NREL/TP-253-4281 • UC Categories: 233,234 • DE91002146

Computer Simulation of Shading and Blocking: Discussion of Accuracy and Recommendations

F.W. Lipps
University of Houston
Houston, Texas

NREL Technical Monitor: R. Gerald Nix

National Renewable Energy Laboratory
(formerly the Solar Energy Research Institute)
1617 Cole Boulevard
Golden, Colorado 80401-3393
A Division of Midwest Research Institute
Operated for the U.S. Department of Energy
under Contract No. DE-AC02-83CH10093

Prepared under Subcontract No. XX-4-04006-1

April 1992

Printed in the United States of America
Available from:
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Microfiche A01
Printed Copy A03

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issue of the following publications which are generally available in most libraries: *Energy Research Abstracts (ERA)*; *Government Reports Announcements* and Index (*GRA* and I); *Scientific and Technical Abstract Reports (STAR)*; and publication NTIS-PR-360 available from NTIS at the above address.

Computer Simulations of Shading and Blocking Events in a Field of Heliostats

F. W. Lipps

Energy Lab., University of Houston

Abstract A field of heliostats suffers losses due to shading and blocking. Shading occurs when a heliostat is shaded from the sun by a neighboring heliostat and similarly, blocking occurs if some of the reflected sunlight hits the back of a neighbor instead of proceeding towards the receiver. The usefully reflecting region is represented by a set of vertices in the plane of the heliostat. This set of points is called the boundary vector. The boundary vector is used to compute the percentage of loss for a heliostat due to its neighbors and also to compute the image of the heliostat on the receiver. The image is needed to determine the receiver interception fraction and the corresponding percentage of loss due to the receiver size.

The complex geometry of multiple shading and blocking events due to several neighbors suggests that a processing type of code is needed to update the boundary vector for each shading or blocking event. A new version called RSABS simulates the split rectangular heliostat (see figure 1). After a discussion of the errors resulting from our simplifying assumptions, we conclude that the dominant error for the given heliostat geometry is due to the departure from planarity of the neighboring heliostats. Systematic errors of several percent can occur in this way. A programmer's manual for RSABS is included.

We recommend that a version of the heliostat simulation be modified to include losses due to non-reflective structural margins, if they occur. See

Figure 6. We also recommend that the heliostat neighbors be given true guidance rather than assumed to be parallel, and that the resulting non-identical quadrilateral images be processed as in HELIOS by ignoring over-lapping events which we find to be extremely rare in optimized fields.

ACKNOWLEDGMENT

This document was prepared with DOE support under Contract EG-77-C-04-3974 and SERI contract XX-4-04006-1. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author and do not necessarily reflect the views of DOE or SERI. The author has developed these codes continuously since June of 1973. The working group has included L. L. Vant-Hull, F. W. Lipps, M. D. Walzel, C. Laurence, and A. Holley.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

1) INTRODUCTION

The loss fraction, F_l, attributed to the shading and blocking of a heliostat is complementary to the fractions of effectiveness for the heliostat, F_e. Clearly, we have

$$F_l + F_e = 1$$

The fraction of effectiveness can be defined in several ways using parallel or stereographic projections of the neighboring heliostats.

It is reasonable to define the fraction of effectiveness for a heliostat, relative to a particular point P on the surface of the receiver, as the ratio of the flux density due to the given heliostat with and without its neighbors. Consequently,

$$F_e(P) = \int_{A_e(P)} dwS(\alpha) / \int_{A_H} dwS(\alpha) ,$$

where dw $S(\alpha)$ is the flux density due to direct beam insolation received from a cone of solid angle dw oriented in a direction α radians from the center of the solar disc. A_H is the reflecting region of the heliostat and $A_e(P)$ is the effective reflecting region after loses due to shading and blocking by the entire set of neighboring heliostats. $A_e(P)$ depends on the point P because the effective region is limited by a stereographic projection of the neighboring heliostats using P as a center. This definition of F_e is inconvenient because it depends on difficult flux density calculations. However, the approximation

$$F_e(P) \cong A_e(P)/A_H$$

holds under the assumptions:

1) The solar disc is uniform, i.e. $S(\alpha)$ is independant of α;

2) The heliostats are relatively small so that the apparent solar disc can fill the heliostat; and

3) The heliostats are well guided and P is relatively near the center of the receiver.

It is usually desirable to define the effective fraction of the heliostat in a receiver independent way. Assuming a point sun, we can write

$$F_e = A_e/A_H$$

where A_H is the area of reflecting surface per heliostat and A_e is the usefully reflecting area of the central heliostat in the neighborhood. A_e is determined by a parallel projection of all relevant neighbors. This definition of F_e is used throughout subsequent discussions. It lacks some precision due to the finite size of the solar disc. However, the effect of the solar disc is very slight and can be estimated by redefining F_e in terms of the total usefully reflected power. Let

$$F_e = \int_{A_e} d\sigma \int_{-\infty}^{\infty} dw S(\alpha)/JA_H$$

where $d\sigma$ is an element of area in a plane perpendicular to the optic axis. A_e is based on the parallel projections, and J is the total direct beam insolation in W/m^2.

$$J = \int dw\ S(\alpha)$$

An inspection of the $d\sigma$ integral reveals regions of zero intensity and regions of maximum intensity equal to J, as well as regions of intermediate intensity corresponding to the familiar umbra and penumbra. The contour of 50% intensity corresponds to the boundary of A_e. The symmetry of the umbra and the penumbra for long straight edges leads to the result that $F_e \equiv A_e/A_H$. However, the vertices of the useful region A_e are exceptional points which introduce small errors.

Unlike procedures which simply sum the area shaded or blocked in each event, without regard for overlapping which may occur, our shading and blocking processor updates the boundary vector of the useful reflecting region each time a shading or a blocking event occurs. There is an initial boundary vector B_0 and a final boundary vector B_1 which determines A_e and F_e. In general a boundary vector has the form

$$B = \{(u_i, v_i) \; i = 1 \ldots N\} ,$$

where (u_i,v_i) represents a point in the plane of the heliostat. The closed region defined by the N vertices of B has the area

$$A = \tfrac{1}{2} \sum_{i=3}^{N} [(u_{i-1}-u_1)(v_i-v_1)-(u_i-u_1)(v_{i-1}-v_1)]$$

The initial boundary vector B_0 gives the area A_H if the u,v coordinates have true scale. However, it is convenient to renormalize the scale of the u,v coordinates so that

$$B_0 \rightarrow A_0 = 1 \text{ and } F_e = A_e.$$

2) PRELIMINARY CONSIDERATIONS

The first shading and blocking processor was developed at the UH Energy Lab in the summer of 1973 for rectangular heliostats in linear neighborhoods with stereographic projections. To our knowledge, this is still the only shading and blocking processor capable of properly handling overlapping images in a general way. The present effort provides an efficient model of the split rectangular heliostat. (See Figure 1). This code is called RSABS.

Early, heliostat design concepts involved square, hexagonal, and circular heliostats. Consequently we proceeded to develop a processor called NGON, which represents regular polygons. This code was used from 1975 to 1979,

