RADIATIVE INPUT to a
HIGH ALTITUDE OBJECT
CALCULATED USING GOES IR DATA
Radiative Input to a High Altitude Object
Calculated Using GOES IR Data

Prepared by

Space Science and Engineering Center
University of Wisconsin-Madison
Madison, Wisconsin 53706

March 1982

Barry B. Hinton
INTRODUCTION

a. **Idealized Problem**

The objective is to compute the thermal radiation input to a body at high altitudes above the earth. For technical reasons the method developed can be validly applied when the altitudes are much greater than eight kilometers. For an object at such altitudes essentially all thermal radiation due to the atmosphere (including both clouds and radiating gasses such as water vapor and carbon dioxide) comes from below. Certainly for altitudes, as considered here, that exceed 80 km it is a very good approximation to imagine that all the thermal radiative input originates from a surface below the object. Thus, a single surface incorporates the radiative properties of both the atmosphere and the earth's surface.

![Diagram showing radiation from earth's surface, cloud, and gasses](image)

To obtain the total input the earth below is divided into many small patches, the strength of the radiation emanating from each determined (per unit area of the patch). The product of the radiating strength and area of each of these patches (or area elements) is then added—if it can be seen from the test object or
body. In this summation weight is also given to the distance from the emitter and test body according to the "inverse square law".

Account must also be taken of the relative look angles between the test body and the area element ("cosine law") since a flat emitter viewed edgewise transfers less energy to a second object than the same emitter viewed straight on from the same distance. That is, a receives less than b from e.

\[ \text{In order to divide the surface below into the small patches an image from GOES is used.}^* \text{ That is, the pixels as defined by the GOES 10.5 micron IR channel (or multiples of them) are the elements of area we have used. The digital count value of each pixel is transformed to a brightness temperature (at 10.5\,\mu m), and the brightness temperature used with an empirical model developed by Smith et al. to obtain the strength of the radiation across the entire spectrum.}^{**} \]


b. Results

We have presented below in Table 1 the results which were obtained using the method discussed in more detail below. In a later section these results are also shown in graphical form. Note that for \( t > 1359 \) the results are given at 10 second intervals.

<table>
<thead>
<tr>
<th>Time(s)</th>
<th>Flux (Wm(^{-2}))</th>
<th>Time(s)</th>
<th>Flux (Wm(^{-2}))</th>
<th>Time(s)</th>
<th>Flux (Wm(^{-2}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1159</td>
<td>207.0</td>
<td>1309</td>
<td>216.5</td>
<td>1399</td>
<td>211.0</td>
</tr>
<tr>
<td>1169</td>
<td>207.7</td>
<td>1319</td>
<td>217.8</td>
<td>1409</td>
<td>208.6</td>
</tr>
<tr>
<td>1179</td>
<td>208.6</td>
<td>1329</td>
<td>218.7</td>
<td>1419</td>
<td>205.8</td>
</tr>
<tr>
<td>1189</td>
<td>209.5</td>
<td>1339</td>
<td>218.0</td>
<td>1429</td>
<td>202.2</td>
</tr>
<tr>
<td>1199</td>
<td>210.5</td>
<td>1349</td>
<td>216.7</td>
<td>1439</td>
<td>197.8</td>
</tr>
<tr>
<td>1209</td>
<td>211.3</td>
<td>1359</td>
<td>217.4</td>
<td>1449</td>
<td>193.1</td>
</tr>
<tr>
<td>1229</td>
<td>213.0</td>
<td>1369</td>
<td>215.4</td>
<td>1459</td>
<td>188.6</td>
</tr>
<tr>
<td>1259</td>
<td>214.5</td>
<td>1379</td>
<td>214.3</td>
<td>1469</td>
<td>185.6</td>
</tr>
<tr>
<td>1279</td>
<td>216.3</td>
<td>1389</td>
<td>213.9</td>
<td>1479</td>
<td>185.3</td>
</tr>
</tbody>
</table>

1 0.3660 should be added to every time entry.

The word flux is used for power per unit area rather than a more precise, but cumbersome, term such as flux density.

