

AD-A066 156

NORTHROP RESEARCH AND TECHNOLOGY CENTER PALOS VERDES --ETC F/G 20/5  
THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSE--ETC(U)  
JAN 79 W B LACINA

N00014-78-C-0499

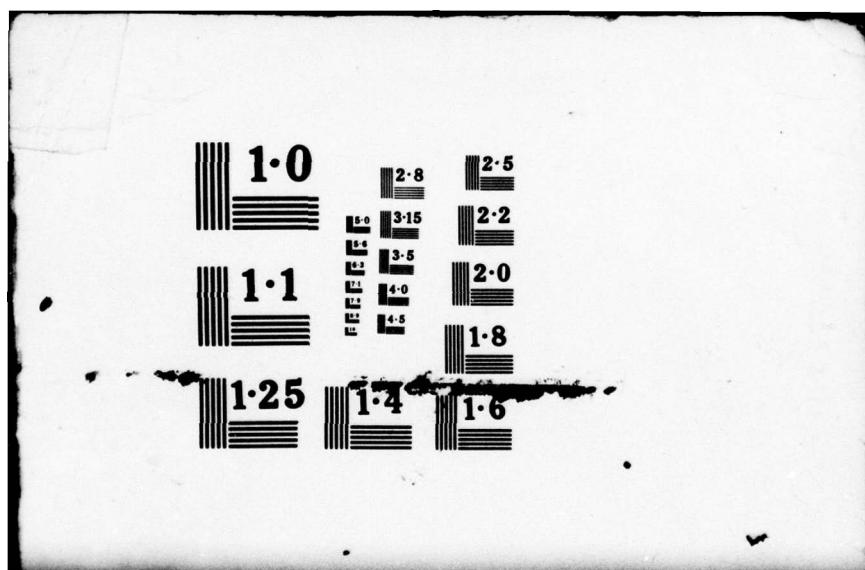
UNCLASSIFIED

NRTC-79-TR-VOL-2

NL

1 OF 2  
ADA  
066156





~~LEVEL~~

(12)  
B.S.

NRTC-79-7R

ADA0 66156

THEORETICAL MODELING OF MOLECULAR AND ELECTRON  
KINETIC PROCESSES

Volume II

FORTRAN Computer Program Listings  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis

January, 1979

DDC FILE COPY

William B. Lacina



Sponsored by

OFFICE OF NAVAL RESEARCH  
Department of the Navy

Contract No. N00014-78-C-0499

This document has been approved  
for public release and sale; its  
distribution is unlimited.

**NORTHROP**

Research and Technology Center

79 03 21 030

(6) THEORETICAL MODELING OF MOLECULAR AND ELECTRON  
KINETIC PROCESSES.

Volume II.

FORTRAN Computer Program Listings,  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis.

(11) January, 1979

(9) Final Technical Rept. 12 Jun 78-15 Jan 79,

(10) William B. Lacina

(14) NRTC-79-7R-VOL-2

NORTHROP RESEARCH AND TECHNOLOGY CENTER  
One Research Park  
Palos Verdes Peninsula, California 90274  
Telephone: (213) 377-4811

(12) Y3YPI

Sponsored by

OFFICE OF NAVAL RESEARCH  
Department of the Navy

Contract No. N00014-78-C-0499

(15)

407 696

alt

ACCESSION for	
NTIS	Ref. Section
DOC	B.R. Section
MANUFACTURER	
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	All Other SPECIAL
A	

## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSES. Vol. II: Fortran Listings, General Laser Kinetics Synthesis and Analysis; Boltzmann Kinetics Analysis		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 12 June 1978 to 15 Jan. 1979
7. AUTHOR(s)  AD - AD006155 William B. Lacina		6. PERFORMING ORG. REPORT NUMBER NRTC-79-7R
8. PERFORMING ORGANIZATION NAME AND ADDRESS NORTHROP Research and Technology Center One Research Park Palos Verdes Peninsula, Ca. 90274		9. CONTRACT OR GRANT NUMBER(S) N00014-78-C-0499
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Dept. of the Navy 800 North Quincy Street Arlington, Virginia 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Attn.: 613C:MAK
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research, Pasadena, California		12. REPORT DATE January, 1979
		13. NUMBER OF PAGES 126
16. DISTRIBUTION STATEMENT (of this Report)  Unlimited.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Refer to Vol. I of this report.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Vol. I of this report describes and documents a comprehensive and reasonably general computer analysis applicable to a broad class of transient, electrically excited laser systems. The theoretical model is based on a coupled analysis of molecular kinetics, electron kinetics, external driving circuit, and radiative extraction. A complete discussion has been given in Vol. I. Vol. II contains the FORTRAN listings for these programs and their subroutines, which were developed for the CDC 6000 and CYBER series computer system.		

THEORETICAL MODELING OF MOLECULAR AND  
ELECTRON KINETIC PROCESSES

Volume I, II

Contract Number: N00014-78-C-0499  
613C:MAK

Principal Investigator: Dr. William B. Lacina  
(213) 377-4811, Ext. 322

Name of Contractor: NOR THROP Corporation  
Northrop Research and  
Technology Center  
One Research Park  
Palos Verdes Peninsula  
California 90274

Scientific Officer: Director, Physics Program  
Physical Sciences Division  
Office of Naval Research  
Department of the Navy  
800 N. Quincy Street  
Arlington, Virginia 22217

Technical Monitor: Dr. Robert Behringer  
Office of Naval Research  
Pasadena, California

Contract Period: 12 June 1978 - 15 Jan. 1979

Reproduction of this document in whole or in part is permitted  
for any purpose of the United States Government.

The views and conclusions contained in this document are those  
of the author and should not be interpreted as necessarily repre-  
senting the official policies, either expressed or implied, of the  
Office of Naval Research or of the United States Government.

THEORETICAL MODELING OF MOLECULAR AND  
ELECTRON KINETIC PROCESSES

Volume II

FORTRAN Computer Program Listings  
Generalized Laser Kinetics Synthesis and Analysis  
Boltzmann Electron Kinetics Analysis

TABLE OF CONTENTS

---

Program LASER .....	1
Subroutines:	
1) SYNTH .....	32
2) ANALYZE .....	54
3) DEKODE .....	56
4) UPDATE .....	57
5) PLASMA .....	61
6) SHAPE .....	64
7) BOLTZ.....	65
8) SIMEQ .....	83
9) GEAR .....	84
10) PLOT .....	96
11) AXIS .....	104
12) INTERP .....	105
13) SIMPSON .....	105
14) EDITOR .....	106
15) COVER .....	107
16) HEADINX .....	108
Program ELECT .....	111

---

PROGRAM LASER (INPUT,OUTPUT,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=OUTPUT,  
 1 TAPE7,TAPE8,TAPE9,TAPE10=INPUT) LASER 2  
 -----
   
 THIS PROGRAM IS A GENERAL LASER KINETICS SYNTHESIS CODE WHICH CAN  
 BE USED TO AUTOMATICALLY GENERATE SUBROUTINES REQUIRED FOR A  
 COUPLED ANALYSIS OF MOLECULAR AND ELECTRON KINETICS, OPTICAL EX-  
 TRACTION, AND EXTERNAL DRIVING CIRCUIT. THE MOLECULAR KINETICS  
 MECHANISMS ARE DEFINED BY AN ARBITRARILY LONG INPUT QUEUE OF SYM-  
 BOLIC REACTIONS, FOLLOWED BY THEIR FORWARD AND/OR REVERSE RATE  
 CONSTANTS, WHICH ARE AUTOMATICALLY TRANSLATED INTO COMPUTER-CODED  
 EQUATIONS. THE SYNTHESIZED SUBROUTINES REQUIRED FOR THE COMPLETE-  
 LY COUPLED ANALYSIS ARE COMPILED AND ADDED TO ALL OTHER REQUIRED  
 SUBROUTINES FOR EXECUTION. A MULTISTEP GEAR INTEGRATION SCHEME  
 FOR STIFF SYSTEMS OF DIFFERENTIAL EQUATIONS IS USED FOR THE FINAL  
 ANALYSIS. THE TREATMENT OF ELECTRON KINETICS CONSISTS OF NUMERI-  
 CAL SOLUTION OF THE BOLTZMANN TRANSPORT EQUATION, WITH SUPERELAS-  
 TIC COLLISIONS AND ELECTRON-ELECTRON SCATTERING INCLUDED. THE  
 RADIATIVE EXTRACTION IS FORMULATED IN TERMS OF THE CAVITY PHOTON  
 DENSITY ( $I = I_0 / C_0 \cdot h \cdot N \cdot U$ ), WITH A SPONTANEOUS EMISSION SOURCE TERM  
 AND AMPLIFICATION FROM STIMULATED EMISSION.  
 -----
   
 THIS CODE WAS DEVELOPED BY --  
 -----
   
 I DR. WILLIAM B. LACINA I LASER 28  
 I NORTHROP RESEARCH AND TECHNOLOGY I LASER 29  
 I ONE RESEARCH PARK I LASER 30  
 I PALOS VERDES PENINSULA, CA 90274 I LASER 31  
 I TEL: (213) 377-4811, EXT. 322 I LASER 32  
 I ----- LASER 33  
 -----
   
 COMPLETE DOCUMENTATION OF THE PRESENT CODE (INCLUDING A DISCUSSION  
 OF THE MATHEMATICAL FORMULATION, DESCRIPTION OF THE SUBROUTINES  
 AND NUMERICAL TECHNIQUES, AND INSTRUCTIONS FOR USAGE) IS AVAILABLE  
 IN PUBLISHED REPORTS (1978). LASER 34  
 -----
   
 DECLARATION STATEMENTS --  
 -----
 DIMENSION KF(200), KR(200), GAS(200), RATEK(200), FLAG(200),  
 1 LEV1(25), LEV2(25), N1(25), N2(25), VSIG(2,25), POWER(25), U(25), LASER 48  
 2 NEL(25), PROCESS(4,25), Q(501,25), QM(501), QMOM(501,2), F(501), LASER 49  
 3 S(501), G(501), A(501,3), B(501), EV(501), NAME(30), MASS(30), LASER 50  
 4 FI(30), E(30), PLOTS(30), PMAX(30), NTIME(30), RPCT(30), POP(32), LASER 51  
 5 NO(32), NDOT(32), DNVDTN(32,8), SCRATCH(32,12), YMAX(32), ERR(32), LASER 52  
 6 PHI(32,65), NOUT(201,2,32), TIME(201), ISUS(201,2), COND(201,2), LASER 53  
 7 IBEAM(201), RAD(201,3), ALPHA(201,4), TB(21), JB(21), COMMENT(5), LASER 54  
 8 Y0(2), DY(2), RATE(2), RNAME(10), IMAGE(8), TITLE(3), KAPTION(4), LASER 55  
 9 IO(10), OUT(10), DUM(5,2), LABEL(5,2), LINE(250), ERROR(10), LASER 56  
 S FATAL(10), MOM(50), ITAU(30) LASER 57  
 -----
 LASER 58

C	REAL NO, N1, N2, NDOT, KF, KR, NTOT, NOUT, NMOL, MASS, MOLWT, KB,	LASER 59
1	LDIDT, KTE, MU, ISUS, NE, IBEAM, INDUCT, KVOLT, LOSS, LENGTH,	LASER 60
2	JBEAM, JSUS, KVCM, IONIZE, JB	LASER 61
C	INTEGER GAS, TYPE, TITLE, RATE, LHS, RHS, TODAY, GENDATE, WARN,	LASER 62
1	WORD	LASER 63
C	LOGICAL CONVRGE, ERROR, REJECT, STOP, FATAL, LIST, OUT, ELECT,	LASER 64
1	EXPAND, TEST, MISSING, OUTSIDE, STIM, END, MODIFY, ILLEGAL,	LASER 65
2	ARC, FLAG, INTRP, PLOTS, REPEAT, ERRORS, FE	LASER 66
C	EQUIVALENCE (B,QM)	LASER 67
C	-----	LASER 68
C	LABELED COMMON BLOCKS --	LASER 69
C	COMMON / DATA / RATEK, KF, KR, VSIG, E	LASER 70
C	COMMON / DISCH / INDUCT, CAPAC, RESIST, MU, AREA, DIST	LASER 71
C	COMMON / GAINS / GNET, GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY	LASER 72
C	COMMON / SOURCE / UPLUS, JBEAM, DVDX, FACTOR, ENERGY, SB, SO	LASER 73
C	COMMON / CONST / NTOT, THOL, FREQ, HNU	LASER 74
C	COMMON / TIMES / TR, TF, TFAIL, TC, TB, JB, INTRP, NPTS, UNITS	LASER 75
C	-----	LASER 76
C	DATA INITIALIZATION --	LASER 77
C	DATA MESH, ITMAX, METHOD, MAXDER, IDEG, IO, NSIZE, LIMIT, NCYCLE,	LASER 78
1	KOUNT / 500, 100, 1, 7, 2, 10*1, 10, 2*200, 20 /	LASER 79
C	DATA EMAX, TMAX, EPS, ETA, RE, ELIMIT, PCT, PER, WARN / 20.0,	LASER 80
1	50.0, 3*0.001, 30.0, 2*5.0, 8HWARNING /	LASER 81
C	DATA KB, EE, PI, EFMT, FFMT / 1.38E-23, 1.602E-19, 3.14159,	LASER 82
1	9H(1PE10.2), 7H(F10.2) /	LASER 83
C	DATA OUT, EXPAND, REPEAT, FATAL, FE, ERROR, ERRORS, PLOTS, FLAG,	LASER 84
1	LIST / 64*.FALSE., 201*.TRUE. /	LASER 85
C	-----	LASER 86
C	NAMELIST PARAMETERS --	LASER 87
C	NAMELIST / CONTROL / EMAX, MESH, EXPAND, ITMAX, THMAX, EPS, IDEG,	LASER 88
1	PCT, KOUNT, NCYCLE, LIMIT, MAXDER, METHOD, ETA, FATAL, IO	LASER 89
C	NAMELIST / PARAM / TPULSE, THOL, TE, PTOT, ATM	LASER 90
C	NAMELIST / RATES / KF, KR	LASER 91
C	-----	LASER 92
C		LASER 93
C		LASER 94
C		LASER 95
C		LASER 96
C		LASER 97
C		LASER 98
C		LASER 99
C		LASER 100
C		LASER 101
C		LASER 102
C		LASER 103
C		LASER 104
C		LASER 105
C		LASER 106
C		LASER 107
C		LASER 108
C		LASER 109
C		LASER 110
C		LASER 111
C		LASER 112
C		LASER 113
C		LASER 114
C		LASER 115

```

NAMELIST / EBEAM / JBEAM, ENERGY, FACTOR, TR, TF, TC, TFALL, UA, LASER 116
 1 UB, JB, TB, UNITS LASER 117
C NAMELIST / CIRCUIT / KVOLT, INDUCT, CAPAC, RESIST, AREA, DIST LASER 118
C NAMELIST / OPTICAL / LOSS, REFLECT, GAMMA, AREA, LENGTH, CAVITY, LASER 119
 1 OMEGA LASER 120
C -----
C ARRAY STORAGE IS DEFINED BY THE FOLLOWING DIMENSION DECLARATORS -- LASER 121
C -----
C MAX = 200 LASER 122
C M1 = MAX+1 LASER 123
C (MAX = MAXIMUM VALUE OF NCYCLE, THE NUMBER OF INTERVALS INTO WHICH LASER 124
C THE TOTAL PULSE DURATION IS SUBDIVIDED FOR BOLTZMANN CALCULATIONS LASER 125
C AND OUTPUT GENERATION.) LASER 126
C MGRID = 500 LASER 127
C (MGRID = MAXIMUM NUMBER OF BINS INTO WHICH THE ELECTRON ENERGY LASER 128
C RANGE MAY BE PARTITIONED IN THE E- KINETICS ANALYSIS.) LASER 129
C KMAX = 200 LASER 130
C (KMAX = MAXIMUM NUMBER OF REACTIONS) LASER 131
C NMAX = 30 LASER 132
C (NMAX = MAXIMUM VALUE OF NTYPE, THE NUMBER OF SPECIES) LASER 133
C NMXP2 = NMAX+2 LASER 134
C (NMXP2 IS MAXIMUM NUMBER OF EQUATIONS FOR SPECIES AND CIRCUIT.) LASER 135
C NKMAX = 25 LASER 136
C (NKMAX = MAXIMUM VALUE OF NK = NUMBER OF SECONDARY E- REACTIONS) LASER 137
C -----
C IF IT IS DESIRED TO CHANGE DIMENSION STORAGE, THE ABOVE DECLARATOR LASER 138
C VALUES MUST BE MODIFIED, AND THE FOLLOWING MISCELLANEOUS ARRAYS LASER 139
C MUST BE CHANGED (AS SHOWN) IN THE DIMENSION STATEMENT --
C -----
C KF(KMAX), KR(KMAX), GAS(KMAX), LEV1(NKMAX), LEV2(NKMAX), N1(NKMAX) LASER 140
C N2(NKMAX), VSIG(2+NKMAX), POWER(NKMAX), U(NKMAX), NEL(NKMAX), LASER 141
C PROCESS(4,NKMAX), Q(MGRID+1,NKMAX), QM(MGRID+1), QMOM(MGRID+1+2), LASER 142
C F(MGRID+1), G(MGRID+1), A(MGRID+1+3), B(MGRID+1), EV(MGRID+1), LASER 143
C FI(NMAX), NAME(NMAX), MASS(NMAX), E(NMAX), PLOTS(NMAX), PMAX(NMAX) LASER 144
C NTIME(NMAX), RPCT(NMAX), NO(NMXP2), POP(NMXP2), NDOT(NMXP2), LASER 145
C DNYDTN(NMXP2,8), NOUT(MAX+1,2,NMAX+2), YMAX(NMXP2), ERR(NMXP2), LASER 146
C SCRATCH(NMXP2,12), PHI(NMAX+2,2*NMAX+5), S(MGRID+1), TIME(MAX+1), LASER 147
C ISUS(MAX+1,2), COND(MAX+1,2), IBEAM(MAX+1), RAD(MAX+1,3), LASER 148
C ALPHA(MAX+1,4), RATEK(KMAX) LASER 149
C -----
C NOTE ALSO THAT DIMENSIONS OF ARRAYS WHICH OCCUR IN LABELLED COMMON LASER 150
C BLOCKS MUST BE ACCORDINGLY MODIFIED IN ANY SUBROUTINE WHERE THEY LASER 151
C OCCUR. IN PARTICULAR, COMMON / DATA / CONTAINS DIMENSIONS IN THE LASER 152
C SYNTHESIZED SUBROUTINES *JACOB* AND *DNOT*. LASER 153
C -----
C IF CERTAIN DIMENSION DECLARATORS DO NOT AGREE WITH THOSE ON AN IN- LASER 154
C PUT DATA FILE (SCRATCH) ACCESSED DURING EXECUTION, AN EXIT OCCURS. LASER 155
C -----
C FILE USAGE --
C INPUT CARD FILE --

```

```

KARDS = 10                                LASER    173
C
C   SUBROUTINE *EDITOR*, WHICH EDITS THE INPUT CARD IMAGES, CREATES AN LASER    174
C   EFFECTIVE INPUT FILE ON TAPE 5, WHICH IS THEN SUBSEQUENTLY USED. LASER    175
C
C   INPUT E- CROSS SECTION FILE --          LASER    176
C   INPUT = 8                               LASER    177
C   REWIND INPUT                           LASER    178
C
C   FILE OF UPDATED E- CROSS SECTIONS --    LASER    179
C   NDATA = 9                               LASER    180
C   REWIND NDATA                           LASER    181
C
C   SCRATCH TAPE USED IN PROGRAM SETUP --   LASER    182
C   NSCRATCH = 4                           LASER    183
C
C   FILE FOR SYNTHESIS OF RATE SUBROUTINES -- LASER    184
C   NTAPE = 7                               LASER    185
C   REWIND NTAPE                           LASER    186
C   MTAPE = 3                               LASER    187
C   REWIND MTAPE                           LASER    188
C   LTAPE = 2                               LASER    189
C   REWIND LTAPE                           LASER    190
C
C   -----
C   DO 7 NK = 1,NKMAX                      LASER    191
C   7 LEV1(NK) = LEV2(NK) = 0               LASER    192
C   DO 1 I = 1,NMAXP2                      LASER    193
C   YMAX(I) = 1.0                           LASER    194
C   DO 1 N = 1,8                           LASER    195
C   1 DNYDTN(I,N) = 0.                      LASER    196
C
C   -----
C   C IS JUST SOME ARBITRARY CONSTANT USED TO TAG RATES WHICH ARE NOT LASER    197
C   EXPLICITLY INITIALIZED LATER BY INPUT -- LASER    198
C   C = -SQRT(1.25963782)                   LASER    199
C   DO 62 K = 1,KMAX                      LASER    200
C   62 KF(K) = KR(K) = C                  LASER    201
C
C   KB = KR/EE                            LASER    202
C   -----
C   CALL DATE (TODAY)                     LASER    203
C   -----
C   REWIND NSCRATCH                      LASER    204
C   READ (NSCRATCH) TITLE, GENDATE       LASER    205
C   IF (EOF(NSCRATCH)) 403              LASER    206
C
C   -----
C   SYNTHESIS OF MOLECULAR KINETICS SUBROUTINES LASER    207
C   -----
C   SYNTHESIZE SUBROUTINE *DNDT* AND *JACOB* TO CALCULATE THE VALUES LASER    208
C   OF DN1/DT AND D/DN1(DN1/DT) AS A FUNCTION OF TIME, AND SUBROUTINE LASER    209
C   *LEVELS* TO DEFINE THE CORRESPONDENCE OF LOWER AND UPPER STATES IN LASER    210
C   THE E- KINETICS ANALYSIS WITH THE MOLECULAR SPECIES LABELS -- LASER    211
C                                         LASER    212
C                                         LASER    213
C                                         LASER    214
C                                         LASER    215
C                                         LASER    216
C                                         LASER    217
C                                         LASER    218
C                                         LASER    219
C                                         LASER    220
C                                         LASER    221
C                                         LASER    222
C                                         LASER    223
C                                         LASER    224
C                                         LASER    225
C                                         LASER    226
C                                         LASER    227
C                                         LASER    228
C                                         LASER    229

```

```

C READ DATA BLOCK NO. 1 -- LASER 230
4 CALL EDITOR (KARDS,LIST) LASER 231
GENDATE = TODAY LASER 232
C CALL SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, INPUT, NSIZE, NMAX, GAS, LASER 233
1 *KMAX, NKMAX, LEV1, LEV2, TODAY) LASER 234
LASER 235
L'ASER 236
C AT THE TERMINATION OF PROGRAM GENERATION, THE SYNTHESIZED SUBROU- LASER 237
TINES (80 BCD CHARACTER RECORDS) ARE STORED ON FILE *MTAPE* = 3, LASER 238
WHICH IS USED AS A SOURCE FILE TO BE COMPILED (AND CATALOGED AS A LASER 239
BCD UPDATE FILE, IF DESIRED). NUMERICAL DATA ASSOCIATED WITH RATE LASER 240
CONSTANTS AND INFORMATION CHARACTERIZING THE REACTION SCHEME EN- LASER 241
COUNTERED (UPON WHICH THE SYNTHESIZED SUBROUTINES ARE BASED) IS LASER 242
STORED ON FILE *NSCRTCH* = 4. IF FUTURE EXECUTION OF THE ANALYSIS LASER 243
IS INTENDED, *NSCRTCH* MUST ALSO BE CATALOGUED AND SAVED. LASER 244
C CALL EXIT LASER 245
C LASER 246
C ----- LASER 247
C ----- LASER 248
C GENERATE AN UPDATED ELECTRON CROSS SECTION FILE -- LASER 249
C READ DATA BLOCK NO. 2 -- LASER 250
3 CALL EDITOR (KARDS,LIST) LASER 251
CALL UPDATE (INPUT, NDATA, NTAPE, .NOT.LIST, TODAY) LASER 252
LASER 253
LASER 254
LASER 255
LASER 256
LASER 257
LASER 258
C READ DATA BLOCK NO. 3 -- LASER 259
CALL EDITOR (KARDS,.NOT.LIST) LASER 260
C ----- LASER 261
C GENERATE COMMENT CARD INFORMATION, IF ANY -- LASER 262
C ----- LASER 263
C ----- LASER 264
LC = 0 LASER 265
55 READ (5+101) IMAGE LASER 266
IF (EOF(5)) 56,59 LASER 267
59 IF (LC.GT.0) GO TO 57 LASER 268
WRITE (6,304) LASER 269
WRITE (6,302) LASER 270
57 LC = LC+1 LASER 271
WRITE (6,303) IMAGE LASER 272
IF (LC.NE.30) GO TO 55 LASER 273
WRITE (6,302) LASER 274
LC = 0 LASER 275
GO TO 55 LASER 276
56 IF (LC.NE.0) WRITE (6,302) LASER 277
C READ DATA BLOCK NO. 4 -- LASER 278
CALL EDITOR (KARDS,LIST) LASER 279
C ----- LASER 280
C ----- LASER 281
C READ GENERAL CONTROL PARAMETERS THAT DEFINE ACCURACIES, SCOPE OF LASER 282
C CALCULATIONS, ITERATION LIMITS, INTERPOLATION ORDER, METHOD OF LASER 283
C INTEGRATION, AND OUTPUT OPTIONS -- LASER 284
----- LASER 285
----- LASER 286

```

```

C
C      -----
C      READ (5,CONTROL)
C      -----
C      IF (METHOD.NE.0.AND.METHOD.NE.2) METHOD = 1
C      IF (MAXDER.GT.8) MAXDER = 8
C      IF (MAXDER.EQ.8.AND.METHOD.NE.0) MAXDER = 7
C      IF (NCYCLE.LE.0) NCYCLE = 1
C      IF (NCYCLE.GT.MAX) NCYCLE = MAX
C      IF (LIMIT.LT.0) LIMIT = 0
C      IF (LIMIT.GT.NCYCLE) LIMIT = NCYCLE
C
C      READ (NSCRTCH) MAXGAS, NTYPE, MAXK, KTYPE, MAXNK, NK, ERRORS
C
C      ERROR(1) = ERRORS
C      ERROR(8) = (MAXGAS.NE.NMAX).OR.(MAXNK.NE.NKMAX).OR.(MAXK.NE.KMAX)
C
C      IF THE FOLLOWING CONDITION OCCURS, DIMENSION STORAGE IS INADEQUATE
C
C      ERROR(9) = (NTYPE.GT.NMAX).OR.(KTYPE.GT.KMAX).OR.(NK.GT.NKMAX)
C
C      TRUNCATE DATA FILE *NSCRTCH* IF NECESSARY --
C      IF (NTYPE.GT.NMAX) NTYPE = NMAX
C      IF (KTYPE.GT.KMAX) KTYPE = KMAX
C      IF (NK.GT.NKMAX) NK = NKMAX
C
C      READ (NSCRTCH) (GAS(N), N = 1,NTYPE)
C      READ (NSCRTCH) (LEV1(N), LEV2(N), N = 1,NK)
C
C      DO 72 I = 1,10
C      72 OUT(I) = IO(I).NE.0
C
C      -----
C      READ GENERAL EXPERIMENTAL PARAMETERS RELATING TO TEMPERATURE,
C      PRESSURE, PULSE LENGTH, AND MODIFICATION OF RATE CONSTANTS --
C
C      PTOT = ATM = TPULSE = TE = 0.
C      TMOL = 300.
C
C      -----
C      READ (5,PARAM)
C      -----
C
C      IF (MESH.GT.MGRID) MESH = MGRID
C      MESHPI = MESH+1
C      IF (TMOL.LE.0.) TMOL = 300.
C      IF (TE.LE.0.) TE = TMOL
C      IF (PTOT.EQ.0.) PTOT = 760.*ATM
C      ATM = PTOT/760.
C      UNIT = 1.0
C      IF (TPULSE.LE.0.) GO TO 42
C      TT = TPULSE
C      43 IF (TT.GT.1) GO TO 42
C      TT = 1000.*TT
C      UNIT = UNIT/1000.

