NASA TECHNICAL NOTE

NASA TN D-5910

c. 1

NASA TN D-5910

A METHOD FOR THE THERMAL ANALYSIS OF SPACECRAFT, INCLUDING ALL MULTIPLE REFLECTIONS AND SHADING AMONG DIFFUSE, GRAY SURFACES

by Doyle P. Swofford

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JULY 1970

1. Report No. NASA TN D-5910	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A METHOD FOR THE THERMAL ANALYSIS OF SPACECRAFT, INCLUDING ALL MULTIPLE REFLECTIONS AND SHADING AMONG DIFFUSE, GRAY SURFACES		5. Report Date July 1970
		6. Performing Organization Code
7. Author(s) Doyle P. Swofford		8. Performing Organization Report No. L-6063
		10. Work Unit No. 124-09-24-03
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A new method which uses finite surface elements has been developed and used to calculate temperature histories for spacecraft of arbitrary physical geometry. This method assumes gray surfaces with diffuse reflection and radiation properties and accounts for all multiple reflections. The analysis is performed in terms of the Cartesian coordinates of the four corners of each plane quadrilateral element. Shading, or optical blocking, is accounted for in the radiant heat exchange between each pair of surfaces and in the thermal fluxes from external sources which are incident on each surface. This method has been programed for a digital computer and the program can be used as is for the thermal analysis of any spacecraft whose surfaces may be approximated by planar quadrilateral elements which obey Lambert's cosine law. A listing of the program and its auxiliary programs is included in the report. An example problem of a complex, multiwinged earth-orbiting satellite is also presented.

17. Key Words (Suggested by Author(s)) Spacecraft thermal analysis; Multiple diffuse reflections; Digital computer programs	18. Distribution Statement Unclassified – Unlimited		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 63	22. Price* $3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

A METHOD FOR THE THERMAL ANALYSIS OF SPACECRAFT, INCLUDING ALL MULTIPLE REFLECTIONS AND SHADING AMONG DIFFUSE, GRAY SURFACES

By Doyle P. Swofford
Langley Research Center

SUMMARY

A new method which uses finite surface elements has been developed and used to calculate temperature histories for spacecraft of arbitrary physical geometry. This method assumes gray surfaces with diffuse reflection and radiation properties and accounts for all multiple reflections. The analysis is performed in terms of the Cartesian coordinates of the four corners of each plane quadrilateral element. Shading, or optical blocking, is accounted for in the radiant heat exchange between each pair of surfaces and in the thermal fluxes from external sources which are incident on each surface.

This method has been programed for a digital computer and the program can be used as is for the thermal analysis of any spacecraft whose surfaces may be approximated by planar quadrilateral elements which obey Lambert's cosine law. A listing of the program and its auxiliary programs is included in the report.

An example problem of a complex, multiwinged earth-orbiting satellite is also presented.

INTRODUCTION

For a spacecraft which has extended members with large surface areas or has large enclosed spaces containing personnel or sensitive electronic equipment, shading and multiple reflections among its surfaces can be critical to its thermal design.

The projected areas of the external surfaces of the spacecraft and the shape factors between all surfaces show the effect of shading of spacecraft surfaces by one another. The usual method of obtaining projected areas is by photographs of a model of the spacecraft. Shape factors may also be obtained by a photographic method (ref. 1, pp. 399-402).

For complex spacecraft in which temperatures are required at literally hundreds of nodal points, computer calculation methods are the only practical approach because of the large number of shape factors and projected areas that must be obtained.

When the computer program in this report was written, other programs were already available which calculate projected areas and shape factors for configurations with shading and which account for multiple diffuse reflections of thermal radiation as well. (See ref. 2.) However, since these programs require much time to prepare and run, a more simple, yet flexible program, such as the one in the present investigation, was desired for development use in a wide variety of spacecraft. The thermal design may be fairly well fixed by use of the simpler program and then confirmed by use of a more sophisticated program.

The treatment of the multiple reflections is made tractable by the usual approximation that all the surfaces reflect and emit radiation diffusely (i.e., according to Lambert's cosine law). Once the shape factors and projected areas are calculated, the quadrature solution to this diffuse problem (developed by Gebhart, ref. 3) can be used. The basis of the approach used here is to divide the irregular quadrilateral surfaces of the spacecraft into grids by the use of two-dimensional coordinate systems embedded in each separate surface. The division allows the required integrations to be carried out in two-dimensional spaces. A shading test for each grid element is made by determining whether a line from the source point to the grid point is intercepted by any other surface belonging to the spacecraft.

Calculated temperatures resulting from the shape factors and projected areas can be steady state or time dependent. If radiant heat transfer is predominant over conductive heat transfer, equilibrium temperatures may be found by solving simultaneous equations for the fourth power of the temperature of each surface. The nonequilibrium case (varying external and internal heat loads) is solved by integrating the time derivatives of the temperatures. In this case heat conduction is included in the calculation.

SYMBOLS

A	area
$[A]$	diagonal matrix of the areas of a group of surfaces
A_{proj}	projected area of a surface
$[a]$	diagonal matrix of surface absorptivities
B	thermal flux incident on one surface from others
$\{B\}$	one-dimensional array of thermal flux incident on each surface from the other surfaces

c	specific heat of a material
D_{ij}	radiation coefficient between two identically shaped, closely spaced parallel plates, $\dfrac{\sigma}{\frac{1}{e_i} + \frac{1}{e_j} - 1}$
$[D]$	matrix of the radiation coefficients D_{ij}
e	hemispherical emissivity of a surface
$[e]$	diagonal matrix of the emissivities of a group of surfaces
F_{ij}	shape factor between surface i and surface j
$[F]$	square matrix of shape factors between pairs of a group of surfaces
$f_{a,i}$	shape factor of a spacecraft surface i for the albedo flux from a planet surface
$f_{p,i}$	shape factor of a spacecraft surface i for planet-emitted radiation
$[G]$	square matrix which yields $\{B\}$ when it operates upon $\{\epsilon\}$ (see eq. (6))
H	sum of the thermal flux emitted by a surface and the flux reflected by it
$\{H\}$	one-dimensional array of total thermal flux moving away from each surface
h_{ij}	conduction coefficient between nodes i and j
$[h]$	square matrix of conduction coefficients between pairs of a group of nodes
$[I]$	identity matrix
I_{sh}	shading indicator $\left(I_{sh} = 0\text{: shaded by an intervening member; } I_{sh} = 1\text{: not shaded}\right)$
K	sunlight indicator ($K = 1$: sunlit; $K = 0$: shaded by planet)
m	mass

$\bar{n}$	unit normal vector of a surface
q	net heat flow rate to a node
q_{cond}	net rate of heat flow to a node by conduction
q_{int}	rate of internal heat generation in a node
q_{rad}	net rate of heat flow between close parallel nodes by radiation
$\bar{r}$	position vector of a point
$\bar{r}_{ij}$	position vector of point j relative to point i, $\bar{r}_j - \bar{r}_i$
S	solar flux at position of planet or spacecraft
T	absolute temperature, ^{o}K
T_p	equivalent blackbody temperature of a planet, ^{o}K
t	time
$\overline{V}_c$	position vector of the centroid of an elemental area
$\overline{V}_{j,k}$	position vector of the lower left corner of the jkth grid element of a plane quadrilateral
$\overline{V}_p$	unit vector directed toward the center of a planet from an orbiting spacecraft
[W]	diagonal matrix with diagonal $W_i = \sum_j D_{ij}$
z_i	coefficient for heat conduction away from a node, $z_i = \sum_j h_{ij}$
[z]	diagonal matrix of coefficients for heat conduction away from each node
α, β, γ	parameters which each give the ratio of the lengths of two colinear vectors having a common origin

α', β'	abscissa and ordinate of a point in a normalized skewed coordinate system
ϵ	sum of the thermal emission flux from a spacecraft surface and the reflected portion of incident thermal flux arriving at the surface directly from sources external to the spacecraft
$\{\epsilon\}$	one-dimensional array of the fluxes ϵ of the surfaces of a spacecraft
θ_i	angle formed by vectors $\bar{n}_i$ and $\bar{r}_{ij}$
θ_j	angle formed by vectors $\bar{n}_j$ and $\bar{r}_{ij}$
θ_s	angle formed at the center of a planet by lines to the sun and to an orbiting spacecraft
λ_1, λ_2	parameters giving any vector lying in a given plane as a linear combination of two given vectors in the plane
μ	ratio of the projected area of a surface to its total area (unit projected area)
$\{\mu\}$	one-dimensional array of unit projected areas
ρ	reflectivity of a surface
$[\rho]$	diagonal matrix of reflectivities of surfaces
ρ_p	reflectivity, or albedo, of a planet surface for solar radiation
σ	Stefan-Boltzmann constant
Φ_{ji}	total radiative heat flow from surface j which is incident directly upon surface i
$d\phi_{ji}$	heat flux incident upon elemental area dA_i which is received from dA_j
ϕ_a	planet solar albedo flux incident upon a surface of a spacecraft
ϕ_p	planet thermal emission incident on a spacecraft surface

ϕ_{rad} net thermal radiation flux received by a plate i from identical plates j parallel to and near it, $D_{ji}\left(T_j^4 - T_i^4\right)$

ϕ_s solar radiation flux incident upon a spacecraft surface

ψ_{ji} function which is integrated to yield F_{ji}, $-\dfrac{\left(\bar{n}_i \cdot \bar{r}_{ij}\right)\left(\bar{n}_j \cdot \bar{r}_{ij}\right)}{\pi\left(\bar{r}_{ij} \cdot \bar{r}_{ij}\right)^2}$

ω half the angle subtended by a planet from the position of an orbiting spacecraft

Subscripts:

a albedo

p planetary

s solar

t thermal radiation emitted by spacecraft surfaces or planet surface

When an expression with an asterisk affixed as a superscript has a negative value, it is set equal to zero; positive values are unchanged.

A bar over a symbol indicates a vector quantity.

ANALYSIS

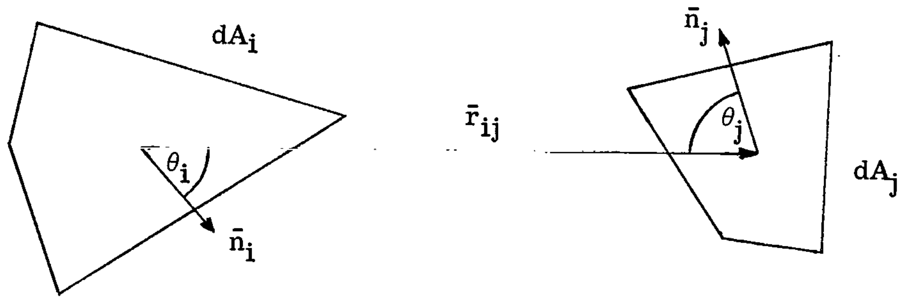
Radiative Heat Balance

The analysis of the heat transfer by diffuse emission and reflection of thermal radiation between one surface and the remainder of a group of surfaces is introduced by first considering just two surfaces i and j exchanging thermal radiation. The two surfaces are assumed to be isothermal.

The heat flux incident upon the elemental area dA_i of surface i which is received from dA_j of surface j is given by

$$d\phi_{ji} = \frac{H_j}{\pi} \frac{dA_j}{r_{ij}^2} \cos\theta_i \cos\theta_j$$

where H_j is the heat flux leaving dA_j, r_{ij} is the distance between dA_i and dA_j, and θ_i and θ_j are the angles made with the normals to dA_i and dA_j, respectively, by the line $\bar{r}_{ij}$ between them as shown in sketch A:



Sketch A

If $\bar{r}_i$ is the position vector of dA_i, and $\bar{r}_j$ that of dA_j, then

$$\cos\,\theta_i = \bar{n}_i \cdot \frac{\bar{r}_j - \bar{r}_i}{\left|\bar{r}_j - \bar{r}_i\right|} = \frac{\bar{n}_i \cdot \bar{r}_{ij}}{\left|\bar{r}_{ij}\right|}$$

and

$$\cos\,\theta_j = n_j \cdot \frac{\bar{r}_i - \bar{r}_j}{\left|\bar{r}_i - \bar{r}_j\right|} = \bar{n}_j \cdot \frac{-\left(\bar{r}_j - \bar{r}_i\right)}{\left|\bar{r}_{ij}\right|} = -\frac{\bar{n}_j \cdot \bar{r}_{ij}}{\left|\bar{r}_{ij}\right|}$$

where $\bar{n}_i$ and $\bar{n}_j$ are unit normal vectors to dA_i and dA_j. Thus,

$$d\phi_{ji} = -\frac{H_j}{\pi}\,\frac{\left(\bar{n}_i \cdot \bar{r}_{ij}\right)\left(\bar{n}_j \cdot \bar{r}_{ij}\right)}{r_{ij}^4}\,dA_j$$

Now, let

$$-\frac{\left(\bar{n}_i \cdot \bar{r}_{ij}\right)\left(\bar{n}_j \cdot \bar{r}_{ij}\right)}{\pi r_{ij}^4} = \psi_{ji} = \psi_{ij}$$

so that

$$d\phi_{ji} = H_j \psi_{ij} \, dA_j$$

If Φ_{ji} is the total heat flow incident upon surface i from surface j, then

$$d^2\Phi_{ji} = d\phi_{ji} \, dA_i = H_j \psi_{ij} \, dA_i \, dA_j$$

and

$$\Phi_{ji} = \int_{A_j} H_j \int_{A_i} \psi_{ij} \, dA_i \, dA_j$$

Assuming H_j constant over surface j gives

$$\Phi_{ji} = H_j \int_{A_j} \int_{A_i} \psi_{ij} \, dA_i \, dA_j = H_j F_{ij} A_i A_j$$

where the shape factor F_{ij} is the average value of ψ_{ij} over the two surfaces. Unlike the conventional shape factor, F_{ij} is not nondimensional and must be multiplied by A_j to yield the conventional quantity f_{ij}. Also, $F_{ji} = F_{ij}$. The average flux incident on surface i from surface j is defined as

$$B_i = \frac{\Phi_{ji}}{A_i} = F_{ij} A_j H_j$$

This is the flux incident on surface i from a single surface j. For more than one surface j, the total flux incident on i is given by the sum over j of the individual contributions:

$$B_i = \sum_j F_{ij} A_j H_j \tag{1}$$

If surface i is concave, B_i includes a contribution from surface i to itself. When B_i and H_i are treated as the ith components of linear arrays $\{B\}$ and $\{H\}$, it follows directly from equation (1) that

$$\{B\} = [F][A]\{H\} \tag{2}$$

where $[F]$ is a square matrix with elements F_{ij}, and $[A]$ is a diagonal matrix whose diagonal A_i is the area of surface i.

The flux H_j leaving surface j is made up of ϵ_j (the emitted flux and the reflected portion of flux incident from sources external to the spacecraft) and the reflected part $\rho_j B_j$ of the incident flux from other surfaces, where ρ_j is the reflectivity of surface j. Thus,

$$H_j = \epsilon_j + \rho_j B_j \tag{3}$$

and ϵ_j includes the reflected part of any external radiation directly incident on j, such as sunlight. In the array form,

$$\{H\} = \{\epsilon\} + [\rho]\{B\} \tag{4}$$

where $[\rho]$ is a diagonal matrix of the reflectivities of the surfaces.

Equations (2) and (4) are two matrix equations in the two arrays $\{B\}$ and $\{H\}$; therefore, unique solutions may be obtained for them. Substituting equation (4) into equation (2) gives

$$\{B\} = [F][A]\{\epsilon\} + [F][A][\rho]\{B\} \tag{5}$$

and, on solving for $\{B\}$,

$$\{B\} = [I - FA\rho]^{-1}[FA]\{\epsilon\} = [G]\{\epsilon\} \tag{6}$$

where $[I]$ is an identity matrix. Since $[G]$ is a function only of the surface properties and geometry, it need only be evaluated once for each configuration.

The net heat fluxes are given by the difference $\{B\} - \{H\}$ between the incident flux and the flux leaving the surfaces. From equation (4),

$$\{B\} - \{H\} = [I - \rho]\{B\} - \{\epsilon\}$$

or

$$\{B - H\} = [a]\{B\} - \{\epsilon\} \tag{7}$$

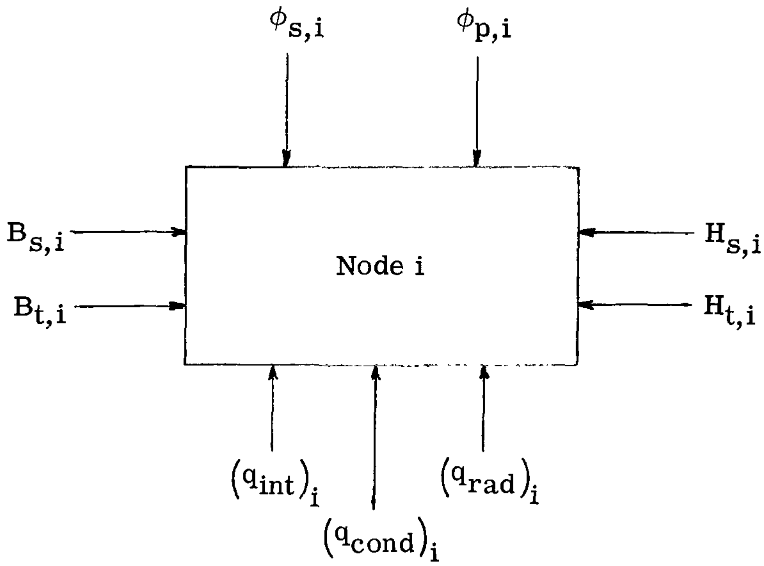
When equation (7) is substituted into equation (6), the following equation is obtained:

$$\{B - H\} = [aG - I]\{\epsilon\} \tag{8}$$

where [a] is a diagonal matrix of the absorptivities of the surfaces and the matrix [G] is given in equation (6).

