

Data Set Catalog # 208

Quiet Sun Mag. Field Comp.
Program

PM-10A

1 tape

Table of Contents

1. Introduction
2. Errata/Change Log
3. LINKS TO RELEVANT INFORMATION IN THE ONLINE NSSDC
INFORMATION SYSTEM
4. Catalog Materials
 - a. Associated Documents
 - b. Core Catalog Materials

1. INTRODUCTION:

The documentation for this data set was originally on paper, kept in NSSDC's Data Set Catalogs (DSCs). The paper documentation in the Data Set Catalogs have been made into digital images, and then collected into a single PDF file for each Data Set Catalog. The inventory information in these DSCs is current as of July 1, 2004. This inventory information is now no longer maintained in the DSCs, but is now managed in the inventory part of the NSSDC information system. The information existing in the DSCs is now not needed for locating the data files, but we did not remove that inventory information.

The offline tape datasets have now been migrated from the original magnetic tape to Archival Information Packages (AIP's).

A prior restoration may have been done on data sets, if a requestor of this data set has questions; they should send an inquiry to the request office to see if additional information exists.

2. ERRATA/CHANGE LOG:

NOTE: Changes are made in a text box, and will show up that way when displayed on screen with a PDF reader.

When printing, special settings may be required to make the text box appear on the printed output.

Version	Date	Person	Page	Description of Change
01				
02				

3 LINKS TO RELEVANT INFORMATION IN THE ONLINE NSSDC
INFORMATION SYSTEM:

<http://nssdc.gsfc.nasa.gov/nmc/>

[NOTE: This link will take you to the main page of the NSSDC Master Catalog. There you will be able to perform searches to find additional information]

4. CATALOG MATERIALS:

- a. Associated Documents To find associated documents you will need to know the document ID number and then click here.
<http://nssdcftp.gsfc.nasa.gov/miscellaneous/documents/>

- b. Core Catalog Materials

MODELING PROGRAMS

This data set catalog consists of one 9-track, 1600 BPI, ASCII magnetic tape created on the MODCOMP IV computer. It contains 8 files of various modeling programs as listed below.

<u>D#</u>		<u>C#</u>
D-34471		C-20664

<u>FILE</u>	<u>PROGRAM NAME</u>	<u>NSDF ID</u>
1	QUIET SUN MAG. FIELD COMP. PROGRAM	PS-10A
2	JACKSON'S BOTTOMSIDE PROGRAM	PI-11A
3	JACKSON'S TOPSIDE PROGRAM (SHORT VERSION)	PI-21A
4	JACKSON'S TOPSIDE PROGRAM (LONG VERSION)	PI-21B
5	M-FACT	PI-31A
6	ORB	PX-21A
7	ORBIT	PX-21B
8	CARPET PLOT (CRP) PACKAGE	PX-30B

DOCUMENTATION

<u>FILE</u>	<u>PROGRAM NAME</u>	<u>DOCUMENT</u>
1	QUIET SUN MAG. FIELD COMP PROGRAM	
2	JACKSON'S BOTTOMSIDE PROGRAM	X-625-71-186
3	JACKSON'S TOPSIDE PROGRAM (SHORT VERSION)	"
4	JCAKSON'S TOPSIDE PROGRAM (LONG VERSION)	
5	M-FACT	NSSDC 70-13 B06843-000A
6	ORB	LOCATED ON SHELF IN ROOM 121B
7	ORBIT	"
8	CARPET PLOT PACKAGE	"
9	BOW SHOCK & MAGNETOPOUSE CROSSING	X-692-70-452

QUIET SUN MAG. FIELD COMP.

PROGRAM

PM-10A

This data set catalog consists of one tape containing the Quiet Sun Magnetic Field Computations Program. The tape is 556 BPI, BCD, 7-track, 1 file. It was created on the 360/30 using the 7094 utility card to tape program.

This catalog contains a listing of the program.

D#

D-11293

C#

C-08462

NSDF FREE TEXT ENTRY SHEET (Working Copy)

NSDF ID PM-10A

Originator _____ Date _____

Short Name Quiet Sun Mag. Filed ComputationsAgent Assigned CDW Date 11/8/72

File Coordinator _____ Date _____

☒ Brief Description	☒ New	☐ Addition	☐ Replacement
☐ Full Name	☐ New	☐ Addition	☐ Replacement
☐ Data Set Documentation	☐ New	☐ Addition	☐ Replacement

This data set is a computer program, written in fortran, that can be used to generate or draw coronal field lines based upon the current sheet model of the solar corona. Solar magnetogram data of the longitudinal photospheric magnetic field is used as input. The program yields the vector magnetic field anywhere from 1 to 20 or 30 solar radii from the center of the sun.

```

DEFINE FILE 25(110,1944,U,IV) 00000100
DEFINE FILE 25(110,1944,U,IV) 00000100
REAL LONG 00000110
INTEGER RS,RMAX 00000120
DIMENSION BRS(24,27),BTS(24,27),BPS(24,27) 00000130
DIMENSION R(30),G(10,10,2),H(10,10,2),B(24,27),BS(24,27,3) 00000140
DIMENSION BUFF(4001) 00000150
COMMON NN, LONG, IX, IY, ARG(24), ANG(24), COSARG(24), G, H 00000160
COMMON /GHHG/ GH(100,100) 00000165
EQUIVALENCE (BS(1,1,1),BRS(1,1)), (BS(1,1,2),BTS(1,1)) 00000170
EQUIVALENCE (BS(1,1,3),BPS(1,1)) 00000180
DATA RAD/57.2957795131/, DPH/13.33333/, TST/'TEST'/ 00000220
C 00000223
C INITIALIZE CONSTANTS 00000226
C 00000229
DO 3 K=1,2 00000242
DO 2 I=1,10 00000243
DO 1 J=1,10 00000244
G(I,J,K)=0.0 00000246
1 H(I,J,K)=0.0 00000247
2 CONTINUE 00000248
3 CONTINUE 00000249
WRITE(6,151) 00000290
READ(5,154) N,RMAX,RS,IX,IY, LONG,LAT1,LAT2,LATD, LN1, LN2, LND, TEST
1, SCL
READ(5,155) (R(I),I=1,RMAX)
NN=N+1
MAX=NN*NN
G(1,3,1)=R(1)
G(1,3,2)=R(RS)
IF(IX.LE.0) GO TO 6
C 00000300
C INITIALIZE PLOTS AND DRAW GRID 00000310
C 00000320
CALL PLOTS(BUFF,16000) 00000330
CALL PLOT(15.,-30.,-3) 00000340
CALL FACTOR(SCL) 00000345
CALL GRID(R(1),R(RS),RAD) 00000350
C 00000360
C GENERATE SOURCE FIELD 00000370
C 00000380
6 DO 10 I=1,24 00000399
COSARG(I)=(12.5-11)/12.0 0000400
ARG(I)=ARCOS(COSARG(I)) 0000410
ANG(I)=ARG(I)*RAD 0000420
10 CONTINUE 0000470
IF(TEST.EQ.TST) GO TO 5
READ(5,156) B
GO TO 9
5 DO 8 I=1,24
B(I,1)=5.0*COSARG(I)*SIN(ARG(I))/(2.0*(R(1)**3))
DO 7 J=2,27
7 B(I,J)=B(I,J-1)
8 CONTINUE
C 00000480
C COMPUTE FOURIER-LEGENDRE COEFFICIENTS ON SUN SURFACE 00000490
C 00000500
9 CALL GANDH(G,H,B) 00000520
C 00000540
C COMPUTE FIELD AT BOUNDRY R=R(RS) SOLAR RADII 00000550
C 00000560
DO 22 J=1,27 00000565

```

	PHI=AMOD(J*DPH,360.)	00000570
	DO 20 I=1,24	00000580
	CALL FIELD(G,H,R(RS),ARG(I),PHI,BR,BT,BP)	00000590
	IF(BR.LT.0.0) GO TO 12	00000595
	BRS(I,J)=BR	00000600
	BTS(I,J)=BT	00000610
	BPS(I,J)=BP	00000620
	GO TO 14	00000622
12	BRS(I,J)=-BR	00000624
	BTS(I,J)=-BT	00000626
	BPS(I,J)=-BP	00000628
14	CONTINUE	00000630
20	CONTINUE	00000690
22	CONTINUE	00000695
C		00000700
C	COMPUTE FOURIER-LEGENDRE COEFFICIENTS OUTSIDE BOUNDRY	00000710
C		00000720
	CALL FITGH(BS,G(1,1,2),H(1,1,2),GH,MAX)	00000730
	WRITE(6,150) B,G,H	00000750
C		
C	COMPUTE FIELD AT EACH R,TH,PH REQUESTED	
C		
	KK=1	00000760
24	DO 40 K=1,RMAX	00000790
	GO TO 27	00000800
25	KK=2	00000810
27	CONTINUE	00000820
	WRITE(6,153)	
	DO 35 J=LN1,LN2,LND	
	PHI=(J*DPH)	
	DO 30 I=LAT1,LAT2,LATD	00000830
	CALL FIELD(G(1,1,KK),H(1,1,KK),R(K),ARG(I),PHI,BR,BT,BP)	00000840
	WRITE(6,152) R(K),ANG(I),PHI,BR,BT,BP	00000850
30	CONTINUE	00000860
35	CONTINUE	
	IF((K.EQ.1.AND.K.EQ.RS) GO TO 25	00000890
40	CONTINUE	00000900
	IF(IX.LE.0) RETURN	
C		00000903
C	PLOT FIELD LINES	00000906
C		00000909
	DO 70 J=LN1,LN2,LND	00000910
	DO 60 I=LAT1,LAT2,LATD	00000912
	THE=ARG(I)	00000913
	PHI = (J*DPH)/RAD	
	CALL BL INE(THE,PHI)	00000920
60	CONTINUE	00000930
70	CONTINUE	00000932
	CALL PLOT(10.,-5.,-3)	00000934
	CALL PLOT(-0.,-0.,999)	00000938
	RETURN	00000940
150	FORMAT('1 BLOS(24,27) ',27(2(12F7.2//))//	00000950
	1 '1G(N,M) ',/,20(1X,10E12.4//)	00000960
	2 '1H(N,M) ',/,20(1X,10E12.4//)	00000970
151	FORMAT(1H1)	00000980
152	FORMAT(1X,3F7.2,3E12.4)	00000990
153	FORMAT('1 R LAT LONG BR BT BP')	
154	FORMAT(5I3,F5.1,6I3,A4,F5.1)	
155	FORMAT(15F5.1,/,15F5.1)	
156	FORMAT(12F6.3)	00001005
	END	00001010
	SUBROUTINE BLINE(THETA,PHI)	00000100
	AMAG(A,B,C)=SQRT(A*A+B*B+C*C)	00000110
	LOGICAL POSTIV	00000120

