

SAND80-8027

Unlimited Release

# Instruction Manual for the Program "Shapefactor"

**MASTER**

A. F. Emery

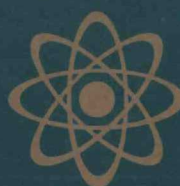
Prepared by Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550 for the United States Department of Energy under Contract DE-AC04-76DP00789.

Printed October 1980

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED



**Sandia Laboratories**  
**energy report**



Issued by Sandia Laboratories, operated for the United States Department of Energy by Sandia Corporation.

---

---

#### NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States 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.

## **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.**

MASTER

SAND80-8027  
Unlimited Release  
Printed October 1980

INSTRUCTION MANUAL FOR THE PROGRAM "SHAPEFACTOR"

A. F. Emery  
Thermal Sciences Division  
Sandia National Laboratories, Livermore\*

ABSTRACT

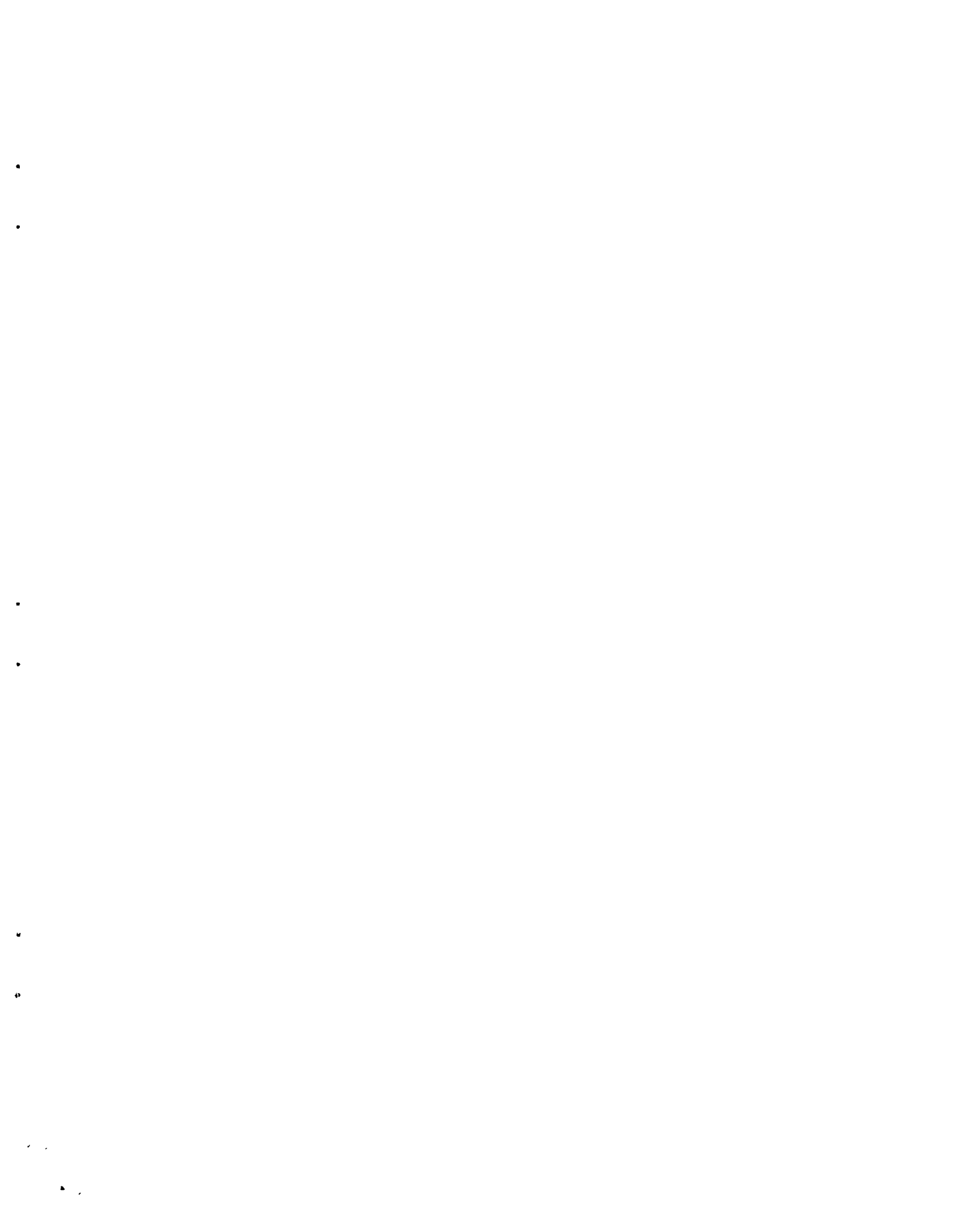
A numerical technique is presented for evaluating the geometrical radiant exchange factors (also called shape or view factors) between surfaces with interposed obstructions. Since the program is developed for plane surfaces, arbitrary surfaces are expressed by the juxtaposition of plane surfaces; to simplify the input and output, the program respectively combines and decomposes these elemental surfaces. The data input format, although designed for manual input, is ideally suited for standard three-dimensional mesh-generated programs. When obstructions are not present, the calculated exchange factors are accurate to within tenths of a percent; but when obstructions are present, the accuracy depends on the nature of the problem, the refinement of the elemental area mesh, and the amount of computation called for by the user.

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.

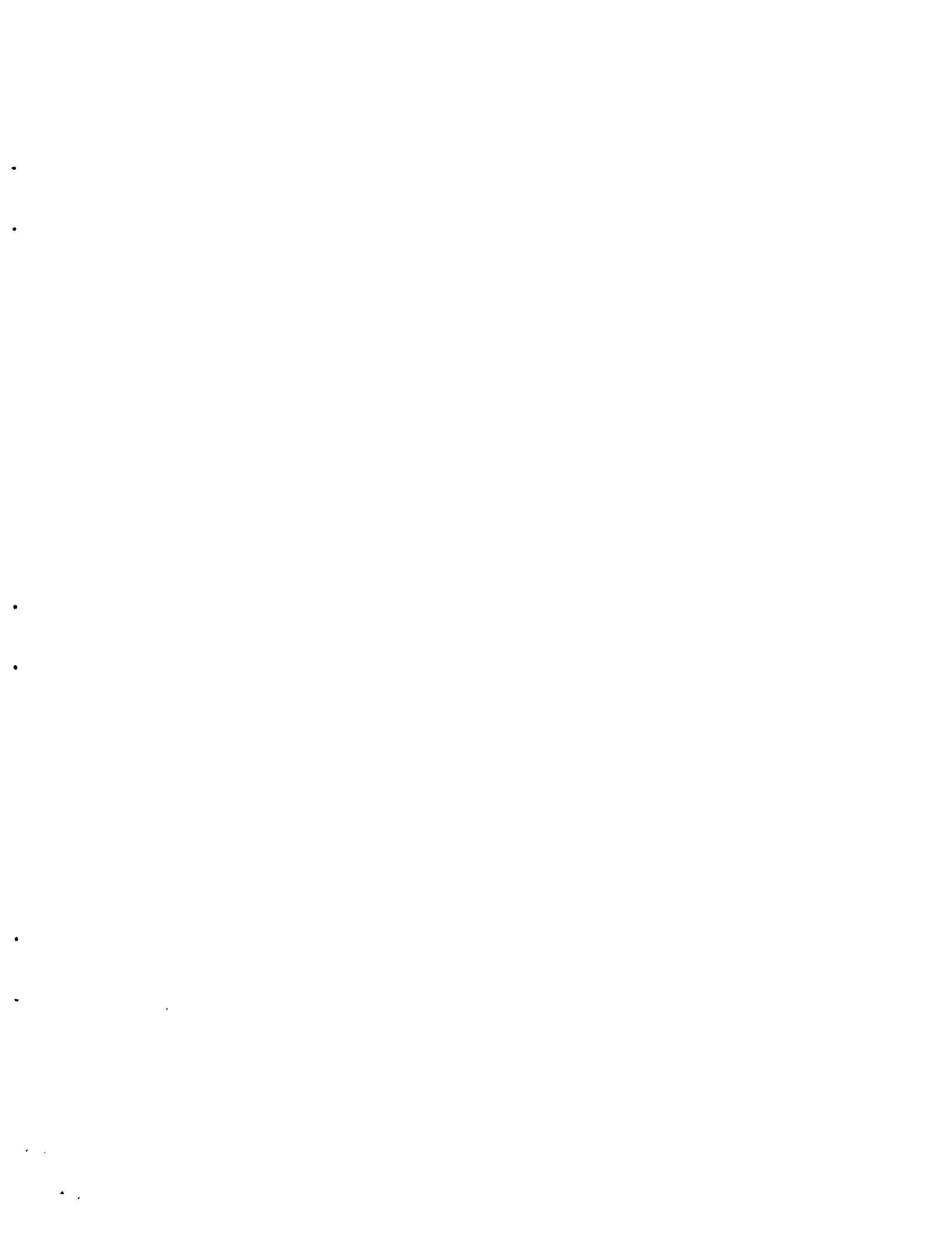
\*Permanent address: Department of Mechanical Engineering, University of Washington, Seattle, Washington 98197