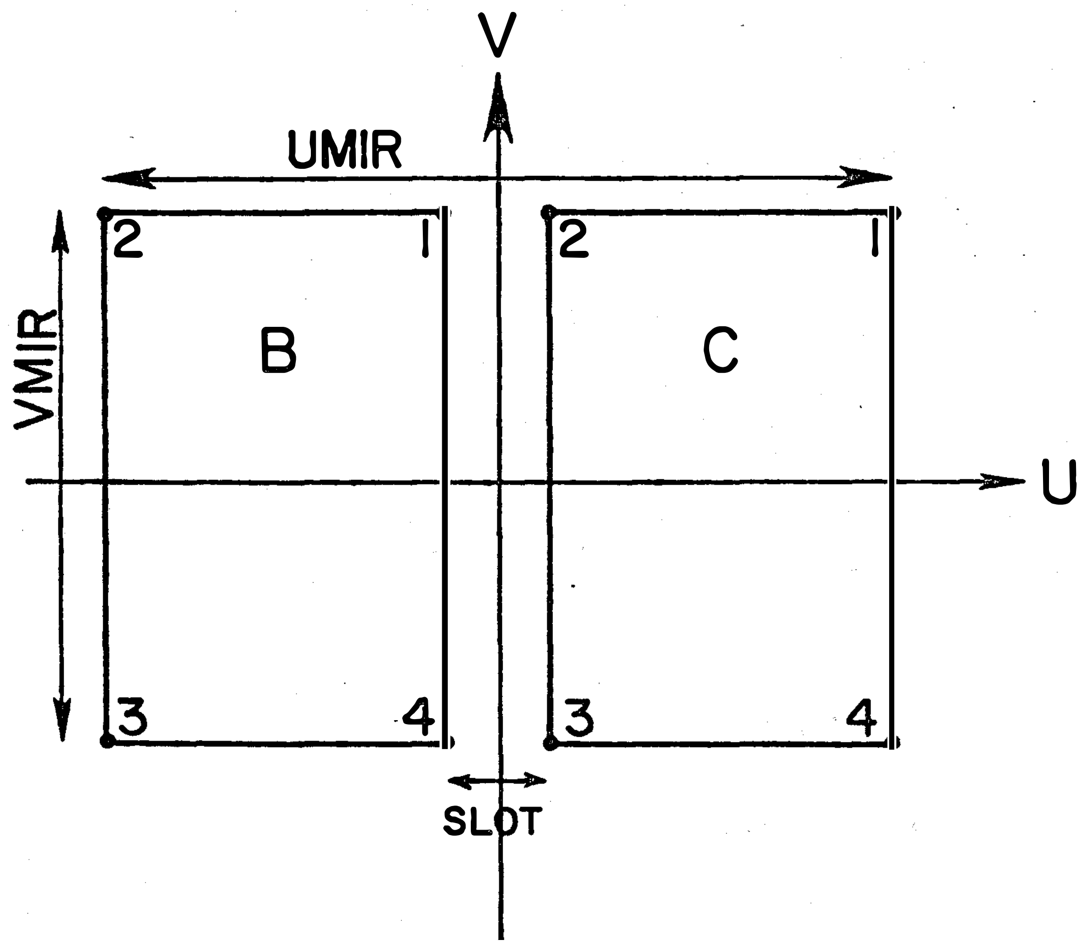

Figure 1) Geometry of Split Rectangular Heliostat.

The regions B and C each have a boundary vector with 4 points initially. There are three input variables; namely, UMIR, VMIR, and SLOT as shown. An additional glass sequent in the slot converts the heliostat into an H or U shaped heliostat.

even though most heliostat designs had clipped corners and median slots which are ignored by NGON. IN 1979 we developed RSABS to include the slot. We do not have a processor for circular heliostats, although we have developed a code (called BL) that represents a bubble enclosed circular heliostat with isolated shading and blocking events. The isolated shading and blocking events include interference by neighboring bubbles as well as by the enclosed heliostat disks. Bubble transmission as a function of angle of incidence for both the enclosing and neighboring bubbles is included. If no bubbles are present circular heliostats could be handled by setting the transmittance function equal to 1.0, but a new code version would be preferable.

The central receiver system concept assumes that the heliostats are mass produced and are structurally similar. Consequently, each heliostat and its neighbors are expected to have the same size and shape. The similarity of the heliostat profiles vastly simplifies the shading and blocking processor. Another simplification, which we use depends on the plane of the collector field. We usually assume that the heliostat centers are coplanar, so that the collector field will have a well defined plane, although it may not be a level plane. Given adverse terrain the collector field may become contoured but we will continue to assume that the neighborhoods are approximately planar. Notice that both the sun and the receiver are expected to be above the plane of the collector field (i.e. planes of the neighborhoods), so that the highest point of a heliostat can not be obscured by a shading or a blocking event. Consequently, the high point of the heliostat is chosen as a fixed point of the processor and is labelled the first vertex.

The boundary vector concept introduced in section (1) assumes that the heliostats have a polygonal profile. It is possible to develop a processor for arbitrary polygons. For example, if X, Y, and Z are polygons having boundary vectors X, $\bar{Y}$, and $\bar{Z}$, then the processor must determine $\bar{Z}$ so that

$$Z = X \cap Y, \text{ ($\cap$ denotes the logical intersection).}$$

In this example, if X represents the reflective area of a central heliostat and Y the boundary of a nearly neighbor, then Z represents their overlap and the polygon X - Z is the new reflective area, which may be subject to additional events. However, this approach will not necessarily lead to efficient calculations and, therefore, it is desirable to develop processors for a series of progressively more complex cases. For instance, we may consider

1) Rectangles (See Figure 2 for typical events),
2) Split Rectangles (See Figure 1 for basic geometry),
3) Various shapes depending on the shape of the slot, and finally
4) Regular polygons such as octagons.

The first two cases will be developed in this report. The third case is an easy extension of the split rectangle geometry to include a third rectangular piece of glass in the slot. The case of regular polygons is essentially more complex but not as bad as arbitrary polygons.

Before going into the design of the code, it desirable to consider the variety of its applications. For generality, all heliostat geometry is referred to an orthonormal basis $(\hat{u},\hat{v},\hat{w})$ which is fixed in the frame of the heliostat. The unit vector $\hat{w}$ is normal to the plane of the heliostat, hence $\hat{w}$ depends on the positions of the sun and the receiver. However, u and v also depend on the choice of the heliostat mounting system which plays a role in determining the high point of the heliostat.

Figure 3 shows a rectangular heliostat with two events. The cross hatched region remains useful after the events. If event (I) comes first in the processor, then event (II) pinches the useful region into two discon-

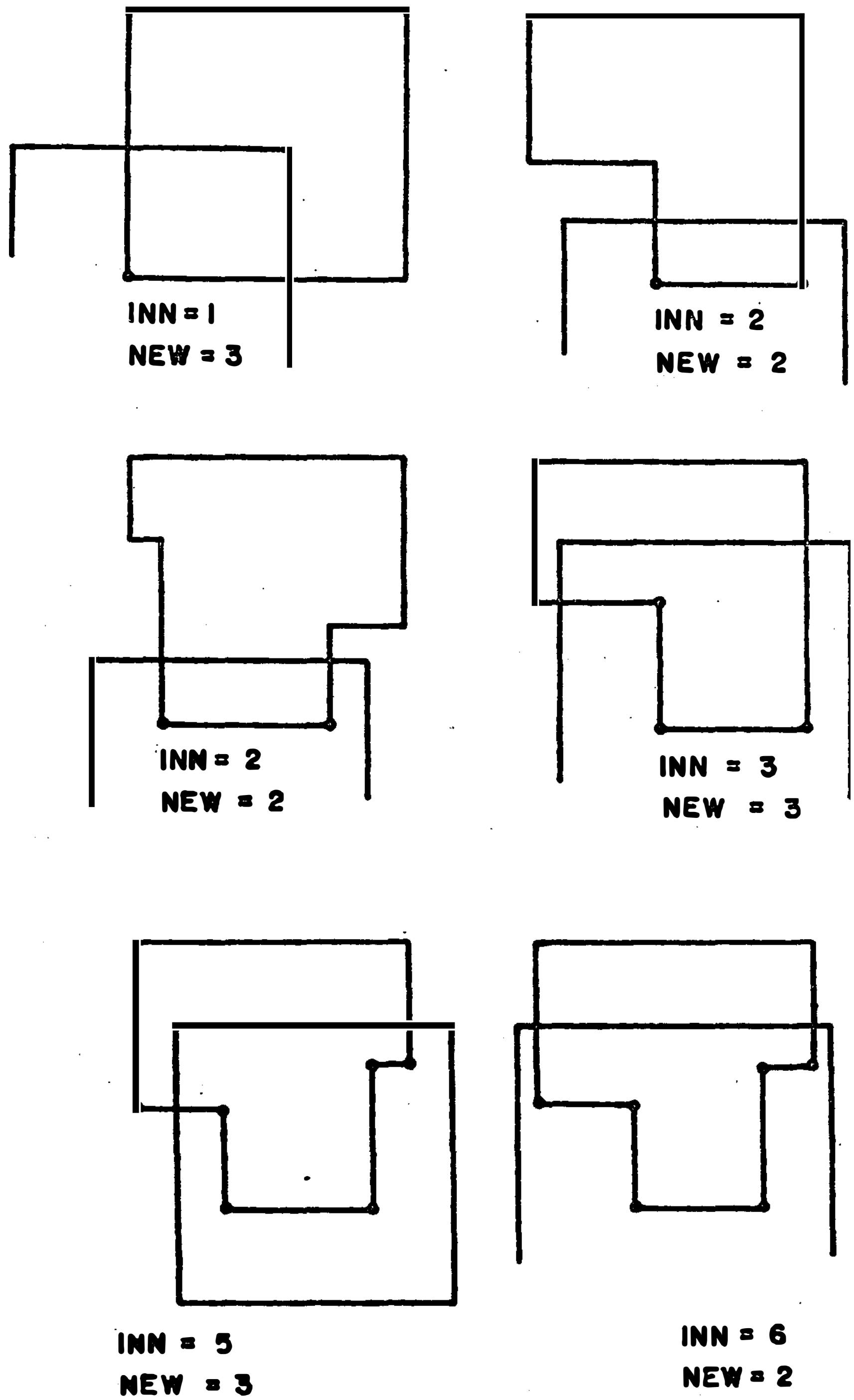

Figure 2) Some Typical Shading and Blocking Events.

The closed curve represents the boundary status before processing the event which is represented by the open curve. The dots represent vertices which must be deleted to process the current event. Crossed lines become new vertices. The current image caused NEW points, and INN points must be deleted.