```

LASER 287  
LASER 288  
LASER 289  
LASER 290  
LASER 291  
LASER 292  
LASER 293  
LASER 294  
LASER 295  
LASER 296  
LASER 297  
LASER 298  
LASER 299  
LASER 300  
LASER 301  
LASER 302  
LASER 303  
LASER 304  
LASER 305  
LASER 306  
LASER 307  
LASER 308  
LASER 309  
LASER 310  
LASER 311  
LASER 312  
LASER 313  
LASER 314  
LASER 315  
LASER 316  
LASER 317  
LASER 318  
LASER 319  
LASER 320  
LASER 321  
LASER 322  
LASER 323  
LASER 324  
LASER 325  
LASER 326  
LASER 327  
LASER 328  
LASER 329  
LASER 330  
LASER 331  
LASER 332  
LASER 333  
LASER 334  
LASER 335  
LASER 336  
LASER 337  
LASER 338  
LASER 339  
LASER 340  
LASER 341  
LASER 342  
LASER 343

```

GO TO 43
42 TUUT = TPULSE/NCYCLE
DTIME = TUUT/UNIT
C -----
C READ OPTICAL RESONATOR PARAMETERS (REFLECTIVITY, LOSS, LENGTH,
C OMEGA, ETC.) --
C -----
C LOSS = REFLECT = GAMMA = 0.
LENGTH = CAVITY = AREA = OMEGA = 0.
C -----
C READ (5+OPTICAL)
C -----
IF (LENGTH.LE.0.) LENGTH = 0.01
IF (CAVITY.LE.0.) CAVITY = LENGTH
IF (CAVITY.LT.LENGTH) CAVITY = LENGTH
STIM = REFLECT.GT.0.
IF (REFLECT.LE.0.) REFLECT = 1.E-20
IF (LOSS.LT.0.) LOSS = 0.
R = REFLECT/100.
LOSS = LOSS/100.
IF (GAMMA.GT.0.) LOSS = 0.
IF (OMEGA.LE.0.) OMEGA = AREA/CAVITY**2
IF (GAMMA.EQ.0.) GAMMA = (LOSS - 0.5*ALOG(R))/LENGTH
TCAVITY = CAVITY/(30.*LENGTH*GAMMA)
PASS = 100.*LOSS
OMEGA4P = OMEGA/4./PI
C -----
C READ EXPERIMENTAL ELECTRICAL AND CIRCUIT PARAMETERS --
C -----
C TR = 0.
TF = TC = 10000.
TFALL = 1.0
DU 517 I = 1.21
S17 TB(I) = JB(I) = 0.
UNITS = 1.0E-09
JBEAM = ENERGY = 0.
FACTOR = 1.0
SU = DU = 0.
UA = UB = 0.
DEPOSIT = 0.
C -----
C READ (5+EBEAM)
C -----
IF (UB.GT.EMAX) UB = EMAX
FOR THE SQUARE WAVE S(U) = 1, UA < U < UB, THE AVERAGE ENERGY
UPLUS = <U+> IS GIVEN BY --
UPLUS = (UA + UB)/2.
SMAX = JB(1)
IF (SMAX.LT.0.) GO TO 519
Laser 344
Laser 345
Laser 346
Laser 347
Laser 348
Laser 349
Laser 350
Laser 351
Laser 352
Laser 353
Laser 354
Laser 355
Laser 356
Laser 357
Laser 358
Laser 359
Laser 360
Laser 361
Laser 362
Laser 363
Laser 364
Laser 365
Laser 366
Laser 367
Laser 368
Laser 369
Laser 370
Laser 371
Laser 372
Laser 373
Laser 374
Laser 375
Laser 376
Laser 377
Laser 378
Laser 379
Laser 380
Laser 381
Laser 382
Laser 383
Laser 384
Laser 385
Laser 386
Laser 387
Laser 388
Laser 389
Laser 390
Laser 391
Laser 392
Laser 393
Laser 394
Laser 395
Laser 396
Laser 397
Laser 398
Laser 399
Laser 400

```

```

NPTS = 1                                LASER    401
T0 = TB(1)                               LASER    402
DO 515 I = 2,21                           LASER    403
T1 = TB(I)                               LASER    404
IF (T1.LE.T0) GO TO 518                 LASER    405
NPTS = NPTS+1                           LASER    406
SI = JB(I)                               LASER    407
IF (SI.LT.0.) GO TO 519                 LASER    408
IF (SI.GT.SMAX) SMAX = SI               LASER    409
515 T0 = T1                             LASER    410
518 INTRP = NPTS.GT.1                  LASER    411
C COMPUTE NORMALIZED E-BEAM CURRENT DENSITY SHAPE FUNCTION --
DO 521 I = 1,NPTS                      LASER    412
521 JB(I) = JB(I)/SMAX                LASER    413
GO TO 520                             LASER    414
519 INTRP = .FALSE.                     LASER    415
LASER    416
LASER    417
C 520 KVOLT = 0.                         LASER    418
PDISCH = 0.                            LASER    419
AREA = DIST = 1.                       LASER    420
INDUCT = RESIST = 0.                   LASER    421
CAPAC = 1.0                            LASER    422
LASER    423
C -----
C READ (5,CIRCUIT)                     LASER    424
C -----
C VOLT = 1000.*KVOLT                  LASER    425
ELECT = KVOLT.NE.0.                    LASER    426
IF (DIST.EQ.0.) DIST = 1.0            LASER    427
LASER    428
LASER    429
LASER    430
LASER    431
C ELECT = ELECT.AND.NK.NE.0          LASER    432
NEQ = NTYPE                           LASER    433
IF (ELECT) NEQ = NEQ+2              LASER    434
NP1 = NTYPE+1                         LASER    435
NP2 = NTYPE+2                         LASER    436
LASER    437
C -----
C READ UPDATED RATES FOR ALL PROCESSES -- LASER    438
C -----
C C READ NUMERICAL RATE DATA PROVIDED AT THE TIME OF PROGRAM SYNTHESIS LASER    439
C (STORED ON TAPE NSCRTCH), AND (POSSIBLY) MODIFIED BY INPUT FROM LASER    440
C THE SRATES ... $ CARD. NOTE THAT CERTAIN RATES ARE ABSENT IF THEY LASER    441
C DID NOT APPEAR AT PROGRAM SYNTHESIS. THESE RATES ARE NOT ACCESSIBLE LASER    442
C BY INPUT, AND ARE IGNORED IF AN ATTEMPT IS MADE TO SPECIFY LASER    443
C THEM ON THE SRATES ... $ CARD. HOWEVER, FOR SECONDARY ELECTRON LASER    444
C COLLISIONS (FOR WHICH RATES ARE NORMALLY OBTAINED BY DEFAULT TO LASER    445
C E-KINETICS CALCULATIONS), FIXED INPUT VALUES FOR RATES MAY BE LASER    446
C ASSIGNED BY THE SRATES ... $ CARD IF THERE IS NO ELECTRIC FIELD LASER    447
C (I.E., EVCM = 0 SPECIFIED ON THE FOREGOING SCIRCUIT ... $ CARD). LASER    448
C -----
C READ (5,RATES)                        LASER    449
C -----
REWIND MTAPE                           LASER    450
LASER    451
LASER    452
LASER    453
LASER    454
LASER    455
LASER    456
LASER    457

```

```

STOP = .FALSE.
K0 = 24
IF (KOUNT.LE.0) KOUNT = K0
IF (KOUNT.GT.K0) KOUNT = K0
NSKIP = K0 - KOUNT
IF (NSKIP.LE.0) NSKIP = 1
ENCODE (10+109,PAGE) NSKIP
L = LC = K = N = 0
ILLEGAL = MODIFY = .FALSE.
71 READ (INSCRTH) LSUM, LSUM, LABEL, RATE, FK, RK, RNAME, COMMENT
IF (EOF(INSCRTH)) 70,91
91 NFLAG = 1H
TEST = .FALSE.
K = K+1
DO 84 I = 1,NTYPE
NL = NR = 0
DO 503 J = 1,5
IF (LABEL(J,1).EQ.1) NL = NL+1
503 IF (LABEL(J,2).EQ.1) NR = NR+1
NI = NR-NL
IF (I.EQ.1.AND.NL.EQ.1.AND.NR.EQ.1) NI = 1
84 NTIME(I) = NI
IMAGE(1) = RATE(1)
IMAGE(2) = RATE(2)
DECODE (1+100,RATE(1)) R1
DECODE (1+100,RATE(2)) R2
IF (R2.EQ.1HV) IMAGE(2) = 1H
IF (R1.NE.1HV.AND.R2.NE.1HV) GO TO 73
N = N+1
VSIG(1+N) = VSIG(2,N) = 0.
C COUNT NET NUMBER OF ELECTRONS (RHS-LHS) --
C
NEL(N) = 0
DO 92 M = 1,5
IF (LABEL(M,1).EQ.2) NEL(N) = NEL(N)-1
92 IF (LABEL(M,2).EQ.2) NEL(N) = NEL(N)+1
C
IF (.NOT.ELECT) GO TO 74
IF (R1.EQ.1HV) IMAGE(1) = 1H
IF (R2.EQ.1HV) IMAGE(2) = 1H
GO TO 73
C
C (IF EVCM = 0, AND IF KF OR KR ARE NOT SPECIFIED FOR A SECONDARY
C ELECTRON PROCESS, IT IS ASSUMED BY THE PROGRAM THAT THERE WAS AN
C IMPLIED INPUT OF KF = 0 AND/OR KR = 0 FOR THAT PROCESS.)
C
74 IF (R1.NE.1HV) GO TO 76
IF (KF(K).EQ.0) KF(K) = 0.
VSIG(1,N) = KF(K)
76 IF (R2.NE.1HV) GO TO 73
IF (KR(K).EQ.0) KR(K) = 0.
VSIG(2,N) = KR(K)
73 IF (KF(K).EQ.0) GO TO 67
IF (IMAGE(1).NE.1H) GO TO 67
KF(K) = C
L = L+1

```

LASER	458
LASER	459
LASER	460
LASER	461
LASER	462
LASER	463
LASER	464
LASER	465
LASER	466
LASER	467
LASER	468
LASER	469
LASER	470
LASER	471
LASER	472
LASER	473
LASER	474
LASER	475
LASER	476
LASER	477
LASER	478
LASER	479
LASER	480
LASER	481
LASER	482
LASER	483
LASER	484
LASER	485
LASER	486
LASER	487
LASER	488
LASER	489
LASER	490
LASER	491
LASER	492
LASER	493
LASER	494
LASER	495
LASER	496
LASER	497
LASER	498
LASER	499
LASER	500
LASER	501
LASER	502
LASER	503
LASER	504
LASER	505
LASER	506
LASER	507
LASER	508
LASER	509
LASER	510
LASER	511
LASER	512
LASER	513
LASER	514

```

J = IMF           LASER  515
NFLAG = 2H**     LASER  516
TEST = ILLEGAL = .TRUE.   LASER  517
ENCODE (10,305,LINE(L)) J, K   LASER  518
STOP = .TRUE.     LASER  519
67 IF (KR(K).EQ.C) GO TO 68   LASER  520
IF (IMAGE(2).NE.1H) GO TO 68   LASER  521
KR(K) = C          LASER  522
L = L+1          LASER  523
J = IHR          LASER  524
NFLAG = 2H**     LASER  525
TEST = ILLEGAL = .TRUE.   LASER  526
ENCODE (10,305,LINE(L)) J, K   LASER  527
STOP = .TRUE.     LASER  528
C      AT THIS POINT, WITH THE EXCEPTION OF CHANGES ENCOUNTERED AND PER- LASER  529
C     MITTED BY INPUT, VECTORS KF AND KR CONTAIN ALL C-VALUES, AND LASER  530
C      LINE(L), CONTAINS NAMES OF RATES INACCESSIBLE BY INPUT FOR THE PRO- LASER  531
C     GRAM EXECUTION.   LASER  532
C      LASER  533
C      LASER  534
68 IF (KF(K).EQ.C) GO TO 63   LASER  535
ENCODE (50,323,COMMENT)       LASER  536
MODIFY = .TRUE.               LASER  537
IF (.NOT.TEST) NFLAG = 2H *   LASER  538
ENCODE (10,301,RATE(1)) KF(K)   LASER  539
IF (KF(K).EQ.0.) RATE(1) = 1H   LASER  540
GO TO 64          LASER  541
63 KF(K) = FK          LASER  542
64 IF (KR(K).EQ.C) GO TO 69   LASER  543
ENCODE (50,323,COMMENT)       LASER  544
MODIFY = .TRUE.               LASER  545
IF (.NOT.TEST) NFLAG = 2H *   LASER  546
ENCODE (10,301,RATE(2)) KR(K)   LASER  547
IF (KR(K).EQ.0.) RATE(2) = 1H   LASER  548
GO TO 10          LASER  549
69 KR(K) = RK          LASER  550
10 WRITE (MTAPE) (NTIME(I), I = 1,NTYPE), RATE, RNAME   LASER  551
IF (.NOT.OUT(10)) GO TO 71   LASER  552
IF (LC.NE.0) GO TO 2         LASER  553
IF (K.EQ.1) GO TO 83        LASER  554
WRITE (6,102)                 LASER  555
IF (MODIFY) WRITE (6,123)     LASER  556
IF (ILLEGAL) WRITE (6,124)    LASER  557
83 ILLEGAL = MODIFY = .FALSE.   LASER  558
WRITE (6,PAGE)                LASER  559
WRITE (6,103) GENDATE        LASER  560
IF (ELECT) WRITE (6,219)      LASER  561
WRITE (6,105)                 LASER  562
2 LC = LC+1                 LASER  563
IF (LC.EQ.KOUNT) LC = 0       LASER  564
WRITE (6,104) NFLAG, K, (RNAME(J), J = 1,5), RATE, COMMENT   LASER  565
GO TO 71          LASER  566
C      LASER  567
70 IF (.NOT.OUT(10)) GO TO 93   LASER  568
WRITE (6,102)                 LASER  569
IF (MODIFY) WRITE (6,123)     LASER  570
IF (ILLEGAL) WRITE (6,124)    LASER  571

```

```

C 93 ERROR(7) = STOP
C -----
C BEGIN GENERATION OF ERROR DIAGNOSTICS --
C -----
C LINES = 6
C      WRITE (6,212)
C
C IF (.NOT.ERROR(1)) GO TO 514
C      WORD = WARN
C      IF (FATAL(1)) WORD = 6HFATAL:
C      WRITE (6,201) WORD
C      LINES = LINES+5
C
C 514 IF (.NOT.ERROR(8)) GO TO 516
C      WORD = WARN
C      IF (FATAL(8)) WORD = 6HFATAL:
C      WRITE (6,208) WORD, MAXGAS, MAXK, MAXNK
C      LINES = LINES+5
C
C 516 IF (.NOT.ERROR(9)) GO TO 525
C      WORD = WARN
C      IF (FATAL(9)) WORD = 6HFATAL:
C      WRITE (6,209) WORD, NKMAX, KMAX, NMAX, NK, KTYPE, NTYPE
C      LINES = LINES+5
C
C 525 IF (.NOT.ERROR(7)) GO TO 75
C      WORD = WARN
C      IF (FATAL(7)) WORD = 6HFATAL:
C      WRITE (6,207) WORD
C      LINES = LINES + 5 + (L+7)/8
C      WRITE (6,108) (LINE(I), I = 1,L)
C -----
C
C 75 DO 5 I = 1,NTYPE
C      MASS(I) = 0.
C      5 E(I) = NO(I) = C
C
C READ INPUT DATA: NAMES, CONCENTRATIONS, ENERGIES, MASSES --
C
C      L = NGAS = 0
C      PRESS = 0.
C      IONIZE = 1.0E-12
C
C 20 READ (5,101) TYPE, RATE, DUMMY, DUMMY, NPLOT
C      IF (EOF(5)) 11,31
C
C 31 BACKSPACE 5
C      READ (5,112) P0, E0, MOLWT
C      IF (TYPE.NE.3HRAD) GO TO 34
C
C SPECIES *1* CORRESPONDS TO RADIATION --
C      NO(1) = P0
C      GO TO 20
C
C 34 IF (TYPE.NE.4HE(-)) GO TO 39
C SPECIES *2* CORRESPONDS TO ELECTRONS --
C      NO(2) = P0
C      PLOTS(2) = NPLOT.EQ.4HPLT
C
C      LASER 572
C      LASER 573
C      LASER 574
C      LASER 575
C      LASER 576
C      LASER 577
C      LASER 578
C      LASER 579
C      LASER 580
C      LASER 581
C      LASER 582
C      LASER 583
C      LASER 584
C      LASER 585
C      LASER 586
C      LASER 587
C      LASER 588
C      LASER 589
C      LASER 590
C      LASER 591
C      LASER 592
C      LASER 593
C      LASER 594
C      LASER 595
C      LASER 596
C      LASER 597
C      LASER 598
C      LASER 599
C      LASER 600
C      LASER 601
C      LASER 602
C      LASER 603
C      LASER 604
C      LASER 605
C      LASER 606
C      LASER 607
C      LASER 608
C      LASER 609
C      LASER 610
C      LASER 611
C      LASER 612
C      LASER 613
C      LASER 614
C      LASER 615
C      LASER 616
C      LASER 617
C      LASER 618
C      LASER 619
C      LASER 620
C      LASER 621
C      LASER 622
C      LASER 623
C      LASER 624
C      LASER 625
C      LASER 626
C      LASER 627
C      LASER 628

```

```

      GO TO 20
39 IF (TYPE.NE.6HIONIZE) GO TO 81
  IONIZE = P0
  PLOTS(2) = NPLOT.EQ.4HPLOT
  GO TO 20
81 PRESS = PRESS + P0
  IF (P0.LE.0.) GO TO 58
  NGAS = NGAS+1
  NAME(NGAS) = TYPE
  MASS(NGAS) = MOLWT
  FI(NGAS) = P0
58 DO 8 I = 3,NTYPE
  IF (TYPE.EQ.GAS(I)) GO TO 9
  CONTINUE
  L = L+1
  LINE(L) = TYPE
  ERROR(2) = .TRUE.
  GO TO 20
9 IF (RATE(1).NE.1H ) NO(1) = P0*0.965E 19/THOL
  IF (RATE(2).NE.1H ) E(1) = E0
  PLOTS(1) = NPLOT.EQ.4HPLOT
  GO TO 20
C   11 IF (.NOT.ERROR(2)) GO TO 526
    WORD = WARN
    IF (FATAL(2)) WORD = 6HFATAL:
    WRITE (6,201) WORD
    LINES = LINES + 5 + (L+7)/8
    WRITE (6,108) (LINE(I), I = 1,L)
C   526 DO 6 I = 1,NGAS
  6 FI(I) = FI(I)/PRESS
  IF (PTOT.EQ.0.) GO TO 27
  DO 28 I = 3,NTYPE
28 IF (NO(I).NE.C) NO(I) = NO(I)*PTOT/PRESS
  GO TO 26
27 PTOT = PRESS
26 NMOL = NTOT = 0.965E 19*PTOT/THOL
  IF (NO(2).LE.0.) NO(2) = IONIZE*NMOL
C   CHECK INITIALIZATION OF POPULATION DENSITIES --
  L = 0
  DO 12 I = 1,NTYPE
  IF (NO(I).NE.C) GO TO 12
  ERROR(3) = .TRUE.
  L = L+1
  LINE(L) = GAS(I)
  NO(I) = 0.
12 DNYDTN(I,1) = NO(I)
C   IF (.NOT.ERROR(3)) GO TO 527
    WORD = WARN
    IF (FATAL(3)) WORD = 6HFATAL:
    WRITE (6,203) WORD
    LINES = LINES + 5 + (L+7)/8
    WRITE (6,108) (LINE(I), I = 1,L)
C

```

LASER      629  
 LASER      630  
 LASER      631  
 LASER      632  
 LASER      633  
 LASER      634  
 LASER      635  
 LASER      636  
 LASER      637  
 LASER      638  
 LASER      639  
 LASER      640  
 LASER      641  
 LASER      642  
 LASER      643  
 LASER      644  
 LASER      645  
 LASER      646  
 LASER      647  
 LASER      648  
 LASER      649  
 LASER      650  
 LASER      651  
 LASER      652  
 LASER      653  
 LASER      654  
 LASER      655  
 LASER      656  
 LASER      657  
 LASER      658  
 LASER      659  
 LASER      660  
 LASER      661  
 LASER      662  
 LASER      663  
 LASER      664  
 LASER      665  
 LASER      666  
 LASER      667  
 LASER      668  
 LASER      669  
 LASER      670  
 LASER      671  
 LASER      672  
 LASER      673  
 LASER      674  
 LASER      675  
 LASER      676  
 LASER      677  
 LASER      678  
 LASER      679  
 LASER      680  
 LASER      681  
 LASER      682  
 LASER      683  
 LASER      684  
 LASER      685

```

C      CHECK INITIALIZATION OF ENERGIES --
527 L = 0                                LASER   686
      E(1) = E(2) = 0.                      LASER   687
      DO 15 I = 3,NTYPE                   LASER   688
      IF (E(I).NE.C) GO TO 15              LASER   689
          E(I) = 0.
          ERROR(4) = .TRUE.
          L = L+1
          LINE(L) = GAS(I)
15 CONTINUE
C
      IF (.NOT.ERROR(4)) GO TO 528        LASER   690
          WORD = WARN
          IF (FATAL(4)) WORD = 6HFATAL:
          WRITE (6,204) WORD
          LINES = LINES + 5 + (L+7)/8
          WRITE (6,108) (LINE(I), I = 1,L)
C
      528 IF (.NOT.ELECT) GO TO 80        LASER   691
C
C -----
C
C      IF THE BOLTZMANN ANALYSIS BELOW DOES NOT CONVERGE, CONTROL RETURNS
C      TO THIS POINT TO EXPAND THE ELECTRON ENERGY RANGE --
C
      TA = 0.                                LASER   692
      IF (.NOT.REPEAT) GO TO 79             LASER   693
95 IF (EMAX.GE.S.) GO TO 87               LASER   694
      EMAX = EMAX*2.
      IF (EMAX.GT.S.) EMAX = S.
      GO TO 89
87 EMAX = EMAX + 5.
89 IF (EMAX.LE.ELIMIT) GO TO 79         LASER   695
      WRITE (6,220) EMAX, ELIMIT
      GO TO 97
79 DE = EMAX/MESH
      E0 = 0.
      NA = UA/DE + 1
      NB = UB/DE + 1
      IF ((UA.NE.UB).AND.(NA.EQ.NB)) NB = NA+1
      UA = DE*(NA-1)
      UB = DE*(NB-1)
      DU = UB-UA
      SU = 0.
      IF (DU.GT.0.) SU = 1./DU
      DO 29 I = 1,MESHPI
      QMOM(I,1) = QMOM(I,2) = 0.
      EV(I) = E0
      S(I) = 0.
      IF (I.LT.NA.OR.I.GT.NB) GO TO 29
      S(I) = SU
29 E0 = E0 + DE
C
C      S(I) IS THE NORMALIZED SOURCE FUNCTION FOR SECONDARY ELECTRON
C      CREATION: INT(DU S(U)) = 1.
C
      REWIND NSCRTCH

```

```

READ (NSCRTCH) TITLE LASER 743
READ (NSCRTCH) LASER 744
READ (NSCRTCH) LASER 745
READ (NSCRTCH) LASER 746
C ----- LASER 747
C ----- LASER 748
C ----- LASER 749
C IF THE SYNTHESIZED PROGRAM WAS CONSTRUCTED TO DEFAULT TO ELECTRON LASER 750
C KINETICS CALCULATIONS AS THE SOURCE FOR SECONDARY ELECTRON COLLISION LASER 751
C RATES, AND IF THE EXPERIMENTAL SITUATION CORRESPONDS TO AN LASER 752
C ELECTRIC DISCHARGE (EVCM ≠ 0), THE ELECTRON CROSS SECTION FILE IS LASER 753
C PROCESSED -- LASER 754
C ----- LASER 755
C CALL SECOND (T0) LASER 756
IN = NTYPE LASER 757
K = N = 0 LASER 758
J = 1 LASER 759
STOP = .FALSE. LASER 760
14 READ (NSCRTCH) LHS, RHS, LABEL, RATE, FK, RK, RNAME LASER 761
IF (EOF(NSCRTCH)) 60,505 LASER 762
505 DECODE (1,100,RATE(1)) R1 LASER 763
DECODE (1,100,RATE(2)) R2 LASER 764
IF (R1.NE.1HV.AND.R2.NE.1HV) GO TO 14 LASER 765
N = N+1 LASER 766
ENCODE (40,101,PROCESS(1,N)) (RNAME(L), L = 1,4) LASER 767
C ----- LASER 768
C PROCESS THE INELASTIC ELECTRON CROSS SECTION FILE -- LASER 769
C ----- LASER 770
C ----- LASER 771
C CALL PLASMA (INDATA, MGRID+1, MESH, LHS, RHS, RNAME, EV, F, G, LASER 772
1 Q(1,N), U0, UM, IN, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(B)) LASER 773
C TEST = MISSING.OR.REJECT.OR.OUTSIDE LASER 774
IF (.NOT.TEST) GO TO 32 LASER 775
STOP = STOP.OR.TEST LASER 776
K = K+1 LASER 777
IF (J.GT.231) GO TO 33 LASER 778
ENCODE (50,322,LINE(J)) LASER 779
J = J+5 LASER 780
ENCODE (50,129,LINE(J)) K, (PROCESS(L,N), L = 1,4) LASER 781
J = J+5 LASER 782
IF (MISSING) ENCODE (50,115,LINE(J)) LASER 783
IF (MISSING) J = J+5 LASER 784
IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX LASER 785
IF (OUTSIDE) J = J+5 LASER 786
IF (REJECT) ENCODE (50,117,LINE(J)) EMAX LASER 787
IF (REJECT) J = J+5 LASER 788
IF (J.LE.225) GO TO 33 LASER 789
ENCODE (100,128,LINE(J)) LASER 790
J = J+10 LASER 791
GO TO 33 LASER 792
C 32 DO 24 L = 1,MESHPI LASER 793
24 Q(L,N) = EV(L)*Q(L,N) LASER 794
33 U(N) = 0. LASER 795
L1 = LEV1(N) LASER 796