UNLIMITED  
*See*



## CONTENTS

	<u>Page</u>
Introduction	7
General Comments	8
Definitions	9
Program Flow	10
Description of Routines	12
Routines SHAPFA and SHAPE	12
Routine OBSTRUC	12
Routine ENORMAL	13
Routine WPOINTS	13
Routine INTRCPT	13
Subroutine INCLUDE	14
Routine COMBINE	14
Routine MASK	14
Data	15
Data Input Format	15
Data Input Required	16
APPENDIX A--COMMENTS AND CAVEATS	19
APPENDIX B--UWENSOL MANUAL--EXCERPT	21
REFERENCES	37





## INSTRUCTION MANUAL FOR THE PROGRAM "SHAPEFACTOR"

### Introduction

In the absence of an absorbing medium, radiative heat exchange between surfaces is a function of the surface conditions and the optical view that each surface has of the others. Often the major difficulty in calculating the heat transfer rests with the accurate determination of the surface conditions, and the geometrical view between the surfaces is regarded as a lesser problem. For multi-surface enclosure problems, or for radiation between surfaces whose mutual views are obstructed, the evaluation of the view factor (also known as the shape factor or exchange factor) is a major effort. Usually graphical methods, approximate numerical techniques, or substantial simplification of the structures' geometry are utilized in estimating the shape factors.

For many situations, as in solar receivers or furnaces, such approximations are not adequate. For these, in which radiation is the dominant heat transfer mechanism, an accurate determination of the radiant flux is necessary. This is particularly true of the solar receiver, for which the changing position of the sun leads to strong variations in the solar flux and in internal shading patterns, and thus different receiver surfaces become radiantly important at different times, leading to the requirement of an accurate determination of the shape factors for all surfaces.

To accomplish this end, the computer code SHAPEFACTOR from the group of University of Washington energy-solving programs (UWENSOL) has been adapted for solving central-receiver cavity problems. SHAPEFACTOR is designed to compute the long-wave radiant energy exchange factors between surfaces which make up an enclosure. In its original form, the program computes the shape factors for a sequence of enclosures. As a stand-alone program, however, it is most likely to be used to compute the shape factor for a single enclosure. The abbreviated instructions in this manual pertain only to a single enclosure. To treat more than one enclosure the user should refer to the UWENSOL input manual.<sup>1</sup>

This description of SHAPEFACTOR is arranged in the following format: (1) shape factors and some definitions, (2) program flow, (3) subroutines, (4) the data necessary to run the program, (5) special considerations (Appendix A), and (6) program details (Appendix B).

## General Comments

For SHAPEFACTOR to compute the shape factors between the walls (surfaces) which make up a room, each wall must be flat and have either three or four edges (i.e., be a triangle or a quadrilateral).<sup>\*</sup> The room may have surfaces which obstruct radiation between other walls (see Figure 1). Surfaces are defined by entering the coordinates (locations) of each of the corners and by entering a list of corner node numbers. With this input data format the program is ideally suited for use with a finite-element mesh generation program.

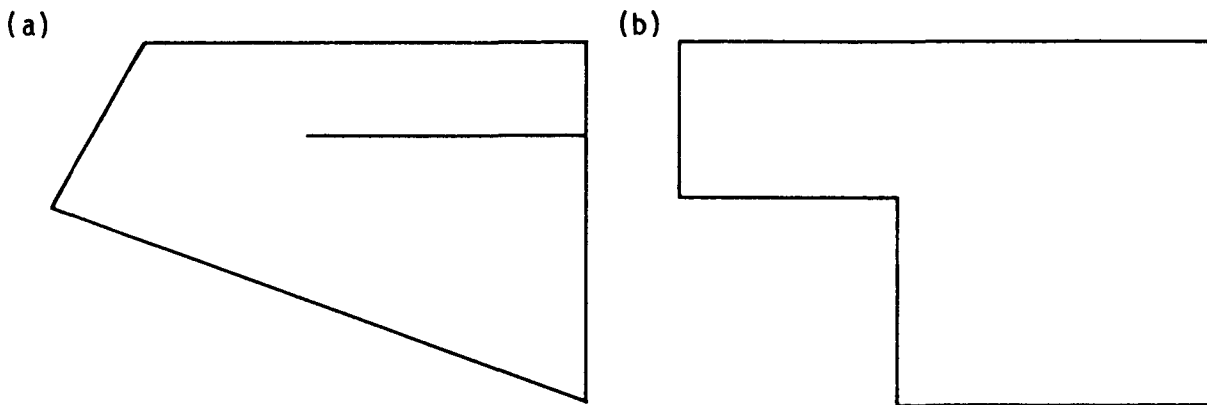


Figure 1. Types of Obstructions

If the surfaces close, i.e., there are no areas through which radiative can escape, the program can correct the shape factors to ensure that the sum of the shape factors from each surface to all other surfaces is one. This correction attempts to compensate for inaccuracies in the calculational procedure and data input by adjusting the shape factors so that the sum of all shape factors for one surface is exactly 1. If the room is not closed as in the case of radiation between two planes or in a solar receiver, no correction technique is available and the accuracy of the results primarily depends on the accuracy with which the surfaces are described by the user.

The shape factors are computed by means of the contour integration technique originally developed by Mittal and Stevenson.<sup>2</sup> This method produces accurate results even for surfaces which have a common edge. However, when the program is instructed that there is the possibility of a surface which obstructs the radiation between any two walls, their shape factor is calculated by means of a double area integral technique. Because this technique suffers a severe loss in accuracy whenever two points are very close together, the shape factor between surfaces with a common edge may be

---

<sup>\*</sup>This is not as restrictive as it sounds. See Appendix B for means of treating arbitrary polygons.

incorrectly evaluated. To avoid such a problem, the user must define the space in such a way that no two surfaces whose mutual view is obstructed share a common edge. The section on data input contains examples of such problems and the techniques for avoiding them.

### Definitions

Wall	A generic term for a flat plane with two surfaces. For the outer walls of a room, one surface is presumed to face into the room (the active surface), the other surface to face out (the inactive surface). SHAPEFACTOR requires that walls be either triangles or quadrilaterals.*
Room	A series of contiguous walls enclosing a space that may possibly contain other surfaces. A room may consist of as many walls and other surfaces as desired.
Active Surface	That surface of a wall which faces into the room. It is presumed to be the surface upon which radiation falls and from which radiation is emitted.
Inactive Surface (Passive)	That surface of a wall which faces out of the room and to and from which no radiation is presumed to flow.
Edges	The straight lines, terminating in corners, that define a wall.
Corners	The point at which two edges of a wall intersect. All walls and surfaces are defined by either three or four corners and edges.
Shape Factor Matrix	The matrix of shape factors which represent the fraction of energy leaving one surface and arriving at another.
Original Shape Factor Matrix	The matrix of shape factors as originally computed by the program.
Corrected Shape Factor Matrix	The matrix of shape factors which has been corrected to ensure that the sum of the shape factors from any one surface to all of the other surface is identically equal to 1.
Obstructing Surface	Any surface so placed that it intercepts radiation passing between any two active surfaces.
Included Surface	A surface that can be entirely contained within a larger surface. The two surfaces must be coplanar, but the program does not check to ensure that they are coplanar.

---

\*A larger number of edges can be used by recompiling the program with only one dynamic dimension and one variable change.

Combined Surfaces	Two or more surfaces may be combined to give one larger surface. The assemblage of the smaller surfaces is said to be a combined surface.
Coordinates	The x, y, z values defining a corner.
Coordinate System	A right-handed coordinate system whose location and orientation are arbitrary, to which all coordinates refer.

### Program Flow

The program treats the data in the following sequential fashion:

- A. Scanning the data, the program determines the number of corners and surfaces, the number of surfaces to be combined, and the number of surfaces which obstruct the view of other surfaces. The program then dynamically dimensions the required storage.
- B. The input information is read and all information stored in the appropriate dimensioned variables.
- C. The length of each variable and its location in the unlabeled common block is written to the output file.
- D. Using the contour integration technique of Mittal and Stevenson,<sup>2</sup> the program calculates the shape factor between surface one and every other surface, then surface two and every other surface but number one, and so on. Thus, shape factors are evaluated only for the upper triangular portion of shape factor matrix; the lower triangular entries are calculated through the reciprocity theorem.

The contour integration technique gives either positive or negative values of the shape factor, depending upon whether the surface is traversed in a clockwise or counter-clockwise direction. The program disregards the sign as calculated and assumes that the active surface of each wall sees the active surface of every other wall. Consequently, there is no checking to determine whether any two surfaces can actually see each other.