Mathematical Description

In this section we define the geometrical quantities and derive the expressions required to program the algorithm outlined in the previous section. Matters pertaining to radiative
quantities (e.g. emissivity, Plank function, etc.) are referred to
the books of Liou and Fleagle and Businger.* Following this is a
listing of the computing program. Parts of the program are not
easy to understand, because much use is made of the McIDAS system
subroutines and related terminology. Documentation for these is
unpublished material obtainable from the University of Wisconsin
Space Science and Engineering Center program librarian.

Figure 1 describes the transformation from latitude,
longitude and altitude coordinates to a cartesian system (X, Y,
Z). Figure 2 is intended to illustrate the general relationship
of the test body (located at the "observation point") to a patch
(or pixel) of area dA.

Let \( \mathbf{r}_p \) be the vector OP. The earth centered cartesian
coordinates are given by \( \mathbf{r}_p = (X_p, Y_p, Z_p) \) where,
\[
X_p = (a + \xi) (\cos \theta_p \cos \phi_p)
\]
\[
Y_p = (a + \xi) (\cos \theta_p \sin \phi_p)
\]
\[
Z_p = (a + \xi) \sin \theta_p
\]
The energy which crosses dA in time dt and which is confined
to the solid angle d \( \Omega \) (see Fig. 3) is:

\[
dE = I_\lambda \cos \theta dA \Omega dt d\lambda \tag{1}
\]

where \( I_\lambda \) is the intensity. If A emits isotropically, we may take

* Liou, K-N, An Introduction to Atmospheric Radiation, Academic
Fleagle, R. G. and J. A. Businger, An Introduction to

** McIDAS stands for Man-Computer Interactive Data Access System.
Figure 1. Earth centered cartesian coordinate system. An arbitrary point, \( p \), is shown at an altitude \( \zeta \) above the earth of radius \( a \).
Figure 2. General relationships of GOES, observation point (0) and a patch on the surface (dA).
\[ I_\lambda = \varepsilon_\lambda B_\lambda (T) \]  

(2)

Where \( \varepsilon_\lambda \) is the emissivity and \( B_\lambda \) is the radiant intensity, or Plank, function of a black body. Thus, if \( \varepsilon B = \int \varepsilon_\lambda B_\lambda (\lambda) d\lambda \)  

(1) becomes,

\[ dE = \varepsilon B \cos \theta \, dA \, d\Omega \, dt \]  

(3)

In this equation \( E \) is an 'effective' or mean emissivity.

The energy incident on \( dA_0 \) from \( dA \) is the intensity emitted by \( dA \) within the solid angle \( d\Omega_0 \) times \( dA_0 \) and \( dt \) as well as \( d\Omega \) itself See Figure 4.

\[ dE = \varepsilon B \cos \theta \, dA \, dt \, d\Omega_0 \]  

(3')

However, in this case, \( d\Omega_0 \) can also be represented as,

\[ d\Omega_0 = dA_0 \cos \theta_0 / r^2_{Ao} \]

where \( r_{Ao} \) is the distance from \( dA \) to \( dA_0 \). Consequently

\[ \frac{dE}{dt} = \frac{dA \cos \theta \, dA_0 \cos \theta_0 \, \varepsilon B}{r^2_{Ao}} \]  

(4)

Actually, it is best to recast this equation in a vector notation. Let \( \hat{r}_A \) be the position vector of the element \( dA \) and \( \hat{r}_O \) that of \( dA_0 \).
Also, let \( \hat{r}_{ao} \) be a vector from \( da \) to \( da_0 \), and \( r_{oa} = -\hat{r}_{ao} \) a vector from \( da_0 \) to \( da \). We may therefore write (4) as,

\[
\frac{\partial E}{\partial t} = \frac{da(\vec{n} \cdot \hat{r}_{ao}) da_0(\vec{n}_0 \cdot \hat{r}_{oa}) \xi B}{(r_{ao})^4}
\]

or,

\[
\frac{\partial E}{\partial t} = \frac{da(\vec{n}_A \cdot \hat{r}_{ao}) da_0(\vec{n}_o \cdot \hat{r}_{oa}) \xi B}{(r_{ao})^2}
\]

As can be seen in Figures 3 and 4, \( n_A \) is the unit normal to \( da \) and the unit normal to \( da_0 \), \( \hat{r}_{ao} = \vec{r}_{ao}/|\vec{r}_{ao}| \) and \( \hat{r}_{oa} = \vec{r}_{oa}/|\vec{r}_{oa}| \).