COMMON NN, LONG, IX, IY, ARG(24), D(48), G(10,10,2), H(10,10,2)	00000130
DIMENSION XYZ(3), B3(3), DX(3)	00000140
EQUIVALENCE (RS, G(1,3,2))	
EQUIVALENCE (B2, B3(3)), (BY, B3(2)), (BX, B3(1))	00000150
DATA PI/6.28318530/, BMIN/1.0E-3/, RAD/57.2957795131/	00000160
K=1	00000162
RL=RS-0.15	
RM=RS+0.15	
LOOP=0	00000163
R=1.0	00000164
PHD=PHI*RAD	00000165
CALL FIELD(G,H,R,THETA,PHD,BR1,BT1,BP1)	00000166
IF(ABS(BR1).LT.BMIN) RETURN	00000168
XYZ(1)=SIN(THETA)*COS(PHI)	00000180
XYZ(2)=SIN(THETA)*SIN(PHI)	00000190
XYZ(3)=COS(THETA)	00000200
CALL PLOT(XYZ(IX), XYZ(IY), 3)	00000210
POSTIV=.TRUE.	00000220
IF(BR1.LT.0.0) POSTIV=.FALSE.	00000230
GO TO 5	00000240
4 K=1	00000250
IF(R.GE.RS) K=2	00000255
PHD=PHI*RAD	00000258
CALL FIELD(G(1,1,K), H(1,1,K), R, THETA, PHD, BR1, BT1, BP1)	00000260
5 BC=R/(AMAG(BR1,BT1,BP1)*20.0)	00000270
IF(POSTIV) GO TO 8	00000273
IF(K.EQ.2) GO TO 8	00000276
BC=-BC	00000279
8 DX(1)=BR1*BC	00000280
DX(2)=BT1*BC/R	
DX(3)=BP1*BC*RAD/(R*SIN(THETA))	
L=1	00000303
IF((R+DX(1)).GE.RS) L=2	00000305
PHD=PHI*RAD	00000308
CALL FIELD(G(1,1,L), H(1,1,L), R+DX(1), THETA+DX(2), PHD+DX(3),	00000310
1 BR2,BT2,BP2)	00000320
IF(POSTIV.OR.K.EQ.L) GO TO 9	00000322
BR2=-BR2	00000324
BT2=-BT2	00000326
BP2=-BP2	00000328
9 BR=(BR1+BR2)/2.0	00000330
BT=(BT1+BT2)/2.0	00000340
BP=(BP1+BP2)/2.0	00000350
BC=R/(AMAG(BR,BT,BP)*20.0)	00000360
IF(POSTIV.OR.K.EQ.2) GO TO 11	00000362
BC=-BC	00000363
11 BRC=BR*BC	00000364
DX(1)=BRC+R	00000370
IF(DX(1).LE.5.0) GO TO 12	00000380
SCALE=(5.0-R)/ABS(BRC)	00000390
GO TO 13	00000393
12 IF(DX(1).GE.1.0) GO TO 14	00000396
SCALE=(R-1.0)/ABS(BRC)	00000399
13 BR=BR*SCALE	00000400
BT=BT*SCALE	00000410
BP=BP*SCALE	00000420
14 CALL ROTATE(BR,BT,BP,R,THETA,PHI,BX,BY,BZ)	00000430
IF(R.GT.RL.AND.R.LT.RM) BC=BC/4.0	00000435
DX(1)=BX*BC	00000440
DX(2)=BY*BC	00000450
DX(3)=BZ*BC	00000460
IF((BR.LT.0.0).OR.((.NOT.POSTIV).AND.K.EQ.2)) GO TO 18	00000470
GO TO 20	00000475
18 CALL PLOT(XYZ(IX)+DX(IX)/2, XYZ(IY)+DX(IY)/2, 3)	00000520

```

20 XYZ(1)=XYZ(1)+DX(1) 00000530
   XYZ(2)=XYZ(2)+DX(2) 00000540
   XYZ(3)=XYZ(3)+DX(3) 00000550
   R=AMAG(XYZ(1),XYZ(2),XYZ(3)) 00000552
   R1=SQRT(XYZ(IX)*XYZ(IX)+XYZ(IY)*XYZ(IY)) 00000554
   ICODE=2 00000555
   IZ=6-IX-IY
   Q=1.
   IF((IY-IX).EQ.-1.OR.(IY-IX).EQ.2) Q=-1.
   IF(R1.LE.1.0.AND.(Q*XYZ(IZ)).LT.0.) ICODE =3
   CALL PLOT(XYZ(IX),XYZ(IY),ICODE) 00000560
   LOOP=LOOP+1 00000576
   IF(_LOOP.GE.60) GO TO 40 00000578
30 IF(R.GE.5.0.OR.R.LE.1.0) RETURN 00000580
   THETA=ARCOS(XYZ(3)/R) 00000620
   PHI=AMOD(ATAN2(XYZ(2),XYZ(1))+PI,PI) 00000630
   GO TO 4 00000640
40 WRITE(6,170) LOOP 00000642
170 FORMAT(' *** ABNORMAL CONDITION ***', 00000644
1 /, ' *** TERMINATE LINE DUE TO SEGMENT COUNT OF',I3,' ***') 00000646
   RETURN 00000648
   END 00000650
   SUBROUTINE FIELD(G,H,R,THETA,PHI,BR,BT,BP) 00000100
   DIMENSION G(10,10),H(10,10) 00000110
   COMMON NN 00000115
   DATA RAD/57.2957795131/ 00000120
C 00000122
C COMPUTE MAGNETIC FIELD FROM GIVEN POSITION AND 00000124
C FOURIER-LEGENDRE COEFFICIENTS 00000126
C 00000128
   BR=0.0 00000130
   BT=0.0 00000140
   BP=0.0 00000150
   RSR=G(1,3)/R 00000155
   DO 50 N1=1,NN 00000160
   N=N1-1 00000170
   RR=RSR**(N+2) 00000180
   DO 40 M1=1,N1 00000190
   M=M1-1 00000200
   ARG=AMOD(M*PHI,360.0)/RAD 00000210
   SINMP=SIN(ARG) 00000220
   COSMP=COS(ARG) 00000230
   CORE1=RR*(G(N1,M1)*COSMP+H(N1,M1)*SINMP) 00000240
   X=CJS(THETA) 00000250
   PNM=P(N1,M1,X) 00000260
   BR=BR+N1*CORE1*PNM 00000270
   BT=BT-CORE1*DP(N1,M1,X) 00000280
   BP=BP+M*RR*(G(N1,M1)*SINMP-H(N1,M1)*COSMP)*FNM 00000290
40 CONTINUE 00000300
50 CONTINUE 00000310
   BP=BP/SIN(THETA) 00000320
   RETURN 00000330
   END 00000340
   SUBROUTINE GANDH(G,H,D) 00000100
   REAL LONG 00000110
   COMMON NN,LONG,IX,IY,ARG(24),ANG(24),COSARG(24) 00000115
   DIMENSION D(24,27),G(10,10),H(10,10),CSC(24) 00000120
   DATA R/57.2957795/,DPH/13.333333/ 00000130
C 00000135
C COMPUTE FOURIER-LEGENDRE COEFFICIENTS FROM 00000140
C RADIAL COMPONENT, BR, ONLY 00000145
C 00000150
   DEL=0.0 00000160
   RS=G(1,3) 00000162

```

```

K=1
IF(2S.GT.1.0) GO TO 40
DO 10 I=1,24
10 CSC(I)=1.0/SIN(ARG(I))
K=0
SUM=0.0
DO 30 I=1,24
DS=0.0
DO 20 J=1,27
20 DS=DS+D(I,J)
DEL=DEL+DS*CSC(I)
30 SUM=SUM+CSC(I)
DEL=DEL/(27.0*SUM)
40 CONTINUE
DO 80 N1=1,NN
N=N1-1
DO 70 M1=1,N1
M=M1-1
COREG=0.0
COREH=0.0
DO 60 I=1,24
PNM=P(N1,M1,COSARG(I))
IF(K.EQ.0) PNM=PNM*CSC(I)
DO 50 J=1,27
S=(D(I,J)-DEL)*PNM
AR=AMOD(M*(-LONG+J*DPH),360.0)/R
COREG=COREG+S*COS(AR)
COREH=COREH+S*SIN(AR)
50 CONTINUE
60 CONTINUE
COEF=(2*N+1)/((N+1)*648.0)
G(N1,M1)=COREG*COEF
H(N1,M1)=COREH*COEF
70 CONTINUE
80 CONTINUE
RETURN
END
SUBROUTINE FITGH(D,G,H,GH,N)
DIMENSION D(24,27,3),G(10,10),H(10,10),C1(648,3),C2(648,3)
DIMENSION A1(100),A2(100),X(100),C3(648,3)
DIMENSION Y(100)
COMMON NN
DIMENSION GH(N,N)
EQUIVALENCE (C1(1,1),A1(1)), (C1(1,2),A2(1)), (C1(1,3),X(1))
DATA PIVMI/1.0E-6/

C
C
C FIND FOURIER-LEGENDRE COEFFICIENTS OUTSIDE
C BOUNDRY BY 'LEAST MEAN SQUARE FIT' SOLUTION OF
C OVERDETERMINED LINEAR SYSTEM OF 1944 EQUATIONS AND
C NN*NN UNKNOWN(S) (G'S + H'S)
C
C (GH)X = Y
C
C VECTOR X CONTAINS G'S + H'S
C
C NOTE FOR M=0, THE H(N1,1) ELEMENTS MUST BE OMITTED
C OR GH BECOMES SINGULAR.GH USED HERE REFERS TO AB IN WRITE-UP,
C Y REFERS TO GH IN WRITE-UP.
NM=(N+NN)/2
IND=NM
DO 20 N1=1,NN
DO 10 M1=1,N1
C
C EACH CALL COMPUTES ONE COLUMN OF ALPHAS AND ONE

```

```

00000163
00000165
00000166
00000167
00000168
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000245
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000315
00000318
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000263
00000266
00000269
00000270
00000280
00000285
00000290
00000300
00000310
00000320

```

C	COLUMN OF BETAS	00000330
C	CALL AANDB(N1,M1,IND)	00000340
10	CONTINUE	00000350
20	CONTINUE	00000360
C		00000370
C	COMPUTE INNER PRODUCTS OF COLUMNS TO FIND TRANSFORM	00000380
C	MATRIX GH AND VECTOR Y	00000390
C		00000400
C	WORK WITH TWO COLUMNS SIMULTANEOUSLY TO REDUCE THE	00000403
C	NUMBER OF READ OPERATIONS BY (N/2)**2	00000406
C	NOTE . N MUST BE ODD	00000409
	DO 40 N1=1,N,2	00000410
	READ(25*N1) C1	00000420
	GH(V1,N1)=XTY(C1,C1,1944)	00000440
	N2=V1+1	00000450
	READ(25*N2) C2	00000451
	GH(N2,N2)=XTY(C2,C2,1944)	00000452
	GH(V1,N2)=XTY(C1,C2,1944)	00000453
	GH(N2,N1)=GH(N1,N2)	00000454
	IF(N2.GE.N) GO TO 35	00000455
	N3=V1+2	00000456
	DO 30 M1=N3,N	00000457
	READ(25*M1) C3	00000470
	GH(V1,M1)=XTY(C1,C3,1944)	00000490
	GH(N2,M1)=XTY(C2,C3,1944)	00000500
	GH(M1,N2)=GH(N2,M1)	00000506
	GH(M1,N1)=GH(N1,M1)	00000509
30	CONTINUE	00000510
	FIND(25*N3)	00000520
35	Y(N1)=XTY(C1,D,1944)	00000550
	Y(N2)=XTY(C2,D,1944)	00000560
40	CONTINUE	00000565
		00000570
C		00000600
C	SOLVE LINEAR SYSTEM BY LU DECOMPOSITION	00000610
C		00000611
C	IF=-2 GIVES PARTIAL PIVCTING WHICH CAN BE USED	00000612
C	WITHOUT DEGRADING RESULT BECAUSE GH IS POSITIVE-DEFINITE	00000613
C		00000614
	IF=-2	00000620
	CALL LUDCO(GH,A1,A2,N,PIVMI,IF,AN,RHO)	00000630
	CALL LINSY(GH,X,Y,A1,A2,N)	00000640
C		00000710
C	SAVE G'S + H'S IN PROPER LOCATIONS	00000720
C		00000730
	J=NM	00000735
	I=0	00000740
	DO 60 N1=1,NN	00000750
	DO 50 M1=1,N1	00000760
	I=I+1	00000770
	G(N1,M1)=X(I)	00000780
	IF(M1.EQ.1) GO TO 45	00000784
	J=J+1	00000788
	H(N1,M1)=X(J)	00000800
	GO TO 50	00000804
45	H(N1,M1)=0.0	00000808
50	CONTINUE	00000810
60	CONTINUE	00000820
	RETURN	00000830
	END	00000840
	SUBROUTINE AANDB(N1,M1,IND)	00000100
	DATA INDEX/0/	00000105
	COMMON NN, LONG, IX, IY, ARG(24), ANG(24), COSARG(24)	00000110
	COMMON /GHHG/ GH(100,100)	00000120

