TRNSHD - A PROGRAM FOR SHADING AND INSOLATION CALCULATIONS

by

MARION D. E. HILLER

A thesis submitted in partial fulfillment
of the requirements for a degree of

MASTER OF SCIENCE
(Mechanical Engineering)

at the
UNIVERSITY OF WISCONSIN - MADISON
1996

ABSTRACT

For analyzing systems with high solar gains it is important that design tools have a reliable and accurate means of predicting the solar radiation on surfaces. Since solar radiation on a surface is often greatly influenced by self-associated facade obstructions, neighbor buildings and the surrounding landscape, a prerequisite of solar modeling is the ability to predict shaded and unshaded parts as a function of solar position and geometry. The objective of this thesis is to improve the shading and insolation calculations for simulating buildings simulations with TRNSYS.

In order to select the appropriate shading and insolation algorithms for TRNSYS, a literature search was carried out on calculation techniques and their implementation in software applications for thermal building simulation. Based on the results of the literature search a computer program called TRNSHD was developed. In this program, surfaces are projected onto a plane and appear as a series of polygons. The heart of the program is a "polygon clipper" that determines the boundary of subpolygons when two or more polygons overlap one another. For a given geometric configuration, TRNSHD performs shading calculations of external surfaces. In addition, TRNSHD can determine the distribution of beam radiation that passes through a window and onto inside surfaces of a room. The results of an analysis with TRNSHD are sunlit fractions which are defined as the ratio of the radiation on a surface including shading effects to the radiation on the surface without shading. The radiation source can either be beam radiation or diffuse sky radiation. Although TRNSHD was developed for building simulations with TRNSYS, it is a stand-alone tool that is not restricted to either buildings or TRNSYS and thus can be used to solve other shading problems.

For validation, the results of TRNSHD were compared with other algorithms. The major assumptions of TRNSHD and their effect on the results are discussed. The major assumptions concern the number of "patches" for diffuse sky radiation calculations and the frequency at which the beam radiation shading calculations are performed. The conclusion is that a total number of 72 sky patches are needed and a frequency of 24 hours (one average day) of each month is sufficient for an annual simulation. The comparative validation process showed that TRNSHD functions correctly and the results of TRNSHD are in good agreement with expected values. The external shading procedure for beam radiation has been validated against an ASHRAE algorithm for overhang and wingwall shading and against a proposed European standard. In addition, comparative calculations with another software application developed by the Danish Building Research Institute were performed. For validating the external shading procedure for diffuse radiation, results of TRNSHD have been compared to those of TRNSYS's TYPE 34 (overhang and wingwall shading) for different shading configurations. A comparison of TRNSHD with experimental measurements and computer simulations from the University of Michigan-Ann Arbor has been performed for the internal solar distribution procedure.

A complex shading study of a building with atrium has been performed by using TRNSHD. The study gives an example of the use of the program and demonstrates its capabilities.

ACKNOWLEDGMENTS

This research was supported by the Solar Energy Laboratory (SEL) at the University of Wisconsin-Madison and TRANSSOLAR Energietechnick GmbH, Stuttgart (the German distributor of TRNSYS). The financial support from TRANSSOLAR and SEL is gratefully acknowledged. Special thanks go to Professor William A. Beckman and Matthias Schuler for offering me the opportunity to do this research, which has been a very challenging, enjoyable and rewarding experience.

None of this would have been possible without the skillful guidance of my advisors Professor William A. Beckman and John W. Mitchell: Thank you for your confidence, your criticism and your patience. Thanks to all Lab students (including Nate, our TRNSYS wiz) for their support in sometimes desperate battles against computers and printers and for the fun time we had together. Special thanks go to Shirley Quamme who was a tremendous help in negotiating bureaucratic jungle.

I also would like to thank Kjeld Johnsen from the Danish Building Research Institute and Fred Winkelmann from the Lawrence Berkeley Laboratory for immediately answering my requests and the volumes of detailed information.

Finally, thank you Heiko for all your love and encouragement through the ups and downs of this project.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A	Area
b	Conduction transfer function coefficient
c	Conduction transfer function coefficient
d	Conduction transfer function coefficient
E	Equation of time
f	Sunlit fraction
G	Irradiance
H	Height
h	Coefficient of heat transfer by long-wave radiation and convection
I	Hourly solar radiation incident on surface
L	Length
q	Energy per unit time and unit area
R	Difference between long-wave radiation incident on the surface from sky and surroundings and radiation emitted by blackbody at outdoor air temperature
SHFT	Difference between standard meridian for local time zone and the local longitude
T	Temperature
t	Time
W	Width
X	x coordinate
Y	y coordinate
Z	z coordinate

Greek

α	Angle (defined locally), absorptance of solar radiation
γ_s	Solar azimuth angle
δ	Declination, interval
φ	Rotation angle
θ_z	Solar zenith angle
ϕ	Latitude
ω	Hour angle, solid angle

Subscripts

beam	Beam
dfu	Diffuse
ex	External
in	Internal
N	North
P	Projected
o	Outside
plane	Plane of the current receiving wall
rc	constant indoor room temperature
sun	Sun direction
sunlit	Sunlit
total	Total
unshaded	completely sunlit
z	Zenith

CHAPTER
ONE

INTRODUCTION

Objective

The energy consumption of U.S. buildings accounts 36% of the country's energy supply at a cost of $193 billion and the energy use is growing at a rate of 3.3% a year. Heating and cooling equipment consumes 42% of all building energy use at a cost of $81 billion. Buildings that incorporate a passive solar design can save as much as 50% on heating bills for only 1% more construction cost. [eren, 1996]

These facts illustrate the capability and the importance of passive solar building design with respect to conservation of both energy and money. The idea of passive solar is simple: take advantage of natural processes like radiation, conduction and convection and avoid them, when undesired. But applying passive solar effectively does require information and attention to the details of design and construction. Nowadays, design tools are available to analyze the thermal behavior of buildings in detail and give recommendations for design strategies. It is therefore necessary, that design tools have a reliable and accurate means of predicting the solar radiation on surfaces. Since solar radiation on a surface is often greatly influenced by self-associated facade obstructions, neighbor buildings and the surrounding landscape, a prerequisite of solar modeling is the ability to predict shaded and insolated parts as a function of solar position and geometry.

The objective of this thesis is to improve the shading and insolation calculations for building simulations with TRNSYS. Due to its intensive development in the last years, the simulation program TRNSYS has become an attractive tool for the thermal analysis of buildings. A new graphical interface called TRNSCAD offers a convenient way to describe the building geometry in three dimensions. This new input device allows advanced methods to be incorporated for calculating the external and internal shading of surfaces, because now the complete three dimensional geometric information is available. The new shading model is able to handle planar polygons without holes as radiation receiving and shadow casting surface. All external building surfaces and additional defined surfaces are potential shadow casting surfaces. The result of the analysis is the sunlit fraction of external building surfaces for beam and diffuse radiation. The sunlit fraction is defined as the ratio of radiation on a surface including shading effects to the radiation on the surface without shading. The sunlit fraction for beam radiation is a function of the solar position.

In addition to the shading calculations, the program is also able to determine insolation of internal surfaces, especially internal windows, that results from direct solar penetration through windows.

Introduction to TRNSYS for Building Simulation

The simulation program, for which the shading and insolation calculations are improved, is called TRNSYS. The following section gives a brief survey of TRNSYS with emphasis on the simulation of buildings.

TRNSYS is a transient system simulation program with a modular structure [SEL, 1995]. The modular structure of TRNSYS gives the program tremendous flexibility, and facilitates the addition to the program of mathematical models not including in the standard library. In TRNSYS, a system is described in a special input language that connects components together. The file containing the system description is called DECK. External data files, i.e. weather data, can be assigned to the simulation in the DECK. Each component of a system is associated with a TYPE number, which identifies the components function (i.e. TYPE 16 is the solar radiation processor), and a UNIT number which distinguishes the component from all other components in a system. A component receives three types of information: INPUTS, PARAMETERS and TIME. INPUT variables may vary with time and are generally OUTPUTS from other components. PARAMETERS are assumed to not vary throughout the whole simulation. Time is a TRNSYS internal variable and is neither an INPUT nor a PARAMETER. The information flowing out of a component are called OUTPUTs. For creating a DECK two different graphical interfaces, PRESIM and IISIBAT, can be used for assistance.

TRNSYS provides different methods to simulate the thermal behavior of a building. The first one is a very simple Degree-Day model (TYPE 12). A more detailed method is offered by a one zone model (TYPE 19). For modeling the thermal behavior of a building having up to 25 thermal zones the multiple-zone building model (TYPE 56) can be used.

The heat flows in TYPE 56 is based on energy balances. In order to reduce computing time the walls of a building are modeled according to the transfer function relationships of Mitalas and Arseneault [Mitalas and Arseneault, 1967] defined from surface to surface. Long wave radiation and convective heat transfer are calculated using the star network approach developed by Seem [Seem, 1987]. Moisture balances are performed in addition to the energy balances. TYPE 56 represents a non geometrical building model, because the only required geometric information are the area of each surface, the relationship of a surface to the thermal zones, the orientation of external surfaces and the volume and capacity of a thermal zone. The number of required INPUTs for TYPE 56 strongly depends on the building. In general, the following INPUTs are needed:

- ambient temperature
- ambient humidity ratio
- incident radiation for each existing orientation of external surfaces
- incident beam radiation for each existing orientation of external surfaces
- incident angle for each existing orientation of external surfaces
- other specific inputs, i.e. for control

Type 56 provides a large list of optional OUTPUTs for each thermal zone, like air temperature, ventilation, infiltration, convective and radiative gains, humidity ratios etc. The required PARAMETERs concern the building description.

Due to the complexity of a building description with multiple zones, a separate preprocessing is performed first to create 'PARAMETER' files which are used by TYPE 56 during the simulation as shown in Figure 1.1. A 3D graphical interface called TRNSCAD can be used to create or only digitalize the geometry of a building. TRNSCAD writes an output file, which is further processed by PREBID, an interactive input program. In PREBID the building description is completed by specifying the characteristics of the building walls and windows and data like ventilation, infiltration, gains, heating mode and cooling mode. Also, the desired outputs are specified within PREBID. Then, the output file of PREBID is processed by the program BID. BID generates three files: one file containing the transfer functions for the walls, a second file containing the remaining building description and a third file for user information only. The first two output files from BID are used as parameters by TYPE 56 in the actual TRNSYS DECK.

Up to now the blocking of solar radiation by other surfaces is performed by TYPE 34, overhang and wingwall shading. In TYPE 34, the shading of beam, diffuse and ground reflected radiation of vertical rectangular receiver surface due to rectangular overhang and wingwalls is determined. Therefore, for each radiation source a multiplier for the irradiance is computed representing the blocking effect. For the beam radiation shading, TYPE 34 determines an irradiated fraction, which is defined as the sunlit area taking into account blocking effects over the total area of the receiver surface. The implemented algorithm is based on a work by Sun [ASHRAE, 1975] and applies analytical geometrical principles in order to determine the sunlit area.

Figure 0.1 Preprocessing for building simulation with TRNSYS

The multipliers for diffuse sky and ground reflected radiation are equal to the view factors between the receiver surface and the sky and the ground, respectively, assuming an isotropic distribution of the radiation. For calculating these view factors, TYPE 34 takes the view factor of a vertical unshaded surface to the sky and ground, respectively, and subtracts the view factor of the receiving surface to the overhang and wingwalls. The view factor between the receiver surface and the wingwall is computed by integrating the differential surface area radiation view factor of the wingwall over the receiver surface area. The receiver radiation view factor of the overhang is computed in a similar manner.

The disadvantages of this approach of shading modeling are its restriction to a special geometrical configuration of surfaces and the large effort needed to specify the geometric information, especially when many surfaces might be shaded. Due to this restriction, the shading of tilted surfaces cannot be determined. Also, shadows from opposite or tilted surfaces, building parts and the surrounding landscape cannot be represented.

For calculating the insolation on inside surfaces, TYPE 56 uses constant absorptance weighted area ratios. This simple method works fine, if all internal walls are opaque. However, when internal windows are also considered, the penetration of direct solar radiation from one building zone to another can not be modeled. The capability of modeling such effects is especially important for buildings with high solar gains and internal glazing, like sunspaces and atria.

CHAPTER TWO

LITERATURE SEARCH

In order to select the appropriate shading and insolation algorithms for TRNSYS, a literature search was carried out on shading and insolation calculation techniques and their implementation in software application for thermal building simulation. A paper from Gueymard [Gueymard, 1993] provides a good overview of algorithms used in thermal building simulation. Additionally, literature references are given in ASHRAE RP-741 [Spitler, 1994]. For investigating the details of a particular program, the program manuals were used.

Shading Calculation Techniques

In general, shading calculations are performed to determine the solar radiation on an object taking into account the blocking effects of other objects. The object, for which the solar access is determined, is called the "receiving" object. The objects, which cause the blocking, are called "shadow casting" objects or simply "shader" objects. In order to perform the calculations, the geometry of the receiving and the shadow casting objects as well as their location with respect to each other must be defined. Most shading algorithms consider planar polygons with linear edges as shadow casting and receiving surfaces. The polygons might be restricted to certain shapes depending on the applied method. Three dimensional solids and curved surfaces or edges are approximated by polygons. Additionally, the location of the radiation source with respect to both the receiving and shadow casting objects must be known. Also, the source of radiation, i.e. beam, diffuse or reflected radiation, and sometimes the intensity needs to be specified. The solar access to the receiving object is then obtained by applying analytical geometric principles (i.e. coordinate transformation, projection) and geometric computational methods (i.e. polygon intersection). The computed results depend on the algorithm, and can be either the sunlit area, the shaded area, the incident energy including the effect of shadow casting surfaces, or the ratio of either quantity to the value obtained without shading. In this work, the latter ratio is simply called shaded and sunlit fraction, respectively. The above mentioned results can be used to determine the average radiation level over the whole receiving object from the unshaded radiation level of the receiving object. Additionally, some algorithms are able to represent the shape of the remaining sunlit parts of the receiving polygon. This is especially of interest if the remaining sunlit parts of the receiving polygon are needed for further calculations, i.e. of a window for determining the internal insolation on surfaces. The complexity of shading calculations mainly depends on the considered geometric configuration of the receiving and shadow casting objects and the radiation source.

The state of shading calculation methods of different categories are presented in the following paragraphs. The existing shading algorithms are divided into three categories according to the different sources of radiation:

- shading of beam radiation,
- shading of diffuse radiation and
- shading of reflected radiation.

The literature shows that these three categories may be subdivided into algorithms that handle special situations only, such as overhangs, wing walls, etc. and algorithms with a general approach, where the shape and location of a shadow casting surface with respect to the receiving surface is less restricted. For a thermal building analysis program like TRNSYS, the method should be general, because it is necessary to consider all kind of shadow casting objects like building parts, external obstructions and the surrounding landscape and the calculations have to be performed for each external wall and window of the building. Thus, the literature search focused only on general methods. The methods reviewed are discussed with respect to the following criteria:

- characteristics of shadow casting and receiving objects,
- representation of radiation source,
- available output,
- accuracy and
- speed.