- E. The program determines if any obstructing surfaces exist. If no obstructions exist, the program proceeds to a consideration of the combine option. If obstructions are found to exist, then the program considers the effect of these obstructions according to the following method.
- F. Obstructions are not determined by the program but are input by the user. If the user is uncertain as to whether a given surface is an obstruction, the user may define all surfaces as obstructions. In this case every surface is treated as a possible obstruction, with a consequent increase in computational time and storage requirements.

The procedure for treating obstructions is as follows:

- (1) First,  $n^2$  points are located in each surface such that the areas associated with each point are as equal as possible.
  - (2) In each surface, a local coordinate system is defined such that its z axis points toward the inside of the enclosure. This coordinate system is defined by means of the numbering of the corners of the surface; all surfaces are presumed to be viewed from inside the enclosure, with the surface traced out from corner to corner in a counter-clockwise fashion. If the direction is reversed, the effect will be to define the active surface of a wall to be looking out of, rather than into, the enclosure. Note that if the obstruction option is not exercised, the direction of traversal is immaterial.
  - (3) A ray is drawn from the first of the  $n^2$  points in surface 1 to the first of the  $n^2$  points in wall surface 2. This ray is compared with the directions of the inward-facing normals from surfaces 1 and 2, and the program determines whether the active surfaces of walls 1 and 2 actually see each other regardless of whether an obstruction between them may exist. If the two active surfaces cannot see each other, the shape factor is set to zero and the program proceeds to examine the shape factor between surfaces 1 and 3, etc.
  - (4) If the program determines that the active surface of wall 1 can see the active surface of wall 2, it then asks if there are any obstructions. If no obstructions are declared, the program then continues by treating the radiant exchange between surfaces 1 and 3, 1 and 4, etc.
- G. If an obstruction has been defined, the program extends the ray between the first of the  $n^2$  points in surface 1 and the first of the  $n^2$  points in surface 2 and determines the point at which this ray intersects the obstructing surface, call it surface 0. Then, using the local coordinate system of surface 0, the program determines whether this point of intersection lies inside or outside the area of surface 0 as determined by the enclosing sides. If none of the rays between surfaces 1 and 2 lie in the area of surface 0, the shape factor between surfaces 1 and 2 is unchanged. If any one of the rays is found to lie in the area of surface 0 and therefore to be obstructed, then the program recomputes the shape factor between surfaces 1 and 2 by using the double area integration technique. In this integration, a ray is extended from each of the  $n^2$  points in surface 1 to each of  $n^2$  points in surface 2, giving a total of  $n^4$  such rays passing through surfaces 1 and 2. Each ray is checked for interception with surface 0; if there is an interception, the contribution of that ray to the double integral is set equal to zero. The result then is an approximate value of the shape factor between surfaces 1 and 2.
- H. The program determines if any surfaces are to be combined. If they are, then the appropriate shape factors for the combined surfaces to all other surfaces are calculated, the surfaces are renumbered, and the renumbering of the surfaces and the new areas are printed out.

- I. If requested, once the program has completed the shape-factor calculation, either with or without consideration of obstructions, the shape factor matrix is corrected. The program examines each row of the matrix and checks that all terms in that row add to 1.0. If the sum is less than .9 or greater than 1.1, a diagnostic message is printed, the computations continue, but the program is aborted at the end of the correction phase and no final output is given. If the sum is found to be between 0.9 and 1.1, then all of the terms to the right of the diagonal are scaled upward or downward such that the sum is exactly 1.0. The consequence of this correction algorithm is that the very last surface in the matrix has no entries to the right of its diagonal and consequently cannot be corrected to ensure that the sum is 1.0. By looking at the corrected shape factor matrix and the sum of the last row of this matrix, the difference between this sum and 1.0 is an immediate indication as to the success of the program. Typically, the error should be of the order of 0.001, but in some conditions it may exceed this.

## Description of Routines

### Routines SHAPFA and SHAPE

SHAPFA scans the data to determine the storage requirements of the program and dynamically allocates storage in the unlabeled common to accommodate all the requirements of the program. It then calls subroutine SHAPE, the main routine. All input data are interpreted in this routine and all computations are performed, either by calling the appropriate subroutines, or by SHAPE itself. SHAPE also carries out printing of the final results and writing on tape.

SHAPE calls the following routines: MASK, PLANEQ, OBSTRUC, INCLUDE, and COMBINE. Routine PLANEQ is used to define the equation of a plane in the form  $ax + by + cz + d = 0$ . It returns the coefficients a, b, c, d with the first coefficient being positive. Only the coordinates of the first three corners of the surface are used. The input to PLANEQ is through the argument list.

### Routine OBSTRUC

OBSTRUC calls the following subroutines: CHECK, ENORMAL, INTRCPT, PLANEQ, and WPOINTS. OBSTRUC is the driving routine for the computation of the effect of obstructing surfaces on radiation between any two other surfaces. Its organization is as follows: (a) Determines the coefficients of the plane of each surface by calling PLANEQ. (b) Sets up the array of  $n^2$  points on each surface by calling WPOINTS. (c) Establishes a new coordinate system in each one of the prospective obstructing surfaces by calling ENORMAL. (d) Examines the shape factors between each one of the surfaces. If the shape factor is zero, no further processing of the radiation between those two surfaces is done. If the shape factor is not zero, OBSTRUC examines the vector drawn between each of the  $n^2$  points on surface 1 to each of the  $n^2$  points on surface 2. A variable XMIST is incremented by one if the subroutine OBSTRUC determines that the two points under question cannot see each other.

This is done by taking the product of the ray between the two points and the outward normal to the active side of each surface. If the points can see each other, the program checks to determine the point of intersection of the ray between the two points and any obstructing surfaces by calling the subroutine INTRCPT. After all the points are examined, the shape factor is then defined to be either the original shape factor if all points were found to be able to see each other or determined to be the result of the double integration if any of the  $n^4$  rays were found to be obstructed.

#### Routine ENORMAL

This routine determines a new coordinate system in which the z axis is normal to the active surface of a wall. The new routine returns the transformation matrix necessary to rotate the original x,y,z coordinate into the new coordinate system x',y',z'. The rotation is about the x axis and then about the y axis such that the z' axis is normal to the active surface of the wall. The routine then computes the location of all of the corners of the obstructing surface in terms of the x',y',z' coordinates system, finds the centroid of the surface, and then moves the origin of the x',y',z' coordinate system to the centroid. It also defines a normal to the edges of the obstructing surface, ensuring that these normals are outward normals as seen when traversing the obstructing surface in a counter-clockwise direction.

#### Routine WPOINTS

This routine defines the x,y,z location of the  $n^2$  points to be located in each surface such that the area associated with each of the  $n^2$  points is equal. If the surface is a triangle or a rectangle, each point will have an equal area associated with it. If the surface is a quadrilateral, but not rectangle, the area of each point will differ. In effect, each point is the centroid of an area defined by drawing isoparametric lines in the element.

#### Routine INTRCPT

This routine determines if a ray actually intercepts an obstructing surface. It first finds the point of intersection of the ray with a plane, infinite in extent, in which the obstructing surface lies. It then uses the coordinate transformation defined in ENORMAL to determine if the point of intersection is in the obstructing surface. This is done by drawing a ray from the centroid of the obstructing surface to the point of intersection. The dot product is then taken between this ray and the normal to the edge of the obstructing surface. If the dot product is less than zero, then the point is to the left of the edge of the obstructing wall as traversed in the counter-clockwise direction. If the dot product of this ray with the normal at any one of the corners is positive, then the point is outside of the obstructing surface and no obstruction for that ray is defined.

### Subroutine INCLUDE

The routine INCLUDE is used if it is desired to define one surface to be included in a second surface. In effect the program computes the shape factor from the first (inner) surface to all the remaining surfaces, computes the shape factor from the second (encompassing) surface to all the remaining surfaces. Using shape factor algebra the program then deducts the shape factor of the first surface from the shape factor of the second surface, leaving two shape factors: that of the second surface being simply the shape factor from the area outside of the first surface and that of the first surface being the originally calculated value. It is assumed that the two surfaces are coplanar, but no check is made for this. Consequently, if surfaces 1 and 2 are not coplanar but surface 1 is said to include surface 2, the results may be in error. The amount of the error depends upon the non-coplanar nature of the two surfaces. A warning that you are using the INCLUDE option is printed in the output, but no other checking is done.