	DIMENSION HG(648,7),SINP(27),COSP(27)	00000130
	DIMENSION ALPHA(24,27), BETA(24,27), GAMMA(24,27)	00000140
	DIMENSION DELTA(24,27), ETA(24,27), ZETA(24,27)	00000150
	DIMENSION A(1944),B(1944)	00000155
	EQUIVALENCE (GH(1,1),HG(1,1),ALPHA(1,1),A(1))	00000160
	EQUIVALENCE (HG(1,4),BETA(1,1),B(1)), (HG(1,2),GAMMA(1,1))	00000170
	EQUIVALENCE (HG(1,5),DELTA(1,1)), (HG(1,3),ETA(1,1))	00000180
	EQUIVALENCE (HG(1,6),ZETA(1,1)), (HG(1,7),SINP(1))	00000190
	EQUIVALENCE (HG(30,7),COSP(1))	00000200
	M=M1-1	00000210
	DO 10 J=1,27	00000220
	AX=AMOD(M*(J*13.3333),360.0)/57.2957795131	00000230
	SINP(J)=SIN(AX)	00000240
10	COSP(J)=COS(AX)	00000250
	DO 30 I=1,24	00000260
	PNM=P(N1,M1,COSARG(I))	00000270
	DPNM=DP(N1,M1,COSARG(I))	00000280
	SINARG=M/SIN(ARG(I))	00000290
	DO 20 J=1,27	00000300
	ALPHA(I,J)= N1*COSP(J)*PNM	00000310
	GAMMA(I,J)=-COSP(J)*DPNM	00000313
	ETA(I,J) = SINARG*SINP(J)*PNM	00000316
	IF(M.EQ.0) GO TO 20	00000318
	BETA(I,J) = N1*SINP(J)*PNM	00000320
	DELTA(I,J)=-SINP(J)*DPNM	00000340
	ZETA(I,J) =-SINARG*COSP(J)*PNM	00000360
20	CONTINUE	00000370
30	CONTINUE	00000380
	INDEX=INDEX+1	00000390
	WRITE(25*INDEX) A	00000400
	IF(M.EQ.0) RETURN	00000420
	IND=IND+1	00000430
	WRITE(25*IND) B	00000460
	RETURN	00000510
	END	00000520
	SUBROUTINE GRID(RSUN,RB,RAD)	00000100
	CALL PLOT(0.,5.5,-3)	00000110
	CALL PLOT(1.,0.,3)	00000120
	DO 2 I=1,361	00000130
	ANG=I/RAD	00000140
	X=RSUN*COS(ANG)	00000150
	Y=RSUN*SIN(ANG)	00000160
2	CALL PLOT(X,Y,2)	00000170
	CALL PLOT(RB,0.,3)	00000180
	DO 3 I=1,361	00000190
	ANG=I/RAD	00000200
	X=RB*COS(ANG)	00000210
	Y=RB*SIN(ANG)	00000220
	ICODE=3-MOD(MOD(I,3),2)	00000230
	CALL PLOT(X,Y,ICODE)	00000240
3	CONTINUE	00000250
	CALL PLOT(3.,0.,3)	00000252
	CALL PLOT(5.,0.,2)	00000254
	CALL PLOT(0.,3.,3)	00000256
	CALL PLOT(0.,5.,2)	00000258
	RETURN	00000260
	END	00000270
	SUBROUTINE ROTATE(BR,BT,BP,R,THETA,PHI,BX,BY,BZ)	00000100
	SINP=SIN(PHI)	00000110
	COSP=COS(PHI)	00000120
	SINT=SIN(THETA)	00000130
	COST=COS(THETA)	00000140
	BX= BR*SINT*COSP + BT*COST*COSP - BP*SINP	00000203
	BY= BR*SINT*SINP + BT*COST*SINP + BP*COSP	

	BZ = BR*COST - BT*SINT	00000209
	RETURN	00000210
	END	00000220
	FUNCTION DP(N1,M1,X)	00000100
	N=N1-1	00000110
	M=M1-1	00000120
	XX=X**2	00000130
	IF(XX-1.0) 10,20,20	00000140
10	DP=(N*XX*P(N1,M1,X)-(N+M)*P(N,M1,X))/SQRT(1.0-XX)	00000150
	RETURN	00000160
20	DP=0.0	00000170
	RETURN	00000180
	END	00000190
	FUNCTION P(N,M,ANG)	00000100
	REAL*8 X,XX,NORM,FACTOR(19)	00000110
	DATA FACTOR/1.0D0, 1.0D0, 2.0D0, 6.0D0, 24.0D0, 120.0D0,	00000111
	1 720.0D0, 5040.0D0, 4032.0D+1, 3.6288D+5, 3.6288D+6,	00000112
	2 3.9917D+7, 4.79D+8, 6.227D+9, 8.7178D+10, 1.3077D+12,	00000114
	3 2.0923D+13, 3.5569D+14, 6.4024D+15/	00000116
	IF(M.EQ.0) GO TO 150	00000120
	IF(M.GT.N) GO TO 150	00000130
	IF(N.GT.10) GO TO 150	00000140
	NORM=DSQRT((2.0D0* FACTOR(N-M+1))/FACTOR(N+M-1))	00000145
C	X=DCOS(ANGD)	00000150
	X=ANG	00000155
	GO TO(10,20,30,40,50,60,70,80,90,100),N	00000160
C	P(0,0)	00000170
10	P=1.0D0	00000180
	RETURN	00000190
20	IF(M-2) 22,24,150	00000200
C	P(1,0)	00000210
22	P=X	00000220
	RETURN	00000230
C	P(1,1)	00000240
24	P=DSQRT(1.0D0-X**2)*NORM	00000250
	RETURN	00000260
30	IF(M-2) 32,34,36	00000270
C	P(2,0)	00000280
32	P=0.5D0*(3.0D0*(X**2)-1.0D0)	00000290
	RETURN	00000300
C	P(2,1)	00000310
34	P=3.0D0*X*DSQRT(1.0D0-X**2)*NORM	00000320
	RETURN	00000330
C	P(2,2)	00000340
36	P=3.0D0*(1.0D0-X**2)*NORM	00000350
	RETURN	00000360
40	GO TO(42,44,46,48),M	00000370
C	P(3,0)	00000380
42	P=0.5D0*(5.0D0*X*X-3.0D0)*X	00000390
	RETURN	00000400
C	P(3,1)	00000410
44	P=1.5D0*DSQRT(1.0D0-X**2)*(5.0D0*(X**2)-1.0D0)*NORM	00000420
	RETURN	00000430
C	P(3,2)	00000440
46	P=15.0D0*X*(1.0D0-X**2)*NORM	00000450
	RETURN	00000460
C	P(3,3)	00000470
48	P=15.0D0*((1.0D0-X**2)**{1.5})*NORM	00000480
	RETURN	00000490
50	GO TO(51,52,53,54,55),M	00000500
C	P(4,0)	00000510
51	P=(35.0D0*(X**4)-30.0D0*(X**2)+3.0D0)/8.0D0	00000520
	RETURN	00000530
C	P(4,1)	00000540

52	P=2.50D0*DSQRT(1.0D0-X*X)*((7.0D0*X*X-3.0D0)*X*NORM RETURN	00000550
C	P(4,2)	00000560
53	P=7.50D0*((7.0D0*X*X-1.0D0)*(1.0D0-X*X)*NORM RETURN	00000570
C	P(4,3)	00000580
54	P=105.0D0*X*((1.0D0-X*X)**(1.5))*NORM RETURN	00000590
C	P(4,4)	00000600
55	P=105.0D0*((1.0D0-X*X)**2)*NORM RETURN	00000610
60	GO TO(61,62,63,64,65,66),M	00000620
C	P(5,0)	00000630
61	P=((63.0D0*X*X-70.0D0)*X*X+15.0D0)*X)/8.0D0 RETURN	00000640
C	P(5,1)	00000650
62	P=15.0D0*DSQRT(1.0D0-X*X)*NORM A*((21.0D0*X*X-14.0D0)*X*X+1.0D0)/8.0D0 RETURN	00000660
C	P(5,2)	00000670
63	P=52.5D0*X*(1.0D0-X*X)*(3.0D0*X*X-1.0D0)*NORM RETURN	00000680
C	P(5,3)	00000690
64	P=52.5D0*((1.0D0-X*X)**1.5)*(9.0D0*X*X-1.0D0)*NORM RETURN	00000700
C	P(5,4)	00000710
65	P=945.0D0*X*((1.0D0-X*X)**2)*NORM RETURN	00000720
C	P(5,5)	00000730
66	P=945.0D0*((1.0D0-X*X)**2.5)*NORM RETURN	00000740
70	GO TO(71,72,73,74,75,76,77),M	00000750
C	P(6,0)	00000760
71	P=((231.0D0*X*X-315.0D0)*X*X+105.0D0)*X*X-5.0D0)/16.0D0 RETURN	00000770
C	P(6,1)	00000780
72	P=21.0D0*DSQRT(1.0D0-X*X)*NORM A*((33.0D0*X*X-30.0D0)*X*X+5.0D0)*X)/8.0D0 RETURN	00000790
C	P(6,2)	00000800
73	P=105.0D0*(1.0D0-X*X)*((33.0D0*X*X-18.0D0)*X*X+1.0D0)*NORM/8.0D0 RETURN	00000810
C	P(6,3)	00000820
74	P=157.5D0*X*((1.0D0-X*X)**1.5)*(11.0D0*X*X-3.0D0)*NORM RETURN	00000830
C	P(6,4)	00000840
75	P=945.0D0*((1.0D0-X*X)**2)*(11.0D0*X*X-1.0D0)*NORM/2.0D0 RETURN	00000850
C	P(6,5)	00000860
76	P=10395.0D0*X*((1.0D0-X*X)**2.5)*NORM RETURN	00000870
C	P(6,6)	00000880
77	P=10395.0D0*((1.0D0-X*X)**3)*NORM RETURN	00000890
80	GO TO(81,82,83,84,85,86,87,88),M	00000900
C	P(7,0)	00000910
81	P((((429.0D0*X*X-693.0D0)*X*X+315.0D0)*X*X-35.0D0)*X)/16.0D0 RETURN	00000920
C	P(7,1)	00000930
82	P((((429.0D0*X*X-495.0D0)*X*X+135.0D0)*X*X-5.0D0)*NORM A*DSQRT(1.0D0-X*X)*7.0D0/16.0D0 RETURN	00000940
C	P(7,2)	00000950
83	P=(1.0D0-X*X)*((((143.0D0*X*X-110.0D0)*X*X+15.0D0)*X)*63.0D0*NORM/8.0D0	00000960

```

RETURN
C P(7,3)
84 P=((429.D0**X-198.D0)**X+9.0D0)*((1.0D0-X)**1.5)*105.D0/8.0D0
P=P*NORM
RETURN
C P(7,4)
85 P=3465.D0*X*((1.0D0-X)**2)*(13.0D0*X-3.0D0)*NORM/2.0D0
RETURN
C P(7,5)
86 P=10395.D0*((1.0D0-X)**2.5)*(13.0D0*X-1.0D0)*NORM/2.0D0
RETURN
C P(7,6)
87 P=135135.0D0*X*((1.0D0-X)**3)*NORM
RETURN
C P(7,7)
88 P=135135.0D0*((1.0D0-X)**3.5)*NORM
RETURN
90 GJ TJ(91,92,93,94,95,96,97,98,99),M
C P(8,0)
91 P((((6435.D0**X-12012.D0)**X+6930.D0)**X-1260.D0)**X+35.D0)
A /128.0D0
RETURN
C P(8,1)
92 P((((715.0D0*X-1001.D0)**X+385.0D0)**X-35.0D0)**X*NORM
A *9.0D0*DSQRT(1.0D0-X)/16.0D0
RETURN
C P(8,2)
93 P((((143.0D0*X-143.0D0)**X+33.0D0)**X-1.0D0)*NORM
A *315.D0*(1.0D0-X)/16.0D0
RETURN
C P(8,3)
94 P((((429.D0*X-286.D0)**X+33.D0)**X)*315.D0*((1.0D0-X)**1.5)
A *NORM/8.0D0
RETURN
C P(8,4)
95 P=((65.D0*X-26.D0)**X+1.0D0)*10395.D0*((1.0D0-X)**2)/8.0D0
P=P*NORM
RETURN
C P(8,5)
96 P=135135.D0*X*((1.0D0-X)**2.5)*(5.0D0*X-1.0D0)*NORM/2.0D0
RETURN
C P(8,6)
97 P=135135.D0*((1.0D0-X)**3)*(15.0D0*X-1.0D0)*NORM/2.0D0
RETURN
C P(8,7)
98 P=2027025.D0*X*((1.0D0-X)**3.5)*NORM
RETURN
C P(8,8)
99 P=2027025.D0*((1.0D0-X)**4)*NORM
RETURN
100 GJ TJ(101,102,103,104,105,106,107,108,109,110),M
C P(9,0)
101 P((((48620.D0*X-102960.D0)**X+72072.D0)**X-18480.D0)
A **X+1260.D0)**X/512.0D0
RETURN
C P(9,1)
102 P((((2431.0D0*X-4004.0D0)**X+2002.0D0)**X-308.0D0)
A **X+7.0D0)*45.D0*DSQRT(1.0D0-X)*NORM/128.0D0
RETURN
C P(9,2)
103 P((((221.0D0*X-273.0D0)**X+91.0D0)**X-7.0D0)**X *NORM
A *495.D0*(1.0D0-X)/16.0D0
RETURN
C P(9,3)