Beam Radiation Shading

The most important category of shading algorithms concerns the beam radiation, because this greatly affects the solar load of buildings. Beam radiation is the solar radiation received from the sun without having been scattered by the atmosphere. For determining the sun's position for beam radiation, standard models are used that take into account the geographic location of the building, i.e. latitude and longitude, the local time and the time zone (see i.e. [Duffie and Beckman, 1991]). Calculation techniques with different approaches are available for determining the shaded and sunlit parts of a receiving surface. For simple shading shapes, where no multiple overlap can occur, the calculations are relatively straightforward. However, general methods need a more sophisticated method that includes complex multiple overlaps and more general surface shapes and locations. The major general methods are presented briefly.

Discrete Element Analysis with Grids

Groth and Lokmanhekim [Groth and Lokmanhekim, 1969] proposed an algorithm with discrete element analysis of grids (see Figure 2.1). This approach is used in NBLSD and NECAP, precursor of DOE-2 and BLAST. A discretization of the receiving surface in two dimensional elements is also used in ESP-r [Clarke et al., 1993]. The discrete element analysis allows a representation of the shape of the remaining receiving polygon. The consideration of semi-transparent shadow casting surfaces is possible. The simplicity of this method is advantageous, but the discrete element analysis is not accurate and is time consuming, especially for higher resolution and many surfaces.

The following steps are performed by the method of Groth and Lokmanhekim. First, the problem is reduced from three dimensional space to a plane by coordinate transformation and shadow projection on the receiving surface. In order to avoid false shadows, the part of the projected shading surface that is below the apparent horizon of the receiving surface is removed. The receiving surface is divided into two dimensional grid elements, each represented by its center point. Each point is tested for enclosure by the shadow projections to determine the sunlit and shaded area. The sum over the sunlit or shade grid elements is used to obtain the sunlit and shaded fractions, respectively.

Figure 0.1 Discrete element analysis with grids

Discrete Element Analysis with bars

The bar-method [LBL, 1982], which is currently used in DOE-2, is an improvement of the method of Groth and Lokmanhekim. The grid technique of the latter is replaced by a bar method, increasing both speed and accuracy. However, the speed is still depending on the desired accuracy. The capability of representing the shape of the remaining receiving surface and the consideration of transparent shaders are maintained.

Figure 0.2 Discrete element analysis with bars

The bar-polygon technique is described briefly. The receiving surface is enclosed in a rectangle. The rectangle is divided into a large number of bars or strips with equal width. The shape of each shadow polygon on the receiving surface is approximated by calculating and storing the points where the shadow polygon crosses the midline of each bar as shown in Figure 2.2. Thus each shadow polygon is approximated by a group of bar segments. In addition to the 'crossing' points, which define the bar segments, the relative intensity of the direct radiation reaching each bar segment is stored. The intensity of radiation over each bar segment is assumed to be constant. The amount of direct radiation on the receiving polygon, and therewith a shaded fraction, is obtained by using the bar segments and the relative intensities of direct radiation.

Convex Polygon Clipping with Homogeneous Coordinates

The method presented by Walton [Walton, 1979], which is currently used in BLAST, also improves the method of Groth and Lokmanhekim. The algorithm uses the same coordinate transformation and projection techniques, but replaces the discrete element analysis with a method using homogeneous coordinates. In homogeneous coordinates an n dimensional space is represented by n+1 dimensions, i.e. two dimensional data where the position of a point is given by the duple (x,y) is represented by three coordinates (hx,hy,h), where h is an arbitrary number [Rogers, 1976]. In Walton's method, the area of the sunlit parts of the receiving polygon are determined indirectly, but exactly, by evaluating the boundary and the area of overlaps of shading surfaces and the receiving surface and shading surfaces among each other. The resulting boundary of the sunlit or shaded parts of the receiving polygon itself cannot be determined. Also, compared to the method of Groth and Lokmanhekim, there is a change in the way this method considers surfaces. The surfaces are considered to be convex polygons and are represented by an ordered list of its vertices. Additionally, only surfaces of a solid object that are not illuminated are considered shadow casting surfaces. In general, this reduces the number of possible shadow casting surfaces, because a building is standing on the ground and the floor surface is not consider a shadow casting surface.

The algorithm proposed by Walton is exact. However, it is restricted to the use of convex polygons and the resulting shaded or sunlit surface parts cannot be presented in the same way as the original receiving and shading polygons. According to the author, this algorithm is fast, but some methods which sacrifice accuracy such as grid techniques with very few point, could be faster. Transparent shadow casting surfaces can be incorporated within this method.

Figure 0.3 Convex polygon clipping with homogeneous coordinates

The following describes Walton's method using homogeneous coordinates. The algorithm is based on the calculation of the overlap of two polygons as shown Figure 2.3. First, the cartesian coordinates are transformed into homogeneous coordinates. The advantage of homogeneous coordinates with respect to cartesian is that the calculation of the intersecting points of lines and of the location of a point versus a line is much easier. After the transformation in homogeneous coordinates, the algorithm finds the vertices of one polygon within the other and visa versa. Afterwards, the intersecting points of the boundary of both polygons are determined. The result of these two steps are all vertices of the overlap. The vertices are then retransformed into cartesian coordinates, ordered clockwise and the area is computed. Due to the restriction of convex polygons, every overlap found for a shading polygon needs to be checked for multiple shadow overlaps with every other encountered overlap.

Boundary Evaluation with 2D Concave Polygon Clipping

The boundary evaluation method is based on a technique which was developed in context of hidden surface removal for applications in computer graphics by Weiler and Atherton [Weiler and Atherton, 1977]. The method was adapted by S.A. Johnston [Johnston, 1985] in a subroutine called SOLTST for SOLAR GRAPHICS and by K. Grau and K. Johnsen [Grau and Johnsen, 1995] in a program called XSUN. Like in the previous methods, the polygons are represented by an ordered list of its vertices. In order to apply the boundary evaluation method the three dimensional problem is first reduced to a two dimensional one by coordinate transformation and projection. The sunlit and shaded parts of the receiving polygon are found by a boundary traversal and are represented the same way as the original receiving polygon, by an ordered list of their vertices.

This quite complex, but powerful method allows accurate calculations for all kinds of overlap of planar polygons in two dimensions. Another advantage of this algorithm is that the remaining sunlit or shaded polygons are represented the same way as the original receiving polygon. Therefore, the output polygons can be directly used for further calculations. No information was found in literature concerning the computing time. But, by looking how the algorithm works, the algorithm appears to be faster than Walton's method because the checking for multiple overlap requires less computational effort. The consideration of transparent shaders is possible and is implemented in the program Xsun [Grau and Johnsen, 1995]. Because of these advantages, the method of Weiler and Atherton was chosen as the basic algorithm in the new shading program.

Polygon clipping is described here briefly. A more detailed description is presented in Section 3.2.7. As mentioned before the receiving and shading polygons are presented by an ordered list of their vertices. The way the polygons are ordered determines whether the resulting polygons represent the overlapping or the non-overlapping part of the receiving polygon. The first step in the algorithm is to determine the intersecting points of both polygons. The encountered intersections are classified into entering intersections, where the shadow casting polygon enters the receiving polygon, and leaving intersections, where the shadow casting polygon leaves the receiving polygon. Afterwards, the intersections are stored in two lists. The first list holds the intersections in the order they occur on the shadow casting polygon and the second list holds the intersections in the order they occur on the receiving polygon. The basic boundary traversal starts at the first entering intersection on the shading polygon and follows the boundary of the shader until the next intersection is found. At the second intersection the algorithm switches to the receiving polygon and follows its boundary until the next intersection is found. Again the algorithm switches, this time to the shading polygon and the shader boundary is then followed. The boundary traversal is continued until the starting point is reached as shown in Figure 2.4. The clipping process is not finished until all intersections have been stored during the boundary traversal. If some intersections have not been stored, then more than one remaining sunlit polygon exists and a new boundary traversal starts at the first unstored intersection on the shader polygon.

Figure 0.4 Boundary evaluation with 2D concave polygon clipping

Diffuse Radiation Shading

The solar radiation received from the sun after its direction has been changed by scattering by the atmosphere is called diffuse radiation. Many thermal building simulation programs neglect the blocking of diffuse radiation on external surfaces by obstructions because it is generally a secondary effect on the total incident radiation. However, in certain cases the blocking effect may be of importance.

The common way to take the diffuse shading effect into account is to assume an isotropic source and to calculate view factors for the receiving surface to the sky. For special configuration cases like overhangs and wing walls there is a straightforward approach to calculating the view factors from known formulas or by numerical integration.

A more general approach is used in DOE-2 [Winkelmann, 1996]. The sky is still assumed as an isotropic source, but the configuration of shadow casting and receiving objects is generalized. The sky, represented by a half hemisphere, is discretizised into patches as shown in Figure 2.5. Each patch is assumed to have the same radiance. The shading calculation for beam radiation is used to determine how much radiation from a given patch is blocked by shadow casting surfaces. In this calculation the diffuse sky radiation is assumed to originate at the center point of the patch. The sum of irradiance from all patches gives the sky irradiance on a surface. The output of the algorithm is the ratio of the diffuse sky radiation on a surface including shading effects over a diffuse sky radiation on a surface without shading. This approach is adopted in the new shading program for TRNSYS.

Figure 0.5 Discretization of the sky for diffuse shading calculations

Reflected Radiation Shading

Reflected radiation is generated by objects reflecting beam and diffuse radiation. For discussing reflected radiation the differentiation between purely diffuse reflection, purely specular reflection and a combination of both must be made. If the reflection process is purely diffuse and isotropic, the problem can be solved with view factors as described in the previous section. Some programs, like DOE-2, assume the ground reflection to be purely diffuse and isotropic. However, the problem is complicated when specular reflections are involved. Algorithms handling such situations have not been integrated into building simulation tools. A current ASHRAE research project is addressing the issue. The resulting algorithms will be implemented in DOE-2.

Integration of Shading Calculations

The second part of the literature search concerned the integration of shading algorithms into building simulation programs. Shading calculations can be very time and memory consuming, depending on the complexity of the shading geometry's and the frequency at which the calculations are performed. Therefore, in most software applications certain assumptions are made to retain reasonable accuracy and to reduce both time and memory requirements. The integration of shading calculations in simulation tools is investigated for the following three programs: BLAST, DOE-2 and ESP-r. These three programs are widely used for analyzing the thermal behavior of a multiple-zone building. Each of these programs applies a different shading calculation technique, as discussed above. DOE-2 is the only program that performs a diffuse shading calculation. The following subjects of the integration are discussed in more detail: the considered receiving and shadow casting objects, preliminary tests and frequency and placement of the shading calculations.

Considered receiving and shadow casting objects

Due to the fact, that shading calculations can be very time consuming, the three programs offer different levels of shading calculations. The user may specify two categories of shading objects: attached shading devices, like overhangs and wingwalls, which are restricted to a certain group of receiving surfaces, and detached shading devices, like neighbor buildings, which may shade several external surfaces of building. The receiving and shadow casting objects and their characteristics for the three simulation programs are shown in table 2.1.

The main difference between the programs is the level of required user input. In ESP-r, the user needs to select the receiving as well as the shadow casting objects. The receiving objects are determined automatically in DOE-2, but external walls of the building are not specified automatically as possible shadow casting surfaces. The user has to define them. BLAST handles surfaces of a building zone that cast shadows onto other surfaces automatically [BLAST, 1991]. The shape of the receiving and shadow casting objects is restricted most in DOE-2. Since only rectangular surfaces are allowed. BLAST requires all surfaces to be convex. ESP-r restricts the shape of shadow casting objects to box-shaped solids. Both BLAST and DOE-2 allow transparent objects for detached shadow casting surfaces and also the level of transparency to vary it over time.

Table 0.1 Considered receiving and shadow casting objects

	BLAST	DOE-2	ESP-r
receiving objects			
shape	convex, planar, simply connected polygons	rectangular, planar, polygons	planar, non selfintersecting, simple connected polygons
for attached shader	extr. wall and all windows and doors on that wall	extr. wall and all windows and doors on that wall	window to which attached
for detached shader	all external surfaces	all external surfaces	user specified surfaces
shadow casting objects			
attached	overhang, wingwalls	overhang, wingwalls	overhang, wingwalls
detached	convex, single sided, planar, simply connected polygons	rectangular, planar, polygons	box-shaped solids
building surface	automatically specified as detached device	needs to be specified as detached device	needs to be specified as detached device
transparency	for detached devices (time dependency possible)	for detached devices (time dependency possible)	opaque only

Preliminary tests

In general, simulation programs perform preliminary tests before executing the actual shading calculations. The objective of these tests is to minimize the unnecessary execution of shading routines and to avoid false shading calculations. One mandatory test is a

sunlit/ shaded test. This test determines whether the considered receiving surface is sunlit or not. Another mandatory test is a front/behind test and its objective is to examine whether a shadow casting surface lies completely or partially behind the receiving surface or not. The projection of such shadow casting surfaces may causes false shadows as shown in Figure 2.6. If a shadow casting surface lies completely behind the receiving surface, no shading can occur and thus shading calculations are not necessary. But for shadow casting surfaces which lie partially behind the receiving surface, a correction needs to be performed. A common method, which is used in BLAST and DOE-2, is to clip off the parts of the surface that lie behind. In order to improve the automatic determination of shadow casting building surfaces, BLAST considers only the unlit surfaces of a building as potential shadow casting surfaces .

Figure 0.6 False shadow projection of a shadow casting surface

Frequency and placement of shading calculations

In ESP-r the beam shading calculations are performed for each hour of one day of each month. The chosen day corresponds to the condition of average solar declination of that month. Thus, the results of the shading calculations for the same hour of each day are assumed constant over the month. The restriction of the shading calculations is unlikely to have a significant impact on simulation predictions and is necessary to restrict the computing time and file size [Clarke et al., 1993]. The shading calculations are performed before the time loop of the actual building simulation. The results are stored in a file for each month. One advantage of this technique is that the shading results do not have to be determined during every simulation, but they can be reused for different simulation runs with the same geometry. Another advantage is that at later date additional shading results, for other surfaces or months, can be added to the file.

Like ESP-r, DOE-2 performs the beam radiation shading only for one day of each month. However, the chosen day is simply the first day of the month. The calculation is integrated in the hourly loop of the building simulation. The diffuse shading is computed only once, before the hourly loop of the actual building simulation.

BLAST performs the shading and insolation calculations in a preprocessing. Unfortunately, no information was available how often such calculations are performed.

Inside Surface Insolation

Solar radiation is transmitted into a zone through windows and strikes inside surfaces. The distribution of solar radiation depends on the geometry of the zone, the varying sun position and the varying fraction of beam, diffuse and reflected radiation. Many thermal building simulation programs neglect the dynamic character of the inside surface insolation and use constant distribution factors because the computation is time and memory consuming and the increased accuracy is in many cases low. However, in some situations the loads of a zone will be affected greatly by the distribution of solar energy. This is especially true for buildings with high passive solar gains. The proper placement of thermal mass requires an accurate calculation of inside sunlit surfaces. Also, the solar distribution on the inside surfaces of a building zone becomes important when inside surfaces are composed of opaque and transparent portions, with the latter allowing the onward transmission of solar radiation to a connected zone.

The approach for calculating the internal surface insolation is similar to the shading calculation and the same basic calculation techniques of shading determination can be applied. The basic steps are the following.