### Routine COMBINE

The routine COMBINE is used to combine several surfaces into a single surface. If one, for example, says that surfaces 1, 3, and 4 are to be combined, then the resulting shape factor will be the algebraic sum of the shape factors associated with 1, 3, and 4, accounting for their respective areas. In combining the surfaces, the routine renumbers the surfaces and a table of old numbers and new numbers, and old areas and new areas is printed. To avoid renumbering important surfaces, the user should input the primary surfaces first and then the surfaces which are to be combined. Thus, for example, if you wish to combine surfaces 5 and 1 and surfaces 6 and 2, the program will do the combination and eliminate surfaces 5 and 6 from consideration, leaving surfaces 1, 2, 3, and 4. However, surface 7 will now be renumbered as surface 5; consequently, any reference to surface 7 in a subsequent program will not be directed to the original surface 7 but to the original surface 9 (9 reduces to 7 when surfaces 5 and 6 are eliminated). If you started with nine surfaces and intend to eliminate surfaces 5 and 6, you should number the surfaces such that surface 8 is the one you expect to combine with 1 and surface 9 the one you expect to combine with 2. Upon combination you will be left with seven surfaces, new surface 1 will be the originals 1 plus 8, new 2 will be originals 2 plus 9, and the numbering of surfaces 3-7 will be as originally input.

### Routine MASK

MASK examines 70 alphanumeric characters and extracts from them the integers, reals, and alphanumeric variables. A description of this free form input is given in the next section. The program can consider only 20 reals and 20 integer or alphanumeric quantities. An alphanumeric quantity is considered to be an integer from the point of view of the program. Consequently, one must be careful not to write more than a combination of 20 integers or alphanumeric quantities.



## Data

### Directive

A line of data (often called a card or card image) which causes the program to perform a prescribed action.

### Data Input Format

All input is in the form of key word data: The first ten columns of each directive are reserved for a descriptor which characterizes the type of data on that card. Columns 11 through 80 specify the necessary data in a free-field format, with each datum being ten characters or less and with a maximum of ten integers and ten real numbers per data card.

To describe the form of the input data, it is convenient to define the following terms:

Digit	The numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0.
Integer	Any sequence of digits, which may be preceded by a + or - sign, whose total length is 10 characters or less (example: 1).
Real	Same as an integer except it must contain a decimal point [.] or an exponentiation symbol [E] within its field of 10 characters (examples: 1. or 1.E-3).
Delimiter	Any Hollerith symbol, or string of Hollerith symbols, not containing any of the following: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, +, -, E, or period. For example, the following symbols, or entire string of symbols, are interpreted as delimiters: [; : > < * / \$ ( ) ,]. Blanks may also be used as delimiters.

Each card requires a certain number of integers and reals to be specified in the data field. These numbers must be separated by delimiters. Of the possible delimiters, commas and strings of blanks will be found to be the most convenient. Although the data may start anywhere in Columns 11 through 80, the correct order of input variables is important. The order of the integers and reals with respect to each other is immaterial; that is,  $I_1, R_1, I_2, R_2$ , or  $I_1, I_2, R_1, R_2$  will both produce the same result. However,  $I_1, I_2, R_2, R_1$  will not produce the same result as the previous set.

All variables are preset to zero and all input quantities required for each card must be specified. When the end of the card is reached, any remaining integers or real quantities which were not specified will remain at their initial values of zero.

The input cards may be assembled into a data deck in any order. The last card in each data deck must be an END card. The formats below give several examples of data cards that all produce the same result. (By use of the key word descriptor, the order of the data input is not important, unless otherwise noted.)

```
COORDINATE  1 0. 0. 0.  2 0. 16. 10.  3 16. 16. 10.
COORDINATE  1,0.,0.,0., 2,0.,16.,10., 3,16.,16.,10.
COORDINATE  1,2,3,0.,0.,0., 0.,16.,10., 16.,16.,10.
COORDINATE  1,2,0.,0.,0., 3,0.,16.,10., 16.,16.,10.
```

### Data Input Required

The data input necessary for this program is as follows. The first data card must have the key word SPACE and must list the number of surfaces and the number of corners to be considered. Since the numbers listed are used to allocate storage, they may be larger than the numbers actually used but not smaller. The next data cards can be given in any order. They are composed of the following cards.

Coordinate Cards--Using the key word COORDINATE, one lists the three coordinates associated with each corner point in the form: corner point number, x,y,z. From one to six corner points may be listed on any coordinate card.

Surface Cards--Using the key word SURFACE, one enters the corner point numbers for each surface. Only the data for one surface may be entered per card. The corners may be entered in the counter-clockwise or clockwise direction, but if the OBSTRUCT option is used, they must be entered in the counter-clockwise direction. If one inadvertently interchanges corner numbers (making an x-shaped quadrilateral), the program will not detect that you have described an unusual shape and will attempt to compute.

Intercept Point--The key word INTPT defines how many points you wish to locate in each surface when considering the OBSTRUCT option. At the current time, the program will accept either 1, 4, 9, or 16 points (i.e., two 1x1, 2x2, 3x3, or 4x4 grids of points). The data entry for these are, respectively, one, two, three, or four. Default is three.

Obstruction--Using the key word OBSTRUCT the user enters the number of the surface which is to be considered as an obstructing surface. If you are not sure whether a surface is an obstructing surface or not you should enter it. If the problem is so complex that many surfaces are likely to be obstructing surfaces, enter the word ALL and every surface will be considered to be an obstructing surface. If you do this, storage is required for each one of these surfaces and the program will execute very slowly.

Include--Using the key word INCLUDE, list the surface number which is to include other surfaces, then list the surface numbers for those surfaces which are included. The program will not check to determine if two included surfaces overlap each other, that is, if surfaces 2 and 3 are included in surface 1 and one has inadvertently overlapped them such that the areas of surfaces 2 and 3 exceed that of surface 1, the program will not recognize this fact.

Combination--Using the key word COMBINE, the user enters the surface numbers for those surfaces which are to be combined into one.

Enclosure--The key word ENCLOSURE causes the program to modify the calculated shape factors so that the sum of each row is 1.

End--The data input for the program ends with a card with the key word END.



## APPENDIX A

### COMMENTS AND CAVEATS

#### Internal Obstructing Walls

Any wall within a room has two surfaces, both of which must be declared as (active) surfaces. Furthermore, these surfaces must be distinct, that is they cannot be defined by the same corner coordinates. It is easiest to imagine that the wall has a slight thickness.

#### Gaps

The program corrects the shape factors to ensure that the sum of the shape factors for all the surfaces is unity. If the room is not closed (i.e., if it has gaps between the outer walls), the corrected shape factors will be in error. Because of the form of the correcting algorithm, the presence of gaps is usually indicated by one or more surfaces having a sum much different than 1.0.

#### Coplanar Surfaces

When calculating the shape factors, if two surfaces are such that surface 1 is extremely large and surface 2 is extremely small, and furthermore surface 2 is nearly coplanar with surface 1, then the shape factor  $F_{21}$  will be nearly equal to one. Consequently, according to the reciprocity theorem there will be a non-zero shape factor from surface 1 to surface 2, even though the true value should be close to zero. The program does no checking for such abnormal physical placement of surfaces, and the user should be sure that if a situation does exist, that the large surface is numbered before the small surface. In this way, the shape factor  $F_{12}$  will be computed to be nearly zero and the shape factor  $F_{21}$  will also turn out to be essentially zero.

#### Coordinate System for Obstructions

Although the coordinate system is arbitrary, the corners of surfaces must be numbered in a counter-clockwise direction as seen from inside the enclosure.

## Included Surfaces

If a surface, 1, includes another surface, 2, and surface 1 is an obstructing surface, do not declare surface 2 as an obstructing surface. Such a declaration will not affect the computed shape factors, but will increase execution time.

## Included Obstructing Surfaces

If surface 2 is included in surface 1 and surface 1 is an obstructing surface, then the shape factor from any other surface to surface 2 will be incorrectly calculated if surfaces 1 and 2 are exactly coplanar. This occurs since a ray drawn to a point in surface 2 will fall in the plane of surface 1 and will be considered to be obstructed. Since the include option automatically sets  $F_{12} = F_{21} = 0$  upon the assumption that surfaces 1 and 2 are coplanar, if you displace surface 2 an infinitesimal amount away from surface 1, the results will be correct.

## Obstructing Surfaces

The program tends to work best when the obstructing surfaces are entered last (i.e., the highest numbered).

## Error in the Sum

If the Include or Obstruct option is exercised, the sum of the shape factors at each row is not checked to see if it is close to 1.0, so the user should check this sum for the obstructing walls especially. If a sum is too far from 1.0, the number of rays should be increased. Increasing the number of rays, however, is not a guaranteed method of improving the accuracy.

## Data Input

Be careful not to enter more than 20 reals or 20 integers on a data card.

## APPENDIX B

### PROGRAM DETAILS

#### Introduction

The following information is abstracted from the UWENSOL Users's Manual to amplify some of the discussion in the main text, prescribe the precise form of the data input, describe some additional directives, and illustrate a typical example treated by the code. Although the example is from the set of architectural cases described in the UWENSOL manual, the example contains all the features relevant to a typical engineering problem.

#### Information Needed to Use SHAPEFACTOR

The user must provide data relating the location or coordinates of the corners of all subsurfaces in the room and then a listing of each subsurface in terms of these corners. SHAPEFACTOR then calculates the shape factors and reports them to the user as well as placing them in a file called SHAPE, which can be stored and then accessed at a later date.