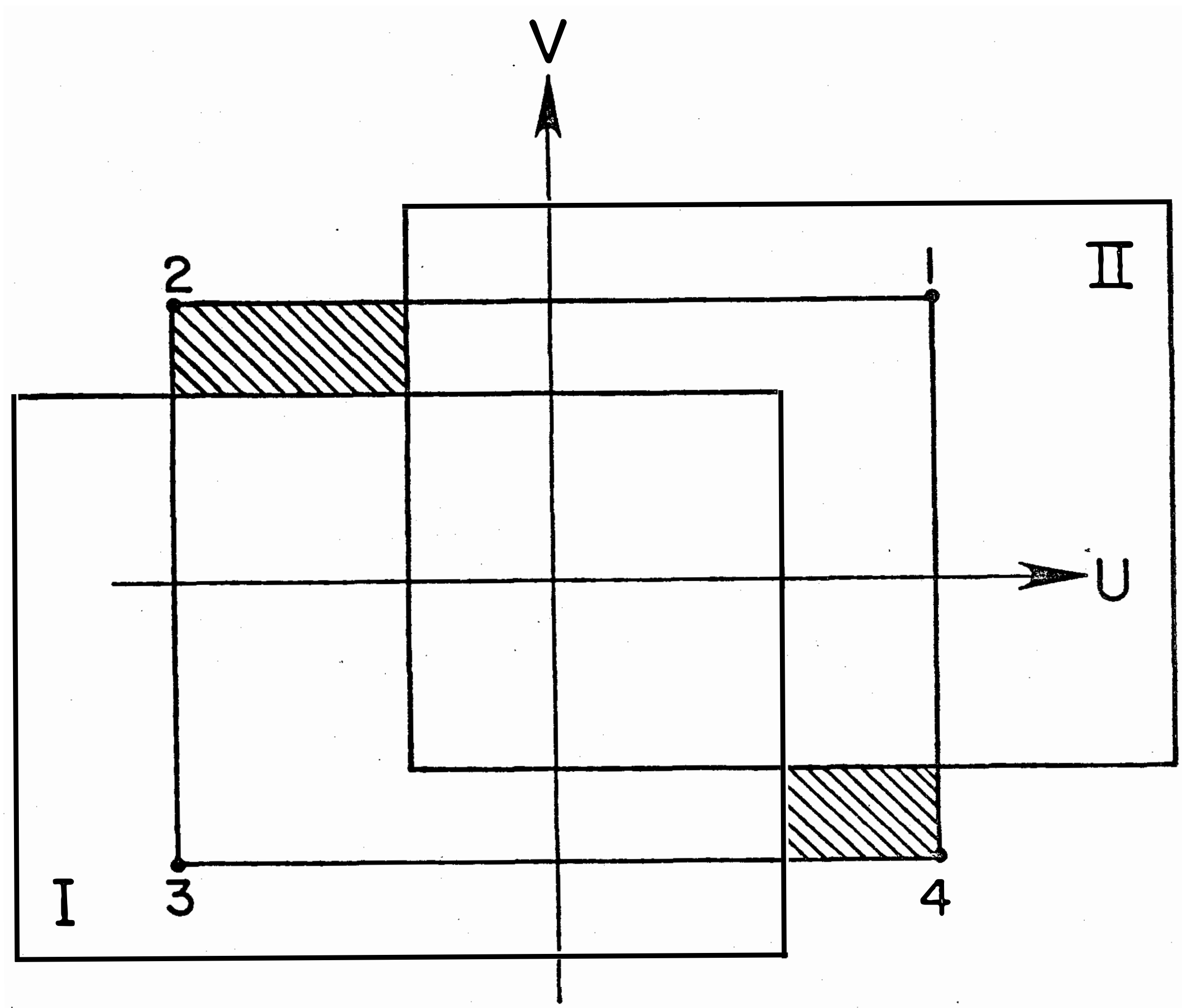

Figure 3) Rectangular Heliostat in (u,v) Plane.

Two events are shown to cause a split region. Points (1,2,3,4) are the vertices of the central heliostat under analysis. The cross hatched area is usefully reflecting. O_I and O_{II} are the images of the center of heliostats I and II under either a shading or a blocking projection. Event II could not cross side 1 → 2 with altazimuthal mountings.

nected areas resulting in a boundary having two cycles. This situation cannot be handled by a single boundary vector and will be ignored for the present. This type of overlap problem can not occur for alt-azimuthal mountings since the û axis is parallel to the ground and therefore the whole upper edge is immune to events. For other mountings the pinch-off problem can occur in realistic collector fields for low solar elevations but very little error occurs if the lower area is deleted. We can afford to ignore it, since all current designs call for alt-azimuthal mountings.

The shading and blocking events depend on the location of the neighbors with respect to the central heliostat. In cell wise models of the collector field, the displacements of the neighboring heliostats are generated independently for each representative heliostat. The neighborhood geometry depends on the incline of the collector field, the type of neighborhood (i.e. radial stagger, N-S stagger, radial cornfield, N-S cornfield, etc.) and the number of neighbors. In an individual heliostat model of the collector field each heliostat has a position vector which locates it with respect to the base of the tower, so that the incline of the field and the type of neighborhood are no longer choices available within the shading and blocking program. However, the size of the neighborhood (i.e., the number of neighboring heliostats) remains a choice under suitable input controls.

The shading and blocking subroutine RSABS and its helper EVENT are designed to handle split rectangular heliostats in the framework of the cell-wise performance code system. See figure 3. Conversion to other code environments is easy. Appendix B contains a list of RSABS and EVENT.

We should also consider the variety of outputs which may be required from a shading and blocking program. The boundary vector is always available if needed by the heliostat image generator. The boundary vector leads quickly to the area of the useful region and the fraction of glass which is usefully reflecting. The useful fraction is frequently the only required output. However, if the heliostat has a sun sensor for guidance purposes, it may be necessary to have a special test in order to determine whether the sensor is exposed to the (reflected) sun at any given time. Current heliostat designs call for computer guidance, so this feature has been omitted. It may be desirable to know the fraction of shading and the fraction of blocking separately, although this feature is not built into the code at this time. We have an additional output array, giving the serial number of the neighbors which are responsible for the first and second shading and blocking events. This output is occasionally useful for interpreting results.

3) TASK STRUCTURE OF CODE

The following list indicates the purpose of each functional element required for a rectangular heliostat shading and blocking subroutine.

1) An initial phase to construct a standard initial boundary vector in dimensionless units. Auxiliary constants are set.
2) The high point of the heliostat is located and its index in the standard boundary vector is determined.
3) The initial boundary vector is loaded, so that its first point is the high point. The initial number of vertices is set.
4) A neighbor heliostat is selected and its displacement from the central heliostat is computed.
5) The displacement is tested to determine whether the neighbor lies on the reflecting or the shading side of the central heliostat.
6) The neighboring heliostat is given a shading and a blocking projection into the plane of the central heliostat. The (u,v) coordinates of the central image points are computed.

7) The remote images are bypassed. (Remote means $u_O^2 + v_O^2$ is large, See Figure 3 for example of a non-remote event).

8) Each vertex in the current boundary vector is tested to see if it lies inside of an image heliostat (i.e. inside of a shaded or blocked region). Basic process control variables are set here. See Figure 2.

9) If no event occurred, the program proceeds to the next image.

10) Dependent control variables are set and the tail-end of the boundary vector is shifted by the number of vertices which are required to accommodate the event. New points will occur and old points will be lost. A net gain or loss is possible.

11) The new points are computed and loaded into the appropriate components of the boundary vector. The current number of vertices is reset.

12) After all images are processed, the area of the region defined by the boundary vector is computed.

13) Subroutine returns with the shading and blocking fraction for a single heliostat at a single time.

The over-all structure of the code is shown in Figure 4.

A slotted-rectangular heliostat requires two boundary vectors $\bar{B}$ and $\bar{C}$, and the useful area is the sum of useful areas due to $\bar{B}$ and $\bar{C}$. See figure 1. In the list above, items (4-11) are inside of the loop over neighbors and their images. If the heliostat is slotted (i.e. split) each neighbor has two rectangular images which can interact with the two rectangular regions in the central heliostat. Each of the four combinations can occur. Hence, it is convenient to build items (8-11) into a subroutine called EVENT which can be called four times for the slotted heliostat case. An H or U-shaped heliostat

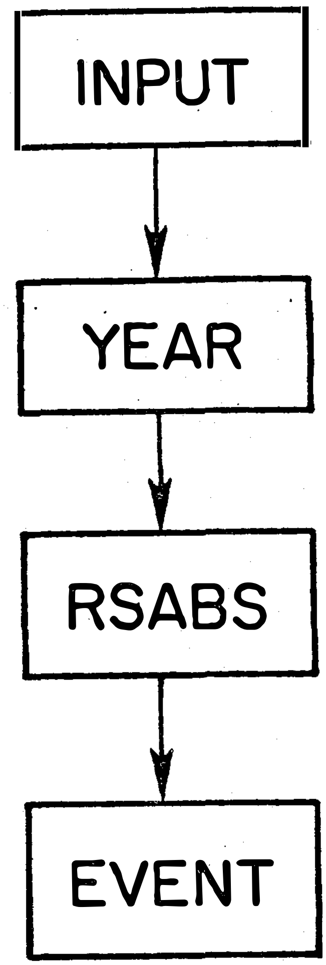

Figure 4) Nest of Calls for RSABS.