```

```

00001190
00001200
00001210
00001215
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001415
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001525
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800

```

104	P=((221.0D0*X*X-195.0D0)*X*X+39.0D0)*X*X-1.0D0)*NORM	00001810
	A *3465.0D0*((1.0D0-X*X)**1.5)/16.0D0	00001820
	RETURN	00001830
C	P(9,4)	00001840
105	P=((17.0D0*X*X-10.0D0)*X*X+1.0D0)*X*135135.0D0*NORM	00001850
	A *((1.0D0-X*X)**2)/8.0D0	00001860
	RETURN	00001870
C	P(9,5)	00001880
106	P=((35.0D0*X*X-30.0D0)*X*X+1.0D0)*((1.0D0-X*X)**2.5)*NORM	00001890
	A *135135.0D0/8.0D0	00001900
	RETURN	00001910
C	P(9,6)	00001920
107	P=(17.0D0*X*X-3.0D0)*X*((1.0D0-X*X)**3)*675675.0D0*NORM/2.0D0	00001930
	RETURN	00001940
C	P(9,7)	00001950
108	P=(17.0D0*X*X-1.0D0)*((1.0D0-X*X)**3.5)*2027025.0D0*NORM/2.0D0	00001960
	RETURN	00001970
C	P(9,8)	00001980
109	P=34459425.0D0*X*((1.0D0-X*X)**4)*NORM	00001990
	RETURN	00002000
C	P(9,9)	00002010
110	P=34459425.0D0*((1.0D0-X*X)**4.5)*NORM	00002020
	RETURN	00002030
150	CONTINUE	00002040
	P=0.0	00002045
	RETURN	00002050
	END	

//STEP2 EXEC FORTRANF,PARM='EBCDIC,LIST,XREF,ID,MAP,OPT=1'

//SOURCE.SYSIN DD *

SUBROUTINE

C LUDCO (A,K,L,N,PIVMI ,IFLAG,ANORM,RHO)

C
C THIS SUBROUTINE EFFECTIVELY COMPUTES THE LU DECOMPOSITION
C OF MATRIX A BY GAUSSIAN ELIMINATION IN AN EXCHANGE ALGORITHM
C WHICH EMPLOYS EITHER PARTIAL OR COMPLETE PIVOTING.
C THIS SUBROUTINE COMPUTES THE FOLLOWING
C 1 THE INFINITY NORM, ANORM, OF MATRIX A. REF. 1 PAGE 98
C 2 RHO, A PARAMETER WHICH CAN BE USED TO ESTIMATE THE ROUND OFF
C ERRORS AND FLOATING POINT ARITHMETIC ERRORS INTRODUCED
C BY GAUSSIAN ELIMINATION REF.1 PAGE 102
C WHEN CALLING THIS SUBROUTINE....
C 1 A 1 TO A N CONTAIN THE FIRST ROW OF MATRIX A.
C A N+1 TO A 2N CONTAIN THE SECOND ROW OF MATRIX A. ETC.
C HENCE ARRAY A HAS AT LEAST DIMENSION N**2.
C 2 ARRAYS K AND L WILL CONTAIN PIVOTING INFORMATION
C REQUIRED BY THE COMPATIBLE SUBROUTINES IDENTIFIED BELOW.
C K I CONTAINS THE NUMBER OF THE ITH PIVOT ROW
C L I CONTAINS THE NUMBER OF THE ITH PIVOT COLUMN.
C THE CALLING PROGRAM MUST DIMENSION K AND L AT LEAST N.
C UPON RETURN TO THE CALLING PROGRAM, THE LU DECOMPOSED MATRIX A
C WILL RESIDE IN ARRAY A AND WILL BE IN THE PROPER FORM TO COMPUTE
C THE SOLUTION OF AX=B BY CALLING LINSY.

REFERENCES

1. FORSYTHE AND MOLER, COMPUTER SOLUTION OF LINEAR ALGEBRAIC
SYSTEMS

2. STIEFEL, INTRODUCTION TO NUMERICAL ANALYSIS
R.F.THOMPSON, APRIL 1968

DIMENSION A 1 ,K 1 ,L 1

LEAP=2

IF(IFLAG+2) 36,35,36

35 LEAP=1

36 CONTINUE

DO 1 I 1,N

K I 0

1 L I 0

RHO 0.0

NP IV 0

GO TO 20

C

C

C

SEARCH FOR MAXIMUM PIVOT

2 NP IV NP IV+1

PIV 0.0

GO TO (30,40),LEAP

C

C

C

PARTIAL PIVOTING

30 LT=NP IV

DO 32 KK=1,N

DO 33 I=1,NP IV

IF(K(I)-KK) 33,32,33

33 CONTINUE

M=LT+(KK-1)*N

IF(PIV-ABS(A(M))) 31,32,32

31 PIV=ABS(A(M))

KT<K

32 CONTINUE

GO TO 42

C

C

C

COMPLETE PIVOTING

40 DO 6 KK 1,N

DO 3 I 1,NP IV

IF(K(I)-KK) 3,6,3

3 CONTINUE

MM KK-1 *N

DO 5 LL 1,N

DO 4 I 1,NP IV

IF(L(I)-LL) 4,5,4

4 CONTINUE

M MM+LL

IF(PIV-ABS(A(M))) 41,5,5

41 PIV ABS A M

KT < K

LT LL

5 CONTINUE

6 CONTINUE

42 K NP IV KT

L NP IV LT

IF(PIV-RHO) 62,62,61

61 RHO=PIV

62 IF(PIV-PIVMI) 63,63,7

63 CONTINUE

C

C

C

EXIT, SYSTEM ILL CONDITIONED

IFLAG=NP IV-1

GO TO (37,38),LEAP

37 WRITE(6,18) IFLAG

18 FORMAT(5X,'**** PARTIAL PIVOTING. ONLY' I5,' EXCHANGES PERFORMED')

RETURN

38 WRITE(6,19) IFLAG

19 FORMAT(5X,'**** THE RANK OF A IS APPROXIMATELY' I5)

RETURN

7 CONTINUE
IF(NPIV-N) 72,71,72
71 RHO=RHO/ANORM
RETURN

EXCHANGE OF REDUCED MATRIX

72 CONTINUE
MK N* K NPIV -1
MP MK+L NPIV
DO 9 LL 1,N
DO 8 I 1,NPIV
IF(LL-L(I)) 8,9,8
8 CONTINUE
M MK+LL
A M -A M /A MP
9 CONTINUE
DO 12 KK 1,N
DO 10 I 1,NPIV
IF((K-K(I)) 10,12,10
10 CONTINUE
M N* KK-1
KPC M+L NPIV
APC A KPC
DO 11 LL 1,N
DO 11 I 1,NPIV
IF(L(I)-LL) 11,11,111
111 CONTINUE
NM M+LL
MKLL MK+LL
A NM A NM +APC*A MKLL
11 CONTINUE
12 CONTINUE
GO TO 2

COMPUTE INFINITY NORM OF A

20 CONTINUE
ANORM 0.0
SUM 0.0
DO 22 NROW 1,N
MK N* NROW-1
DO 21 NCOL 1,N
M MK+NCOL
21 SUM SUM+ABS A M
IF(SUM-ANORM) 212,212,211
211 ANORM=SUM
212 CONTINUE
SUM 0.0
22 CONTINUE
GO TO 2
END
SUBROUTINE LINSY (A,X,B,K,L,N)

THIS SUBROUTINE SOLVES THE LINEAR EQUATION AX=B.
A IS AN N BY N MATRIX IN LU FORM SEE SUBROUTINE LUDCOM.
THE FIRST N ELEMENTS OF A CONTAIN THE FIRST ROW OF THE MATRIX,
THE SECOND N ELEMENTS OF ARRAY A CONTAIN THE SECOND ROW OF
THE MATRIX, ETC.
DURING COMPUTATION, THE CONTENTS OF ARRAY B ARE ESSENTIALLY
REPLACED BY INVERSE OF L *B, WHICH IS REFERRED TO AS AUGMENTAT90N.
USING THIS AUGMENTED FORM OF B, X IS COMPUTED AS THE
SOLUTION TO AX=B BY SIMPLE BACK SUBSTITUTION.

AFTER EXECUTION, X WILL CONTAIN THE DIRECT SOLUTION OF
AX B. AN ITERATIVE IMPROVEMENT TO X CAN BE ATTEMPTED BY
CALLING SUBROUTINE ITERAT.

R.F. THOMPSON, APRIL, 1968

DIMENSION A 1 ,X 1 ,B 1 ,K 1 ,L 1

N1 N-1
DO 13 I 1,N1
MM L I +N* K I -1
LL L I
KK K I
B KK -B KK /A MM
DO 15 J 1,N
DO 14 JJ 1,I
IF(K(JJ)-J) 14,15,14

14 CONTINUE
M LL+N* J-1
B J B J +B KK *A M
15 CONTINUE
16 CONTINUE

BACK SUBSTITUTION

MM L N +N* K N -1
KK < N
LL L N
X LL B KK /A MM

DO 18 I 1,N1

II N-I

KK < II

LL - II

X LL -B KK

JJ I

DO 17 J 1,I

JJ JJ-1

MK N* KK-1

NN N-JJ

LLL L NN

MM MK+LLL

17 X LL X LL +A MM *X LLL

18 CONTINUE

RETURN

END

// EXEC ASSEMBLR

//SOURCE.SYSIN DD *

* FUNCTION XTY(X,Y,N)
* THIS OS/360 ASSEMBLY PROGRAM CAN BE REFERENCED
* AS A FORTRAN FUNCTION SUBPROGRAM.
* THIS PROGRAM ACCUMULATES THE INNER PRODUCT OF
* VECTORS X AND Y, EACH HAVING N COMPONENTS. THE
* ACCUMULATION IS DONE IN DOUBLE PRECISION
* AND THE FINAL VALUE IS CHOPPED TO SINGLE PRECISION.
* (REF J.H. WILKINSON, ROUNDING ERRORS IN ALGEBRAIC PROCESSES,
* PRENTICE HALL, 1963)

NOTE. IN THE CALLING PROGRAM, FOR I=1,...,N,

X(I) AND Y(I) MUST BE REAL VARIABLES

OF LENGTH 4, AND N MUST BE AN INTEGER

VARIABLE OF LENGTH 4.