- Project each inside surface on the window or project the window on each internal surface
- Determine the sunlit area and fraction

For a non-convex building zone, one inside surface can block another one. In order to incorporate those blocking effects, the inside surfaces must be processed in a certain order or ”inside” obstructions must be defined. The correct order can be determined by a preliminary depth sort which is not a trivial approach in three dimensions.

The integration of insolation algorithms in building simulation tools is similar to that previously discussed for shading calculations. Due to the time and memory consumption of the calculations, many simulation programs offer these calculations only as an option. In DOE-2, the internal solar distribution calculations are restricted to zones defined as ”SUNSPACES”. The blocking effect of inside surfaces is handled by defining inside obstructions. BLAST offers the internal solar distribution optional for every ordinary zone. In ESP-r, these calculations are only performed for selected external windows.

Conclusions

The literature search showed different ways to perform external shading and internal solar distribution calculations. Coordinate transformation and projection are common procedures to reduce the three dimensional problem to two dimensions. The similarities between both external shading and inside insolation calculations for beam radiation permit the use of the same basic algorithm to solve the remaining two dimensional problem. The Weiler-Atherton method seems to be superior to the other discussed methods with respect to the combination of speed, accuracy and generality. However, this method is quite complex with its rules for the boundary traversal. For diffuse radiation shading, the approach implemented in DOE-2 has the merit of simplicity and generality. Another advantage of this approach is that the same basic algorithm used for beam radiation shading can be used.

Shading/insolation calculations can be very time consuming. Certain assumptions must be made in software applications in order to retain reasonable accuracy and reduce both time and memory requirements. It is a common approach to perform the shading/insolation calculations only for one day of each month. Further, time and memory savings are achieved by minimizing the number of potential shadow casting and receiving surfaces of a building by pretesting and selection. For example, the internal solar distribution calculations can be restricted to certain zones of interest or only shaded building surfaces are considered as potential shadow casting surfaces. The integration of shading/insolation calculations in building simulation tools as a preprocessing operation offers the possibility to reuse previously calculated results. It saves therewith simulation time, especially during parameter studies with the same geometry.

CHAPTER
THREE

THE PROGRAM TRNSHD

In simple terms, the program TRNSHD (pronounced TRANSHADE) performs external shading and internal insolation calculations for a given geometric configuration. TRNSHD is a stand-alone tool developed for building simulations with TRNSYS. However, TRNSHD is not restricted to buildings or TRNSYS and can be used to solve other shading problems. The results of an analysis with TRNSHD are the sunlit fractions of external building walls and their attached windows as a function of radiation source and shadows originated by external building walls and defined obstructions. The sunlit fraction is defined as the ratio of radiation on a surface including shading effects to the radiation on the surface without shading. The radiation source can be either beam radiation or diffuse sky radiation. Additionally, TRNSHD can determine the distribution of beam radiation that passes through a window and onto inside surfaces of a room. The program is described in the following sections.

Input and Output of TRNSHD

The structure of the input and output of TRNSHD is designed with respect to building simulations with TRNSYS. TRNSHD requires an input file containing the geometric information of the building and its obstructions. In addition, TRNSHD allows the user to select the desired calculation level in an interactive way. The output of TRNSHD, sunlit fractions, are written to output files. Example input and output files are in Appendix A.

Figure 3.1 shows the data structure of the required geometric data. The data can be divided into two parts, a combinatorial and a metric part. Only the metric part varies for different coordinate systems, whereas the combinatorial part is independent of the coordinate system.

The building is composed of modules, which are three dimensional polyhedral solids. The modules can be disjoint or touching each other at a common vertex, edge or face, but a module is not allow to penetrate another one. The faces of the module are external and "internal" walls. An internal wall is a wall that is shared by two modules and thus lies within the building. A direction vector is associated with each wall that represents the outwarding normal vector of the wall with respect to the module. Therefore, an internal wall must be defined twice, once in each module. In order to associate the direction vectors with cardinal points, a north vector for the whole system must be defined additionally. Windows can be defined on both types of walls. Windows are not allow to penetrate each other and they must lie completely within the wall‘s boundary. External windows posses a link to the module to which they belong. This link is used for the internal solar distribution of beam radiation. If the link is not provided, then no internal solar distribution of beam radiation is performed for that window. Obstructions are considered as two dimensional planar polygons with a zero transparency.

Figure 0.1 Input data structure of TRNSHD

All walls, windows, modules and obstructions are identified unambiguously by an associated identification number. External and internal walls are not allow to posses the same identification number. Similar, external and internal windows are not allow to posses the same identification number. The identification numbers are positive integer values equal to or bigger than one. They do not have to be successive.

All walls, windows and obstructions are considered as planar non-selfintersecting polygons without holes. The edges of a polygon are restricted to straight lines. The boundary of the polygon is represented by an ordered closed list of its vertices. The order can be clockwise or counter clockwise. A ”closed” list means that the last vertex must be the same as the first vertex.

All metric information must be entered in the same three dimensional coordinate system, the world coordinate system. The floors of the building and the north vector must lie in planes parallel to the xy plane of the world coordinate system. Then, the z coordinate represents the height.

In addition to the geometric input via file, TRNSHD requires information about the desired calculation mode. This information is entered in an interactive way. The advantage of separating both kinds of input is that the geometrical input does not need to be changed for different levels of shading calculations. With regard to the computation time, the lowest acceptable level of detail should be selected. The following calculation modes are available:

- beam radiation shading calculation of external surfaces on/off and
- diffuse radiation shading calculations of external surfaces on/off.

If the calculations of beam radiation shading are turned on, then TRNSHD needs information about the location of the building in order to determine the sun's position correctly. The following input is required:

- Latitude of the location in question,
- a switch to indicate that the program should treat the local time as solar time or
- the difference between the standard meridian for the local time zone and the longitude of the location in question (called SHFT) and
- a switch to turn on/off the internal insolation distribution of beam radiation.

The output of TRNSHD are sunlit fractions, which are written to output files. The program creates two files for external radiation shading, one for external walls and another one for external windows. These files contain one sunlit fraction for diffuse radiation and, if beam radiation shading was selected, 288 sunlit fractions for beam radiation. The 288 values for beam radiation are calculated on an hourly base for twelve typical days throughout the year. The file also includes the hour of year when each hourly sunlit fraction was computed. For internal solar distribution a file is created for each external window. The file contains 288 sunlit fractions for beam radiation for each internal wall and window of a module.

Operation of TRNSHD

The structure and operation of TRNSHD are shown in a simplified block diagram in Figure 3.2. In order to keep the block diagram easy to understood, only the main iteration loops are shown. Each block represents a distinct operation or a closely related sequence of operations.

After reading the input, TRNSHD creates sun vectors for beam and diffuse radiation. These sun vectors are stored in temporary files. Then, the program selects each external wall in turn as the receiving wall for the shading calculations. The shading calculations begin with a coordinate transformation of all vertices to a more convenient coordinate system. Then, the quadrant test eliminates shadow casting surfaces which lie behind the current receiving wall. The remaining shadow casting surfaces are classified into four categories according to their location with respect to the current receiving wall. TRNSHD considers all external walls and obstructions as potential shadow casting surfaces. Now, the program sets up loops over all solar positions. Within the loop, the program checks first, whether the current receiving wall is sunlit or not. Afterwards, TRNSHD projects the shadow casting surfaces on the plane of the current receiving wall according to the sun's ray. Thereby, the three dimensional problem is reduced to two dimensions. A clipping procedure determines the sunlit parts of the current receiving wall according to a method proposed by [Weiler and Atherton, 1977]. Subsequent to the wall clipping, the program selects each attached window of the wall in turn, performs the clipping process and determines the sunlit fractions. If the internal solar distribution is turned on, then TRNSHD computes the sunlit fractions of the window that strikes each internal surface of the module. After finishing the attached window loop, the sunlit fractions of the receiving wall and its attached windows are computed for diffuse radiation. Before the next external wall is selected as receiving wall, all calculated sunlit fractions are written to output files. The following sections describe the operation performed during each block in more detail.

Figure 0.2 Simplified block diagram of TRNSHD

Sun Vector for Beam Radiation

The solar position is determined for each hour of one day of each month. The chosen day corresponds to the average day of that month according to [Klein, 1977]. For each hour the average solar position is used. The coordinates of the sun vector are stored in a temporary file.

The position of the sun in the sky is specified by two solar angles, the solar zenith angle and the solar azimuth angle. The zenith angle is the angle between the vertical and the line to the sun. The solar azimuth angle is the angle between the local meridian and the projection of the line of sight of the sun onto the horizontal plane. Zero solar azimuth is facing the equator, west is positive, while east is negative. For determining both solar angles TYPE 16 (solar radiation processor) of TRNSYS [SEL, 1995] was modified and implemented as a subroutine. Thereby, the solar angles are found using the following trigonometric relationship:

$$\cos\theta_{Z} = \sin\delta\sin\phi + \sin\phi\cos\delta\cos\omega \tag{0.1}$$

$$\sin\gamma_{S} = \frac{\cos\delta\sin\omega}{\sin\theta_{Z}} \tag{0.2}$$

where
θ_Z = solar zenith angle,
γ_S = solar azimuth,
δ = declination,
ϕ = local latitude,
ω = hour angle.
In order to use the sun-angle relationship above, the local time is converted to solar time:

$$\mathrm{solartime} = \mathrm{t} + \mathrm{E} + \mathrm{SHFT}/15 \tag{0.3}$$

where t is the local time in hours and SHFT is the difference between the standard meridian for the local time zone and the longitude of the location in question. E in Equation 3.3 is called the equation of time and accounts for the eccentricity of the earth's orbit (see chapter 2 [Duffie and Beckman, 1991]).

The coordinates of the sun direction vector in the world coordinate system are obtained from the solar angles and the angle α_N between the x-axis of the worlds coordinate system and the given north vector:

$$\begin{bmatrix} \mathrm{X}_{\mathrm{sun}} \\ \mathrm{Y}_{\mathrm{sun}} \\ \mathrm{Z}_{\mathrm{sun}} \end{bmatrix} = \begin{bmatrix} \cos(\alpha_{N} - \gamma_{S})\cdot\sqrt{1-(\cos\theta_{Z})^{2}} \\ \cos(\alpha_{N} - \gamma_{S} - \mathrm{pi}/2)\cdot\sqrt{1-(\cos\theta_{Z})^{2}} \\ -\cos\theta_{Z} \end{bmatrix} \tag{0.4}$$

Figure 3.3 shows a sun vector in the world coordinate system. In the solar angle calculation, the zenith angle has to be equal to or smaller than 90 degrees in order to assure that the sun is above the horizon. If the sun is below the horizon, then the zenith angle is set arbitrarily to 99 degrees and the sun vector is set equal to the zero vector. No shading and insolation calculations are performed in this case.

Figure 0.3 Sun vector in world coordinate system

Sun Vector for Diffuse Radiation

For the diffuse radiation shading, the sky is represented by a half hemisphere, where the building is placed in its center (see Figure 2.5). This half hemisphere is subdivided into 12 x 6 patches. The increment angle for the azimuth angle is set to 30°; the increment angle of the zenith angle to 15°. It is assumed that the patches are rather small and far away, thus the diffuse radiation leaving each patch can be treated as parallel radiation with the direction from its center point to the center of the hemisphere. Similar to the sun's position for beam radiation, a "sun's position" for diffuse radiation can then be defined as the center point of a patch and described by the same two solar angles, the solar zenith and azimuth angle. A routine creates the center points of all patches by defining the solar zenith and azimuth angle and calculates then the sun vector according to Equation 3.4. Finally, the coordinates of all diffuse sun vectors are stored in a temporary file.

Coordinate Transformation

For each external wall, which is considered as the current receiving wall, TRNSHD transforms the world coordinate system to a more convenient one, where the positive direction of the x-axis is equal to the direction of the outwarding normal vector of the wall and thus the wall is lying in a plane parallel to the yz-plane (see Figure 3.4). Due to the coordinate transformation the projection of shadow casting surfaces and the front/behind classification of vertices are easier to perform. All vertices and later the sun direction vectors are transformed to the new coordinate system by multiplication with a transformation matrix. The transformation matrix represents a three dimensional rotation about a rotation axis, which is rectangular to both the x-axis of the world coordinate system and the outwarding normal vector of the current wall. In the example shown in Figure 3.4, the rotation axis is equal to the z-axis. The transformation matrix is given by the following equation according to [Rogers et. al., 1976]:

$$[\mathrm{M}] = \begin{bmatrix} \mathrm{R}_1^2 + (1-\mathrm{R}_1^2)\cos\varphi & \mathrm{R}_1\mathrm{R}_2(1-\cos\varphi) + \mathrm{R}_3\sin\varphi & \mathrm{R}_1\mathrm{R}_3(1-\cos\varphi) - \mathrm{R}_2\sin\varphi \\ \mathrm{R}_1\mathrm{R}_2(1-\cos\varphi) - \mathrm{R}_3\sin\varphi & \mathrm{R}_2^2 + (1-\mathrm{R}_2^2)\cos\varphi & \mathrm{R}_2\mathrm{R}_3(1-\cos\varphi) + \mathrm{R}_1\sin\varphi \\ \mathrm{R}_1\mathrm{R}_3(1-\cos\varphi) + \mathrm{R}_2\sin\varphi & \mathrm{R}_2\mathrm{R}_3(1-\cos\varphi) - \mathrm{R}_1\sin\varphi & \mathrm{R}_3^2 + (1-\mathrm{R}_3^2)\cos\varphi \end{bmatrix} \tag{0.5}$$

where the terms are defined as the following:

R_1 = cosine of the angle between the rotation axis and the x-axis,
R_2 = cosine of the angle between the rotation axis and the y-axis,
R_3 = cosine of the angle between the rotation axis and the z-axis and
φ = angle between the outwarding normal and the x-axis.

The rotation axis is determined by computing the vector product of the outwarding normal vector and the unit vector in x-direction of the world coordinate system. However, if both vectors are collinear, then no rotation axis can be obtained by the vector product. For identical vectors, the transformation matrix is equal to the unit matrix. In the other case, the z-axis is chosen arbitrarily as the rotation axis. The new coordinate system is called the wall coordinate system and is used in the following operations of the program.

Figure 0.4 Coordinate transformation

Quadrant Test

The quadrant test is a pretesting procedure, which is performed only once for each receiving wall. The test combines the mandatory front/behind test (see Section 2.2) with a preliminary rectangle test. The front/behind test eliminates shadow casting surfaces which lie completely behind the current receiving wall. Also, shadow casting surfaces which lie partially behind the receiving wall are detected and marked for special treatment during the projection procedure. The remaining shadow casting surfaces of the front/behind test are presorted and classified into four groups. Each group corresponds to a “sun quadrant” which is the quadrant of the wall coordinate system where the sun is currently located. The sun quadrants are shown in Figure 3.5. Due to the subdivision of the shadow casting surfaces into four groups, the number of potential shadow casting surfaces for a specific sun’s position can be greatly reduced. Also, the results of the quadrant test make it easy to decide at an early stage whether the receiving wall can actually be shaded or not. If there are no remaining shadow casting polygons after the quadrant test, no external shading calculations need to be performed at all. The sunlit fraction for the receiving wall and all windows on the wall are set to 1 for all sun’s position. Only the internal solar distribution calculations must be performed

Figure 0.5 Definition of sun quadrants

In order to determine the potential shadow casting surfaces of each sun quadrant, a bounding rectangle of the receiving wall is generated first. Then each vertex of a shadow casting surface is tested as to whether it lies in front, behind, right, left, above or below the bounding rectangle. These tests are performed by comparing x, y and z coordinates of the shadow casting surface and the bounding rectangle. If all vertices are found to be behind the receiving wall, then the shadow casting surfaces is eliminated for all quadrants. Depending on the results from the rectangle test, the shadow casting surface is stored for the different quadrants according to rules shown in Table 3.1. For example, if all vertices of a shadow casting surface are found to be below the bounding rectangle, then the shadow casting surfaces can cause no shadows when the sun is in quadrant one and two. The shadow casting surface is not stored for these two quadrants.