A similar situation exists in the individual heliostat performance model but in the cellwise optimization program, YEAR is replaced by RCELL which calls SBDATA which then calls RSABS.

can be handled in a similar way, with nine calls to EVENT and three boundary vectors. The test for remote images works so efficiently that the additional complexity of the code is barely noticeable in CPU time. See conclusions. As the drive and pedestal occupy the lower half of the slot they will rarely cast a shadow on a neighbor or block a reflected beam, thus their presence can be ignored.

4) TEST PHILOSOPHY FOR EVENTS

A neighboring heliostat may or may not cause an event. The call to EVENT is bypassed if the projected image of the neighbor lies outside of the initial boundary vector which represents the maximum extent of the reflecting region in question. This bypass excludes the so called remote images which are easy to detect.

A subroutine listing of RSABS and, starting at line 244, of EVENT is given in appendix B and fortran variables are defined in appendix A. Just inside EVENT the loop 150 goes over the current list of boundary points (K=1, NGON) and tests to see if any points are inside of the image of the neighbor. The first point found to be inside the image sets

IGON = K (I-GO-IN),

and the last inside point sets

IGOT = K (I-GO-OUT).

If no point is inside, then IGON=0 and RETURN occurs. The K=1 point (selected as the highest point of the heliostat) is not tested because it is a fixed point and in RSABS we assume an alt-azimuthal mounting so that K=2 is also fixed.

All other control variables depend on IGON and IGOT. The list of boundary points splits into several parts. See Table 1. In general, the

Table 1) Shows Correspondence between Current and Updated Points

CURRENT POINTS			UPDATED POINTS .		
BU(1)	· 1	1	· BU(1)		
BU(2)	· 2	2	· BU(2)	Unchanged
	· LAST	LAST	· BU(LAST)	
1	· IGON		· SU(1)	
2	· Deleted		· SU(2)	Update
INN	· IGOT		· SU(NEW)	
	· IFRT	KFRT	· BU(IFRT)	
BU(ICORN)	· ICORN	KORN	· BU(ICORN)	Shifted

initial group of boundary points is unchanged. The middle group is replaced by the newly computed points and the final group is unchanged but must be shifted to join the last new point. The initial group must occur. However, no final group occurs if the last current point is inner, so that the new points are final. The following variables refer to the current boundary vector, which does not include the inserted points.

```
LAST=IGON-1         (last of first group)
IGON                (first of deleted group)
IGOT                (last of deleted group)
IFRT=IGOT+1         (first of final group)
ICORN               (last of final group)
```

The following variables are required to construct the updated boundary vector.

```
INN=IGOT-IGON+1     (number of inner points)
NEW ={2 if INN is even
      3 if INN is odd
JCORN=NEW-INN       (change in ICORN)
KFRT=IFRT+JCORN     (first of final group)
KORN=ICORN+JCORN    (last updated point)
     →ICORN
```

The odd-even rule for NEW is valid for rectangular images. For other shapes, a more complicated test is needed to determine NEW. It is important to shift the final group before loading the new points. However, no shift occurs if

JCORN=0, or IGOT=ICORN.

The new points are either corners of the neighboring heliostat image or combinations of components from the neighboring heliostat image and the current boundary vector. In order to select the right combination we have to know whether the first new point belongs to constant U or a constant V segment of the current boundary vector. Let

JNDEX=INDEX+LAST,

so that JNDEX is even if the LAST leads a constant U segment and JNDEX is odd if LAST leads a constant V segment. One further test is needed to decide between the larger and smaller alternatives for a component, so that the new point will go inside of the glass region. This is a relatively fussy construction and if there is any doubt about how it is working one can set

IPSAB=2

in order to output the boundary vector for closer study.

5) INPUT REQUIREMENTS AND GEOMETRY OF SPLIT-RECTANGULAR HELIOSTAT

Figure 1 shows the geometry of the split (or slotted) rectangular heliostat. There are three basic linear dimensions: UMIR, VMIR, and SLOT as shown in the figure. According to this model, the reflecting area of the heliostat, is given by

AHELI=VMIR * (UMIR-SLOT),

however, various minor features such as clipped corners and gaps between the segments may cause AHELI to differ from the actual reflecting area, which is always called HGLASS.

Shading and blocking also depends on the geometry of the heliostat neighborhood. For a radial stagger neighborhood, the radial and azimuthal spacing parameters are given by

R=XGRND*DMIR

Z=YGRND*DMIR

where XGRND and YGRND are determined by the optimization program and are independent of DMIR which represents heliostat size. We define

$$DMIR=[\tfrac{1}{2}(UMIR^2+VMIR^2)]^{\frac{1}{2}}$$

so that for a square heliostat

$$DMIR=UMIR=VMIR.$$

DMIR is a convenient scale, it is not simply related to the diagonal. The fraction of ground coverage is given by

$$FGRND=.5*RHELI/(XGRND*YGRND)$$

where

$$RHELI=HGLASS/DMIR^2.$$

Notice that DMIR cancels out of FGRND, and HGLASS introduces the true heliostat area. These identities remain true for the split rectangular heliostat, however, the value of the mechanical limit (in DMIR units) may have to be adjusted for a new geometry.

The fraction of shading and blocking is called FMIRX. A look at the listings in appendices A and B reveals that FMIRX is transmitted to the external program by common group GRP6. In previous shading and blocking codes, the components of the boundary vectors were in meters, so that

$$FMIRX= .5* POLYA/AHELI,$$

where POLYA is twice the area of the region enclosed by the boundary vector. In this case POLYA and AHELI have the dimensions meters squared. However, in RSABS the components of the boundary vectors are in dimensionless units of DMIR. Consequently, we have

$$FMIRX=FINVER*(POLYA_1+POLYA_2)$$

where

$$FINVER=.250*DMIR^2/AHELI$$

and the two boundary vectors give a total useful area of $POLYA_1+POLYA_2$. (There are two factors of .5: one factor from the area of triangles which is omitted from POLYA; the other factor is a normalization for the two boundary vectors of the slotted heliostat).

6) THE DOMINANT ERROR

The assumptions which have been discussed up to this point lead to very small errors in the shading and blocking calculations. However, several problems relating to heliostat guidance must be considered. The effect of ignoring heliostat guidance errors is very small, but the effect of assuming parallel neighbors leads to errors of several percent in the fraction of useful area, FMIRX.

Guidance errors are typically about 3 milliradians. The resulting errors in FMIRX are small and will average to zero in a reasonable time period. Furthermore, the errors in FMIRX are not strongly correlated with receiver flux spillage so that the assumption of perfect guidance appears to be justified for FMIRX.

The projection of a neighboring heliostat onto the plane of the central heliostat depends on the plane of the neighbor, as well as the plane of the central heliostat. For simplicity we assume that the plane of the neighbors is parallel to the central heliostat. Consequently, the shape of the images is similar to the shape of the central heliostat and the test for events is greatly simplified. Actually, the planes are not parallel due to the independent tracking of the heliostats and the independence of their guidance errors. Assuming perfect tracking, the angle between the planes of adjacent heliostats is roughly equal to the azimuthal spacing parameter divided by the slant range to the receiver. For the Barstow pilot plant, this angle ranges from .035 radians to .148 radians. The worst case error in FMIRX is given by

$$\delta \text{FMIRX} \cong \pm \tfrac{1}{2}\, \varepsilon \tan i$$

where i is the angle of incidence on the heliostat and ε is the angle between adjacent heliostat planes. If $\varepsilon = .148$ and $\tan i = 1.0$, then

$\delta FMIRX \cong 7\%$,

but this will be a rare case. Unfortunately we do not expect these errors to cancel out in a sum over heliostats since all neighbors tilt toward the central heliostat (simulating a parabola). Hence, a correct calculation is desirable. A glance at figure 5, shows that a correct calculation turns the images into parallelograms, so that a more sophisticated approach to the code would be required.

The exact projections are available in the HELIOS program from SANDIA/ Albuquerque (Ref. 3). However, in that code the overlap of quadrilaterals is handled for single events, instead of providing a multi-event processor as is done here. HELIOS is very accurate, but CPU time requirements become troublesome for large numbers of heliostats.

7) SUMMARY AND COMPARISON

The RSABS shading and blocking code derives from the following assumptions

1) The effective fraction is defined by a parallel projection from the center of the solar disc. Umbra and penumbra effects are ignored. The effective fraction refers to total redirected sunlight.
2) The heliostat geometry is split-rectangular. There are two boundary vectors. This technique can be reduced to the simple rectangular case or extended to the U and the H shapes.
3) Any mounting system is allowed but alt-azimuthal is less troublesome because it cannot produce split regions. For other mounting system splits can occur, and the lower region will be thrown away. This occurs very rarely.
4) The effect of guidance errors is negligible, and therefore is ignored in the calculation of FMIRX.