GEN REG 1 CONTAINS ADDRESS

OF PARAMETER LIST.

ADDRESS OF X(1) AT 0(1)

Y(1) AT 4(1)

N AT 8(1)

00000110

00000120

00000130

00000140

00000150

00000160

00000170

00000180

00000190

00000200

00000210

00000220

00000230

00000240

00000250

00000260

00000270

00000280

00000290

00000300

START 0

EQU 8

BC 15,12(15)

DC X*7

DC CL7*XTY

STM 14,12,12(13)

BALR B,0

	CAL FACTOR(SCL)	00000345
	CALL GRID(R(1),R(RS),RAD)	00000350
C		00000360
C	GENERATE SOURCE FIELD	00000370
C		00000380
6	DO 10 I=1,24	00000399
	COSARG(I)=(12.5-I)/12.0	00000400
	ARG(I)=ARCOS(COSARG(I))	00000410
	ANG(I)=ARG(I)*RAD	00000420
10	CONTINUE	00000470
	IF(TEST.EQ.TST) GO TO 5	
	READ(5,156) B	
	GO TO 9	
5	DO 8 I=1,24	
	B(I,1)=5.0*COSARG(I)*SIN(ARG(I))/(2.0*(R(1)**3))	
	DO 7 J=2,27	
7	B(I,J)=B(I,J-1)	
8	CONTINUE	
C		00000480
C	COMPUTE FOURIER-LEGENDRE COEFFICIENTS ON SUN SURFACE	00000490
C		00000500
9	CALL GANDH(G,H,B)	00000520
C		00000540
C	COMPUTE FIELD AT BOUNDRY R=R(RS) SOLAR RADII	00000550
C		00000560
	DO 22 J=1,27	00000565
	PHI=AMOD(J*DPH,360.)	00000570
	DO 20 I=1,24	00000580
	CALL FIELD(G,H,R(RS),ARG(I),PHI,BR,BT,BP)	00000590
	IF(BR.LT.0.0) GO TO 12	00000595
	BRS(I,J)=BR	00000600
	BTS(I,J)=BT	00000610
	BPS(I,J)=BP	00000620
	GO TO 14	00000622
12	BRS(I,J)=-BR	00000624
	BTS(I,J)=-BT	00000626
	BPS(I,J)=-BP	00000628
14	CONTINUE	00000630
20	CONTINUE	00000690
22	CONTINUE	00000695
C		00000700
C	COMPUTE FOURIER-LEGENDRE COEFFICIENTS OUTSIDE BOUNDRY	00000710
C		00000720
	CALL FITGH(BS,G(1,1,2),H(1,1,2),GH,MAX)	00000730
	WRITE(6,150) B,G,H	00000750
C		
C	COMPUTE FIELD AT EACH R,TH,PH REQUESTED	
C		
	KK=1	00000760
24	DO 40 K=1,RMAX	00000790
	GO TO 27	00000800
25	KK=2	00000810
27	CONTINUE	00000820
	WRITE(6,153)	
	DO 35 J=LN1,LN2,LND	
	PHI=(J*DPH)	
	DO 30 I=LAT1,LAT2,LATD	00000830
	CALL FIELD(G(1,1,KK),H(1,1,KK),R(K),ARG(I),PHI,BR,BT,BP)	00000840
	WRITE(6,152) R(K),ANG(I),PHI,BR,BT,BP	00000850
30	CONTINUE	00000860
35	CONTINUE	
	IF((K.EQ.1.AND.K.EQ.RS) GO TO 25	00000890
40	CONTINUE	00000900
	IF(IX.LE.0) RETURN	

	USING *,B		00000310
	L 7,8(1)		00000320
	MVC N(4),0(7)		00000330
	L 7,N	LOOP LIMIT IN REG 7 (N)	00000340
	S 7,=F*1*		00000344
	SLL 7,2		00000348
	LA 5,0	LOOP COUNT IN REG 5 (I)	00000350
	LA 6,4	INCREMENT IN REG 6 (I)	00000360
	L 2,0(1)	X(I) ADDRESS INTO REG 2	00000370
	L 3,4(1)	Y(I) ADDRESS INTO REG 3	00000380
	SDR 0,0	FL REG 0 WILL ACCUMULATE (SUM=0.)	00000390
*			00000400
LOOP	LE 2,0(5,2)	LOAD X(I) INTO FL REG 2	00000430
	ME 2,0(5,3)	X(I)*Y(I) NOW IN FL REG 2	00000450
	ADR 0,2	(SUM=SUM+X(I)*Y(I))	00000460
	BXLE 5,6,LOOP	(I=I+1 IF(I.GT.N) GO TO OUT)	00000470
*		PREPARE FOR NEXT PASS THROUGH LOOP	00000480
*			00000520
*		EXIT ROUTINE	00000530
*			00000540
OUT	STE 0,RESULT	STORE FUNCTION VALUE IN RESULT	00000550
	L 0,RESULT	STORE RESULT IN GEN REG 0	00000560
	LM 2,12,28(13)	RESTORE REGISTERS	00000570
	MVI 12(13),X*FF*	FLAG RETURN TO CALLING PROGRAM	00000580
	BCR 15,14	RETURN TO CALLING PROGRAM	00000590
*		(RETURN ADDRESS IN GEN REG 14)	00000600
N	DS 1F		00000610
RESULT	DS 1F		00000620
	END		00000630
	DEFINE FILE 25(110,1944,U,IV)		00000100
	REAL LONG		00000110
	INTEGER RS,RMAX		00000120
	DIMENSION BRS(24,27),BTS(24,27),BPS(24,27)		00000130
	DIMENSION R(30),G(10,10,2),H(10,10,2),B(24,27),BS(24,27,3)		00000140
	DIMENSION BUFF(4001)		00000150
	COMMON NN, LONG, IX, IY, ARG(24), ANG(24), COSARG(24), G, H		00000160
	COMMON /GHHG/ GH(100,100)		00000165
	EQUIVALENCE (BS(1,1,1),BRS(1,1)), (BS(1,1,2),BTS(1,1))		00000170
	EQUIVALENCE (BS(1,1,3),BPS(1,1))		00000180
	DATA RAD/57.2957795131/, DPH/13.33333/, TST/'TEST'/		00000220
C			00000223
C	INITIALIZE CONSTANTS		00000226
C			00000229
	DJ 3 K=1,2		00000242
	DJ 2 I=1,10		00000243
	DJ 1 J=1,10		00000244
	G(I,J,K)=0.0		00000246
1	H(I,J,K)=0.0		00000247
2	CONTINUE		00000248
3	CONTINUE		00000249
	WRITE(6,151)		00000290
	READ(5,154) N,RMAX,RS,IX,IY, LONG,LAT1,LAT2,LATD, LN1, LN2, LND, TEST		
1	SCL		
	READ(5,155) (R(I),I=1,RMAX)		
	NN=N+1		
	MAX=NN*NN		
	G(1,3,1)=R(1)		
	G(1,3,2)=R(RS)		
	IF(IX.LE.0) GO TO 6		
C			00000300
C	INITIALIZE PLOTS AND DRAW GRID		00000310
C			00000320
	CALL PLOTS(BUFF,16000)		00000330
	CALL PLOT(15.,-30.,-3)		00000340

C					00000903
C					00000906
C					00000909
					00000910
					00000912
					00000913
					00000920
					00000930
60	CONTINUE				00000932
70	CONTINUE				00000934
	CALL PLOT(10.,-5.,-3)				00000938
	CALL PLOT(-0.,-0.,999)				00000940
	RETURN				00000950
150	FORMAT('1 BLOS(24,27) ',/ ,27(2(12F7.2)/) /				00000960
	1 '1G(N,M) ',/ ,20(1X,10E12.4)/ //				00000970
	2 '1H(N,M) ',/ ,20(1X,10E12.4)/ //				00000980
151	FORMAT(1H1)				00000990
152	FORMAT(1X,3F7.2,3E12.4)				
153	FORMAT('1 R LAT LONG BR BT BP')				
154	FORMAT(S13,F5.1,6I3,A4,F5.1)				
155	FORMAT(15F5.1,/,15F5.1)				
156	FORMAT(12F6.3)				00001005
	END				00001010
	SUBROUTINE BLINE(THETA,PHI)				00000100
	AMAG(A,B,C)=SQRT(A*A+B*B+C*C)				00000110
	LOGICAL POSTIV				00000120
	COMMON NN, LONG, IX, IY, ARG(24), D(48), G(10,10,2), H(10,10,2)				00000130
	DIMENSION XYZ(3), B3(3), DX(3)				00000140
	EQUIVALENCE (RS, G(1,3,2))				
	EQUIVALENCE (BZ, B3(3)), (BY, B3(2)), (BX, B3(1))				00000150
	DATA PI/6.28318530/, BMIN/1.0E-3/, RAD/57.2957795131/				00000160
	K=1				00000162
	RL=RS-0.15				
	RM=RS+0.15				
	LOOP=0				00000163
	R=1.0				00000164
	PHD=PHI*RAD				00000165
	CALL FIELD(G,H,R,THETA,PHD,BR1,BT1,BP1)				00000166
	IF(ABS(BR1).LT.BMIN) RETURN				00000168
	XYZ(1)=SIN(THETA)*COS(PHI)				00000180
	XYZ(2)=SIN(THETA)*SIN(PHI)				00000190
	XYZ(3)=COS(THETA)				00000200
	CALL PLOT(XYZ(IX),XYZ(IY),3)				00000210
	POSTIV=.TRUE.				00000220
	IF(BR1.LT.0.0) POSTIV=.FALSE.				00000230
	GO TO 5				00000240
4	K=1				00000250
	IF(R.GE.RS) K=2				00000255
	PHD=PHI*RAD				00000258
	CALL FIELD(G(1,1,K),H(1,1,K),R,THETA,PHD,BR1,BT1,BP1)				00000260
5	BC=R/(AMAG(BR1,BT1,BP1)*20.0)				00000270
	IF(POSTIV) GO TO 8				00000273
	IF(K.EQ.2) GO TO 6				00000276
	BC=-BC				00000279
8	DX(1)=BR1*BC				00000280
	DX(2)=BT1*BC/R				
	DX(3)=BP1*BC*RAD/(R*SIN(THETA))				
	L=1				00000303
	IF((R+DX(1)).GE.RS) L=2				00000305
	PHD=PHI*RAD				00000308
	CALL FIELD(G(1,1,L),H(1,1,L),R+DX(1),THETA+DX(2),PHD+DX(3),				00000310
	1 BR2,BT2,BP2)				00000320
	IF(POSTIV.OR.K.EQ.L) GO TO 9				00000322
	BR2=-BR2				00000324

```

BT2=-BT2
BP2=-BP2
9 BR=(BR1+BR2)/2.0
BT=(BT1+BT2)/2.0
BP=(BP1+BP2)/2.0
BC=R/(AMAG(BR,BT,BP)*20.0)
IF(POSTIV.OR.K.EQ.2) GO TO 11
BC=-BC
11 BRC=BR*BC
DX(1)=BRC+R
IF(DX(1).LE.5.0) GO TO 12
SCALE=(5.0-R)/ABS(BRC)
GO TO 13
12 IF(DX(1).GE.1.0) GO TO 14
SCALE=(R-1.0)/ABS(BRC)
13 BR=BR*SCALE
BT=BT*SCALE
BP=BP*SCALE
14 CALL ROTATE(BR,BT,BP,R,THETA,PHI,BX,BY,BZ)
IF(R.GT.RL.AND.R.LT.RM) BC=BC/4.0
DX(1)=BX*BC
DX(2)=BY*BC
DX(3)=BZ*BC
IF((BR.LT.0.0).OR((.NOT.POSTIV).AND.K.EQ.2)) GO TO 18
GO TO 20
18 CALL PLOT(XYZ(IX)+DX(IX)/2,XYZ(IY)+DX(IY)/2,3)
20 XYZ(1)=XYZ(1)+DX(1)
XYZ(2)=XYZ(2)+DX(2)
XYZ(3)=XYZ(3)+DX(3)
R=AMAG(XYZ(1),XYZ(2),XYZ(3))
R1=SQRT(XYZ(1)*XYZ(1)+XYZ(2)*XYZ(2)+XYZ(3)*XYZ(3))
ICODE=2
IZ=6-IX-IY
Q=1.
IF((IY-IX).EQ.-1.OR.(IY-IX).EQ.2) Q=-1.
IF(R1.LE.1.0.AND.(Q*XYZ(IZ)).LT.0.) ICODE=3
CALL PLOT(XYZ(IX),XYZ(IY),ICODE)
LOOP=LOOP+1
IF(LOOP.GE.60) GO TO 40
30 IF(R.GE.5.0.OR.R.LE.1.0) RETURN
THETA=ARCOS(XYZ(3)/R)
PHI=AMOD(ATAN2(XYZ(2),XYZ(1))+PI,PI)
GO TO 4
40 WRITE(6,170) LOOP
170 FORMAT(' *** ABNORMAL CONDITION ***',
1 /,' *** TERMINATE LINE DUE TO SEGMENT COUNT OF',I3,' ***')
RETURN
END
SUBROUTINE FIELD(G,H,R,THETA,PHI,BR,BT,BP)
DIMENSION G(10,10),H(10,10)
COMMON NN
DATA RAD/57.2957795131/

C
C COMPUTE MAGNETIC FIELD FROM GIVEN POSITION AND
C FOURIER-LEGENDRE COEFFICIENTS
C

BR=0.0
BT=0.0
BP=0.0
RSR=G(1,3)/R
DO 50 N1=1,NN
N=N1-1
RR=RSR**(N+2)
DO 40 M1=1,N1