In the program at hand \( da \) is a unit area, i.e. \( da = 1 \), \( da \) is the variable AREA, \( (X0, Y0, Z0) = \vec{r}_o \) \( (XA, YA, ZA) = \vec{r}_A \), \( \vec{r}_o = \vec{r}_o \vec{r}_o \), \( \hat{r}_{ao} = (XUOA, YUOA, ZUOA), \hat{r}_{oa} = (XUOA, YUOA, ZUOA) \).

In this particular application, both \( da_0 \) and \( da \) are considered horizontal hence, \( \vec{n}_o = -\vec{r}_o = -\hat{r}_o/|\hat{r}_o| \) and \( \vec{n}_A = \hat{r}_A/|\hat{r}_A| \).

Finally, the quantity \( \xi B \), which is to be deduced from the GOES IR image is rendered by the equivalent expression in terms of flux density \( F \) (in Watt \( m^{-2} \)),

\[
\frac{F}{\pi} = \xi B
\]

According to the empirical study of Smith et al cited above,

\[
F = 0.543 \ T_{\text{GOES}}^4 + 44.538
\]

\( T_{\text{GOES}} \) is the GOES window IR temperature and the Stephan Boltzman constant \( = 5.66 \times 10^{-8} \text{W/M}^2 \text{deg}^{-4} \).
Figure 3. Terms relating to an emitting surface $A$.

Figure 4. Radiant energy flux incident on an area $dA_0$. 
If \( CT \) is a GOES digital count value \( T_{\text{GOES}} \) is given by (8),

\[
T = \begin{cases} 
330 - CT/2, & \text{CT} < 176 \\
418 - CT, & \text{CT} > 176 
\end{cases}
\]  

To obtain the total power incident on \( dA_o \), one must sum over all the \( dA \) which can be seen. The \( dA \) are taken to be IR pixels in the GOES image. In order to be seen the pixel locations must satisfy the two conditions below.

**Cond. 1:** \( \hat{n}_o \cdot \hat{r}_{OA} > 0 \)  

If this condition is satisfied the look direction is downward, as shown in Figure 5.

**Cond. 2:** \( \hat{n}_A \cdot \hat{r}_{AO} > 0 \)  

If this condition is not satisfied, it means that the radiation intercepts the earth as shown in Figure 6.

---

**Figure 5.** Illustration for condition 1.

**Figure 6.** Illustration for condition 2.
Program Listing

1 $JOB X U1000
2 $FORTRAN
3 SUBROUTINE MAIN
4 C
5 C +-------------------------------------------------------------+
6 C ! --->NAME
7 C ! GTFLUX
8 C ! --->SYNOPSIS
9 C ! COMPUTE TEMPERATURE AT OBSERVATION POINT.
10 C ! --->DESCRIPTION
11 C ! FLUX IS INTEGRATED OVER A GIVEN AREA, "AS VIEWED" BY AN
12 C ! OBSERVATION POINT. RADIATIVE FLUX PER UNIT AREA IS WEIGHED.
13 C ! THERE ARE SIX INPUT PARAMETERS:
14 C ! ISTART = MIN(1) = SERIAL NUMBER OF FIRST AREA.
15 C ! NUMOFA = MIN(2) = TOTAL NUMBER OF AREAS.
16 C ! LAT = MIN(3) = LATITUDE OF OBSERVATION POINT.
17 C ! LON = MIN(4) = LONGITUDE OF OBSERVATION POINT.
18 C ! IALT = MIN(5) = ALTITUDE OF OBSERVATION POINT.
19 C ! NEW = MIN(6) = ACTUAL NUMBER OF PIXELS/LINE AS OPPOSED TO
20 C ! THOSE DISPLAYED BY THE LA COMMAND. THIS
21 C ! MAY BE PROCURED WITH THE T KEY AND FROM
22 C ! THE RASTOR SCREEN.
23 C +-------------------------------------------------------------+
24 C
25 DIMENSION MIN(10),INBUF(840),IOUTBF(2520)
26 DIMENSION NOUT(24),IGRAY(24)
27 LOGICAL TRACE
28 LOGICAL CRTRCE
29 COMMON /NAVNUM, FTIME, BETAIN, BETDOT, ATFRAC, IYPE, INAV
30 COMMON /COMP, XO, YO, ZO, XA, YA, ZA
31 COMMON /MAINIT, NL, NE, ILIN, ILE, IDAY, JTIME, LIN, IEL
32 COMMON /IRESS, LINRES, IELRES
33 DATA TRACE / .TRUE. /.
34 DATA CRTRCE / .TRUE. /
35 DATA A / 6372.10E03/
36 DATA XKME2ME / 1000.0 /
37 DATA LNOFST, EOFST / 1, 1 /
38 DATA MIN / 'GTFLUX', 8*0 /
39 CALL IQ(MIN)
40 CALL OPN(6)
41 ISTART = MIN(1)
42 NUMOFA = MIN(2)
43 LAT = MIN(3)
44 LON = MIN(4)
45 IALT = MIN(5)
46 NEW = MIN(6)
47 C
48 IF (TRACE) PRINT 37
49 37 FORMAT(1X, ' -- COMMAND PARAMETERS: ')
50 IF (TRACE) PRINT 38, ISTART, NUMOFA, LAT, LON, IALT, NEW
51 38 FORMAT(1X, ' ISTART=', I2, ', NUMOFA=', I1, ', LAT=', I7,
52 * LON=', I8, ', IALT=', I4, ', NEW=', I3)
DIST = FLOAT(IALTI) * XKM2ME
IEND = ISTART + NUMOFA - 1