#### Consequences of Using SHAPEFACTOR

To use SHAPEFACTOR you will need to break the room into as many pieces (component surfaces) as necessary to describe the section and orientation changes. In addition, because the program can only recognize surfaces bounded by three or four edges, arbitrary subsurfaces may need to be defined in order to correctly describe the component surface. These are usually generated by extending known boundary lines to form a checkerboard or plaid pattern.

To correctly describe the wall in Figure A-1a, one must define the two surfaces shown in Figure A-1b. Some other alternatives are shown in Figure A-2. In any case, one "real" component surface here requires the use of two "artificial" subsurfaces for correct input. Be especially careful that no two surfaces have the same corner coordinates.

With the use of the COMBINE option, discussed below, it is possible to merge the two subsurfaces back into the one component surface.

The "three- and four-sided figures only" rule causes other difficulties, too, as seen in Figure A-3. The problem is to describe the location of the "window" with surfaces of only three or four edges. There are two ways to handle the situation, both illustrated in Figures A-4 and A-5.

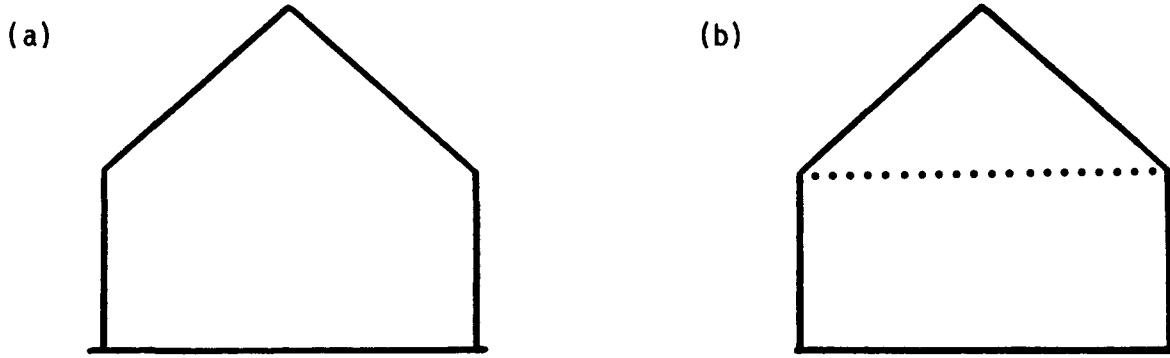


Figure A-1. Wall with Five Edges Artificially Broken into a Triangle and a Rectangle

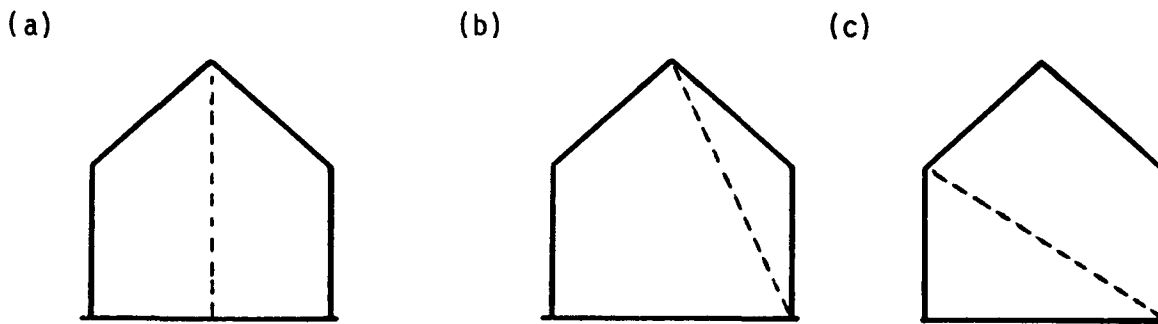


Figure A-2. Other Ways of Inputting the Wall Shown in Figure A-1a

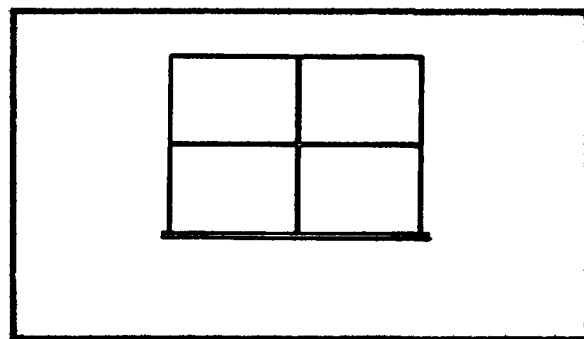


Figure A-3. Wall with a Window



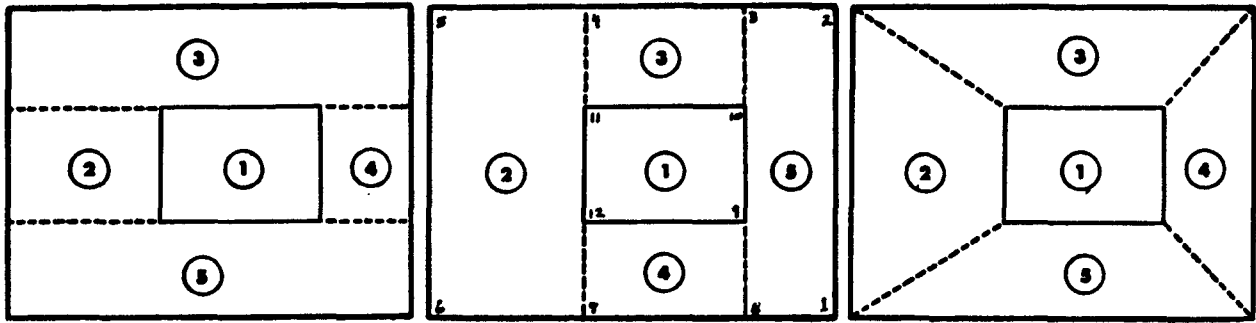


Figure A-4. Possible Ways to Treat a Wall with a Window

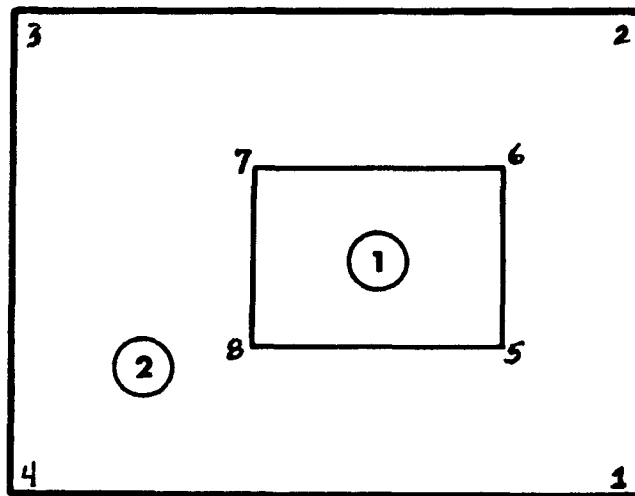


Figure A-5. Another Way to Treat a Wall with a Window

### Using COMBINE

Divide the wall surface into a number of subsurfaces and later COMBINE them again. A number of alternate divisions are possible, as shown in Figure A-5, but the example uses the middle one to illustrate the use of the COMBINE directive.

See the section on directives, later in this appendix, for clarification of the meaning or syntax if any of the directives is unclear.

<u>Associated Directives</u>	<u>Interpretation</u>
(only part of a data set)	
.	
.	
.	
SURFACE 1 9 10 11 12	Surface 1 is bounded by points 9, 10, 11, 12. Surface 2 is bounded by points 7, 4, 5, 6. Et cetera.
SURFACE 2 7 4 5 6	
SURFACE 3 10 3 4 11	
SURFACE 4 8 9 12 7	
SURFACE 5 1 2 3 8	
.	
.	
.	
COMBINE 2 3 4 5	Combine, into one surface, surfaces 2, 3, 4, and 5 (the combination will hereafter be referred to as "surface 2").

### Using INCLUDE

Treat the two surfaces independently, but state that the window is INCLUDED in the larger wall surface. Again, compare Figure A-5 with the associated directives below, and their interpretation.

<u>Associated Directives</u>	<u>Interpretation</u>
(only part of a data set)	
.	
.	
.	
SURFACE 1 5 6 7 8	Surface 1 is bounded by points 5, 6, 7, and 8. Surface 2 is bounded by points 1, 2, 3, and 4.
SURFACE 2 1 2 3 4	
.	
.	
.	
INCLUDE 2 1	Surface 2 wholly includes surface 1.