```

```

00000326
00000328
00000330
00000340
00000350
00000360
00000362
00000363
00000364
00000370
00000380
00000390
00000393
00000396
00000399
00000400
00000410
00000420
00000430
00000435
00000440
00000450
00000460
00000470
00000475
00000520
00000530
00000540
00000550
00000552
00000554
00000555
00000560
00000576
00000578
00000580
00000620
00000630
00000640
00000642
00000644
00000646
00000648
00000650
00000100
00000110
00000115
00000120
00000122
00000124
00000126
00000128
00000130
00000140
00000150
00000155
00000160
00000170
00000180
00000190

```

```

M=M1-1
ARG=AMOD(M*PHI,360.0)/RAD
SINMP=SIN(ARG)
COSMP=COS(ARG)
CORE1=RR*(G(N1,M1)*COSMP+H(N1,M1)*SINMP)
X=COS(THETA)
PNM=P(N1,M1,X)
BR=BR-N1*CORE1*PNM
BT=BT-CORE1*DP(N1,M1,X)
BP=BP+M*RR*(G(N1,M1)*SINMP-H(N1,M1)*COSMP)*PNM
40 CONTINUE
50 CONTINUE
BP=BP/SIN(THETA)
RETURN
END
SUBROUTINE GANDH(G,H,D)
REAL LONG
COMMON NN, LONG, IX, IY, ARG(24), ANG(24), COSARG(24)
DIMENSION D(24,27), G(10,10), H(10,10), CSC(24)
DATA R/57.2957795/, DPH/13.333333/

      COMPUTE FOURIER-LEGENDRE COEFFICIENTS FROM
      RADIAL COMPONENT, BR, ONLY

DEL=0.0
RS=G(1,3)
K=1
IF(RS.GT.1.0) GO TO 40
DO 10 I=1,24
  CSC(I)=1.0/SIN(ARG(I))
  K=0
  SUM=0.0
  DO 30 J=1,24
    DS=0.0
    DO 20 J=1,27
      DS=DS+D(I,J)
    DEL=DEL+DS*CSC(I)
  SUM=SUM+CSC(I)
  DEL=DEL/(27.0*SUM)
40 CONTINUE
  DO 80 N1=1,NN
    N=N1-1
    DO 70 M1=1,N1
      M=M1-1
      COREG=0.0
      COREH=0.0
      DO 60 I=1,24
        PNM=P(N1,M1,COSARG(I))
        IF(K.EQ.0) PNM=PNM*CSC(I)
        DO 50 J=1,27
          S=(D(I,J)-DEL)*PNM
          AR=AMOD(M*(-LONG+J*DPH),360.0)/R
          COREG=COREG+S*COS(AR)
          COREH=COREH+S*SIN(AR)
50 CONTINUE
60 CONTINUE
      COEF=(2*N+1)/((N+1)*648.0)
      G(N1,M1)=COREG*COEF
      H(N1,M1)=COREH*COEF
70 CONTINUE
80 CONTINUE
RETURN
END
SUBROUTINE FITGH(D,G,H,GH,N)

```

```

00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000100
00000110
00000115
00000120
00000130
00000135
00000140
00000145
00000150
00000160
00000162
00000163
00000165
00000166
00000167
00000168
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000245
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000315
00000318
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000100

```

```

DIMENSION D(24,27,3),G(10,10),H(10,10),C1(648,3),C2(648,3)
DIMENSION A1(100),A2(100),X(100),C3(648,3)
DIMENSION Y(100)
COMMON NN
DIMENSION GH(N,N)
EQUIVALENCE (C1(1,1),A1(1)), (C1(1,2),A2(1)), (C1(1,3),X(1))
DATA PIVMI/1.0E-6/

      FIND FOURIER-LEGENDRE COEFFICIENTS OUTSIDE
      BOUNDARY BY 'LEAST MEAN SQUARE FIT' SOLUTION OF
      OVERDETERMINED LINEAR SYSTEM OF 1944 EQUATIONS AND
      NN*NN      UNKNOWN(S) (G'S + H'S)

      (GH)X = Y

      VECTOR X CONTAINS G'S + H'S

      NOTE FOR M=0, THE H(N1,1) ELEMENTS MUST BE OMITTED
      OR GH BECOMES SINGULAR.GH USED HERE REFERS TO AB IN WRITE-UP,
      Y REFERS TO GH IN WRITE-UP.

      NM=(1+NN)/2
      IND=NM
      DO 20 N1=1,NN
      DO 10 M1=1,N1

      EACH CALL COMPUTES ONE COLUMN OF ALPHAS AND ONE
      COLUMN OF BETAS

      CALL AANDB(N1,M1,IND)
10    CONTINUE
20    CONTINUE

      COMPUTE INNER PRODUCTS OF COLUMNS TO FIND TRANSFORM
      MATRIX GH AND VECTOR Y

      WORK WITH TWO COLUMNS SIMULTANEOUSLY TO REDUCE THE
      NUMBER OF READ OPERATIONS BY (N/2)**2
      NOTE . N MUST BE ODD
      DO 40 N1=1,N,2
      READ(25*N1) C1
      GH(N1,N1)=XTY(C1,C1,1944)
      N2=N1+1
      READ(25*N2) C2
      GH(N2,N2)=XTY(C2,C2,1944)
      GH(N1,N2)=XTY(C1,C2,1944)
      GH(N2,N1)=GH(N1,N2)
      IF(N2.GE.N) GO TO 35
      N3=N1+2
      DO 30 M1=N3,N
      READ(25*M1) C3
      GH(N1,M1)=XTY(C1,C3,1944)
      GH(N2,M1)=XTY(C2,C3,1944)
      GH(M1,N2)=GH(N2,M1)
      GH(M1,N1)=GH(N1,M1)
30    CONTINUE
      FIND(25*N3)
35    Y(N1)=XTY(C1,D,1944)
      Y(N2)=XTY(C2,D,1944)
40    CONTINUE

      SOLVE LINEAR SYSTEM BY LU DECOMPOSITION

      IF=-2 GIVES PARTIAL PIVOTING WHICH CAN BE USED
      WITHOUT DEGRADING RESULT BECAUSE GH IS POSITIVE-DEFINITE

```

```

00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000263
00000266
00000269
00000270
00000280
00000285
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000403
00000406
00000409
00000410
00000420
00000440
00000450
00000451
00000452
00000453
00000454
00000455
00000456
00000457
00000470
00000490
00000500
00000506
00000509
00000510
00000520
00000550
00000560
00000565
00000570
00000600
00000610
00000611
00000612
00000613

```

C
C
C
45
50
60

10

20
30

IF=-2
CALL LUDCO(GH,A1,A2,N,PIVMI,IF,AN,RHO)
CALL LINSY(GH,X,Y,A1,A2,N)

SAVE G'S + H'S IN PROPER LOCATIONS

J=NM
I=0
DO 60 N1=1,NN
DO 50 M1=1,N1
I=I+1
G(N1,M1)=X(I)
IF(M1.EQ.1) GO TO 45
J=J+1
H(N1,M1)=X(J)
GO TO 50
H(N1,M1)=0.0
CONTINUE
CONTINUE
RETURN
END
SUBROUTINE AANDB(N1,M1,IND)
DATA INDEX/0/
COMMON NN, LONG, IX, IY, ARG(24), ANG(24), COSARG(24)
COMMON /GHHG/ GH(100,100)
DIMENSION HG(648,7), SINP(27), COSP(27)
DIMENSION ALPHA(24,27), BETA(24,27), GAMMA(24,27)
DIMENSION DELTA(24,27), ETA(24,27), ZETA(24,27)
DIMENSION A(1944), B(1944)
EQUIVALENCE (GH(1,1), HG(1,1), ALPHA(1,1), A(1))
EQUIVALENCE (HG(1,4), BETA(1,1), B(1)), (HG(1,2), GAMMA(1,1))
EQUIVALENCE (HG(1,5), DELTA(1,1)), (HG(1,3), ETA(1,1))
EQUIVALENCE (HG(1,6), ZETA(1,1)), (HG(1,7), SINP(1))
EQUIVALENCE (HG(30,7), COSP(1))
M=M1-1
DO 10 J=1,27
AX=AMOD(M*(J*13.3333),360.0)/57.2957795131
SINP(J)=SIN(AX)
COSP(J)=COS(AX)
DO 30 I=1,24
PNM=P(N1,M1,COSARG(I))
DPNM=DP(N1,M1,COSARG(I))
SINARG=M/SIN(ARG(I))
DO 20 J=1,27
ALPHA(I,J)= N1*COSP(J)*PNM
GAMMA(I,J)=-COSP(J)*DPNM
ETA(I,J) = SINARG*SINP(J)*PNM
IF(M.EQ.0) GO TO 20
BETA(I,J) = N1*SINP(J)*PNM
DELTA(I,J)=-SINP(J)*DPNM
ZETA(I,J) =-SINARG*COSP(J)*PNM
CONTINUE
CONTINUE
INDEX=INDEX+1
WRITE(25*INDEX) A
IF(M.EQ.0) RETURN
IND=IND+1
WRITE(25*IND) B
RETURN
END
SUBROUTINE GRID(RSUN,RB,RAD)
CALL PLOT(0.,5.5,-3)
CALL PLOT(1.,0.,3)

00000614
00000620
00000630
00000640
00000710
00000720
00000730
00000735
00000740
00000750
00000760
00000770
00000780
00000784
00000788
00000800
00000804
00000808
00000810
00000820
00000830
00000840
00000100
00000105
00000110
00000120
00000130
00000140
00000150
00000155
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000313
00000316
00000318
00000320
00000340
00000360
00000370
00000380
00000390
00000400
00000420
00000430
00000460
00000510
00000520
00000100
00000110
00000120