Table 0.1 Storing rules of shadow casting surfaces in the quadrant test

	Vertices of shadow casting surface				
	all above	**all below**	**all right**	**all left**	**remaining combination**
Quadrant 1	stored	not stored	stored	not stored	stored
Quadrant 2	stored	not stored	not stored	stored	stored
Quadrant 3	not stored	stored	not stored	stored	stored
Quadrant 4	not stored	stored	stored	not stored	stored

Sunlit/shaded Test for Receiving Wall

The sunlit/shaded test determines, whether the sun is behind the receiving wall and thus the receiving wall is shaded. In this case, no further external and internal shading calculations are performed for that sun's position. For testing, the angle between the sun vector and the normal vector of the receiving wall is calculated. If this angle is equal to or smaller than 90 degrees, then the sun is behind the receiving wall. Figure 3.6 shows the sunlit/shaded classification in two dimensions.

Figure 0.6 Sunlit/shaded classification of walls

Projection of Shadow Casting Polygons

The projection procedure projects the shadow casting surfaces on the plane of the current receiving wall. The three dimensional problem is thereby reduced to two dimensions. Besides the actual projection process, the procedure involves several other small tests. The following steps are performed by the projection procedure:

- Quadrant selection
- Orientation test
- Clipping of shadow casting surfaces which are partially behind
- Actual projection

- Rectangle test
- Order test
- Checking for remaining shadow casting surfaces

Quadrant selection

The projection procedure begins with the selection of the group of shadow casting surfaces corresponding to the current sun vector. Therefore, the procedure determines in which quadrant of the wall coordinate system the sun is currently located. The definition of the quadrants is shown in Figure 3.5. The determination of the quadrant is performed by checking the sign of the y and z coordinates of the transformed sun vector according to Table 3.2. According to the quadrant one of the four groups of shadow casting polygons is selected.

Table 0.2 Quadrant selection

	y-coordinate of sun vector	**z-coordinate of sun vector**
Quadrant 1	-	-
Quadrant 2	+	-
Quadrant 3	-	+
Quadrant 4	+	+

Orientation test

After the selection of the corresponding quadrant, an orientation test is performed for all potential shadow casting surfaces in order to avoid the degeneration of a surface into a line after projection. The orientation test calculates the angle between the normal vector of the shadow casting surface and the sun vector. If this angle lies between 89° and 91°, then the surface is disregarded for that sun position. Furthermore, the orientation test eliminates shadow casting surfaces that are sunlit external walls of the building because it is not necessary to consider both sunlit and shaded external walls. As mentioned in Section 2.1, it is more convenient to consider the shaded walls only, because a building possesses fewer shaded surfaces than sunlit ones. If the previously determined angle between the surface normal vector and the sun vector is smaller than 90 degrees, the surface is shaded and thus further processed (see Figure 3.6). The later test is not performed for 2D obstructions because they are not necessarily part of a 3D solid.

Clipping of shadow casting surfaces which are partially behind

Before the actual projection, TRNSHD checks the shadow casting surfaces for a marker indicating that the surface lies partially behind the receiving wall. In order to avoid false shadow projection, the parts of the polygon that lie partially behind the receiving wall must be clipped off before the projection.

In order to clip off the ”behind” parts, the shadow casting surface is projected on the zx-plane or the xy plane of the wall coordinate system. The zx-plane is selected if the angle between the normal vector of the shadow casting surface and the z-axis is greater then the angle between the normal vector of the shadow casting surface and the y-axis; otherwise the xy-plane is used. The projection is performed by ignoring the y coordinate and z coordinate, respectively. Afterwards, the bounding rectangle of the projected shadow casting surface is determined. The rectangle is then divided into two subrectangles by a line representing the plane of the current receiving wall. The result is that one subrectangle lies in front of the plane of the current receiving wall and the other one lies behind the plane. Now, the projected shadow casting polygon is clipped against the subrectangle that lies behind the plane of the receiving wall. The technique used for clipping is described in Section 3.2.7. The remaining parts of the shadow casting surface after the clipping process represent the “front” parts of the surface. These parts are reprojected to their original location in the wall coordinate system.

Actual Projection

The “front” surfaces are projected on the receiving wall by projecting each of their vertices along the direction of the sun vector. For vertex projection, a line in parametric representation is established with the vertex and the sun direction vector. The coordinates of the projected vertex are obtained by intersecting the line with the plane of the receiving wall:

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \begin{bmatrix} X_{plane} \\ Y + r \cdot Y_{sun} \\ Z + r \cdot Z_{sun} \end{bmatrix} \tag{0.6}$$

$$r = (X_{plane} - X) / X_{sun} \tag{0.7}$$

where
X, Y, Z = coordinates of the vertex before projection,
X_P, Y_P, Z_P = coordinates of the projected vertex,
$X_{sun}, Y_{sun}, Z_{sun}$ = coordinates of the sun vector,
X_{plane} = plane of the receiving wall.

Rectangle test

For efficiency reasons another rectangle test is performed after the actual projection and before the clipping process. Many cases in which no shadow is cast on the receiving wall will be eliminated through this step. The rectangle test checks whether a projected shadow casting surface lies completely below, above, right or left of the receiving wall. If this is the case, then the shadow casting surface and the receiving wall are disjoint and no shading can occur and the shadow casting surface is disregarded in the clipping procedure for this sun position. Like the quadrant test, the rectangle test is performed by creating a bounding rectangle of the receiving polygon and a subsequently coordinate comparison. Figure 3.7 illustrates the result of the rectangle test. Two out of the four shown shadow casting surfaces are eliminated.

Figure 0.7 Rectangle test

Order test

Finally, the actual order of the vertices of the projected surface is checked, because shadow casting surface must be ordered clockwise for the clipping process. The actual order can be determined by the sign of the area A of the polygon [O'Rourke, 1994]:

$$A = \frac{1}{2}\sum_{k=1}^{n}(Y_k Z_{k-1} - Y_{k-1} Z_k) \qquad (0.8)$$

where
Y_k, Z_k = coordinates of the vertex k, and $Y_1 = Y_n$, $Z_1 = Z_n$
n = number of vertices of the surface
If the area is negative, the vertices are ordered counter clockwise and a reordering procedure is applied.

Checking for remaining shadow casting surfaces

When the projection procedure is finished, the program checks again for remaining shadow casting surfaces. For no remaining shadow casting surfaces, no clipping is performed and the sunlit fraction for beam radiation of the receiving wall and attached windows are set to 1 for the current sun's position.

Clipping

The clipping process is the heart of TRNSHD. The remaining sunlit parts of the receiving wall or window are determined here according to a boundary evaluation method proposed by [Weiler and Atherton, 1977] (see also Section 2.1.1).

Before the actual clipping process begins, the program checks the order of the receiving surface by the sign of its area according to Equation 3.8. The receiving surface must be ordered counter clockwise for the clipping process. After the clipping process the sunlit area of the remaining receiving surface is calculated according Equation 3.8. The sunlit fraction $f_{beam,\ ex}$ for beam radiation can then be calculated by the following equation:

$$f_{beam,ex} = \frac{A_{sunlit}}{A_{total}} \qquad (0.9)$$

where
A_{sunlit} = sunlit area of the receiving surface
A_{total} = total area of the receiving surface
For walls, the sunlit and total area are corrected by the sunlit and total area of all windows attached to the wall.

The clipping procedure itself is performed by a subroutine based on a MS thesis of S.A. Johnston [Johnston, 1985] at the University of Wisconsin-Milwaukee. Fundamental improvements in Johnston's program, especially at the intersection part, were made in order to obtain correct results. In the following description of the procedure, the receiving wall or window will simply be called receiving polygon. Similarly, the shadow casting surfaces will be called shader polygons. The procedure consists of two main parts:

- intersection testing and
- boundary evaluation.

A more detailed, but still simplified structured is shown in Figure 3.8. The subroutine is terminated if no receiving polygons remain or all shadow casting polygons have been used in the clipping process.

Figure 0.8 Simplified block diagram of the clipping procedure

Intersection testing

In order to determine whether a shader polygon actually overlaps the receiving polygon, intersection testing is performed. The intersection testing consists of three main steps.

First, the procedure determines the intersections of both polygons by intersecting each edge of the shader polygon with each edge of the receiving polygon. If intersections are encountered, they are stored in an ordered list. The order of the list is equal to the order they occur on the shader polygon.

In order for the Weiler-Atherton algorithm to work correctly, care must be taken with the identification and placement of the intersections. Therefore in a second step, the intersections are classified into entering intersections, where the shader polygon enters the receiving polygon, and leaving intersections, where the shader polygon leaves the receiving polygon. Figure 3.9 illustrates the classification of intersections. Entering and leaving intersections must occur alternating. If none of the intersections is a vertex of one of the polygons, the classification of the intersections is quite simple. In this case, the classification of the first intersection and therewith of all other intersections is obtained by testing the first vertex of the shader polygon with respect to the receiving polygon. For this point membership test the Jordan Curve theorem is applied. The method works by ”drawing” a straight line from the point in any direction and counting the number of intersections with the polygon boundary. If the number of intersections is even, the point is outside the polygon; otherwise it is inside. However, if intersections are vertices of one of the two polygons, neighbor point tests must be performed in order to determine their correct classification.

In a last step, a second intersection list is established, where the intersections are stored in the order they occur on the receiving polygon. The second list is mapped to the first one in order to enable the switching between the vertices list of the shader and the receiving polygon for the following boundary traversal.

Figure 0.9 Classification of intersections

Boundary evaluation

The way the boundary evaluation is performed depends on the number of encountered intersections. If more than one intersection is detected, then a boundary traversal is performed in order to obtain the resulting polygons. The basic boundary traversal is described in Section 2.1.1. If one or zero intersections are encountered, both polygons are either disjoint (no intersection) or touching each other (one intersection). This means that the receiving polygon might be completely shaded, completely unshaded or possess an inscribed shader. In order to distinguish those cases a point membership classification of a boundary point of the shader polygon with respect to the receiving polygon and visa versa are performed. Again, the Jordan Curve theorem is used for the point membership classification. If both points are found to be out, the polygons are disjoint and no shading occurs. If the boundary point of the receiving polygon is found to be inside the shader polygon, but the boundary point of the shader polygon is out, then the receiving polygon lies inside the shader polygon and is completely shaded. The remaining case is an inscribed shader polygon. In order to avoid polygons with holes, the remaining receiving polygon is subdivided into two simple connected polygons.

Internal Insolation Distribution

After the clipping of an external window, the distribution of the beam radiation that passes through the window onto internal surfaces can be determined. TRNSHD calculates for each internal surface the sunlit fraction of the window that falls on the surface in question. The internal insolation distribution is currently restricted to convex modules. This restriction simplifies the calculation technique greatly because no internal surface can shade another internal surface (see also Section 2.3).

The calculation steps and technique used are similar to the external surface shading procedure. A projection procedure reduces the problem to two dimensions. Like for the shadow casting surfaces in the external shading calculations, a sunlit/shaded test for all walls of the module is performed first. Only sunlit walls are going to be projected and shaded walls are disregarded. Before the projection of a wall, all attached windows of that wall are projected. Afterwards, the wall itself is projected. Due to this order the necessary area corrections for walls with attached windows is simplified.

In the clipping procedure, every window and wall of the module is clipped against the remaining (sunlit) parts of the receiving window from the external shading calculation. The walls and windows, for which the internal insolation is calculated, are treated as "shadow casting objects" in the clipping procedure. The remaining receiving window from the external shading calculations is treated as the "receiving polygon". After each clipping process the remaining sunlit area of the receiving window is calculated and therewith the sunlit fraction of the internal wall and window, respectively:

$$f_{\mathrm{beam,in}} = \frac{A_{\mathrm{sunlit,total}} - A_{\mathrm{sunlit,after}}}{A_{\mathrm{sunlit,total}}} \qquad (0.10)$$

where
$A_{\mathrm{sunlit,\ after}}$ = sunlit area of the receiving window after the clipping process
$A_{\mathrm{sunlit,\ total}}$ = sunlit area of the receiving window after the external shading calculations

If the "shadow casting object" of the last clipping process was a wall with attached windows, the sunlit area after the clipping process is correct by the sunlit areas of the attached windows. The sum of internal sunlit fractions of all inside surfaces of module for an external window must add up to one, because all solar radiation which enters a room must hit an inside surface. TRNSHD checks for the sum the internal sunlit fractions and determines the routine if the sum is found to be equal to one.

Diffuse Sunlit Fraction

TRNSHD evaluates the shading effects of diffuse radiation on external surfaces by computing diffuse sunlit fractions. The diffuse sunlit fraction is defined as the ratio of the diffuse sky radiation on a surface including shading effects over the diffuse sky radiation without shading. As mentioned before the sky is subdivided into patches for this calculation such that the diffuse radiation on a surface is given by the sum of radiation from all patches (see Figure 2.5). The diffuse fraction $f_{dfu,ex}$ of external surfaces is determined by the following equations:

$$f_{dfu,ex} = \frac{\sum_{k=1}^{n} \cos\alpha_k \cdot d\omega_k \cdot f_{beam,k}}{\sum_{k=1}^{n} \cos\alpha_k \cdot d\omega_k} \qquad (0.11)$$

$$d\omega_k = \sin\theta_{Z,k} \cdot d\theta \cdot d\gamma \qquad (0.12)$$

where
n = number of patches where receiving surface is sunlit,
α_k = angle between the surface normal vector and the sun vector of patch k,
$\theta_{Z,k}$ = solar zenith angle of patch k,
$f_{beam,k}$ = "beam" sunlit fraction of patch k,
$d\omega_{Z,}$ = increment of the solid angle of patch k,
$d\gamma$ = increment of the solar azimuth angle of patch,
$d\theta_{Z,}$ = increment of the solar zenith angle of patch.

TRNSHD and the TRNSYS Family

TRNSHD is designed to be a part of the TRNSYS family for building simulation. The proposed integration of TRNSHD in the building simulation process with TRNSYS is shown in Figure 3.10. It can be seen that TRNSHD is used as a preprocessing tool at the hierarchical level of BID (see Section 1.2). The advantage of preprocessing the data is that results from TRNSHD can be reused in several simulation runs provided that the geometry remains the same. This reuse of results greatly reduces the computing time, especially during parametric studies. Another advantage of the preprocessing is, that for each month an average day can be used thus improving the accuracy (see Section 4.5.2). If TRNSHD would be integrated in the hourly loop of the building simulation, then the first day of the month must be used like in DOE-2.

The input file of TRNSHD containing the geometric information is created by TRNSCAD. As mentioned earlier, TRNSCAD is used to create or digitalize the geometry of a building. TRNSHD creates files with the sunlit fractions for external and internal walls and windows. These output files are used during the actual building simulation by TYPE 56, the multiple zone building model.

