT1(3),POINT2(3),POINT(3),TOINT(3)
  DIMENSION VN(3),RC(3),RR(3),S(3),EPS(3),CC(10,3)
  TOINT(3)=0.
  I=IFR
  J=JFR
  IV=IVFR
  IO=IVC(IV,I)
  JO=IVC(IV,J)
  N1=NVAL(1)
  N2=NVAL(2)

  C..
  C..LOOP OVER THE GRID ELEMENTS OF SURFACE I
  DO 100 IN=1,N1
    C..
    C..DEFINE THE GRID ELEMENT IN IN THE BUILDING SYSTEM.CALL IT POINT1
    DO 1 K=1,2
      TOINT(K)=VAL(1,IN,K)
    CONTINUE
    CALL CTRNSM(2,IO,POINT1,TOINT)
    C..
    C..LOOP OVE THE GRID ELEMENTS OF SURFACE J
    DO 95 JN=1,N2
      C..
      C..DEFINE THE GRID ELEMENT JN IN BUILDING SYSTEM. CALL IT POINT2
      DO 2 K=1,2
        TOINT(K)=VAL(2,JN,K)
      CALL CTRNSM(2,JO,POINT2,TOINT)
      C..
      C..FIND THE UNIT VECTOR OF THE LINE GOING FROM POINT1 TO POINT2
      DO 3 K=1,3
        L(K)=POINT2(K)-POINT1(K)
      CALL DOT(L,L,SL2)
      SL=SQRT(SL2)
      DO 4 K=1,3
        L(K)=L(K)/SL
      C..
      C..LOOP OVER ALL SURFACES OF SPACE IV TO FIND WHICH ONE,IF ANY,
      C..INTERSECTS THE LINE SEGMENT FROM POINT1 TO POINT2.
      DO 80 IL=1,NW
        IF(IL.EQ.I.OR.IL.EQ.J)GO TO 80
        CALL POINTR(1,IV,IL,I,ILI)
        CALL POINTR(1,IV,IL,J,ILJ)
        IF(EOM(ILJ).LT.1.E-5.AND.EOM(ILI).LT.1.E-8)GO TO 80
      C..
      C..DEFINE THE VECTOR NORMAL (VN) AND ORIGIN (RC) OF SURFACE IL
      ILO=IVC(IV,IL)
      ILS=(1+ILO)/2
      UNQ=1.

```

```
B      43.      IF(2*ILS.EQ.ILO)UNO=-1.
B      44.      DO 5 K=1,3
B      45.      VN(K)=UNO*U(ILS,3,K)
B      46. 5     RC(K)=PTS(ILO,1,K)
      C..
      C..FIND INTERSECTION OF PLANE IL WITH LINE POINT2-POINT1
B      47.      CALL PLVNE(RC,VN,POINT1,L,RR,INTR)
B      48.      IF(INTR.EQ.0)GO TO 80
      C..
      C..FIND IF THE INTRSECTION (R) IS WITHIN THE SEGMENT DEFINED
      C..BY POINT1 AND POINT2
B      49.      DO 6 K=1,3
B      50. 6     S(K)=RR(K)-POINT1(K)
B      51.      CALL DOT(S,L,SL)
B      52.      IF(SL.LT.1.E-8)GO TO 80
B      53.      DO 7 K=1,3
B      54. 7     S(K)=RR(K)-POINT2(K)
B      55.      CALL DOT(S,L,SL)
B      56.      IF(SL.GT.1.E-8)GO TO 90
      C..
      C..FIND IF THE INTRSECTION (R) IS WITHIN THE BOUNDS OF SURFACE IL.
      C.. TO DO SO FIRST FIND THE CORNERS OF SURFACE IL AND THE INTRSECTION
      C.. POINT R IN THE LOCAL COORDINATE SYSTEM OF SURFACE IL.
B      57.      MM=NOPTS(ILO)
B      58.      DO 10 M=1,MM
B      59.      DO 8 K=1,3
B      60. 8     POINT(K)=PTS(ILO,M,K)
B      61.      CALL CTRNSM(1,ILO,POINT,TOINT)
B      62.      DO 9 K=1,3
B      63. 9     CC(M,K)=TOINT(K)
B      64. 10    CONTINUE
B      65.      CALL CTRNSM(1,ILO,RR,EPS)
B      66.      CALL INTER(EPS(1),EPS(2),MM,CC,INT1)
B      67.      IF(INT1.EQ.0)GO TO 80
      C..
      C..FIND THE COORDINATES OF THE 4 CORNERS OF EACH GRID ELEMENT IN
      C..THE BUILDING SYSTEM AND CALCULATE THE GEOMETRIC FACTOR
B      68.      CALL LCLPNT(3,IO,IN)
B      69.      CALL LCLPNT(4,JO,JN)
B      70.      CALL GFDRVR(SUMM,IO,JO)
B      71.      CALL POINTR(1,IV,I,J,IJ)
B      72.      EOM(IJ)=EOM(IJ)-SUMM
B      73.      GO TO 95
B      74. 80    CONTINUE
B      75. 95    CONTINUE
B      76. 100   CONTINUE
B      77.      RETURN
B      78.      END
```

SUBROUTINE TRNSFM(KI)

COMMON/BLKU/U(77,3,3)

COMMON/BLKP/PCN(154,3),PTS(154,10,3)

COMMON/BLKT/P(10,3)

DIMENSION V1(3),V2(3),V3(3)

C..

C..GENERATES THE TRANSFORMATION MATRIX FROM THE BUILDING

C..COORDINATE SYSTEM TO THE LOCAL COORDINATE SYSTEM OF

C..SURFACE K. NOTE, HOWEVER, THAT SURFACE 1 OF WALL KW

C..IS ACTUALLY CALCULATED. THE TRANSFORMATION OF SURFACE 2

C..IS DERIVED FROM THAT OF SURFACE 1 AS IT IS NEEDED

C..IT ALSO DETERMINES THE ORIGIN OF SURFACE K

C..

K=KI

KW=(K+1)/2

DO 1 L=1,3

V1(L)=P(2,L)-P(1,L)

V2(L)=P(3,L)-P(2,L)

CALL CROSS(V1,V2,V3)

ABS1=ABS3=0.

DO 2 L=1,3

ABS1=ABS1+V1(L)*V1(L)

ABS3=ABS3+V3(L)*V3(L)

ABS1=SQRT(ABS1)

ABS3=SQRT(ABS3)

IF(ABS1.LT.1.E-8.OR.ABS3.LT.1.E-8) GO TO 5

IF(2*KW.EQ.K)RETURN

DO 3 L=1,3

U(KW,3,L)=V3(L)/ABS3

U(KW,1,L)=V1(L)/ABS1

V1(L)=U(KW,3,L)

V2(L)=U(KW,1,L)

CALL CROSS(V1,V2,V3)

DO 4 L=1,3

U(KW,2,L)=V3(L)

RETURN

PRINT 101

FORMAT(*1ABORT IN TRNSFM - DUPLICATE POINTS*)

STOP203

END