Equation (8) gives the net flux on each surface due to all emissions and reflections of thermal radiation from all the surfaces. Now, only the fluxes from external radiation sources which are directly incident on the surfaces remain to be accounted for.

In the specific case of a spacecraft immersed in a real environment, there will be two regimes of thermal radiation flux – the principally short-wavelength solar flux and the principally long-wavelength emission from the spacecraft surfaces and from the planet surface. The external thermal radiation sources are direct and planet-reflected (albedo) solar flux and emission from the surface of a planet due to its temperature. Heat conduction as well as radiation will be accounted for. A diagram of the heat transfer for a single node is shown in sketch B:



Sketch B

If q_i is the net rate of heat input to member i, then the heat balance is given by

$$\frac{q_i}{A_i} = \frac{m_i c_i}{A_i} \frac{dT_i}{dt} = (B_s - H_s)_i + (B_t - H_t)_i + (\phi_s + \phi_p)_i + \frac{(q_{int})_i + (q_{cond})_i}{A_i} + (\phi_{rad})_i \tag{9}$$

where

m_i	mass of i
c_i	specific heat of i
T_i	absolute temperature of i
$\phi_{s,i}$	solar flux directly incident on i, both planet reflected and direct from the sun
$\phi_{p,i}$	flux emitted from the planet surface and directly incident on i
$(\phi_{rad})_i$	net radiation flux to i from j, which is parallel to and near i
$(q_{int})_i$	rate of internal heat generation
$(q_{cond})_i$	net rate of conduction to i from other members

Equation (8) gives for the solar spectrum

$$\{B_s - H_s\} = [a_s G_s - I]\{\epsilon_s\} \tag{10}$$

where $\epsilon_{s,i}$ is the reflected part of the incident solar and albedo flux on i, that is, $(\rho_s \phi_s)_i$. For long-wavelength flux

$$\{B_t - H_t\} = \left[eG_t - I \right]\{\epsilon_t\} \tag{11}$$

where $\epsilon_{t,i}$ is made up of reflected planet-emitted flux and the thermal emission of i. Thus

$$\epsilon_{t,i} = \sigma e_i T_i^4 + \rho_{t,i} \phi_{p,i}$$

or in array form,

$$\{\epsilon_t\} = \sigma[e]\left\{T^4\right\} + [I - e]\left\{\phi_p\right\} \tag{12}$$

where σ is the Stefan-Boltzmann constant, $[e]$ is a diagonal matrix of the emissivities of the surfaces, and $\left\{T^4\right\}$ is an array of the fourth powers of the temperatures.

Equation (9) becomes, in the array form,

$$\left\{\frac{q}{A}\right\} = \{B_s - H_s\} + \{B_t - H_t\} + \{\phi_s + \phi_p\} + \left\{\frac{q_{int}}{A}\right\} + \left\{\frac{q_{cond}}{A}\right\} + \{\phi_{rad}\} \tag{13}$$

Substituting equations (10), (11), and (12) into equation (13) gives

$$\left\{\frac{q}{A}\right\} = [a_sG_s - I][\rho_s]\{\phi_s\} + \sigma\left[eG_t - I\right][e]\left\{T^4\right\} + \left[eG_t - I\right][I - e]\left\{\phi_p\right\} + \left\{\phi_s + \phi_p\right\}$$
$$+ \left\{\frac{q_{int}}{A}\right\} + \left\{\frac{q_{cond}}{A}\right\} + \left\{\phi_{rad}\right\} \tag{14}$$

and after simplification,

$$\left\{\frac{q}{A}\right\} = [a_s][I + G_s\rho_s]\{\phi_s\} + [e]\left[I + G_t(I - e)\right]\left\{\phi_p\right\} - \sigma[e][I - G_te]\left\{T^4\right\}$$
$$+ \left\{\frac{q_{int} + q_{cond}}{A}\right\} + \left\{\phi_{rad}\right\} \tag{15}$$

The net rate of heat conduction to member i is given by

$$\left(q_{cond}\right)_i = \sum_j h_{ij}\left(T_j - T_i\right) = \sum_j \left(h_{ij}T_j\right) - \left(\sum_j h_{ij}\right)T_i$$

where h_{ij} is the conduction coefficient between nodes i and j. In array form,

$$\{q_{cond}\} = [h]\{T\} - [z]\{T\} = [h - z]\{T\} \tag{16}$$

where z is a diagonal matrix with $z_i = \sum_j h_{ij}$.

The net radiant heat to i from the identical panel j with which it forms an isolated enclosure is given by

$$\left(\phi_{rad}\right)_i = \sum_j D_{ji}\left(T_j^4 - T_i^4\right)$$

where D_{ji} is the radiation coefficient between j and i, given by

$$D_{ji} = \frac{\sigma}{\frac{1}{e_i} + \frac{1}{e_j} - 1} = D_{ij}$$

Thus,

$$\{\phi_{rad}\} = [D - W]\{T^4\} \tag{17}$$

where W is a diagonal matrix with $W_i = \sum_j D_{ij}$.

Substituting equations (16) and (17) into equation (15) gives the heat balance as

$$\begin{aligned}\left\{\frac{q}{A}\right\} = [a_s][I + G_s\rho_s]\{\phi_s + \phi_a\} + [e]\left[I + G_t(I - e)\right]\{\phi_p\} \\ - \sigma[e]\left[I - G_t e\right]\{T^4\} + [D - W]\{T^4\} \\ + \left[\frac{1}{A}\right][h - z]\{T\} + \left\{\frac{q_{int}}{A}\right\}\end{aligned} \tag{18}$$

Only the external sources $\phi_{s,i}$, $\phi_{a,i}$, and $\phi_{p,i}$ remain to be evaluated. The solar flux $\phi_{s,i}$ incident on i is the simplest, being proportional to the projection of the sunlit area of surface i upon a plane perpendicular to a line to the sun $\left(A_{proj}\right)_i$:

$$A_i\phi_{s,i} = \left(A_{proj}\right)_i KS$$

$$\phi_{si} = \left(\frac{A_{proj}}{A}\right)_i KS = \mu_{s,i} KS$$

Or, in array form,

$$\{\phi_s\} = KS\{\mu_s\} \tag{19}$$

where K has the value 0 if the spacecraft is in the shadow of the planet and the value 1 if not; and S is the local solar flux.

The planet-emitted flux is given by

$$\phi_{p,i} \approx f_{p,i}\sigma T_p^4$$

and, in array form,

$$\left\{\phi_p\right\} = \sigma T_p^4 \left\{f_p\right\} \tag{20}$$

where $f_{p,i}$ is the shape factor of surface i for the planet-emitted radiation, and T_p is the equivalent blackbody temperature of the planet $\left(\sigma T_p^4\right.$ is equal to the average value over the planet of the emitted thermal radiation $\left.\text{flux}\right)$.

The planet solar albedo flux on surface i is given by

$$\phi_{a,i} \approx \rho_p S f_{a,i} \cos^* \theta_s$$

or, in array form,

$$\{\phi_a\} = \rho_p S \cos^* \theta_s \{f_a\} \tag{21}$$

where ρ_p is the mean reflectivity of the planet for sunlight, $f_{a,i}$ is the shape factor of surface i for the planet albedo flux, θ_s is the angle formed at the center of the planet by the lines to the spacecraft and the sun, and $\cos^* \theta_s = \cos \theta_s$ if $\cos \theta_s$ is positive and equal to 0 if $\cos \theta_s$ is negative.

The factors f_p and f_a are difficult to calculate, even without the added complication of partial shading of a surface. An approximation for them – exact when the panel is exposed to the entire portion of the planet surface which is visible from the position of the spacecraft – is

$$f_{a,i} \approx f_{p,i} \approx \left(\bar{n}_i \cdot \bar{V}_p\right)^* \sin^2 \omega \approx \mu_{p,i} \sin^2 \omega$$

Here $\bar{V}_p$ is a unit vector pointing from the spacecraft toward the center of the planet, and ω is the angle included by $\bar{V}_p$ and a tangent from the spacecraft to the planet surface. The factor $\mu_{p,i}$ approximates the effect of shading on $f_{p,i}$ and $f_{a,i}$.

When the relations for the external fluxes (eqs. (19) to (21)) are substituted into equation (18), the heat balance becomes

$$\begin{aligned} \left\{\frac{q}{A}\right\} = {} & KS[a_s][I + G_s \rho_s]\{\mu_s\} + \rho_p S \cos^* \theta_s \sin^2 \omega [a_s][I + G_s \rho_s]\{\mu_p\} \\ & + \sigma T_p^4 \sin^2 \omega [e]\left[I + G_t(I - e)\right]\{\mu_p\} - \sigma[e]\left[I - G_t e\right]\left\{T^4\right\} \\ & + [D - W]\left\{T^4\right\} + \left[\frac{1}{A}\right][h - z]\{T\} + \left\{\frac{q_{int}}{A}\right\} \end{aligned} \tag{22}$$

For the equilibrium case, the net heat flow to each node is zero, and with negligible conduction, equation (22) can be solved for $\{T^4\}$:

$$\sigma\{T^4\} = \left[e\left(I - G_t e\right) + \frac{1}{\sigma}(D - W)\right]^{-1} \left\{KS[a_s][I + G_s \rho_s]\{\mu_s\} + \rho_p S \cos^*\theta_s \sin^2\omega [a_s][I + G_s \rho_s]\{\mu_p\} + \sigma T_p^4 \sin^2\omega [e]\left[I + G_t(I - e)\right]\{\mu_p\} + \left\{\frac{q_{int}}{A}\right\}\right\} \tag{23}$$

For a multifaceted node which is nearly isothermal, although its faces receive different thermal radiation fluxes, the total net heat flow rate is given by

$$q_i = \sum_{j=1}^{n} \left(\frac{q}{A}\right)_j A_j$$

where n is the number of faces of node i. For the nonequilibrium case,

$$\frac{dT_i}{dt} = \frac{A_i}{m_i c_i}\left(\frac{q}{A}\right)_i \tag{24}$$

Equation (24) may be integrated numerically to yield the temperature history, beginning with given initial temperatures.

Listings of the three computer programs which calculate shape factors, projected areas, and temperatures are shown in appendix A. They are written in the FORTRAN IV language. The shape factors and projected areas are run in separate programs because they need to be calculated only once for each configuration. In the main program, surface properties, materials, heat loads, attitude, and flight trajectory can be varied for a fixed configuration without recalculating the shape factors and projected areas. Punched-card outputs from the other two programs supply shape factor and projected-area inputs to the main program.

Computation of Geometric Shape Factors and Projected Areas

Routines were written to enable the digital computer to calculate $[F]$ and $\{\mu\}$ using as input the coordinates of the corners of each plane quadrilateral into which the spacecraft has been divided. The factor which makes the calculation of the projected area of a flat plate complex is the shading by intercepting surfaces.

In order to calculate the unshaded projected area of a plane quadrilateral, it is divided into an $n \times n$ grid. Formulas for the coordinates of the grid points and areas of the grid elements are developed in appendix B. The ratio of the projected area to the total area is given by

$$\mu_i = \frac{1}{A_i}\int_{A_i}\left(\bar{V}_{source} \cdot \bar{n}_i\right)^* I_{sh,i}\, dA_i$$

$$\approx \frac{1}{A_i}\left(\bar{V}_{source} \cdot \bar{n}_i\right)\sum_i I_{sh,i}\, \Delta A_i \tag{25}$$

where $\bar{V}_{source}$ is the unit vector in the direction of the source and

$$I_{sh} = \begin{cases} 0 & \text{for shading} \\ 1 & \text{for no shading} \end{cases}$$

The method of determining shading is developed in appendix C.

In the computation of the shape factor between a pair of plane quadrilaterals, both are divided into $n \times n$ grids as before, and each element of one is paired successively with each element of the other. The shading test is performed by the same scheme as for the projected areas. The shape factor F_{ij} between surfaces i and j is given by the average (taken over both surfaces) of the function ψ_{ij}:

$$F_{ij} = \frac{1}{A_i A_j}\int_{A_i}\int_{A_j} \psi_{ij}\, dA_i\, dA_j$$

$$\approx \frac{1}{A_i A_j}\sum_{i,j} \psi_{ij}\, \Delta A_i\, \Delta A_j$$

$$\approx \frac{1}{A_i A_j}\sum_{i,j} \frac{-\left(\bar{n}_i \cdot \bar{r}_{ij}\right)\left(\bar{n}_j \cdot \bar{r}_{ij}\right)}{\pi r_{ij}^4}\left(I_{sh}\right)_{ij}\, \Delta A_i\, \Delta A_j \tag{26}$$

The position vector $\bar{r}_{ij}$ is given by $\bar{r}_j - \bar{r}_i$, where $\bar{r}_i$ and $\bar{r}_j$ are the position vectors of the centroids of the elemental areas ΔA_i and ΔA_j. The same formulas as for the projected-area routine are used for the area of ΔA_i and its centroid $\bar{r}_i$.

If a surface k is nonplanar and is approximated by a set of N planar surfaces, then μ_k is

$$\mu_k = \frac{\sum_{i=1}^{N} \mu_i A_i}{\sum_{i=1}^{N} A_i} \tag{27}$$

and F_{kj} between the nonplanar surface and any other surface j is

$$F_{kj} = \frac{\sum_{i=1}^{N} A_i F_{ij}}{\sum_{i=1}^{N} A_i} \tag{28}$$

In this investigation the nodal surfaces into which a spacecraft is divided are called panels. A large member may be divided into several pieces to conform more nearly to the assumption of constant temperature over each panel. Surfaces of the spacecraft which can intercept thermal radiation which otherwise would impinge on any of the panels are called shaders. There will be fewer shaders than panels if any of the plane surfaces is subdivided into more than one panel.

EXAMPLE PROBLEM

An example of the application of the computer program is the prediction of temperatures on the proposed Meteoroid Technology Satellite. Figure 1 is a photograph of a model of the spacecraft shown attached to the last stage of its booster, with meteoroid-detector panels deployed. The cubical modules attached to the central structure are experiments for measuring velocities of meteoroids. The octagonal prism at the top is mainly solar cell area. In this example, the spacecraft is spinning about its axis of symmetry.

The projected areas and shape factors are calculated in separate programs to be used later as inputs to the temperature-prediction program. The Cartesian coordinates of panel corners and shader corners are used as input for the projected-area and shape factor programs. Actual areas of the panels are also provided by the projected-area program as input to the main program.

Direct inputs to the main program are absorptivities for solar radiation, hemispherical emissivities, the product of mass and specific heat for each panel, radiation coefficients between back-to-back panels, and orbit and spacecraft-attitude parameters. A circular orbit with an altitude of 300 nautical miles (556 km) and an inclination of 38° to the equator was assumed.

In this example, 104 nodes were used. No conduction is accounted for in this example. Because of the small thicknesses and large surface areas of the nodes, radiation will outweigh conduction by far.

It was necessary to use a very small time increment (0.05 min) in the numerical integration of the temperatures. Some of the nodes consist of a sheet of plastic film 0.00025 inch (0.00064 cm) thick, with a very small thermal capacity; therefore, the solution is unstable for larger time increments.

The temperatures all converged within two or three orbits, without oscillation, to a solution which repeated itself on subsequent orbits. The temperatures compare reasonably with those obtained by a similar heat-transfer computer program which does not take multiple reflections into account.

The temperature history is plotted for seven nodes, whose locations are indicated in figure 1. The temperature plot for the outer face of one of the upper velocity detectors (see fig. 2) follows the heat inputs to it very closely because it has very low thermal inertia. At the far left of the plot, the decline from maximum value of the earth albedo and earth thermal fluxes is noted, while the solar input remains constant. The computed value of the albedo and earth thermal flux declines to zero at about 24 min after perigee, after which the albedo remains zero and the earth thermal flux increases. At about 31 min, the spacecraft enters the earth's shadow. During shadow, the earth thermal flux passes through a maximum and goes to zero again at about 71 min. Sunlight appears again at about 66 min. Albedo flux increases from zero at around 71 min to a maximum about midway through the sunlit period. The times of maximums and minimums in the thermal fluxes incident from external sources and in the temperature do not coincide exactly, since the node receives radiation from other parts of the spacecraft also.

Figures 3 and 4 show the temperature variation over the orbit for a horizontal detector and a vertical detector, respectively. A schematic illustration of the cross section of these meteoroid detectors is given in figure 5, with the bumper shields on each side shown.

CONCLUDING REMARKS

A method has been developed for the thermal analysis of geometrically complicated spacecraft whose surfaces can be approximated by plane quadrilaterals with gray surfaces having diffuse reflection and radiation properties. Optical blocking between surface elements is accounted for automatically. Multiple diffuse reflections are also accounted for. Listings of the computer programs which perform the calculations are included.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., May 1, 1970.