```

```

IF (L1.EQ.0) GO TO 19      LASER    800
U(N) = - E(L1)             LASER    801
GO TO 13                  LASER    802
19 DO 21 L = 1,5           LASER    803
  I = LABEL(L,1)            LASER    804
  IF (I.EQ.0) GO TO 13     LASER    805
21 U(N) = U(N)-E(I)       LASER    806
13 L2 = LEV2(N)            LASER    807
  IF (L2.EQ.0) GO TO 23    LASER    808
  U(N) = U(N) + E(L2)     LASER    809
  GO TO 38                 LASER    810
23 DO 22 L = 1,5           LASER    811
  I = LABEL(L,2)            LASER    812
  IF (I.EQ.0) GO TO 38     LASER    813
22 U(N) = U(N)+E(I)       LASER    814
38 CONTINUE                LASER    815
  GO TO 14                 LASER    816
C
60 ERROR(5) = STOP         LASER    817
  FE = FATAL(5).AND.ERROR(5)
C
C-----PROCESS THE MOMENTUM TRANSFER CROSS SECTIONS -- -----
C
IN = NTYPE                LASER    820
TWOM = 2./1837.              LASER    821
STOP = .FALSE.               LASER    822
DO 17 I = 1,NGAS            LASER    823
  IF (FI(I).EQ.0.) GO TO 17
  FRACT = FI(I)
  MISSING = REJECT = OUTSIDE = .FALSE.
  ENCODE (40,106,RNAME) NAME(I)
C
  ENCODE (50+120,IMAGE) NAME(I), NAME(I)
  DECODE (50+100,IMAGE) (MOM(L), L = 1,50)
  CALL DEKODE (GAS, MOM, LHS, RHS, LABEL, DUM, 10, IN, 50)
C
  CALL PLASMA (INDATA, MGRID+1, MESH, LHS, RHS, RNAME, EV, F, G, QM,
1 U0, UM, IN, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(8))
C
TEST = MISSING.OR.REJECT.OR.OUTSIDE.OR.(MASS(I).LE.0.).OR.
1   (UM.LT.EMAX)
  IF (.NOT.TEST) GO TO 36
  K = K+1
  IF (J.GT.221) GO TO 17
  ENCODE (50+322+LINE(J))
  J = J+5
  ENCODE (50+129+LINE(J)) K, (RNAME(L), L = 1,4)
  J = J+5
  IF (MISSING) ENCODE (50+115+LINE(J))
  IF (MISSING) J = J+5
  IF (OUTSIDE) ENCODE (50+116+LINE(J)) EMAX
  IF (OUTSIDE) J = J+5
  IF (REJECT)  ENCODE (50+117+LINE(J)) EMAX
  IF (REJECT)  J = J+5
  IF (UM.LT.EMAX) ENCODE (50+118+LINE(J)) EMAX

```

```

IF (UM.LT.EMAX) J = J+5                                LASER    857
IF (MASS(I).LE.0.) ENCODE (50,119,LINE(J))             LASER    858
IF (MASS(I).LE.0.) J = J+5                                LASER    859
IF (J.LE.225) GO TO 17                                 LASER    860
ENCODE (100,128,LINE(J))                               LASER    861
J = J+10                                              LASER    862
GO TO 17                                              LASER    863
C                                                       LASER    864
C                                                       LASER    865
C CONSTRUCT TWO MOMENTUM TRANSFER FUNCTIONS WHICH OCCUR IN THE LASER    866
C BOLTZMANN EQUATION --                                LASER    867
C                                                       LASER    868
C                                                       LASER    869
36 DO 44 L = 1,MESHPI                                LASER    870
FQ = FRACT*QM(L)                                     LASER    871
QMOM(L,1) = QMOM(L,1) + FQ                           LASER    872
44 QMOM(L,2) = QMOM(L,2) + FQ/MASS(I)                LASER    873
C                                                       LASER    874
17 STOP = STOP.OR.TEST                               LASER    875
C                                                       LASER    876
ERROR(6) = STOP                                     LASER    877
FE = FE.OR.(FATAL(6).AND.ERROR(6))                  LASER    878
ERRORS = ERROR(5).OR.ERROR(6)                         LASER    879
C                                                       LASER    880
C GENERATE ADDITIONAL WARNING DIAGNOSTICS FOR E- DATA LASER    881
C                                                       LASER    882
C                                                       LASER    883
C                                                       LASER    884
IF (.NOT.ERRORS) GO TO 98                            LASER    885
WORD = WARN                                         LASER    886
IF (FE) WORD = 6HFATAL:                             LASER    887
J = J-1                                             LASER    888
LINES = LINES + 5 + (J*4)/5                         LASER    889
IF (REPEAT.OR.OUT(8).OR.OUT(9).OR.(LINES.GT.55))   LASER    890
1      WRITE (6,212)                                LASER    891
LINES = 6                                           LASER    892
WRITE (6,205) WORD                                LASER    893
WRITE (6,206) (LINE(L), L = 1,J)                   LASER    894
C                                                       LASER    895
98 FATAL(10) = .TRUE.                               LASER    896
DO 46 L = 1,MESHPI                                LASER    897
X = EV(L)                                         LASER    898
XSQ = TWOM*X*X                                     LASER    899
ERROR(10) = QMOM(L,1).LE.0.                          LASER    900
IF (ERROR(10)) GO TO 47                           LASER    901
A(L,1) = X/NMOL/QMOM(L,1)                          LASER    902
46 A(L,2) = XSQ*NMOL*QMOM(L,2)                    LASER    903
C                                                       LASER    904
XBAR = DE/2.                                       LASER    905
DO 41 I = 1,MESH                                    LASER    906
CALL INTERP (2, XBAR, QMOM(I,1), EV, A(1,1), 1, MESHPI)
CALL INTERP (2, XBAR, QMOM(I,2), EV, A(1,2), 1, MESHPI)
41 XBAR = XBAR + DE                                LASER    907
QMOM(MESHPI,1) = QMOM(MESH,1)                      LASER    908
QMOM(MESHPI,2) = QMOM(MESH,2)                      LASER    909
C                                                       LASER    910
47 WORD = WARN                                     LASER    911
C                                                       LASER    912
C                                                       LASER    913

```

```

IF (FATAL(10)) WORD = 6HFATAL:
IF (ERROR(10)) WRITE (6,210) WORD
C
CALL SECOND (T1)
TA = TA + (T1 - T0)
C
KTE = KB*TE
EXPON = EXP(-DE/KTE)
FB = 1.
DO 37 I = 1,MESHPI
F(I) = FB
37 FB = FB*EXPON
C
C PROHIBIT FURTHER PLOTS OR TABULATIONS OF E- CROSS SECTION DATA --
C
OUT(8) = OUT(9) = .FALSE.
C
C -----
C
C TEST FOR ERROR CONDITIONS --
80 FE = ERRORS = .FALSE.
C
DO 94 I = 1,10
ERRORS = ERRORS.OR.ERROR(I)
FE = FE.OR.(ERROR(I).AND.FATAL(I))
C
RESET ERROR FLAG --
94 ERROR(I) = .FALSE.
IF (FE) WRITE (6,300)
IF (FE) GO TO 99
C
IF (REPEAT) GO TO 90
IF (SU.EQ.0.) GO TO 45
C
C PLOT OF NORMALIZED EXTERNAL IONIZATION SOURCE FUNCTION --
C
WRITE (6,130) UPLUS
Y0(1) = DY(1) = 0.
CALL PLOT (1, MESH+1, 1, S, Y0, DY, EV, 0., 0., .TRUE., .TRUE.,
1 .TRUE., .TRUE., .TRUE., TITLE, 1, 0)
WRITE (6,131)
C
45 MU = 1000.
TBOLTZ = NBOLTZ = 0
VMAX = 0.9999*KVOLT
NO(NP1) = CAPAC*VOLT
NO(NP2) = 0.
RD = DIST/AREA/(EE*MU)
IF (INDUCT.EQ.0.) NO(NP2) = -VOLT/(RESIST + RD)
DNYDTN(NP1+1) = NO(NP1)
DNYDTN(NP2+1) = NO(NP2)
HMIN = TOUT/1000.
HMAX = TOUT
H = HMIN
DELTA = ETA
JSTART = 0
T = TP = NP = 0
TT = TOUT
Laser 914
Laser 915
Laser 916
Laser 917
Laser 918
Laser 919
Laser 920
Laser 921
Laser 922
Laser 923
Laser 924
Laser 925
Laser 926
Laser 927
Laser 928
Laser 929
Laser 930
Laser 931
Laser 932
Laser 933
Laser 934
Laser 935
Laser 936
Laser 937
Laser 938
Laser 939
Laser 940
Laser 941
Laser 942
Laser 943
Laser 944
Laser 945
Laser 946
Laser 947
Laser 948
Laser 949
Laser 950
Laser 951
Laser 952
Laser 953
Laser 954
Laser 955
Laser 956
Laser 957
Laser 958
Laser 959
Laser 960
Laser 961
Laser 962
Laser 963
Laser 964
Laser 965
Laser 966
Laser 967
Laser 968
Laser 969
Laser 970

```

```

      ENCODE (40+322,KAPTION)          LASER    971
C-----LASER    972
C-----LASER    973
C-----LASER    974
C-----LASER    975
C-----LASER    976
C-----LASER    977
C-----LASER    978
C-----LASER    979
C-----LASER    980
C-----LASER    981
C-----LASER    982
C-----LASER    983
C-----LASER    984
C-----LASER    985
C-----LASER    986
C-----LASER    987
C-----LASER    988
C-----LASER    989
C-----LASER    990
C-----LASER    991
C-----LASER    992
C-----LASER    993
C-----LASER    994
C-----LASER    995
C-----LASER    996
C-----LASER    997
C-----LASER    998
C-----LASER    999
C-----LASER   1000
C-----LASER   1001
C-----LASER   1002
C-----LASER   1003
C-----LASER   1004
C-----LASER   1005
C-----LASER   1006
C-----LASER   1007
C-----LASER   1008
C-----LASER   1009
C-----LASER   1010
C-----LASER   1011
C-----LASER   1012
C-----LASER   1013
C-----LASER   1014
C-----LASER   1015
C-----LASER   1016
C-----LASER   1017
C-----LASER   1018
C-----LASER   1019
C-----LASER   1020
C-----LASER   1021
C-----LASER   1022
C-----LASER   1023
C-----LASER   1024
C-----LASER   1025
C-----LASER   1026
C-----LASER   1027

C-----INTEGRATE EQUATIONS FROM T = 0 TO T = TPULSE --
C-----DO 30 I = 1,9
C-----HOUT = TOUT - TT
C-----IF (H.LT.HOUT) GO TO 18
C-----DO 16 I = 1,9
C-----OUT(I) = .FALSE.
C-----IF (IO(I).EQ.0) GO TO 16
C-----OUT(I) = NP.EQ.IO(I)*(NP/IO(I))
C-----CONTINUE
C-----NP = NP+1
C-----TP = T + HOUT
C-----TIME(NP) = TP/UNIT
C-----IBEAM(NP) = JBEAM*SHAPE(TP)
C-----IF (LIMIT.NE.0) ENCODE (40+121,KAPTION) TP
C-----CALCULATE EXTRAPOLATED VALUES OF THE POPULATION DENSITIES --
C-----SH = HOUT/H
C-----SJ = 1.
C-----JP1 = JSTART+1
C-----DO 25 I = 1,NP2
C-----25 PUP(I) = 0.
C-----DO 65 J = 1,JP1
C-----DO 66 I = 1,NP2
C-----66 PUP(I) = PUP(I) + DNYDTN(I,J)*SJ
C-----65 SJ = SJ*SH
C-----NE = PUP(2)

C-----ELECTRON KINETICS ANALYSIS
C-----IF (.NOT.ELECT) GO TO 48
C-----CALL "DNDT" TO DETERMINE S0 AND SB PRIOR TO CALLING THE BOLTZMANN
C-----ANALYSIS. THE ELECTRON SOURCE FUNCTION IS: SEXT(U) = S0*DELTA(U)
C-----+ SB*S(U).
C-----CALL DNDT (NEO, TP, POP, NDOT)
C-----IF (DU.EQ.0.) S0 = S0 + SB
C-----IF (DU.EQ.0.) SB = 0.
C-----CHARGE = POP(NP1)
C-----CURRENT = -POP(NP2)
C-----RD = DIST/AREA/(EE*MU)
C-----IF (NE.GT.0.) RD = RD/NE
C-----VOLT = CURRENT*RD
C-----ITER = ITMAX

```

```

IF (VOLT.EQ.0.) ITER = 0 LASER 1028
EVCM = ABS(VOLT/DIST) LASER 1029
ESQ = EVCM*EVCM LASER 1030
DNEDT = NDUT(2) LASER 1031
C LASER 1032
NOUT(NP,1,NP1) = VC = CHARGE/CAPAC/1000. LASER 1033
NOUT(NP,2,NP1) = KVOLT = VOLT/1000. LASER 1034
NOUT(NP,1,NP2) = VR = CURRENT*RESIST/1000. LASER 1035
NOUT(NP,2,NP2) = LDIDT = VC - VR - KVOLT LASER 1036
C LASER 1037
CALL LEVELS (N1, N2, POP) LASER 1038
C LASER 1039
IF (NP.EQ.1) GO TO 90 LASER 1040
DP = NE*ELASTIC + EE*(UBAR*DNEDT - NE*MU*ESQ - UPLUS*SB)
DEPOSIT = IBEAM(NP)*DVDX LASER 1041
PDISCH = NE*EE*MU*ESQ LASER 1042
PCOLL = 0. LASER 1043
DO 524 J = 1,NK LASER 1044
IF (NEL(J).GE.0) GO TO 530 LASER 1045
PWR = N1(J)*POWER(J)
GO TO 524 LASER 1046
530 PWR = EE*U(J)*(N1(J)*VSIG(1,J) - N2(J)*VSIG(2,J))
524 PCOLL = PCOLL + PWR LASER 1047
DP = DP + NE*PCOLL LASER 1048
BEFORE = 100.*DP/(PDISCH + DEPOSIT) LASER 1049
C CALL SECOND (T0) LASER 1050
C -----
C CALL BOLTZ (MGRID+1, MESH, NK, NAME, FI, NGAS, NMOL, TMOL, ITER,
1 TMAX, EPS, KAPTION, TODAY, OUT, EVCM, NE, PROCESS, U, N1, N2,
2 NEL, S, SB, SG, EV, Q, QMOM, F, G, A, B, VSIG, POWER, PCOLL,
3 PDISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE, ATTACH,
4 VD, MIJ, D, EK, AMPS, UBAR, TE, CONVRGE, PERCENT) LASER 1051
C -----
C E(2) = UBAR LASER 1052
CALL SECOND (T1) LASER 1053
TBOLTZ = TBOLTZ + (T1 - T0) LASER 1054
NBOLTZ = NBOLTZ + 1 LASER 1055
C -----
C IF (ITER.LE.0) PERCENT = 0. LASER 1056
REJECT = (.NOT.CONVRGE).OR.(PERCENT.GT.PCT) LASER 1057
REPEAT = REJECT.AND.EXPAND LASER 1058
IF (REPEAT) GO TO 95 LASER 1059
IF (REJECT) GO TO 99 LASER 1060
C OBTAIN ELECTRON PARAMETERS, NORMALIZED WITHOUT POPULATION DENSITY: LASER 1061
IF (NE.NE.0.) ELASTIC = ELASTIC/NE LASER 1062
DO 523 J = 1,NK LASER 1063
IF (NEL(J).GE.0) GO TO 523 LASER 1064
IF (N1(J).NE.0.) POWER(J) = POWER(J)/N1(J) LASER 1065
523 IF (NE.NE.0.) POWER(J) = POWER(J)/NE LASER 1066
C -----

```

```

C THE RATES PRODUCED BY SUBROUTINE RATES HERE ARE USED FOR OUTPUT      LASER 1085
C INFORMATION ONLY --                                                 LASER 1086
C                                                               LASER 1087
C                                                               LASER 1088
C -----                                                 LASER 1089
48 CALL DNDR (NEQ, TP, POP, NDOT)                                                 LASER 1090
C                                                               LASER 1091
C                                                               LASER 1092
C (NOTE: DO NOT INTERCHANGE POP AND NO.)                                                 LASER 1093
RD = DIST/AREA/(EE*MU)                                                 LASER 1094
IF (NE.GT.0.) RD = RD/NE                                                 LASER 1095
COND(NP,1) = SIGMA = EE*MU*NE                                                 LASER 1096
COND(NP,2) = RD                                                 LASER 1097
JSUS = EE*NE*VD                                                 LASER 1098
JSUS(NP,1) = SIGMA*EVCM                                                 LASER 1099
JSUS(NP,2) = - NDOT(NP2)/AREA                                                 LASER 1100
C E(1) = HNU                                                 LASER 1101
C CAVITY PHOTON DENSITY --                                                 LASER 1102
PHOTON = POP(1)                                                 LASER 1103
DNPHDT = NDOT(1)                                                 LASER 1104
LASER 1105
C EFFECTIVE PHOTON DENSITY IN MEDIUM --                                                 LASER 1106
NDOT(1) = (CAVITY/LENGTH)*NDOT(1)                                                 LASER 1107
POP(1) = (CAVITY/LENGTH)*POP(1)                                                 LASER 1108
LASER 1109
C ETOT = DUDT = 0.                                                 LASER 1110
DO 86 I = 1,NTYPE                                                 LASER 1111
ITAU(I) = 1H                                                 LASER 1112
TAU = 0.                                                 LASER 1113
IF (NDOT(I).NE.0.) TAU = POP(I)/NDOT(I)                                                 LASER 1114
TAU = ABS(TAU)                                                 LASER 1115
TAU = TAU/UNIT                                                 LASER 1116
IF (TAU.NE.0.) ENCODE (10,FFMT,ITAU(I)), TAU                                                 LASER 1117
IF (TAU.GT.1.E 04) ENCODE (10,EFMT,ITAU(I)), TAU                                                 LASER 1118
NOUT(NP,1,I) = POP(I)                                                 LASER 1119
NOUT(NP,2,I) = NDOT(I)                                                 LASER 1120
ETOT = ETOT + E(I)*POP(I)                                                 LASER 1121
86 DUDT = DUDT + E(I)*NDOT(I)                                                 LASER 1122
ETOT = EE*ETOT                                                 LASER 1123
DUDT = EE*DUDT                                                 LASER 1124
LASER 1125
C NOUT(I) = DNPHDT                                                 LASER 1126
POP(I) = PHOTON                                                 LASER 1127
RAD(NP,1) = RADIATE = 3.0E 10*HNU*PHOTON                                                 LASER 1128
RAD(NP,2) = 3.0E 10*HNU*DNPHDT                                                 LASER 1129
RAD(NP,3) = PBEAM = DVDX*IBEAM(NP)                                                 LASER 1130
PSTIM = GAMMA*RADIATE                                                 LASER 1131
HEAT = PDISCH + PBEAM - DUDT - PSTIM                                                 LASER 1132
P = PBEAM                                                 LASER 1133
LASER 1134
C ALPHA(NP,1) = GNET                                                 LASER 1135
ALPHA(NP,2) = GAMMA                                                 LASER 1136
ALPHA(NP,3) = GAIN                                                 LASER 1137
ALPHA(NP,4) = ABSORB                                                 LASER 1138
LASER 1139
C IF (.NOT.OUT(6)) GO TO 61                                                 LASER 1140
LASER 1141

```

```

C-----LASER 1142
C-----LASER 1143
C-----LASER 1144
C-----LASER 1145
C-----LASER 1146
C-----LASER 1147
C-----LASER 1148
C-----LASER 1149
C-----LASER 1150
C-----LASER 1151
C-----LASER 1152
C-----LASER 1153
C-----LASER 1154
C-----LASER 1155
C-----LASER 1156
C-----LASER 1157
C-----LASER 1158
C-----LASER 1159
C-----LASER 1160
C-----LASER 1161
C-----LASER 1162
C-----LASER 1163
C-----LASER 1164
C-----LASER 1165
C-----LASER 1166
C-----LASER 1167
C-----LASER 1168
C-----LASER 1169
C-----LASER 1170
C-----LASER 1171
C-----LASER 1172
C-----LASER 1173
C-----LASER 1174
C-----LASER 1175
C-----LASER 1176
C-----LASER 1177
C-----LASER 1178
C-----LASER 1179
C-----LASER 1180
C-----LASER 1181
C-----LASER 1182
C-----LASER 1183
C-----LASER 1184
C-----LASER 1185
C-----LASER 1186
C-----LASER 1187
C-----LASER 1188
C-----LASER 1189
C-----LASER 1190
C-----LASER 1191
C-----LASER 1192
C-----LASER 1193
C-----LASER 1194
C-----LASER 1195
C-----LASER 1196
C-----LASER 1197
C-----LASER 1198

C-----OUTPUT OF POPULATION DENSITIES AND THEIR RATES OF CHANGE, AND
C-----MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS --
C-----ENCODE OPTICAL AND ELECTRICAL PARAMETERS --
L = 1
ENCODE (120,307,LINE(L)) IBEM(NP), FACTOR, ENERGY
L = L+12
IF (ENERGY.LE.0.) L = L-4
DVDX = DVDX/1000.
PBEM = PBEM/1000.
ENCODE (80,325,LINE(L)) DVDX, PBEM
L = L*8
ENCODE (80,308,LINE(L)) SB, SO
L = L*8
ENCODE (40,322,LINE(L))
L = L*4
IF (.NOT.ELECT) GO TO 53
C-----KVCM = KVOLT/DIST
PDISCH = PDISCH/1000.
P = P + PDISCH
IF (P.NE.0.) AFTER = PERCENT*(DEPOSIT + PDISCH)/P
C-----ENCODE (120,317,LINE(L)) AREA, DIST, COND(NP,1)
L = L+12
ENCODE (40,309,LINE(L)) RD
L = L*4
ENCODE (120,310,LINE(L)) CHARGE, CURRENT
L = L+12
ENCODE (120,311,LINE(L)) JSUS, KVCM, PDISCH
L = L+12
ENCODE (120,312,LINE(L)) VC, KVOLT
L = L+12
IF (RESIST.EQ.0.) GO TO 52
ENCODE (40,313,LINE(L)) VR
L = L*4
52 IF (INDUCT.EQ.0.) GO TO 53
ENCODE (40,314,LINE(L)) LDIDT
L = L*4
53 IF (.NOT.STIM) GO TO 54
ENCODE (120,318,LINE(L)) REFLECT, PASS
L = L+12
IF (CAVITY.EQ.LENGTH) GO TO 504
ENCODE (80,324,LINE(L)) TCAVITY, CAVITY
L = L*8
504 ENCODE (120,319,LINE(L)) LENGTH, OMEGA4P, GAMMA
L = L+12
ENCODE (120,321,LINE(L)) GNET, GAIN, ABSORB
L = L+12
ENCODE (40,322,LINE(L))
L = L*4
IF (P.EQ.0.) P = 1.E 99
PSTIM = PSTIM/1000.
EFF = 100.*PSTIM/P

```

```

        ENCODE (120,315,LINE(L)) RADIATE, PSTIM, EFF          LASER    1199
        L = L+12                                              LASER    1200
        IF (ABS(EFF).GT.100.) L = L-4                          LASER    1201
C
      54 DEDT = DEDT/1000.
        DUDT = DUDT/1000.
        HEAT = HEAT/1000.
        ENCODE (120,316,LINE(L)) DEDT, DUDT, ETOT          LASER    1202
        L = L+12                                              LASER    1203
        ENCODE (40,327,LINE(L)) HEAT                         LASER    1204
        L = L+4                                              LASER    1205
        IF (.NOT.ELECT) GO TO 529                           LASER    1206
        ENCODE (80,326,LINE(L)) BEFORE, AFTER                LASER    1207
        L = L+8                                              LASER    1208
      529 ENCODE (40,322,LINE(L))                           LASER    1209
        L = L+4                                              LASER    1210
        ENCODE (120,320,LINE(L))                           LASER    1211
        LMAX = L+3                                         LASER    1212
C
      L = 0                                              LASER    1213
      WRITE (6,110) KAPTION, UNIT                         LASER    1214
      DO 35 I = 1,NTYPE                                    LASER    1215
        WRITE (6,400) I, GAS(I), E(I), POP(I), NDOT(I), ITAU(I),
        1 (LINE(L+K), K = 1,4)                            LASER    1216
      35 IF (L.LT.LMAX) L = L+4                          LASER    1217
        LC = NTYPE+6                                     LASER    1218
        WRITE (6,401) (LINE(L+K), K = 1,4)               LASER    1219
        L = L+4                                              LASER    1220
        STEP = H/UNIT                                    LASER    1221
        WRITE (6,107) STEP, (LINE(L+K), K = 1,4)          LASER    1222
        L = L+4                                              LASER    1223
        WRITE (6,114) JSTART, (LINE(L+K), K = 1,4)         LASER    1224
        L = L+4                                              LASER    1225
        WRITE (6,401) (LINE(L+K), K = 1,4)               LASER    1226
        L = L+4                                              LASER    1227
        WRITE (6,107) STEP, (LINE(L+K), K = 1,4)          LASER    1228
        L = L+4                                              LASER    1229
        WRITE (6,114) JSTART, (LINE(L+K), K = 1,4)         LASER    1230
        L = L+4                                              LASER    1231
        WRITE (6,401) (LINE(L+K), K = 1,4)               LASER    1232
        L = L+4                                              LASER    1233
        WRITE (6,111) (LINE(L+K), K = 1,4)               LASER    1234
        L = L+4                                              LASER    1235
      51 IF (L.GE.LMAX) GO TO 77                         LASER    1236
        WRITE (6,401) (LINE(L+K), K = 1,4)               LASER    1237
        LC = LC+1                                         LASER    1238
        L = L+4                                              LASER    1239
        GO TO 51                                         LASER    1240
      77 NSKIP = 43-LC
        IF (NSKIP.LT.1) NSKIP = 1
        ENCODE (80,500,IMAGE) NSKIP
        WRITE (6,IMAGE) TODAY                           LASER    1241
C
C   -----
C   SENSITIVITY ANALYSIS OF REACTION SCHEME --
C   -----
C
      61 IF (OUT(7).AND.NP.NE.1) CALL ANALYZE (NTYPE, KTYPE, RATEK, NTIME,
        1 RPCT, FLAG, PMAX, GAS, PER, KAPTION, LTAPE, MTAPE, NTAPE) LASER    1242
C
        ARC = NE/NTOT.GT.RE
        END = END.OR.ARC
        IF (.NOT.END) GO TO 78                           LASER    1243
                                                LASER    1244
                                                LASER    1245
                                                LASER    1246
                                                LASER    1247
                                                LASER    1248
                                                LASER    1249
                                                LASER    1250
                                                LASER    1251
                                                LASER    1252
                                                LASER    1253
                                                LASER    1254
                                                LASER    1255