	DO 2 I=1,361	00000130
	ANG=I/RAD	00000140
	X=RSUN*COS(ANG)	00000150
	Y=RSUN*SIN(ANG)	00000160
2	CALL PLOT(X,Y,2)	00000170
	CALL PLOT(RB,0.,3)	00000180
	DO 3 I=1,361	00000190
	ANG=I/RAD	00000200
	X=RB*COS(ANG)	00000210
	Y=RB*SIN(ANG)	00000220
	ICODE=3-MOD(MOD(I,3),2)	00000230
	CALL PLOT(X,Y,ICODE)	00000240
3	CONTINUE	00000250
	CALL PLOT(3.,0.,3)	00000252
	CALL PLOT(5.,0.,2)	00000254
	CALL PLOT(0.,3.,3)	00000256
	CALL PLOT(0.,5.,2)	00000258
	RETURN	00000260
	END	00000270
	SUBROUTINE ROTATE(BR,BT,BP,R,THETA,PHI,BX,BY,BZ)	00000100
	SINP=SIN(PHI)	00000110
	COSP=COS(PHI)	00000120
	SINT=SIN(THETA)	00000130
	COST=COS(THETA)	00000140
	BX= BR*SINT*COSP + BT*COST*COSP - BP*SINP	00000203
	BY= BR*SINT*SINP + BT*COST*SINP + BP*COSP	
	BZ= BR*COST - BT*SINT	00000209
	RETURN	00000210
	END	00000220
	FUNCTION DP(N1,M1,X)	00000100
	N=N1-1	00000110
	M=M1-1	00000120
	XX=X**2	00000130
	IF(XX-1.0) 10,20,20	00000140
10	DP=(V*XP(N1,M1,X)-(N+M)*P(N,M1,X))/SQRT(1.0-XX)	00000150
	RETURN	00000160
20	DP=0.0	00000170
	RETURN	00000180
	END	00000190
	FUNCTION P(N,M,ANG)	00000100
	REAL*8 X,XX,NORM,FACTOR(19)	00000110
	DATA FACTOR/1.0D0, 1.0D0, 2.0D0, 6.0D0, 24.0D0, 120.0D0,	00000111
	1 720.0D0, 5040.0D0, 4032.0D+1, 3.6288D+5, 3.6288D+6,	00000112
	2 3.9917D+7, 4.79D+8, 6.227D+9, 8.7178D+10, 1.3077D+12,	00000114
	3 2.0923D+13, 3.5569D+14, 6.4024D+15/	00000116
	IF(M.EQ.0) GO TO 150	00000120
	IF(M.GT.N) GO TO 150	00000130
	IF(M.GT.10) GO TO 150	00000140
	NORM=DSQRT((2.0D0* FACTOR(N-M+1))/FACTOR(N+M-1))	00000145
C	X=DCOS(ANG)	00000150
	X=ANG	00000155
	GO TO(10,20,30,40,50,60,70,80,90,100),N	00000160
C	P(0,0)	00000170
10	P=1.0D0	00000180
	RETURN	00000190
20	IF(M-2) 22,24,150	00000200
C	P(1,0)	00000210
22	P=X	00000220
	RETURN	00000230
C	P(1,1)	00000240
24	P=DSQRT(1.0D0-X**2)*NORM	00000250
	RETURN	00000260
30	IF(M-2) 32,34,36	00000270
C	P(2,0)	00000280

32	P=0.5D0*(3.0D0*(X**2)-1.0D0)	00000290
	RETURN	00000300
C	P(2,1)	00000310
34	P=3.0D0*X*DSQRT(1.0D0-X**2)*NORM	00000320
	RETURN	00000330
C	P(2,2)	00000340
36	P=3.0D0*(1.0D0-X**2)*NORM	00000350
	RETURN	00000360
40	GO TO(42,44,46,48),M	00000370
C	P(3,0)	00000380
42	P=0.5D0*(5.0D0*X*X-3.0D0)*X	00000390
	RETURN	00000400
C	P(3,1)	00000410
44	P=1.5D0*DSQRT(1.0D0-X**2)*(5.0D0*(X**2)-1.0D0)*NORM	00000420
	RETURN	00000430
C	P(3,2)	00000440
46	P=15.0D0*X*(1.0D0-X**2)*NORM	00000450
	RETURN	00000460
C	P(3,3)	00000470
48	P=15.0D0*((1.0D0-X**2)**(1.5))*NORM	00000480
	RETURN	00000490
50	GO TO(51,52,53,54,55),M	00000500
C	P(4,0)	00000510
51	P=(35.0D0*(X**4)-30.0D0*(X**2)+3.0D0)/8.0D0	00000520
	RETURN	00000530
C	P(4,1)	00000540
52	P=2.5D0*DSQRT(1.0D0-X*X)*(7.0D0*X*X-3.0D0)*X*NORM	00000550
	RETURN	00000560
C	P(4,2)	00000570
53	P=7.5D0*(7.0D0*X*X-1.0D0)*(1.0D0-X*X)*NORM	00000580
	RETURN	00000590
C	P(4,3)	00000600
54	P=105.0D0*X*((1.0D0-X*X)**(1.5))*NORM	00000610
	RETURN	00000620
C	P(4,4)	00000630
55	P=105.0D0*((1.0D0-X*X)**2)*NORM	00000640
	RETURN	00000650
60	GO TO(61,62,63,64,65,66),M	00000660
C	P(5,0)	00000670
61	P=((63.0D0*X*X-70.0D0)*X*X+15.0D0)*X/8.0D0	00000680
	RETURN	00000690
C	P(5,1)	00000700
62	P=15.0D0*DSQRT(1.0D0-X*X)*NORM	00000710
	A*((21.0D0*X*X-14.0D0)*X*X+1.0D0)/8.0D0	00000720
	RETURN	00000730
C	P(5,2)	00000740
63	P=52.5D0*X*(1.0D0-X*X)*(3.0D0*X*X-1.0D0)*NORM	00000750
	RETURN	00000760
C	P(5,3)	00000770
64	P=52.5D0*((1.0D0-X*X)**1.5)*(9.0D0*X*X-1.0D0)*NORM	00000780
	RETURN	00000790
C	P(5,4)	00000800
65	P=945.0D0*X*((1.0D0-X*X)**2)*NORM	00000810
	RETURN	00000820
C	P(5,5)	00000830
66	P=945.0D0*((1.0D0-X*X)**2.5)*NORM	00000840
	RETURN	00000850
70	GO TO(71,72,73,74,75,76,77),M	00000860
C	P(6,0)	00000870
71	P=((231.0D0*X*X-315.0D0)*X*X+105.0D0)*X*X-5.0D0/16.0D0	00000880
	RETURN	00000890
C	P(6,1)	00000900
72	P=21.0D0*DSQRT(1.0D0-X*X)*NORM	00000910
	A*((33.0D0*X*X-30.0D0)*X*X+5.0D0)*X/8.0D0	00000920

	RETURN	00000930
C	P(5,2)	00000940
73	P=105.00*(1.00-X*X)*((33.00*X*X-18.00)*X*X+1.00)*NORM/8.000	00000950
	RETURN	00000960
C	P(5,3)	00000970
74	P=157.500*X*(1.000-X*X)**1.5*(11.000*X*X-3.000)*NORM	00000980
	RETURN	00000990
C	P(5,4)	00001000
75	P=945.00*((1.000-X*X)**2)*(11.000*X*X-1.000)*NORM/2.000	00001010
	RETURN	00001020
C	P(5,5)	00001030
76	P=10395.000*X*((1.000-X*X)**2.5)*NORM	00001040
	RETURN	00001050
C	P(5,6)	00001060
77	P=10395.000*((1.000-X*X)**3)*NORM	00001070
	RETURN	00001080
80	GO TO(81,82,83,84,85,86,87,88),M	00001090
C	P(7,0)	00001100
81	P=((((429.00*X*X-693.00)*X*X+315.00)*X*X-35.00)*X)/16.000	00001110
	RETURN	00001120
C	P(7,1)	00001130
82	P=((429.000*X*X-495.000)*X*X+135.000)*X*X-5.000)*NORM	00001140
	A=DSQRT(1.000-X*X)*7.000/16.000	00001150
	RETURN	00001160
C	P(7,2)	00001170
83	P=(1.000-X*X)*(((143.00*X*X-110.00)*X*X+15.00)*X)*63.00*NORM/8.000	00001180
	RETURN	00001190
C	P(7,3)	00001200
84	P=((429.00*X*X-198.00)*X*X+9.000)*((1.000-X*X)**1.5)*105.00/8.000	00001210
	P=P*NORM	00001215
	RETURN	00001220
C	P(7,4)	00001230
85	P=3465.00*X*((1.000-X*X)**2)*(13.000*X*X-3.000)*NORM/2.000	00001240
	RETURN	00001250
C	P(7,5)	00001260
86	P=10395.00*((1.00-X*X)**2.5)*(13.000*X*X-1.000)*NORM/2.000	00001270
	RETURN	00001280
C	P(7,6)	00001290
87	P=135135.000*X*((1.000-X*X)**3)*NORM	00001300
	RETURN	00001310
C	P(7,7)	00001320
88	P=135135.000*((1.000-X*X)**3.5)*NORM	00001330
	RETURN	00001340
90	GO TO(91,92,93,94,95,96,97,98,99),M	00001350
C	P(8,0)	00001360
91	P((((6435.00*X*X-12012.00)*X*X+6930.00)*X*X-1260.00)*X*X+35.00)	00001370
	A/128.000	00001380
	RETURN	00001390
C	P(8,1)	00001400
92	P((((715.000*X*X-1001.000)*X*X+385.000)*X*X-35.000)*X*NORM	00001410
	A*9.000*DSQRT(1.000-X*X)/16.000	00001415
	RETURN	00001420
C	P(8,2)	00001430
93	P((((143.000*X*X-143.000)*X*X+33.000)*X*X-1.000)*NORM	00001440
	A*315.00*(1.000-X*X)/16.000	00001450
	RETURN	00001460
C	P(8,3)	00001470
94	P(((429.00*X*X-286.00)*X*X+33.000)*X)*315.00*((1.00-X*X)**1.5)	00001480
	A*NORM/8.000	00001490
	RETURN	00001500
C	P(8,4)	00001510
95	P(((65.00*X*X-26.00)*X*X+1.000)*10395.00*((1.000-X*X)**2)/8.000	00001520
	P=P*NORM	00001525
	RETURN	00001540

C	P(8,5)	00001550
96	P=135135.D0**X*((1.0D0-X**2.5)*(5.0D0*X**X-1.0D0)*NCRM/2.0D0	00001560
	RETURN	00001570
C	P(8,6)	00001580
97	P=135135.D0*((1.0D0-X**3)*(5.0D0*X**X-1.0D0)*NCRM/2.0D0	00001590
	RETURN	00001600
	P(8,7)	00001610
98	P=2027025.D0*X*((1.0D0-X**3.5)*NORM	00001620
	RETURN	00001630
C	P(8,8)	00001640
99	P=2027025.D0*((1.0D0-X**4)*NORM	00001650
	RETURN	00001660
100	GJ TD(101,102,103,104,105,106,107,108,109,110),M	00001670
C	P(9,0)	00001680
101	P=(((((48620.D0*X**X-102960.D0)*X**X+72072.D0)*X**X-18480.D0)	00001690
	A *X**X+1260.D0)*X/512.0D0	00001700
	RETURN	00001710
C	P(9,1)	00001720
102	P=(((((2431.0D0*X**X-4004.0D0)*X**X+2002.0D0)*X**X-308.0D0)	00001730
	A *X**X+7.0D0)*45.D0*D SQRT(1.0D0-X**X)*NORM/128.0D0	00001740
	RETURN	00001750
C	P(9,2)	00001760
103	P=(((((221.0D0*X**X-273.0D0)*X**X+91.0D0)*X**X-7.0D0)*X *NORM	00001770
	A *495.D0*(1.0D0-X**X)/16.0D0	00001780
	RETURN	00001790
C	P(9,3)	00001800
104	P=(((((221.0D0*X**X-195.0D0)*X**X+39.0D0)*X**X-1.0D0)*NORM	00001810
	A *3465.D0*((1.0D0-X**X)**1.5)/16.0D0	00001820
	RETURN	00001830
C	P(9,4)	00001840
105	P=((17.0D0*X**X-10.0D0)*X**X+1.0D0)*X*135135.0D0*NCRM	00001850
	A *((1.0D0-X**X)**2)/8.0D0	00001860
	RETURN	00001870
C	P(9,5)	00001880
106	P=((85.0D0*X**X-30.0D0)*X**X+1.0D0)*((1.0D0-X**X)**2.5)*NORM	00001890
	A *135135.0D0/8.0D0	00001900
	RETURN	00001910
C	P(9,6)	00001920
107	P=(17.0D0*X**X-3.0D0)*X*((1.0D0-X**X)**3)*675675.D0*NCRM/2.0D0	00001930
	RETURN	00001940
C	P(9,7)	00001950
108	P=(17.0D0*X**X-1.0D0)*((1.0D0-X**X)**3.5)*2027025.D0*NCRM/2.0D0	00001960
	RETURN	00001970
C	P(9,8)	00001980
109	P=34459425.D0*X*((1.0D0-X**X)**4)*NCRM	00001990
	RETURN	00002000
C	P(9,9)	00002010
110	P=34459425.D0*((1.0D0-X**X)**4.5)*NORM	00002020
	RETURN	00002030
150	CONTINUE	00002040
	P=0.0	00002045
	RETURN	00002050
	END	