```
      SUBROUTINE CTRNSM(IWAY,IS,POINT,TOINT)
B      2.      COMMON/BLKP/PCN(154,3),PTS(154,10,3)
B      3.      DIMENSION POINT(3),TOINT(3)
B      4.      COMMON/BLKU/U(77,3,3)
      C..IWAY=1..GIVEN POINT IN BUILDING COORDINATE SYSTEM IT TRANSFORMS
      C..      IT TO THE LOCAL COORDINATE SYSTEM OF SURFACE I; AND IT
      C..      RETURNS IT IN ARRAY TOINT.
      C..IWAY=2..GIVEN TOINT IN THE LOCAL COORDINATE SYSTEM IT TRANSFORMS
      C..      IT TO THE BUILDING COORDINATE SYSTEM: IT RETURNS IT IN
      C..      ARRAY POINT.
      C..
      C..BOTH SURFACES OF A WALL USE THE SAME TRANSFORMATION MATRIX.GIVEN
      C..THE TRANSFORMATION MATRIX FOR SIDE 1,THE X AND Z AXIS OF SIDE 2
      C..ARE TAKEN TO BE THE NEGATIVE FOR THOSE OF SIDE 1.BOTH SIDES USE
      C..THE SAME ORIGIN POINT R.
      C..
B      5.      I=(IS+1)/2
B      6.      ISIDE=1
B      7.      IF(2*I.EQ.IS)ISIDE=2
      C..
B      8.      IF(IWAY.EQ.2)GO TO 20
      C..TRANSFORM FROM BUILDING TO SURFACE COORDINATE SYSTEM
B      9.      DO 2 K=1,3
B     10.      TOINT(K)=0.
B     11.      UNO=1.
B     12.      IF(ISIDE.EQ.1)GO TO 15
B     13.      IF(K.EQ.2)GO TO 15
B     14.      UNO=-1.
B     15. 15    CONTINUE
B     16.      DO 1 L=1,3
B     17.      TOINT(K)=TOINT(K)+UNO*U(I,K,L)*(POINT(L)-PTS(IS,1,L))
B     18. 1    CONTINUE
B     19. 2    CONTINUE
B     20.      RETURN
      C..
      C..TRANSFORM FROM SURFACE TO BUILDING COORDINATE SYSTEM
B     21. 20    CONTINUE
B     22.      DO 4 K=1,3
B     23.      POINT(K)=PTS(IS,1,K)
B     24.      DO 3 L=1,3
B     25.      UNO=1.
B     26.      IF(ISIDE.EQ.1)GO TO 16
B     27.      IF(L.EQ.2)GO TO 16
B     28.      UNO=-1.
B     29. 15    CONTINUE
B     30.      POINT(K)=POINT(K)+UNO*U(I,L,K)*TOINT(L)
B     31. 3    CONTINUE
B     32. 4    CONTINUE
B     33.      RETURN
B     34.      END
```

```

SUBROUTINE LCLPNT(IN,I,LNV)
  2.  COMMON/BLKU/U(77,3,3)
  3.  COMMON/BLKV/VAL(2,100,2),NVAL(2),NP
  4.  COMMON/BLKV2/DELX(2),DELY(2)
  5.  COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2
  6.  COMMON/BLKN/NOVOL,NOWALS,NOPTS(154)
  7.  COMMON/BLKP/PCN(154,3),PTS(154,10,3)
  8.  DIMENSION POINT(3),TOINT(3)
      C..PREPARES THE COORDINATES IN THE BUILDING SYSTEM FOR THE
      C..GEOMETRIC FACTOR CALCULATION. IF IN=1 OR =2 THE CORNERS
      C..ARE FOR THE ENTIRE BOUNDED SURFACE. IF IN=3 OR =4 THE
      C..CORNERS ARE FOR THE IVL GRID ELEMENT OF SURFACE I
  9.  IF(IN.GT.2)GO TO 3
      C..
      C..GIVEN THE CORNERS OF SURFACE I (PTS), IT STORES THE DATA
      C..IN THE LOCAL ARRAY PNTS. IT IS EXCLUSIVELY FOR GFDRVR.
 10.  NT=NOPTS(I)
 11.  IF(IN.EQ.1)GO TO 5
 12.  NP2=NT
 13.  NSTRG2=NT
 14.  GO TO 5
 15. 5  NP1=NT
 16.  NSTRG1=NT
 17. 6  CONTINUE
 18.  DO 1 NP=1,NT
 19.  DO 2 K=1,3
 20.  PNTS(IN,NP,K)=PTS(I,NP,K)
 21.  STRG(IN,NP,K)=PTS(I,NP,K)
 22. 2  CONTINUE
 23. 1  CONTINUE
 24.  UNO=1.
 25.  IW=(I+1)/2
 26.  IF(2*IW.EQ.I)UNO=-1.
 27.  DO 10 K=1,3
 28. 10  UVN(IN,K)=UNO*U(IW,3,K)
 29.  RETURN
      C..
      C..THE CORNERS OF THE GRID ELEMENT
 30. 3  CONTINUE
 31.  JN=1
 32.  IF(IN.EQ.4)JN=2
 33.  NP1=4
 34.  NP2=4
 35.  NSTRG1=4
 36.  NSTRG2=4
 37.  DO 15 NP=1,4
 38.  UNOX=-.5
 39.  UNOY=-.5
 40.  IF(NP.EQ.2.OR.NP.EQ.3)UNOX=.5
 41.  IF(NP.GT.2)UNOY=.5
 42.  TOINT(1)=VAL(JN,LNV,1)+UNOX*DELX(JN)
 43.  TOINT(2)=VAL(JN,LNV,2)+UNOY*DELY(JN)
 44.  TOINT(3)=0.
 45.  CALL CTRNSM(2,I,POINT,TOINT)
 46.  DO 14 K=1,3
 47.  PNTS(JN,NP,K)=POINT(K)
 48.  STRG(JN,NP,K)=POINT(K)

```

B	49. 14	CONTINUE
B	50. 15	CONTINUE
B	51.	RETURN
B	52.	END

SUBROUTINE FRMSTR