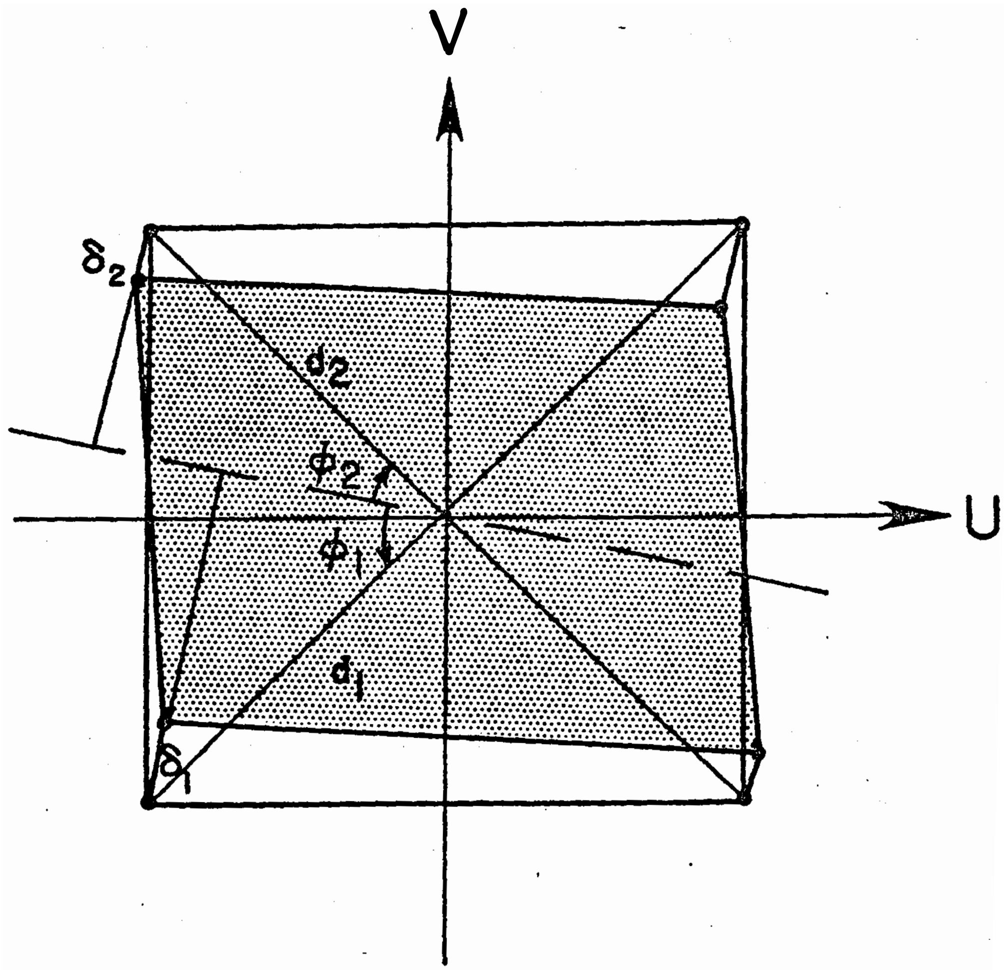

Figure 5) Image of a Non-Parallel Neighbor.

Dashed line shows intersection of heliostat plane with plane of neighbor. Dotted region is parallelgogram. For a rectangular heliostats $d_1 = d_2$ but the image becomes a parallelogram.

$$\delta_1 = \varepsilon d_1 \cos\phi_1 \tan i$$

where ε is angle between the plane of the neighbor and the plane of the central heliostat and i is angle of incidence for the heliostat.

5) The effect of independent tracking, with the consequence of non-coplanar neighbors, is ignored. This approximation gives the dominant error. This error would be insignificant for a commercial size system, but in the smaller Barstow plant it may cause significant errors in the performance of selected heliostats having large incidence angles.

6) Near the singular point in the collector field, the azimuthal orientation of neighboring heliostats varies greatly as the heliostats slew in azimuth to track the sun. However, in this region the heliostats are horizontal, so no shading or blocking occurs. In all other parts of the collector field, the edges of neighboring heliostats are nearly parallel so our current processor is nearly correct.

7) RSABS is a faster subroutine than its predecessors, NGON, etc., in spite of the complexity introduced by the slot.

8) The effect of the slot variable is shown in Table 3. Slotting is not favored from a shading and blocking viewpoint since the deleted part of the heliostat is usually effective. However it may be necessary if there is a valid requirement for mirror inversion (safety or dust aversion).

TABLE 3
Effect of Slot Size Using RSABS
(Equinox Afternoon)

Hours Past Solar Noon	Solar Elev. (deg.)	AV.FMIRX SLOT Width 0.0 m.	AV.FMIRX SLOT Width 1.0 m.	MAX.DEV. of FMIRX
0.000	56.3	.988	.986	-.004
.874	54.2	.990	.988	-.004
1.749	48.4	.993	.992	-.004
2.632	40.3	.996	.996	
3.498	30.8	.989	.988	-.006
4.372	20.6	.945	.948	+.057
5.246	10.0	.817	.852	+.112
"	"	.799*		

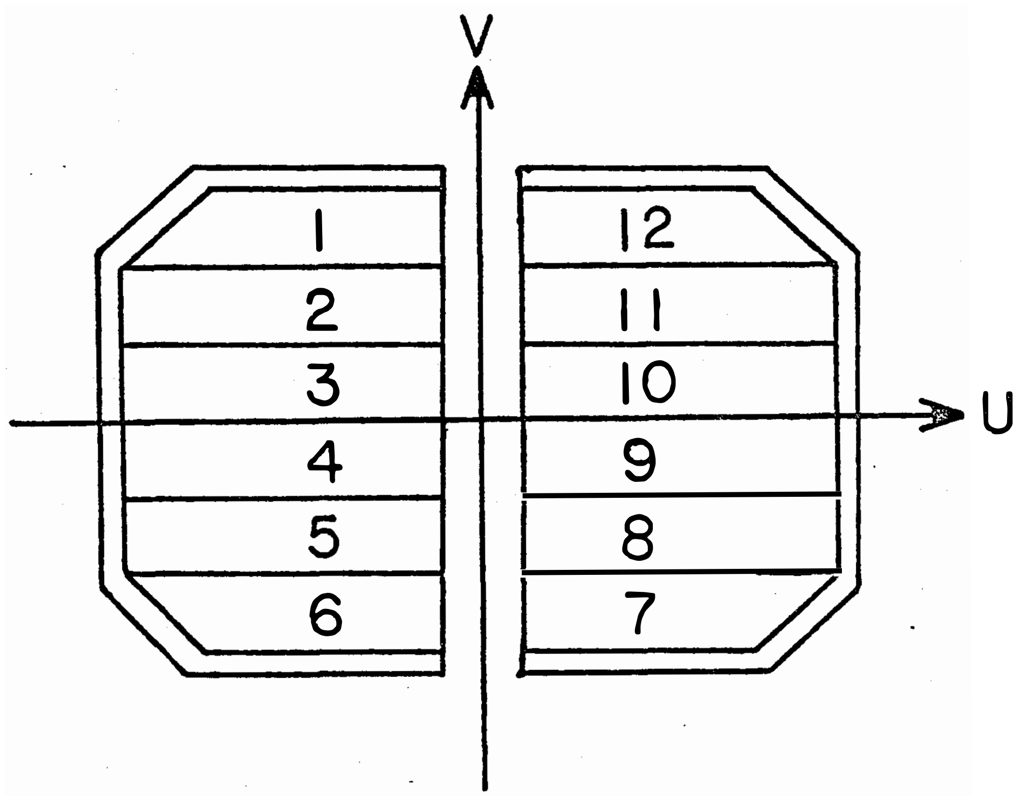

Figure 6) Realistic Heliostat Geometry

There are 12 panels, four clipped corners and 10 gaps between panels in addition to the central slot. A nonreflecting margin is indicated in black. The margin contributes to shading and blocking loses and must be included in the heliostat profile. It is reasonable to consider the margin and the gaps as a default FMIRX < 1 for no shading and blocking. However, this is not now in the code.

This data describes a heliostat which is 10 meters square. The surround field has a radius of 120 heliostat widths, the receiver is 12 heliostat width high and located 1/3 diameter from the south boundary.

*At 10° degrees of elevation, more neighbors are required to calculate FMIRX correctly. The second value is obtained using 24 neighbors instead of 8, as previously. Significant overlap of shading events has been processed at these low sun elevations. The additional neighbors have no effect at the higher elevations in this table.