APPENDIX A

COMPUTER PROGRAM LISTINGS

Spacecraft Temperature Program

```
      PROGRAM ORBTEMP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C                                                                       0090
C     HEAT TRANSFER PROGRAM FOR DIFFUSE RADIATION ON MULTIPANELED SPACECRAFT080
C                                                                       0070
      DIMENSION AR1(104),AR2(104),AP1(104),APP1(104),APP2(104),CSH(104),
     1ETAS(104),ETAE(104),FSP1(104,104),FSPP1(104,104),FSPP2(104,104),
     2HCOND(104,104),WATE(104),RRAD(104,104),T(104),XMU(104),TETAS(20),
     3YMU(104),TXMU(104,20),TTXMU(20),DT(104),THELAM(3,3),URTH(3,3),
     4             IPIVOT(104),INDEX(104,2)
      DIMENSION BTHETA(20),BALPHA(20),BBETA(20),RM3PN(3,3),RM1I(3,3),
     1RM3OP(3,3),RM3A(3,3),RM1B(3,3),RM2O(3,3),RM3NTH(3,3),RMOB(3,3),
     2RMEB(3,3),UVEC(3,104),VEC(3)
      DIMENSION              XMUSUN(42, 7,19),TPHIS(36),PHIS(36),AXIS1(
     13,50),AXIS2(3,50),TXMUS(360),VE1(3),VE2(3)
      DIMENSION VE3(3),RMIPHI(3,3),RMPIF(3,3)
      DIMENSION PHIE(104)
 14   FORMAT(12F6.2)
 15   FORMAT(5E16.8)                                                    0930
 16   FORMAT(6F13.3)                                                    0940
 17   FORMAT(2X,10E12.5)                                                0950
 18   FORMAT(//6H TIME=E15.8,18X,22HINTERNAL TEMPERATURE =E15.6/)       0960
 19   FORMAT(1H1,9X,9HTIME(MIN),7X,14HTHETA(RADIANS),5X,15HALTITUDE(MILE 0970
     1S),5X,15HH 0=SHADE,1=SUN,9X,3HAMP,18X,7HETAS(1)//)                0980
 23   FORMAT(1H1,10X,14HVALUES OF UVEC//)                               0990
 24   FORMAT(1H1,10X,14HPROGRAM INPUTS//)                               1000
 25   FORMAT(2X,6E20.6)                                                 1010
 1984 FORMAT(3(2I4,E16.8))
10101 FORMAT(7F11.8)
1856  FORMAT(7I10)
 1905 FORMAT(8E16.8)
      NTIME=1
      READ(5,1856) NPANEL,NFAC,NHCOND,NRRAD,NINSFC,IFTLUP,ISPIN
      DO 3 I=1,NPANEL
      AR2(I)=0.
      APP2(I)=0.
      DO 3 J=1,NPANEL
      HCOND(I,J)=0.
      FSP1(I,J)=0.
      FSPP1(I,J)=0.
 3    FSPP2(I,J)=0.
      READ(5,1856) KEXT,KINT,KRRAD,KHCOND
      READ (5,15)RADE,ALT,VELTO,VELRO,TI,DTI,TID,S                      1030
      READ(5,15)TE,BOLT,G,AU,RE
 1322 FORMAT(1H058HSOLAR CONSTANT,S-B CONSTANT,EARTH TEMP.,EARTH REFLECT
```

```
     1IVITY)
      WRITE(6,1322)
      WRITE(6,17) S,BOLT,TE,RE
      READ(5,15)ABETA,AINCE,AINCO,ALPHA,COMEGA,DOMEGA,OMEGAP,PHIN,THETAO
      IF(IFTLUP) 11,11,12
 12   READ(5,15)BALPHA
      READ(5,15)BBETA
      READ(5,15)BTHETA
 11   READ(5,15) (AR1(J),J=1,NPANEL)
      READ(5,14) (AP1(J),J=1,NPANEL)
      READ(5,14) (APP1(J),J=1,NPANEL)
      READ(5,14) (CSH(J),J=1,NPANEL)
      READ(5,14) (WATE(J),J=1,NPANEL)
      WRITE(6,1313)
 1313 FORMAT(1H010HEXT. AREAS)
      WRITE(6,17)  AR1
 1314 FORMAT(1H025HEXT. SURF. ABSORPTIVITIES)
      WRITE(6,1314)
      WRITE(6,17) AP1
 1316 FORMAT(1H023HEXT. SURF. EMISSIVITIES)
      WRITE(6,1316)
      WRITE(6,17) APP1
 1318 FORMAT(1H07HWEIGHTS)
      WRITE(6,1318)
      WRITE(6,17) WATF
 1320 FORMAT(1H014HSPECIFIC HEATS)
      WRITE(6,1320)
      WRITE(6,17) CSH
      READ(5,14) (T(J),J=1,NPANEL)
      IF(KEXT.EQ.0) GO TO 1111
      DO 1492 K=1,NFAC
      READ(5,1984) I1,J1,FSPP1(I1,J1),I2,J2,FSPP1(I2,J2),I3,J3,FSPP1(I3,
     1J3)
      FSPP1(J1,I1)=FSPP1(I1,J1)
      FSPP1(J2,I2)=FSPP1(I2,J2)
 1492 FSPP1(J3,I3)=FSPP1(I3,J3)
      DO 1776 I=1,NPANEL
      DO 1776 J=1,NPANEL
      FSP1(I,J)=FSPP1(I,J)*AR1(J)
      FSPP1(I,J)=FSP1(I,J)
      RRAD(I,J)=-FSP1(I,J)*(1.-AP1(J))
1776  IF(I.EQ.J) RRAD(I,J)=RRAD(I,J)+1.
      CALL SIMEQ(RRAD,NPANEL,FSP1,NPANEL,DETERM,IPIVOT,104,ISCALE)
 1861 FORMAT(1H160X6HFSP1-S///)
```

```
      WRITE(6,1861)
      WRITE(6,1905) (( FSP1(J,K),K=1,NPANEL),J=1,NPANEL)
      DO 1849 I=1,NPANEL
      DO 1849 J=1,NPANEL
      RRAD(I,J)=-FSPP1(I,J)*(1.-APP1(J))
1849  IF(I.EQ.J) RRAD(I,J)=RRAD(I,J)+1.
      CALLSIMEQ(RRAD,NPANEL,FSPP1,NPANEL,DETERM,IPIVOT,104,ISCALE)
 1862 FORMAT(1H160X7HFSPP1-S///)
      WRITE(6,1862)
      WRITE(6,1905) ((FSPP1(J,K),K=1,NPANEL),J=1,NPANEL)
 1111 CONTINUE
      RADO = RADE + ALT
      THETA = THETAO
      P = RADO*VELTO
      COSEGA=COS(COMEGA)
      SINEGA=SIN(COMEGA)
      COINCE=COS(AINCE)
      SINCE=SIN(AINCE)
      IF(KINT.EQ.0) GO TO 1033
      READ(5,14)(AR2(J),J=1,NPANEL)
      READ(5,14) (APP2(J),J=1,NPANEL)
      DO 7011 I=1,NINSFC
      READ(5,1984) I1,J1,FSPP2(I1,J1),I2,J2,FSPP2(I2,J2),I3,J3,FSPP2(I3,
     1J3)
      FSPP2(J1,I1)=FSPP2(I1,J1)
      FSPP2(J2,I2)=FSPP2(I2,J2)
 7011 FSPP2(J3,I3)=FSPP2(I3,J3)
      DO 1929 I=1,NPANEL
      DO 1929 J=1,NPANEL
      FSPP2(I,J)=FSPP2(I,J)*AR2(J)
      RRAD(I,J)=-FSPP2(I,J)*(1.-APP2(J))
1929  IF(I.EQ.J) RRAD(I,J)=RRAD(I,J)+1.
      CALLSIMEQ(RRAD,NPANEL,FSPP2,NPANEL,DETERM,IPIVOT,104,ISCALE)
 1863 FORMAT(1H160X7HFSPP2-S///)
      WRITE(6,1863)
      WRITE(6,1905) ((FSPP2(J,K),K=1,NPANEL),J=1,NPANEL)
1033  IF(KHCOND.EQ.0) GO TO 1030
      DO 1010 I=1,NHCOND
      READ(5,1984)I1,J1,HCOND(I1,J1),I2,J2,HCOND(I2,J2),I3,J3,HCOND(I3,J
     13)
      HCOND(J1,I1)=HCOND(I1,J1)
      HCOND(J2,I2)=HCOND(I2,J2)
      HCOND(J3,I3)=HCOND(I3,J3)
 1010 CONTINUE
```

```
 1324 FORMAT(1H123HCONDUCTION COEFFICIENTS)
      WRITE(6,1324)
      WRITE(6,17) HCOND
 1030 CONTINUE
      DO 207 I=1,NPANEL
      DO 207 J=1,NPANEL
 207  RRAD(I,J)=0.
      IF(KRRAD.EQ.0) GO TO 1020
      DO 1011 I=1,NRRAD
      READ(5,1984)I1,J1,RRAD(I1,J1),I2,J2,RRAD(I2,J2),I3,J3,RRAD(I3,J3)
      RRAD(J1,I1)=RRAD(I1,J1)
      RRAD(J2,I2)=RRAD(I2,J2)
      RRAD(J3,I3)=RRAD(I3,J3)
 1011 CONTINUE
 1020 CONTINUE
      READ(5,1856) NETAS,NPHIS
      IF(ISPIN) 1312,1312,1311
 1311 DO 420 I=1,NPANEL
 420  READ(5,10101) (TXMU(I,J),J=1,NETAS)
 4891 FORMAT(1H012HMU BAR TABLE)
      WRITE(6,4891)
      WRITE(6,17) ((TXMU(I,J),J=1,NETAS),I=1,NPANEL)
 1312 READ(5,14) (TETAS(M),M=1,NETAS)
 4893 FORMAT(1H010HETAS TABLE)
      WRITE(6,4893)
      WRITE(6,17) (TETAS(M),M=1,NETAS)
      DO 7 M=1,NETAS
  7   TETAS(M)=TETAS(M)*.0174532925
      IF(ISPIN.EQ.1) GO TO 103
 532  DO 4 I=1,NPANEL
      DO 4 J=1,NETAS
  4   READ(5,10101) (XMUSUN(I,J,K),K=1,NPHIS)
 4895 FORMAT(1H120HMU(ETAS,PHIS) TABLES)
      WRITE(6,4895)
      DO 2849 I=1,NPANEL
      DO 2849 J=1,NETAS
 2849 WRITE(6,1701) (XMUSUN(I,J,K),K=1,NPHIS)
 1701 FORMAT(//2X,10E12.5)
      READ(5,14) (TPHIS(NP),NP=1,NPHIS)
 4897 FORMAT(1H010HPHIS TABLE)
      WRITE(6,4897)
      WRITE(6,17) (TPHIS(NP),NP=1,NPHIS)
      DO 246 M=1,NPHIS
 246  TPHIS(M)=TPHIS(M)*.01745329252
```

```
 531  READ(5,16) ((AXIS1(K,J),K=1,3),J=1,NPANEL)
      READ(5,16) ((AXIS2(K,J),K=1,3),J=1,NPANEL)
      READ(5,16) ((UVEC(K,J),K=1,3),J=1,NPANEL)
      DO 8 L=1,NPANEL
      STOR=AXIS1(1,L)
      AXIS1(1,L)=AXIS1(2,L)
      AXIS1(2,L)=AXIS1(3,L)
      AXIS1(3,L)=STOR
      STOR=AXIS2(1,L)
      AXIS2(1,L)=AXIS2(2,L)
      AXIS2(2,L)=AXIS2(3,L)
      AXIS2(3,L)=STOR
      STOR=UVEC(1,L)
      UVEC(1,L)=UVEC(2,L)
      UVEC(2,L)=UVEC(3,L)
   8  UVEC(3,L)=STOR
C        PROGRAM TO CALCULATE ENVIRONMENT OF SPACECRAFT
C
C
C     ROTMTX(M,N,ETA) COMPUTES THE ELEMENTS OF A ROTATION MATRIX M ABOUT AXIS N
C                    (N=1 TO 3) THROUGH ANGLE ETA
C
 103  CALL ROTMTX(RM3PN,3,PHIN)
      CALL ROTMTX(RM1I,1,AINCO)
      CALL ROTMTX(RM3OP,3,OMEGAP)
      CALL MULT(RMIPHI,RM1I,RM3PN)
      CALL MULT(RMPIF,RM3OP,RMIPHI)
 100  IF(IFTLUP) 102,102,101
 101  CALL FTLUP(THETA,ABETA,1,20,BTHETA,BBETA)
      CALL FTLUP(THETA,ALPHA,1,20,BTHETA,BALPHA)
 102  SINTHE = RADE/RADO
      COSTHE=(1.-SINTHE**2)**.5
      OMEGA=TI*DOMEGA
      THETAN=-THETA
      CALL ROTMTX(RM3NTH,3,THETAN)
      CALL ROTMTX(RM3A,3,ALPHA)
      CALL ROTMTX(RM1B,1,ABETA)
      CALL ROTMTX(RM2O,2,OMEGA)
C
C     MULT(MP,M1,M2) MULTIPLIES 3X3 MATRICES M1 AND M2,GIVING (MP)=(M1)(M2)
C
      CALL MULT(RMEB,RM3A,RMPIF)
      CALL MULT(RMEB,RM1B,RMEB)
      CALL MULT(RMEB,RM2O,RMEB)
```

```
C          RMEB IS COMPLETED
C   ROTATION MATRIX RMEB ROTATES THE INERTIAL REFERENCE FRAME (Z-AXIS COINCIDENT
C   WITH PLANET SPIN AXIS,X-AXIS PASSES THROGH THE VERNAL EQUINOX) TO COINCIDE
C   WITH THE SPACECRAFT BODY AXES
C
      DO 170 J=1,3
  170 VEC(J)=RMEB(J,1)*COSEGA+RMEB(J,2)*SINEGA*COINCE+RMEB(J,3)*SINEGA*
     1 SINCE
      CALL MULT(RMOB,RM3A,RM3NTH)
      CALL MULT(RMOB,RM1B,RMOB)
      CALL MULT(RMOB,RM20,RMOB)
C
C   ROTATION MATRIX RMOB GIVES ORBIT POSITION. COLUMN 1 GIVES COMPONENTS OF UNIT
C   VECTOR IN DIRECTION OF CRAFT FROM CENTER OF PLANET
C
      IF(ISPIN.EQ.1) GO TO 1
      DO 160 I=1,NPANEL
      DO 9 J=1,3
      VE1(J)=AXIS1(J,I)
    9 VE2(J)=AXIS2(J,I)
      COSE=-UVEC(1,I)*RMOB(1,1)-UVEC(2,I)*RMOB(2,1)-UVEC(3,I)*RMOB(3,1)
      SINE=(1.-COSE*COSE)**.5
      IF(     COSE .LT..1E-7) ETAE(I)=1.57079633
      IF(     COSE .GE..1E-7) ETAE(I)=ATAN2(SINE,COSE)
      IF((1.-ABS(COSE)).LT.1.E-8) GO TO 1809
      COSE=-RMOB(1,1)*VE1(1)-RMOB(2,1)*VE1(2)-RMOB(3,1)*VE1(3)
      SINE=-RMOB(1,1)*VE2(1)-RMOB(2,1)*VE2(2)-RMOB(3,1)*VE2(3)
      PHIE(I)=ATAN2(SINE,COSE)
      GO TO 1810
 1809 PHIE(I)=0.
 1810 CONTINUE
      COSE=UVEC(1,I)*VEC(1)+UVEC(2,I)*VEC(2)+UVEC(3,I)*VEC(3)
      SINE=(1.-COSE*COSE)**.5
      IF(     COSE .LT..1E-7) ETAS(I)=1.57079633
      IF(     COSE .GE..1E-7) ETAS(I)=ATAN2(SINE,COSE)
      IF((1.-ABS(COSE)).LT.1.E-8) GO TO 1918
      COSE=VE1(1)*VEC(1)+VE1(2)*VEC(2)+VE1(3)*VEC(3)
      SINE=VE2(1)*VEC(1)+VE2(2)*VEC(2)+VE2(3)*VEC(3)
      PHIS(I)=ATAN2(SINE,COSE)
      GO TO 160
 1918 PHIS(I)=0.
  160 CONTINUE
      GO TO 2
    1 CONTINUE
```

```
      COSE=VEC(2)
      SINE=(1.-COSE*COSE)**.5
      IF(ABS(COSE).LT..1E-7) ETASUN=1.57079633
      IF(ABS(COSE).GE..1E-7) ETASUN=ATAN2(SINE,COSE)
      COSE=-RMOB(2,1)
      SINE=(1.-COSE*COSE)**.5
      IF(ABS(COSE).LT..1E-7) ETARTH=1.57079633
      IF(ABS(COSE).GE..1E-7) ETARTH=ATAN2(SINE,COSE)
      DO 6 I=1,NPANEL
      ETAS(I)=ETASUN
    6 ETAE(I)=ETARTH
    2 CONTINUE
      AMP=RMOB(1,1)*VEC(1)+RMOB(2,1)*VEC(2)+RMOB(3,1)*VEC(3)
      H=1.0
      IF(-AMP.GT.COSTHE) H=0.0
      IF(AMP.LT.0.0) AMP=0.0
C
      DO 1000 J=1,NPANEL
      IK=0
      IF(ISPIN) 1357,1357,1358
 1357 DO 680 I=1,NETAS
      DO 680 K=1,NPHIS
      IK=IK+1
  680 TXMUS(IK)=XMUSUN(J,I,K)
      NY=NETAS*NPHIS
      CALL DISCOT(ETAS(J),PHIS(J),TETAS,TXMUS,TPHIS,22,NY,NPHIS,XMU(J))
      CALL DISCOT(ETAF(J),PHIE(J),TETAS,TXMUS,TPHIS,22,NY,NPHIS,YMU(J))
      GO TO 1000
 1358 DO 1001 I=1,NETAS
 1001 TTXMU(I)=TXMU(J,I)
      CALL FTLUP(ETAS(J),XMU(J),2,NETAS,TETAS,TTXMU)
      CALL FTLUP(ETAE(J),YMU(J),2,NETAS,TETAS,TTXMU)
 1000 CONTINUE
* AT THIS  POINT PROJECTED AREAS FOR SOLAR AND PARENT BODY ARE AVAILABLE
      DO 1002 I=1,NPANEL
      DT(I)=H*S*AR1(I)*XMU(I)*AP1(I)+BOLT*TE**4*SINTHE**2*AR1(I)*YMU(I)
     1*APP1(I)+S*RE*AMP*SINTHE**2*AR1(I)*YMU(I)*AP1(I)
C
C     (DIR. SOLAR+DIR. PLANET EMISSION+ PLANET-REFLECTED SOLAR)
C
      TEMP=0.
      DO 1003 K=1,NPANEL
      TEMP=TEMP+(APP1(K)*BOLT*T(K)**4+BOLT*TE**4*SINTHE**2*YMU(K)*(1.-
     1APP1(K)))       *FSPP1(I,K)*AR1(I)*APP1(I)
```

```
 1003 CONTINUE
C
C     (TEMP ADDS EXTERIOR LONG WAVELENGTH CONTRIB. TO PANEL I FROM ALL PANELS,
C                DIRECTLY AND BY ALL MULTIPLE DIFFUSE REFLECTIONS. CONSISTS OF
C                  EXT. THERMAL EMISS. BY PANELS + REFLECTED PORTION OF PLANET EM
C
      SEMP=0.
      DO 1004 K=1,NPANEL
      SEMP=SEMP+(H*S*XMU(K)+S*RE*AMP*SINTHE**2*YMU(K))*(1.-AP1(K))            2975
     1 *FSP1(I,K)*AR1(I)*AP1(I)
 1004 CONTINUE
C
C     (SEMP ADDS SOLAR SPECTRUM CONTRIB. FROM ALL PANELS INCL. ALL REFLECTIONS
C              CONSISTS OF PANEL-REFLECTED PORTION OF DIR. AND PLANET-REFL. SOLAR
C
      UEMP=0.
      DO 1005 K=1,NPANEL
      UEMP=UEMP+BOLT*APP2(K)*T(K)**4        *FSPP2(I,K)*AR2(I)*APP2(I)        2995
 1005 CONTINUE
C
C      (UEMP ADDS INTERIOR LONG WAVELENGTH CONTRIB. TO I IF IT IS PART OF AN
C                ENCLOSURE . CONSISTS OF EMISSIONS AND ALL REFL. FROM SURF. OF
C                 ENCLOSURE)
C
      VEMP=0.
      DO 1006 K=1,NPANEL
      VEMP=VEMP+HCOND(K,I)*(T(K)-T(I))
 1006 CONTINUE
C
C     (VEMP IS THE CONDUCTION TO I  )
C
      WEMP=0.
      DO 1007 K=1,NPANEL
      WEMP=WEMP+RRAD(K,I)*(T(K)**4-T(I)**4)
 1007 CONTINUE
      WEMP=WEMP*AR1(I)
C
C     (WEMP GIVES RADIATION FROM ANOTHER PANEL PARALLEL TO I AND FORMING A FLAT
C                ENCLOSURE WITH I. EXAMPLE IS FLAT PRESSURE CELL METEOROID DETEC.
C
      XEMP=-BOLT*T(I)**4*(APP1(I)*AR1(I)+APP2(I)*AR2(I))
C
C     (XEMP GIVES LOSS BY EMISSION FROM I  )
C
```

```
      DT(I)=DT(I)+TEMP+SEMP+UEMP+VEMP+WEMP+XEMP
      DT(I)=DT(I)/WATE(I)/CSH(I)
 1002 CONTINUE
      DO 9843 I=1,NPANEL
 9843 T(I)=T(I)+DT(I)*DTI
      IF(MOD(NTIME,100).EQ.0)GO TO 113
      GO TO 114
  113 CONTINUE
      WRITE(6,17) (T(I),I=1,NPANEL)
  114 IF(TI.LT.TID) GO TO 99
      GO TO 999
C
C     ROUTINE TO CALCULATE RADIUS IN ELLIPTICAL ORBIT
C
   99 CONTINUE
      DDTI=.1*DTI
      DO 992 J = 1,10
      ACCEL = (P**2)/RADO**3 -G*(RADE/RADO)**2
      ACCEL1 =ACCEL
      DO 991 I = 1,5
      RAD1=RADO+VELRO*DDTI+(2.*ACCEL+ACCEL1)*DDTI**2/6.
      VELR1=VELRO+.5*(ACCEL+ACCEL1)*DDTI
  991 ACCEL1 = (P**2)/RAD1**3 - G*(RADE/RAD1)**2
      RADH = RAD1-RADE
      ANVELO =P/RADO**2
      ANVEL1 = P/RAD1**2
      THETA = THETA + (ANVELO+ANVEL1)/2.0*(DTI/10.0)
      RADO = RAD1
  992 VELRO = VELR1
      RATIO = RADH/ALT
      HIHT = RADH/5280.0
      TI=DTI*FLOAT(NTIME)
      IF(MOD(NTIME,20).NE.0) GO TO 30
      WRITE(6,25) TI,THETA,HIHT,H,AMP,ETAS(1)
   30 NTIME=NTIME+1
      GO TO 100
  999 STOP
      END
      SUBROUTINE ROTMTX(ROTOR,M,ROTANG)
      DIMENSION ROTOR(3,3)
      ROTOR(M,M)=1.0
      COSE=COS(ROTANG)
      SINE=SIN(ROTANG)
      DO 1 I=1,3
```

```
      IF(I.EQ.M) GO TO 1
      ROTOR(I,M)=.0
      ROTOR(M,I)=.0
      ROTOR(I,I)=COSE
    1 CONTINUE
      IF(M.EQ.1) GO TO 2
      IF(M.EQ.2) GO TO 3
      ROTOR(1,2)=SINE
      ROTOR(2,1)=-SINE
      RETURN
    2 ROTOR(2,3)=SINE
      ROTOR(3,2)=-SINE
      RETURN
    3 ROTOR(1,3)=-SINE
      ROTOR(3,1)=SINE
      RETURN
      END
C
C   SUBROUTINE ROTMTX CALCULATES THE ELEMENTS OF A ROTATION MATRIX,GIVEN THE
C     AXIS NO. AND THE CCW ANGLE OF ROTATION
      SUBROUTINE MULT(A,B,C)
      DIMENSION A(3,3),B(3,3),C(3,3),D(3,3)
      DO 1 I=1,3
      DO 1 J=1,3
      D(I,J)=0.0
      DO 1 K=1,3
    1 D(I,J)=D(I,J)+B(I,K)*C(K,J)
      DO 2 I=1,3
      DO 2 J=1,3
    2 A(I,J)=D(I,J)
      RETURN
      END
```