```
B      2.      COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2
      C..
      C..THE COORDINATES AND THE NUMBER OF CORNERS OF EACH SURFACE PAIR
      C..ARE STORED IN A DOBLE ENTRY.BEFORE THE REPLACEMENT SEQUENCE DUE
      C..TO VISUAL OCCULTATION, THE COORDINATES (PNTS),AND THE NUMBER
      C..OF CORNERS ARE RESET IN THIS SUBROUTINE TO THEIR INITIAL VALUES.
      C..
B      3.      NP1=NSTRG1
B      4.      NP2=NSTRG2
B      5.      DO 10 IC=1,2
B      6.      NT=NP1
B      7.      IF(IC.EQ.2)NT=NP2
B      8.      DO 9 N=1,NT
B      9.      DO 8 K=1,3
B     10.      PNTS(IC,N,K)=STRG(IC,N,K)
B     11. 8      CONTINUE
B     12. 9      CONTINUE
B     13. 10     CONTINUE
B     14.      RETURN
B     15.      END
```

SUBROUTINE REPLCE(KSEE)

```
3      2.      REAL L,LP
3      3.      COMMON/H/VN1(3),VN2(3),R1(3),R2(3),F(3),FP(3),POINT(3)
3      4.      COMMON/BCH2/MPS(2),L(3),LP(3),JSEE(20),X(3),XRP(20,3),IRP(20)
3      5.      COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2
3      6.      COMMON/BLKU/U(77,3,3)
      C..THIS IS A CLIPPING ROUTINE. GIVEN TWO SURFACES DEFINED BY THEIR
      C..CORNERS (PNTS),AND THEIR UNIT VECTORS (UVN) IT CLIPS THAT PORTION
      C..OF ONE SURFACE THAT CANNOT BE SEEN FROM THE OTHER.
B      7.      KSEE=0
      C..KSEE=0 MEANS THAT SURFACES ARE NOT FACING EACH OTHER
      C.
      C..
      C..RESET THE COORDINATES AND THE NUMBER OF CORNERS
B      8.      DO 10 K=1,3
B      9.      VN1(K)=UVN(1,K)
B     10.      VN2(K)=UVN(2,K)
B     11. 10    CONTINUE
      C..IF IT RETURNS AT THIS POINT THE SURFACES ARE NOT FACING
      C
B     12.      KSEE=1
      C.
      C..CALCULATE THE PARAMETERS OF THE LINE OF INTERSECTION OF PLANES
B     13.      DO 15 K=1,3
B     14.      R1(K)=PNTS(1,1,K)
B     15.      R2(K)=PNTS(2,1,K)
B     16. 15    CONTINUE
B     17.      CALL TWOPLN(VN1,VN2,R1,R2,L,F)
      C
      C..CONSIDER ONE SURFACE AT THE TIME AS THE OBSERVING PLANE
B     18.      DO 50 IS=1,2
B     19.      CALL VEAMOS(IS,N,NP)
B     20.      KSEE=0
B     21.      IF(N.EQ.0)RETURN
      C..N=0..MEANS ALL THE POINTS ARE OUT OF VIEW. GEOM FAC=0
      C.
B     22.      KSEE=1
B     23.      IF(N.EQ.NP)GO TO 50
      C..IF N=NP ALL THE POINTS CAN BE SEEN.NONE NEED TO BE REPLACED.
      C
      C..OTHERWISE REPLACE THOSE WHOSE VALUE OF JSEE=0
      C..CONSIDER ONE EDGE AT THE TIME
B     24.      CALL CAMBIA(IS,NP)
B     25.      NMAX=MPS(IS)
B     26.      DO 45 N=1,NMAX
B     27.      DO 46 K=1,3
B     28.      PNTS(IS,N,K)=XRP(N,K)
B     29. 46    CONTINUE
B     30. 45    CONTINUE
B     31. 50    CONTINUE
B     32.      RETURN
B     33.      END
```

SUBROUTINE VEAPOS(ISL,N,NPL)

```
2.  DIMENSION VN(3),POINT1(3)
3.  REAL L,LP
4.  COMMON/H/VN1(3),VN2(3),R1(3),R2(3),F(3),FP(3),POINT(3)
5.  COMMON/BCH2/MPS(2),L(3),LP(3),JSEE(20),X(3),XRP(20,3),IRP(20)
6.  COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2
7.  COMMON/BLKU/U(77,3,3)
```

C..

C..CALLED FROM REPLCE. IT COUNTS HOW MANY OF THE CORNERS OF

C..SURFACE IS (ISL) CAN BE SEEN FROM THE OTHER ONE.

C..