During the design process of TRNSHD care was taken to limit the necessary adjustments of other programs of the TRNSYS family. However, some adjustments must be performed in TRNSCAD and TYPE 56. The output routines from TRNSCAD must be extended in order to create the input file for TRNSHD. The distributor of TRNSCAD (TRANSSOLAR) is already working on the generation of the output file. In Type 56, two categories of adjustments must be performed: formal adjustments, like the reading of the sunlit fractions and the surface identification process, and conceptional adjustments of the routines computing the solar gains. The sunlit fractions of TRNSHD are going to be used in TYPE 56 to calculate the average irradiance level over the receiving surface from the irradiance level of the unshaded surface. This can be done by a simple multiplication:

Figure 0.10 Integration of TRNSHD in the building simulation process

$$\overline{G} = G_{\text{unshaded}} \cdot f \tag{0.13}$$

where

$\overline{G}$ = average irradiance level over the whole receiver surface
G_{unshaded} = irradiance level of the unshaded receiver surface
f = sunlit fraction

For external surfaces, the integration of Equation 3.13 is simple because TYPE 56 receives the beam and diffuse irradiance level for each external surface as an input from TYPE 16, the solar radiation processor. The adjustments for the internal solar distribution are more extensive because the current model treats all entering solar radiation in a zone as diffuse radiation and distributes it via constant absorptance weighted area ratios. A distinction between entering beam and diffuse radiation must be implemented. For beam radiation, the calculation of the average irradiance level of each internal wall must be performed during each time step. If the surface is an interior window, then the transmission and absorption properties of the glazing must be used to determine the solar gain through the window into the adjacent zone. If the surface is opaque, then the absorbed and reflected radiation needs to be determined. The internal distribution of diffuse radiation remains the same. The adjustments of TYPE 56 are also going to be performed by TRANSSOLAR.

CHAPTER
FOUR

VALIDATION

In the previous chapter, the program TRNSHD was introduced. In this chapter TRNSHD is used to study different shading geometries. For validation, the results of TRNSHD are compared with other algorithms: TYPE 34 (overhang and wingwall shading) of TRNSYS, Xsun, a proposed European Standard (CEN) procedure and experimental measurements. In addition, the major assumptions of TRNSHD and their effects on the results are discussed.

Comparison with TYPE 34

Up to now, the blocking of solar radiation in TRNSYS has been performed by TYPE 34, overhang and wingwall shading. As described earlier in Section 1.2, TYPE 34 calculates the shading of beam, diffuse and ground reflected radiation of a vertical rectangular receiver surface due to rectangular overhangs and wingwalls.

For comparison, shading calculations of beam and diffuse radiation for simple geometric configurations were carried out with both programs TYPE 34 with TRNSYS and TRNSHD. The dimensions of the example receiving surface and its obstructions are shown in Figure 4.1. The receiving surface is a vertical south-facing window with the dimensions 2 m by 2 m. The window can be shaded by up to three obstructions: an overhang and two wingwalls. The overhang and wingwalls are at a distance of 0.5 m from the edges of the window and extend 0.7 m from the plane of the window. The window is located in Madison, WI at a latitude of 43.1°N and a longitude of 89.4°.

Figure 0.1 Dimensions of the shading configuration

Beam radiation

Both programs TYPE 34 with TRNSYS and TRNSHD use the same algorithm to calculate the solar position, but the algorithms that determine the sunlit area are different. TYPE 34 uses an ASHRAE algorithm [ASHRAE, 1975] that was contributed by T.Y. Sun, whereas TRNSHD employs the Weiler-Atherton method (see Section 3.2.7). The shading calculations of beam radiation were performed on July 17th. Four combinations of the basic geometric configuration shown in Figure 4.1 were chosen:

- a south-facing window with an overhang,
- a south facing window with a left wingwall,

- a south-facing window with a right wingwall and
- a south-facing window with an overhang and two wingwalls.

The resulting sunlit fractions are shown in Figure 4.2. The results of both programs show good agreement, they give the same results to five significant digits.

Figure 0.2 Sunlit fractions of the window for the four shading configurations

Diffuse radiation

For diffuse radiation shading, the two programs use different approaches. TYPE 34 calculates the view factor from the receiver surface to the sky by subtracting the view factor from the window to the overhang and wingwalls from the view factor of an unshaded vertical surface to the sky (see Section 1.2). For determining the view factors from the window to the overhang and wingwalls, TYPE 34 integrates over the receiver area. TRNSHD determines the sunlit fraction of a receiver surface for diffuse radiation by discretizing the sky into patches and a summing over all patches. In order to obtain comparable results, the sunlit fractions of TRNSHD are converted to view factors by multiplication with the view factor of an unshaded window to the sky. Again, three combinations of the basic geometric configuration shown in Figure 4.1 were chosen:

- a south-facing window with an overhang,
- a south facing window with a wingwall and,
- a south-facing window with an overhang and two wingwalls.

Additionally, the number of patches is varied for TRNSHD. Figure 4.3 presents the resulting view factors of the window to the sky. The results show that an increasing number of patches in TRNSHD improves the accuracy and yields to a good agreement with the results of TYPE 34. For 240 and 7200 patches, both programs gave the same first three digits and a deviation of less than 0.1%. However, 72 patches also yields satisfactory results for a wingwall. For a single overhang, a deviation of 0.4% is obtained. A deviation of 1.0% results for a single wingwall. The largest deviation of 2.7% is encountered when all three obstructions are used together. The impact of the number of patches used on simulation results is further investigated in Section 4.5.1.

Figure 0.3 View factors of the window to the sky for different shading configurations

Comparison with Xsun

Xsun is a software application developed by the Danish Building Research Institute [Grau and Johnsen, 1995]. For beam radiation, Xsun determines the sunlit and shaded areas of windows located on a plane surface, i.e. the facade of a building, by polygon clipping. The implemented clipping algorithm is based on the Weiler-Atherton method like the one used TRNSHD (see Seciton 2.1.1). The algorithms for determining the solar position are not discussed in detail.

For comparison, the shading configuration presented in Section 4.1 is used. Again, a single south-facing window can be shaded by up to three obstructions as shown in Figure 4.1. The window is located in Madison, WI at a latitude of 43.1°N and a longitude of 89.4°. Three cases of shading configurations are selected:

- a south-facing window with an overhang,

- a south facing window with a wingwall and,
- a south-facing window with an overhang and two wingwalls.

The shading calculations are performed with both programs Xsun and TRNSHD for June 13th and July 17th. On June 13th, the solar time is approximately equal to local time according to Xsun. July 17th represents the average day of the month July. For the comparison, the averaging of the hour angle in TRNSHD is turned off, because Xsun calculates instantaneous values of the sunlit fractions.

Figure 4.4 shows the graphical output of Xsun for a window with a single wingwall at 8 am on June 13th. The results of the calculations are shown in Table 4.1 and 4.2. It can be seen that a good agreement of the calculated sunlit fractions is obtained for June 13th. In general, both programs give the same first three digits. The remaining deviations result from the fact that the local time used in Xsun is only approximately equal to solar time, whereas in TRNSHD the local time is equal to solar time. For July 17th, deviations of up to 6% are obtained. These deviations can be caused by the conversion of local time into solar time, because the results of both programs showed a good agreement on June 13th when the local time is approximately equal to solar time. Small deviations of the solar zenith and azimuth angle can result in relative high errors in the sunlit fraction. Unfortunately, the algorithms used for calculating the solar position in Xsun are not published, so that further examinations are not possible.

Figure 0.4 Graphical output of Xsun on June 13th at 8 am

Table 0.1 Sunlit fractions for the three cases on June 13th

	Window with overhang		Window with wingwall		Window with overhang and wingwalls	
Hour	TRNSHD	Xsun	TRNSHD	Xsun	TRNSHD	Xsun
8.00	0.678	0.678	0.570	0.570	0.248	0.248
9.00	0.406	0.406	0.777	0.777	0.183	0.183
10.00	0.221	0.221	0.922	0.922	0.143	0.143
11.00	0.244	0.244	1.000	1.000	0.244	0.244
12.00	0.383	0.384	1.000	1.000	0.283	0.284
13.00	0.244	0.244	1.000	1.000	0.244	0.244
14.00	0.221	0.221	1.000	1.000	0.143	0.143
15.00	0.406	0.406	1.000	1.000	0.183	0.183
16.00	0.678	0.677	1.000	1.000	0.248	0.248

Table 0.2 Sunlit fractions for the three cases on July 17th

	Window with overhang		Window with wingwall		Window with overhang and wingwalls	
Hour	TRNSHD	Xsun	TRNSHD	Xsun	TRNSHD	Xsun
8.00	0.714	0.730	0.533	0.514	0.246	0.246
9.00	0.446	0.471	0.752	0.737	0.199	0.207
10.00	0.321	0.322	0.908	0.897	0.229	0.220
11.00	0.343	0.339	0.999	0.996	0.342	0.335
12.00	0.378	0.378	1.000	1.000	0.378	0.378
13.00	0.351	0.356	1.000	1.000	0.351	0.356
14.00	0.320	0.322	1.000	1.000	0.243	0.253
15.00	0.411	0.388	1.000	1.000	0.185	0.176
16.00	0.687	0.669	1.000	1.000	0.247	0.248

Validation with CEN Standard Procedure

The proposed European CEN standard [CEN, 1995] specifies the input data and the procedures for calculation under transient conditions of the internal temperature in summer of a single room without any cooling equipment. A validation procedure for the determination of the sunlit area of a window due to external obstructions is included in the proposed standard. The validation procedure requires the evaluation of the sunlit factor defined as the ratio of the sunlit area of a plane surface to its total area for six defined tests. The sunlit factor used in the CEN standard is equivalent to the sunlit fraction used in this work. The following tests are required by the proposed standard:

- Test 1: South-facing window with an overhang,
- Test 2: South-facing window with two wingwalls,
- Test 3: South-facing window with an overhang and two wingwalls,
- Test 4: South-facing window with an opposite obstruction,
- Test 5: East-facing window with an overhang and
- Test 6: East-facing window with an opposite obstruction.

The dimensions of the window and its obstructions for the six tests are shown Figure 4.5. The location of the window is at a latitude of 50°N and a longitude of -15°. The validation procedure requires that for each test instantaneous values of the sunlit factor should be determined on July 15th at each hour or each half hour from 7 a.m. to 12 p.m. solar time. Due to the resolution of TRNSHD hourly values are used for validation. In order to obtain instantaneous sunlit fractions, the averaging of the hour angle in TRNSHD is turned off.

Figure 0.5 Dimensions of the window and its obstructions acc. to the proposed standard

The results of the calculations with TRNSHD and the reference values given in the proposed standard are shown in Table 4.3. The validation procedure allows a deviation of 0.05 for each value. It can be seen that good agreement of the results is obtained and that TRNSHD meets the requirements of the proposed CEN standard. In general, the rounded sunlit fractions of TRNSHD are equal to the given values.

Table 0.3 Sunlit fractions for six tests according to the proposed CEN standard

	Test 1		Test 2		Test 3	
Hour	TRNSHD	CEN	TRNSHD	CEN	TRNSHD	CEN
7.00	0.000	0.00	0.000	0.00	0.000	0.00
8.00	0.529	0.53	0.471	0.47	0.000	0.00
9.00	0.238	0.24	0.762	0.76	0.000	0.00
10.00	0.210	0.21	0.967	0.97	0.178	0.18
11.00	0.299	0.30	1.000	1.00	0.299	0.30
12.00	0.327	0.33	1.000	1.00	0.327	0.33
	Test 4		Test 5		Test 6	
Hour	TRNSHD	CEN	TRNSHD	CEN	TRNSHD	CEN
7.00	0.000	0.00	1.000	1.00	0.000	0.00
8.00	1.000	1.00	0.890	0.89	0.000	0.00
9.00	1.000	1.00	0.714	0.71	0.071	0.07
10.00	1.000	1.00	0.391	0.39	0.719	0.72
11.00	0.901	0.90	0.024	0.00	1.000	1.00
12.00	0.847	0.84	0.000	0.00	1.000	1.00

Validation of the Internal Solar Distribution Procedure

Messadi [Messadi, 1990] presents experimental measurements and computer simulations of the solar distribution of beam radiation in a room with a single window. The experimental set-up was located at the University of Michigan-Ann Arbor.

Figure 0.6 Dimensions of the test room

Figure 4.6 shows the dimensions of the test room. The test room is a rectangle parallelipipedic enclosure with a single, south oriented window. Experiments for two different window sizes and locations on the south-facing wall, case A and case B, were performed. Instantaneous measurements of the sunlit area on the internal surfaces of the room were taken every hour in solar time. For the window configuration A the measurements were recorded on October 29th, for case B the measurements were taken on November 24th. In addition to the measured values, calculated values of the sunlit area of internal surfaces are presented by Messadi. The used program uses simultaneously tests for angular boundaries that are established for each of the 64 sunlit configurations that can occur in a rectangle, parallelipipedic enclosure with a single, south oriented window. In order to obtain comparable results, the sunlit fractions of TRNSHD must be converted into sunlit areas on the internal surfaces by the following equation:

$$A_{\text{sunlit},k} = f_{\text{beam},\text{in},k} \cdot A_{\text{window}} \cdot \left| \frac{\cos \alpha_{\text{window}}}{\cos \alpha_k} \right| \tag{0.1}$$

where
$A_{\text{sunlit,k}}$ = sunlit area of the internal surface k
$f_{\text{beam,in,k}}$= sunlit fraction of the internal surface k for beam radiation
A_{window}= total area of the window
α_{window} = angle between the sun vector and the normal vector of the window
α_k = angle between the sun vector and the normal vector of the internal surface k
Tables 4.4 and 4.5 show the resulting sunlit areas of the internal walls of the test room from the experimental measurements and calculations performed by Messadi and the calculations performed by TRNSHD. It can be seen that the sunlit areas calculated by Messadi and the sunlit areas calculated by TRNSHD show good agreement. In general, they give the same first two digits. The remaining deviation between the results can be caused by the determination of the declination because the exact latitude used by Messadi is not known. Comparing the measured values with the calculated values of TRNSHD, deviations of up to 0.3 ft^2 are obtained. These deviations can also be caused by errors related to the conditions of the experiment. According to Messadi, the main error source rises from the fact that the boundaries of the sunlit areas are hard to determine due to a lack of sharpness of the line between shadow and light. Also, the measurements were taken manually with a steel graduated ruler and simultaneous measurements of all dimensions of the sunlit configuration could not be possibly achieved, especially when the latter falls over three internal surfaces. Additional error sources are defects in the construction model (lack of orthogonality, horizontaly and verticality), deviation of the window orientation from the true south and the conversion between local time and solar time. A relative error between the measurements and the results of TRNSHD of the total sunlit area of all internal walls is calculated for each hour of the day. For case A, the largest encountered hourly error amounts to 8%, for case B the largest error turned out to be 2%. These hourly errors are acceptable considering the errors related to the conditions of the experiment. Also, the results show that the sunlit areas appropriately appeared at the expected location on the internal surfaces of the room.