Definitions and Instructions for the Temperature Program

```
C      HEAT TRANSFER PROGRAM FOR DIFFUSE RADIATION ON MULTIPANELED SPACECRAFT
C
C     SYMBOLS USED IN PROGRAM
C          ABETA      ANGLE OUT OF ORBIT PLANE OF SPIN AXIS (YAW)
C          AINCE      ANGLE BETWEEN ECLIPTIC PLANE AND EQUATORIAL PLANE
C          AINCO      ORBIT INCLINATION ANGLE
C          ALPHA      NEGATIVE ELEVATION ANGLE  (PITCH)
C          ALT        ALTITUDE OF SPACECRAFT (FT)
C     AMP          FOR DAYLIGHT, AMP=COS(ANGLE BETWEEN LINES TO THE SUN AND SPACECR
C                    AFT DRAWN FROM THE CENTER OF THE EARTH) FOR DARKNESS,AMP=0
*      AP1(J)       ABSORPTIVITY OF EXTERNAL CAVITY TO P ENERGY,DIMENSIONLESS
*      APP1(J)      ABSORPTIVITY OF EXTERNAL CAVITY TO PP ENERGY,DIMENSIONLESS
*      APP2(J)      ABSORPTIVITY OF  INTERNAL CAVITY TO PP ENERGY,DIMENSIONLESS
*      AR1(J)       EXTERNAL CAVITY AREA,INCH SQUARE
*      AR2(J)       INTERNAL CAVITY AREA,INCH SQUARE
C         AU        DISTANCE FROM SUN (ASTRONOMICAL UNITS)
C     AXIS1        UNIT VECTOR IN PLANE OF PANEL
C     AXIS2        CROSS PRODUCT OF UVEC AND AXIS1
C         BALPHA      PITCH TABLE (20 PARTS)
C         BBETA       YAW TABLE (20 PARTS)
C         BTHETA      ORBIT ANGLE TABLE (20 PARTS)
*      BOLT           STEFAN-BOLTZMANN CONSTANT
C         COMEGA      SUN ANGLE IN ELLIPTIC PLANE FROM VERNAL EQUINOX
*      CSH(J)       SPECIFIC HEAT OF JTH COMPONENT,BTU/POUND/DEG
C         DDEL        TIME DERIVATIVE OF ORBIT ANGLE (RADIANS/MIN)
C         DEL         ORBIT ANGLE (RADIANS)
C        DOMEGA     SPIN RATE OF SPACECRAFT ABOUT 2-AXIS
C         DT(I)       TEMPERATURE INCREMENT OF THE ITH PANEL
C         DTI         TIME INCREMENT (MIN)
C         EPI         ANGLE OF SUN OUT OF ORBIT PLANE
*      ETAE(J)      PARENT BODY ASPECT ANGLE
*      ETAS(J)      SOLAR ASPECT ANGLE
C        THE SHAPE FACTORS INPUT TO THE PROGRAM ARE ONLY GEOMETRIC. THE FACTORS
C    WHICH TAKE INTO ACCOUNT ALL REFLECTIONS AND ALL RE-REFLECTIONS ARE COMPUTED
C    IN THE PROGRAM USING GEOMETRIC FACTORS AND THE REFLECTIVITIES OF EACH NODE.
*      FSP1(K,J)    EXTERNAL SHAPE FACTOR ELEMENT FOR P ENERGY, /INCH SQUARE
*      FSPP1(K,J)   EXTERNAL SHAPE FACTOR ELEMENT FOR PP ENERGY, /INCH SQUARE
*      FSPP2(K,J)    INTERNAL SHAPE FACTOR ELEMENT FOR PP ENERGY, /INCH SQUARE
C         G           ACCELRATION OF GRAVITY  (FT/MIN**2)
*      H            SUN INDICATOR,DIMENSIONLESS
*      HCOND(K,J)   LINEAR HEAT CONDUCTION COEF,BTU/MIN/DEG
C         IFTLUP       INTEGER TO INDICATE USE OF BBETA,BTHETA,BALPHA
C                      USE 0 FOR CONSTANTS AND 1 FOR TABLES
C    ISPIN IS ZERO FOR A STABILIZED VEHICLE,NON-ZERO FOR SPINNING.
```

```
C     KEXT,KINT,KRRAD,AND KHCOND INDICATE WHETHER THERE ARE NON-ZERO ELEMENTS OF
C          FSP1/FSPP1,FSPP2,RRAD,AND HCOND RESPECTIVELY
C      NETAS          NO. OF VALUES IN TABLE OF ETAS
C     NFAC IS THE NO. OF CARDS USED TO LOAD THE NON-ZERO GEOMETRIC SHAPE FACTORS
C       BETWEEN EXTERNAL SURFACES(3 TO A CARD).FSPP1 IS INITIALLY USED AS A DUMMY
C       NAME FOR THIS ARRAY.
C     NHCOND IS SIMILARLY THE NO. OF CARDS FOR HCOND,NRRAD FOR RRAD,NINSFC FOR THE
C       INTERNAL GEOM. SHAPE FACTORS(READ IN AS FSPP2).
C           NPANEL     NUMBER OF PANELS                                         0510
C      NPHIS          NO. OF VALUES IN TABLE OF PHIS
C          OMEGA      ROLL ANGLE OF BODY
C           OMEGAP     ARGUMENT OF PERIGEE                                      0530
C        P DENOTES SOLAR SPECTRUM,PP DENOTES THERMAL RADIATION SPECTRUM PEAKING A
C              T MUCH LONGER WAVELENGTHS
C           PHIN       ARGUMENT OF ASCENDING NODE                               0560
C      PHIS(J)         SOLAR AZIMUTH ANGLE
C           RADE       EARTH RADIUS (FT)                                        0590
C           RADH       ALTITUDE OF SPACECRAFT (FT)                              0600
C          RADO       INJECTION RADIUS OF ELLIPTICAL ORBIT (FT)                 0610
*      REFE           REFLECTIVITY OF PARENT BODY TO SOLAR ENERGY,DIMENSIONLESS
C      RRAD(K,J)      RADIATION COEFFFICIENT,WITH NET HEAT TRANSFER RATE=
C                      AREA*RRAD(K,J)*(T(K)**4-T(J)**4)
*      S              SOLAR CONSTANT,BTU/INCH SQUARE/MINUTE
C           SINTHE     SINE OF THE EARTH VIEW ANGLE                             0690
*      T(J)           TEMPERATURE OF JTH COMPONENT,DEG
*      TE             TEMPERATURE OF PARENT BODY,DEG
C      TETAS,TPHIS,TXMU,TXMUS    TABLES OF ETAS,PHIS,XMU
C           THETA      ORBIT ANGLE FROM PERIGEE                                 0720
C          THETAO     INITIAL THETA (ARBITRARY)                                 0730
C           TI         TIME  (MIN)                                              0760
C           TID        CALCULATION TIME LIMIT                                   0770
C          UVEC      UNIT NORMAL VECTOR TO PANEL
C          VELRO      INJECTION VELOCITY NORMAL TO THE EARTH (FT/MIN)           0810
C          VELTO      INJECTION VELOCITY TANGENTIAL TO THE EARTH(FT/MIN)        0820
*      WATE(J)        MASS OF JTH COMPONENT,POUNDS
*      XMU(J)         UNIT PROJECTED AREA,DIMENSIONLESS,SOLAR
C      XMUSUN(K,I,J)         TABLE OF XMU OF EACH PANEL FOR GIVEN ETAS AND PHIS
*      YMU(J)         UNIT PROJECTED AREA,DIMENSIONLESS,PARENT BODY
C                                                                               0850
C
C     ARRAY DIMENSIONS FOR AN INDIVIDUAL CASE ARE-
C       BTHETA,BALPHA,BBETA-SIZE OF TABLES OF PITCH AND YAW VERSUS ORBIT ANGLE.
C       UVEC(3,NPANEL)
C       AR1,AR2,AP1,APP1,APP2,CSH,ETAS,ETAE-ALL(NPANEL).
```

```
C       FSP1,FSPP1,FSPP2,HCOND,RRAD,(NPANEL,NPANEL).
C       WATE,T,XMU,YMU-ALL  (NPANEL).
C       TETAS(NETAS),TXMU(NPANEL,NETAS),TTXMU(NETAS),DT(NPANEL).
C       XMUSUN(NPANEL,NETAS,NPHIS),TPHIS(NPHIS),PHIS(NPHIS).
C       AXIS1(3,NPANEL),AXIS2(3,NPANEL),TXMUS(NETAS*NPHIS),FAR(NPANEL,NPANEL).
C       IPIVOT(NPANEL),INDEX(NPANEL,2).
C
C
C    INPUT DATA LOADING OPDER
C       1.(8I10) NPANEL,NFAC,NHCOND,NRRAD,NINSFC,IFTLUP,ISPIN.
C       2. (8I10) KEXT,KINT,KRRAD,KHCOND.
C       3. (5E16.8) RADE,ALT,VELTO,VELRO,TI / DTI,TID,S.
C       4. (5E16.8) TE,BOLT,G,AU,RE.
C       5. (5E16.8) ABETA,AINCE,AINCO,ALPHA,COMEGA/DOMEGA,OMEGAP,PHIN,THETAO.
C       6. IF(IFTLUP.GT.0),(5E16.8) BALPHA,BBETA,BTHETA.
C       7. (5E16.8) AR1   8. (12F6.2) AP1   9.(12F6.2)APP1   10.(12F6.2)CSH
C       11.(12F6.2)WATE     12.(12F6.2) T
C       13. IF(KEXT.GT.0) FSPP1  (3(2I4,E16.8))
C       14. IF(KINT.GT.0) (12F6.2) AR2/APP2  15. IF(KINT.GT.0) FSPP2  (3(2I4,E16.8
C       16. IF(KHCOND.NE.0) HCOND  (3(2I4,E16.8))
C       17. IF(KRRAD.NE.0) RRAD  (3(2I4,E16.8))
C       18. (8I10)  NETAS,NPHIS
C       19. (7F11.8) TXMU 20. (12F6.2) TETAS
C       IF(ISPIN.EQ.0) - 21. (7F11.8) XMUSUN,22. (12F6.2) TPHIS, 23. (6F13.3)
C                                                        AXIS1,AXIS2,UVEC
C
C
C    PROGRAM OUTPUT-
C          1. IF(KEXT.NE.0),(8E16.8) FSP1,FSPP1-CONVERTED TO ACCOUNT FOR REFLECTIO
C          2.IF(KINT.NE.0),(8E16.8) FSPP2-ALSO CONVERTED
C          3. EACH MINUTE,TI(MIN.),THETA,ALT.(STAT.MI.),H(1. FOR SUNLIT),AMP(MEA-
C          SURE OF PLANET-REFLECTED SUNLIGHT),ETAS(1).  (2X,6E20.6)
C          4. EVERY TEN MINUTES,TEMP. OF EACH PANEL,DEG.R  (2X,10E12.5)
C
```