```
8.  IS=ISL
9.  JS=1
10. IF(IS.EQ.1)JS=2
11. DO 13 K=1,3
12. 13 POINT1(K)=PNTS(JS,1,K)
13.  NP=NP1
14.  DO 14 K=1,3
15.  14 VN(K)=VN2(K)
16. IF(IS.EQ.1)GO TO 16
17.  NP=NP2
18.  DO 15 K=1,3
19.  15 VN(K)=VN1(K)
20. 16 CONTINUE
21.  N=0
22.  DO 18 N1=1,NP
23.  DO 17 K=1,3
24.  XRP(N1,K)=0.
25.  17 POINT(K)=PNTS(IS,N1,K)
26.  CALL SIVEO(POINT,POINT1,VN,ISEE)
27.  JSEE(N1)=ISEE
28.  IRP(N1)=0
29.  N=N+ISEE
30. 18 CONTINUE
31.  NPL=NP
32.  RETURN
33.  END
```

SUBROUTINE CAMBIA(ISL,NP)

2. REAL L,LP

3. COMMON/H/VN1(3),VN2(3),R1(3),R2(3),F(3),FP(3),POINT(3)

4. COMMON/BCH2/MPS(2),L(3),LP(3),JSEE(20),X(3),XRP(20,3),IRP(20)

5. COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2

6. COMMON/BLKU/U(77,3,3)

C..

C..CALLED FROM REPLCE.IT REPLACES THE CORNERS THAT CAN NOT
C..BE SEEN WITH THE POINTS OF INTERSECTION. IT RESETS ALSO
C..THE NUMBER OF CORNERS.

C..

7. IS=ISL

8. M=0

9. DO 42 N1=1,NP

10. N2=N1+1

11. IF(N1.EQ.NP)N2=1

12. J1=JSEE(N1)

13. J2=JSEE(N2)

14. IF(J1.EQ.J2)GO TO 33

15. N=N1

16. IF(J2.EQ.1)N=N2

17. IF(N.EQ.N2)GO TO 34

18. IF(IRP(N).EQ.1)GO TO 34

19. IRP(N)=1

20. M=M+1

21. DO 33 K=1,3

22. 33 XRP(M,K)=PNTS(IS,N,K)

23. 34 CONTINUE

24. DO 36 K=1,3

25. LP(K)=PNTS(IS,N2,K)-PNTS(IS,N1,K)

26. R1(K)=PNTS(IS,N1,K)

27. 36 CONTINUE

28. CALL CROSS(R1,LP,FP)

29. CALL TWOLIN(L,LP,F,FP,X,INTR)

30. IF(INTR.EQ.0)GO TO 42

31. M=M+1

32. DO 37 K=1,3

33. XRP(M,K)=X(K)

34. 37 CONTINUE

35. IF(N.EQ.N1)GO TO 42

36. IF(IRP(N).EQ.1) GO TO 42

37. IRP(N)=1

38. M=M+1

39. DO 53 K=1,3

40. 53 XRP(M,K)=PNTS(IS,N,K)

41. GO TO 42

42. 38 CONTINUE

43. IF(J1.EQ.0)GO TO 42

44. IF(IRP(N1).EQ.1)GO TO 51

45. IRP(N1)=1

46. M=M+1

47. DO 39 K=1,3

48. 39 XRP(M,K)=PNTS(IS,N1,K)

49. 51 IF(IRP(N2).EQ.1)GO TO 42

50. IRP(N2)=1

51. M=M+1

52. DO 40 K=1,3

MNF I=DRBGF5 J E Z

V-22 UT2D 6600 15

```
B 53. 40 XRP(M,K)=PNTS(IS,N2,K)
B 54. 42 CONTINUE
B 55.     MPS(IS)=M
B 56.     IF(IS.EQ.1)NP1=M
B 57.     IF(IS.EQ.2)NP2=M
B 58.     RETURN
B 59.     END
```



```
SUBROUTINE SIVEO(POINT,POINT1,VN,ISEE)
```

```
2. DIMENSION POINT(3),POINT1(3),VN(3)
```

```
C
```

```
C..GIVEN A POINT IN THE BUILDING COORDINATE SYSTEM, THIS
```

```
C..SUBROUTINE DETERMINES IF SUCH POINT IS ON THE POSITIVE (ISEE=1)
```

```
C..OR ON THE NEGATIVE (ISEE=0) SIDE OF A SURFACE THAT GOES
```

```
C..THROUGH POINT1 AND WHOSE UNIT VECTOR NORMAL IS VN.
```

```
C
```

```
3. ISEE=1
```

```
4. RH=0.
```

```
5. DO 10 K=1,3
```

```
6. RH=RH+VN(K)*(POINT(K)-POINT1(K))
```

```
7. 10 CONTINUE
```

```
8. IF(RH.LT.1.E-8)ISEE=0
```

```
9. RETURN
```

```
10. END
```

SUBROUTINE GFDRVR(SUM,I1,I2)

2. COMMON/B/UVN(2,3),STRG(2,20,3),NSTRG1,NSTRG2,PNTS(2,20,3),NP1,NP2

3. COMMON/G1/C1(3),C2(3),CP1(3),CP2(3),TERM,D(3),DP(3),DM,DPM,

1 DDP,DEL(3),DPERP(3),DPER,OPAR,X(4)

4. PI=3.141592654

5. SUM=0.

C

C.....EXECUTE THE SUM OVER THE CORNERS OF THE UNPRIME SURFACE

C.....NP1=NUMBER OF CORNERS OF UNPRIME SURFACE

6. DO 15 N1=1,NP1

7. N2=N1+1

8. IF(N1.EQ.NP1)N2=1

9. DO 1 K=1,3

10. C1(K)=PNTS(1,N1,K)

11. C2(K)=PNTS(1,N2,K)

12. 1 CONTINUE

C

C.....EXECUTE THE INNER SUM. I.E. THE SUM OVER THE PRIME SURFACE

C.....NP2=NUMBER OF CORNERS OF PRIME SURFACE

13. DO 14 N1P=1,NP2

14. N2P=N1P+1

15. IF(N1P.EQ.NP2)N2P=1

16. DO 2 K=1,3

17. CP1(K)=PNTS(2,N1P,K)

18. CP2(K)=PNTS(2,N2P,K)

19. 2 CONTINUE

C

C ----- CALL THE SEGMENT INTEGRATOR

20. CALL SGPAIR

21. SUM=SUM+TERM

22. 14 CONTINUE

23. 15 CONTINUE

C

C.....DOUBLE SUM IS COMPLETE. NOW NORMALIZE THE SUM.

24. SUM=SUM/(4.*PI)

25. RETURN

26. END