DO 500 IA = ISTART,IEND

CALL CARTES(LAT,LON,DIST+A,XO,YO,Z0)

CALL INIT(IA)
INDEX = 0
SUMFLX = 0.0
LINE = 1
IYL = 1
IF (CRTRCE) CALL TQMS( 'NL=$' , NL)
IF (CRTRCE) CALL TQMS( 'NEW=$' , NEW)
NSR = NSECL(NE)

POSITION AT BEGINING OF LINE AND READ "NEW" NUMBER OF BRIGHTNESS VALUES INTO INBUF, THEN UNPACK IT AND PUT IT IN OUTBUF.

ISSEC = (LINE-1) * NSR
CALL READA(IA,ISEC,NEW/3,INBUF)
CALL CRACK(NEW,INBUF,OUTBF)

FOR EACH PIXEL IN THE LINE, COMPUTE EARTH AND VECTOR COORDINATES, NEXT COMPUTE THE AREA AND FLUX, AND FINALLY, INTEGRATE

INDEX = INDEX + 1
FSLIN = FLOAT(LINE*LINRES+ILIN-1)
FSELE = FLOAT(IYL*IYLRES+IYLE-1)
CALL SATEAR(PTIME,FSLIN,FSELE,FLAT,FLON,ITYPE,INAV, ...
IF ((FLAT.EQ.100.0).AND.(FLON.EQ.200.0)) GOTO 300
ICOUNT = IOUTBF(IEL)
LAT = ILALO(FLAT)
LON = ILALO(FLON)
call cartes(LAT,LON,XA,YA,ZA)
AREA=STAREA(FLAT,FLON)*(FLOAT(LINRES))*(FLOAT(IELRES))
IF (AREA .EQ. 0.0) GOTO 300
F = WEIGH(ICOUNT)
SUMFLX = SUMFLX + FLUX(AREA,F)
I = IEL + IEOSTF
IF (IEL.LE.NEW) GOTO 357
IEL = 1
IF((LINE/25)*25.EQ.LINE) CALL TQMS('PROCESSED LINE$',LINE)
LINE = LINE + LNOSTF
IF (LINE.LE.NL) GOTO 356
CONTINUE
IF (TRACE) PRINT 198
FORMAT(1X,'---RESULT(S):')
IF (TRACE) PRINT 199,SUMFLX
FORMAT('NET FLUX = ',E14.7)
ENCOD(72,99,NOUT) SUMFLX
FORMAT('NET FLUX = ',E14.7)
CALL TQ(NOUT)
CALL TQMS('INX=',INDEX)
RETURN
END

FUNCTION WEIGH(ICOUNT)

! THIS FUNCTION COMPUTES THE WEIGHTED RADIATIVE FLUX PER UNIT AREA WHEN GIVEN THE BRIGHTNESS VALUE FOR THE PIXEL.