Shape Factor Program

```
      PROGRAM SHAPE(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT)
C       SHAPE FACTOR PROGRAM FOR DIFFUSE RADIATION
C THE FOLLOWING INSTRUCTIONS MUST BE FOLLOWED WHEN USING THIS PROGRAM
C IN THE MAIN PROGRAM CHANGE DIMENSION STATESMENT TO READ
C      DIMENSION C(M,4,3),D(N,5,3),NUMB1(M),NUMB2(N),COEFFA(L,L,4),COEFFR(L,L,4)
C IN SUBROUTINE ZAP CHANGE DIMENSION STATEMENT TO READ
C      DIMENSION Z(N,5,3)
C IN SUBROUTINE AREA,CHANGE DIMENSIONS TO READ
C      DIMENSION COEFFA(L,L,4),COEFFR(L,L,4)
C IN SUBROUTINE BLOCK CHANGE DIMENSION STATEMENT TO READ
C      DIMENSION B(N,5,3),NUMB2(N)
C WHERE M IS THE NUMBER OF PANELS AND N IS THE NUMBER OF BLOCKING PANELS. L IS
C      THE GRID SIZE.
C VARIABLE DEFINITIONS,NO IS THE NUMBER OF PANELS,NBLOCK IS THE NUMBER
C OF BLOCKING PANELS. C IS AN ARRAY CONTAINING THE COORDINATES OF PANELS.
C D IS AN ARRAY CONTAINING THE COORDINATES OF THE BLOCKING PANELS
C NUMB1 IS AN ARRAY OF THE NUMBERS ASSIGNED TO THE PLANES OF EACH PANEL
C NUMB2 IS AN ARRAY OF THE NUMBERS ASSIGNED TO THE PLANES OF EACH BLOCKER
C
C   PUNCHED OUTPUT-
C        (3(2I4,E16.8)) NON-ZERO SHAPE FACTORS FOR PANELS I AND J,3 TO A CARD
C        I1,J1,SHPFAC(I1,J1),I2,J2,SHPFAC(I2,J2),I3,J3,SHPFAC(I3,J3)
C
      DIMENSIONC(100,4,3),D(75,5,3),A(4,3),B(4,3)
      DIMENSION NUMB1(100),NUMB2(75)
      DIMENSION TENT(3),NZ1(3),NZ2(3)
      DIMENSION COEFFA(10,10,4),COEFFR(10,10,4)
      COMMON/ONE/D
      COMMON/TWO/NUMB2
      COMMON /THREE/ COEFFA,COEFFR
 86   FORMAT(7I10)
 10   FORMAT(3E16.8,2I4)
   11 FORMAT(12F6.1)
 330  FORMAT (3(2I4,E16.8))
      READ(5,86)NO,NBLOCK
      READ(5,11)(((C(I,J,K),K=1,3),J=1,4),I=1,NO)
      READ(5,11)(((D(I,J,K),K=1,3),J=1,4),I=1,NBLOCK)
      READ(5,86)(NUMB1(I),I=1,NO)
      READ(5,86)(NUMB2(I),I=1,NBLOCK)
      WRITE(6,5)
  5   FORMAT(1H 14HNO. OF PANELS,15HNO. OF BLOCKERS)
      WRITE(6,86) NO,NBLOCK
      WRITE(6,7)
    7 FORMAT(1H017HPANEL COORDINATES)
```

```
      WRITE(6,11) (((C(I,J,K),K=1,3),J=1,4),I=1,NO)
      WRITE(6,19)
   19 FORMAT(1H 19HBLOCKER COORDINATES)
      WRITE(6,11) (((D(I,J,K),K=1,3),J=1,4),I=1,NBLOCK)
      WRITE(6,6)
 6    FORMAT(1H016HPANEL PLANE NOS.)
      WRITE(6,86) (NUMB1(I),I=1,NO)
      WRITE(6,8)
 8    FORMAT(1H018HBLOCKER PLANE NOS.)
      WRITE(6,86) (NUMB2(I),I=1,NBLOCK)
      WRITE(6,9)
 9    FORMAT(1H14H    I4H    J3X9HAREA OF I7X9HAREA OF J)
      NZ=1
      NG=5
      NG2=NG*NG
      FOURN2=FLOAT(4*NG2)
      TWON4=FLOAT(2*NG2*NG2)
      DO 75 M=1,NG
      I1=M-1
      I21=I1+M
      NI=NG-M
      NI1=2*NI+1
      DO 75 N=1,NG
      J1=N-1
      J21=J1+N
      NJ=NG-N
      NJ1=2*NJ+1
      COEFFA(M,N,1)=FLOAT(NI*NJ+(NI+1)*(NJ+1))/TWON4
      COEFFA(M,N,2)=FLOAT(I1*(NJ+1)+M*NJ)/TWON4
      COEFFA(M,N,3)=FLOAT((NI+1)*J1+NI*N)/TWON4
      COEFFA(M,N,4)=FLOAT(I1*J1+M*N)/TWON4
      COEFFR(M,N,1)=FLOAT(NI1*NJ1)/FOURN2
      COEFFR(M,N,2)=FLOAT(I21*NJ1)/FOURN2
      COEFFR(M,N,3)=FLOAT(I21*J21)/FOURN2
 75   COEFFR(M,N,4)=FLOAT(NI1*J21)/FOURN2
      CALL ZAP(NBLOCK)
      IX=NO-1
      DO 3 I=1,IX
      JX=I+1
      DO 3 J=JX,NO
      DO 1 K=1,3
      DO 1 L=1,4
      A(L,K)=C(I,L,K)
    1 B(L,K)=C(J,L,K)
```

```
      NA=NUMB1(I)
      NB=NUMB1(J)
      IF(NA.EQ.NB) SAB=0.0
      IF(NA.EQ.NB) GO TO 4
      CALL FAKTOR(A,B,SAB,SDA,SDB,NA,NB,NG,NBLOCK,NG2)
4     CONTINUE
      IF(SAB.GT.1.E-10)GO TO 333
      IF((I.EQ.IX).AND.(J.EQ.NO)) GO TO 333
      GO TO 3
333   TENT(NZ)=SAB
      NZ1(NZ)=I
      NZ2(NZ)=J
      WRITE(6,213) I,J,SAB
213   FORMAT(2I4,E16.8,8H VIEWFAC)
      NZ=NZ+1
      IF(NZ.EQ.4) GO TO 335
      IF((I.EQ.IX).AND.(J.EQ.NO)) GO TO 335
      GO TO 3
335   PUNCH 330,NZ1(1),NZ2(1),TENT(1),NZ1(2),NZ2(2),TENT(2),NZ1(3),NZ2(3
     1),TENT(3)
      NZ=1
3     CONTINUE
      STOP
      END
      SUBROUTINE ZAP(NBLOCK)
      DIMENSION Z(75,5,3),C1(3),C2(3),C3(3),C4(3),C5(3),AM1(3),AM2(3),
     1V(3)
      COMMON/ONE/Z
      I=1
5     DO 1 J=1,3
      C2(J)=Z(I,2,J)-Z(I,1,J)
1     C3(J)=Z(I,4,J)-Z(I,1,J)
      CALL CROSS(C5,C2,C3)
      CALL CROSS(C1,C3,C5)
      CALL DOT(DET,C1,C2)
      CALL CROSS(AM2,C2,C5)
      DO 2 K=1,3
      AM1(K)=C1(K)/DET
    2 AM2(K)=AM2(K)/DET
      DO 3 J=1,3
3     V(J)=Z(I,1,J)+Z(I,3,J)-Z(I,2,J)-Z(I,4,J)
      C4(1)=AM 1(1)*V(1)+AM 1(2)*V(2)+AM 1(3)*V(3)
      C4(2)=AM 2(1)*V(1)+AM 2(2)*V(2)+AM 2(3)*V(3)
      C4(3)=0.
```

```
      DO 6 J=1,3
      Z(I,2,J)=C2(J)
      Z(I,3,J)=C3(J)
      Z(I,4,J)=C4(J)
6     Z(I,5,J)=C5(J)
      I=I+1
      IF(I.LT.(NBLOCK+1)) GO TO 5
      RETURN
      END
      SUBROUTINE FAKTOR(A,B,SIJ,SDA,SDB,NA,NB,NG,NBLOCK,NG2)
      DIMENSION A(4,3),B(4,3),DA(100),DB(100),RA(100,3),RB(100,3),AV(3),
     1BV(3),CV(3),DV(3),EV(3),FV(3),X(3),Y(3)
      DIMENSION XA(3),CB(3),RIJ(16,3),RDOTN(16,2),GV(3),HV(3)
      SIJ=0.0
      I=1
      DO 1 N=1,3
      AV(N)=A(1,N)
      BV(N)=A(2,N)
      CV(N)=A(3,N)
      DV(N)=B(1,N)
      EV(N)=B(2,N)
      FV(N)=B(3,N)
      GV(N)=A(4,N)
      HV(N)=B(4,N)
      RIJ(1,N)=DV(N)-AV(N)
      RIJ(2,N)=EV(N)-AV(N)
      RIJ(3,N)=FV(N)-AV(N)
      RIJ(4,N)=HV(N)-AV(N)
      RIJ(5,N)=DV(N)-BV(N)
      RIJ(6,N)=EV(N)-BV(N)
      RIJ(7,N)=FV(N)-BV(N)
      RIJ(8,N)=HV(N)-BV(N)
      RIJ(9,N)=DV(N)-CV(N)
      RIJ(10,N)=EV(N)-CV(N)
      RIJ(11,N)=FV(N)-CV(N)
      RIJ(12,N)=HV(N)-CV(N)
      RIJ(13,N)=DV(N)-GV(N)
      RIJ(14,N)=EV(N)-GV(N)
      RIJ(15,N)=FV(N)-GV(N)
      RIJ(16,N)=HV(N)-GV(N)
      AV(N)=BV(N)-AV(N)
      BV(N)=CV(N)-BV(N)
      DV(N)=FV(N)-DV(N)
    1 EV(N)=FV(N)-FV(N)
```

```
      CALL CROSS(CV,AV,BV)
      CALL CROSS(FV,DV,EV)
      DO 14 J=1,16
      DO 14 N=1,2
   14 RDOTN(J,N)=0.
      DO 15 J=1,16
      DO 13 N=1,3
      RDOTN(J,1)=RDOTN(J,1)+RIJ(J,N)*CV(N)
   13 RDOTN(J,2)=RDOTN(J,2)+RIJ(J,N)*FV(N)
      IF((RDOTN(J,1).GE.0.1).AND.(RDOTN(J,2).LE.-0.1)) GO TO 5
   15 CONTINUE
      SIJ=0.
      SDA=1.
      SDB=1.
      RETURN
    5 CALL DOT(SA,CV,CV)
      CALL DOT(SB,FV,FV)
      SA=SA**.5
      SB=SB**.5
      DO 2 N=1,3
      CV(N)=CV(N)/SA
2     FV(N)=FV(N)/SB
      CALL AREA(A,DA,RA,NG,NG2)
      CALL AREA(B,DB,RB,NG,NG2)
      DO 3 L=1,NG2
      DO 3 J=1,NG2
      DO 4 N=1,3
      X(N)=RA(L,N)
      XA(N)=X(N)
      Y(N)=RB(J,N)
    4 X(N)=X(N)-Y(N)
      CALL DOT(S1,CV,X)
      CALL DOT(S2,FV,X)
      IF((S1.GE.0.0).OR.(S2.LE.0.0)) GO TO 3
      CALL BLOKK(XA,Y,NBLOCK,G,NA,NB,I)
      IF(G.LT..5) GO TO 3
      CALL DOT(S3,X,X)
      S3=S3**2
      SIJ=-S1*S2/S3*DA(L)*DB(J)+SIJ
3     CONTINUE
11    SDA=0.0
      SDB=0.0
      DO 7 N=1,NG2
      SDA=SDA+DA(N)
```

```
    7 SDB=SDB+DB(N)
      SIJ=SIJ/(SDA*SDB*3.1415926536)
      IF(((SIJ*SDA).LT.1.E-5).AND.((SIJ*SDB).LT.1.E-5)) SIJ=0.
      RETURN
      END
      SUBROUTINE AREA(A,DA,RA,NG,NG2)
      DIMENSION A(4,3),DA(100),RA(100,3),AV(3),BV(3),CV(3),DV(3)
     1,T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3),T8(3)
      DIMENSION COEFFA(10,10,4),COEFFR(10,10,4)
      COMMON /THREE/ COEFFA,COEFFR
   18 FORMAT(4E16.8)
      DO 2 N=1,3
       AV(N)=A(1,N)
      BV(N)=A(2,N)
      CV(N)=A(3,N)
      DV(N)=A(4,N)
      T1(N)=BV(N)-AV(N)
      T2(N)=CV(N)-BV(N)
      T3(N)=CV(N)-DV(N)
    2 T4(N)=DV(N)-AV(N)
      CALL CROSS(T5,T4,T1)
      CALL CROSS(T6,T2,T1)
      CALL CROSS(T7,T4,T3)
      CALL CROSS(T8,T2,T3)
      CALL DOT(Z1,T5,T5)
      CALL DOT(Z2,T6,T6)
      CALL DOT(Z3,T7,T7)
      CALL DOT(Z4,T8,T8)
      Z1=Z1**.5
      Z2=Z2**.5
      Z3=Z3**.5
      Z4=Z4**.5
      K=1
      DO 1967 I=1,NG
      DO 1967 J=1,NG
      DA(K)=Z1*COEFFA(I,J,1)+Z2*COEFFA(I,J,2)+Z3*COEFFA(I,J,3)+Z4*COEFFA
     1(I,J,4)
      DO 3 M=1,3
    3 RA(K,M)=AV(M)*COEFFR(I,J,1)+BV(M)*COEFFR(I,J,2)+CV(M)*COEFFR(I,J,3
     1)+DV(M)*COEFFR(I,J,4)
 1967 K=K+1
      RETURN
      END
      SUBROUTINE BLOKK(UV,VV,NBLOCK,G,NU,NV,I)
```

```
      DIMENSION UV(3),VV(3),B(75,5,3),CI2(3),CI3(3),CI5(3)
      DIMENSION NUMB2(75),AM(3),AMI1(3),AMI2(3),AMI3(3),RIMINA(3)
      COMMON/ONE/B
      COMMON/TWO/NUMB2
      EPI=1.0E-06
      IJ=1
      CALL SUB(AM,UV,VV)
    3 CONTINUE
      IF((NU.EQ.NUMB2(I)).OR.(NV.EQ.NUMB2(I)))GO TO 100
      DO 7 L=1,3
      CI2(L)=B(I,2,L)
      CI5(L)=B(I,5,L)
    7 CI3(L)=B(I,3,L)
      CALL DOT(DET,AM,CI5)
      IF(DET**2.LT.EPI) GO TO 100
      CALL CROSS(AMI1,AM,CI2)
      CALL CROSS(AMI2,CI3,AM)
      DO 8 M=1,3
      AMI1(M)=AMI1(M)/DET
      AMI2(M)=AMI2(M)/DET
      AMI3(M)=B(I,5,M)/DET
    8 RIMINA(M)=UV(M)-B(I,1,M)
      CALL DOT(V3,AMI3,RIMINA)
      IF((V 3 .GE.1.0).OR.(V 3 .LE.0.)) GO TO 100
      CALL DOT(V1,AMI1,RIMINA)
      IF(V 1 .LT.0.0) GO TO 100
      CALL DOT(V2,AMI2,RIMINA)
      IF(V 2 .LT.0.0) GO TO 100
      IF(((V 2 -1.0)*(1.0+B(I,4,2))-V 1 *B(I,4,1)).GT.0.0) GO TO 100
      IF(((V 1 -1.0)*(1.0+B(I,4,1))-V 2 *B(I,4,2)).GT.0.0) GO TO 100
      G=0.0
      GO TO 60
100   CONTINUE
      IF(IJ.EQ.NBLOCK) GO TO 98
      IF(I.EQ.NBLOCK) GO TO 10
      I=I+1
      GO TO 11
   10 I=1
   11 IJ=IJ+1
      GO TO 3
98    G=1.0
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SUB(C,A,B)
      DIMENSION C(3),A(3),B(3)
      DO 1 J=1,3
1     C(J)=A(J)-B(J)
      RETURN
      END
      SUBROUTINE CROSS(C,A,B)
      DIMENSION C(3),A(3),B(3)
      C(1)=A(2)*B(3)-B(2)*A(3)
      C(2)=B(1)*A(3)-A(1)*B(3)
      C(3)=A(1)*B(2)-B(1)*A(2)
      RETURN
      END
      SUBROUTINE DOT(C,A,B)
      DIMENSION A(3),B(3)
      C=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
      RETURN
      END
```