```

```

NF = NP-1
WRITE (6,225) NF, TP
IF (NBOLTZ.NE.0) WRITE (6,127) TA, NBOLTZ, TBOLTZ
IF (ARC) WRITE (6,224) NE, RE
GO TO 97
C
C-----CONTINUE TO INTEGRATE THE EQUATIONS FROM A DEAD START FROM THIS
C CYCLE POINT --
C-----78 T = TP
DO 49 I = 1,NEQ
49 DNYDTN(I,1) = POP(I)
JSTART = NFLAG = 0
DELTA = ETA
C-----18 CALL GFAR (NEQ, T, DNYDTN, SCRATCH, H, HMIN, HMAX, DELTA, METHOD,
I YMAX, ERR, KFLAG, JSTART, MAXDER, NMAXP2, PHI)
C-----TT = T - TP
IF (KFLAG.EQ.1) GO TO 30
JSTART = 0
NFLAG = NFLAG+1
IF (NFLAG.EQ.1) GO TO 18
H = H/100.
IF (INFLAG.LE.5) GO TO 18
DELTA = 2.*DELTA
IF (INFLAG.LE.10) GO TO 18
C 97 IF (NP.LT.15) GO TO 99
C-----OUTPUT GENERATION
C-----SUMMARY OF UNIMPORTANT REACTIONS --
C-----REWIND MTAPE
KOUNT = 25
LC = 0
DO 507 K = 1,KTYPE
READ (MTAPE) (INTIME(I), I = 1,NTYPE), RATE, RNAME
IF (.NOT.FLAG(K)) GO TO 507
IF (LC.EQ.0) WRITE (6,402) PER
LC = LC+1
IF (LC.EQ.KOUNT) LC = 0
WRITE (6,403) K, RATE, (RNAME(L), L = 1,4)
507 CONTINUE
C-----SUMMARY OF IMPORTANT REACTIONS --
C-----REWIND MTAPE
LC = 0
DO 501 K = 1,KTYPE

```

LASER	1256
LASER	1257
LASER	1258
LASER	1259
LASER	1260
LASER	1261
LASER	1262
LASER	1263
LASER	1264
LASER	1265
LASER	1266
LASER	1267
LASER	1268
LASER	1269
LASER	1270
LASER	1271
LASER	1272
LASER	1273
LASER	1274
LASER	1275
LASER	1276
LASER	1277
LASER	1278
LASER	1279
LASER	1280
LASER	1281
LASER	1282
LASER	1283
LASER	1284
LASER	1285
LASER	1286
LASER	1287
LASER	1288
LASER	1289
LASER	1290
LASER	1291
LASER	1292
LASER	1293
LASER	1294
LASER	1295
LASER	1296
LASER	1297
LASER	1298
LASER	1299
LASER	1300
LASER	1301
LASER	1302
LASER	1303
LASER	1304
LASER	1305
LASER	1306
LASER	1307
LASER	1308
LASER	1309
LASER	1310
LASER	1311
LASER	1312

```

READ (MTAPE) (INTIME(I), I = 1,NTYPE), RATE, RNAME
IF (FLAG(K)) GO TO 501
IF (LC.EQ.0) WRITE (6,404) PER
LC = LC+1
IF (LC.EQ.KOUNT) LC = 0
WRITE (6,403) K, RATE, (RNAME(L), L = 1,4)
501 CUNTINUE
C
      WRITE (6,216)
      CALL PLOT (M1, NP, 1, IBEAM, 0., 0., TIME, 0., DTIME, .FALSE.,
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 1, 0)
      WRITE (6,306) UNIT, TODAY
C
      IF (.NOT.STIM) GO TO 40
      KAPTION(1) = 10HINTENSITY
      KAPTION(2) = SHDI/DT
      WRITE (6,218)
      CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 2, 0)
      WRITE (6,306) UNIT, TODAY
C
      40 KAPTION(1) = 10HNET GAIN
      KAPTION(2) = 10HTHRESHOLD
      NPLOT = 2
      IF (.NOT.STIM) NPLOT = 1
      WRITE (6,113)
      CALL PLOT (M1, NP, 1, ALPHA, 0., 0., TIME, 0., DTIME, .FALSE.,
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)
      WRITE (6,306) UNIT, TODAY
C
      KAPTION(1) = 10HLASER GAIN
      KAPTION(2) = 10HABSORPTION
      WRITE (6,126)
      CALL PLOT (M1, NP, 1, ALPHA(1,3), 0., 0., TIME, 0., DTIME, .FALSE.,
1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, 2, 0)
      WRITE (6,306) UNIT, TODAY
C
      NI = 0
      IF (.NOT.STIM) GO TO 502
      NI = NI+1
      KAPTION(1) = 10HOPTICAL
      DO 82 I = 1,NP
      ALPHA(I,3) = ALPHA(I,4) = 0.
      RAD(I+1) = ALPHA(I,1)*RAD(I+1)
A2 ALPHA(I+1) = RAD(I,1)/1000.
502 IF (.NOT.ELECT) GO TO 511
      NI = NI+1
      KAPTION(NI) = 10HELECTRICAL
      DO 88 I = 1,NP
      KVCM = NOUT(I,2,NP1)/DIST
      AMPS = ISUS(I,1)
      ALPHA(I,4) = KVCM*AMPS
      RR RAD(I+NI) = 1000.*ALPHA(I,4)
511 IF (JBEAM.EQ.0.) GO TO 96
      NI = NI+1
      KAPTION(NI) = 10HE-BEAM
      DO 512 I = 1,NP

```

LASER	1313
LASER	1314
LASER	1315
LASER	1316
LASER	1317
LASER	1318
LASER	1319
LASER	1320
LASER	1321
LASER	1322
LASER	1323
LASER	1324
LASER	1325
LASER	1326
LASER	1327
LASER	1328
LASER	1329
LASER	1330
LASER	1331
LASER	1332
LASER	1333
LASER	1334
LASER	1335
LASER	1336
LASER	1337
LASER	1338
LASER	1339
LASER	1340
LASER	1341
LASER	1342
LASER	1343
LASER	1344
LASER	1345
LASER	1346
LASER	1347
LASER	1348
LASER	1349
LASER	1350
LASER	1351
LASER	1352
LASER	1353
LASER	1354
LASER	1355
LASER	1356
LASER	1357
LASER	1358
LASER	1359
LASER	1360
LASER	1361
LASER	1362
LASER	1363
LASER	1364
LASER	1365
LASER	1366
LASER	1367
LASER	1368
LASER	1369

```

RAD(I,NI) = RAD(I,3)
512 ALPHA(I,3) = RAD(I,3)/1000.
96 IF (NI.EQ.0) GO TO 508
NPLOT = NI
WRITE (6,122)
CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,
I .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)
WRITE (6,306) UNIT, TODAY
C
IF (.NOT.STIM) GO TO 509
C COMPUTE POWER EFFICIENCY --
DO 510 I = 1,NP
P = RAD(I,3)
IF (ELECT) P = P + RAD(I,2)
IF (P.EQ.0.) P = 1.E 99
ALPHA(I,2) = 100.*RAD(I,1)/P
510 IF (ABS(ALPHA(I,2)).GT.100.) ALPHA(I,2) = 0.
C
C -----
C INTEGRATE POWER DENSITIES --
C -----
C
509 DO 506 I = 1,3
B1 = RAD(1,I)
RAD(1,I) = 0.
DO 506 K = 2,NP
K1 = K-1
B2 = B1
B1 = RAD(K,I)
506 RAD(K,I) = RAD(K+1,I) + 0.5E 03*TOUT*(B1 + B2)
C
WRITE (6,125)
CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,
I .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)
WRITE (6,306) UNIT, TODAY
C
C -----
C TABULAR SUMMARY OF MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS
C -----
C
I2 = 1
522 I1 = I2
I2 = I2+50
IF (I2.GT.NP) I2 = NP
WRITE (6,223) UNIT
WRITE (6,226) (TIME(I), IBEAM(I), ALPHA(I,3), NOUT(I,2,NP),
I ISUS(I,1), ALPHA(I,4), ALPHA(I,1), ALPHA(I,2), I = I1:I2)
IF (I2.LT.NP) GO TO 522
C
IF (.NOT.STIM) GO TO 508
DO 513 I = 1,NP
ALPHA(I,3) = RAD(I,1)
P = RAD(I,3)
IF (ELECT) P = P + RAD(I,2)
IF (P.EQ.0.) P = 1.E 99
513 ALPHA(I,4) = 100.*RAD(I,1)/P

```

LASER	1370
LASER	1371
LASER	1372
LASER	1373
LASER	1374
LASER	1375
LASER	1376
LASER	1377
LASER	1378
LASER	1379
LASER	1380
LASER	1381
LASER	1382
LASER	1383
LASER	1384
LASER	1385
LASER	1386
LASER	1387
LASER	1388
LASER	1389
LASER	1390
LASER	1391
LASER	1392
LASER	1393
LASER	1394
LASER	1395
LASER	1396
LASER	1397
LASER	1398
LASER	1399
LASER	1400
LASER	1401
LASER	1402
LASER	1403
LASER	1404
LASER	1405
LASER	1406
LASER	1407
LASER	1408
LASER	1409
LASER	1410
LASER	1411
LASER	1412
LASER	1413
LASER	1414
LASER	1415
LASER	1416
LASER	1417
LASER	1418
LASER	1419
LASER	1420
LASER	1421
LASER	1422
LASER	1423
LASER	1424
LASER	1425
LASER	1426

```

C ----- PLOT OPTICAL POWER DENSITY AND EFFICIENCY -- ----- C
C KAPTION(1) = 10HPOWER/VOL C
C KAPTION(2) = 10HEFFICIENCY C
C WRITE (6,221) C
C CALL PLOT (M1, NP, 1, ALPHA, 0., 0., TIME, 0., DTIME, .FALSE..,
C .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)
C WRITE (6,306) UNIT, TODAY C
C ----- PLOT OPTICAL ENERGY DENSITY AND EFFICIENCY -- ----- C
C KAPTION(1) = 10HENERGY/VOL C
C KAPTION(2) = 10HEFFICIENCY C
C WRITE (6,222) C
C CALL PLOT (M1, NP, 1, ALPHA(1:3), 0., 0., TIME, 0., DTIME, .FALSE..,
C .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)
C WRITE (6,306) UNIT, TODAY C
C 508 IF (.NOT.ELECT) GO TO 85
C KAPTION(1) = 6HSUS C
C KAPTION(2) = 8HDISUS/DT C
C WRITE (6,214) C
C CALL PLOT (M1, NP, 1, ISUS, 0., 0., TIME, 0., DTIME, .FALSE..,
C .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)
C WRITE (6,306) UNIT, TODAY C
C KAPTION(1) = 6HSIGMA C
C KAPTION(2) = 10HRD (OHM) C
C Y0(1) = Y0(2) = DY(1) = 0. C
C DY(2) = 0.5 C
C WRITE (6,215) C
C CALL PLOT (M1, NP, 1, COND, Y0, DY, TIME, 0., DTIME, .FALSE..,
C .FALSE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0)
C WRITE (6,306) UNIT, TODAY C
C NPOINT = 4 C
C KAPTION(1) = 10HCAPACITOR C
C KAPTION(2) = 10HDISCHARGE C
C KAPTION(3) = 10HRESISTANCE C
C KAPTION(4) = 10HINDUCTANCE C
C IF (INDUCT.EQ.0.) NPOINT = 3 C
C VMIN = 0. C
C IF (NOUT(NP,1,NP1).LT.VMIN) VMIN = NOUT(NP,1,NP1) C
C IF (NOUT(NP,2,NP1).LT.VMIN) VMIN = NOUT(NP,2,NP1) C
C IF (NOUT(NP,1,NP2).LT.VMIN) VMIN = NOUT(NP,1,NP2) C
C IF (NOUT(NP,2,NP2).LT.VMIN) VMIN = NOUT(NP,2,NP2) C
C CALL AXIS (.TRUE., VMAX, VMIN, Y0, DY, YDC) C
C Y0(2) = DY(2) = 0. C
C TEST = YDC.NE.0. C
C WRITE (6,217) C
C CALL PLOT (M1, NP, 1, NOUT(1:1,NP1), Y0, DY, TIME, 0., DTIME, C

```

```

1 .FALSE., TEST, .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0) LASER 1484
      WRITE (6,306) UNIT, TODAY LASER 1485
C
C      85 DO 50 I = 2,NTYPE LASER 1486
      IF (.NOT.PLOTS(I)) GO TO 50 LASER 1487
      ENCODE (20,211,KAPTION) I, I LASER 1488
      WRITE (6,227) GAS(I) LASER 1489
      CALL PLOT (M1, NP, 1, NOUT(1,1,I), 0., 0., TIME, 0., DTIME,
      1 .FALSE., .TRUE., .FALSE., .TRUE., .TRUE., KAPTION, 2, 0) LASER 1490
      WRITE (6,306) UNIT, TODAY LASER 1491
      50 CONTINUE LASER 1492
C----- FORMAT STATEMENTS ----- LASER 1493
C
C      100 FORMAT (80A1) LASER 1494
C
C      101 FORMAT (8A10) LASER 1495
C
C      102 FORMAT (/1X,134(1H-)//) LASER 1496
C
C      103 FORMAT (18X,*SUMMARY OF UPDATED RATES FOR INPUT REACTION SCHEME OF LASER 1497
      1 SYNTHETIC KINETICS CODE GENERATED ON*A10/25X*DR. WILLIAM B. LACIN LASER 1498
      2A, NORTHROP RESEARCH AND TECHNOLOGY CENTER, PALOS VERDES, CALIFORN LASER 1499
      3IA*) LASER 1500
C
C      104 FORMAT (/1X,A2,I4+3X+4A10,A5+2X,A10,3X,A10,5X,5A10) LASER 1501
C
C      105 FORMAT (/6X*I*6X*REACTION(I)*36X*KF(I)*8X*KR(I)*20X*REFERENCES OR LASER 1502
      2COMMENTS*/1X,134(1H-)) LASER 1503
C
C      106 FORMAT (*MOMENTUM TRANSFER FOR *,A10) LASER 1504
C
C      107 FORMAT (43X*INTEGRATION STEP SIZE ==F10.3+6X+4A10) LASER 1505
C
C      108 FORMAT ((30X,8A10)) LASER 1506
C
C      109 FORMAT (*(*1H1*I2*(/))* ) LASER 1507
C
C      110 FORMAT (1H1/55X+4A10//24X*POPULATION DENSITIES AND RATES OF CHANG LASER 1508
      1E*20X*ELECTRICAL AND OPTICAL PARAMETERS//13X*I*5X*SPECIES*6X*E(I) LASER 1509
      2*8X*N(I)*7X*DN(I)/DT*7X*TAU(I)*/20X*NAME*8X*(EV)*7X*(CM-3)*5X*(CM- LASER 1510
      33/SEC)*1PE10.1* SEC*7X*PARAMETER*6X*VALUE*5X*UNITS*/9X,70(1H-)+3X+
      440(1H-)/) LASER 1511
C
C      111 FORMAT (9X,70(1H-),3X+4A10) LASER 1512
C
C      112 FORMAT (10X,3E10.3) LASER 1513
C
C      113 FORMAT (1H1/53X*INSTANTANEOUS NET GAIN COEFFICIENT (CM-1)*//) LASER 1514
C
C      114 FORMAT (44X*ORDER OF INTEGRATION ==I10,6X+4A10) LASER 1515
C
C      115 FORMAT (10X*NO E- CROSS SECTION DATA WAS FOUND.* ) LASER 1516
C
C      116 FORMAT (10X*SIGMA = 0 IN THE RANGE {0.+F4.1}* EV.* ) LASER 1517
C
C      117 FORMAT (10X*ERRORS OCCURRED IN CROSS SECTION DATA.* ) LASER 1518
C
C

```

C	118 FORMAT (10X*CROSS SECTION DOES NOT SPAN [0,*F4.1*] EV.*)	LASER	1541
C	119 FORMAT (10X*THE MASS ENTERED FOR THIS SPECIES IS ≤ 0*)	LASER	1542
C	120 FORMAT (A10* + E + E + *A10)	LASER	1543
C	121 FORMAT (8X*TIME T = *.*1PE10.3* SEC*)	LASER	1544
C	122 FORMAT (1H1,56X*POWER DENSITY (WATT/CM3)*//)	LASER	1545
C	123 FORMAT (45X** THE ORIGINAL RATE CONSTANT(S) HAVE BEEN MODIFIED#)	LASER	1546
C	124 FORMAT (44X** ILLEGAL ATTEMPT TO MODIFY RATE CONSTANT(S) WAS REJECTED#)	LASER	1547
C	125 FORMAT (1H1,56X*ENERGY DENSITY (JOULE/LITER)*//)	LASER	1548
C	126 FORMAT (1H1,42X*LASER GAIN AND MEDIUM ABSORPTION COEFFICIENTS (CM-11)*//)	LASER	1549
C	127 FORMAT (40X*THE PROCESSING OF ELECTRON CROSS SECTIONS REQUIRED*, 1F5.1* CP SEC.*/40X*AND*13* BOLTZMANN ELECTRON CALCULATIONS CONSUME 2D*.F6.1* CP SEC.*/)	LASER	1550
C	128 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)	LASER	1551
C	129 FORMAT (I2* *4A10.5X)	LASER	1552
C	130 FORMAT (1H1,24X,*NORMALIZED EXTERNAL SOURCE FUNCTION FOR CREATION 10F ELECTRONS IN THE ENERGY RANGE (U, U + DU)*/43X*AVERAGE ENERGY 0 2F CREATED ELECTRONS = < U+ > = *F6.2* EV*/)	LASER	1553
C	131 FORMAT (/62X*ELECTRON ENERGY U (EV)*)	LASER	1554
C	201 FORMAT (/15X,A10*ERRORS WERE DETECTED IN PROCESSING THE INPUT REACTION SCHEME. MODIFICATIONS OF THE REACTION*/25X*SCHEME, CORRECTIO 2NS IN REACTION SYNTAX, CHANGES IN DIMENSION STORAGE, OR ADDITIONS 3TO THE E-*/25X*CROSS SECTION FILE MAY BE REQUIRED TO REMOVE ALL OF 4 THE ERROR CONDITIONS.*)	LASER	1555
C	202 FORMAT (/15X,A10*THE FOLLOWING SPECIES WERE INITIALIZED BY INPUT 1DATA, BUT DID NOT OCCUR IN THE KINETIC SYSTEM.*/25X*IF THE PROGRAM 2 IS EXECUTED, THEY WILL BE IGNORED, BUT INCLUDED IN THE TOTAL PRES 3SURE (AS BUFFER*/25X*GASES IN 3-BODY COLLISIONS) AND FOR MOMENTUM 4TRANSFER IN E- KINETICS ANALYSIS:*)	LASER	1556
C	203 FORMAT (/15X,A10*INITIAL POPULATION DENSITIES FOR THE FOLLOWING SPECIES WERE NOT DEFINED. IF THE PROGRAM IS EXECUTED, NO(I) 2 = 0 WILL BE ASSUMED:*)	LASER	1557
C	204 FORMAT (/15X,A10*ENERGIES FOR THE FOLLOWING SPECIES WERE NOT DEFINED. IF PROGRAM IS EXECUTED, E(I) = 0 WILL BE ASSUMED:*)	LASER	1558
C	205 FORMAT (/15X,A10*ERRORS OCCURRED FOR THE INPUT CROSS SECTIONS FOR THE FOLLOWING ELECTRON COLLISION PROCESSES.*/25X*IF THE PROGRAM IS EXECUTED, A ZERO CROSS SECTION OVER ALL ENERGY WILL BE ASSUMED:*)	LASER	1559

3/1

C	206 FORMAT ((40X,5A10))	LASER	1598
C	207 FORMAT ((//15X,A10*AN ATTEMPT WAS MADE TO ENTER. BY SRATES ... S IN INPUT, THE FOLLOWING RATES, WHICH ARE NOT ACCESSIONS-*/25X*ISIBLE BY INPUT 2 FOR THE SYNTHETIC PROGRAM THAT WAS GENERATED. IF THE PROGRAM EXECUTES, THE AT-*/25X*TEMPTED MODIFICATIONS WILL BE IGNORED, AND THE 4 ORIGINAL RATES USED*))	LASER	1599
C	208 FORMAT ((//15X,A10*SYNTHESIZED SUBROUTINES AND DATA FILE WERE GENERATED WITH DIMENSION DECLARATORS NMAX =*I4*, /*25X*KMAX =*I4*, AND N 2KMAX =*I4*. THESE MUST AGREE WITH THE CORRESPONDING DIMENSION DECLARATORS /*25X*IN THE MAIN PROGRAM (LASER). EXECUTION MAY BE POSSIBLE IF PRESENT STORAGE EXCEEDS ORIGINAL, BUT /*25X*CAUTION IS ADVISED TO INSURE THAT LABELED COMMON BLOCKS AGREE WITH THOSE IN THE 6 SYNTHETIC/*25X*SUBROUTINES ASSOCIATED WITH THE DATA FILE GENERATE 7D ON TAPE4.*))	LASER	1600
C	209 FORMAT ((//15X,A10*TAPE4 DATA FILE VECTORS EXCEEDED DIMENSION STORE 1GE IN MAIN PROGRAM, AND WERE TRUNCATED DURING /*25X*READ. NKMAX =*I4*, KMAX =*I4*, AND NMAX =*I4*. TAPE4 CONTAINS NK =*I4*, KTYPE = 3*I4*, AND /*25X*NTYPE =*I4*. CAUTION IS ADVISED.))	LASER	1601
C	210 FORMAT ((//15X,A10*MOMENTUM TRANSFER COLLISION FREQUENCY IS ZERO AT 1 SOME POINT. E- ANALYSIS CONTAINS 1/QM TERMS.))	LASER	1602
C	211 FORMAT (2HN(,I2,IH)*5X,3HDN(,I2+4H)/DT,1X)	LASER	1603
C	212 FORMAT (1H1/55X*SUMMARY OF POSSIBLE ERROR CONDITIONS*///15X*SEVER 1)TY*45X*DESCRIPTION /*15X,105(1H-))	LASER	1604
C	214 FORMAT (1H1,25X*DISCHARGE CURRENT DENSITY JSUS (AMP/CM2), AND ITS RATE OF CHANGE D/DT(JSUS) (A/CM2/S)*//)	LASER	1605
C	215 FORMAT (1H1,20X*PLASMA CONDUCTIVITY SIGMA = F*NE*MU (/CM/OHM), AND 1 DISCHARGE IMPEDANCE RD = D/AREA/SIGMA (OHM)*//)	LASER	1606
C	216 FORMAT (1H1/40X*E-BEAM CURRENT DENSITY (AMP/CM2) AS A FUNCTION OF 1TIME*//)	LASER	1607
C	217 FORMAT (1H1/56X*CIRCUIT VOLTAGES (KVOLT)*//)	LASER	1608
C	218 FORMAT (1H1/27X*INTRACAVITY RADIATION INTENSITY I (W/CM2) AND ITS RATE OF CHANGE DI/DT (W/CM2/SEC)*//)	LASER	1609
C	219 FORMAT (/13X*(VSIG(K,I) ARE FORWARD (K=1) OR REVERSE (K=2) RATES FOR THE ITH INELASTIC PROCESS IN THE E- KINETICS ANALYSIS*))	LASER	1610
C	220 FORMAT (1H1,20(/),40X*ELECTRON ENERGY RANGE EMAX =*,0PF6.2* EXCEED 1S MAXIMUM VALUE =*0PF6.2)	LASER	1611
C	221 FORMAT (1H1,40X*OPTICALLY EXTRACTED POWER DENSITY (KW/CM3) AND EFFICIENCY (%)*//)	LASER	1612
C	222 FORMAT (1H1,35X*OPTICALLY EXTRACTED ENERGY DENSITY (JOULE/LITER) AND EFFICIENCY (%)*//)	LASER	1613
		LASFR	1614
		LASER	1615
		LASER	1616
		LASER	1617
		LASER	1618
		LASER	1619
		LASER	1620
		LASER	1621
		LASER	1622
		LASER	1623
		LASER	1624
		LASER	1625
		LASER	1626
		LASER	1627
		LASER	1628
		LASER	1629
		LASER	1630
		LASER	1631
		LASER	1632
		LASER	1633
		LASER	1634
		LASER	1635
		LASER	1636
		LASER	1637
		LASER	1638
		LASER	1639
		LASER	1640
		LASER	1641
		LASER	1642
		LASER	1643
		LASER	1644
		LASER	1645
		LASER	1646
		LASER	1647
		LASER	1648
		LASER	1649
		LASER	1650
		LASER	1651
		LASER	1652
		LASER	1653
		LASFR	1654