```
      SUBROUTINE SGPAIR
B      2.      COMMON/G1/C1(3),C2(3),CP1(3),CP2(3),TERM,D(3),DP(3),DM,DPM,
      1      DDP,DEL(3),DPERP(3),DPER,OPAR,X(4)
B      3.      TERM=0.
B      4.
      C RELATIVE POSITION OF POINT2 FROM POINT1 FOR EACH SEGMENT (D & DP)
      C AND OF UPRIMED POINT1 FROM PRIMED POINT1
B      5.      DO 1 K=1,3
B      6.      D(K)=C2(K)-C1(K)
B      7.      DP(K)=CP2(K)-CP1(K)
B      8.      DEL(K)=C1(K)-CP1(K)
B      9. 1     CONTINUE
B     10.
      C MAGNITUDE OF EACH SEGMENT
B     11.      CALL DOT(D,D,DM)
B     12.      DM=SQRT(DM)
B     13.      CALL DOT(DP,DP,DPM)
B     14.      DPM=SQRT(DPM)
B     15.
      C UNIT VECTOR OF EACH SEGMENT
B     16.      DO 2 K=1,3
B     17.      D(K)=D(K)/DM
B     18.      DP(K)=DP(K)/DPM
B     19. 2     CONTINUE
B     20.
      C DOT PRODUCT OF THE PRIME AND UPRIME UNIT VECTORS
B     21.      CALL DOT(D,DP,DDP)
B     22.
      C IF THE SEGMENTS ARE PERPENDICULAR TO EACH OTHER
B     23.      IF(ABS(DDP).LT.1.E-5)RETURN
B     24.      IF((1.-ABS(DDP)).LT.1.E-5) GO TO 5
B     25.
      C IF THE SEGMENTS ARE NEITHER PERPENDICULAR NOR PARALLEL TO
      C EACH OTHER
B     26.      CALL ANYDOT
B     26.      RETURN
B     27.
      C IF THE SEGMENTS ARE PARALLEL TO EACH OTHER
B     28. 5     CALL PRALEL
B     29.      RETURN
B     30.      END
```

SUBROUTINE PRAEL

2. COMMON/61/C1(3),C2(3),CP1(3),CP2(3),TERM,D(3),DP(3),DM,DPM,
1 DDP,DEL(3),DPERP(3),DPER,DPAR,X(4)

C CALCULATE THE CONTRIBUTION TO THE GEOMETRIC FACTOR WHEN THE
C SEGMENTS ARE PARALLEL

3. UNO=1.

4. CALL DOT(DEL,D,DPAR)

5. DO 2 K=1,3

6. DPERP(K)=DEL(K)-DPAR*D(K)

7. 2 CONTINUE

8. CALL DOT(DPERP,DPERP,DPER2)

9. DPER=SQRT(DPER2)

10. X(1)=DPAR-DDP*DPM

11. X(2)=X(1)+DM

12. X(3)=DPAR+DM

13. X(4)=DPAR

14. TERM=0.

15. UNO=-1.

16. DO 5 L=1,4

17. UNO=-UNO

18. XX=X(L)

19. X2=XX*XX

20. G=0.

21. IF(X2.LT.1.E-10.AND.DPER2.LT.1.E-10)GO TO 4

22. G=.5*(X2+DPER2)*(ALOG(X2+DPER2)-1.)-X2

23. GXTRA=0.

24. IF(DPER2.LT.1.E-10)GO TO 50

25. GXTRA=2.*DPER2*((XX/DPER)*ATAN(XX/DPER)-.5*ALOG(1.+X2/DPER2))

26. 50 CONTINUE

27. G=G+GXTRA

28. 4 CONTINUE

29. TERM=TERM+UNO*G

30. 5 CONTINUE

31. RETURN

32. END

SUBROUTINE ANYDCT

```
B      2.      DIMENSION V(3)
B      3.      COMMON/G1/C1(3),C2(3),CP1(3),CP2(3),TERM,D(3),DP(3),DM,DPM,
      1      DDP,DEL(3),DPERP(3),DPER,DPAR,X(4)
      C CARRIES OUT NUMERICAL INTEGRATION FOR SEGMENTS WHICH ARE NEITHER
      C PERPENDICULAR NOR PARALLEL TO EACH OTHER
B      4.
      C THE NUMBER OF INTEGRATION STEPS FOR UMPRIME SEGMENT
B      5.      NSTEP=100
B      6.      STEP=NSTEP
B      7.      DSTEP=DM/STEP
B      8.      NPLUS=NSTEP+1
B      9.      UNQ=-1.
B     10.      TERM=-2.*DPM*DM
B     11.      DO 20 IP=1,2
B     12.      PP=DPM
B     13.      IF(IP.EQ.2)PP=0.
B     14.      PP2=PP*PP
B     15.      UNQ=-UNQ
B     16.      G=0.
B     17.      DO 10 L=1,NPLUS
B     18.      LM=L-1
B     19.      FACTOR=1.
B     20.      IF(LM.EQ.0.OR.LM.EQ.NSTEP)FACTOR=.5
B     21.      XLM=LM
B     22.      P=XLM*DSTEP
B     23.      DO 2 K=1,3
B     24.      V(K)=DEL(K)+P*D(K)
B     25. 2      CONTINUE
B     26.      CALL DOT(V,V,V2)
B     27.      CALL DOT(V,DP,VPAR)
B     28.      VPER2=V2-VPAR*VPAR
B     29.      PPD=PP-VPAR
B     30.      IF(ABS(PPD).LT.1.E-8)GO TO 10
B     31.      G=G+PPD*ALOG(VPER2+PPD*PPD)*FACTOR
B     32.      IF(VPER2.LT.1.E-8)GO TO 10
B     33.      VPER=SQRT(VPER2)
B     34.      G=G+2.*VPER*ATAN(PPD/VPER)*FACTOR
B     35. 10      CONTINUE
B     36.      TERM=TERM+UNQ*G*DSTEP
B     37. 20      CONTINUE
B     38.      TERM=TERM*DDP
B     39.      RETURN
B     40.      END
```

SUBROUTINE CYCLY(M)

2. COMMON/BLKT/P(10,3)

3. DIMENSION PP(10,3)

C..

C..GIVEN A SET OF POINTS (P) IT CYCLICLY PERMUTES THEM.

C..

4. DO 3 K=1,3

5. DO 1 J=1,M

6. 1 PP(J,K)=P(J,K)

7. N=M-1

8. DO 2 J=1,N

9. 2 P(J,K)=PP(J+1,K)

10. 3 P(M,K)=PP(1,K)

11. RETURN

12. END

SUBROUTINE AREAS(CNTR,C,M,ARA)

2. DIMENSION CNTR(3), C(10,3)

3. CNTR(3)=0.

C..

C..GIVEN THE COORDINATES OF THE CORNERS OF A POLYGON,IT CALCULATES

C..THE AREA AND GEOMETRICAL CENTER OF THE POLYGON.

C.. -

4. L=M-1

5. AREA= ((C(1,2)-C(M,2))*C(M,1)-(C(1,1)-C(M,1))*C(M,2))/2.

6. CNTR(1)=AREA*(C(M,1)+C(1,1))/3.

7. CNTR(2)=AREA*(C(M,2)+C(1,2))/3.

8. DO 1 J=1,L

9. N=J+1

10. AR= ((C(N,2)-C(J,2))*C(J,1)-(C(N,1)-C(J,1))*C(J,2))/2.

11. AREA=AREA+AR

12. CNTR(1)=CNTR(1)+AR*(C(N,1)+C(J,1))/3.

13. CNTR(2)=CNTR(2)+AR*(C(N,2)+C(J,2))/3.

14. 1 CONTINUE

15. CNTR(1)=CNTR(1)/AREA

16. CNTR(2)=CNTR(2)/AREA

17. ARA=AREA

18. RETURN

19. END

SUBROUTINE VOLUME