Projected Area Program

```
      PROGRAM PROJAR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PUNCH)
C     PROGRAM FOR PROJECTED AREAS OF PLANE QUADRILATERALS,ACCOUNTING FOR SHADING
C        ARRAY DIMENSIONS WILL BE AS FOLLOWS-
C MAIN PROGRAM- A(NPNL,4,3),B(NSHDR,5,3),NUMP(NPNL),NUMS(NSHDR),ETAS(NM),
C   PHIS(NN),RETAS(NM),RPHIS(NN),ARPAN(NPNL),XMUAV(NPNL,NM),AND UNRMVEC(NPNL,3),
C SUBROUTINE PLANE-Z(NSHDR,5,3)
C SUBROUTINE SHADE-B(NSHDR,5,3),NUMB2(NSHDR)
C         WHERE NPNL IS THE NUMBER OF PANELS,NSHDR IS THE NUMBER OF SHADERS,NM IS
C THE NUMBER OF VALUES OF ETA-S,AND NN IS THE NUMBER OF VALUES OF PHI-S.
C         IN THE PROJECTED AREA PROGRAM,NG IS THE NUMBER OF EQUAL SEGMENTS INTO
C WHICH EACH SIDE OF EACH PANEL IS DIVIDED TO FORM A GRID OF ELEMENTAL AREAS.
C IF NG IS DIFFERENT FROM 10,THE DIMENSIONS MUST BE DA(NG**2),RG(NG**2,3),
C COEFFA(NG,NG,4),COEFFR(NG,NG,4) IN THE PROGRAM-AND IN THE SUBROUTINE AREA,
C DA(NG**2),RA(NG**2,3),COEFFA(NG,NG,4),AND COEFFR(NG,NG,4).
C            DEFINITIONS OF VARIABLES-
C   COEFFA(M,N,K) IS ONE OF THE FOUR COEFFICIENTS IN THE FORMULA FOR THE AREA
C       OF GRID ELEMENT M,N OF A PANEL
C   COEFFR(M,N,K) IS THE COEFFICIENT OF THE COORDINATES OF THE KTH CORNER OF THE
C        PANEL IN THE FORMULA FOR THE CENTROID OF ELEMENT M,N
C NUMP IS AN ARRAY OF NUMBERS IDENTIFYING THE PLANE OF EACH PANEL
C NUMS IS A SIMILAR ARRAY FOR THE SHADERS
C A(I,J,K)GIVES THE CARTESIAN COORDINATES OF CORNER J(J=1 TO 4) OF PANEL I
C K=1 TO 3 CORRESPONDS TO X,Y,AND Z COORDINATES RESPECTIVELY.
C B GIVES THE SHADER COORDINATES.
C ETAS IS THE ARRAY OF ETA-S VALUES,PHIS OF THE PHI-S VALUES.
      DIMENSION A(50,4,3),B(60,5,3),NUMP(50),NUMS(60),ETAS(37)
     1,PHIS(72),C(4,3),C1(3),C2(3),C3(3),C4(3),VNORM(3),V2(3),SUN(3),
     2RETAS(37),RPHIS(72),DA(100),RG(100,3),RSUN(3),RGR(3)
      DIMENSION COEFFA(10,10,4),COEFFR(10,10,4)
      DIMENSION ARPAN(50),XMUAV(  19  ),UNRMVEC(50,3)
      DIMENSION VEE1(3),VEE2(3)
      DIMENSION XMUPHI(72)
      DIMENSION AXIS1(50,3),AXIS2(50,3)
      DIMENSION CSET(37),SNET(37),SNPH(72),CSPH(72)
      COMMON /ONE/B /TWO/NUMS
      COMMON /THREE/ COEFFA,COEFFR
   30 FORMAT(7I10)
   31 FORMAT(12F6.1)
      READ(5,30) NPNL,NSHDR,ISPIN
      READ(5,30) NG
      READ(5,30) (NUMP(I),I=1,NPNL)
      READ(5,30) (NUMS(I),I=1,NSHDR)
      READ(5,31) (((A(I,J,K),K=1,3),J=1,4),I=1,NPNL)
      READ(5,31) (((B(I,J,K),K=1,3),J=1,4),I=1,NSHDR)
```

```
      WRITE(6,32)
 32   FORMAT(1H 10HNO. PANELS11HNO. SHADERS)
      WRITE(6,30) NPNL,NSHDR
      WRITE(6,34)
 34   FORMAT(1H012HSIZE OF GRID)
      WRITE(6,30) NG
      WRITE(6,35)
 35   FORMAT(1H016HPANEL PLANE NOS.)
      WRITE(6,30) (NUMP(I),I=1,NPNL)
      WRITE(6,36)
 36   FORMAT(1H017HSHADER PLANE NOS.)
      WRITE(6,30) (NUMS(I),I=1,NSHDR)
      WRITE(6,37)
 37   FORMAT(1H017HPANEL COORDINATES)
 97   FORMAT(12F7.1)
      WRITE(6,97) (((A(I,J,K),K=1,3),J=1,4),I=1,NPNL)
      WRITE(6,38)
 38   FORMAT(1H018HSHADER COORDINATES)
      WRITE(6,97) (((B(I,J,K),K=1,3),J=1,4),I=1,NSHDR)
      NG2=NG*NG
      FOURN2=FLOAT(4*NG2)
      TWON4=FLOAT(2*NG2*NG2)
      DO 75 M=1,NG
      I1=M-1
      I21=I1+M
      NI=NG-M
      NI1=2*NI+1
      DO 75 N=1,NG
      J1=N-1
      J21=J1+N
      NJ=NG-N
      NJ1=2*NJ+1
      COEFFA(M,N,1)=FLOAT(NI*NJ+(NI+1)*(NJ+1))/TWON4
      COEFFA(M,N,2)=FLOAT(I1*(NJ+1)+M*NJ)/TWON4
      COEFFA(M,N,3)=FLOAT((NI+1)*J1+NI*N)/TWON4
      COEFFA(M,N,4)=FLOAT(I1*J1+M*N)/TWON4
      COEFFR(M,N,1)=FLOAT(NI1*NJ1)/FOURN2
      COEFFR(M,N,2)=FLOAT(I21*NJ1)/FOURN2
      COEFFR(M,N,3)=FLOAT(I21*J21)/FOURN2
 75   COEFFR(M,N,4)=FLOAT(NI1*J21)/FOURN2
      CALL PLANE(NSHDR)
 51    READ(5,30) NM,NN
      IF(EOF,5)998,999
 998  STOP
```

```
 999 CONTINUE
      READ(5,31) (ETAS(I),I=1,NM)
      READ(5,31) (PHIS(I),I=1,NN)
     WRITE(6,33)
  33 FORMAT(1H012HNO. OF ETA-S12HNO. OF PHI-S)
     WRITE(6,30) NM,NN
     WRITE(6,39)
  39 FORMAT(1H014HETA-SUN VALUES)
     WRITE(6,31) (ETAS(I),I=1,NM)
     WRITE(6,40)
  40 FORMAT(1H014HPHI-SUN VALUES)
     WRITE(6,31) (PHIS(I),I=1,NN)
     TOT=FLOAT(NN)
     ISHDR=1
     DO 150 LM=1,NM
     RETAS(LM)=ETAS(LM)*1.745329252E-2
     CSET(LM)=COS(RETAS(LM))
 150 SNET(LM)=SIN(RETAS(LM))
     DO 175 MN=1,NN
     RPHIS(MN)=PHIS(MN)*1.745329252E-2
     SNPH(MN)=SIN(RPHIS(MN))
 175 CSPH(MN)=COS(RPHIS(MN))
     DO 1 IP=1,NPNL
     NI=NUMP(IP)
     DO 2 JA=1,3
     DO 8 JC=1,4
   8 C(JC,JA)=A(IP,JC,JA)
     C1(JA)=C(2,JA)-C(1,JA)
     C2(JA)=C(4,JA)-C(1,JA)
     C3(JA)=C(4,JA)-C(3,JA)
   2 C4(JA)=C(2,JA)-C(3,JA)
     CALL CROSS(VNORM,C1,C2)
     CALL CROSS(V2,C3,C4)
     CALL DOT(D,VNORM,VNORM)
     CALL DOT(D1,V2,V2)
     D=D**.5
     DO 68 JQ=1,3
  68 UNRMVEC(IP,JQ)=VNORM(JQ)/D
     D1=D1**.5
     ARMAC=.5*(D+D1)
     ARPAN(IP)=ARMAC
     CALL AREA(C,DA,RG,NG,NG2)
     ARTOT=0.
     DO 13 J=1,NG2
```

```
   13 ARTOT=ARTOT+DA(J)
      WRITE(6,16) IP
   16 FORMAT(1H111H PANEL NO.=I2)
      WRITE(6,15) ARMAC
   15 FORMAT(1H020X,5HAREA=E16.8,2X,7HSQ. IN.)
      IF((ABS(ARTOT/ARMAC-1.)).GE..01) WRITE(6,14)
   14 FORMAT(1H037HAREAS DO NOT AGREE WITHIN ONE PERCENT)
      WRITE(6,25) ARTOT
   25 FORMAT(1H 23HSUM OF ELEMENTAL AREAS=E16.8,2X,7HSQ. IN.)
      DO 3 KN=1,3
    3 VNORM(KN)=VNORM(KN)/D
      DO 4 LM=1,NM
      SUM=0.
      WRITE(6,17)
   17 FORMAT(1H03X4HETAS4X4HPHIS3X6HARPROJ10X2HMU)
      DO 5 MN=1,NN
      ARPROJ=0.
      IF(ISPIN.EQ.1) GO TO 27
      GO TO 21
   27 SUN(1)=SNET(LM)*CSPH(MN)
      SUN(2)=SNET(LM)*SNPH(MN)
      SUN(3)=CSET(LM)
      CALL DOT(SDOTN,VNORM,SUN)
      IF(SDOTN.LE.0.) GO TO 23
      GO TO 47
   21 CONTINUE
      DO 29 K=1,3
   29 VEE1(K)=C1(K)
      SDOTN=CSET(LM)
      CALL DOT(VV1,VEE1,VEE1)
      VV1=VV1**.5
      DO 24 K=1,3
   24 VEE1(K)=VEE1(K)/VV1
      CALL CROSS(VEE2,VNORM,VEE1)
      DO 26 K=1,3
      AXIS1(IP,K)=VEE1(K)
      AXIS2(IP,K)=VEE2(K)
   26 SUN(K)=VNORM(K)*CSET(LM)+VEE1(K)*SNET(LM)*CSPH(MN)+VEE2(K)*SNET(LM
     1)*SNPH(MN)
   47 DO 7 N=1,3
    7 SUN(N)=1000.*SUN(N)
      DO 9 J=1,NG2
      DO 10 K=1,3
      RSUN(K)=RG(J,K)+SUN(K)
```

```
   10 RGR(K)=RG(J,K)
      CALL SHADE(RGR,RSUN,NSHDR,H,NI,ISHDR)
      IF(H.LT..5) GO TO 9
      ARPROJ=ARPROJ+DA(J)
    9 CONTINUE
      ARPROJ=ARPROJ*SDOTN
   23 XMU=ARPROJ/ARTOT
      WRITE(6,18) ETAS(LM),PHIS(MN),ARPROJ,XMU
      XMUPHI(MN)=XMU
   18 FORMAT(2F8.1,2E16.8)
      IF(ISPIN) 5,5,100
  100 SUM=SUM+ARPROJ
    5 CONTINUE
      IF(ISPIN) 77,77,78
   77 PUNCH 88,(XMUPHI(LP),LP=1,NN)
      GO TO 4
   78 ARPJBR=SUM/TOT
      XMUBAR=ARPJBR/ARTOT
      XMUAV(LM)=XMUBAR
      WRITE(6,19)
   19 FORMAT(1H015HMEAN PROJ. AREA,7X,10HAVERAGE MU)
      WRITE(6,20) ARPJBR,XMUBAR
   20 FORMAT(2E16.8)
    4 CONTINUE
      IF(ISPIN.EQ.0) GO TO 1
      PUNCH 88,(XMUAV(LM),LM=1,NM)
    1 CONTINUE
   88 FORMAT(7F11.8)
   49 FORMAT(5E16.9)
      PUNCH 49,(ARPAN(IP),IP=1,NPNL)
   50 FORMAT(6F13.9)
      IF(ISPIN) 72,72,73
   72 PUNCH 50,((UNRMVEC(IP,JQ),JQ=1,3),IP=1,NPNL)
      PUNCH 50,((AXIS1(IP,K),K=1,3),IP=1,NPNL)
      PUNCH 50,((AXIS2(IP,K),K=1,3),IP=1,NPNL)
   73 PUNCH 30, NM,NN
      PUNCH 31,(ETAS(M),M=1,NM)
      PUNCH 31,(PHIS(N),N=1,NN)
      GO TO 51
      END
      SUBROUTINE PLANE(NBLOCK)
      DIMENSION Z(60,5,3),C1(3),C2(3),C3(3),C4(3),C5(3),AM1(3),AM2(3),
     1V(3)
      COMMON/ONE/Z
```

```
      I=1
  5   DO 1 J=1,3
      C2(J)=Z(I,2,J)-Z(I,1,J)
  1   C3(J)=Z(I,4,J)-Z(I,1,J)
      CALL CROSS(C5,C2,C3)
      CALL CROSS(C1,C3,C5)
      CALL CROSS(AM2,C5,C2)
      CALL DOT(DET,C1,C2)
      DO 2 K=1,3
      AM1(K)=C1(K)/DET
    2 AM2(K)=AM2(K)/DET
      DO 3 J=1,3
  3   V(J)=Z(I,1,J)+Z(I,3,J)-Z(I,2,J)-Z(I,4,J)
      C4(1)=AM 1(1)*V(1)+AM 1(2)*V(2)+AM 1(3)*V(3)
      C4(2)=AM 2(1)*V(1)+AM 2(2)*V(2)+AM 2(3)*V(3)
      C4(3)=0.
      DO 6 J=1,3
      Z(I,2,J)=C2(J)
      Z(I,3,J)=C3(J)
      Z(I,4,J)=C4(J)
  6   Z(I,5,J)=C5(J)
      I=I+1
      IF(I.LT.(NBLOCK+1)) GO TO 5
      RETURN
      END
C   SUBR. PLANE OPERATES ON ARRAY Z (BODY COORDINATES OF THE 4 CORNERS OF EACH
C      SHADER) TO FIND PARAMETERS NEEDED BY THE SHADING SUBR.(SHADE)
      SUBROUTINE AREA(A,DA,RA,NG,NG2)
      DIMENSION A(4,3),DA(100),RA(100,3),AV(3),BV(3),CV(3),DV(3)
     1,T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3),T8(3)
      DIMENSION COEFFA(10,10,4),COEFFR(10,10,4)
      COMMON /THREE/ COEFFA,COEFFR
 18   FORMAT(4E16.8)
      DO 2 N=1,3
       AV(N)=A(1,N)
      BV(N)=A(2,N)
      CV(N)=A(3,N)
      DV(N)=A(4,N)
      T1(N)=BV(N)-AV(N)
      T2(N)=CV(N)-BV(N)
      T3(N)=CV(N)-DV(N)
    2 T4(N)=DV(N)-AV(N)
      CALL CROSS(T5,T4,T1)
      CALL CROSS(T6,T2,T1)
```

```
      CALL CROSS(T7,T4,T3)
      CALL CROSS(T8,T2,T3)
      CALL DOT(Z1,T5,T5)
      CALL DOT(Z2,T6,T6)
      CALL DOT(Z3,T7,T7)
      CALL DOT(Z4,T8,T8)
      Z1=Z1**.5
      Z2=Z2**.5
      Z3=Z3**.5
      Z4=Z4**.5
      K=1
      DO 1967 I=1,NG
      DO 1967 J=1,NG
      DA(K)=Z1*COEFFA(I,J,1)+Z2*COEFFA(I,J,2)+Z3*COEFFA(I,J,3)+Z4*COEFFA
     1(I,J,4)
      DO 3 M=1,3
    3 RA(K,M)=AV(M)*COEFFR(I,J,1)+BV(M)*COEFFR(I,J,2)+CV(M)*COEFFR(I,J,3
     1)+DV(M)*COEFFR(I,J,4)
 1967 K=K+1
      RETURN
      END
C    SUBR. AREA,USING THE COORDINATES OF EACH CORNER AND THE ARRAYS COEFFA AND
C        COEFFR,COMPUTES THE COORDINATES OF THE CENTROID AND THE AREA OF EACH GRID
C        ELEMENT OF A PANEL.
      SUBROUTINE SHADE(UV,VV,NBLOCK,G,NU,I)
      DIMENSION UV(3),VV(3),B(60,5,3),CI2(3),CI3(3),CI5(3)
      DIMENSION NUMB2(60),AM(3),AMI1(3),AMI2(3),AMI3(3),AMINX(3)
      COMMON/ONE/B
      COMMON/TWO/NUMB2
      EPI=1.0E-06
      IJ=1
      CALL SUB(AM,VV,UV)
    3 CONTINUE
      IF (NU.EQ.NUMB2(I))GO TO 100
      DO 7 L=1,3
      CI2(L)=B(I,2,L)
      CI5(L)=B(I,5,L)
    7 CI3(L)=B(I,3,L)
      CALL DOT(DET,AM,CI5)
      IF(DET**2.LT.EPI) GO TO 100
      CALL CROSS(AMI1,AM,CI3)
      CALL CROSS(AMI2,CI2,AM)
      DO 8 M=1,3
      AMI1(M)=AMI1(M)/DET
```

```
      AMI2(M)=AMI2(M)/DET
      AMI3(M)=B(I,5,M)/DET
  8   AMINX(M)=B(I,1,M)-UV(M)
      CALL DOT(V3,AMI3,AMINX)
      IF((V 3 .GE.1.0).OR.(V 3 .LE.0.)) GO TO 100
      CALL DOT(V1,AMI1,AMINX)
      IF(V 1 .LT.0.0) GO TO 100
      CALL DOT(V2,AMI2,AMINX)
      IF(V 2 .LT.0.0) GO TO 100
      IF(((V1-1.)*(1.+B(I,4,2))-V2*B(I,4,1)).GT.0.) GO TO 100
      IF(((V2-1.)*(1.+B(I,4,1))-V1*B(I,4,2)).GT.0.) GO TO 100
      G=0.0
      GO TO 60
100   CONTINUE
      IF(IJ.EQ.NBLOCK) GO TO 98
      IF(I.EQ.NBLOCK) GO TO 10
      I=I+1
      GO TO 11
   10 I=1
   11 IJ=IJ+1
      GO TO 3
 98   G=1.0
 60   CONTINUE
      RETURN
      END
C    SUBR. SHADE DETERMINES WHETHER THE LINE BETWEEN TWO GIVEN POINTS INTERSECTS
C      ANY OF THE SHADERS.
      SUBROUTINE SUB(C,A,B)
      DIMENSION A(3),B(3),C(3)
      DO 1 I=1,3
    1 C(I)=A(I)-B(I)
      RETURN
      END
C    SUBR. SUB GIVES VECTOR C=B-A
      SUBROUTINE CROSS(C,A,B)
      DIMENSION A(3),B(3),C(3)
      C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END
C    SUBR. CROSS GIVES VECTOR C=CROSS PRODUCT OF VECTORS A AND B
      SUBROUTINE DOT(C,A,B)
      DIMENSION A(3),B(3)


      C=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
      RETURN
      END
C    SUBR. DOT GIVES C=DOT PRODUCT OF VECTORS A AND B.
```

APPENDIX B

DERIVATION OF CENTROID AND AREA FORMULAS FOR ELEMENTAL GRID SECTIONS OF A PLANE QUADRILATERAL

The plane quadrilateral is divided into an $n \times n$ grid by dividing each side into n equal segments and connecting the corresponding dividing points on opposite sides. (See fig. 6.) Counterclockwise from the lower left, the position vectors of the vertexes are $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$. Any vertex may be taken as the starting point, so long as the order is counterclockwise. This gives the proper sense to the computed normal vector to the plane.