C	223 FORMAT (1H1.45X#SUMMARY OF ELECTRICAL AND OPTICAL PARAMETERS// 123X#TIME#7X#JBEAM#7X#PBEAM#6X#VOLTAGE#7X#JSUS#7X#DISCH#7X#OUTPUT# 26X#OPT EFF#/18X#(*,1PE8.2,* SEC)* 1X*(A/CM2)*4X*(KW/CM3)*7X*(KV)*. 36X*(A/CM2)*4X*(KW/CM3)*5X*(KW/CM3)*7X*(%)*/18X.99(1H-)/)	LASER	1655
C	224 FORMAT (40X#ELECTRON DENSITY NE =*1PE10.3* EXCEEDS LIMIT NE/NMOL S 1 *1PE10.3)	LASER	1656
C	225 FORMAT (/40X#CALCULATION IS TERMINATED AT CYCLE NP =*I3*, T = *1PE 110.3,* SEC.*)	LASER	1657
C	226 FORMAT (18X,F9.2,5F12.2,F13.2,F11.2)	LASER	1658
C	227 FORMAT (1H1/32X,*POPULATION DENSITY N (CM-3) AND RATE OF CHANGE DN 1/DT (CM-3/S) FOR *,A10/)	LASER	1659
C	300 FORMAT (///25X#PROGRAM IS TERMINATED FOR ERRORS SPECIFIED FATAL.*)	LASER	1660
C	301 FORMAT (1PE10.4)	LASER	1661
C	302 FORMAT (///25X,B6(1H*)///)	LASER	1662
C	303 FORMAT (28X,8A10)	LASER	1663
C	304 FORMAT (1H1.5(/))	LASER	1664
C	305 FORMAT (*K*,A1,*(*,I3,*))	LASER	1665
C	306 FORMAT (/62X#TIME (*1PE9.3* SEC)*/100X#DR. WILLIAM B. LACINA,*A1/ 1100X*NORTHROP RESEARCH AND TECHNOLOGY*)	LASER	1666
C	307 FORMAT (3X#JBEAM#5X* = *0PF10.2* AMP/CM2*,8X#DEPOSITION = *F10.2, 117X#ENERGY#4X* = *F10.0* KEV*,9X)	LASER	1667
C	308 FORMAT (3X*S(U > 0) = *1PE10.3* CM-3/SEC*7X*S(U = 0) = * 1 1PE10.3* CM-3/SEC*4X)	LASER	1668
C	309 FORMAT (3X*R(DISCH)*2X* = *1PE10.3* OHM*)	LASER	1669
C	310 FORMAT (43X*Q*9X* = *.1PE10.3* COULOMB*8X*I(DISCH)*2X* = *.1PE10. 13* AMP*9X)	LASER	1670
C	311 FORMAT (3X*J(DISCH)*2X* = *1PE10.3* AMP/CM2*8X*ESUS*6X* = *0PF10. 13* KVOLT/CM*7X*JSUS = #0PF10.2* KW/CM3*)	LASER	1671
C	312 FORMAT (43X*Q/C*7X* = *0PF10.3* KVOLT*10X*V(DISCH)*2X* = *0PF10.3 1* KVOLT*)	LASER	1672
C	313 FORMAT (3X*V(RESIST) = *0PF10.3* KVOLT*)	LASER	1673
C	314 FORMAT (3X*L*DI/DT*3X* = *0PF10.3* KVOLT*)	LASER	1674
C	315 FORMAT (3X*INTENSITY = *1PE10.3* WATT/CM2*.7X*OPTICAL*3X* = *. 10PF10.2* KW/CM3*9X*EFFICIENCY = *.0PF10.2* %*)	LASER	1675
C	316 FORMAT (3X*E<U>DNE/DT = #1PE10.3* KW/CM3*9X*DE(TOT)/DT = *E10.3* LASER	LASER	1676
			1677
			1678
			1679
			1680
			1681
			1682
			1683
			1684
			1685
			1686
			1687
			1688
			1689
			1690
			1691
			1692
			1693
			1694
			1695
			1696
			1697
			1698
			1699
			1700
			1701
			1702
			1703
			1704
			1705
			1706
			1707
			1708
			1709
			1710
			1711

C	1* KW/CM3*9X*E(TOT)*4X* = *E10.3* J/CM3*7X)	LASER	1712
C	317 FORMAT (3X*AREA*6X* = *0PF10.2* CM2*12X*DIST*6X* = *0PF10.2* CM* 113X*CONDUCT*3X* = *1PE10.3* /OHM/CM*)	LASER	1713
C	318 FORMAT (43X*REFLECT*3X* = *0PF10.2* %*14X*LOSS*6X* = *0PF10.2* % 1/PASS*)	LASER	1714
C	319 FORMAT (3X*LENGTH*4X* = *0PF10.2* CM*13X*OMEGA/4/PI = *1PE10.3* 117X*THRESH*4X* = *1PE10.3* CM-1*)	LASER	1715
C	320 FORMAT (40(1H-),80X)	LASER	1716
C	321 FORMAT (3X*NET GAIN = *1PE10.3.* CM-1*11X*LASER GAIN = *1PE10.3* 1* CM-1*11X*ABSORPTION = *,1PE10.3* CM-1*)	LASER	1717
C	322 FORMAT (10X)	LASER	1718
C	323 FORMAT (****** THE ORIGINAL RATE HAS BEEN MODIFIED ******)	LASER	1719
C	324 FORMAT (3X*T(CAVITY) = *F10.1* NS*13X*L(CAVITY) = *F10.2* CM*)	LASER	1720
C	325 FORMAT (3X*DVOX*6X* = *F10.2* KV/CM*10X*P(BEAM)*3X* = *F10.2* 1* KW/CM3*6X)	LASER	1721
C	326 FORMAT (3X*DP(BEFORE) = *F10.2* %*14X*DP(AFTER) = *F11.2* %*11X)	LASER	1722
C	327 FORMAT (3X*HEAT + SP = *1PE10.3* KW/CM3*6X)	LASER	1723
C	400 FORMAT (9X,I5,5X,A10,0PF7.2,1P2E14.3,2X,A10,6X,4A10)	LASER	1724
C	401 FORMAT (82X,4A10)	LASER	1725
C	402 FORMAT (1H1,36X*THE FOLLOWING REACTIONS CONTRIBUTE LESS THAN *F3.0 1* %//29X*K8X*KF(K)*10X*KR(K)*15X*REACTION(K)*/27X,75(1H-)/)	LASER	1726
C	403 FORMAT (25X,I5,5X,A10,5X,A10,5X,4A10/)	LASER	1727
C	404 FORMAT (1H1,40X*THE FOLLOWING REACTIONS WERE IMPORTANT (> *F3.0, 1* %//29X*K8X*KF(K)*10X*KR(K)*15X*REACTION(K)*/27X,75(1H-)/)	LASER	1728
C	500 FORMAT (1H1,I2,#/190X*DR. WILLIAM B. LACINA,*A11/90X*NORTHROP RES 1EARCH AND TECHNOLOGY*)#)	LASER	1729
C-----		LASER	1730
C	99 CALL EXIT	LASER	1731
C	END	LASER	1732
		LASER	1733
		LASER	1734
		LASER	1735
		LASER	1736
		LASER	1737
		LASER	1738
		LASER	1739
		LASER	1740
		LASER	1741
		LASER	1742
		LASER	1743
		LASER	1744
		LASER	1745
		LASER	1746
		LASER	1747
		LASER	1748
		LASER	1749
		LASER	1750
		LASER	1751
		LASER	1752
		LASER	1753
		LASER	1754
		LASER	1755
		LASER	1756
		LASER	1757
		LASER	1758
		LASER	1759

SUBROUTINE SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, NDATA, NSIZE,  
 1 MAXGAS, GAS, KMAX, NKMAX, LEVI, LEV2, DATE) ..... SYNTH 2  
 ..... SYNTH 3  
 ..... SYNTH 4  
 ..... SYNTH 5  
 ..... SYNTH 6  
 ..... SYNTH 7  
 ..... SYNTH 8  
 ..... SYNTH 9  
 ..... SYNTH 10  
 ..... SYNTH 11  
 ..... SYNTH 12  
 ..... SYNTH 13  
 ..... SYNTH 14  
 ..... SYNTH 15  
 ..... SYNTH 16  
 ..... SYNTH 17  
 ..... SYNTH 18  
 ..... SYNTH 19  
 ..... SYNTH 20  
 ..... SYNTH 21  
 ..... SYNTH 22  
 ..... SYNTH 23  
 ..... SYNTH 24  
 ..... SYNTH 25  
 ..... SYNTH 26  
 ..... SYNTH 27  
 ..... SYNTH 28  
 ..... SYNTH 29  
 ..... SYNTH 30  
 ..... SYNTH 31  
 ..... SYNTH 32  
 ..... SYNTH 33  
 ..... SYNTH 34  
 ..... SYNTH 35  
 ..... SYNTH 36  
 ..... SYNTH 37  
 ..... SYNTH 38  
 ..... SYNTH 39  
 ..... SYNTH 40  
 ..... SYNTH 41  
 ..... SYNTH 42  
 ..... SYNTH 43  
 ..... SYNTH 44  
 ..... SYNTH 45  
 ..... SYNTH 46  
 ..... SYNTH 47  
 ..... SYNTH 48  
 ..... SYNTH 49  
 ..... SYNTH 50  
 ..... SYNTH 51  
 ..... SYNTH 52  
 ..... SYNTH 53  
 ..... SYNTH 54  
 ..... SYNTH 55  
 ..... SYNTH 56  
 ..... SYNTH 57  
 ..... SYNTH 58

THIS SUBROUTINE WILL EDIT THE INPUT FILE OF KINETIC REACTIONS AND AUTOMATICALLY GENERATE SUBROUTINES REQUIRED FOR THE MOLECULAR KINETICS ANALYSIS AND ITS LINKAGE TO COUPLED ELECTRON KINETICS CALCULATIONS. SYMBOLIC REACTIONS ARE TRANSLATED INTO COMPUTER CODED EQUATIONS. SUBROUTINES SYNTHESIZED ARE: (1) RATES OF CHANGE DNI/DT OF POPULATION DENSITIES, (2) THE JACOBIAN D(DNI/DT)/D(NJ), AND (3) DEFINITION OF THE CORRESPONDENCE OF MOLECULAR STATES WHICH OCCUR IN THE ELECTRON-MOLECULE SCATTERING PROCESSES FOR THE E- KINETICS. INPUT CONSISTS OF AN ARBITRARILY LONG SEQUENCE OF CARD PAIRS OF THE FORM

1) A1 + A2 + A3 + ... + B1 + B2 + B3 + ...  
 2) KF, KR, KOMMNT (2E10.3,5X,5A10)

WHERE A1, A2, ..., B1, B2, ... ARE THE PHYSICAL NAMES OF THE REACTANT SPECIES (FOR ANY SPECIES NAME, THE FIRST \*NSIZE\* CHARACTERS ARE RECOGNIZED, AND OTHERS IGNORED). EACH SIDE OF THE REACTION MAY CONTAIN UP TO \*MAX\* SPECIES. THERE ARE NO RESTRICTIONS ON THE REACTION FORMAT (WHICH MAY INCLUDE EMBEDDED BLANKS) EXCEPT FOR THE FOLLOWING --

1) ELECTRONS ARE DENOTED BY EITHER E, E-, OR E(-).  
 2) HIGH ENERGY ELECTRONS (E-BEAM) ARE DENOTED BY HE-.  
 3) BUFFER GASES ARE DENOTED BY M.  
 4) NOISE PHOTONS ARE DENOTED BY HNU. IF HNU APPEARS ON THE LEFT HAND SIDE, IT IS REJECTED. ALL PROCESSES WHICH CONTRIBUTE NOISE TO THE BUILDUP OF STIMULATED EMISSION ARE TO BE DESCRIBED WITH HNU ON THE RIGHT HAND SIDE OF THE REACTION.  
 5) STIMULATED EMISSION OR ABSORPTION PROCESSES ARE DESCRIBED IN TERMS OF PHOTON NUMBER DENSITY, AND ARE RECOGNIZED BY THE APPEARANCE OF RAD.  
 6) THERE ARE NO SPECIAL RESTRICTIONS ON SPECIES NAMES EXCEPT THAT IONS MUST EXPLICITLY EXHIBIT THEIR CHARGE IN THEIR NAME (E.G. F-, KR2(+), AR(++) CO-, ETC.). IF AN ION IS POSITIVE, THE + SYMBOL MUST BE IMMEDIATELY FOLLOWED BY EITHER ANOTHER + OR ). IN ORDER TO AVOID CONFUSION WITH THE NORMAL USAGE OF + IN WRITING THE REACTION AS ABOVE.

IF KF = 0 (KR = 0), NO TRANSLATION OF THE FORWARD (REVERSE) REACTION TERM OCCURS (THUS, NULL OPERATIONS CONTAINING UNNECESSARY MU; TIPPLICATIONS BY ZERO) ARE ELIMINATED. RATE CONSTANTS USED FOR SYNTHESIS CAN BE CHANGED IN SUBSEQUENT EXECUTION, HOWEVER, SO A NONZERO VALUE SHOULD BE USED DURING SYNTHESIS FOR ANY REACTION PROCESS WHICH IS NOT TO BE PERMANENTLY NEGLECTED.

REACTIONS ARE SUBJECT TO SEVERAL TESTS TO DETERMINE WHETHER THEY SYNTH

C SHOULD BE RETAINED FOR CONSTRUCTING THE KINETIC EQUATIONS IN THE  
 C SUBROUTINE WHICH IS GENERATED. THESE INCLUDE THE FOLLOWING: SYNTH 59  
 C  
 C 1) HIGH ENERGY ELECTRONS MUST BALANCE ON LHS AND RHS. SYNTH 60  
 C  
 C 2) BUFFER GAS M MUST BALANCE ON BOTH SIDES OF EQUATION. SYNTH 61  
 C  
 C 3) CHARGE CONSERVATION MUST NOT BE VIOLATED. SYNTH 62  
 C  
 C 4) NO REVERSE PROCESS FOR SPONTANEOUS RADIATION ALLOWED. SYNTH 63  
 C  
 C 5) DUPLICATE REACTIONS (EVEN WRITTEN BACKWARDS) ARE IGNORED SYNTH 64  
 C  
 C 6) DETAIL BALANCE FOR BINARY MOLECULAR COLLISIONS ENFORCED. SYNTH 65  
 C  
 C 7) SECONDARY E- COLLISION PROCESSES MAY HAVE FIXED RATE CON- SYNTH 66  
 C STANTS, OR ZERO MAY BE ENTERED. IN WHICH CASE THEY WILL BE SYNTH 67  
 C AUTOMATICALLY LINKED BY DEFAULT TO AN E- KINETICS ANALY- SYNTH 68  
 C SIS. DURING SYNTHESIS, THE ELECTRON CROSS SECTION FILE SYNTH 69  
 C WILL BE SCANNED TO DETERMINE WHETHER DATA FOR THE PROCESS SYNTH 70  
 C IS AVAILABLE. AND IF NOT, AN INFORMATIVE DIAGNOSTIC ISSUED SYNTH 71  
 C  
 C 8) STIMULATED EMISSION (AND ABSORPTION) ARE DENOTED BY THE SYNTH 72  
 C SYNTAX: B + RAD = A + RAD. (IT IS ASSUMED THAT THE FOR- SYNTH 73  
 C FORWARD REACTION DENOTES EMISSION.) ABSORPTION PROCESSES CAN SYNTH 74  
 C BE ENTERED AS: X + RAD = Y. (ONLY THE FORWARD REACTION IS SYNTH 75  
 C RECOGNIZED. AND CORRESPONDS TO ABSORPTION.) SYNTH 76  
 C  
 C THERE ARE A VARIETY OF ERRORS RELATED TO EXCEEDED LIMITS, BAD SYNTH 77  
 C SYNTAX, BAD PHYSICS, ETC. WHICH ARE RECOGNIZED AND FLAGGED. BOTH SYNTH 78  
 C FATAL AND NON-FATAL WARNING CONDITIONS ARE GENERATED DURING SYNTH 79  
 C SYNTHESIS, AND ARE PROVIDED IN AN EDITED OUTPUT SUMMARY OF THE REACTION SYNTH 80  
 C SCHEME WHICH WAS PROCESSED. A CROSS-REFERENCE LISTING OF THE SYNTH 81  
 C OCCURRENCE OF SPECIES IN THE REACTION SCHEME IS ALSO GENERATED. SYNTH 82  
 C THE SYNTHESIZED SUBROUTINES ARE DOCUMENTED WITH COMMENT CARDS. SYNTH 83  
 C  
 C ..... SYNTH 84  
 C  
 C DIMENSION IMAGE(80), KAR(10), REFER(40), LINE(100), NBUFF(2), SYNTH 85  
 1 NHE(2), GAS(1), KINETIC(10), LABEL(5,2), LL(160), FORM(5), SYNTH 86  
 2 LEV1(1), LEV2(1), VSIG(2), TITLE(3), NAME(100), COMM(5), KODE(8), SYNTH 87  
 3 LDUM(10) SYNTH 88  
 C  
 C REAL KF, KR, KB SYNTH 89  
 C  
 C INTEGER GAS, SIGN, RATE(2), E, HNU, RAD, LHS, RHS, HE, TITLE, DATE SYNTH 90  
 C  
 C LOGICAL ELECT(2), RADIATE, REJECT, REVERSE, FORWARD, DETAIL, EXIT, SYNTH 91  
 1 SOURCE, BUFFER, PHOTO(2), OPTICAL, LASER, TEST SYNTH 92  
 C  
 C DATA E, HE, HNU, RAD, F, R, SKIP / 4HE(-), SHHE(-), 3HHNU, 3HRAD, SYNTH 93  
 1 7HFORWARD, 7HREVERSE, SH(1H1) / SYNTH 94  
 C  
 C DATA KB, EO, H, C / 1.38 E-23, 1.602 E-19, 6.625 E-34, 3.0 E 10 / SYNTH 95  
 C  
 C NTAPE = FILE FOR GENERATION OF SUBROUTINES RATES, LEVELS SYNTH 96  
 C MTAPE = FILE FOR GENERATION OF SUBROUTINE JACOB SYNTH 97

```

C      LTAPE = SCRATCH FILE FOR EDITING REACTION STRUCTURE           SYNTH    116
C      AT THE TERMINATION OF THE SYNTHESIS, NTAPE IS COPIED ONTO MTAPE.   SYNTH    117
C      NUATA = FILE CONTAINING UPDATED ELECTRON CROSS SECTION DATA    SYNTH    118
C      NSCRTCH = SCRATCH FILE FOR STORING REACTIONS AND RATE DATA     SYNTH    119
C
C      KB = KB/E0
C      EXIT = REJECT = .FALSE.
C      IN = NTYPE = 2
C      NAME(1) = GAS(1) = RAD
C      NAME(2) = GAS(2) = E
C      LASER = .FALSE.
C      NPHOTON = 1
C      READ (5,102) TITLE
C      CALL COVER (TITLE,2)
C      REWIND NSCRTCH
C      WRITE (NSCRTCH) TITLE, DATE
C
C      GENERATE SYMBOLIC RATE SUBROUTINE TO BE EXECUTED --
C
C      WRITE (NTAPE,400)
C      WRITE (NTAPE,105)
C      WRITE (NTAPE,406)
C      WRITE (NTAPE,105)
C      WRITE (NTAPE,605)
C      WRITE (NTAPE,105)
C      WRITE (NTAPE,460)
C      WRITE (NTAPE,406)
C      WRITE (NTAPE,105)
C      WRITE (MTAPE,510)
C      WRITE (MTAPE,105)
C      WRITE (MTAPE,406)
C      WRITE (MTAPE,105)
C      WRITE (MTAPE,505)
C      WRITE (MTAPE,105)
C      WRITE (MTAPE,460)
C      WRITE (MTAPE,406)
C      WRITE (MTAPE,105)
C      ENCODE (80,401,KODE)
C      WRITE (NTAPE,102) KODE
C      ENCODE (80,504,KODE)
C      WRITE (MTAPE,102) KODE
C      WRITE (NTAPE,402)
C      WRITE (NTAPE,105)
C      WRITE (NTAPE,104) KMAX, KMAX, KMAX, NKMAX, MAXGAS
C      WRITE (NTAPE,512)
C      WRITE (MTAPE,402)
C      WRITE (MTAPE,105)
C      WRITE (MTAPE,104) KMAX, KMAX, KMAX, NKMAX, MAXGAS
C      WRITE (MTAPE,512)
C      WRITE (NTAPE,421) KB, E0, H, C
C      WRITE (MTAPE,421) KB, E0, H, C
C
C      WRITE (NTAPE,403)
C      WRITE (NTAPE,407)
C      WRITE (NTAPE,105)
C      WRITE (NTAPE,415)
C      WRITE (NTAPE,105)

```

WRITE (NTAPE,422)	SYNTH	173
WRITE (NTAPE,406)	SYNTH	174
WRITE (NTAPE,105)	SYNTH	175
ENCODE (80+208,KODE)	SYNTH	176
WRITE (NTAPE,102) KODE	SYNTH	177
WRITE (NTAPE,105)	SYNTH	178
C	SYNTH	179
WRITE (MTAPE,403)	SYNTH	180
WRITE (MTAPE,507)	SYNTH	181
WRITE (MTAPE,105)	SYNTH	182
WRITE (MTAPE,422)	SYNTH	183
WRITE (MTAPE,406)	SYNTH	184
WRITE (MTAPE,105)	SYNTH	185
WRITE (MTAPE,102) KODE	SYNTH	186
WRITE (MTAPE,105)	SYNTH	187
C	SYNTH	188
C FORWARD DEFAULT FOR SECONDARY ELECTRON COLLISIONS --	SYNTH	189
M1 = 1	SYNTH	190
M2 = 2	SYNTH	191
INTEGER = 4H(I1)	SYNTH	192
INITIAL = 7HR = KF(	SYNTH	193
NTOT = 5H*NTOT	SYNTH	194
MULT = 4H*NO(	SYNTH	195
LAST = 6H - KR(	SYNTH	196
IF (NSIZE.GT.10) NSIZE = 10	SYNTH	197
MAX = 5	SYNTH	198
M0 = 5	SYNTH	199
M10 = 10*M0	SYNTH	200
M20 = 2*M10	SYNTH	201
LC = 0	SYNTH	202
NK = 0	SYNTH	203
INPUT = 0	SYNTH	204
K = 1	SYNTH	205
1 K = K-1	SYNTH	206
C READ HOLERITH STATEMENT OF REACTION NUMBER K --	SYNTH	207
10 K = K+1	SYNTH	208
READ (5,100) IMAGE	SYNTH	209
IF (EOF(5)) 5,76	SYNTH	210
76 READ (5,101) VSIG, COMM	SYNTH	211
DO 61 L = 1,M0	SYNTH	212
61 REFER(L) = COMM(L)	SYNTH	213
INPUT = INPUT+1	SYNTH	214
EXIT = EXIT.OR.REJECT	SYNTH	215
REWIND LTAPE	SYNTH	216
ENCODE (10+209,NUMBER) K	SYNTH	217
REJECT = DETAIL = .FALSE.	SYNTH	218
RATE(1) = RATE(2) = 1H	SYNTH	219
MESS = M0 + 1	SYNTH	220
IF (REFER(1).EQ.1H) MESS = 1	SYNTH	221
BUFFER = SOURCE = RADIATE = ELECT(1) = ELECT(2) = .FALSE.	SYNTH	222
PHOTO(1) = PHOTO(2) = .FALSE.	SYNTH	223
DO 9 L = 1,MAX	SYNTH	224
DO 9 M = 1,2	SYNTH	225
NBUFF(M) = NHE(M) = 0	SYNTH	226
9 LABEL(L,M) = 0	SYNTH	227
DO 22 L = 1,100	SYNTH	228
22 LINE(L) = 1H	SYNTH	229

```

DO 21 L = 1,160                               SYNTH   230
21 LL(L) = 1H                                 SYNTH   231
NKAR = 6                                       SYNTH   232
DECODE (7,100,INITIAL) (LL(NKAR+L), L = 1,7)  SYNTH   233
NKAR = NKAR+7                                SYNTH   234
K2 = K/10                                     SYNTH   235
K3 = K - 10*K2                                SYNTH   236
K1 = K2/10                                    SYNTH   237
K2 = K2 - 10*K1                                SYNTH   238
IF (K1,EQ.0) GO TO 11                           SYNTH   239
NKAR = NKAR+1                                SYNTH   240
ENCODE (10,INTEGER,LL(NKAR)) K1               SYNTH   241
11 IF (K1*K2,EQ.0) GO TO 29                  SYNTH   242
NKAR = NKAR+1                                SYNTH   243
ENCODE (10,INTEGER,LL(NKAR)) K2               SYNTH   244
29 NKAR = NKAR+1                                SYNTH   245
ENCODE (10,INTEGER,LL(NKAR)) K3               SYNTH   246
NKAR = NKAR+1                                SYNTH   247
LL(NKAR) = 1H                                  SYNTH   248
M = 1                                         SYNTH   249
N0 = NTYPE                                    SYNTH   250
NE = NCH = 0                                   SYNTH   251
NM = 1                                         SYNTH   252
LP = I = J = N = 0                             SYNTH   253
C SCAN THE 80 BCD CHARACTERS TO DETERMINE SPECIES --
2 IF (I.EQ.80) GO TO 4                         SYNTH   254
I = I+1                                       SYNTH   255
C IMBEDDED BLANKS ARE IGNORED.
IF (IMAGE(I).EQ.1H) GO TO 2                  SYNTH   256
IF (IMAGE(I).NE.1H) GO TO 3                  SYNTH   257
MM = 1                                         SYNTH   258
GO TO 4                                       SYNTH   259
C
3 IF (IMAGE(I).NE.1H) GO TO 6                SYNTH   260
IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H+) GO TO 6
    MM = 0                                       SYNTH   261
    GO TO 4                                     SYNTH   262
C
C CONTINUE TO ADD NONBLANK BCD CHARACTERS TO THE GAS NAME. NAMES
C ARE TRUNCATED TO IGNORE ALL BUT THE FIRST NSIZE CHARACTERS.
C
6 IF (J.EQ.NSIZE) GO TO 2                     SYNTH   263
J = J+1                                       SYNTH   264
LP = LP+1                                     SYNTH   265
LINE(LP) = KAR(J) = IMAGE(I)                 SYNTH   266
IF (KAR(J).EQ.1H) NCH = NCH + NN            SYNTH   267
IF (KAR(J).EQ.1H-) NCH = NCH - NN          SYNTH   268
GO TO 2                                       SYNTH   269
C
C DUMP THE CONTENTS OF THE NAME AFTER A DELIMITER (+, *, OR COL. 80)
C HAS BEEN ENCOUNTERED --
C
4 IF (J.EQ.0) GO TO 60                        SYNTH   270
IF (M.GT.2) GO TO 60                          SYNTH   271
IF (I.EQ.80.AND.M.NE.2) GO TO 60            SYNTH   272
IF (N.LT.MAX) GO TO 38                        SYNTH   273
ENCODE (M10,214,REFER(MESS)) MAX           SYNTH   274
                                                SYNTH   275
                                                SYNTH   276
                                                SYNTH   277
                                                SYNTH   278
                                                SYNTH   279
                                                SYNTH   280
                                                SYNTH   281
                                                SYNTH   282
                                                SYNTH   283
                                                SYNTH   284
                                                SYNTH   285
                                                SYNTH   286