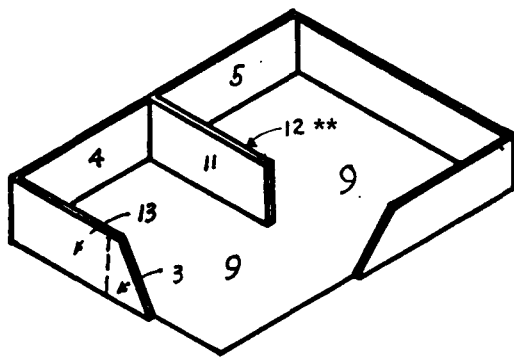
### How They Work

INCLUDE subtracts the effects of the included surface from the including surface; in effect it replaces part of the larger surface with the smaller. The boundaries of the included surface must be wholly contained within the boundaries of the including surface, or share only all or part of a side with the including surface. The including surface must be given by a single SURFACE directive, not built up with COMBINE. COMBINE adds together the effects of the subsurfaces to produce a single, larger surface.

CAUTION: A given surface should be treated by either the COMBINE or the INCLUDE procedure, but not by both.

SHAPEFACTOR has the ability to calculate shape factors for surfaces wholly or partially hidden from one another, but the algorithm which does the calculations is severely hampered if a surface has an edge in common with a surface it does not totally see. In Figure A-6 such a condition exists between the floor (9) and the walls 3, 4, 5, and 13. The partition "sits" on the floor, but no part of the floor can see all of the partition.\* Figure A-6b is the solution; break the floor into two subsurfaces, both of which see their side of the partition, with which they share an edge, but do not see all of the opposite side of the partition, with which they do not share an edge.

(a)



(b)

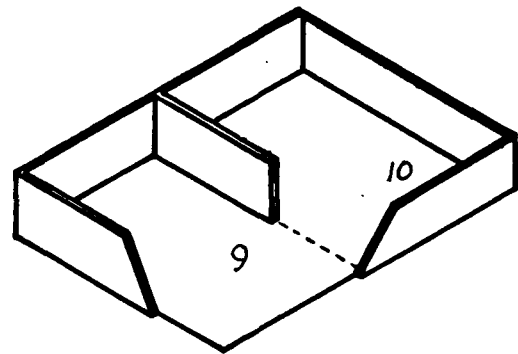


Figure A-6. Room with a Partition

### Coordinates

Each entry gives the coordinates of the appropriate point in space, using a right-handed coordinate system with origin anywhere, though a floor corner is best. There is no limit on the number of coordinates which can be assigned.

---

\*Note that the projecting partition has been described by two surfaces (11 and 12). This is an exception to the dictum stated in the definition of surfaces, but is necessary in this shape factor algorithm to establish the direction in which the surface faces (established by the counter-clockwise sequence of corners, described in the section on directives later in this appendix).

\*\*Note that surfaces 11 and 12 must be distinct (i.e., have different corner coordinates).

### Surface Name

This is the number by which the surface is known to succeeding programs, and therefore it is vitally important that the correct number refer to the correct surface in all programs. Note that if a COMBINE option is used, the surfaces will be renumbered to cover any gaps in the numbering caused by the COMBINE. Be sure to check the output column headed "New Surface Numbers" since the shape factors produced by the program are those associated with the new, not the old, surface numbers.

### Corners

The three or four corners, or vertices, of a subsurface are used to locate it in the model. The order in which they are given implies the direction in which the surface is said to "face." Consequently, the following suggestion: always input the corners in counter-clockwise order, as seen from the "inside" of the room in question, starting with the lower right-hand corner. Be especially careful of ceilings, where looking "down" is often a difficult habit to break.

### Includes

This column is used to give the numbers of any surfaces which are wholly included within the bounds of the surface in question.

### Combines With

If a component surface has been artificially divided into a number of subsurfaces, their numbers should be recorded to be combined with the lowest numbered one of the lot.

### Obstructs

If the surface in question is in a position where it is possible to intersect it with a straight line drawn between two other surfaces in the space, it obstructs both of those surfaces.

## Directives

### TITLE Directive (optional)

The TITLE directive is used to label the output. It is identical to the other TITLE directives in use and limitation.

```

-----
0      1      2      3      4      5      6      7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
TITLE   :[Any user-defined text, up to 70 characters per directive, up to
:        two directives in the data set]
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0      1      2      3      4      5      6      7
-----

```

### SPACE Directive

The SPACE directive establishes the number of component surfaces in a room and the number of surface corners necessary to describe those surfaces. If the entered numbers differ from the number of SURFACE or COORDINATE directives, the program will generate an error message and stop. The SPACE directive must precede all but the title cards.

```

-----
0      1      2      3      4      5      6      7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
SPACE   : I-surfaces, I-corners
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0      1      2      3      4      5      6      7
-----

```

### COORDINATE Directive

The location of the defining corners of each component surface is input through use of the COORDINATE directive. The coordinates must be measured in feet, with unique values within each SPACE, i.e., there can be only one coordinate system per room.

```

-----
0      1      2      3      4      5      6      7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
COORDINATE: (I-coord.no., X-coord., Y-coord., Z-coord.), . . .
:
COORDINATE: 1 (0.,0.,0.) 2 (0.,0.,10.) 3 (0.,10.,10.)
[or]
COORDINATE: 1 2 3 (0.,0.,0.) (0.,0.,10.) (0.,10.,10.)
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0      1      2      3      4      5      6      7
-----

```

## SURFACE Directive

The SURFACE directive assigns an identifying number to each subsurface of the room and identifies its boundaries in terms of the coordinates of its corners. Note the direction in which the surface faces is implied from the order in which the bounding corners are given. They MUST be given in counter-clockwise order, as seen from the inside of the space. We recommend that the user always begin numbering from the lower-right-hand corner of the surface.

```
-----  
0          1          2          3          4          5          6          7  
1234567890:12345678901234567890123456789012345678901234567890123456789012345678  
:  
SURFACE   : I-surf.no., (I-coordinate no.), . . .  
:  
SURFACE   :      1      1 2 3 4      [SURFACES MUST HAVE THREE]  
SURFACE   :      2      4 3 2 1      [ OR FOUR SIDES ONLY ]  
SURFACE   :      3      5 4 3  
:  
1234567890:12345678901234567890123456789012345678901234567890123456789012345678  
0          1          2          3          4          5          6          7  
-----
```

In the examples above, surfaces 1 and 2 are opposite sides of an infinitely thin wall, or plane, but both must appear if the wall protrudes into the SPACE, because radiation cannot reach the "back" of a surface. The third example illustrates a three-sided surface.

## COMBINE Directive

Having established the subsurfaces of a larger component surface, it is possible to obliterate the imaginary lines of the patchwork; those drawn to maintain the "three- and four-sided figures" rule, but not marking a thermal change. This is done by use of the COMBINE option.

This directive is used to combine contiguous, coplanar subsurface (i.e., of the same material composition) into a single component surface. The directive reads "combine all these subsurfaces into one component surface and renumber all surfaces in the room."

The example refers to Figure A-5, reconstructing the original configuration of Figure A-3.

```
-----  
0          1          2          3          4          5          6          7  
1234567890:12345678901234567890123456789012345678901234567890123456789012345678  
:  
COMBINE   : I-in surface no., (I-surface no.), . . .  
:  
COMBINE   : 2 3 4 5  
:  
1234567890:12345678901234567890123456789012345678901234567890123456789012345678  
0          1          2          3          4          5          6          7  
-----
```

Note, because the program automatically renumbers the surfaces after a COMBINE option is exercised, two strategies are suggested:

1. Number all component surfaces first, then add in and number the subsurfaces necessary to execute the program (there is nothing sacred about the order of numbering). When the COMBINE directive is used, be sure to combine the extra surfaces into the original (i.e., lower numbered) surface. The renumbering will then have no effect on the numbers you have assigned to the component surfaces.
2. IN ANY EVENT, examine the output from SHAPEFACTOR before moving on to CONSTRUCT and pay special attention to the "New Surface Number" column because it is by these numbers that you must henceforth refer to the surfaces of the model if the results are to have any meaning.

### INCLUDE Directive

The INCLUDE directive permits the user to specify that a given subsurface or surfaces is or are wholly included in a larger sub- or component surface. As many surfaces as necessary may be included in this fashion, but you may not "nest" inclusions as shown in Figure A-8a. Also, care should be taken that the surfaces are distinct and coplanar. Either of the cases shown below would produce incorrect results, without generating a FATAL error.