```
2. COMMON/BLKP/PCN(154,3),PTS(154,10,3)
3. COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
4. COMMON/BLKU/U(77,3,3)
5. COMMON/BLKA/AR(154),SUMG(154)
6. COMMON/BLK1/IVOL(154)
7. COMMON/BLKN/NOVOL, NOWALS, NOPTS(154)
8. COMMON/BLKVOL/CNTV(9,3),VOL(9)
```

C..

C..GIVEN THE AREA AND THE ORIGINS OF THE POLYGONS ENCLOSING A

C..POLYHEDRON IT CALCULATES THE VOLUME OF THE POLYHEDRON. ALSO

C..GIVEN THE GEOMETRICAL CENTER OF EACH POLYGON IT CALCULATES

C..THE GEOMETRICAL CENTER OF THE POLYHEDRON

C..

```
9. NCUM=0
10. NCU(1)=0
11. DO 12 IV=1,NOVOL
12. CNTV(IV,1)=CNTV(IV,2)=CNTV(IV,3)=0.
13. NCUM=NCUM+IWC(IV)*IWC(IV)
14. IF(IV.LT.NOVOL)NCU(IV+1)=NCUM
15. VOL(IV)=0.
16. DO 10 I=1,NOWALS
17. IF(IVOL(I).NE.IV) GOTO 10
18. NPT=NOPTS(I)
19. DO 5 K=1,3
20. IO=(I+1)/2
21. UNO=1.
22. IF(2*IO.EQ.I)UNO=-1.
23. FACTOR=UNO*U(IO,3,K)*PTS(I,1,K)*AR(I)
24. DO 88 LV=1,3
25. CNTV(IV,LV)=CNTV(IV,LV)+FACTOR*PCN(I,LV)
26. 88 CONTINUE
27. VOL(IV)=VOL(IV)-FACTOR/3.
28. 5 CONTINUE
29. 10 CONTINUE
30. RECIP=-1./(4.*VOL(IV))
31. DO 89 LV=1,3
32. 89 CNTV(IV,LV)=RECIP*CNTV(IV,LV)
33. PRINT 9999,IV,VOL(IV),(CNTV(IV,LV),LV=1,3)
34. 9999 FORMAT(X,* VOL(*,I2,*)=*,F9.1,* CENTER=*,3F8.2)
35. 12 CONTINUE
36. RETURN
37. END
```


SUBROUTINE PLNLNE(RC,N,R1,L,R,INTR)

2. REAL RC(3),N(3),R1(3),L(3),R(3),X(3)

C

C..GIVEN A PLANE WHOSE UNIT VECTOR NORMAL IS N AND WHOSE KNOWN

C..POINT IS RC; AND GIVEN A LINE WHOSE DIRECTION VECTOR IS L

C..AND WHOSE KNOWN POINT IS R1, THIS SUBR. CALCULATES THE POINT

C..OF INTERSECTION OF THE PLANE WITH THE LINE (R), PROVIDED THE

C..INTERSECTION EXISTS (INTR=1)

C

3. INTR=0

4. CALL DOT(L,N,DLN)

5. IF(ABS(DLN).LT.1.E-8)RETURN

C..IF DLN=0,THE LINE IS PERPENDICULAR TO THE VECTOR NORMAL OF THE

C..SURFACE,CONSEQUENTLY THERE IS NO INTERSECTION

C

6. INTR=1

7. DO 10 I=1,3

8. X(I)=RC(I)-R1(I)

9. 10 CONTINUE

10. CALL DOT(X,N,XDN)

11. DO 20 I=1,3

12. R(I)=R1(I)+L(I)*XDN/DLN

13. 20 CONTINUE

14. RETURN

15. END

SUBROUTINE TWOLIN(L,LP,F,FP,R,INTR)

2. DIMENSION S(3),G(3),GP(3)

3. REAL L(3),LP(3),F(3),FP(3),R(3),FPF(3)

C

C..GIVEN TWO LINES IN SPACE THIS SUBROUTINE CALCULATES THE

C..COORDINATES OF THEIR POINT OF INTERSECTION.

C..LINE 1 IS DEFINED BY ITS ORIENTATION (L) AND POSITION (F)

C..VECTORS((R)X(L)=(F)); AND LINE 2 BY LP AND FP [(R)X(LP)=(FP)].

C..THEIR POINT OF INTERSECTION BY VECTOR R.

C..INTR=0 THERE IS NO INTERSECTION

C.. =1 THERE IS AN INTERSECTION

C.

4. INTR=0

5. CALL CROSS(L,LP,S)

6. CALL DOT(S,S,S2)

7. IF(S2.LT.1.E-10)RETURN

C..S2=0. MEANS SEGMENTS ARE PARALLEL. THUS THEY DONT INTERSECT

C.

8. CALL DOT(FP,L,FPL)

9. CALL DOT(F,LP,FLP)

10. IF(ABS(FPL+FLP).GT.1.E-10)RETURN

C..FPL+FLP NOT=0 MEANS SEGMENTS NOT COPLANAR. THUS THEY DONT INTERSECT

C.

11. INTR=1

12. CALL CROSS(FP,F,FPF)

13. CALL DOT(FPF,FPF,VFPF)

14. IF(ABS(FPL).LT.1.E-8.AND.VFPF.LT.1.E-8)GO TO 40

C..

C..IF BOTH THE NUMERATOR (FPF), AND DENOMINATOR (FPL) ARE ZERO

C..THE DIRECT CALCULATION YIELDS AN UNDEFINED RESULT. IN SUCH

C..CASE GO TO 40 TO CARRY OUT THE CALCULATION IN OTHER C.S.

C..

15. DO 10 I=1,3

16. R(I)=FPF(I)/FPL

17. 10 CONTINUE

18. RETURN

C.

C..IF BOTH FPF AND FPL ARE ZERO ZERO DO THE CALCULATION IN A NEW

C..COORDINATE SYSTEM TRANSLATED ONE ALONG THE DIRECTION (L)X(LP).

19. 40 CONTINUE

20. CALL CROSS(S,L,G)

21. CALL CROSS(S,LP,GP)