```

	MESS = MESS + M0	SYNTM	287
60	REJECT = .TRUE.	SYNTM	288
	ENCODE (M10,216,REFER(MESS))	SYNTM	289
	MESS = MESS+M0	SYNTM	290
	LB = 0	SYNTM	291
	DO 53 L = 1,80	SYNTM	292
	IF (IMAGE(L).EQ.1H) GO TO 53	SYNTM	293
	LB = LB+1	SYNTM	294
	IMAGE(LB) = IMAGE(L)	SYNTM	295
53	CONTINUE	SYNTM	296
	LB = LB+1	SYNTM	297
	IF (LB.GT.80) LB = 80	SYNTM	298
	DO 55 L = LB,80	SYNTM	299
55	IMAGE(L) = 1H	SYNTM	300
	NUMBER = 1H	SYNTM	301
C	ENCODE UNRECOGNIZABLE REACTION --	SYNTM	302
	ENCODE (100,100,KINETIC) (IMAGE(L), L = 1,80)	SYNTM	303
	GO TO 26	SYNTM	304
38	ENCODE (10,100,NGAS) (KAR(L), L = 1,J)	SYNTM	305
	IF (NGAS.EQ.1HE) NCH = NCH - NN	SYNTM	306
	IF (NGAS.EQ.1HE.OR.NGAS.EQ.2HE-) NGAS = E	SYNTM	307
	IF (NGAS.EQ.3HHE-) NGAS = HE	SYNTM	308
	IF (NGAS.EQ.HE) NCH = NCH + NN	SYNTM	309
	J = 0	SYNTM	310
	IF (NGAS.EQ.1HM) GO TO 8	SYNTM	311
	IF (NGAS.EQ.HE) GO TO 8	SYNTM	312
	IF (NGAS.NE.HNU) GO TO 30	SYNTM	313
	IF (M.EQ.1) GO TO 67	SYNTM	314
	IF (RADIADE) GO TO 67	SYNTM	315
	IF (PHOTO(1).OR.PHOTO(2)) GO TO 67	SYNTM	316
	RADIATE = .TRUE.	SYNTM	317
	VSIG(2) = 0.	SYNTM	318
	GO TO 16	SYNTM	319
30	IF (INGAS.NE.RAD) GO TO 45	SYNTM	320
	IF (PHOTO(M)) GO TO 67	SYNTM	321
	PHOTO(M) = .TRUE.	SYNTM	322
	IF (RADIADE.AND.(PHOTO(1).OR.PHOTO(2))) GO TO 67	SYNTM	323
	GO TO 45	SYNTM	324
67	REJECT = .TRUE.	SYNTM	325
	ENCODE (M10,227,REFER(MESS))	SYNTM	326
	MESS = MESS + M0	SYNTM	327
	GO TO 16	SYNTM	328
45	N = N+1	SYNTM	329
	IF (INGAS.EQ.E) NE = NE+1	SYNTM	330
	DO 15 L = 1,NO	SYNTM	331
	IF (GAS(L).NE.NGAS) GO TO 15	SYNTM	332
	LABEL(N,M) = L	SYNTM	333
	GO TO 8	SYNTM	334
15	CONTINUE	SYNTM	335
	IF (NO.NE.MAXGAS) GO TO 39	SYNTM	336
	IF (REJECT) GO TO 16	SYNTM	337
	REJECT = .TRUE.	SYNTM	338
	ENCODE (M20,213,REFER(MESS)) MAXGAS	SYNTM	339
	MESS = MESS + 2*M0	SYNTM	340
	GO TO 16	SYNTM	341
39	NO = NO + 1	SYNTM	342
	GAS(NO) = NGAS	SYNTM	343

```

LABEL(N,M) = NO                               SYNTH    344
8 NEW = NKAR - 80*(NKAR/80) + 7               SYNTH    345
IF (NEW.LE.72) GO TO 17                      SYNTH    346
NEW = NEW+8                     SYNTH    347
NKAR = 80*(NEW/80) + 6                     SYNTH    348
LL(NKAR) = 1H                                SYNTH    349
17 DECODE (4,100,MULT) (LL(NKAR+L), L = 1,4)  SYNTH    350
IF (NGAS.NE.1HM) GO TO 70                   SYNTH    351
NBUFF(M) = NBUFF(M) + 1                     SYNTH    352
BUFFER = .TRUE.                            SYNTH    353
DECODE (5,100,NTOT) (LL(NKAR+L), L = 1,5)  SYNTH    354
NKAR = NKAR+5                           SYNTH    355
GO TO 16                                 SYNTH    356
70 IF (NGAS.NE.NE) GO TO 20                 SYNTH    357
NHE(M) = NHE(M) + 1                     SYNTH    358
SOURCE = .TRUE.                         SYNTH    359
GO TO 16                                 SYNTH    360
20 NKAR = NKAR+4                     SYNTH    361
N1 = LABEL(N,M)/10                    SYNTH    362
N2 = LABEL(N,M) - 10*N1              SYNTH    363
IF (N1.EQ.0) GO TO 13                SYNTH    364
NKAR = NKAR+1                     SYNTH    365
ENCODE (10,INTEGER,LL(NKAR)) N1      SYNTH    366
13 NKAR = NKAR+1                     SYNTH    367
ENCODE (10,INTEGER,LL(NKAR)) N2      SYNTH    368
NKAR = NKAR+1                     SYNTH    369
LL(NKAR) = 1H                          SYNTH    370
16 IF (I.EQ.80) GO TO 27             SYNTH    371
C
C   DELIMITERS (+, =) ENCODED INTO HOLERITH LINE TO DEFINE REACTION --
C
LP = LP+1                                SYNTH    372
LINE(LP) = 1H                             SYNTH    373
LP = LP+1                                SYNTH    374
LINE(LP) = IMAGE(I)                      SYNTH    375
LP = LP+1                                SYNTH    376
LINE(LP) = 1H                             SYNTH    377
M = M + MM                            SYNTH    378
IF (MM.EQ.0) GO TO 2                  SYNTH    379
NL = NE                                SYNTH    380
IF (.NOT.SOURCE) GO TO 62            SYNTH    381
IF (NHE(1).EQ.1) GO TO 62            SYNTH    382
REJECT = .TRUE.                         SYNTH    383
ENCODE (M10,226,REFER(MESS))        SYNTH    384
MESS = MESS + MO                      SYNTH    385
62 IF (.NOT.(PHOTO(1).AND.NE.NE.0)) GO TO 64
REJECT = .TRUE.                         SYNTH    386
ENCODE (M10,227,REFER(MESS))        SYNTH    387
MESS = MESS + MO                      SYNTH    388
64 ELECT(1) = .NOT.(PHOTO(1).OR.SOURCE.OR.BUFFER)
DETAIL = .NOT.(PHOTO(1).OR.SOURCE)
DETAIL = DETAIL.AND.(NE.EQ.0).AND.(N+NBUFF(1).EQ.2)
ELECT(1) = (NE.EQ.1).AND.(N.EQ.2).AND.ELECT(1)
IF (VSIG(1).NE.0.) ENCODE (10,103,RATE(1)) VSIG(1)
FORWARD = (VSIG(1).NE.0.).AND.(NE.LE.1)
FORWARD = FORWARD.OR.ELECT(1)
LEFT = NKAR                            SYNTH    390
SYNTH    391
SYNTH    392
SYNTH    393
SYNTH    394
SYNTH    395
SYNTH    396
SYNTH    397
SYNTH    398
SYNTH    399
SYNTH    400

```

```

NN = - 1           SYNTH  401
NE = N = 0         SYNTH  402
IF (.NOT.FORWARD) NKAR = 9  SYNTH  403
C
C   CONSTRUCT REVERSE REACTION TERM --
C
      IF (REJECT) GO TO 2           SYNTH  404
      DECODE (6+100, LAST) (LL(NKAR+L), L = 1,6)  SYNTH  405
      NKAR = NKAR+6                SYNTH  406
      IF (K1.EQ.0) GO TO 24        SYNTH  407
      NKAR = NKAR+1                SYNTH  408
      ENCODE (10+INTEGER,LL(NKAR)) K1  SYNTH  409
24    IF (K1+K2.EQ.0) GO TO 28    SYNTH  410
      NKAR = NKAR+1                SYNTH  411
      ENCODE (10+INTEGER,LL(NKAR)) K2  SYNTH  412
28    NKAR = NKAR+1                SYNTH  413
      ENCODE (10+INTEGER,LL(NKAR)) K3  SYNTH  414
      NKAR = NKAR+1                SYNTH  415
      LL(NKAR) = 1H                SYNTH  416
      GO TO 2                      SYNTH  417
C
27    IF (LP.GT.100) LP = 100      SYNTH  418
      ENCODE (100,100,KINETIC) (LINE(L), L = 1,LP)  SYNTH  419
      OPTICAL = PHOTO(1).OR.PHOTO(2)  SYNTH  420
C
      IF (NBUFF(1).EQ.NBUFF(2).AND.NBUFF(1).LE.1) GO TO 56  SYNTH  421
      ENCODE (M10,224,REFER(MESS))  SYNTH  422
      MESS = MESS + M0              SYNTH  423
      ENCODE (M10,216,REFER(MESS))  SYNTH  424
      MESS = MESS + M0              SYNTH  425
      REJECT = .TRUE.               SYNTH  426
      GO TO 23                     SYNTH  427
C
56    IF (NHE(1).EQ.NHE(2).AND.NHE(1).LE.1) GO TO 57  SYNTH  428
      ENCODE (M10,225,REFER(MESS))  SYNTH  429
      MESS = MESS + M0              SYNTH  430
      ENCODE (M10,216,REFER(MESS))  SYNTH  431
      MESS = MESS + M0              SYNTH  432
      REJECT = .TRUE.               SYNTH  433
      GO TO 23                     SYNTH  434
C
57    IF (.NOT.((SOURCE.AND.(RADIATE.OR.OPTICAL)).OR.(.NOT.PHOTO(1).  SYNTH  435
     1 AND.PHOTO(2))) GO TO 66  SYNTH  436
      ENCODE (M10,227,REFER(MESS))  SYNTH  437
      MESS = MESS + M0              SYNTH  438
      REJECT = .TRUE.               SYNTH  439
      GO TO 23                     SYNTH  440
C
C   GENERATE CHECKSUM IDENTIFIER --
C
A6    IF (REJECT) GO TO 23       SYNTH  441
      K1 = K2 = NBUFF(1)*(MAXGAS+1) + NHE(1)*(MAXGAS+2)  SYNTH  442
      K1SQ = K2SQ = K1*K1 - 2*NBUFF(1)*(MAXGAS+1)*NHE(1)*(MAXGAS+2)  SYNTH  443
      DO 31 L = 1,5                SYNTH  444
      K1 = K1 + LABEL(L+1)          SYNTH  445
      K2 = K2 + LABEL(L+2)          SYNTH  446
      K1SQ = K1SQ + LABEL(L+1)*LABEL(L+1)  SYNTH  447
31    K2SQ = K2SQ + LABEL(L+2)*LABEL(L+2)  SYNTH  448
                                         SYNTH  449
                                         SYNTH  450
                                         SYNTH  451
                                         SYNTH  452
                                         SYNTH  453
                                         SYNTH  454
                                         SYNTH  455
                                         SYNTH  456
                                         SYNTH  457

```

```

ENCODE (10+107,LHS) K1, K1SQ      SYNTH 458
ENCODE (10+107,RHS) K2, K2SQ      SYNTH 459
C
C CHECK FOR DUPLICATION --
C
IF (K.EQ.1) GO TO 23              SYNTH 460
K1 = K-1                          SYNTH 461
DO 25 L = 1,K1                   SYNTH 462
READ (LTAPE) L1, L2               SYNTH 463
IF (REJECT) GO TO 23              SYNTH 464
REJECT = LHS.EQ.L1.AND.RHS.EQ.L2  SYNTH 465
REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1  SYNTH 466
IF (RADIAE) REVERSE = .FALSE.    SYNTH 467
REJECT = REJECT.OR.REVERSE       SYNTH 468
IF (.NOT.REJECT) GO TO 25        SYNTH 469
ENCODE (M10,210,REFER(MESS)) L   SYNTH 470
IF (REVERSE) ENCODE (M10,211,REFER(MESS)) L  SYNTH 471
MESS = MESS + M0                  SYNTH 472
25 CUNTINUE                      SYNTH 473
C
23 ELECT(2) = (NE.EQ.1).AND.(N.EQ.2)  SYNTH 474
NR = NE                           SYNTH 475
REVERSE = .NOT.(RADIAE.OR.(PHOTO(1).AND.VSIG(1).EQ.0.))  SYNTH 476
IF (.NOT.REVERSE) VSIG(2) = 0.      SYNTH 477
IF (PHOTO(1).AND..NOT.PHOTO(2)) VSIG(2) = 0.  SYNTH 478
ELECT(2) = .NOT.(BUFFER.OR.RADIAE.OR.OPTICAL.  SYNTH 479
1          OR.SOURCE).AND.ELECT(2)  SYNTH 480
IF (VSIG(2).NE.0.) ENCODE (10,103,RATE(2)) VSIG(2)  SYNTH 481
IF (VSIG(1).NE.0.) ELECT(1) = ELECT(2) = .FALSE.  SYNTH 482
IF (ELECT(1).AND.ELECT(2)) VSIG(1) = VSIG(2) = 0.  SYNTH 483
ELECT(1) = ELECT(1).AND.VSIG(1).EQ.0.  SYNTH 484
ELECT(2) = ELECT(2).AND.VSIG(2).EQ.0.  SYNTH 485
DETAIL = (VSIG(1).NE.0.).AND.(VSIG(2).EQ.0.).AND.(NE.EQ.0.).AND.  SYNTH 486
1          (N+NBUFF(2).EQ.2).AND.DETAIL.AND.REVERSE  SYNTH 487
REVERSE = REVERSE.AND.(DETAIL.OR.(VSIG(2).NE.0.).OR.ELECT(2))  SYNTH 488
REJECT = .NOT.(FORWARD.OR.REVERSE).OR.REJECT  SYNTH 489
REJECT = REJECT.OR.(K.GT.KMAX)  SYNTH 490
IF (K.LE.KMAX) GO TO 43  SYNTH 491
ENCODE (M10,215,REFER(MESS)) KMAX  SYNTH 492
MESS = MESS + M0                  SYNTH 493
C
C TEST FOR CHARGE CONSERVATION --
C
43 IF (INCH.EQ.0) GO TO 85  SYNTH 494
REJECT = .TRUE.  SYNTH 495
ENCODE (M10,204,REFER(MESS))  SYNTH 496
MESS = MESS + M0  SYNTH 497
85 IF (REJECT) GO TO 51  SYNTH 498
IF (.NOT.(ELECT(1).OR.ELECT(2))) GO TO 51  SYNTH 499
IF (NK.LT.NKMAX) GO TO 48  SYNTH 500
REJECT = .TRUE.  SYNTH 501
ENCODE (M20,218,REFER(MESS)) NKMAX  SYNTH 502
MESS = MESS + 2*M0  SYNTH 503
GO TO 51  SYNTH 504
C
C SEARCH FILE OF ELECTRON CROSS SECTIONS --
C

```

48 CALL DEKODE (NAME, IMAGE, L1, L2, LDUM, KAR, 10, IN, 60)	SYNTH	515	
REWIND NDATA	SYNTH	516	
74 READ (IN DATA,100) IMAGE	SYNTH	517	
IF (EOF(IN DATA)) 79,72	SYNTH	518	
72 CALL DEKODE (NAME, IMAGE, LL1, LL2, LDUM, KAR, 10, IN, 60)	SYNTH	519	
IF (L1.NE.LL1.OR.L2.NE.LL2) GO TO 77	SYNTH	520	
M1 = 1	SYNTH	521	
M2 = 2	SYNTH	522	
GO TO 52	SYNTH	523	
C	SYNTH	524	
77 IF (L1.NE.LL2.OR.L2.NE.LL1) GO TO 78	SYNTH	525	
M1 = 2	SYNTH	526	
M2 = 1	SYNTH	527	
GO TO 52	SYNTH	528	
C	SYNTH	529	
C C EXHAUST NUMERICAL DATA FOR THIS PROCESS --	SYNTH	530	
C	SYNTH	531	
78 READ (IN DATA,100)	SYNTH	532	
73 READ (IN DATA,102) LETTERS	SYNTH	533	
IF (LETTERS.EQ.1H) GO TO 74	SYNTH	534	
READ (IN DATA,102)	SYNTH	535	
GO TO 73	SYNTH	536	
C	SYNTH	537	
79 ENCODE (M10,219,REFER(MESS))	SYNTH	538	
MESS = MESS + M0	SYNTH	539	
GO TO 52	SYNTH	540	
C	IF DESIRED, LACK OF E- CROSS SECTIONS CAN BE MADE SUFFICIENT FOR	SYNTH	541
C	REJECTION OF THE REACTION BY THE REMOVAL OF THE ABOVE CARD, AND	SYNTH	542
C	THE REMOVAL OF *C* ON THE FOLLOWING THREE CARDS --	SYNTH	543
C	ELECT(1) = ELECT(2) = .FALSE.	SYNTH	544
C	REJECT = .TRUE.	SYNTH	545
C	GO TO 51	SYNTH	546
52 READ (IN DATA,102) (REFER(L), L = 1,M0)	SYNTH	547	
NK = NK+1	SYNTH	548	
C	KULL(NK) = K	SYNTH	549
IF (.NOT.ELECT(1)) GO TO 54	SYNTH	550	
RATE(1) = 10H(COMPUTED)	SYNTH	551	
LEV1(NK) = K1-2	SYNTH	552	
ENCODE (M10,223,REFER(MESS)), F	SYNTH	553	
MESS = MESS + M0	SYNTH	554	
54 IF (.NOT.ELECT(2)) GO TO 51	SYNTH	555	
RATE(2) = 10H(COMPUTED)	SYNTH	556	
LEV2(NK) = K2-2	SYNTH	557	
ENCODE (M10,223,REFER(MESS)), R	SYNTH	558	
MESS = MESS + M0	SYNTH	559	
C	SYNTH	560	
51 IF (NL.EQ.NR) GO TO 98	SYNTH	561	
NE = NR-NL	SYNTH	562	
IF (ELECT(1).AND.NE.LT.0) GO TO 98	SYNTH	563	
IF (SOURCE) ENCODE (M10,312+REFER(MESS))	SYNTH	564	
IF (.NOT.SOURCE) ENCODE (M10,313+REFER(MESS))	SYNTH	565	
MESS = MESS + M0	SYNTH	566	
98 IF (REJECT) NUMBER = 1H	SYNTH	567	
IF (FORWARD) GO TO 46	SYNTH	568	
ENCODE (M10,220,REFER(MESS))	SYNTH	569	
MESS = MESS + M0	SYNTH	570	
46 IF (REVERSE) GO TO 47	SYNTH	571	

```

IF (.NOT.FORWARD) GO TO 49           SYNTH 572
ENCODE (M10,222,REFER(MESS))        SYNTH 573
MESS = MESS + M0                    SYNTH 574
IF (.NOT.RADIATE) GO TO 26          SYNTH 575
ENCODE (M10,221,REFER(MESS))        SYNTH 576
MESS = MESS + M0                    SYNTH 577
GO TO 26                           SYNTH 578
49 MESS = MESS-M0                  SYNTH 579
ENCODE (M10,212,REFER(MESS))        SYNTH 580
MESS = MESS + M0                    SYNTH 581
IF (.NOT.RADIATE) GO TO 47          SYNTH 582
ENCODE (M10,221,REFER(MESS))        SYNTH 583
MESS = MESS + M0                    SYNTH 584
GO TO 26                           SYNTH 585
47 IF (.NOT.DETAIL) GO TO 26        SYNTH 586
RATE(2) = 10HX E(-E/KT)             SYNTH 587
ENCODE (M10,203,REFER(MESS))        SYNTH 588
MESS = MESS + M0                    SYNTH 589
26 IF (LC.GT.0) GO TO 7            SYNTH 590
IF (K.GT.1) WRITE (6,202)           SYNTH 591
IF (K.GT.1) WRITE (6,470) DATE      SYNTH 592
WRITE (6,200)                      SYNTH 593
7 IF (MESS.EQ.1) MESS = M0+1        SYNTH 594
REFER(MESS) = 1H                   SYNTH 595
LC = LC+1*(MESS-1)/M0              SYNTH 596
IF (LC.GT.40) LC = 0               SYNTH 597
IF (REJECT.AND.VSIG(1).EQ.0.) RATE(1) = 1H
IF (REJECT.AND.VSIG(2).EQ.0.) RATE(2) = 1H
C THE REACTION IS STORED IN LINE(L). WITH BLANKS NEATLY EMBEDDED --
C WRITE (6,201) NUMBER, (LINE(L), L = 1+45), RATE, (REFER(L), L =
1+MESS)
IF (REJECT) GO TO 1                SYNTH 601
NTYPE = NO                         SYNTH 602
KTYPE = K                          SYNTH 603
IF (ELECT(1)) ENCODE (10,110,RATE(1)) M1,NK
IF (ELECT(2)) ENCODE (10,110,RATE(2)) M2,NK
KF = KR = UNDEF                   SYNTH 604
IF (FORWARD.AND.VSIG(1).NE.0.) KF = VSIG(1)
IF (REVERSE.AND.VSIG(2).NE.0.) KR = VSIG(2)
C WRITE (LTAPE) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, (REFER(L),
1 L = 1,5)
PHOTO(2) = PHOTO(2).AND.PHOTO(1)   SYNTH 605
C GENERATE RATE EXPRESSIONS --
C
IF (.NOT.REVERSE) NKAR = LEFT     SYNTH 606
WRITE (INTAPE,404) K, (KINETIC(L), L = 1+6)
WRITE (INTAPE,105)                 SYNTH 607
WRITE (INTAPE,404) K, (KINETIC(L), L = 1+6)
WRITE (INTAPE,105)                 SYNTH 608
IF (ELECT(1)) WRITE (INTAPE,405) K, M1, NK
IF (ELECT(1)) WRITE (INTAPE,405) K, M1, NK
IF (ELECT(2)) WRITE (INTAPE,409) K, M2, NK
IF (ELECT(2)) WRITE (INTAPE,409) K, M2, NK
IF (ELECT(1).OR.ELECT(2)) WRITE (INTAPE,105)
IF (ELECT(1).OR.ELECT(2)) WRITE (INTAPE,105)
IF (SOURCE) GO TO 65              SYNTH 609
SYNTH 610
SYNTH 611
SYNTH 612
SYNTH 613
SYNTH 614
SYNTH 615
SYNTH 616
SYNTH 617
SYNTH 618
SYNTH 619
SYNTH 620
SYNTH 621
SYNTH 622
SYNTH 623
SYNTH 624
SYNTH 625
SYNTH 626
SYNTH 627
SYNTH 628