8) RECOMMENDATIONS

A recent study of shading and blocking using the multievent processor (Ref. 4) shows that practically no over-lapping of events occurs for optimized fields. Hence, the need of an event processer is less than originally expected. Hence, it is practical to consider the exact guidance of the neighbors and process the quadrilateral events as in HELIOS. This would not apply to low sun situations where significant overlap of shading events occurs. For low sun, an accurate treatment requires the more difficult boolean processor with topological complexities but, fortunately, these cases do not contribute significantly to the optimization problem.

Figure 6 shows one of the realistic heliostat geometries. A partially filled slot should also be considered. The geometry of this heliostat is modeled separately for imaging, and for shading and blocking. For imaging the true area of glass is obtained by suming over the reflective areas of the panels. For shading and blocking the outer profile is required which must include the margins as shown in Figure 6. It may also include the corners for convenience. We recommend that the margin effect should be included in the model since it relates to both cost and performance. It would then be possible to do trade studies with and without margins.

REFERENCES

(1) Shading and Blocking Geometry for Rectangular Mirrors, Technical Addendum 4, p. 239, UH/MDAC Progress Report No. 1 "Solar Thermal Power Systems Based on Optical Transmission" NSF/RANN/SE/GI-39456/PR/73/4.

(2) Generalized Shading and Blocking Programs, Technical Addendum 11, p. 204, UH/MDAC Final Report "Solar Thermal Power Systems Based on Optical Transmission" NSF/RANN/SE/GI-39456/FR/75/3.

(3) C.N. Vittitos, F. Biggs, and R. E. Lighthill, HELIOS: A Computer Program for Modeling the Solar Thermal Test Facility, Sandia Laboratories report SAND76-0346, UC-62 distribution. Second Edition June 1977.

(4) Task 3, System Design Studies for Central Receiver Applications p 55, University of Houston Solar Thermal Advanced Research Center" Annual Technical Progress Report for calendar year 1983" DOE/SF/11557--2 NTIS Document number DE85 010917.

Appendix A) List of Fortran Variables used in the Shading and Blocking subroutine RSABS and its subroutine EVENT.

Appendix B contains a complete listing of RSABS and EVENT. Since RSABS is adapted to operation in the cellwise performance program its common groups may contain some irrelevant variables, which will not be discussed here.

RSABS and EVENT are connected by the common group DIMSS, and INPUT is connected to RSABS by HELI2 which contains the basic linear dimensions for the split rectangular heliostat.

Appendix A) List of Fortran Variables used in RSABS and EVENT

EVENT contains the following variables.

Group	Variable	Purpose
ARG	BU	U component of boundary vector for current side
ARG	BV	V component of boundary vector for current side
DIMSS	CL2	Half width of rectangle in U direction
DIMSS	CW2	Half width of rectangle in V direction
ARG	ICORN	Number of boundary points
	IFRT	Index of first point in final group
ARG	IGON	Index of first inside point
	IGOT	Index of last inside point
DIMSS	INDEX	Index of high point
	INN	Number of inside points
	IS	DO200 loop parameter
	IS1	DO202 loop control for JCORN>0
	IS2	DO202 loop control for JCORN<0
	JCORN	Change in ICORN
	JNDEX	Branch control for loading new points
	K	DO150 loop parameter
	KFRT	Index of IFRT after shift
	LAST	Index of last point in fixed group
	NEW	Number of new points.
ARG	UBC	U component of shading/ blocking (center point)

EVENTS (cont'd)

Group	Variable	Purpose
	UBP	U component of shading or blocking (plus side)
ARG	USH	U component of shift for slot
	USHP	U component of shift for slot
	VBC	V component of shading or blocking (center point)
	VBP	V component of shading/ blocking (plus side)

RSABS contains the follow variables in addition to those listed above for EVENT

Group	Variable	Purpose
	A	Intermediate variables
	B	" "
	C	" "
	D	" "
	AU	U component of standard boundary vector.
	AV	V component of standard boundary vector
	BU	U component of boundary vector for left side
	BV	V component of boundary vector for left side
	CU	U component of boundary vector for right side
	CV	V component of boundary vector for right side
DIMSS	CL	Width of Rectangle in U direction
"	CL2	Half width of Rectangle in U direction
"	CW	Width of Rectangle in V direction
"	CW2	Half width of rectangle in V direction
GRP6	COSI	Cosine of incidence for for heliostat
	COSZZ	Cosine of azimuth around tower
	DHNX	X component of displacement of neighbor
	DHNY	Y component of displacement of neighbor
	DHNZ	Z component of displacement of neighbor
HELIO	DMIR	Heliostat size parameters for layout.
	DPN	Projection of DHN onto UNOR.
	DPNC	=DPN/COSI
	DUN	Projection of DNH onto U axis of Helios
	DVN	Projection of DHN onto V axis of Helios
	DUS	Projection of USUN onto U axis of Helios
	DVS	Projection of USUN onto V axis of Helios

SITE	EGRND	Elevation angle for collector field
	FINVER	Normalization factor for FMIRX
GRP6	FMIRX	Fraction of heliostat which is useful
HELIO	HGLASS	Actual reflecting area of heliostat
	I	DO 101 loop parameters, ect.
	IBLOK	Logical varable to control load of SB
	ICORN1	Number of points in B boundary
	ICORN2	Number of points in C boundary
	IGON	≠ 0 if event occurs
GRP6	IMIR	Index of cell in collector field
DIMSS	INDEX	Index of high point in heliostat
	INIT	Branch control for initial phase
PRINT	IPSAB	Output control variable
	J	Do 101 loop parameters
GRP6	JMIR	Index of cell in collector field
	JMIRP	Index for loading SB array
CELL	KORY	=1 for radial orientation, 2 for N-S.
CELL	LRAY	=1 for cornfied, 2 = for staggered neighbors
CELL	NBOR	Number of neighbors
	NCELI	Number of rows in field arrays
	NCELT	Number of columns in field arrays
HELIO	NGON	Number of sides in heliostat geometry
	NGONP	=NGON+1 for SABF2.
GRPSB	ROTXX	Matrix element for rotation of collector field
	ROTXY	Matrix element for rotation of collector field
	ROTXZ	Matrix element for rotation of collector field
	ROTYX	Matrix element for rotation of collector field
	ROTYY	Matrix element for rotation of collector field
	ROTYZ	Matrix element for rotation of collector field
	SB	field array to output neighbors causing events
	SBLOK	intermediate value of SB
	SINZZ	sine of azimuth around tower
	SL	=SLOT/DMIR
	SL2	=SL/2.
HELIZ	SLOT	Size of slot in meters
	SSHAD	intermediate value of SB
	UBC	U component of blocking (center point)
	UC	U component of blocking after shift
	UCNTR	U component of shift for split
	UCNTR2	=2.*UCNTR
GRPSB	UGRNDX	X component of normal to collector field
	URGNDY	Y componet of normal to collector field
	UGRNDZ	Z component of normal to collector field

HELI2	UMIR	U component of heliostat width
GRP5	UNORX	X component of normal to heliostat
	UNORY	Y component of normal to heliostat
	UNORZ	Z component of normal to heliostat
	USC	U component of shading (center point)
	USH	U component of shift for events
GRP5	USUNX	X component of direction towards sun
	USUNY	Y component of direction towards sun
	USUNZ	Z component of direction towards sun
GRP3	UUX	X component of U axis of heliostat
	UUY	Y component of U axis of heliostat
	UUZ	Z component of U axis of heliostat
	UVX	X component of V axis of heliostat
	UVY	Y component of V axis of heliostat
	UVZ	Z component of V axis of heliostat
	VBC	V component of blocking (center point)
	VC	=VBC or VSC
HELI2	VMIR	V component of heliostat width
	VSC	V component of shading (center point)
	WDX	=XGRND(IMIR, JMIR)
	WDY	=YGRND(IMIR, JMIR)
	XA	X component of cell centers
	XAA	X component of intermediate variables
	XCM	X component of data for cornfield neighborhoods
	XGRND	Radial spacing parameters for neighbors
	XS	X component of spacing for neighbors
	XSM	X component of data locating staggering neighborhoods
GRPSB	XTPHI	X component of subreceiver point
	YA	Y component of cell centers
	YAA	Y component of intermediate variables
	YCM	Y component of data for corn-field neighborhoods
	YGRND	Azimuthal spacing parameters for neighbors
	YS	Y component of spacing for neighbors
	YSM	Y component of data locating staggered neighborhoods
GRPSB	YTPHI	Y component of subreceiver point
	ZB	Azimuthal angle of cell center with respect to subreceiver point