The lower left corner of each elemental area is denoted by (j,k), where j and k are integers increasing from 1 to n; j, from left to right; and k, from bottom to top. Now, each segment of the line between $\bar{A}$ and $\bar{B}$ represents a change in position of $\frac{\bar{B} - \bar{A}}{n}$. Thus, the position vector of the point (j, 1) is given by

$$\bar{V}_{j,1} = \bar{A} + (j - 1)\frac{\bar{B} - \bar{A}}{n} = \frac{1}{n}\left[(n - j + 1)\bar{A} + (j - 1)\bar{B}\right] \tag{B1}$$

Similarly, along the line from $\bar{D}$ to $\bar{C}$ (where $k = n + 1$),

$$\bar{V}_{j,n+1} = \frac{1}{n}\left[(n - j + 1)\bar{D} + (j - 1)\bar{C}\right] \tag{B2}$$

The position vectors of the points (1,k) and (n+1,k) correspondingly will be given by

$$\bar{V}_{1,k} = \frac{1}{n}\left[(n - k + 1)\bar{A} + (k - 1)\bar{D}\right] \tag{B3}$$

$$\bar{V}_{n+1,k} = \frac{1}{n}\left[(n - k + 1)\bar{B} + (k - 1)\bar{C}\right] \tag{B4}$$

If a line is drawn between $\bar{V}_{j,1}$ and $\bar{V}_{j,n+1}$ and another is drawn between $\bar{V}_{1,k}$ and $\bar{V}_{n+1,k}$, their intersection will be $\bar{V}_{j,k}$, the position vector of the lower left corner of element jk. The line between $\bar{V}_{j,1}$ and $\bar{V}_{j,n+1}$ is divided by the intersection point in the same ratio as sides BC and AD are divided. Thus, in terms of $\bar{V}_{j,1}$ and $\bar{V}_{j,n+1}$, $\bar{V}_{j,k}$ may be expressed as

$$\bar{V}_{j,k} = \frac{1}{n}\left[(n - k + 1)\bar{V}_{j,1} + (k - 1)\bar{V}_{j,n+1}\right] \tag{B5}$$

Substituting equations (B1) and (B2) into equation (B5) yields

$$\bar{V}_{j,k} = \frac{1}{n^2}\Big[(n - j + 1)(n - k + 1)\bar{A} + (j - 1)(n - k + 1)\bar{B}$$

$$+ (j - 1)(k - 1)\bar{C} + (\bar{n} - j + 1)(k - 1)\bar{D}\Big] \tag{B6}$$

This is the exact value of the position vector of the lower left corner of elemental area jk of the quadrilateral. The position vector of the centroid of the elemental quadrilateral may, except for extremely irregular shapes, be well approximated by

$$\left(\bar{V}_c\right)_{j,k} \approx \bar{V}_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{n^2}\Big[\left(n - j + \frac{1}{2}\right)\left(n - k + \frac{1}{2}\right)\bar{A} + \left(j - \frac{1}{2}\right)\left(n - k + \frac{1}{2}\right)\bar{B}$$

$$+ \left(j - \frac{1}{2}\right)\left(k - \frac{1}{2}\right)\bar{C} + \left(n - j + \frac{1}{2}\right)\left(k - \frac{1}{2}\right)\bar{D}\Big] \tag{B7}$$

This is the mean of the position vectors of the four corners of the grid element and would also be the point of intersection if the grid were twice as fine.

The area of the element jk is found by dividing it into two triangles and taking half the magnitude of the cross product of two sides of each triangle. Since the upper right triangle of element jk is congruent to the lower left triangle of element j+1, k+1, the area of element jk is given by

$$(\Delta A)_{j,k} = \frac{1}{2}\left|\left(\bar{V}_{j+1,k} - \bar{V}_{j,k}\right) \times \left(\bar{V}_{j,k+1} - \bar{V}_{j,k}\right)\right|$$

$$+ \frac{1}{2}\left|\left(\bar{V}_{j+2,k+1} - \bar{V}_{j+1,k+1}\right) \times \left(\bar{V}_{j+1,k+2} - \bar{V}_{j+1,k+1}\right)\right| \tag{B8}$$

Substituting equation (B6) into equation (B8) yields

$$(\Delta A)_{j,k} = \frac{1}{2n^4}\Big|\left[(n - k + 1)(\bar{B} - \bar{A}) + (k - 1)(\bar{C} - \bar{D})\right] \times \left[(n - j + 1)(\bar{D} - \bar{A}) + (j - 1)(\bar{C} - \bar{B})\right]$$

$$+ \left[(n - k)(\bar{B} - \bar{A}) + k(\bar{C} - \bar{D})\right] \times \left[(n - j)(\bar{D} - \bar{A}) + j(\bar{C} - \bar{B})\right]\Big| \tag{B9}$$

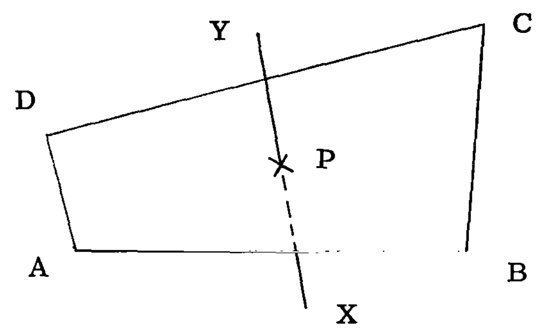
Expanding equation (B9) gives

$$
\begin{aligned}
(\Delta A)_{j,k} = \frac{1}{2n^4} \Bigg\{ & \Big[(n - j + 1)(n - k + 1) + (n - j)(n - k)\Big] \Big|(\bar{B} - \bar{A}) \times (\bar{D} - \bar{A})\Big| \\
& + \Big[(j - 1)(n - k + 1) + j(n - k)\Big] \Big|(\bar{B} - \bar{A}) \times (\bar{C} - \bar{B})\Big| \\
& + \Big[(n - j + 1)(k - 1) + (n - j)k\Big] \Big|(\bar{C} - \bar{D}) \times (\bar{D} - \bar{A})\Big| \\
& + \Big[(j - 1)(k - 1) + jk\Big] \Big|(\bar{C} - \bar{D}) \times (\bar{C} - \bar{B})\Big| \Bigg\}
\end{aligned}
\tag{B10}
$$

APPENDIX C

DETERMINATION OF SHADING

Five tests are made in the computer program to determine whether points X and Y are shaded from one another by the planar quadrilateral ABCD:



The first test determines whether the point of intersection P of the plane of ABCD by the line XY falls between points X and Y. Then for each of the four sides of ABCD, it is found whether P lies toward the inside or the outside of ABCD from that side.

The position vector of a point on the plane of the quadrilateral ABCD may be expressed as a linear combination of two vectors in the plane added to the position vector of a corner, say corner A:

$$\bar{r}_{plane} = \bar{A} + \alpha'(\bar{B} - \bar{A}) + \beta'(\bar{D} - \bar{A})$$

If $\bar{P}$ is the position vector of the point of intersection of the plane by the line connecting X and Y, then

$$\bar{P} - \bar{X} = \gamma(\bar{Y} - \bar{X})$$

Setting $\bar{P} = \bar{r}_{plane}$ gives

$$\alpha'(\bar{A} - \bar{B}) + \beta'(\bar{A} - \bar{D}) + \gamma(\bar{Y} - \bar{X}) = (\bar{A} - \bar{X})$$

or

$$\alpha'\bar{V}_1 + \beta'\bar{V}_2 + \gamma\bar{V}_3 = \bar{V}_4$$

This vector equation is, of course, three simultaneous linear equations in α', β', and γ – one for each vector component.

The solution for γ will be carried out first, since the quadrilateral ABCD will be immediately eliminated as a shader if the condition $0 < \gamma < 1$ is not met:

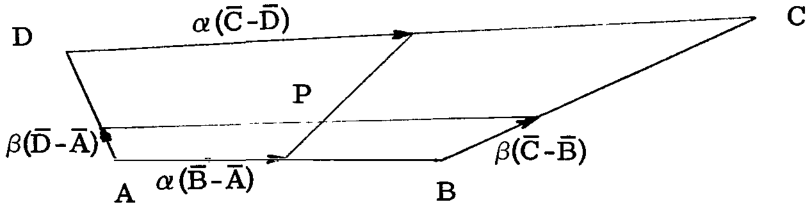
$$\gamma = \frac{\begin{vmatrix} (V_1)_X & (V_2)_X & (V_4)_X \\ (V_1)_Y & (V_2)_Y & (V_4)_Y \\ (V_1)_Z & (V_2)_Z & (V_4)_Z \end{vmatrix}}{\begin{vmatrix} (V_1)_X & (V_2)_X & (V_3)_X \\ (V_1)_Y & (V_2)_Y & (V_3)_Y \\ (V_1)_Z & (V_2)_Z & (V_3)_Z \end{vmatrix}} = \frac{\bar{V}_1 \cdot \left(\bar{V}_2 \times \bar{V}_4\right)}{\bar{V}_1 \cdot \left(\bar{V}_2 \times \bar{V}_3\right)}$$

If $\alpha' < 0$, the point lies to the left of side AD and if $\beta' < 0$, the point lies below side AB. These will also be evaluated one at a time:

$$\alpha' = \frac{\bar{V}_4 \cdot \left(\bar{V}_2 \times \bar{V}_3\right)}{\bar{V}_1 \cdot \left(\bar{V}_2 \times \bar{V}_3\right)} \qquad \beta' = \frac{\bar{V}_1 \cdot \left(\bar{V}_4 \times \bar{V}_3\right)}{\bar{V}_1 \cdot \left(\bar{V}_2 \times \bar{V}_3\right)}$$

If α' and β' are both greater than zero, more testing is required.

The reason for the primes is that α' and β' are actually quadratics in the two linear parameters for the skewed coordinate frame formed by the quadrilateral. Let α be a linear parameter characterizing a point moving from A to B or D to C as α varies from 0 to 1. Let β be the parameter for points along AD or BC as shown in the following sketch:



Then,

$$\bar{P} = \bar{A} + \alpha(\bar{B} - \bar{A}) + \beta\left\{\left[\bar{D} + \alpha(\bar{C} - \bar{D})\right] - \left[\bar{A} + \alpha(\bar{B} - \bar{A})\right]\right\}$$

$$= \bar{A} + \alpha(\bar{B} - \bar{A}) + \beta(\bar{D} - \bar{A}) + \alpha\beta(\bar{A}-\bar{B}+\bar{C}-\bar{D})$$

The vector $(\bar{A}-\bar{B}+\bar{C}-\bar{D})$ lies in the plane of the quadrilateral and thus can be given as a linear combination of $(\bar{B} - \bar{A})$ and $(\bar{D} - \bar{A})$: $(\bar{A}-\bar{B}+\bar{C}-\bar{D}) = \lambda_1(\bar{B} - \bar{A}) + \lambda_2(\bar{D} - \bar{A})$. The parameters λ_1 and λ_2 may be found by use of the vectors reciprocal to $(\bar{B} - \bar{A})$ and $(\bar{D} - \bar{A})$, denoted by superscript R and characterized by

$$(\bar{B} - \bar{A})^R \cdot (\bar{B} - \bar{A}) = 1$$

$$(\bar{B} - \bar{A})^R \cdot (\bar{D} - \bar{A}) = 0$$

$$(\bar{D} - \bar{A})^R \cdot (\bar{B} - \bar{A}) = 0$$

$$(\bar{D} - \bar{A})^R \cdot (\bar{D} - \bar{A}) = 1$$

where, if $\bar{N} = (\bar{B} - \bar{A}) \times (\bar{D} - \bar{A})$, then

$$(\bar{B} - \bar{A})^R = \frac{(\bar{D} - \bar{A}) \times \bar{N}}{\left[(\bar{D} - \bar{A}) \times \bar{N}\right] \cdot (\bar{B} - \bar{A})}$$

$$(\bar{D} - \bar{A})^R = \frac{\bar{N} \times (\bar{B} - \bar{A})}{\left[\bar{N} \times (\bar{B} - \bar{A})\right] \cdot (\bar{D} - \bar{A})}$$

Now,

$$(\bar{B} - \bar{A})^R \cdot (\bar{A}-\bar{B}+\bar{C}-\bar{D}) = \lambda_1$$

and

$$(\bar{D} - \bar{A})^R \cdot (\bar{A}-\bar{B}+\bar{C}-\bar{D}) = \lambda_2$$

Thus,

$$\bar{P} = \bar{A} + \alpha(\bar{B} - \bar{A}) + \beta(\bar{D} - \bar{A}) + \alpha\beta\left[\lambda_1(\bar{B} - \bar{A}) + \lambda_2(\bar{D} - \bar{A})\right]$$

or

$$\bar{P} = \bar{A} + \alpha\left(1 + \lambda_1\beta\right)(\bar{B} - \bar{A}) + \beta\left(1 + \lambda_2\alpha\right)(\bar{D} - \bar{A})$$

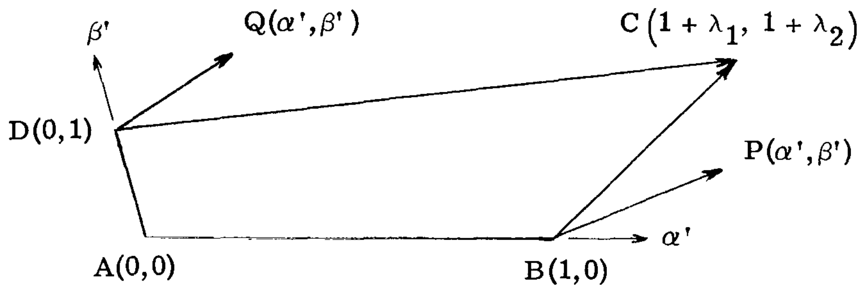
and

$$\alpha' = \left(1 + \lambda_1\beta\right)\alpha$$

$$\beta' = \left(1 + \lambda_2\alpha\right)\beta$$

from which the values of α and β can be found directly.

An advantageous alternative to finding α and β is found by plotting the quadrilateral ABCD in α',β' coordinates:



The α' and β' coordinates of the corners are found from the two equations for α' and β' in terms of α and β, with α and β having a value of 0 or 1 at the corners.

The points P and Q represent possible points of intersection between the plane of ABCD and a line connecting two points of interest. If point P lies on the right-hand side of line BC, then the single nonzero component of the cross product of vector $(\bar{P} - \bar{B})$ upon $(\bar{C} - \bar{B})$ will be positive. Thus for no shading, this third component will be given by

$$\left[(\bar{P} - \bar{B}) \times (\bar{C} - \bar{B})\right]_3 = (\alpha' - 1)\left(1 + \lambda_2\right) - \beta'\lambda_1 > 0$$

If this quantity is zero or negative, $\bar{P}$ lies on or to the left of line BC. Similarly, the condition that the point lie above line CD is

$$\left[(\bar{C} - \bar{D}) \times (\bar{Q} - \bar{D})\right]_3 = \left(1 + \lambda_1\right)(\beta' - 1) - \lambda_2\alpha' > 0$$

The five tests for shading, in the order of execution, are thus

(1) $0 < \gamma < 1$

(2) $\alpha' \geqq 0$

(3) $\beta \geqq 0$

(4) $(\alpha' - 1)(1 + \lambda_2) - \beta' \cdot \lambda_1 \leqq 0$

(5) $(\beta' - 1)(1 + \lambda_1) - \alpha' \cdot \lambda_2 \leqq 0$

If any one of these fails, there is no shading between the two points by the quadrilateral ABCD. They must all hold for shading to occur.

REFERENCES

1. Eckert, E. R. G.; and Drake, Robert M., Jr.: Heat and Mass Transfer. McGraw-Hill Book Co., Inc., 1959.

2. Goerss, R. H.: Computer Analysis of Satellite Thermal Behavior. Spacecraft and Space Systems, Radio Corp. Amer., 1966, pp. 30-33.

3. Gebhart, Benjamin: Heat Transfer. McGraw-Hill Book Co., Inc., c.1961.

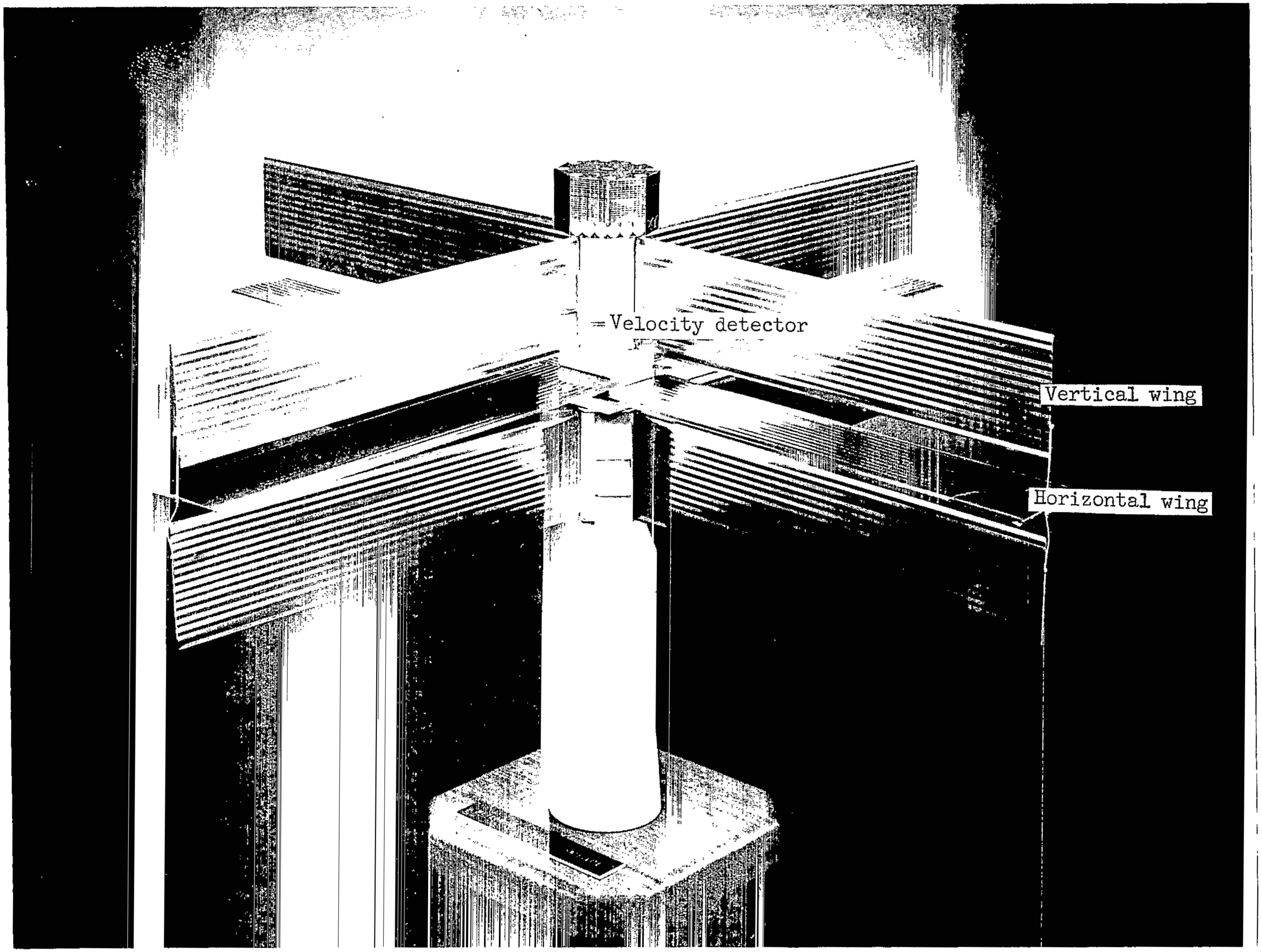


Figure 1.- Location of three sets of sample nodes on the Meteoroid Technology Satellite. L-69-657