```

C	IF (.NOT.DETAIL) GO TO 64	SYNT	629
C	WRITE EXPRESSIONS FOR DETAILED BALANCE RELATION --	SYNT	630
	WRITE (NTAPE,408)	SYNT	631
	WRITE (NTAPE,105)	SYNT	632
	WRITE (MTAPE,408)	SYNT	633
	WRITE (MTAPE,105)	SYNT	634
	GO TO 65	SYNT	635
44	IF (.NOT.PHOTO(1)) GO TO 35	SYNT	636
	IF (PHOTO(2)) WRITE (NTAPE,412) NPHOTON	SYNT	637
	IF (.NOT.PHOTO(2)) WRITE (NTAPE,413) NPHOTON	SYNT	638
	IF (PHOTO(2)) WRITE (NTAPE,412) NPHOTON	SYNT	639
	IF (.NOT.PHOTO(2)) WRITE (NTAPE,413) NPHOTON	SYNT	640
	IF (.NOT.PHOTO(2)) GO TO 35	SYNT	641
	IF (LASER) GO TO 35	SYNT	642
	LASER = .TRUE.	SYNT	643
65	DO 41 M = 1,2	SYNT	644
	ENCODE (10,106,LINE) M	SYNT	645
	DO 42 L = 1,5	SYNT	646
	J = LABEL(L,M)	SYNT	647
	IF (J.EQ.0) GO TO 92	SYNT	648
	L1 = L+1	SYNT	649
	ENCODE (10,108,LINE(L1)) J	SYNT	650
42	CONTINUE	SYNT	651
92	IF (DETAIL) WRITE (MTAPE,109) (LINE(N), N = 1:L)	SYNT	652
41	WRITE (NTAPE,109) (LINE(N), N = 1:L)	SYNT	653
	IF (SOURCE) GO TO 35	SYNT	654
	IF (DETAIL) GO TO 58	SYNT	655
	WRITE (NTAPE,417)	SYNT	656
	WRITE (NTAPE,105)	SYNT	657
	GO TO 35	SYNT	658
58	WRITE (NTAPE,410) K+K	SYNT	659
	WRITE (NTAPE,105)	SYNT	660
	WRITE (MTAPE,410) K+K	SYNT	661
	WRITE (MTAPE,105)	SYNT	662
35	WRITE (NTAPE,100) (LL(L), L = 1:NKAR)	SYNT	663
	IF (SOURCE) WRITE (NTAPE,311)	SYNT	664
	IF (PHOTO(1)) WRITE (NTAPE,416)	SYNT	665
	WRITE (NTAPE,105)	SYNT	666
	WRITE (NTAPE,217) K	SYNT	667
	DO 18 I = 1,NTYPE	SYNT	668
	N1 = N2 = 0	SYNT	669
	DO 14 L = 1,MAX	SYNT	670
	IF (LABEL(L,1).EQ.I) N1 = N1+1	SYNT	671
14	IF (LABEL(L,2).EQ.I) N2 = N2+1	SYNT	672
	N = N2-N1	SYNT	673
C	IT IS UNDERSTOOD THAT, IF RAD APPEARS ON BOTH SIDES OF THE EQUA-	SYNT	674
C	TION, A NET INCREASE IN RADIATION RESULTS.)	SYNT	675
C	IF (I.EQ.NPHOTON.AND.N1.EQ.1.AND.N2.EQ.1) N = 1	SYNT	676
	IF (N.EQ.0) GO TO 18	SYNT	677
	IF (N.GT.0) SIGN = 1H	SYNT	678
	IF (N.LT.0) SIGN = 1H-	SYNT	679
	ENCODE (10,205,NSIGN) SIGN	SYNT	680
	N = IABS(N)	SYNT	681
	IND = 1	SYNT	682
	IF (I.GT.9) IND = 2	SYNT	683
		SYNT	684
		SYNT	685

```

IF (I.GT.99) IND = 3           SYNTH   686
ENCODE (50,500,FORM) IND, IND, MAX  SYNTH   687
WRITE (NTAPE,FORM) I, I, (NSIGN, J = 1,N)  SYNTH   688
18 CONTINUE  SYNTH   689
C           SYNTH   690
C           IF (.NOT.SOURCE) GO TO 95  SYNTH   691
C           WRITE (NTAPE,305)  SYNTH   692
C           DETERMINE IF SOURCE TERM CORRESPONDS TO IONIZATION --  SYNTH   693
C           DO 94 L = 1,MAX  SYNTH   694
C           J = LAREL(L,2)  SYNTH   695
C           IF (J.EQ.0) GO TO 97  SYNTH   696
C           IF (J.EQ.2) GO TO 96  SYNTH   697
C           94 CONTINUE  SYNTH   698
C           SOURCE TERM WAS IONIZATION --  SYNTH   699
C           96 WRITE (NTAPE,310)  SYNTH   700
C           97 WRITE (NTAPE,105)  SYNTH   701
C           WRITE (NTAPE,309)  SYNTH   702
C           WRITE (NTAPE,304)  SYNTH   703
C           GO TO 90  SYNTH   704
C           95 NE = NR-NL  SYNTH   705
C           IF (ELECT(1).OR.ELECT(2).OR.(NE.EQ.0)) GO TO 90  SYNTH   706
C           WRITE (NTAPE,105)  SYNTH   707
C           IF (NE.GT.0) SIGN = 1H+  SYNTH   708
C           IF (NE.LT.0) SIGN = 1H-  SYNTH   709
C           IF (NE.GT.0) WRITE (NTAPE,424)  SYNTH   710
C           IF (NE.LT.0) WRITE (NTAPE,425)  SYNTH   711
C           WRITE (NTAPE,105)  SYNTH   712
C           NE = IABS(NE)  SYNTH   713
C           WRITE (NTAPE,426) (SIGN, L = 1,NE)  SYNTH   714
C           90 IF (RADIATE) WRITE (NTAPE,418)  SYNTH   715
C           C .....*****.....  SYNTH   716
C           C GENERATE JACOBIAN --  SYNTH   717
C           C THE RATE FOR REACTION K, ENCODED INTO THE VECTOR LL ABOVE, IS R.  SYNTH   718
C           C IF SPECIES I OCCURS NR TIMES ON THE RIGHT, NL TIMES ON THE LEFT,  SYNTH   719
C           C OF REACTION K, THEN THE CONTRIBUTION OF REACTION K TO DNDOT(I) IS  SYNTH   720
C           C NI*R, WHERE N = (NR - NL). THE FOLLOWING LOOP CALCULATES THE  SYNTH   721
C           C DERIVATIVE S = DS/DN(J) FOR EVERY SPECIES J WHICH OCCURS. FOR  SYNTH   722
C           C EVERY SPECIES I, REACTION K MAKES A CONTRIBUTION OF NI*S TO THE  SYNTH   723
C           C JACOBIAN PHI(I,J) = DNDOT(I)/DN(J).  SYNTH   724
C           C
C           DO A9 J = 1,NTYPE  SYNTH   725
C           IP = JP = 0  SYNTH   726
C           LINE(I) = 1H(  SYNTH   727
C           NKAR = 1  SYNTH   728
C           M = JN = NJ = 0  SYNTH   729
A3 M = M+1  SYNTH   730
IF (M.GT.2) GO TO 82  SYNTH   731
IF (M.EQ.2) GO TO 88  SYNTH   732
IF (.NOT.FURWARD) GO TO 83  SYNTH   733
NSIGN = 1H  SYNTH   734
KJ = 1HF  SYNTH   735
GO TO 84  SYNTH   736
AA IF (.NOT.REVERSE) GO TO 82  SYNTH   737
NSIGN = 1H-  SYNTH   738

```

C	KJ = 1HR	SYNTH	743
	NJ IS THE NUMBER OF TIMES SPECIES J APPEARS (ON LHS OR RHS) --	SYNTH	744
84	NJ = 0	SYNTH	745
	DO 63 L = 1,MAX	SYNTH	746
	JL = LABEL(L,M)	SYNTH	747
	IF (JL,NE,J) GO TO 63	SYNTH	748
	NJ = NJ+1	SYNTH	749
	LJ = L	SYNTH	750
63	CONTINUE	SYNTH	751
	JN = JN + NJ	SYNTH	752
	IF (INJ,EO,0) GO TO 83	SYNTH	753
	IP = IP+1	SYNTH	754
	NKAR = NKAR+1	SYNTH	755
	LINE(NKAR) = NSIGN	SYNTH	756
	NKAR = NKAR+1	SYNTH	757
	ENCODE (10,306,LINE(NKAR)) KJ, K	SYNTH	758
	IF (NJ,GT,1) ENCODE (10,307,LINE(NKAR)) NJ, KJ, K	SYNTH	759
	DO 91 L = 1,MAX	SYNTH	760
	IF (L,EO,LJ) GO TO 91	SYNTH	761
	JL = LABEL(L,M)	SYNTH	762
	IF (JL,EO,0) GO TO 83	SYNTH	763
	NKAR = NKAR+1	SYNTH	764
	ENCODE (10,308,LINE(NKAR)) JL	SYNTH	765
91	CONTINUE	SYNTH	766
	GO TO 83	SYNTH	767
82	IF (JN,EO,0) GO TO 89	SYNTH	768
	LKAR = NKAR = NKAR+1	SYNTH	769
	LINE(NKAR) = 1H	SYNTH	770
	IF (INHE(1),EO,0) GO TO 68	SYNTH	771
	NKAR = NKAR+1	SYNTH	772
	LINE(NKAR) = 9H*IBEAM/E0	SYNTH	773
	JP = IP	SYNTH	774
68	IF (NBUFF(1),EQ,0) GO TO 69	SYNTH	775
	NKAR = NKAR+1	SYNTH	776
	LINE(NKAR) = NTOT	SYNTH	777
	JP = IP	SYNTH	778
69	IF (JP,NE,2) LINE(1) = LINE(LKAR) = 1H	SYNTH	779
	LKAR = 0	SYNTH	780
	DO 81 L = 1,NKAR	SYNTH	781
	DECODE (10,100,LINE(L)) (KAR(L)), L1 = 1,10	SYNTH	782
	DO 93 L2 = 1,10	SYNTH	783
	IF (KAR(L2),EO,1H) GO TO 93	SYNTH	784
	LKAR = LKAR+1	SYNTH	785
	LL(LKAR) = KAR(L2)	SYNTH	786
93	CONTINUE	SYNTH	787
81	CONTINUE	SYNTH	788
	WRITE (MTAPE,503) (LL(L), L = 1,LKAR)	SYNTH	789
	IF (PHOTO(1)) WRITE (MTAPE,416)	SYNTH	790
	WRITE (MTAPE,105)	SYNTH	791
C	DO 86 I = 1,NTYPE	SYNTH	792
	N1 = N2 = 0	SYNTH	793
	DO 87 L = 1,MAX	SYNTH	794
	IF (LABEL(L,1),EQ,I) N1 = N1+1	SYNTH	795
87	IF (LABEL(L,2),EQ,I) N2 = N2+1	SYNTH	796
	N = N2-N1	SYNTH	797
	IF (I,EO,NPHOTON,AND,N1,EO,1,AND,N2,EO,1) N = 1	SYNTH	798
		SYNTH	799

```

IF (N.EQ.0) GO TO 86          SYNTH 800
IF (N.GT.0) SIGN = 1H+         SYNTH 801
IF (N.LT.0) SIGN = 1H-         SYNTH 802
ENCODE (10,205,NSIGN) SIGN   SYNTH 803
N = IABS(N)                  SYNTH 804
WRITE (NTAPE,502) I, J, I, J, (NSIGN, L = 1,N)  SYNTH 805
IF (I.NE.1.OR.J.NE.1) GO TO 86  SYNTH 806
    WRITE (NTAPE,105)
    WRITE (NTAPE,503) (LL(L), L = 1,LKAR)
    IF (PHOTO(2)) WRITE (NTAPE,411)
    WRITE (NTAPE,420) NSIGN      SYNTH 807
86 CONTINUE                   SYNTH 808
    IF (RADIATE) WRITE (NTAPE,419) J, J
    WRITE (NTAPE,105)             SYNTH 809
C     A9 CONTINUE                SYNTH 810
C
C     R9 CONTINUE                SYNTH 811
C
C     WRITE (NTAPE,105)           SYNTH 812
C     GO TO 10                   SYNTH 813
5    WRITE (NTAPE,423)           SYNTH 814
    WRITE (6,202)
    WRITE (6,470) DATE          SYNTH 815
    NREJ = INPUT-KTYPE          SYNTH 816
    IF (EXIT.AND.LC.GT.30) WRITE (6,SKIP)  SYNTH 817
    WRITE (6,302) INPUT, KTYPE, KMAX, NREJ, NK, NTYPE, MAXGAS
    IF (EXIT) WRITE (6,303)      SYNTH 818
C     EDIT THE REACTIONS TO DETERMINE WHERE EACH SPECIES OCCURS --
C
LC = 0                         SYNTH 819
KOUNT = 40                      SYNTH 820
TEST = .FALSE.                  SYNTH 821
DO 50 I = 1,NTYPE              SYNTH 822
N = 0                           SYNTH 823
REWIND LTAPE                    SYNTH 824
DO 33 K = 1,KTYPE              SYNTH 825
READ (LTAPE) LHS, RHS, LABEL   SYNTH 826
DO 34 M = 1,2                  SYNTH 827
DO 34 L = 1,5                  SYNTH 828
    IF (LABEL(L,M).EQ.I) GO TO 36  SYNTH 829
34 CONTINUE                     SYNTH 830
    GO TO 37                   SYNTH 831
36 N = N+1                      SYNTH 832
    LINE(N) = K                SYNTH 833
37 IF (N.LT.160.AND.K.LT.KTYPE) GO TO 33  SYNTH 834
    IF (N.EQ.0) GO TO 33        SYNTH 835
    IF (LC.NE.0) GO TO 32        SYNTH 836
    IF (TEST) WRITE (6,202)       SYNTH 837
    IF (TEST) WRITE (6,470) DATE  SYNTH 838
    TEST = .TRUE.
    WRITE (6,300) NTYPE          SYNTH 839
    WRITE (6,202)
32 LC = LC + 2 + N/20            SYNTH 840
    WRITE (6,301) I, GAS(I), (LINE(L), L = 1,N)  SYNTH 841
    IF (LC.GT.KOUNT) LC = 0      SYNTH 842
    N = 0                         SYNTH 843
33 CONTINUE                     SYNTH 844
                                SYNTH 845
                                SYNTH 846
                                SYNTH 847
                                SYNTH 848
                                SYNTH 849
                                SYNTH 850
                                SYNTH 851
                                SYNTH 852
                                SYNTH 853
                                SYNTH 854
                                SYNTH 855
                                SYNTH 856

```

50	CONTINUE	SYNTH	857
	WRITE (6,202)	SYNTH	858
	WRITE (6,470) DATE	SYNTH	859
C		SYNTH	860
	WHITE (NSCRTCH) MAXGAS, NTYPE, KMAX, KTYPE, NKMAX, NK, EXIT	SYNTH	861
	WRITE (NSCRTCH) (GAS(I), I = 1,NTYPE)	SYNTH	862
	WRITE (NSCRTCH) (LEV1(I), LEV2(I), I = 1,NK)	SYNTH	863
C		SYNTH	864
C	COPY LTAPE ONTO NSCRTCH --	SYNTH	865
	REWIND LTAPE	SYNTH	866
	DO 59 K = 1,KTYPE	SYNTH	867
	READ (LTAPE) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM	SYNTH	868
	WRITE (NSCRTCH) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM	SYNTH	869
	59 CONTINUE	SYNTH	870
C		SYNTH	871
	NP1 = NTYPE + 1	SYNTH	872
	NP2 = NTYPE + 2	SYNTH	873
C		SYNTH	874
	WRITE (NTAPE,414)	SYNTH	875
	WRITE (NTAPE,105)	SYNTH	876
	WRITE (NTAPE,406)	SYNTH	877
	WRITE (MTAPE,406)	SYNTH	878
	WRITE (NTAPE,105)	SYNTH	879
	WRITE (MTAPE,105)	SYNTH	880
	ENCODE (80,506,KODE) NP1, NP2	SYNTH	881
	WRITE (NTAPE,102) KODE	SYNTH	882
	WRITE (MTAPE,102) KODE	SYNTH	883
	WRITE (NTAPE,105)	SYNTH	884
	WRITE (MTAPE,105)	SYNTH	885
	WRITE (NTAPE,508)	SYNTH	886
	WRITE (MTAPE,508)	SYNTH	887
	ENCODE (80,509,KODE) NP1, NP2	SYNTH	888
	WRITE (NTAPE,102) KODE	SYNTH	889
	ENCODE (80,513,KODE) NP2, NP1, NP2	SYNTH	890
	WRITE (NTAPE,102) KODE	SYNTH	891
	WRITE (NTAPE,440)	SYNTH	892
	WRITE (NTAPE,105)	SYNTH	893
	ENCODE (80,514,KODE) NP1, NP1	SYNTH	894
	WRITE (NTAPE,102) KODE	SYNTH	895
	ENCODE (80,515,KODE) NP2, NP1	SYNTH	896
	WRITE (NTAPE,102) KODE	SYNTH	897
	WRITE (NTAPE,105)	SYNTH	898
C		SYNTH	899
	ENCODE (80,511,KODE) NP1, NP1	SYNTH	900
	WRITE (MTAPE,102) KODE	SYNTH	901
	ENCODE (80,516,KODE) NP1, NP2	SYNTH	902
	WRITE (MTAPE,102) KODE	SYNTH	903
	ENCODE (80,517,KODE) NP2, NP1	SYNTH	904
	WRITE (MTAPE,102) KODE	SYNTH	905
	ENCODE (80,518,KODE) NP2, NP2	SYNTH	906
	WRITE (MTAPE,102) KODE	SYNTH	907
	ENCODE (80,519,KODE) NP2, NP2	SYNTH	908
	WRITE (MTAPE,102) KODE	SYNTH	909
	WRITE (MTAPE,460)	SYNTH	910
	WRITE (MTAPE,105)	SYNTH	911
	ENCODE (80,521,KODE) NP1, NP1	SYNTH	912
	WRITE (MTAPE,102) KODE	SYNTH	913

```

ENCODE (80+520,KODE) NP1, NP1           SYNTH   914
WRITE (MTAPE,102) KODE                  SYNTH   915
WRITE (MTAPE,105)                       SYNTH   916
C
WRITE (INTAPE,406)                      SYNTH   917
WRITE (INTAPE,105)                      SYNTH   918
WRITE (INTAPE,406)                      SYNTH   919
WRITE (INTAPE,105)                      SYNTH   920
ENCODE (80+207,KODE)                   SYNTH   921
WRITE (INTAPE,102) KODE                  SYNTH   922
WHITE (INTAPE,105)                      SYNTH   923
WRITE (INTAPE,102) KODE                  SYNTH   924
WRITE (INTAPE,105)                      SYNTH   925
N2 = 0                                  SYNTH   926
12 N1 = N2+1                           SYNTH   927
IF (N1.GT.NTYPE) GO TO 19              SYNTH   928
N2 = N1+3                            SYNTH   929
IF (N2.GT.NTYPE) N2 = NTYPE            SYNTH   930
ENCODE (80+206,KODE) (L, GAS(L), L = N1,N2)
WRITE (INTAPE,102) KODE                  SYNTH   931
WRITE (INTAPE,102) KODE                  SYNTH   932
GO TO 12                               SYNTH   933
19 WRITE (INTAPE,105)                   SYNTH   934
WRITE (INTAPE,406)                      SYNTH   935
WRITE (INTAPE,105)                      SYNTH   936
WRITE (INTAPE,440)                      SYNTH   937
WRITE (INTAPE,450)                      SYNTH   938
WRITE (INTAPE,105)                      SYNTH   939
WRITE (INTAPE,406)                      SYNTH   940
WRITE (INTAPE,105)                      SYNTH   941
WRITE (INTAPE,406)                      SYNTH   942
WRITE (INTAPE,105)                      SYNTH   943
WRITE (INTAPE,440)                      SYNTH   944
WRITE (INTAPE,450)                      SYNTH   945
C
C CONSTRUCT SUBROUTINE TO COMPUTE POPULATION DENSITIES FOR THE
C LOWER AND UPPER LEVELS TO BE USED IN THE E-KINETICS ANALYSIS --
C
IF (NK.EQ.0) GO TO 75                 SYNTH   946
WRITE (INTAPE,600)                      SYNTH   947
WRITE (INTAPE,105)                      SYNTH   948
WRITE (INTAPE,406)                      SYNTH   949
WRITE (INTAPE,105)                      SYNTH   950
WRITE (INTAPE,604)                      SYNTH   951
WRITE (INTAPE,105)                      SYNTH   952
WRITE (INTAPE,460)                      SYNTH   953
WRITE (INTAPE,406)                      SYNTH   954
WRITE (INTAPE,105)                      SYNTH   955
ENCODE (80+601,KODE)                  SYNTH   956
WRITE (INTAPE,102) KODE                  SYNTH   957
WRITE (INTAPE,105)                      SYNTH   958
DO 80 I = 1,NK                         SYNTH   959
ENCODE (80+602,KODE) M1, I, LEV1(I)    SYNTH   960
IF (LEV1(I).EQ.0) ENCODE (80+603,KODE) M1, I
WRITE (INTAPE,102) KODE                  SYNTH   961
ENCODE (80+602,KODE) M2, I, LEV2(I)    SYNTH   962
IF (LEV2(I).EQ.0) ENCODE (80+603,KODE) M2, I
WRITE (INTAPE,102) KODE                  SYNTH   963
80 WRITE (INTAPE,105)                   SYNTH   964

```

```

      WRITE (NTAPE,440)
      WRITE (NTAPE,450)

C   COPY NTAPE ONTO MTAPE --
C
      75 REWIND NTAPE
      40 READ (NTAPE,102) KODE
         IF (EOF(NTAPE)) 99,71
      71 WRITE (MTAPE,102) KODE
         GO TO 40
      99 REWIND MTAPE
         REWIND NTAPE

C----- FORMAT STATEMENTS -----
C
      100 FORMAT (8A1)
C      101 FORMAT (2E10.3,5X,5A10)
C      102 FORMAT (8A10)
C
      103 FORMAT (1PE10.4)
C
      104 FORMAT (6X*COMMON / DATA / RATE(*I4*), KF(*I4*), KR(*I4*), VSIG(2,
     1*I3*), E(*I2*)*8X)
C
      105 FORMAT (1HC,79X)
C
      106 FORMAT (6X*E*,I1,* =*)
C
      107 FORMAT (I4,I6)
C
      108 FORMAT (* * E(*,I2,*)*)
C
      109 FORMAT (A10,5A8,30X)
C
      110 FORMAT (*VSIG(*,I1,*,*I2,*)*)

C
      200 FORMAT (1H1/17X*SUMMARY OF INPUT: REACTIONS AND RATE CONSTANTS IS
     1EC-1, CM3/SEC, CM6/SEC, ... OR CM2) WITH REFERENCES*//10X*(IF A
     2RATE CONSTANT KF OR KR FOR A BINARY ELECTRON COLLISION IS NOT EXPL
     3ICITLY SPECIFIED, IT WILL BE COMPUTED SELF*/10X*CONSISTENTLY AS A
     4FUNCTION OF E/N, GAS COMPOSITION, AND EXCITED LEVEL DENSITIES FROM
     5A COUPLED ELECTRON ANALYSIS.)*//4X,*I*12X,*REACTION(I)*,34X,
     6*RATE CONSTANTS*14X*RATE REFERENCES AND/OR COMMENTS*/BX*(IGNORED R
     7EACTIONS ARE NOT NUMBERED)*,16X,*KF(I)*8X,*KR(I)*/1X,134(1H-)//)
C
      201 FORMAT (A8+45A1,4X,A10,3X,A10,4X,5A10/(84X+5A10))
C
      202 FORMAT (/1X,134(1H-))
C
      203 FORMAT (*REVERSE RATE IS OBTAINED FROM DETAILED BALANCE.*)
C
      204 FORMAT (*REACTION REJECTED -- CHARGE CONSERVATION VIOLATED.*)
C
      205 FORMAT (1H ,A1,* R*)

```

C 206 FORMAT (*C*4(I4,3X,A10))	SYNTH	1028
C 207 FORMAT (*C THE FOLLOWING MOLECULAR SPECIES (WITH LABELS) WERE INCLUDED --*)	SYNTH	1029
C 208 FORMAT (*C THE FOLLOWING REACTIONS DEFINE THE KINETICS --*)	SYNTH	1030
C 209 FORMAT (I5)	SYNTH	1031
C 210 FORMAT (*REACTION IGNORED -- SAME AS NUMBER*,I4,*.*)	SYNTH	1032
C 211 FORMAT (*REACTION IGNORED -- REVERSE OF NO.,I4,*.*)	SYNTH	1033
C 212 FORMAT (*REACTION IS IGNORED -- KF = KR = 0.*)	SYNTH	1034
C 213 FORMAT (*REACTION IS IGNORED -- MORE THAN*,I3,* DIFFERENT GAS SPECIES ARE NOT PERMITTED WITH PRESENT DIMENSION.*)	SYNTH	1035
C 214 FORMAT (*> *,I1,* SPECIES ON LHS OR RHS NOT PERMITTED.*)	SYNTH	1036
C 215 FORMAT (*MORE THAN *,I3,* REACTIONS ARE IGNORED.*)	SYNTH	1037
C 216 FORMAT (*BAD SYNTAX -- UNRECOGNIZABLE REACTION IS IGNORED.*)	SYNTH	1038
C 217 FORMAT (I1X*RATE(*I3*) = R*56X)	SYNTH	1039
C 218 FORMAT (*REACTION REJECTED -- NO MORE THAN*,I3,* REACTIONS ARE ALLOWED FOR THE COUPLED E- KINETICS ANALYSIS.*)	SYNTH	1040
C 219 FORMAT (*WARNING -- NO E- CROSS SECTION DATA WAS FOUND.*)	SYNTH	1041
C 220 FORMAT (*FORWARD REACTION IS IGNORED -- KF = 0.*)	SYNTH	1042
C 221 FORMAT (*NO REVERSE REACTION ALLOWED FOR RADIATIVE DECAY.*)	SYNTH	1043
C 222 FORMAT (*REVERSE REACTION IS IGNORED -- KR = 0.*)	SYNTH	1044
C 223 FORMAT (A7,* RATE IS OBTAINED FROM E- KINETICS ANALYSIS*)	SYNTH	1045
C 224 FORMAT (*IMPROPER BUFFER GAS SPECIFICATION.*)	SYNTH	1046
C 225 FORMAT (*IMPROPER HIGH ENERGY ELECTRON TERMS.*)	SYNTH	1047
C 226 FORMAT (*THREE-BODY HE- COLLISION NOT ALLOWED.*)	SYNTH	1048
C 227 FORMAT (*THIS RADIATIVE PROCESS NOT ALLOWED.*)	SYNTH	1049
C 300 FORMAT (IHI/35X,*SUMMARY OF REACTIONS FOR WHICH EACH SPECIES OCCURS: NTYPE = *,I3/31X*(THIS EDIT PERMITS RAPID DELETION OF ANY SPECIES FROM THE KINETIC SYSTEM)//7X*,I*5X*GAS(I)*40X*REACTIONS CONTAINING GAS(I*))	SYNTH	1050
C 301 FORMAT (/I8,5X,A10,5X,I4,19(*,*I4)/(28X,I4,19(*,*I4)))	SYNTH	1051
C 302 FORMAT (/4X,*OF*,I4,* INPUT REACTIONS SCANNED*,I4,* WERE RETAINED 1 (MAXIMUM ALLOWED = *,I4,* AND *,I4,* WERE IGNORED FOR REASONS ITEM 2IZED IN THE TABLE.*/4X,*OF THOSE RETAINED,*I3,* REQUIRE RATES FROM	SYNTH	1052
	SYNTH	1053
	SYNTH	1054
	SYNTH	1055
	SYNTH	1056
	SYNTH	1057
	SYNTH	1058
	SYNTH	1059
	SYNTH	1060
	SYNTH	1061
	SYNTH	1062
	SYNTH	1063
	SYNTH	1064
	SYNTH	1065
	SYNTH	1066
	SYNTH	1067
	SYNTH	1068
	SYNTH	1069
	SYNTH	1070
	SYNTH	1071
	SYNTH	1072
	SYNTH	1073
	SYNTH	1074
	SYNTH	1075
	SYNTH	1076
	SYNTH	1077
	SYNTH	1078
	SYNTH	1079
	SYNTH	1080
	SYNTH	1081
	SYNTH	1082
	SYNTH	1083
	SYNTH	1084