Table 0.4 Sunlit areas [ft^2] on the internal surfaces for case A on October 29th

	West wall			East wall		
Hours	measured	calculated	TRNSHD	measured	calculated	TRNSHD
12.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13.00	0.0000	0.0000	0.0000	0.9431	0.9535	0.9553
14.00	0.0000	0.0000	0.0000	2.0496	2.3524	2.2468

15.00	0.0000	0.0000	0.0000	2.4671	2.3429	2.3419
16.00	0.0000	0.0000	0.0000	1.6944	1.6176	1.6156
17.00	0.0000	0.0000	0.0000	N/A	1.0036	1.0014
	Ground wall			North wall		
Hours	measured	calculated	TRNSHD	measured	calculated	TRNSHD
12.00	4.0416	4.0251	4.0266	0.0000	0.0000	0.0000
13.00	3.4459	3.6470	3.6470	0.0000	0.0000	0.0000
14.00	1.8281	1.8569	1.9680	0.0055	0.0065	0.0066
15.00	0.3131	0.3251	0.3230	0.0000	0.0000	0.0000
16.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
measured and calculated values according to [Messadi, 1990]						

Table 0.5 Sunlit areas [ft^2] on the internal surfaces for case B on November 24th

	East wall			West wall		
Hours	measured	calculated	TRNSHD	measured	calculated	TRNSHD
8.00	0.0000	0.0000	0.0000	N/A	1.9083	1.9067
9.00	0.0000	0.0000	0.0000	N/A	2.7158	2.7157
10.00	0.0000	0.0000	0.0000	2.5907	2.8788	2.8829
11.00	0.0000	0.0000	0.0000	0.3881	0.4471	0.4489
12.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13.00	0.6451	0.4468	0.4489	0.0000	0.0000	0.0000
14.00	2.8663	2.8788	2.8829	0.0000	0.0000	0.0000
15.00	2.6132	2.7158	2.7157	0.0000	0.0000	0.0000
16.00	N/A	1.9084	1.9067	0.0000	0.0000	0.0000
	Ground wall			North wall		
Hours	measured	calculated	TRNSHD	measured	calculated	TRNSHD

8.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9.00	N/A	0.3843	0.3811	0.0000	0.0000	0.0000
10.00	2.1994	1.9471	1.9438	0.1388	0.1358	0.1353
11.00	3.4188	3.2798	3.2780	0.8333	0.9259	0.9279
12.00	3.4444	3.3532	3.3512	0.8055	0.9647	0.9672
13.00	3.1909	3.2798	3.2780	0.9062	0.9261	0.9279
14.00	1.9705	1.9471	1.9438	0.1279	0.1358	0.1353
15.00	0.4792	0.3843	0.3811	0.0000	0.0000	0.0000
16.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
measured and calculated values according to [Messadi, 1990]						

Impact of Assumptions in TRNSHD

In TRNSHD, some assumptions were made in order to reduce both time and memory requirements. The major assumptions concern the number of patches used for diffuse sky radiation shading and the frequency at which beam radiation shading is performed. The impact of these assumption on the calculation results is discussed in the following sections.

Number of Patches Used for Diffuse Sky Radiation

The accuracy of the diffuse radiation shading calculations in TRNSHD depend on the number of patches used in the discretization procedure of the sky. A greater number of patches improves the accuracy, but increases also the computing efforts. The number of patches in TRNSHD is currently set to 72. In Section 4.1, a south-facing window with an overhang and two wingwalls was already investigated. The results showed that for a number of 240 patches a deviation of less than 0.1% was obtained compared to the results of TYPE 34 with TRNSYS. A higher number of patches (7200) did barely improve the accuracy. For 72 patches, the largest encountered deviation was 2.7%. In this section, additional typical shading configurations that can occur for buildings are examined.

In the following shading calculations, two levels of sky discretization are used in TRNSHD, 72 and 240 patches. In addition to the calculations with TRNSHD, shading calculations for the same geometric configurations are performed with TYPE 34 of TRNSYS, if possible. As explained in Section 4.1, the sunlit fraction of TRNSHD are multiplied by the view factor of the unshaded receiving surface to the sky in order to obtain comparable results to TYPE 34. The chosen shading configurations are based on the south-facing window shown in Figure 4.1. In the first calculation set, a single wingwall is selected. The length L of the wingwall is varied from 0.7 m to 15.0 m. In a second calculation set, the height of the wingwall was varied from 2.0 m to 10.0 m. A single overhang is selected for the third calculation set. The length L of the overhang is varied from 0.7 m to 10.0 m. Finally, a rectangular vertical obstruction is defined opposite the wall at a distance of 8.0 m from the window as shadow casting surface. The height and width of the obstruction are varied. For this geometric configuration, no calculations are performed with TYPE 34, because TYPE 34 cannot handle opposite obstructions.

The results of the four calculation sets are shown in Figure 4.7 - 4.10. A good agreement of the results is obtained for all calculations. The improvement in accuracy by using 240 patches instead of 72 is small. For 72 patches the largest deviation amounts to 4.2%. In general, a deviation of less than 1% is obtained. The results leads to the conclusion that using number of 72 patches is appropriate for diffuse radiation shading of buildings.

Figure 0.7 View factors from the window to the sky for a wingwall with varying length

Figure 0.8 View factors from the window to the sky for a wingwall with varying height

Figure 0.9 View factors from the window to the sky for a overhang with varying length

Figure 0.10 View factors from the window to the sky for an opposite obstruction with varying size

Frequency of Beam Radiation Shading Calculations

In TRNSHD, the frequency at which the shading calculations of beam radiation are performed is set to each hour of one day for each month. The values calculated for that day are used throughout the month. The chosen day corresponds to the average day of that month according to [Klein, 1977]. A more accurate way is to perform the shading calculations for each day of a month because the declination which represents the angular position of the sun at solar noon varies throughout the month. However, a daily calculation would also increase the computation effort by a factor of 30.4.

Figure 0.11 Declination throughout the year

In order to investigate the effects on the accuracy of the calculation results, the declination was calculated for the first, the average and the last day of each month throughout the year as shown in Figure 4.11. It turned out that the largest deviation of the declination throughout a month exists in March. Thus, the largest deviation between the ”one-day-a-month” method of TRNSHD and an accurate daily calculation is expected in March. Shading calculations are carried out for a south-facing window with an overhang and two wingwalls as described in Section 4.1. In order to approximate the worst case, the shading calculations are performed on March 1st, March 16th (average day) and March 31st. The results are shown in Figure 4.12. Deviations of up to 0.15 between average and either first or last day did occur for certain hours of the day. It can also be seen, that the use of an average day compared to the first day of a month as used in DOE-2 improves the accuracy. For March the solar gains would be always under predicted, if the first day would be used.

Figure 0.12 Sunlit fraction for a window with an overhang and two wingwalls in March

The chosen average day corresponds to the conditions of average solar radiation according to Klein [Klein, 1977]. Because only one day is used in a month, a difference can occur between the sunlit fractions of the average day and the true average of the hourly sunlit fractions of each day of a month. For investigating this difference, hourly sunlit fractions are calculated for every day of the month March. Then, the arithmetic average value is calculated and compared to the sunlit fraction of the average day. The largest encountered deviation of an hourly sunlit fraction amounts to less than 0.01. This shows that using the average day is a good choice.

Averaging of the irradiance level

As mentioned in Section 3.3 the sunlit fractions calculated by TRNSHD are used to calculate an average irradiance level over a whole surface. In this section, the impact of the usage of an average irradiance level, rather than the sunlit and shading areas separately, on the heat gain of a room by conduction through wall or roof is investigated.

The heat gain by conduction through an opaque surfaces can be calculated by the transfer function method [ASHRAE, 1993]. This method uses the sol-air temperature to represent outdoor conditions and a constant indoor air temperature. Furthermore, both indoor and outdoor surface heat transfer coefficients are assumed constant. The heat flux is assumed to be one dimensional. According to the transfer function method the heat gain through a wall or roof is given by:

$$q_{e,t} = A\left[\sum_{n=0} b_n (T_{e,t-n\delta}) - \sum_{n=1} d_n \{(q_{e,t-n\delta}) / A\} - T_{rc}\sum_{n=0} c_n\right] \qquad (0.2)$$

with $$T_{e,t-n\delta} = T_{o,t-n\delta} + \alpha \cdot \bar{I}_{total,t-n\delta} \big/ h_o - \varepsilon \cdot \Delta R \big/ h_o \qquad (0.3)$$

where

$q_{e,t}$ = heat gain through a wall or roof, at calculation hour t
A = indoor surface area of a wall or roof
b_n, c_n, d_n = conduction transfer function coefficients
t = time
δ = time interval
n = summation index (each summation has as many terms as there are non-negligible values of coefficients)
$T_{e,t-n\delta}$ = sol-air temperature at time t-nδ
T_{rc} = constant indoor room temperature
$T_{o,t}$ = outdoor air temperature at calculation hour t
α = absorptance of surface for solar radiation
$\bar{I}_{total,t-n\delta}$ = total solar radiation incident on surface at calculation hour t
h_o = coefficient of heat transfer by long-wave radiation an convection at outer
surface
ε = hemispherical emittance of surface
R = difference between long-wave radiation incident on surface from sky and
surroundings and radiation of blackbody at outdoor air temperature

Equation 4.2 shows that only the first term that contains the sol-air temperature is affected directly by the irradiance level. The irradiance level that appears in Equation 4.3 is an averaged irradiance level over the whole surface area. The second term of Equation 4.2 represents the "history" of the heat flux and the third term is independent of the irradiance. Thus, the following investigation is focused on the first term of Equation 4.2 and the sol-air temperature. Additionally, it is assumed for the following algebra that the total radiation is pure beam radiation. Therewith, the first term can be rewritten:

$$A_{total}\sum_{n=0} b_n (T_{e,t-n\delta}) = A_{total}\sum_{n=0} b_n (\cdot T_{o,t-n\delta} + \alpha \cdot \bar{I}_{beam,t-n\delta} \big/ h_o + \varepsilon \cdot \Delta R / h_o) \qquad (0.4)$$

The average beam irradiance level in Equation 4.4 can be replaced by the product of the sunlit fraction f_{beam} and the beam irradiance level I_{beam} for an unshaded surface:

$$A_{\text{total}} \sum_{n=0} b_n (T_{e,t-n\delta}) = A_{\text{total}} \sum_{n=0} b_n (\cdot T_{o,t-n\delta} + f_{\text{beam},t-n\delta} \cdot \alpha \cdot I_{\text{beam},t-n\delta} / h_o + \varepsilon \cdot \Delta R / h_o) \quad (0.5)$$

In order to avoid averaging the irradiance level, the whole surface area must be subdivided into subareas each with uniform irradiance level. For beam radiation, two types of subareas exist, a sunlit subarea and a shaded subarea. The first term of Equation 4.2 can also be rewritten by using an uniform irradiance level and subareas:

$$\begin{aligned} A_{\text{total}} \sum_{n=0} b_n (T_{e,t-n\delta}) = \sum_{n=0} b_n \{ & (A_{\text{sunlit},t-n\delta} + A_{\text{shaded},t-n\delta}) \cdot T_{o,t-n\delta} \\ & + \alpha / h_o \left(A_{\text{sunlit},t-n\delta} \cdot I_{\text{beam},t-n\delta} + A_{\text{shaded},t-n\delta} \cdot 0 \right) \\ & + (A_{\text{sunlit},t-n\delta} + A_{\text{shaded},t-n\delta}) \cdot \varepsilon \cdot \Delta R / h_o \} \end{aligned}$$

$$= A_{\text{total}} \sum_{n=0} b_n (\cdot T_{o,t-n\delta} + A_{\text{sunlit},t-n\delta} / A_{\text{total}} \cdot \alpha \cdot I_{\text{beam},t-n\delta} / h_o + \varepsilon \cdot \Delta R / h_o)$$

$$(0.6)$$

It can be seen that Equation 4.5 and Equation 4.6 are identical, because the sunlit fraction for beam radiation is defined as the ratio of the sunlit area over the total surface area. Thus for beam radiation, the averaging of the irradiance level over the whole surface area does not yield to errors on the calculation of heat flux through an opaque surface. For diffuse radiation, the algebra is not as easy as for beam radiation. However, the conclusions derived for beam radiation can be transformed to diffuse radiation because the diffuse radiation shading problem can be divided into many "beam" radiation problems by discretizing the sky into patches (see Section 2.1.2 and Section 3.2.9).

CHAPTER

FIVE

SHADING STUDY WITH TRNSHD

After the validation of TRNSHD with relatively simple shading geometries, a more complex shading study is performed in order to demonstrate the capabilities of the program. In addition, the use of TRNSHD for other objects than buildings is discussed.

Shading Study of an Atrium

For the case study, a building with an atrium is chosen as shown in Figure 5.1. Atria and sunspaces have become a major architectural component of a building design and a challenge for energy engineering. In general, atria posses large external glazing surfaces. The judicious admission of sunlight into an atrium creates its high aesthetic values. Besides the ambience of an atrium provided by the admission of sunlight, the large glazing surfaces can contribute to high cooling and heating loads. The design of an atrium with respect to both aesthetic values and energy efficiency is a very complex task and requires a detailed shading study and thermal analysis of the atrium.

The building with atrium used in the following shading study is an L-shaped building with three stories. Figure 5.2 shows a plan view of the building and its surrounding with

Figure 0.1 Building with atrium

the main dimensions. The atrium has a south-east facing vertical glass wall and a sloped glass roof. The building has two neighbor building where one is located south and the other one east. The neighbor buildings are modeled as “boxes” with the same height as the investigated building. The building is located in Madison, WI at a longitude of 89.4° and a latitude of 43.1°N. Shading can occur from the building itself and the neighbor buildings. For the shading study of the atrium three levels of shading calculations are chosen:

- Case 1: no external shading
- Case 2: shading originated by the building itself and
- Case 3: shading originated by the building itself and its neighbor buildings.

The results for beam radiation are shown in Figure 5.3 and 5.4. Due to shading effects the solar access in the atrium is significantly reduced. For the south-east facing

Figure 0.2 Plan view of investigated building and its surrounding

wall, the building itself blocks the beam radiation in January and March only during the afternoon hours. This is due to the fact that the solar azimuth angle is smaller then 90°E and no shading caused by the building itself can occur during the morning. In May and June shading caused by the building itself occurs on the south-east facing atrium wall during the morning and afternoon hours. Shading effects at noon and later are low due to a higher sun position (small zenith angle). The neighbor buildings have the greatest impact in January, where the sun position is low. For the glass roof, the solar gains in the afternoon are reduced by self-shading of the building during all months. The shading effects of the neighbor buildings on the glass roof are smaller than for the vertical glass wall.

Figure 0.3 Sunlit fractions of the glass wall

Figure 0.4 Sunlit fractions of the glass roof

Figure 0.5 Diffuse sunlit fractions for the atrium glazing

Figure 5.5 presents the results for diffuse radiation. The patches of the hemisphere are shown in two dimensions and each patch is colored according to the corresponding sunlit fraction. In addition to the sunlit fraction for each patch, the resulting sunlit diffuse fraction is shown in Figure 5.5. For the vertical glass wall, the blocking of diffuse sky radiation by the building itself is small. 95% of the diffuse radiation still strikes the wall. The impact of the neighbor buildings is relative high. If the neighbor buildings are also considered, then the diffuse radiation access is reduced to 44% compared to an completely unshaded wall. The results illustrate how important it is to consider not only the building in question, but also its surroundings. For the glass roof, the impact of diffuse sky radiation is reduced to 65% by shading through the building itself. Neighbor buildings cause a further reduction of the diffuse radiation down to 57%.