TSUBG = 330.0 - (FLOAT(ICOUNT)/2.0)
IF (ICOUNT.GT.176) TSUBG = FLOAT(418 - ICOUNT)
WEIGH = 0.543*5.66E-08*TSUBG*TSUBG*TSUBG*TSUBG+44.54
RETURN
END

SUBROUTINE INIT(IA)
C ! THIS ROUTINE SETS AREA AND NAVIGATION SPECIFICATIONS. IT CALLS
C ! COAREA, HOWBIG, GETNAV AND GETGAM.
C
DIMENSION ISCRA(1200)
LOGICAL TRACE
COMMON /NAVNUM/ PTIME,BETAIN,BETDOT,ATFRAC,ITYPE,INAV
COMMON /MAINT/ NL,NE,ILIN,IELE,IDAY, JTIME,LINE,IEL
COMMON /IRESOL/ LINRES,IELRES
DATA TRACE / .TRUE. /
CALL COAREA(IAMK, IREEL, IDAY, JTIME, ILIN, IELE, LINRES, IELRES, IGE)
CALL HOWBIG(IANL,NE)

IF (TRACE) PRINT 92
FORMAT(1X, ' -->AREA SPECIFICATIONS: ')
IF (TRACE) PRINT 93, IA, IDAY, JTIME, ILIN, IELE, LINRES, IELRES, NL
FORMAT(1X, 'IA=', I1, ' IDAY=', I7, ' JTIME=', I6, ' ILIN=', I4,

PTIME = FTIME(JTIME)
INAV = 1
ITYPE = 1
ATFRAC = 0.0
CALL GETNAV(IDAY, IEXIST)
CALL GETGAM(IDAY, JTIME, BETAIN, BETDOT)
RETURN

SUBROUTINE CARTES(LAT, LON, DIST, X, Y, Z)

DATA DTR / 0.0174532 /
XLAT = FLALO(LAT) * DTR
XLOL = FLALO(LON) * DTR
X = DIST * COS(XLAT) * COS(XLOL)
Y = DIST * COS(XLAT) * SIN(XLOL)
Z = DIST * SIN(XLAT)
RETURN
END

FUNCTION DOT(XL, YL, ZL, XM, YM, ZM)

C ! THIS FUNCTION TAKES THE SCALAR DOT PRODUCT OF THE TWO VECTORS.
C
C
197 C       DOT = (XL*XU) + (YL*YU) + (ZL*ZU)
198 RETURN
199 END

201 C       SUBROUTINE UNITVC(XM,YM,ZM,XU,YU,ZU)
202 C       +---------------------------------------------------------------------+
203 C       ! THIS SUBROUTINE COMPUTES THE UNIT VECTOR FOR THE GIVEN VECTOR  
204 C       +---------------------------------------------------------------------+
205 C       COMMON /MAINIT/ NL,NE,ILIN,IELE,IDAY,JTIME,LINE,IEL
206 C       DIVISR = SQRT((XM*XM) + (YM*YM) + (ZM*ZM))
207 C       IF (DIVISR .EQ. 0.0) CALL TQMES('0 DIVIDE..LINE =$',LINE)
208 C       IF (DIVISR .EQ. 0.0) CALL TQMES('0 DIVIDE..IEL =$',IEL)
209 C       XU = XM/DIVISR
210 C       YU = YM/DIVISR
211 C       ZU = ZM/DIVISR
212 C       RETURN
213 C       END

216 C       FUNCTION FLUX(AREA,F)
217 C       +---------------------------------------------------------------------+
218 C       ! THIS FUNCTION COMPUTES THE FLUX FOR A PIXEL GIVEN ITS AREA AND  
219 C       ! RADIATIVE FLUX PER UNIT AREA.
220 C       +---------------------------------------------------------------------+
221 C       LOGICAL TRACE
222 C       COMMON /COMP/ XQ,YQ,ZQ,XA,YA,ZA
223 C       DATA TRACE /..FALSE../
224 C       DATA INDX2 /0/
225 C       INDX2 = INDX2 + 1
226 C       XQ = XA - XO
227 C       YQ = YA - YO
228 C       ZQ = ZA - ZO
229 C       IF (TRACE.AND.
230 C       *(MOD(INDX2,250),.EQ.0)) PRINT 222,XQ,YQ,ZQ,XA,YA,ZA,XQ,YA,ZA
231 C       FORMAT(9(1X,E14.7))
232 C       CALL UNITVC(-XQ,-YQ,-ZQ,XNO,YNO,ZNO)
233 C       IF (TRACE.AND.
234 C       *(MOD(INDX2,250),.EQ.0)) PRINT 88,XNO,YNO,ZNO
235 C       FORMAT(3(1X,E14.7))
236 C       CALL UNITVC(XA,YA,ZA,XUA,YUA,ZUA)
C

IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 77, XUA, YUA, ZUA
C
77 FORMAT (3(I2, E14.7))
C
CALL UNITYC (XAO, YAO, ZAO, XUA, YUA, ZUA)
C
IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 777, XUAO, YUAO, ZUAO
C
777 FORMAT (3(I2, E14.7))
C
XUAO = -XUAO
C
YUAO = -YUAO
C
ZUAO = -ZUAO
C
IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 66, XUAO, YUAO, ZUAO
C
66 FORMAT (3(I2, E14.7))
C
PROD = DOT (XUA, YUA, ZUA, XUAO, YUAO, ZUAO)
C
IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 55, PROD
C
55 FORMAT (I2, E14.7)
C
PROD = PROD * DOT (XNO, YNO, ZNO, XUAO, YUAO, ZUAO)
C
IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 44, PROD
C
44 FORMAT (I2, E14.7)
C
PROD = ((PROD*AREA*F)/3.1419526)/(DOT (XAO, YAO, ZAO, XAO, YAO, ZAO))
C
IF (TRACE.AND. * (MOD (INDEX2, 250), EQ. 0)) PRINT 33, PROD, AREA, XAO, YAO, ZAO
C
33 FORMAT (5(I2, E14.7))
C
FLUX = PROD
C
IF ((DOT (XNO, YNO, ZNO, XUAO, YUAO, ZUAO)), GT. 0.0)
C
*FLUX = 0.0
C
IF ((DOT (XUA, YUA, ZUA, XUAO, YUAO, ZUAO)), GT. 0.0)
C
*FLUX = 0.0
C
RETURN
C
END