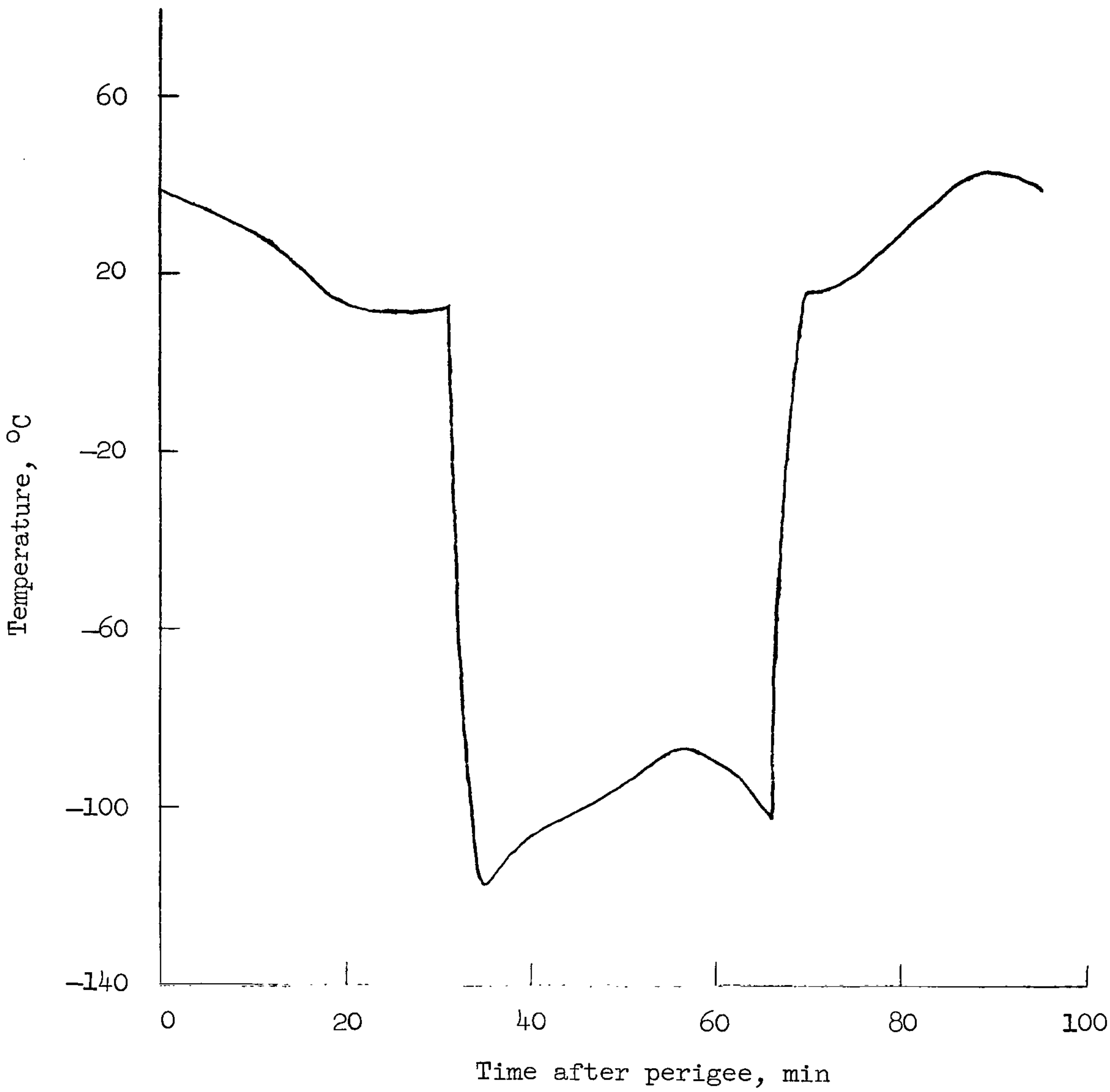


Figure 2.- Temperature history of outer face of upper velocity detector.

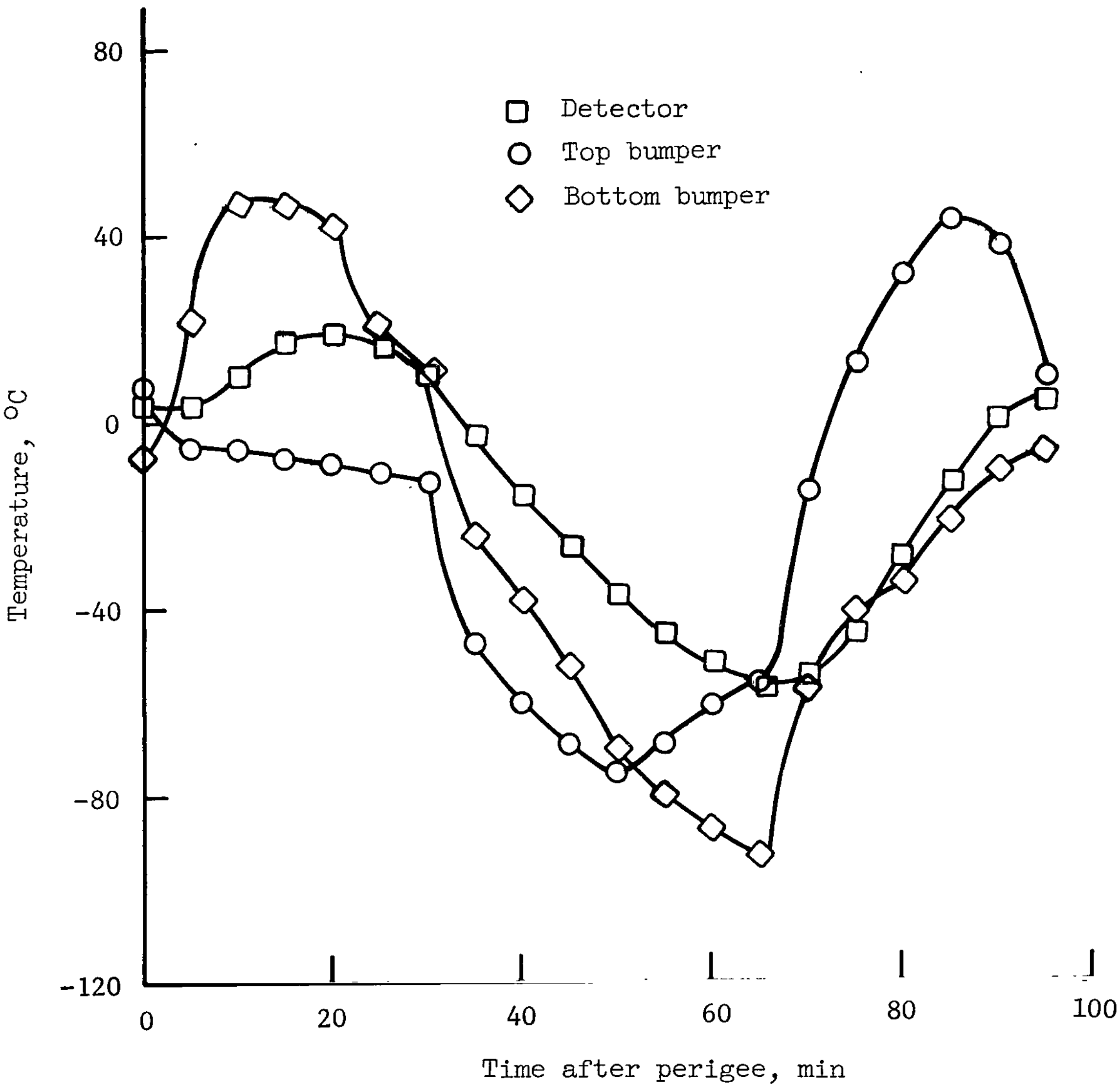


Figure 3.- Temperature histories of horizontal wing detector and its bumper shields.

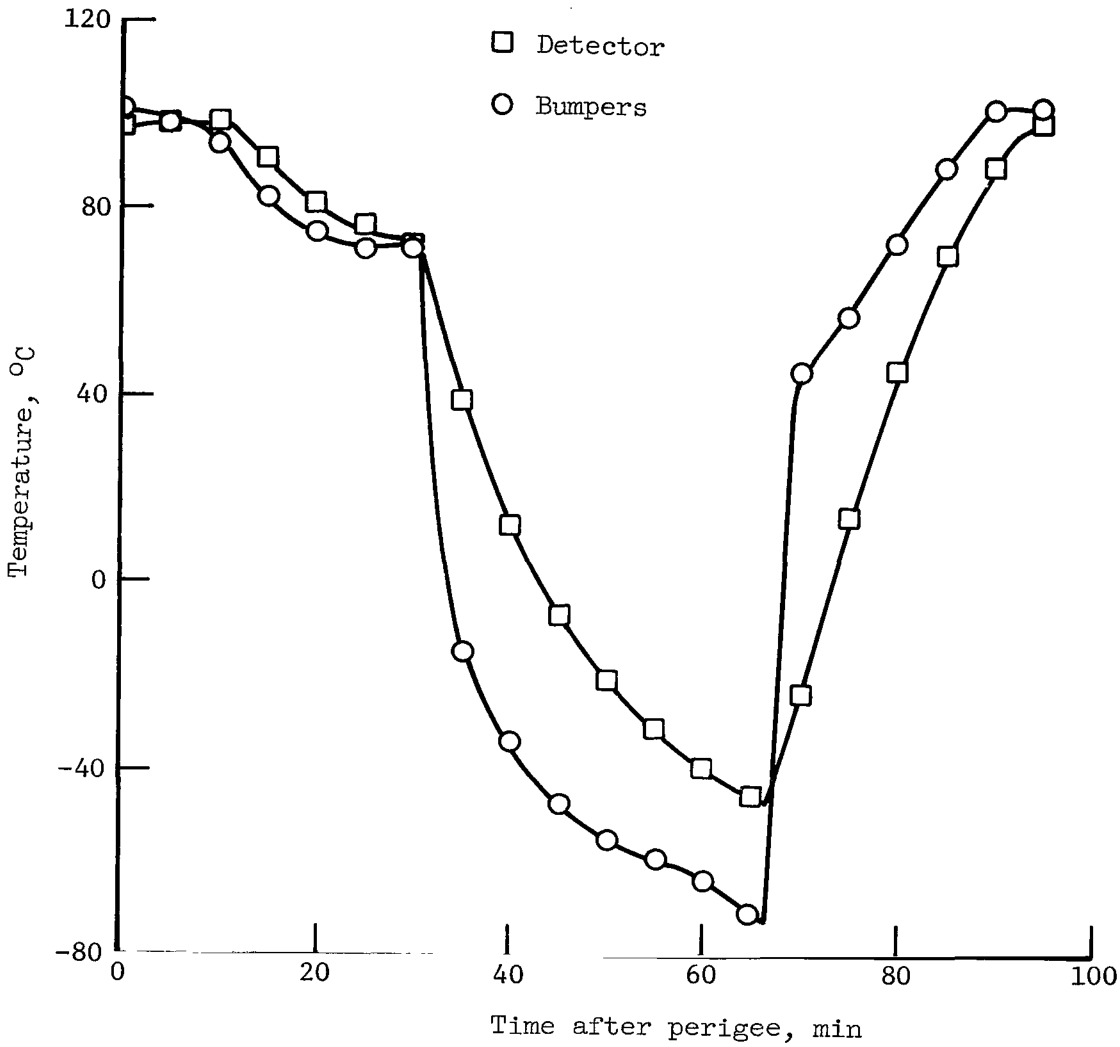


Figure 4.- Temperature history of vertical wing detector and its bumper shields.

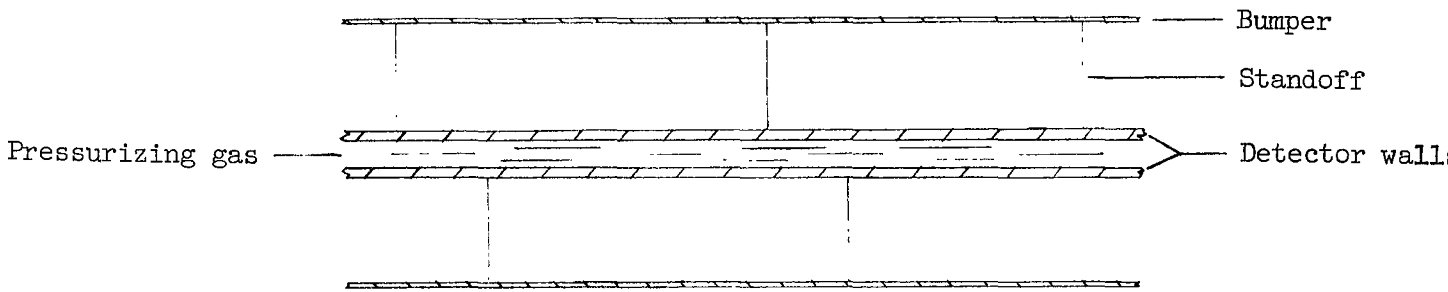


Figure 5.- Schematic cross section of a pressurized-cell meteoroid detector with bumpers.

NASA-Langley, 1970 — 33 L-6063

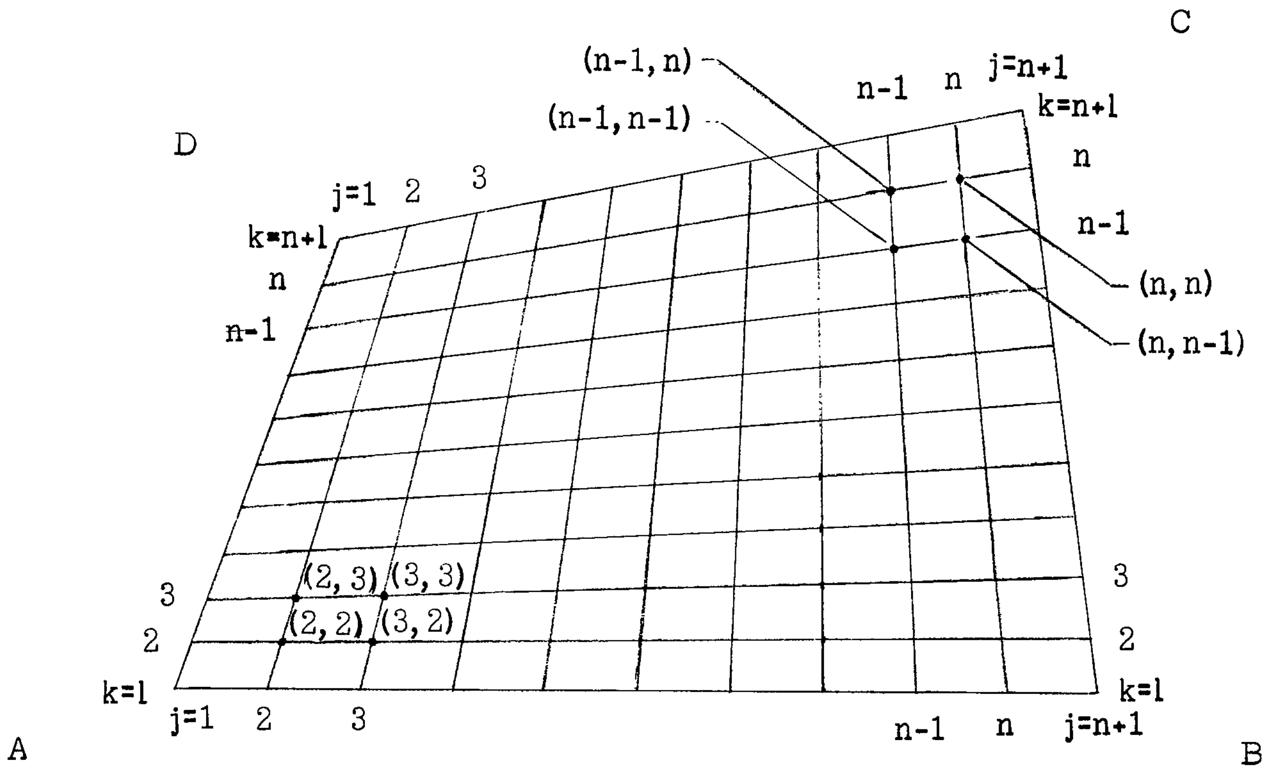


Figure 6.- Plane quadrilateral with $n \times n$ grid formed by dividing each side into n equal segments, with coordinate parameters j and k illustrated.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

05U 001 58 51 3DS 70195 00903
AIR FORCE WEAPONS LABORATORY /WLOL/
KIRTLAND AFB, NEW MEXICO 87117

ATT E. LOU BOWMAN, CHIEF,TECH. LIBRARY

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546