//STEP2 EXEC FORTRANF,PARM='EBCDIC,LIST,XREF,ID,MAP,OPT=1'

//SOURCE.SYS IN DD *

SUBROUTINE

C LUDDO (A,K,L,N,PIVMI ,IFLAG,ANORM,RHO)

THIS SUBROUTINE EFFECTIVELY COMPUTES THE LU DECOMPOSITION
OF MATRIX A BY GAUSSIAN ELIMINATION IN AN EXCHANGE ALGORITHM
WHICH EMPLOYS EITHER PARTIAL OR COMPLETE PIVOTING.

THIS SUBROUTINE COMPUTES THE FOLLOWING

1 THE INFINITY NORM, ANORM, OF MATRIX A. REF. 1 PAGE 98
 2 RHO, A PARAMETER WHICH CAN BE USED TO ESTIMATE THE ROUND OFF
 ERRORS AND FLOATING POINT ARITHMETIC ERRORS INTRODUCED
 BY GAUSSIAN ELIMINATION REF.1 PAGE 102
 WHEN CALLING THIS SUBROUTINE....
 1 A 1 TO A N CONTAIN THE FIRST ROW OF MATRIX A,
 A N+1 TO A 2N CONTAIN THE SECOND ROW OF MATRIX A, ETC.
 HENCE ARRAY A HAS AT LEAST DIMENSION N**2.
 2 ARRAYS K AND L WILL CONTAIN PIVOTING INFORMATION
 REQUIRED BY THE COMPATIBLE SUBROUTINES IDENTIFIED BELOW.
 K I CONTAINS THE NUMBER OF THE ITH PIVOT ROW
 L I CONTAINS THE NUMBER OF THE ITH PIVOT COLUMN.
 THE CALLING PROGRAM MUST DIMENSION K AND L AT LEAST N.
 UPON RETURN TO THE CALLING PROGRAM, THE LU DECOMPOSED MATRIX A
 WILL RESIDE IN ARRAY A AND WILL BE IN THE PROPER FORM TO COMPUTE
 THE SOLUTION OF AX=B BY CALLING LINSY.

REFERENCES

1. FORSYTHE AND MOLER, COMPUTER SOLUTION OF LINEAR ALGEBRAIC
SYSTEMS
2. STIEFEL, INTRODUCTION TO NUMERICAL ANALYSIS
R.F. THOMPSON, APRIL 1968

DIMENSION A 1 ,K 1 ,L 1

LEAP=2
 IF(IFLAG+2) 36,35,36

35 LEAP=1
 36 CONTINUE
 DO 1 I 1,N
 K I 0
 1 L I 0
 RHO 0.0
 NPIV 0
 GO TO 20

SEARCH FOR MAXIMUM PIVOT

2 NPIV NPIV+1
 PIV 0.0
 GO TO (30,40),LEAP

PARTIAL PIVOTING

30 LT=NPIV
 DO 32 KK=1,N
 DO 33 I=1,NPIV
 IF(K(I)-KK) 33,32,33
 33 CONTINUE
 M=LT+(KK-1)*N
 IF(PIV-ABS(A(M))) 31,32,32
 31 PIV=ABS(A(M))
 KT=KK
 32 CONTINUE
 GO TO 42

COMPLETE PIVOTING

40 DO 6 KK 1,N
 DO 3 I 1,NPIV
 IF(K(I)-KK) 3,6,3
 3 CONTINUE
 MM KK-1 *N
 DO 5 LL 1,N

```

DO 4 I 1,NPIV
IF(L(I)-LL) 4,5,4
4 CONTINUE
M MM+LL
IF(PIV-ABS(A(M))) 41,5,5
41 PIV ABS A M
KT KK
LT LL
5 CONTINUE
6 CONTINUE
42 K NPIV KT
L NPIV LT
IF(PIV-RHO) 62,62,61
61 RHO=PIV
62 IF(PIV-PIVMI) 63,63,7
63 CONTINUE

```

C
C
C EXIT, SYSTEM ILL CONDITIONED

```

IFLAG=NPIV-1
GO TO (37,38),LEAP
37 WRITE(6,18) IFLAG
18 FORMAT(5X,'**** PARTIAL PIVOTING. ONLY'IS,' EXCHANGES PERFORMED')
RETURN
38 WRITE(6,19) IFLAG
19 FORMAT(5X,'**** THE RANK OF A IS APPROXIMATELY' IS)
RETURN

```

C
C
7 CONTINUE
IF(NPIV-N) 72,71,72
71 RHO=RHO/ANORM
RETURN

C
C EXCHANGE OF REDUCED MATRIX

```

72 CONTINUE
MK N* K NPIV -1
MP MK+L NPIV
DO 9 LL 1,N
DO 8 I 1,NPIV
IF(LL-L(I)) 8,5,8
8 CONTINUE
M MK+LL
A M -A M /A MP
9 CONTINUE
DO 12 KK 1,N
DO 10 I 1,NPIV
IF(KK-K(I)) 10,12,10
10 CONTINUE
M N* KK-1
KPC M+L NPIV
APC A KPC
DO 11 LL 1,N
DO 111 I 1,NPIV
IF(L(I)-LL) 111,11,111
111 CONTINUE
NM M+LL
MKLL MK+LL
A NM A NM +APC*A MKLL
11 CONTINUE
12 CONTINUE
GO TO 2

```

C
C COMPUTE INFINITY NORM OF A

```

20      CONTINUE
      ANORM 0.0
      SUM 0.0
      DO 22 NROW 1,N
      MK N* NROW-1
      DO 21 NCOL 1,N
      M MK+NCOL
21      SUM SUM+ABS A M
      IF(SUM-ANORM) 212,212,211
211     ANORM=SUM
212     CONTINUE
      SUM 0.0
22      CONTINUE
      GO TO 2
      END
      SUBROUTINE LINSY (A,X,B,K,L,N)

```

```

C      THIS SUBROUTINE SOLVES THE LINEAR EQUATION AX B.
C      A IS AN N BY N MATRIX IN LU FORM  SEE SUBROUTINE LUDCOM .
C      THE FIRST N ELEMENTS OF A CONTAIN THE FIRST ROW OF THE MATRIX,
C      THE SECOND N ELEMENTS OF ARRAY A CONTAIN THE SECOND ROW OF
C      THE MATRIX, ETC.
C      DURING COMPUTATION, THE CONTENTS OF ARRAY B ARE ESSENTIALLY
C      REPLACED BY INVERSE OF L *B, WHICH IS REFERRED TO AS AUGMENTAT90N.
C      USING THIS AUGMENTED FORM OF B, X IS COMPUTED AS THE
C      SOLUTION TO AX B BY SIMPLE BACK SUBSTITUTION.
C      AFTER EXECUTION, X WILL CONTAIN THE DIRECT SOLUTION OF
C      AX B.  AN ITERATIVE IMPROVEMENT TO X CAN BE ATTEMPTED BY
C      CALLING SUBROUTINE ITERAT.

```

R.F.THOMPSON, APRIL, 1968

```

      DIMENSION A 1 ,X 1 ,B 1 ,K 1 ,L 1
      N1 N-1
      DO 16 I 1,N1
      MM - I +N* K I -1
      LL L I
      KK < I
      B KK -B KK /A MM
      DO 15 J 1,N
      DO 14 JJ 1,I
      IF(K(JJ)-J) 14,15,14
14      CONTINUE
      M LL+N* J-1
      B J B J +B KK *A M
15      CONTINUE
16      CONTINUE

```

BACK SUBSTITUTION

```

      MM L N +N* K N -1
      KK K N
      LL L N
      X LL B KK /A MM
      DO 18 I 1,N1
      II N-I
      KK K II
      LL L II
      X LL -B KK
      JJ I
      DO 17 J 1,I
      JJ JJ-1
      MK N* KK-1
      NN N-JJ
      LLL L NN
      MM MK+LLL

```

18 CONTINUE

RETURN

END

```
// EXEC ASSEMBLR
```

//SOURCE.SYS IN DD *

FUNCTION XTY(X,Y,N)

```
*      THIS OS/360 ASSEMBLY PROGRAM CAN BE REFERENCED
*      AS A FORTRAN FUNCTION SUBPROGRAM.
*      THIS PROGRAM ACCUMULATES THE INNER PRODUCT OF
*      VECTORS X AND Y, EACH HAVING N COMPONENTS.  THE
*      ACCUMULATION IS DONE IN DOUBLE PRECISION
*      AND THE FINAL VALUE IS CHOPPED TO SINGLE PRECISION.
*      (REF J.H.WILKINSON, ROUNDING ERRORS IN ALGEBRAIC PROCESSES,
*      PRENTICE HALL, 1963)
```

NOTE. IN THE CALLING PROGRAM, FOR $I=1, \dots, N$,

```
*      X(I) AND Y(I) MUST BE REAL VARIABLES
*      OF LENGTH 4, AND N MUST BE AN INTEGER
*      VARIABLE OF LENGTH 4.
```

XTY	START	0	GEN REG 1 CONTAINS ADDRESS
B	EQU	8	OF PARAMETER LIST.
	BC	15,12(15)	ADDRESS OF X(1) AT 0(1)
	DC	X*7°	Y(1) AT 4(1)
	DC	CL7°XTY °	N AT 8(1)
	STM	14,12,12(13)	
	BALR	B,0	
	USING	*,B	
	L	7,8(1)	
	MVC	N(4),0(7)	
	L	7,N	LOOP LIMIT IN REG 7 (N)
	S	7,=F°1°	
	SLL	7,2	
	LA	5,0	LOOP COUNT IN REG 5 (1)
	LA	6,4	INCREMENT IN REG 6 (1)
	L	2,0(1)	X(1) ADDRESS INTO REG 2
	L	3,4(1)	Y(1) ADDRESS INTO REG 3
	SDR	0,0	FL REG 0 WILL ACCUMULATE (SUM=0.)

[illegible]

```

*
*
EXIT ROUTINE

```

OUT	STE	0,RESULT	STORE FUNCTION VALUE IN RESULT
	L	0,RESULT	STORE RESULT IN GEN REG 0
	LM	2,12,28(13)	RESTORE REGISTERS
	MVI	12(13),X*FF*	FLAG RETURN TO CALLING PROGRAM
	BCR	15,14	RETURN TO CALLING PROGRAM
*			(RETURN ADDRESS IN GEN REG 14)

```

N          DS      1F
RESULT     DS      1F
          END

```

00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000344
00000348
00000350
00000360
00000370
00000380
00000390
00000400
00000439
00000450
00000460
00000470
00000480
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630