FUNCTION ARETRI (A1, A2, A3, B1, B2, B3)
C
C
FUNCTION ARETRI (A1, A2, A3, B1, B2, B3)
C
! THIS FUNCTION RETURNS THE AREA OF HALF THE PARALLELOGRAM ENCLOSED
! BETWEEN THE TWO GIVEN VECTORS.
C +-----------------------------------------------------------------------
292  C      QI = (A2*B3) - (A3*B2)
293  C      QJ = (A3*B1) - (A1*B3)
294  C      QK = (A1*B2) - (A2*B1)
295  C      ARETRI = (SQRT((QI*QI)+(QJ*QJ)+(QK*QK)))/2.0
296  C      RETURN
297  C      END
298
299
300  FUNCTION GTAREA(XLAT1,XLON1)
301  C +-----------------------------------------------------------------------
302  C ! THIS FUNCTION IS MODIFIED SATSPO, THE AREA OF A PIXEL WHOSE
303  C ! TOP LEFT CORNER IS GIVEN, IS DETERMINED IN SQUARE METERS.
304  C +-----------------------------------------------------------------------
305  C +-----------------------------------------------------------------------
306  C COMMON /NAVNUM/ PTIME,BETAIN,BETDOT,ATFRAC,ITYPE,INAV
307  C DATA RADIUS /6372.10E03/
308  C CALL SATEAR(PTIME,XLIN,XELE,XLAT1,XLON1,2,INAV,BETAIN,BETDOT,0.0)
309  C *0)
310  C   ILIN1=IROUND(XLIN)
311  C   IELE1=IROUND(XELE)
312  C   IF(ILIN1.EQ.0)GO TO 1
313  C   ILIN2=ILIN1+1
314  C   IELE2=IELE1
315  C   ILIN3=ILIN1
316  C   IELE3=IELE1+1
317  C   ILIN4=ILIN2
318  C   IELE4=IELE1
319  C   XLIN1=ILIN1
320  C   XELE1=IELE1
321  C   XLIN2=ILIN2
322  C   XELE2=IELE1
323  C   XLIN3=ILIN3
324  C   XELE3=IELE3
325  C   XLIN4=ILIN4
326  C   XELE4=IELE4
327  C   CALL SATEAR(PTIME,XLIN1,XELE1,XLAT1,XLON1,1,INAV,BETAIN,BETDOT,0.0)
328  C *0)
329  C   CALL SATEAR(PTIME,XLIN2,XELE2,XLAT2,XLON2,1,INAV,BETAIN,BETDOT,0.0)
330  C *0)
331  C   CALL SATEAR(PTIME,XLIN3,XELE3,XLAT3,XLON3,1,INAV,BETAIN,BETDOT,0.0)
332  C *0)
333  C   CALL SATEAR(PTIME,XLIN4,XELE4,XLAT4,XLON4,1,INAV,BETAIN,BETDOT,0.0)
334  C *0)
335  C   IF(ABS(XLAT2).GT.90.0.OR.ABS(XLAT3).GT.90.0
336  C * .OR.ABS(XLAT4).GT.90.0) GOTO 1
337  C   LAT1 = ILALO(XLAT1)
338  C   LON1 = ILALO(XLON1)
339  C   LAT2 = ILALO(XLAT2)
341  LON2 = ILALO(XLON2)
342  LAT3 = ILALO(XLAT3)
343  LON3 = ILALO(XLON3)
344  LAT4 = ILALO(XLAT4)
345  LON4 = ILALO(XLON4)
346  CALL CARTES(LAT1, LON1, RADIUS, RX1, RY1, RZ1)
347  CALL CARTES(LAT2, LON2, RADIUS, RX2, RY2, RZ2)
348  CALL CARTES(LAT3, LON3, RADIUS, RX3, RY3, RZ3)
349  CALL CARTES(LAT4, LON4, RADIUS, RX4, RY4, RZ4)
350  X13 = RX3 - RX1
351  Y13 = RY3 - RY1
352  Z13 = RZ3 - RZ1
353  X12 = RX2 - RX1
354  Y12 = RY2 - RY1
355  Z12 = RZ2 - RZ1
356  X42 = RX2 - RX4
357  Y42 = RY2 - RY4
358  Z42 = RZ2 - RZ4
359  X43 = RX3 - RX4
360  Y43 = RY3 - RY4
361  Z43 = RZ3 - RZ4
362  GTAREA = ARETRI(X12, Y12, Z12, X13, Y13, Z13)
363  *+ ARETRI(X43, Y43, Z43, X42, Y42, Z42)
364  RETURN
365  1 GTAREA = 0
366  RETURN
367  END$
Program Verification

Consider Ω, the solid angle which can be seen at an altitude h km above the earth as shown in Figure 7. This solid angle is formed by a cone of half angle ψ with apex at the observing point O.

Figure 7. Visible solid angle

As suggested by a reviewer of a preliminary version of this paper, it is easy to show that the flux onto a flat plate of unit area at O is given by,

\[ F = \int_{0}^{2\pi} \int_{0}^{\psi} \epsilon B \sin \theta \cos \theta \, d\theta \, d\phi = \epsilon B \pi \sin^2 \psi \quad (11) \]
A derivation is also given in the solved problems at the end of Chapter 5 in Fleagle and Businger's book cited above. The above expression, (11), assumes a uniform temperature over the earth. We have used this solution to check the computing program, including the vectors and area computations as well as the completeness of the radiance map. This was done by transforming the GOES data to a uniform count value of 120 using an enhancement program which leaves the data otherwise unchanged. This count value corresponds to 270K. Thus for the point at altitude 746km the value 166.58 watts m\(^{-2}\) would be expected. The value obtained was 167.08 Wm\(^{-2}\) (0.3\% high). For the point at 161 km (11) gives 197.75 Wm\(^{-2}\) while we computed 196.67 (about 0.5\% low).

Use of this test enabled us to find a coding error in the earlier version of this work which resulted in a substantial error. In the present version this error has been corrected by the insertion of line 361, what had been erroneously omitted. We are greatful to the reviewer who suggested this test.

It is concluded that the values obtained in this calculation are reasonable.

Results

Results in tabular form were already presented in the introduction. They are also shown in Figure 8. The first part of the flight is over relatively cloud-free ocean. Thus, the equivalent radiating surface has a relatively high temperature resulting in about 210 Wm\(^{-2}\) input. From \(t = 1350\) onward this drops rapidly as the flight encounters high (cold) clouds to about 185 Wm\(^{-2}\).
Figure 8. Flux as a function of time