The impact of shading on the amount of solar radiation incident on a surface is examined with TRNSYS for all three shading cases. The monthly simulations are performed for Madison, WI at a latitude of 43.1°N and a longitude of 89.4°. For radiation data, a TMY weather data file from Madison is used. Figure 5.6 and Figure 5.7 show the resulting monthly radiation on the vertical glass wall and the glass roof of the atrium. It can be seen that for the glass wall the impact of the neighbor buildings is large compared to the impact of the selfshading of the building, whereas for the glass roof the impact of the selfshading of the building is more important. The figures also show that for this atrium the diffuse radiation contributes a significant portion of the total monthly radiation. Thus, neglecting diffuse radiation shading yields an overprediction of solar gains for the investigated atrium.

A generalization of the results and conclusions obtained for the one individual atrium is not valid, because the results depend strongly on the weather data, the location and geometric configuration.

Figure 0.6 Monthly radiation on the glass wall for the three shading configurations

Figure 0.7 Monthly radiation on the glass roof for the three shading configurations

Shading Study of Other Objects than Buildings

Although TRNSHD was develop for building simulations with TRNSYS, it is a stand-alone tool that can used for problems other than buildings. In general, every type of solar receiver, like a thermal flat-plate collector or a photovoltaic panel, that can be approximated by planar polygons, can be handled by TRNSHD. However, it needs to be assured that an average radiation level over the whole receiver area is an appropriate approach. For solar collectors it is usually adequate to use an average radiation level [Duffie and Beckman, 1991], but for photovoltaic systems the effect on power by using average radiation may not be correct.

The geometric description of the receiver-obstruction configuration must satisfy the data structure and its requirements presented in Section 4.1. The receiver surfaces must be defined as ”external walls”. The number of internal walls, external and internal window and modules can be set to zero. An example input file for a single receiver surface which illustrates the shading by row of thermal collectors is described in Appendix A.

CHAPTER
SIX

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

A computer program for external shading and internal insolation calculations of solar beam and diffuse radiation has been developed. The calculation techniques described in chapter 3 has been implemented in a software application called TRNSHD. TRNSHD is a stand-alone tool, which has been developed for building simulations with TRNSYS. However, TRNSHD is not restricted to buildings or TRNSYS and can be used to solve other shading problems.

The validation process of TRNSHD described in chapter 4 shows that TRNSHD functions correctly and the results of TRNSHD are in good agreement with expected values. Also, it shows that the sunlit areas appropriately appeared at the expected location. The external shading procedure for beam radiation has been validated against an ASHRAE algorithm for overhang and wingwall shading and against a proposed CEN standard. The results of TRNSHD showed a good agreement with both. In addition, the results of TRNSHD have been compared to results of another software application called Xsun. The results of both programs show a good agreement. The results show that the sunlit fractions are sensitive with respect to the determination of the solar angles. For validating the external shading procedure for diffuse radiation, results of TRNSHD have been compared to those of TYPE 34 with TRNSYS for different shading configurations. It has been found that a total number of 72 sky patches yield an acceptable accuracy when calculating diffuse radiation shading. A comparison of TRNSHD with experimental measurements and computer simulations has been performed for the internal solar distribution procedure. The simulated values show a good agreement to the results of TRNSHD. The obtained deviation between the measured values and the predicted values of TRNSHD are acceptable considering the errors related to the conditions of the experiment. A discussion of the frequency at which beam radiation is performed yield the conclusion that calculations on an average day of each month is sufficient.

Finally, a complex shading study of an building with an atrium described in Section 5.1 has demonstrated the use and the capabilities of TRNSHD. Also, the study has illustrated that the surroundings of a building like neighbor buildings and the landscape can have a significant impact on the amount of solar radiation that strikes an external surface. However, the generalization of results of shading calculations is difficult, because they depend strongly on the location, the weather data and the geometry.

Recommendations for Future Work

The computer program TRNSHD is a stand-alone tool that has been developed for building simulations with TRNSYS. The integration of TRNSHD in the TRNSYS family must be performed next. Therefore, adjustments in other preprocessing tools like TRNSCAD and PREBID and in the actual building model, TYPE 56, are necessary. The German distributor of TRNSYS, TRANSSOLAR, is already working on those issues.

Further work can also be done in adding features to TRNSHD and optimization of the calculations. Recommended features are:

- **Semi-transparent obstructions**

Semi-transparent obstructions would permit the modeling of shade trees and other kinds of non-opaque objects in TRNSHD. Therefore, the clipping procedure needs to be extended such that not only the non-overlapping parts but also the overlapping parts are determined.

- **Non-convex room modules**

In the current version of TRNSHD, room modules are restricted to convex 3D solids. An extension to non-convex room modules would greatly improve the generality of the internal solar distribution calculations.

- **Reflected radiation shading**
  Diffuse and beam radiation is reflected by the ground and adjacent buildings. As mentioned in Section 2.1.3, the shading of reflected radiation is a complex problem, but it can have a significant impact on building loads. Thus, an incorporation of reflected radiation shading calculations is suggested.

For further optimization of the program several ideas occurred during this thesis. The following tasks are suggested:

- **Resetting of sunlit fractions when the sun is not above the horizon**

When the sun is not above the horizon TRNSHD sets the sunlit fractions to zero. It can happen that for some day the sun rises an hour earlier than for the average day. A zero value of the sunlit fraction can cause an error for such a configuration. A better approach would be to reset values of the sunlit fraction, when the sun is not above the horizon, to the first and last calculated value of the sunlit fraction of that day.

- **Optimizing the handling of shadow casting surfaces which lie partially behind the receiving surface**

The subroutine which determines the front parts of shadow casting surfaces which lies partially in front and partially behind the current receiving wall is now implemented in such a way that it must be executed for every solar position. A less time consuming approach would be to execute this procedure only once for each shadow casting surface which lies partially behind, store the determined front parts of the surface and reuse them for the following shading calculations of the receiving wall.

- **Optimizing the calculations for beam and diffuse radiation shading**
  For beam radiation hourly values of the sunlit fractions are calculated according to the solar position in the sky. For diffuse radiation shading calculations with TRNSHD the whole sky is subdivided into patches and for each center point of a patch a “beam” radiation shading calculation is performed. Thus, for same patches the same calculation is performed twice, once for beam radiation and once for diffuse radiation. An interesting approach could be to optimize the shading calculations by reusing results from the beam radiation shading calculation for the diffuse radiation shading calculations.

APPENDIX A

In this appendix two examples for the use of TRNSHD for shading and insolation calculations are given. In addition, the limitations on maximum number of elements allowed in the program are listed.

Example 1 - Building

Example 1 illustrates the use of TRNSHD for shading and insolation calculations for a building. Figure A1 shows the building with one obstruction and the chosen world coordinate system. The one story building is composed of two rooms (modules). The two modules share on internal wall with one internal window. The smaller module has one window, the larger module has two windows. One obstruction is located opposite of the west front of the building.

Figure A1 Building used for example 1

The geometric description of the building must be written in an input file for TRNSHD. The description must satisfy the data structure and requirements discussed in Section 3.1 as well as a special format. Figure A2a shows the listing of the corresponding geometric input file called EXAMPLE1.GEO of example 1. The identification numbers of walls, windows, obstructions and vertices used in Figure A2a are illustrated in Figure A2b. As mentioned in Section 3.1 care must be taken in the definition of internal walls. They must be defined twice: once for each module with the corresponding outward drawn normal vector with respect to the module. In addition to the geometric input data file, TRNSHD requires information about the desired calculation mode. This information is entered in an interactive way. In order to minimize computation time, the lowest acceptable level of detail should be selected. The available calculation modes are discussed in Section 3.1. The interactive input for this example is shown in Figure A3. The calculations are performed for Madison, WI in solar time.

The calculation results of TRNSHD, the sunlit fractions, are written in the following files:

- EXWALL.SHA contains the sunlit fractions for all external walls of the building
- EXWINDOW.SHA contains the sunlit fractions for all external windows of the building
- WID*****.ISL contains the internal sunlit fractions of all walls and windows of a module for the external window with the identification number *****. Note, that such a file is created for each external window for which an internal solar distribution calculation is performed. Thus, three insolation files are created for the building of example 1, WID1.ISL, WID2.ISL and WID3.ISL.

The output file EXWALL.SHA for the example building is shown in Figure A4. The first column of this file contains the ID numbers of all defined external walls. The ID numbers appear in the order in which they are entered in geometrical input file. The sunlit fractions for diffuse radiation are written in the second column. Note, that the sunlit fractions for diffuse radiation are set to one if no diffuse radiation shading calculations are performed. The following columns contain the sunlit fractions for beam radiation. The time of the year, in hours, for which the sunlit fractions of beam radiation are calculated is written in the second row. As mentioned in Chapter 3, if no shading of a wall can occur, because no other walls or obstructions are in front of that wall, all sunlit fractions of that wall are set to one for all solar positions as is the case for wall 3, 4, 5, 6, 7, 9, 10, 11 and 12. Figure A5 presents the output file EXWINDOW.SHA for external windows. The format of this output file is similar to the output file EXWALL.SHA for walls. Figure A6 shows the internal insolation file WID1.ISL of window 1 of the example building The first column of the insolation file contains the identification number of all walls and windows of a module. In contrast to the output files of shading calculations (*.SHA) the insolation files (*.ISL) contain only sunlit fractions for beam radiation.. Thus, all the following columns contain sunlit fractions for beam radiation on inside walls and windows. The sum of all values of one column must add up to one, because all solar radiation which enters a room must hit an inside surface. TRNSHD checks the sum of each column and writes on error message on the screen, if the sum is not equal to one.

```
******** NORTH DIRECTION VECTOR ***************************************
0. 1. 0.        /X Y Z COORDINATES OF THE NORTH VECTOR/
******** VERTICES *****************************************************
36               /TOTAL NUMBER OF VERTICES/

1                /ID # OF VERTEX/
 0.  0.  0.      /X Y Z COORDINATES OF VERTEX 1/
2                /ID # OF VERTEX/
 2.  0.  0.      /X Y Z COORDINATES OF VERTEX 2/
3                /ID # OF VERTEX/
 2. -3.  0.      /X Y Z COORDINATES OF VERTEX 3/
4                /ID # OF VERTEX/
 3. -3.  0.      /X Y Z COORDINATES OF VERTEX 4/
5                /ID # OF VERTEX/
 4. -3.  0.      /X Y Z COORDINATES OF VERTEX 5/
6                /ID # OF VERTEX/
 4.  0.  0.      /X Y Z COORDINATES OF VERTEX 6/
7                /ID # OF VERTEX/
 4.  4.  0.      /X Y Z COORDINATES OF VERTEX 7/
8                /ID # OF VERTEX/
 0.  4.  0.      /X Y Z COORDINATES OF VERTEX 8/
9                /ID # OF VERTEX/
 3.5 0.  0.      /X Y Z COORDINATES OF VERTEX 9/
10               /ID # OF VERTEX/
 2.5 0.  0.      /X Y Z COORDINATES OF VERTEX 10/
11               /ID # OF VERTEX/
 0.  0.  3.      /X Y Z COORDINATES OF VERTEX 11/
12               /ID # OF VERTEX/
 2.  0.  3.      /X Y Z COORDINATES OF VERTEX 12/
13               /ID # OF VERTEX/
 2. -3.  3.      /X Y Z COORDINATES OF VERTEX 13/
14               /ID # OF VERTEX/
 3. -3.  3.      /X Y Z COORDINATES OF VERTEX 14/
15               /ID # OF VERTEX/
 4. -3.  3.      /X Y Z COORDINATES OF VERTEX 15/
16               /ID # OF VERTEX/
 4.  0.  3.      /X Y Z COORDINATES OF VERTEX 16/
17               /ID # OF VERTEX/
 4.  4.  3.      /X Y Z COORDINATES OF VERTEX 17/
18               /ID # OF VERTEX/
 0.  4.  3.      /X Y Z COORDINATES OF VERTEX 18/
19               /ID # OF VERTEX/
 3.5 0.  2.      /X Y Z COORDINATES OF VERTEX 19/
20               /ID # OF VERTEX/
 2.5 0.  2.      /X Y Z COORDINATES OF VERTEX 20/
21               /ID # OF VERTEX/
 2. -0.5 1.      /X Y Z COORDINATES OF VERTEX 21/
22               /ID # OF VERTEX/
 2. -2.5 1.      /X Y Z COORDINATES OF VERTEX 22/
23               /ID # OF VERTEX/
 2. -2.5 2.      /X Y Z COORDINATES OF VERTEX 23/
24               /ID # OF VERTEX/
 2. -0.5 2.      /X Y Z COORDINATES OF VERTEX 24/
25               /ID # OF VERTEX/
 0.  3.5 1.      /X Y Z COORDINATES OF VERTEX 25/
26               /ID # OF VERTEX/
 0.  2.5 1.      /X Y Z COORDINATES OF VERTEX 26/
27               /ID # OF VERTEX/
```

Figure A2a Listing of the geometric input file EXAMPLE1.GEO

```
0.  2.5 2.      /X Y Z COORDINATES OF VERTEX 27/
28               /ID # OF VERTEX/
 0.  3.5 2.      /X Y Z COORDINATES OF VERTEX 28/
29               /ID # OF VERTEX/
 0.  1.5 1.      /X Y Z COORDINATES OF VERTEX 29/
30               /ID # OF VERTEX/
 0.  0.5 1.      /X Y Z COORDINATES OF VERTEX 30/
31               /ID # OF VERTEX/
 0.  0.5 2.      /X Y Z COORDINATES OF VERTEX 31/
32               /ID # OF VERTEX/
 0.  1.5 2.      /X Y Z COORDINATES OF VERTEX 32/
35               /ID # OF VERTEX/
-2.  3.  0.      /X Y Z COORDINATES OF VERTEX 35/
36               /ID # OF VERTEX/
-2.  1.  0.      /X Y Z COORDINATES OF VERTEX 36/
37               /ID # OF VERTEX/
-2.  1.  3.      /X Y Z COORDINATES OF VERTEX 37/
38               /ID # OF VERTEX/
-2.  3.  3.      /X Y Z COORDINATES OF VERTEX 38/
******** OBSTRUCTIONS **************************************************
1               /TOTAL NUMBER OF OBSTRUCTIONS/

1               /OBSTRUCTION SET IDENTIFICATION NUMBER/
5               /NUMBER OF VERITCES OF OBSTRUCTION/
35 36 37 38 35  /ORDERED LIST OF VERTICES (ID #) OF OBSTRUCTIONS/
******** EXTERNAL WINDOWS **********************************************
3               /NUMBER OF EXTERNAL WINDOWS/

1               /WINDOW SET IDENTIFICATION NUMBER/
5               /NUMBER OF VERITCES OF WINDOW/
21 22 23 24 21  /ORDERED LIST OF VERTICES (ID #) OF WINDOW/
1               /MODULE IDENTIFICATION NUMBER/

2               /WINDOW SET IDENTIFICATION NUMBER/
5               /NUMBER OF VERITCES OF WINDOW/
29 30 31 32 29  /ORDERED LIST OF VERTICES (ID #) OF WINDOW/
2               /MODULE IDENTIFICATION NUMBER/

3               /WINDOW SET IDENTIFICATION NUMBER/
5               /NUMBER OF VERITCES OF WINDOW/
25 26 27 28 25  /ORDERED LIST OF VERTICES (ID #) OF WINDOW/
2               /MODULE IDENTIFICATION NUMBER/
******** INTERNAL WINDOWS **********************************************
1               /TOTAL NUMBER OF INTERNAL WINDOWS/

4               /WINDOW SET IDENTIFICATION NUMBER/
5               /NUMBER OF VERITCES OF WINDOW/
10 9 19 20 10   /ORDERED LIST OF VERTICES (ID #) OF WINDOW/
******** EXTERNAL WALLS ************************************************
12              /TOTAL NUMBER OF EXTERNAL WALLS/

1               /WALL SET IDENTIFICATION NUMBER/
0. -1. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
1 2 12 11 1     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/
```