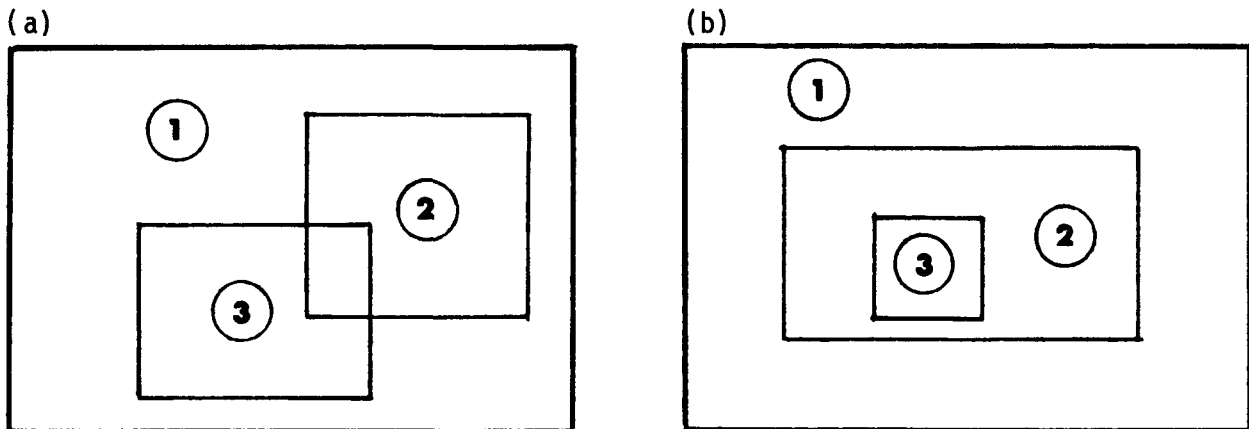


Figure A-8. (a) Incorrectly Nested Surfaces  
(b) Correctly Nested Surfaces

In the example, as noted above, a surface whose boundaries are within another surface's boundaries, or which share some boundary totally or in part, may be INCLUDED in that surface. The window (surface 1) may be treated as described in COMBINE, or by use of the above INCLUDE directive.

```

-----
0           1           2           3           4           5           6           7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
INCLUDE    : I-in surface no., (I-surface no.), . . .
:
INCLUDE    : 2 1
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0           1           2           3           4           5           6           7
-----

```

OBSTRUCT Directive (optional) (DEFAULT = no obstructions)

The OBSTRUCT directive informs the program that the surfaces listed may obstruct the view between other surfaces. Although the condition can exist in section as well as plan, the most common occurrence is that of the L-shaped room: Lines drawn from parts of one surface cannot reach parts of some other without being intercepted by the obstructing surface.

Accuracy of the algorithm which the directive requests is greatly reduced if, the surfaces with an obstructed view of each other share a common edge or side, so this should be avoided, by making an arbitrary subdivision if necessary.

```

-----
0           1           2           3           4           5           6           7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
OBSTRUCT   : I-surface number, I-surface number
[or]       :
OBSTRUCT   : ALL
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0           1           2           3           4           5           6           7
-----

```

INTPT Directive (optional)

Defines the number of rays used by the OBSTRUCT option.

Enclosure Directive

Instructs the program to implement the correcting algorithm to ensure that the sum of the shape factors is unity.

ORIGINAL Directive (optional)

Prints out the original shape factor matrix before it is corrected to ensure that the sum of each row equals 1.



## END Directive

The arrangement of directives ought to resemble the layout below, including as many spaces as necessary. If desired, only those spaces which cannot be calculated by SIMPLE need be done by SHAPEFACTOR, but care should be taken that the order in which such spaces are calculated remains the same in both SHAPEFACTOR and CONSTRUCT. This is because CONSTRUCT reads the shapefactors from the file "SHAPE" in the same order they were written to it by SHAPEFACTOR; thus, the order in which SHAPEFACTOR computes and records the shape factor must be the same as the order in which CONSTRUCT reads them.

```

-----
0           1           2           3           4           5           6           7
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
:
END          : [no data]
:
1234567890:12345678901234567890123456789012345678901234567890123456789012345678
0           1           2           3           4           5           6           7
-----

```

```

TITLE      *[optional]*           [ Two lines maximum.      ]
TITLE      *[optional]*
SPACE <---- Begin space number one.
COORDINATE  Coordinates for the   [ Maximum of three per   ]
.           first space.         [ directive but no limit on ]
.           [ total number of coordinates. ]
.
.
.
SURFACE     [ Three or four sided surfaces]
.           [ only. ]
.
.
.
COMBINE     *[optional]*
INCLUDE     *[optional]*
OBSTRUCT    *[optional]*
SPACE <---- Begin space number two.
COORDINATE  Coordinates for the   [ New coordinate numbers and ]
.           second space, if they [ values supercede those   ]
.           are different.        [ previously input. ]
.
SURFACE
.
.
.           [Et Cetera]           [ Up to the ten space limit. ]
.
.
END          ** DO NOT forget the END directive.**

```

To reduce the storage requirements of the program, it is suggested that each space have its own coordinate system and that the numbering of these coordinates begin with 1.

## Sample Input and Output

### Input

```

SPACE          13 20
TITLE          HYPOTHETICAL HEALTH FACILITY
TITLE          DOUBLE OCCUPANCY BEDROOM
COORDINATE     1 23.  12.  0.    2 23.  12. 10.
COORDINATE     3 23.   0. 10.   4 23.   0.  0.
COORDINATE     5  8.   0.  0.    6  8.   0. 10.
COORDINATE     7  0.   0. 10.   8  0.   0.  0.
COORDINATE     9  0.  12.  0.  10  0.  12. 10.
COORDINATE    11  8.  12. 10.  12  8.  12.  0.
COORDINATE    13 8.01  0.  0.  14 8.01  5.  0.
COORDINATE    15 8.01  5. 10.  16 8.01  0. 10.
COORDINATE    17  8.   5.  0.  18  8.   5. 10.
COORDINATE    19 23.   6.  0.  20 23.   6. 10.
SURFACE        1  2   3   6  11
SURFACE        2  6   7  10  11
SURFACE        3  1   4   3   2
SURFACE        4  4   5   6   3
SURFACE        5  5           8   7   6
SURFACE        6  9  10   7   8
SURFACE        7 12  11  10   9
SURFACE        8 12   1   2  11
SURFACE        9  1  12   5   4
SURFACE       10  5  12   9   8
SURFACE       11 13  14  15  16
SURFACE       12  5   6  18  17
SURFACE       13  4   3  20  19
INCLUDE        3  13
COMBINE        7   8
COMBINE        1   2
COMBINE        9  10
COMBINE        4   5
OBSTRUCT      11  12
END

```

Output

```
#####  
# UNIVERSITY OF WASHINGTON #  
# #  
# PROGRAM #  
# ...UWENSQL... #  
# #  
# ----- #  
# SUB PROGRAM #  
# ..SHAPEFA.. #  
# #  
# VERSION 4.1 #  
# DATED APRIL 26 1978 #  
# #  
# THIS RUN WAS EXECUTED #  
# AT 08.14.42. ON 09/12/79 #  
#####
```

IMAGE OF THE DATA CARDS

```
SPACE 13 20  
TITLE HYPOTHETICAL HEALTH FACILITY  
TITLE DOUBLE OCCUPANCY BEDROOM  
ORIGINAL  
COORDINATE 1 23. 12. 0. 2 23. 12. 10.  
COORDINATE 3 23. 0. 10. 4 23. 0. 0.  
COORDINATE 5 8. 0. 0. 6 8. 0. 10.  
COORDINATE 7 0. 0. 10. 8 0. 0. 0.  
COORDINATE 9 0. 12. 0. 10 0. 12. 10.  
COORDINATE 11 8. 12. 10. 12 8. 12. 0.  
COORDINATE 13 8.01 0. 0. 14 8.01 5. 0.  
COORDINATE 15 8.01 5. 10. 16 8.01 0. 10.  
COORDINATE 17 8. 5. 0. 18 8. 5. 10.  
COORDINATE 19 23. 6. 0. 20 23. 6. 10.  
SURFACE 1 2 3 6 11  
SURFACE 2 6 7 10 11  
SURFACE 3 1 4 3 2  
SURFACE 4 4 5 6 3  
SURFACE 5 5 8 7 6  
SURFACE 6 9 10 7 8  
SURFACE 7 12 11 10 9  
SURFACE 8 12 1 2 11  
SURFACE 9 1 12 5 4  
SURFACE 10 5 12 9 8  
SURFACE 11 13 14 15 16  
SURFACE 12 5 6 18 17  
SURFACE 13 4 3 20 19  
INCLUDE 3 13  
COMBINE 7 8  
COMBINE 1 2  
COMBINE 9 10  
COMBINE 4 5  
OBSTRUCT 11 12  
END
```

THIS PROGRAM IS DIMENSIONED FOR 13 SURFACES AND 20 CORNERS  
 STARTING LOCATIONS OF VARIABLES IN BLANK STORAGE

VARIABLE	LENGTH	LOCATION IN STORE	ADDRESS	
			DEC	OCT
A	195	1	20899	050643
AA	13	196	21094	051146
BB	13	209	21107	051163
CC	13	222	21120	051200
DD	13	235	21133	051215
ADLA	52	248	21146	051232
AL	52	261	21159	051247
BETA	52	313	21211	051333
GAM	52	365	21263	051417
C	60	417	21315	051503
F	169	477	21375	051577
IC	20	646	21544	052050
IS	13	666	21564	052074
IVERT	52	679	21577	052111
NV	13	731	21629	052175
SUMF	13	744	21642	052212
T	52	757	21655	052227
ICOM	12	809	21707	052313
JICOM	10	821	21719	052327
INCL	3	831	21729	052341
JINCL	13	834	21732	052344
IOBST	13	847	21745	052361
XINT	117	860	21758	052376
YINT	117	977	21875	052563
ZINT	117	1094	21992	052750
AEDBS	8	1211	22109	053135
BEDBS	8	1219	22117	053145
RR	12	1227	22125	053155
CENTX	2	1239	22137	053171
CENTY	2	1241	22139	053173
DAREA	13	1243	22141	053175