Appendix B) List of Code for RSABS and EVENT (line 244 ff)

```
1*#RUN *=; HSELCG1496/XY/R-RSABS (BCD,NOGO,OPTZ)
2       SUBROUTINE SAB3(NCELI,NCELJ,NCELJ4,SB,XGRND,YGRND,XA,YA)
3C
4C       * * * * * * SLOTED RECTANGULAR HELIOSTATS * * * * * *
5C
6C      SHADING AND BLOCKING FOR CELL MODEL OF LARGE SYSTEM
7C         INPUTS BOUNDARY OF HELIOSTAT AND ARRANGMENT OF ITS NEIGHBORS
8C         OUTPUTS BOUNDARY OF LUMINOUS REGION AND ITS AREA
9C      IPSAB PRINT OPTIONS - 2 GIVES DIAGNOSTIC OUTPUT
10C      PARALLEL PROJECTIONS
11C      INCLINED FIELD OPTION  (ROT**)
12C      VARIABLE LIST OF NEIGHBORS
13C      FOUR DIFFERENT NEIGHBORHOODS AVAILIBLE
14C         KORY = 1 FOR RADIAL ORIENTATION
15C         KORY = 2 FOR NORTH-SOUTH ORIENT.
16C         LRAY = 1 FOR CORNFIELD NEIGHBORHOODS
17C         LRAY = 2 FOR STAGGERED NEIGHBORHOODS
18C      SB FOR OUTPUT OF EVENTS
19C
20C       * * * * * * * F W LIPPS    JUNE 1979 * * * * * * * *
21C
22C
23       INTEGER SB
24       LOGICAL IBLOK,ISHAD
25       COMMON /CELL/ DA,AC,NTOWI,NTOWJ,NBOR,KORY,LRAY,LGEO,DTRIM,
26      &             NORDER
27       COMMON /HELIO/ NGON,IAXIS,RH,WH,DMIR,DMECH,HGLASS,CHL(3)
28       COMMON /HELI2/ UMIR,VMIR,SLOT
29       COMMON /DIMSS/ CL,CW,CL2,CW2,INDEX
30       COMMON /PRINT/IFINT,JDISK,KDISK,LDISK,IPDAY,IPLONG,IFLUX,
31      &              IPSAB,IPMIR,IPANCT,INERGY,TRMF,TRMI
32       COMMON /SITE/ XLAT,ILAT,HS,EGRND,ZGRND
33       COMMON /GRPSB/UGRNDX,UGRNDY,UGRNDZ,XTPHI,YTPHI,
34      &        ROTXX,ROTYX,ROTXY,ROTYY,ROTXZ,ROTYZ
35       COMMON /GRP3/UUX,UUY,UUZ,UVX,UVY,UVZ,URECX,URECY,URECZ
36       COMMON /GRP4/UXX,UXY,UXZ,UYX,UYY,UYZ,RSQ,RSQ2,SINASQ
37       COMMON /GRP5/UNORX,UNORY,UNORZ,USUNX,USUNY,USUNZ,RSTAR
38       COMMON /GRP6/IMIR,JMIR,COSI,SOLARA,FMIRX
39       DIMENSION BU(18),BV(18),CU(18),CV(18),AU(4),AV(4)
40       DIMENSION SB(NCELI,NCELJ4)
41       DIMENSION XA(NCELI),YA(NCELJ)
42       DIMENSION XGRND(NCELI,NCELJ),YGRND(NCELI,NCELJ)
43       DIMENSION XSM(24),YSM(24),XCM(24),YCM(24)
44       DATA XSM/1.0,0.5,0.0,-.5,-1.,-.5,0.0,0.5,2.0,1.5,1.0,0.5,
45      &        0.0,-.5,-1.,-1.5,-2.,-1.5,-1.,-.5,0.0,0.5,1.0,1.5/
46       DATA YSM/0.0,0.5,1.0,0.5,0.0,-.5,-1.,-.5,0.0,0.5,1.0,1.5,
47      &        2.0,1.5,1.0,0.5,0.0,-.5,-1.,-1.5,-2.,-1.5,-1.,-.5/
48       DATA XCM/1.,1.,0.,-1.,-1.,-1.,0.,1.,2.,2.,2.,1.,
49      &        0.,-1.,-2.,-2.,-2.,-2.,-2.,-1.,0.,1.,2.,2./
50       DATA YCM/0.,1.,1.,1.,0.,-1.,-1.,-1.,0.,1.,2.,2.,
```

```
     &          2.,2.,2.,1.,0.,-1.,-2.,-2.,-2.,-2.,-2.,-1./
      DATA INIT/0/
      IF (INIT.EQ.1) GO TO 103
      INIT = 1
C
      CL=(UMIR-SLOT)/(2.*DMIR)
      CW=VMIR/DMIR
      SL=SLOT/DMIR
      CL2=CL/2.
      CW2=CW/2.
      SL2=SL/2.
      UCNTR2=2.*UCNTR
      FINVER=.250/(CL*CW)
      AU(1)= CL2
      AV(1)= CW2
      AU(2)= -CL2
      AV(2)= CW2
      AU(3)= -CL2
      AV(3)= -CW2
      AU(4)= CL2
      AV(4)= -CW2
      IF(NGON.NE.4)STOP
      NGONP = 5
      PHIM=3.1415965/2.
C
  103 IF(KORY.EQ.2)GO TO 104
C
C  ** FOR RADIAL ORIENTATION **
      ZB = 0
      IF(ABS(YTPHI-YA(JMIR)).GT.1E-6.OR.ABS(XTPHI-XA(IMIR)).GT.1E-6)
     &     ZB = ATAN2(YTPHI-YA(JMIR),XTPHI-XA(IMIR))
      COSZZ = COS(ZB)
      SINZZ = SIN(ZB)
C
  104 WDX = XGRND(IMIR,JMIR)
      WDY = YGRND(IMIR,JMIR)
C
C     INDEX LOADS HIGHEST VERTEX INTO FIRST CORNER OF(BU,BV)
C     CA- VERSION WORKS FOR ALL MOUNTING SYSTEMS WITH K=2,ICORN
C     INDEX = 0 OR 2 FOR ALT-AZIMUTHAL MOUNTINGS WITH K=3;ICORN
C
      INDEX = 0
      A = UGRNDX*UNORX + UGRNDY*UNORY + UGRNDZ*UNORZ
      IF(A .GT. .9999) RETURN
      B = UGRNDX*UVX + UGRNDY*UVY + UGRNDZ*UVZ
      IF(B.LT.0.)INDEX = 2
CA-    C = UGRNDX*UUX + UGRNDY*UUY + UGRNDZ*UUZ
CA-    PHIH = ATAN2(B/UMIR,C/VMIR)/PHIM
CA-     IF(PHIH.LT.0.0) PHIH =  4.0 + PHIH
```

```
CA-     INDEX = INT(PHIH)
        DO 101 I=1,NGON
        J = I+INDEX
        IF(J.GT.NGON)J=J-NGON
        BU(I) = AU(J) - UCNTR
        BV(I) = AV(J)
        CU(I) = AU(J) + UCNTR
        CV(I) = AV(J)
   101  CONTINUE
        DO 102 I=NGONP,18
        BU(I) = 0.
        BV(I) = 0.
        CU(I) = 0.
   102  CV(I) = 0.
        ICORN1 = NGON
        ICORN2 = NGON
        DUS = UUX*USUNX+UUY*USUNY+UUZ*USUNZ
        DVS = UVX*USUNX+UVY*USUNY+UVZ*USUNZ
C
C            SCAN NEIGHBORS FOR SHADING AND BLOCKING OF BOTH SIDES
C
        DO 115 I=1,NBOR
        IF(LRAY.EQ.1) GO TO 116
C
C   ** STAGGERED NEIGHBORS  **
        XS = XSM(I) * WDX
        YS = YSM(I) * WDY
        GO TO 117
C
C   ** CORNFIELD NEIGHBORS **
   116  XS = XCM(I) * WDX
        YS = YCM(I) * WDY
C
   117  IF(KORY.EQ.2) GO TO 118
C
C   ** FOR RADIAL FIELDS **
        XAA = COSZZ*XS - SINZZ*YS
        YAA = SINZZ*XS + COSZZ*YS
        GO TO 119
C
C       FOR N-S FIELDS
   118  XAA = XS
        YAA = YS
C
   119  IF(ABS(EGRND).LT..0001)GO TO 120
C
C   ** FOR INCLINED FIELDS **
        DHNX = XAA*ROTXX + YAA*ROTYX
        DHNY = XAA*ROTXY + YAA*ROTYY
        DHNZ = XAA*ROTXZ + YAA*ROTYZ
```

```
      GO TO 121
C
C     FOR LEVEL FIELDS
  120 DHNX = XAA
      DHNY = YAA
      DHNZ = 0.
C
  121 DPN = UNORX*DHNX+UNORY*DHNY+UNORZ*DHNZ
      IF(DPN.LT..0001)GO TO 115
      DUN = UUX*DHNX+UUY*DHNY+UUZ*DHNZ
      DVN = UVX*DHNX+UVY*DHNY+UVZ*DHNZ
      DPNC= DPN/COSI
      UBC = DUN + DUS*DPNC
      VBC = DVN + DVS*DPNC
      USC = DUN - DUS*DPNC
      VSC = DVN - DVS*DPNC
C
      UC = UBC - UCNTR
      VC = VBC
      IBLOK = .F.
      IF(ABS(VBC).GT.CW)GO TO 122
      IF(ABS(UBC).GT.CL)GO TO 131
      CALL EVENT(UBC,UC,VC,BU,BV,ICORN1,IGON)
      IF(IGON.NE.0)IBLOK = .T.
  131 USH = UBC - UCNTR2
      IF(ABS(USH).GT.CL)GO TO 132
      CALL EVENT(USH,UC,VC,CU,CV,ICORN2,IGON)
      IF(IGON.NE.0)IBLOK = .T.
  132 UC = UBC + UCNTR
      USH = UBC + UCNTR2
      IF(ABS(USH).GT.CL)GO TO 133
      CALL EVENT(USH,UC,VC,BU,BV,ICORN1,IGON)
      IF(IGON.NE.0)IBLOK = .T.
  133 IF(ABS(UBC).GT.CL)GO TO 122
      CALL EVENT(UBC,UC,VC,CU,CV,ICORN2,IGON)
      IF(IGON.NE.0)IBLOK = .T.
C
  122 UC = USC - UCNTR
      VC = VSC
      ISHAD = .F.
      IF(ABS(VSC).GT.CW)GO TO 123
      IF(ABS(USC).GT.CL)GO TO 135
      CALL EVENT(USC,UC,VC,BU,BV,ICORN1,IGON)
      IF(IGON.NE.0)ISHAD = .T.
  135 USH = USC - UCNTR2
      IF(ABS(USH).GT.CL)GO TO 136
      CALL EVENT(USH,UC,VC,CU,CV,ICORN2,IGON)
      IF(IGON.NE.0)ISHAD = .T.
  136 UC = USC + UCNTR
```

```
      USH = USC + UCNTR2
      IF(ABS(USH).GT.CL)GO TO 137
      CALL EVENT(USH,UC,VC,BU,BV,ICORN1,IGON)
      IF(IGON.NE.0)ISHAD = .T.
  137 IF(ABS(USC).GT.CL)GO TO 123
      CALL EVENT(USC,UC,VC,CU,CV,ICORN2,IGON)
      IF(IGON.NE.0)ISHAD = .T.
  123 CONTINUE
C
      IF(.NOT.IBLOK)GO TO 124
      JMIRP = JMIR + NCELJ
      SBLOK = SB(IMIR,JMIRP)
      IF(SBLOK.NE.0)JMIRP = JMIRP + 2*NCELJ
      SB(IMIR,JMIRP)=I
  124 IF(.NOT.ISHAD)GO TO 115
      JMIRP = JMIR
      SSHAD = SB(IMIR,JMIRP)
      IF(SSHAD.NE.0)JMIRP = JMIRP + 2*NCELJ
      SB(IMIR,JMIRP)=I
C
  115 CONTINUE
      FMIRX=0.
      DO 400 I=3,ICORN1
      A=BU(I-1)-BU(1)
      B=BV(I-1)-BV(1)
      C=BU(I)-BU(1)
      D=BV(I)-BV(1)
      FMIRX=FMIRX + A*D-B*C
  400 CONTINUE
      DO 401 I=3,ICORN2
      A=CU(I-1)-CU(1)
      B=CV(I-1)-CV(1)
      C=CU(I)-CU(1)
      D=CV(I)-CV(1)
      FMIRX=FMIRX + A*D-B*C
  401 CONTINUE
      FMIRX = FINVER * FMIRX
      IF(IPSAB.EQ.2)WRITE(6,29)FMIRX,IMIR,BU,JMIR,BV
     &              ,IMIR,CU,JMIR,CV
   29 FORMAT(1X/,5X,F7.3/,(1X,I4,18F7.3))
      RETURN
      END
C
C
      SUBROUTINE EVENT(USH,UBC,VBC,BU,BV,ICORN,IGON)
      COMMON /DIMSS/ CL,CW,CL2,CW2,INDEX
      DIMENSION BU(18),BV(18)
C
C     TEST TO SET CONTROL VARIABLES
      IGON = 0
      DO 150 K=3,ICORN
```

```
      IF(ABS(UBC-BU(K)).GT.CL2.OR.ABS(VBC-BV(K)).GT.CW2)GOTO 150
      IF(IGON.EQ.0)IGON = K
      IGOT = K
  150 CONTINUE
       IF(IGON.EQ.0)RETURN
       USHP= USH + CL2
       UBP = UBC + CL2
       VBP = VBC + CW2
       INN = IGOT - IGON + 1
       NEW = 2
       IF(2*(INN/2).NE.INN)NEW = 3
       JCORN = NEW - INN
       IFRT = IGOT + 1
       LAST = IGON - 1
       KFRT = IFRT + JCORN
       IF(IGOT.EQ.ICORN)KFRT = 1
C
C     TO SHIFT TAIL OF BOUNDARY
      IF(KFRT.EQ.1)GO TO 203
      IF(JCORN.EQ.0)GO TO 203
      IF(JCORN.GT.0)GO TO 201
C
C     LIST DECREASES IF JCORN < 0
      DO 200 IS = IFRT,ICORN
      BU(IS+JCORN) = BU(IS)
  200 BV(IS+JCORN) = BV(IS)
      GO TO 203
C
C     LIST INCREASES IF JCORN > 0
  201 IS1 = ICORN + IFRT
      IS2 = IS1   + JCORN
      DO 202 IS = IFRT,ICORN
      BU(IS2-IS) = BU(IS1-IS)
  202 BV(IS2-IS) = BV(IS1-IS)
C
C     TO CALC AND LOAD NEW POINTS
  203 ICORN = ICORN + JCORN
      IF(ABS(USHP).GT.CL2)UBP = UBC - CL2
      IF(ABS(VBP ).GT.CW2)VBP = VBC - CW2
      JNDEX = INDEX + LAST
      IF(2*(JNDEX/2).EQ.JNDEX)GO TO 204
C
C     LAST POINT LEADS CONST V SEGMENT
      BU(IGON) = UBP
      BV(IGON) = BV(LAST)
      BU(IGON+1) = UBP
      BV(IGON+1) = BV(KFRT)
      IF(NEW.NE.3)GO TO 205
      BV(IGON+1) = VBP
      BU(IGON+2) = BU(KFRT)
```

```
      BV(IGON+2) = VBP
      GO TO 205
C
C     LAST POINT LEADS CONST U SEGMENT
  204 BU(IGON) = BU(LAST)
      BV(IGON) = VBP
      BU(IGON+1) = BU(KFRT)
      BV(IGON+1) = VBP
      IF(NEW.NE.3)GO TO 205
      BU(IGON+1) = UBP
      BU(IGON+2) = UBP
      BV(IGON+2) = BV(KFRT)
C
  205 CONTINUE
      WRITE(6,29)BU,BV
     &              ,INN,NEW,JCORN,ICORN,IFRT,LAST
   29 FORMAT(5X,18F7.3/,5X,18F7.3/,10X,6I4)
      RETURN
      END
C
```