Figure A2a Continuing listing of the geometric input file EXAMPLE1.GEO

```
2               /WALL SET IDENTIFICATION NUMBER/
-1. 0. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
2 3 13 12 2     /ORDERED LIST OF VERTICES (ID #) OF WALL/
1               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
1               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

3               /WALL SET IDENTIFICATION NUMBER/
0. -1. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
3 4 14 13 3     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

4               /WALL SET IDENTIFICATION NUMBER/
0. -1. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
4 5 15 14 4     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

5               /WALL SET IDENTIFICATION NUMBER/
1. 0. 0.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
6 5 15 16 6     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

6               /WALL SET IDENTIFICATION NUMBER/
1. 0. 0.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
7 6 16 17 7     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

7               /WALL SET IDENTIFICATION NUMBER/
0. 1. 0.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
7 8 18 17 7     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

8               /WALL SET IDENTIFICATION NUMBER/
-1. 0. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
8 1 11 18 8     /ORDERED LIST OF VERTICES (ID #) OF WALL/
2               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
2 3             /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

9               /WALL SET IDENTIFICATION NUMBER/
0. 0. 1.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
6               /NUMBER OF VERITCES OF WALL/
18 11 12 16 17 18 /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/
```

Figure A2a Continuing listing of the geometric input file EXAMPLE1.GEO

```
10              /WALL SET IDENTIFICATION NUMBER/
0. 0. 1.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
6               /NUMBER OF VERITCES OF WALL/
12 13 14 15 16 12 /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

11              /WALL SET IDENTIFICATION NUMBER/
0. 0. -1.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
6               /NUMBER OF VERITCES OF WALL/
8 1 2 6 7 8     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

12              /WALL SET IDENTIFICATION NUMBER/
0. 0. -1.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
6               /NUMBER OF VERITCES OF WALL/
2 3 4 5 6 2     /ORDERED LIST OF VERTICES (ID #) OF WALL/
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/
******** INTERNAL WALLS ***********************************************
2               /TOTAL NUMBER OF INTERNAL WALLS/

13              /WALL SET IDENTIFICATION NUMBER/
0. -1. 0.       /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
2 6 16 12 2     /ORDERED LIST OF VERTICES (ID #) OF WALL/
1               /NUMBER OF WINDWOS ON THE WALL; 0= NO WINDOW/
4               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/

14              /WALL SET IDENTIFICATION NUMBER/
0. 1. 0.        /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
2 6 16 12 2     /ORDERED LIST OF VERTICES (ID #) OF WALL/
1               /NUMBER OF WINDWOS ON THE WALL; 0= NO WINDOW/
4               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/
******** MODULES ******************************************************
2               /TOTAL NUMBER OF MODULES/

1               /MODULE SET IDENTIFICATION NUMBER/
7               /NUMBER OF WALLS OF MODULE/
2 10 4 12 3 14 5 /ORDERED LIST OF ALL WALLS (ID #) OF MODULE/

2               /MODULE SET IDENTIFICATION NUMBER/
7               /NUMBER OF WALLS OF MODULE/
1 13 6 7 8 9 11 /ORDERED LIST OF ALL WALLS (ID #) OF MODULE/
***********************************************************************
END
```

Figure A2a Continuing listing of the geomteric input file EXAMPLE1.GEO

Figure A2b ID numbers used in the file EXAMPL1.GEO

```
PLEASE ENTER THE NAME OF THE GEOMETRIC INPUT FILE: EXAMPLE1.GEO

TRNSHD OFFERS DIFFERENT LEVELS OF SHADING CALCULATIONS:

*  DO YOU WANT TO PERFORM BEAM RADIATION SHADING FOR
   EXTERNAL WALLS AND WINDOWS? [Y/N]  Y

*  DO YOU WANT TO PERFORM DIFFUSE RADIATION SHADING FOR
   EXTERNAL WALLS AND WINDOWS? [Y/N]  Y


FOR BEAM RADIATION SHADING, ADDITIONAL INPUT IS REQUIRED:

*  PLEASE ENTER THE LATITUDE IN DEGREES:  43.1

*  SHOULD THE CALCULATIONS BE PERFORMED IN SOLAR TIME? [Y/N]  Y

*  DO YOU WANT TO PERFORM INTERNAL SOLAR DISTRIBUTION
   CALCULATIONS FOR BEAM RADIATION? [Y/N]  Y


READING INPUT FILE
```

Figure A3 Interactive input of TRNSHD for example1

Example 2 - Single receiver surface

As a second example for the use of TRNSHD, a shading problem with a single receiver surface is chosen. The shading problem is based on a multiple-row array of thermal flat-plate collectors as described by Duffie and Beckman in example 1.9.3, [Duffie and Beckman, 1991]. The arrangement of the collector array is shown in Figure A7. The collectors have a rectangular shape with a height of 2.10 m and a length 10.00 m. They are sloped at 60° toward south. The calculation of the sunlit fractions of the collector in row N due to shading by the collector in row M is desired.

Figure A7 Multiple-row collector arrangement of example 2 in 2D

Figure A8 shows the listing of the geometric input file, EXAMPLE2.GEO, for this configuration. The collector of row N is defined as an ”external wall”, because the calculation of the sunlit fractions for this collector row is desired. The collector row M is specified as an obstruction. The number of external and internal windows, internal walls and modules is set to zero. The world coordinate system of the input file is shown in Figure A7. The x axis is parallel to the length of the collector row and the y axis is pointing south.

The interactive input of TRNSHD for example 2 is presented in Figure A9. The calculations of beam and diffuse radiation are performed for Madison, WI in local time. Figure A10 shows the resulting output file containing the sunlit fractions for external walls, EXWALL.SHA.

```
******** NORTH DIRECTION VECTOR ***************************************
0. -100. 0.        /X Y Z COORDINATES OF THE NORTH VECTOR/
******** VERTICES ******************************************************
8                     /TOTAL NUMBER OF VERTICES/

1                     /VERTEX ID #/
  0.00  1.05  0.00    /X Y Z COORDINATES OF VERTEX 1/
2                     /VERTEX ID #/
 10.00  1.05  0.00    /X Y Z COORDINATES OF VERTEX 2/
3                     /VERTEX ID #/
  0.00  0.00  1.82    /X Y Z COORDINATES OF VERTEX 3/
4                     /VERTEX ID #/
 10.00  0.00  1.82    /X Y Z COORDINATES OF VERTEX 4/
5                     /VERTEX ID #/
  0.00  3.92  0.00    /X Y Z COORDINATES OF VERTEX 5/
6                     /VERTEX ID #/
 10.00  3.92  0.00    /X Y Z COORDINATES OF VERTEX 6/
7                     /VERTEX ID #/
  0.00  3.92  1.82    /X Y Z COORDINATES OF VERTEX 7/
8                     /VERTEX ID #/
 10.00  3.92  1.82    /X Y Z COORDINATES OF VERTEX 8/
******** OBSTRUCTIONS **************************************************
1               /TOTAL NUMBER OF OBSTRUCTIONS/

1               /OBSTRUCTION SET IDENTIFICATION NUMBER: 1=SET1/
5               /NUMBER OF VERITCES OF OBSTRUCTION/
5 6 8 7 5       /ORDERED LIST OF VERTICES (ID #) OF OBSTRUCTION
******** EXTERNAL WINDOWS **********************************************
0               /NUMBER OF EXTERNAL WINDOWS/
******** INTERNAL WINDOWS **********************************************
```

```
0               /TOTAL NUMBER OF INTERNAL WINDOWS/
******** EXTERNAL WALLS ***********************************************
1              /TOTAL NUMBER OF EXTERNAL WALLS/

1               /WALL SET IDENTIFICATION NUMBER: 1=SET1/
0.00 1.00 0.58  /X Y Z COORDINATES OF THE OUTWARD DRAWN NORMAL VECTOR/
5               /NUMBER OF VERITCES OF WALL/
```

Figure A8 Listing of the geometric input file EXAMPLE2.GEO

```
1 2 4 3 1       /ORDERED LIST OF VERTICES (ID #) OF WALL
0               /NUMBER OF WINDWOS ON THE WALL; 0 = NO WINDOW/
0               /LIST OF WINDOWS (ID #); 0 = NO WINDOW/
******** INTERNAL WALLS ***********************************************
0               /TOTAL NUMBER OF INTERNAL WALLS/
******** MODULES ******************************************************
0               /TOTAL NUMBER OF MODULES/
***********************************************************************
END
```

Figure A8 Continuing listing of the geometric input file EXAMPLE2.GEO

```
PLEASE ENTER THE NAME OF THE GEOMETRIC INPUT FILE: EXAMPLE2.GEO

TRNSHD OFFERS DIFFERENT LEVELS OF SHADING CALCULATIONS:

*  DO YOU WANT TO PERFORM BEAM RADIATION SHADING FOR
   EXTERNAL WALLS AND WINDOWS? [Y/N]  Y

*  DO YOU WANT TO PERFORM DIFFUSE RADIATION SHADING FOR
   EXTERNAL WALLS AND WINDOWS? [Y/N]  Y


FOR BEAM RADIATION SHADING, ADDITIONAL INPUT IS REQUIRED:

*  PLEASE ENTER THE LATITUDE IN DEGREES:  43.1

*  SHOULD THE CALCULATIONS BE PERFORMED IN SOLAR TIME? [Y/N]  N

*  PLEASE ENTER THE DIFFERENCE BETWEEN THE STANDARD MERIDIAN
   FOR LOCAL TIME ZONE AND LONGITUDE OF LOCATION IN DEGREES:  0.6

*  DO YOU WANT TO PERFORM INTERNAL SOLAR DISTRIBUTION
   CALCULATIONS FOR BEAM RADIATION? [Y/N]  N


READING INPUT FILE
```

Figure A9 Interactive input of TRNSHD for example 2

Limitation on the maximum number of elements in TRNSHD

The limitation on the maximum number on elements in TRNSHD are presented in Table A1. They should be sufficient for the use of TRNSHD as a stand-alone tool. However, adjustments can be performed easily by corresponding changes in the source code.

Table A1 Limitations on the maximum number of elements in TRNSHD

Description	**Abbr.**	**Value**
Maximum number of external and internal walls	MAXW	100
Maximum number of internal walls	MAXIW	30
Maximum number of vertices per wall	MAXVW	10
Maximum number of windows per wall	MAXFW	5
Maximum number of external and internal windows	MAXF	50
Maximum number of external windows	MAXEF	40
Maximum number of vertices per window	MAXVF	5
Maximum number of modules	MAXM	5
Maximum number of walls per modules	MAXWM	20
Maximum number of obstructions	MAXO	10
Maximum number of vertices per obstruction	MAXVO	10
Maximum number of vertices (total)	MAXV	300

APPENDIX B

Source code of TRNSHD

BIBLIOGRAPHY

ASHRAE, 1975. Subroutine Algorithms for Heating and Cooling Loads to Determine Building Energy Requirements. Compiled by subcommittee for Heating and Cooling Loads. ASHRAE Task Group on Energy Requirements.

BLAST, 1991. *User Reference. Blast Support Office*. Department of Mechanical and Industrial Engineering. Univ. of Illinois at Urbana-Champaign.

CEN, 1995. Thermal Performance of Buildings - Internal Temperatures in Summer of a room without mechanical cooling - General criteria and calculation procedures. CEN TC89. European Committee for Standardization, Brussels.

Clarke, J., P. Strachan, J. Hand, J. Hensen, C. Pernot and E.Aasem, 1993. *ESP-r, a Program for Building Energy Simulation Version 8 Series*. ESRU Energy Simulation Research Unit, University of Strathclyde, Glasgow, Scotland.

Duffie, J.A. and W.A Beckman, 1991. *Solar Engineering of Thermal Processes*. 2nd Edition. Wiley & Sons.

EREN, 1996. ”Energy Efficiency and Renewable Energy Network.” U.S. Department of Energy. http:/www.eren.doe.gov/

Grau, K. and K. Johnsen, 1995. ”General Shading Model for Solar Building Design.” *ASHRAE Trans*. 101(2):1298-1310

Groth, C.G. and M. Lokmanhekim, 1969. SHADOW - ”A New Technique for the Calculation of Shadow Shapes and Areas by Digital Computer.” *Proc. Hawaii 2nd Int. Conf. on System Sciences*, Univ. of Hawaii: 471-474.

Gueymard, C., 1993. *Shading Calculations for Hourly Building Simulations: Survey of Existing Methods and a New Simplified Algorithm*. Florida Solar Energy Center. FSEC-PF-250-93.

Klein, S.A., 1977. ”Calculation of Monthly Average Insolation on Tilted Surfaces.” *Solar Energy*, 19, 325.

LBL, 1982. *DOE-2 Engineers Manual*. Rpt. LBL-11353, Lawrence Berkeley Lab.

Messadi, M.T., 1990. ”Experimental Validation of The Procedure to Determine the Internal Sunlit Configurations.” *Proc. of the 15th National Passive Solar Conference*. American Solar Energy Society. 251-256.

Mitalas, G.P. and J.G. Arsenault, 1967. FORTRAN IV Program to calculate z-Transfer Functions for the Calculation of Transient Heat Transfer Through Walls and Roofs. Division of National Research Council of Canada, Ottawa.

O’Rourke, J., 1994. *Computational Geometry in C*. Cambridge University Press.

Rogers, D.F. and J.A. Adams, 1976. *Mathematical Elements for Computer Graphics*. McGraw-Hill Book Company.

Johnson, S.A., 1985. *Solar Test (SOLTST): Subroutine to Solar Graphics*. Master Thesis. School of Architecture and Urban Planning. University of Wisconsin-Milwaukee.

Seem, J.E. 1987. *Modeling of Heat in Buildings*. Ph.D. Thesis, Solar Energy Laboratory, Univ. of Wisconsin-Madison.

SEL, 1995*.* *TRNSYS Manual Version 14.1*. Solar Energy Lab. Univ. of Wisconsin-Madison.

Spitler, J.F., 1994. *Annotated Guide to Load Calculations Models and Algorithms (Draft)*. ASHRAE RP 741.

Walton, G.N., 1979. ”The application of homogeneous coordinates to shadowing calculations.” *ASHRAE Trans*., 85(1): 174-180.

Weiler, K and P. Atherton, 1977. ”Hidden Surface Removal Using Polygon Area Sorting.” *Computer Graphics* 11:214-222.

Winkelmann, F.,1996. Personal communication.