OLD CH REQ= 22912 DEC      NEW CH REQ= 22160DEC      NEW STORAGE= 2000DEC  
 OLD CH REQ=0000054600 OCT      NEW CH REQ=0000053230 OCT      NEW STORAGE=0000003720 OCT

\*\*\*\*\*  
 WARNING--THE COMBINED SURFACE OPTION  
 IS USED--THE SURFACES ARE ASSUMED TO BE COPLANAR, BUT NO  
 CHECKING IS DONE  
 \*\*\*\*\*

\*\*\*\*\*  
 WARNING--THE INCLUDED SURFACE OPTION  
 IS USED--THE SURFACES ARE ASSUMED TO BE COPLANAR, BUT NO  
 CHECKING IS DONE  
 \*\*\*\*\*

\*\*\*\*\*  
 WARNING--THE OBSTRUCTION OPTION IS BEING USED  
 PLEASE CHECK THE MANUAL FOR DETAILED INSTRUCTIONS AND HINTS  
 \*\*\*\*\*

SHAPEFACTOR DATA AND RESULTS FOR ROOM 1  
 HYPOTHETICAL HEALTH FACILITY  
 DOUBLE OCCUPANCY BEDROOM  
 THE NUMBER OF SURFACES = 13

SURFACE= 1	23.000	12.000	10.000	23.000	0.000	10.000	8.000	0.000	10.000	8.000	12.000	10.000
SURFACE= 2	8.000	0.000	10.000	0.000	0.000	10.000	0.000	12.000	10.000	8.000	12.000	10.000
SURFACE= 3	23.000	12.000	0.000	23.000	0.000	0.000	23.000	0.000	10.000	23.000	12.000	10.000
SURFACE= 4	23.000	0.000	0.000	8.000	0.000	0.000	8.000	0.000	10.000	23.000	0.000	10.000
SURFACE= 5	8.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	10.000	8.000	0.000	10.000
SURFACE= 6	0.000	12.000	0.000	0.000	12.000	10.000	0.000	0.000	10.000	0.000	0.000	0.000
SURFACE= 7	8.000	12.000	0.000	8.000	12.000	10.000	0.000	12.000	10.000	0.000	12.000	0.000
SURFACE= 8	8.000	12.000	0.000	23.000	12.000	0.000	23.000	12.000	10.000	8.000	12.000	10.000
SURFACE= 9	23.000	12.000	0.000	8.000	12.000	0.000	8.000	0.000	0.000	23.000	0.000	0.000
SURFACE= 10	8.000	0.000	0.000	8.000	12.000	0.000	0.000	12.000	0.000	0.000	0.000	0.000
SURFACE= 11	8.010	0.000	0.000	8.010	5.000	0.000	8.010	5.000	10.000	8.010	0.000	10.000
SURFACE= 12	8.000	0.000	0.000	8.000	0.000	10.000	8.000	5.000	10.000	8.000	5.000	0.000
SURFACE= 13	23.000	0.000	0.000	23.000	0.000	10.000	23.000	6.000	10.000	23.000	6.000	0.000

AREAS OF SURFACES 1 THROUGH 13

1	180.0000
2	96.0000
3	120.0000
4	150.0000
5	80.0000
6	120.0000
7	80.0000
8	150.0000
9	180.0000
10	96.0000
11	50.0000
12	50.0000
13	60.0000

4 COMBINE CARDS WERE READ

SURFACE 7 CONTAINS THE SUBSURFACES

8

SURFACE 1 CONTAINS THE SUBSURFACES

3

SURFACE 7 CONTAINS THE SUBSURFACES

10

SURFACE 4 CONTAINS THE SUBSURFACES

5

1 INCLUDE CARDS WERE READ

SURFACE 3 INCLUDES SUBSURFACES

13

NEW AREAS AFTER THE INCLUDE OPTION

1 180.00000  
 2 96.00000  
 3 60.00000  
 4 150.00000  
 5 90.00000  
 6 120.00000  
 7 80.00000  
 8 150.00000  
 9 180.00000  
 10 96.00000  
 11 50.00000  
 12 50.00000  
 13 60.00000

THE AREAS AND THE OLD SURFACE NUMBERS ARE

NEW SURFACE NUMBER	NEW AREA	OLD SURFACE NUMBERS
1	276.00000	1 2
2	60.00000	3
3	230.00000	4 5
4	120.00000	6
5	230.00000	7 8
6	276.00000	9 10
7	50.00000	11
8	50.00000	12
9	60.00000	13

CORRECTED SHAPE FACTOR MATRIX (LAST COLUMN IS SUM OF THE ROW)

0.0000	.0542	.1922	.1019	.2176	.3040	.0420	.0344	.0537	1.0000
.2495	0.0000	.1115	.0449	.2982	.2509	.0450	0.0000	0.0000	1.0000
.2306	.0291	0.0000	.0866	.2163	.2329	.0677	.0613	.0755	1.0000
.2343	.0224	.1660	0.0000	.2134	.2334	0.0000	.1171	.0133	1.0000
.2611	.0778	.2163	.1114	0.0000	.2634	.0224	.0129	.0347	1.0000
.3040	.0545	.1941	.1015	.2195	0.0000	.0408	.0334	.0522	1.0000
.2317	.0540	.3112	0.0000	.1030	.2250	0.0000	0.0000	.0749	1.0000
.1898	0.0000	.2819	.2812	.0593	.1843	0.0000	0.0000	0.0000	.9965
.2472	0.0000	.2895	.0266	.1332	.2401	.0624	0.0000	0.0000	.9991

\*\*\*\*\*

THIS PROGRAM HAS SUCCESSFULLY RUN

\*\*\*\*\*

## REFERENCES

1. The Incorporation of Energy Conservation Principles into the Design of State Buildings, Vol. 2, UWENSOL Users' Manual, Ver. 4.1, August 1979, Built-Environment Teaching & Research Group, Institute for Environmental Studies, University of Washington, Seattle, Washington.
2. G. P. Mitalas and D. G. Stephenson, "FORTRAN IV Programs to Calculate Radiant Energy Interchange Factors," National Research Council of Canada, Computer Program No. 25 of Division of Building Research, Ottawa, 1966.

UNLIMITED RELEASE

INITIAL DISTRIBUTION

A. Emery (5)

Department of Mechanical Engineering  
University of Washington  
Seattle, WA 98195

J. H. Scott, 4700

G. E. Brandvold, 4710

B. W. Marshall, 4713

V. L. Dugan, 4720

J. V. Otts, 4721

J. F. Banas, 4722

W. P. Schimmel, 4723

J. A. Leonard, 4725

A. C. Ratzel, 4512

B. W. Marshall, 4713

R. Stromberg, 4714

R. H. Braasch, 4715

D. Hayes, 5510

J. Nunziatto, 5511

D. Mcvey, 5512

D. Larson, 5513

T. B. Cook, 8000; Attn: A. N. Blackwell, 8200

W. J. Spencer, 8100; Attn: J. F. Barham, 8110

J. L. Wirth, 8150

R. D. Cozine, 8160

W. F. Alzheimer, 8120; Attn: B. A. Benedetti, 8121

C. S. Hoyle, 8122

W. D. Zinke, 8123

R. J. Gallagher, 8124 (10)

D. Siebers, 8124

M. Abrams, 8124

J. Kraabel, 8124

D. Bell, 8124

A. Ortega, 8124

B. Meyer, 8124

P. Falcone, 8124

M. Kanouff, 8124

B. F. Murphey, 8300

R. L. Rinne, 8320

C. Melus, 8326; Attn: J. J. Iannucci

M. J. Fish

G. W. Anderson, 8330

R. J. Kee, 8331

V. Gabrielson, 8331

R. E. Huddleston, 8332

D. Hartley, 8350

B. R. Sanders, 8354

L. Gutierrez, 8400

C. S. Selvage, 8420

R. C. Wayne, 8450



A. C. Skinrood, 8452

W. G. Wilson, 8453; Attn: R. Ellis

E. Cull

C. Schafer

T. T. Bramlette, 8453

Publications Division, 8265, for TIC (2)

Publications Division, 8265/Technical Library Processes and Systems Division, 3141

Technical Library Processes and Systems Division, 3141 (2)

Library and Security Classification Division, 8266-2 (3)