22. DO 20 I=1,3

23. G(I)=G(I)+F(I)

24. GP(I)=GP(I)+FP(I)

25. 20 CONTINUE

26. CALL CROSS(GP,G,FPF)

27. CALL DOT(GP,L,FPL)

28. DO 30 I=1,3

29. R(I)=FPF(I)/FPL-S(I)

30. 30 CONTINUE

31. RETURN

32. END

SUBROUTINE TWOPLN(N,NP,R,RP,L,F,INTR)

2. REAL N(3),NP(3),L(3),F(3),R(3),RP(3),LDL

C

C..GIVEN TWO PLANES IN SPACE THIS SUBROUTINE CALCULATES THE

C..PARAMETERS THAT DEFINE THE LINE GENERATED BY THEIR INTERSECTION

C..GIVEN THE VECTOR NORMAL FOR EACH SURFACE (N, AND NP); AND A

C..POINT KNOWN TO BE ON EACH SURFACE (R, AND RP), WE CALCULATE

C..THE DIRECTION VECTOR OF THE LINE (L), AND ITS POSITION VECTOR (F)

C..IF P A POINT ON THE LINE, THE EQUATION FOR THE LINE IS GIVEN BY

C.. $(P) \times (L) = (F)$

C..INTR=0 MEANS THERE IS NO INTERSECTION

C..INTR=1 MEANS THERE IS AN INTERSECTION

C

3. INTR=0

4. CALL CROSS(N,NP,L)

5. CALL DOT(L,L,LDL)

6. IF(LDL.LT.1.E-8)RETURN

C..LDL=0 MEANS THE TWO PLANES ARE PARALLEL;THUS NO INTERSECTION

C

7. INTR=1

8. LDL=SQRT(LDL)

9. CALL DOT(NP,RP,SP)

10. CALL DOT(N,R,S)

11. DO 10 I=1,3

12. $F(I)=(SP*N(I)-S*NP(I))/LDL$ 13. $L(I)=L(I)/LDL$

14. 10 CONTINUE

15. RETURN

C..NOTE THAT THE ORIENTATION VECTOR (L) HAS BEEN NORMALIZED INTO

C..A UNIT VECTOR

16. END

SUBROUTINE CROSS(V1,V2,V3)

2. DIMENSION EPS(3,3,3),V1(3),V2(3),V3(3)

C THIS SUBROUTINE TAKES THE CROSS PRODUCT OF VECTORS

C V1 AND V2.....I.E. V3=V1XV2

3. DO 1 I=1,3

4. DO 1 J=1,3

5. DO 1 K=1,3

6. 1 EPS(I,J,K)=0

7. EPS(1,2,3)=EPS(2,3,1)=EPS(3,1,2)=1

8. EPS(2,1,3)=EPS(1,3,2)=EPS(3,2,1)=-1

9. DO 2 I=1,3

10. V3(I)=0

11. DO 2 J=1,3

12. DO 2 K=1,3

13. 2 V3(I)=V3(I)+EPS(I,J,K)*V1(J)*V2(K)

14. RETURN

15. END

SUBROUTINE DOT(V1,V2,RESULT)

2. DIMENSION V1(3),V2(3)

3. RESULT=0.

4. DO 1 K=1,3

5. RESULT=RESULT+V1(K)*V2(K)

6. 1 CONTINUE

7. RETURN

8. END

SUBROUTINE MINMAX(A,JMAX,AMAX,AMIN)

2. DIMENSION A(154)

3. AMIN=AMAX=A(1)

4. DO 1 I=2,JMAX

5. IF(AMAX.LT.A(I)) AMAX=A(I)

6. IF(AMIN.GT.A(I)) AMIN=A(I)

7. 1 CONTINUE

8. RETURN

9. END.

SUBROUTINE INTER(X0,Y0,JMAX,CC,INT1)

2. DIMENSION CC(10,2)

C INT1=1 X0,Y0,IS LOCATED IN THE WALL

C INT1 =0 X0,Y0,IS NOT LOCATED IN THE WALL

3. IS=0

4. DO 6 J=1,JMAX

5. IF(J.NE.JMAX) GO TO 2

6. X2=CC(1,1)

7. X1=CC(JMAX,1)

8. Y2=CC(1,2)

9. Y1=CC(JMAX,2)

10. GO TO 3

11. 2 X1=CC(J,1)

12. X2=CC(J+1,1)

13. Y2=CC(J+1,2)

14. Y1=CC(J,2)

15. 3 CALL INTE1(X0,Y0,X1,Y1,X2,Y2,I)

16. IF(I-1)6,5,7

17. 5 IS=IS+1

18. 6 CONTINUE

19. ITE=IS/2

20. ITE2=ITE*2

21. INT1=0

22. IF(IS.NE.ITE2) INT1=1

23. GO TO 8

24. 7 INT1=1

25. 8 CONTINUE

26. RETURN

27. END

SUBROUTINE INTE1 (XX0,YY0,XX1,YY1,XX2,YY2,I)

B	2.	X0=XX0
B	3.	X1=XX1
B	4.	X2=XX2
B	5.	Y0=YY0
B	6.	Y1=YY1
B	7.	Y2=YY2
	C	I=0 NO INTERSECTION
	C	I=1 YES INTERSECTION
	C	I=2 POINT IN WALL EDGE
B	8.	IF(Y0.GE.Y1.AND.Y0.LE.Y2) GO TO 1
B	9.	IF(Y0.LE.Y1.AND.Y0.GE.Y2) GO TO 1
B	10.	GO TO 20
B	11.	1 IF(Y1.NE.Y2) GO TO 2
B	12.	IF(Y0.NE.Y1) GO TO 20
B	13.	IF(X0.GE.X1.AND.X0.LE.X2) GO TO 22
B	14.	IF(X0.LE.X1.AND.X0.GE.X2) GO TO 22
B	15.	GO TO 20
B	16.	2 IF(X1.NE.X2) GO TO 3
B	17.	IF(ABS(X0-X1).LT.1.E-05) GO TO 22
B	18.	IF(X0.LT.X1) GO TO 21
B	19.	GO TO 20
B	20.	3 X=X1+(Y0-Y1)*(X2-X1)/(Y2-Y1)
B	21.	IF(ABS(X0-X).LT.1.E-05) GO TO 22
B	22.	IF(X0.LT.X) GO TO 21
B	23.	20 I=0
B	24.	RETURN
B	25.	21 I=1
B	26.	RETURN
B	27.	22 I=2
B	28.	RETURN
B	29.	END