Document Control Page	1. SERI Report No.	2. NTIS Accession No.	3. Recipient's Accession No.
	SERI/TP-253-4281	DE91002146	

4. Title and Subtitle Computer Simulation of Shading and Blocking: Discussion of Accuracy and Recommendations	**5. Publication Date** April 1992
	6.
7. Author(s) F. Lipps	**8. Performing Organization Rept. No.**
9. Performing Organization Name and Address University of Houston 4800 Calhoun Houston, Texas 77004	**10. Project/Task/Work Unit No.**
	11. Contract (C) or Grant (G) No. (C) XX-4-04006-1 (G)
12. Sponsoring Organization Name and Address Solar Energy Research Institute 1617 Cole Boulevard Golden, Colorado 80401-3393	**13. Type of Report & Period Covered** Technical report
	14.

15. Supplementary Notes
SERI Technical Monitor: R. Gerald Nix, (303)231-1757

16. Abstract (Limit: 200 words)
A field of heliostats suffers losses caused by shading and blocking by neighboring heliostats. The complex geometry of multiple shading and blocking events suggests that a processing code is needed to update the boundary vector for each shading or blocking event. A new version, RSABS, (programmer's manual included) simulates the split-rectangular heliostat. Researchers concluded that the dominant error for the given heliostat geometry is caused by the departure from planarity of the neighboring heliostats. It is recommended that a version of the heliostat simulation be modified to include losses due to nonreflective structural margins, if they occur. Heliostat neighbors should be given true guidance rather than assumed to be parallel, and the resulting nonidentical quadrilateral images should be processed, as in HELIOS, by ignoring overlapping events, rare in optimized fields.

17. Document Analysis

a. Descriptors
Computer simulation; shading; blocking; heliostats; split-rectangular heliostat

b. Identifiers/Open-Ended Terms

c. UC Categories
233 & 234

18. Availability Statement National Technical Information Service U.S. Department of Commerce 5285 Port Royal Road Springfield, VA 22161	**19. No. of Pages** 44
	20. Price A03

Form No. 0069E (6-30-87)