```
      SUBROUTINE CHNGVF(IOPT,NWL,IVL)
B      2.      COMMON/BLKA/AR(154),SUMG(154)
B      3.      COMMON/BLKG/EOM(200),GEOM(1000),GSKY(154)
B      4.      COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B      5.      COMMON/SOMF/SOM(400),VIEWF(400)
B      6.      IV=IVL
B      7.      NW=NWL
B      8.      IF(IOPT.EQ.2)GO TO 25
B      9.      DO 20 IC=1,NW
B     10.      I=IVC(IV,IC)
B     11.      DO 19 JC=1,NW
B     12.      CALL PONTR1(1,IV,IC,JC,IJ)
B     13.      SOM(IJ)=0.
B     14.      IF(JC.EQ.IC)GO TO 19
B     15.      CALL POINTR(1,IV,IC,JC,KC)
B     16.      J=IVC(IV,JC)
B     17.      SOM(IJ)=EOM(KC)/AR(J)
B     18. 19      CONTINUE
B     19. 20      CONTINUE
B     20.      RETURN
B     21. 25      CONTINUE
B     22.      NWL=NW-1
B     23.      DO 30 IC=1,NWL
B     24.      JC1=IC+1
B     25.      DO 29 JC=JC1,NW
B     26.      CALL PONTR1(1,IV,IC,JC,IJ)
B     27.      CALL POINTR(2,IV,IC,JC,KC)
B     28.      GEOM(KC)=SOM(IJ)*AR(IVC(IV,JC))
B     29. 29      CONTINUE
B     30. 30      CONTINUE
B     31.      RETURN
B     32.      END
```

SUBROUTINE POINTR(L,IV,IC,JC,KC)

```
B 2. COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)
B 3. I=IC
B 4. J=JC
B 5. IF(I.LT.J)GO TO 1
B 6. I=JC
B 7. J=IC
B 8. 1 CONTINUE
B 9. KC=IWC(IV)*(I-1)+(J-I)-(I-1)*I/2
B 10. IF(L.EQ.1.OR.IV.EQ.1)RETURN
B 11. IVL=IV-1
B 12. DO 2 JV=1,IVL
B 13. KC=KC+IWC(JV)*(IWC(JV)-1)/2
B 14. 2 CONTINUE
B 15. RETURN
B 16. END
```

SUBROUTINE PONTR1(L,IV,IC,JC,KC)

2. COMMON/BLKS/NCU(20),IVC(20,24),IVCR(154),IWC(20)

3. KC=IWC(IV)*(IC-1)

4. IF(L.EQ.1)GO TO 2

5. KC=KC+NCU(IV)

6. 2 KC=KC+JC

7. RETURN

8. END

SUBROUTINE DAVID

2. PRINT 190

3. 190 FORMAT(1H1,////////15X,*P R O G R A M D R B G F*,///10X,
1 10H*****,* THE DEROB SYSTEM *,10H*****,//)

4. PRINT 191

5. 191 FORMAT(7X,*COMPUTER APPLICATIONS IN ARCHITECTURE BY THE*,
2 //10X,*NUMERICAL SIMULATION LABORATORY OF THE*,//8X,
3 *UNIVERSITY OF TEXAS SCHOOL OF ARCHITECTURE*//,
4 7X,*DIRECTOR OF RESEARCH*,5X,*FRANCISCO ARUMI-NOE*)

6. CALL JOBINFO(1,J1)

7. CALL JOBINFO(2,J2)

8. READ (1,170)NAMEN

9. PRINT 99,NAMEN

10. PRINT 100,J2,J1

11. 99 FORMAT(///20X,*RUN TITLE:*A10)

12. 100 FORMAT(///11X,*PROGRAM RUN AT*,A9,* ON*,A10,////////45X,*DON*)

13. 170 FORMAT(A10)

14. RETURN

15. END