3 AN E- KINETICS ANALYSIS. \*\*I3,\* SEPARATE SPECIES WERE ENCOUNTERED SYNT 1085  
 4 (MAXIMUM ALLOWED =\*,I3,\*).\*/)  
 C  
 303 FORMAT (4X\*ERRORS WHICH WERE DETECTED IN PROCESSING THE INPUT REAC SYNT 1086  
 1T1N SCHEME MAY CAUSE PROGRAM TERMINATION IF THEY HAVE BEEN SPECIF SYNT 1087  
 2IED TO\*/4X,\*BE TREATED AS FATAL. MODIFICATIONS OF THE REACTION SC SYNT 1088  
 3HEME, CORRECTIONS IN REACTION SYNTAX, CHANGES IN DIMENSION STORAGE SYNT 1089  
 4, OR\*/4X,\*ADDITIONS TO THE E- CROSS SECTION FILE MAY BE REQUIRED T SYNT 1090  
 50 REMOVE ALL OF THE ERROR DECLARATIONS.\*)" SYNT 1091  
 C  
 304 FORMAT (1IX#DVDX = DVDX + U\*S#52X) SYNT 1092  
 C  
 305 FORMAT (1IX\*U = (E2 - E1)\*.56X) SYNT 1093  
 C  
 306 FORMAT (\*K\*,A1,\*(\*,I3,\*)) SYNT 1094  
 C  
 307 FORMAT (I1+3H.\*K,A1,\*(\*,I3,\*)) SYNT 1095  
 C  
 308 FORMAT (4H\*NO(,I2,\*)) SYNT 1096  
 C  
 309 FORMAT (\*C\*5X\*E-BEAM ENERGY DEPOSITION --#47X) SYNT 1097  
 C  
 310 FORMAT (\*C\*79X/\*C\*5X\*SECONDARY ELECTRON CREATION--#45X/1IX\*SB = SB SYNT 1098  
 1 + R\*5RX/1IX\*U = U + UPLUS#56X) SYNT 1099  
 C  
 311 FORMAT (6X\*S = R#69X/6X\*R = R\*IBEAM/E0#60X) SYNT 1100  
 C  
 312 FORMAT (\*E- CREATION ASSUMED TO BE OVER ENERGY DISTRIBUTION\*) SYNT 1101  
 C  
 313 FORMAT (\*E- CREATED (OR LOST) ASSUMED TO BE AT ZERO ENERGY\*) SYNT 1102  
 C  
 400 FORMAT (\*#DECK,DNDT#,70X/6X\*SURROUNTING DNDT (N, T, NO, NDOT)\*42X) SYNT 1103  
 C  
 401 FORMAT (6X\*DIMENSION NO(1), NDOT(1)\*50X) SYNT 1104  
 C  
 402 FORMAT (6X\*REAL NO, NTOT, NDOT, NOISE, NE, KF, KR, KB, KT, MU, LO,  
 1 IBEAM,\*12X/5X\*1 JBEAM, LENGTH#60X) SYNT 1105  
 C  
 403 FORMAT (6X\*IBEAM = JBEAM\*DEPOSIT\*SHAPE(T)\*44X/6X\*KT = KB\*TMOL#62X) SYNT 1106  
 C  
 404 FORMAT (IHC,I3.2X,6A10,5X) SYNT 1107  
 C  
 405 FORMAT (\*C FORWARD RATE IS OBTAINED FROM E(-) KINETICS ANALYSIS  
 1S\*.22X/1IX,\*KF(\*,I3,\*)) = VSIG(\*,I1,\*,\*I2,\*)\*,48X) SYNT 1108  
 C  
 406 FORMAT (\*C \*.69(IH.)\*8X) SYNT 1109  
 C  
 407 FORMAT (6X\*DO 1 I = 1,N#62X/4X\*1 NDOT(I) = 0.\*62X/6X\*NDOT(1) = - C SYNT 1110  
 1\*GAMMA\*NO(1)\*49X/6X\*ALPHA = GAIN = HNU = FREQ = NOISE = DVDX = 0.\*  
 229X/6X\*SB = SO = 0.0\*61X) SYNT 1111  
 C  
 408 FORMAT (\*C\*.5X,\*REVERSE RATE IS OBTAINED FROM DETAIL BALANCE ---, SYNT 1112  
 127X) SYNT 1113  
 C  
 409 FORMAT (\*C REVERSE RATE IS OBTAINED FROM E(-) KINETICS ANALYSIS  
 1S\*.22X/1IX,\*KR(\*,I3,\*)) = VSIG(\*,I1,\*,\*I2,\*)\*,48X) SYNT 1114  
 C  
 SYNT 1115  
 SYNT 1116  
 SYNT 1117  
 SYNT 1118  
 SYNT 1119  
 SYNT 1120  
 SYNT 1121  
 SYNT 1122  
 SYNT 1123  
 SYNT 1124  
 SYNT 1125  
 SYNT 1126  
 SYNT 1127  
 SYNT 1128  
 SYNT 1129  
 SYNT 1130  
 SYNT 1131  
 SYNT 1132  
 SYNT 1133  
 SYNT 1134  
 SYNT 1135  
 SYNT 1136  
 SYNT 1137  
 SYNT 1138  
 SYNT 1139  
 SYNT 1140  
 SYNT 1141

```

C 410 FORMAT (*      KR(*,I3,*) = KF(*,I3,18H)*EXP(-(E1-E2)/KT)*40X)      SYNTH 1142
C 411 FORMAT (11X*GAIN = GAIN + R*54X)                                         SYNTH 1143
C 412 FORMAT (*C*5X*(STIMULATED EMISSION PROCESS, WITH NO(*I2*) = INTEN/    SYNTH 1144
  1C/HNU)*17X/*C*79X)                                                       SYNTH 1145
C 413 FORMAT (*C*5X*(RADIATIVE ABSORPTION PROCESS, WITH NO(*I2*) = INTEN    SYNTH 1146
  1/C/HNU)*16X/*C*79X)                                                       SYNTH 1147
C 414 FORMAT (*C*5X*COMPUTE FINAL EXPRESSION FOR D/DT(PHOTON DENSITY) --   SYNTH 1148
  1*22X/*C*79X/6X*NDOT(1) = (LENGTH/CAVITY)*(NDOT(1) + NOISE)*31X/*C*   SYNTH 1149
  279X/6X*ABSORB = GAIN - ALPHA*53X/6X,*DVDX = DEPOSIT*DWDX*55X)          SYNTH 1150
C 415 FORMAT (*C*5X*GAIN = SIGMA*(N2-N1) IS THE LASER TRANSITION GAIN#   SYNTH 1151
  1 23X/*C*5X*ABSORB = SUMK(SIGMA(K)*NK) IS THE TOTAL ABSORPTION OF T   SYNTH 1152
  2HE MEDIUM*10X/*C*5X*ALPHA = (GAIN-ABSORB) IS THE NET GAIN IN THE   SYNTH 1153
  3MEDIUM*22X/*C*5X*GAMMA = THRESHOLD GAIN COEFFICIENT (CM-1)*31X)        SYNTH 1154
C 416 FORMAT (6X*R = R*C*67X)                                                 SYNTH 1155
C 417 FORMAT (6X*HNU = E0*(E1 - E2)*56X/6X*FREQ = HNU/H*62X)               SYNTH 1156
C 418 FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE --*   SYNTH 1157
  131X/*C*79X/11X*NOISE = NOISE + R*OMEGA/4./PI*40X)                      SYNTH 1158
C 419 FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE --*   SYNTH 1159
  131X/*C*79X/11X*R = R*OMEGA/4./PI*51X/11X*PHI(1.*I2*) = PHI(1.*I2   SYNTH 1160
  2*) + R*44X)                                                               SYNTH 1161
C 420 FORMAT (11X*ALPHA = ALPHA*A4*52X)                                         SYNTH 1162
C 421 FORMAT (6X*DATA KB, E0, H, C, PI /*1PE10.3*, *1PE10.3*, *1PE10.3*, *   SYNTH 1163
  118X/5X*1 *1PE10.3*, 3.14159 /*49X/*C*79X)                                SYNTH 1164
C 422 FORMAT (*C*5X*CAVITY = MIRROR SEPARATION (CM)*44X/*C*5X*LENGTH = L   SYNTH 1165
  LENGTH OF ACTIVE MEDIUM (CM)*37X/*C*5X*OMEGA = AREA/CAVITY**2*51X/   SYNTH 1166
  2*C*5X*AREA = AREA OF OPTICS (CM2)*45X/*C*5X*GAMMA = [LOSS + LN(   SYNTH 1167
  31/R)/2]/LENGTH*40X/*C*79X)                                              SYNTH 1168
C 423 FORMAT (6X*RATIO = LENGTH/CAVITY*53X/6X*DO 3 I = 1,N*62X/4X#3 PHI(   SYNTH 1169
  11*I) = RATIO*PHI(1,I)*49X/*C*79X)                                         SYNTH 1170
C 424 FORMAT (*C*5X*CREATION OF (ZERO ENERGY) SECONDARY ELECTRONS--*27X)   SYNTH 1171
C 425 FORMAT (*C*5X*LOSS OF (ZERO ENERGY) SECONDARY ELECTRONS --*30X)       SYNTH 1172
C 426 FORMAT (11X*S0 = S0 *A1* R*58X)                                         SYNTH 1173
C 440 FORMAT (6X*RETURN*68X)                                                 SYNTH 1174
C 450 FORMAT (6X*END*71X)                                                   SYNTH 1175
C 460 FORMAT (*C*5X*THE GENERAL KINETICS SYNTHESIS PROGRAM WHICH AUTOMAT   SYNTH 1176
  ICALLY GEN-*11X/*C*5X*ERATED THIS SUBROUTINE WAS DEVELOPED BY --*   SYNTH 1177
  232X/*C*79X/*C*15X*40(IH-1)*24X/*C*15X*I*38X*I*24X/*C*15X*I DR. WI   SYNTH 1178
  3LLIAM B. LACINA*I4X*I*24X/*C*15X*I NORTHRUP RESEARCH AND TECHNOL   SYNTH 1179
                                         SYNTH 1180
                                         SYNTH 1181
                                         SYNTH 1182
                                         SYNTH 1183
                                         SYNTH 1184
                                         SYNTH 1185
                                         SYNTH 1186
                                         SYNTH 1187
                                         SYNTH 1188
                                         SYNTH 1189
                                         SYNTH 1190
                                         SYNTH 1191
                                         SYNTH 1192
                                         SYNTH 1193
                                         SYNTH 1194
                                         SYNTH 1195
                                         SYNTH 1196
                                         SYNTH 1197
                                         SYNTH 1198

```

40GY I=24X/\*C=15X\*I ONE RESEARCH PARK\*18X\*I=24X/\*C=15X\*I PALO SYNT 1199  
 55 VERDES PENINSULA, CA 90274 I=24X/\*C=15X\*I TEL: (213) 377-481 SYNT 1200  
 61\* EXT. 322\*6X\*I=24X/\*C=15X\*I=38X\*I=24X/\*C=15X,40(1H-),24X/\*C=79X) SYNT 1201  
 C 470 FORMAT (/12X\*GENERALIZED KINETICS SYNTHESIS CODE: DR. WILLIAM B. L SYNT 1202  
 IACINA, NORTHROP RESEARCH AND TECHNOLOGY. DATE:\*A1) SYNT 1203  
 C 500 FORMAT (\*11X,5HNDOT(.I\*,I3\*,\*9H) = NDOT(.I\*,I3\*,\*1H),\*,I2,\*A4)\*) SYNT 1204  
 C 502 FORMAT (11X\*PHI(\*,I2\*,\*,I2\*) = PHI(\*,I2,\*,\*,I2,\*)\*.10A4,6X) SYNT 1205  
 C 503 FORMAT (6X,\*R = \*.62A1,8X/(5X,\*\$\*10X,56A1,8X)) SYNT 1206  
 C 504 FORMAT (6X\*DIMENSION PHI(N,1), NO(1)\*49X) SYNT 1207  
 C 505 FORMAT (\*C=5X\*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT SYNT 1208  
 FILE OF SYM-\*10X/\*C=5X\*BOLIC REACTIONS WHICH DEFINE A COUPLED SYST SYNT 1209  
 2EM OF ELECTRON AND\*13X/\*C=5X\*MOLECULAR KINETICS EQUATIONS. IT RET SYNT 1210  
 3URNS THE JACOBIAN MATRIX,\*11X/\*C=79X/\*C=20X,\*PHI(I,J) = D(DOT(I)) SYNT 1211  
 4/D(NO(J))\*.29X/\*C=79X/\*C=5X\*WHERE I,J = 1,2,3,...,NTYPE. N IS THE D SYNT 1212  
 5IMENSION DECLARATOR FOR PHI\*8X/\*C=5X\*IN THE CALLING PROGRAM. THE SYNT 1213  
 6RATE CONSTANTS KF AND KR HAVE UNITS\*10X/\*C=5X\*OF CM2\* SEC-1\* CM3/S SYNT 1214  
 7EC\* CM6/SEC. ... AS APPROPRIATE.\*,22X) SYNT 1215  
 C 506 FORMAT (\*C=5X\*EXTERNAL CIRCUIT EQUATIONS (Q = NO(\*,I2,\*), AND I = SYNT 1216  
 INQ(\*,I2,\*)) -- \*15X) SYNT 1217  
 C 507 FORMAT (6X\*DO 1 I = 1,N\*,62X/6X\*DO 1 J = 1,N\*,62X/4X\*I PHI(I,J) = SYNT 1218  
 10.\*61X/6X\*PHI(I,J) = - C\*GAMMA\*54X) SYNT 1219  
 C 508 FORMAT (6X\*NE = NO(2)\*64X/6X\*IF (NE.EQ.0.) NE = 1.0\*52X/6X\*CONDUCT SYNT 1220  
 1 = NE\*E0\*MU\*56X/6X\*RD = D/AREA/CONDUCT\*55X/\*C=79X/6X\*IF (L0.EQ.0.) SYNT 1221  
 2 GO TO 2\*53X) SYNT 1222  
 C 509 FORMAT (6X\*DQDT = NDOT(\*I2\*) = NO(\*I2\*)\*) SYNT 1223  
 C 510 FORMAT (\*DECK,JACOB\*69X/6X\*SUBROUTINE JACOB (N, T, NO, PHI)\*42X) SYNT 1224  
 C 511 FORMAT (6X\*PHI(\*,I2\*,\*,I2\*) = 0.\*59X) SYNT 1225  
 C 512 FORMAT (6X\*COMMON / CONST / NTOT, THOL, FREQ, HNU\*36X/6X\*COMMON / SYNT 1226  
 1DISCH / L0, C0, R0, MU, AREA, D\*34X/6X\*COMMON / SOURCE / UPLUS, JB SYNT 1227  
 2EAM, DVDX, DEPOSIT, ENERGY, SB, S0\*13X/6X\*COMMON / GAINS / ALPHA, SYNT 1228  
 3GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY\*8X/\*C=79X) SYNT 1229  
 C 513 FORMAT (6X\*DIDT = NDOT(\*I2\*) = (-NO(\*I2\*)/C0 - (R0 + RD)\*1H\*,\*NO(\* SYNT 1230  
 1I2\*)/L0\*22X) SYNT 1231  
 C 514 FORMAT (4X\*2 DQDT = NDOT(\*I2\*) = - NO(\*I2\*)/C0/(R0 + RD)\*) SYNT 1232  
 C 515 FORMAT (6X\*CURRENT = NO(\*I2\*) = NDOT(\*I2\*)\*) SYNT 1233  
 C 516 FORMAT (6X\*PHI(\*I2\*,\*,I2\*) = 1.0\*) SYNT 1234  
 C 517 FORMAT (6X\*PHI(\*I2\*,\*,I2\*) = -1./L0/C0\*52X) SYNT 1235  
 C 518 FORMAT (6X\*PHI(\*I2\*,\*,I2\*) = -(R0 + RD)/L0\*48X) SYNT 1236

```

C      519 FORMAT (6X*PHI(*I2*, 2) = NO(*I2*)*IH*, *RD/NE/L0*46X)      SYNTH 1256
C      520 FORMAT (6X*PHI(*I2*, 2) = NO(*I2*)*DIDQ*RD/NE/(R0 + RD)*)    SYNTH 1257
C      521 FORMAT (4X*2 PHI(*I2*,*I2*) = DIDQ = - 1./C0/(R0 + RD)*)      SYNTH 1258
C      600 FORMAT (*#DECK,LEVELS#68X/6X*SUBROUTINE LEVELS (N1, N2, NO)*44X) SYNTH 1259
C      601 FORMAT (6X,*REAL N1(1), N2(1), NO(1)*)
C      602 FORMAT (6X,*N*, I1,*(*,I2,*)) = NO(*,I2,*)
C      603 FORMAT (6X,*N*, I1,*(*,I2,*)) = 0.**
C      604 FORMAT (*C      THIS SUBROUTINE DETERMINES THE POPULATION DENSITIES 1 SYNTH 1260
C           1 N1(I), N2(I)*10X/*C*5X*OF THE (LOWER AND UPPER) LEVELS INVOLVED IN SYNTH 1261
C           2N THE ITH INELASTIC*13X/*C*5X*SCATTERING PROCESS INCLUDED IN THE C SYNTH 1262
C           20UPLED E- KINETICS ANALYSIS.*10X) SYNTH 1263
C      605 FORMAT (*C*5X*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT 1 SYNTH 1264
C           1FILE OF SYM-*10X/*C*5X*BOLIC REACTIONS WHICH DEFINE A COUPLED SYST SYNTH 1265
C           2EM OF ELECTRON AND*13X/*C*5X*MOLECULAR KINETICS. IT RETURNS THE RA SYNTH 1266
C           3TES NDOT(I) = (D/DT)NO(I),*11X/*C*5X*I = 1,2...NTYPE (CM-3/SEC). SYNTH 1267
C           4RATE CONSTANTS KF AND KR HAVE UNITS*10X/*C*5X*OF CM2, SEC-1, CM3/S SYNTH 1268
C           SEC, CM6/SEC, ... AS APPROPRIATE.*922X) SYNTH 1269
C
C      -----
C      RETURN
C      END

```

```

SUBROUTINE ANALYZE (NTYPE, KTYPE, RATE, NTIME, RPCT, FLAG, PMAX, ANALYZE 2
1 GAS, PCT, KAPTION, LTAPE, MTAPE, NTAPE) ANALYZE 3
C ..... ANALYZE 4
C ..... ANALYZE 5
C ..... ANALYZE 6
C ..... THIS SUBROUTINE ANALYZES THE CONTRIBUTIONS OF ALL REACTIONS TO ANALYZE 7
C EVERY SPECIES, AND PRINTS OUT DIAGNOSTICS SUMMARIZING THE SENSITIVITY ANALYZE 8
C OF EACH REACTION TO THE TOTAL CALCULATION. ANALYZE 9
C ..... ANALYZE 10
C ..... ANALYZE 11
C ..... ANALYZE 12
C ..... DIMENSION RATE(1), NTIME(1), KAPTION(4), KODE(10), PMAX(1), ANALYZE 13
C           1 RPCT(1), FLAG(1), GAS(1) ANALYZE 14
C ..... LOGICAL FLAG, TEST ANALYZE 15
C
C      REWIND LTAPE ANALYZE 16
C      REWIND MTAPE ANALYZE 17
C      REWIND NTAPE ANALYZE 18
C      DO 8 I = 1,NTYPE ANALYZE 19
C      8 PMAX(I) = 0. ANALYZE 20
C      DO 1 K = 1,KTYPE ANALYZE 21
C      R = RATE(K) ANALYZE 22
C      READ (MTAPE) (NTIME(L), L = 1,NTYPE) ANALYZE 23
C      DO 2 I = 1,NTYPE ANALYZE 24
C      RPCT(I) = 0. ANALYZE 25
C      IF (R.EQ.0.) GO TO 2 ANALYZE 26
C      NI = NTIME(I) ANALYZE 27
C      IF (NI.EQ.0) GO TO 2 ANALYZE 28
C      RPCT(I) = NI*R ANALYZE 29
C      PABS = ABS(RPCT(I)) ANALYZE 30
C      IF (PABS.GT.PMAX(I)) PMAX(I) = PABS ANALYZE 31
C      2 CONTINUE ANALYZE 32
C      1 WRITE (NTAPE) (RPCT(L), L = 1,NTYPE) ANALYZE 33
C      REWIND NTAPE ANALYZE 34
C

```

```

C          DO 3 K = 1, KTYPE          ANALYZE 37
C          PABS = 0.                ANALYZE 38
C          READ (INTAPE) (RPCT(L), L = 1, NTYPE)    ANALYZE 39
C          DO 4 I = 1, NTYPE          ANALYZE 40
C          IF (PMAX(I).EQ.0.) GO TO 4    ANALYZE 41
C          PERCENT = RPCT(I) = 100.*RPCT(I)/PMAX(I)    ANALYZE 42
C          PERCENT = ABS(PERCENT)      ANALYZE 43
C          IF (PERCENT.GT.PCT) FLAG(K) = .FALSE.    ANALYZE 44
C          IF (PERCENT.GT.PABS) PABS = PERCENT    ANALYZE 45
C          4 CONTINUE                  ANALYZE 46
C          3 WRITE (LTAPE) (RPCT(L), L = 1, NTYPE), PABS    ANALYZE 47
C
C          NA = 1                      ANALYZE 48
C          5 NB = NA*9                ANALYZE 49
C          IF (NB.GT.NTYPE) NB = NTYPE    ANALYZE 50
C          NUASH = 37 + 11*(NB-NA)      ANALYZE 51
C          NX = (138-NDASH)/2         ANALYZE 52
C          REWIND LTape               ANALYZE 53
C          TEST = .FALSE.             ANALYZE 54
C          DO 6 K = 1, KTYPE          ANALYZE 55
C          K1 = K-1                  ANALYZE 56
C          IF (K1.NE.50*(K1/50)) GO TO 9    ANALYZE 57
C          IF (K1.EQ.0) GO TO 7        ANALYZE 58
C          WRITE (6,100) NX, NDASH     ANALYZE 59
C          IF (TEST) WRITE (6,101) PCT    ANALYZE 60
C          7 WRITE (6,102) KAPTION, NX, (GAS(I), I = NA,NB)    ANALYZE 61
C          WRITE (6,100) NX, NDASH     ANALYZE 62
C          TEST = .FALSE.             ANALYZE 63
C          9 READ (LTape) (RPCT(L), L = 1, NTYPE), PCTMAX    ANALYZE 64
C          R = RATE(K)                ANALYZE 65
C          NFLAG = 1H                 ANALYZE 66
C          IF (FLAG(K)) NFLAG = 1H+    ANALYZE 67
C          IF (FLAG(K)) TEST = .TRUE.   ANALYZE 68
C          DO 10 I = NA,NB            ANALYZE 69
C          I1 = I-NA+1                ANALYZE 70
C          KODE(I1) = 1H              ANALYZE 71
C          IF (RPCT(I).EQ.0.) GO TO 10    ANALYZE 72
C          ENCODE (10,104,KODE(I1)) RPCT(I)    ANALYZE 73
C          10 CONTINUE                  ANALYZE 74
C          6 WRITE (6,103) NX, NFLAG, K, R, PCTMAX, (KODE(I), I = 1,I1)    ANALYZE 75
C          WRITE (6,100) NX, NDASH     ANALYZE 76
C          IF (TEST) WRITE (6,101) PCT    ANALYZE 77
C          NA = NB*1                  ANALYZE 78
C          IF (NA.LE.NTYPE) GO TO 5    ANALYZE 79
C
C          ----- FORMAT STATEMENTS -----
C
C          100 FORMAT (//=X,=(1H-1)/)    ANALYZE 80
C
C          101 FORMAT (1BX* THIS REACTION CONTRIBUTES LESS THAN #F3.0* % TO ALL ISPECIES THROUGHOUT THE ENTIRE CALCULATION SO FAR*)    ANALYZE 81
C
C          102 FORMAT (1H1,47X,4A10,/29X*PERCENTAGE CONTRIBUTION OF REACTION K TO 1 DN(I)/DT, EXPRESSED (FOR EACH SPECIES)*/33X*AS A PERCENTAGE OF TH 2E MAXIMUM RATE OCCURRING FOR ALL REACTIONS INCLUDED*//=X,4X*K*4X, 3*RATE(K)*4X*MAX %*5X*8(1X,A10),A10,A7)    ANALYZE 82
C
C          103 FORMAT (=X,A1,I4,1PE12.3,0PF8.1,10(1X,A10))    ANALYZE 83
C
C          104 FORMAT (F10.3)            ANALYZE 84
C
C          -----
C          RETURN                     ANALYZE 85
C          END                         ANALYZE 86
C
C          -----
C
C          ANALYZE 87
C          ANALYZE 88
C          ANALYZE 89
C          ANALYZE 90
C          ANALYZE 91
C          ANALYZE 92
C          ANALYZE 93
C          ANALYZE 94
C          ANALYZE 95
C          ANALYZE 96
C          ANALYZE 97
C          ANALYZE 98
C          ANALYZE 99
C          ANALYZE 100
C          ANALYZE 101
C          ANALYZE 102

```

```

SUBROUTINE DEKODE (NAME, IMAGE, LHS, RHS, LABEL, GAS, NSIZE,
1 NTYPE, LONG)                                DEKODE   2
C
DIMENSION NAME(1), IMAGE(1), GAS(5,2), LABEL(5,2), KAR(10)    DEKODE   3
INTEGER LHS, RHS, GAS, E, HNU                  DEKODE   4
E = 4HE(-)                                     DEKODE   5
HNU = 3HNU                                     DEKODE   6
NO = NTYPE                                     DEKODE   7
DO 1 L = 1,5                                    DEKODE   8
DO 1 M = 1,2                                    DEKODE   9
GAS(L,M) = 1H                                   DEKODE  10
1 LABEL(L,M) = 0                               DEKODE  11
M = 1                                         DEKODE  12
I = J = N = MM = 0                            DEKODE  13
2 IF (I.EQ.LONG) GO TO 4                      DEKODE  14
I = I+1                                       DEKODE  15
IF (IMAGE(I).EQ.1H) GO TO 2                  DEKODE  16
IF (IMAGE(I).NE.1H) GO TO 3                  DEKODE  17
MM = 1                                         DEKODE  18
GO TO 4                                       DEKODE  19
3 IF (IMAGE(I).NE.1H) GO TO 6                  DEKODE  20
IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H+) GO TO 6    DEKODE  21
MM = 0                                         DEKODE  22
GO TO 4                                       DEKODE  23
6 IF (J.EQ.NSIZE) GO TO 2                  DEKODE  24
J = J+1                                       DEKODE  25
KAR(J) = IMAGE(I)                           DEKODE  26
GO TO 2                                       DEKODE  27
4 IF (J.EQ.0) GO TO 99                      DEKODE  28
IF (M.GT.2) GO TO 99                      DEKODE  29
ENCODE (10,100,NGAS) (KAR(L), L = 1,J)      DEKODE  30
100 FORMAT (10A1)                           DEKODE  31
IF (NGAS.EQ.1HE.OR.NGAS.EQ.2HE-1) NGAS = E    DEKODE  32
J = 0                                         DEKODE  33
N = N+1                                       DEKODE  34
GAS(N,M) = NGAS                           DEKODE  35
IF (NGAS.EQ.HNU) GO TO 7                  DEKODE  36
IF (NO.EQ.0) GO TO 9                      DEKODE  37
DO 5 L = 1,NO                           DEKODE  38
IF (NGAS.NE.NAME(L)) GO TO 5            DEKODE  39
LABEL(N,M) = L                           DEKODE  40
GO TO 7                                       DEKODE  41
5 CONTINUE                                     DEKODE  42
9 NO = NO+1                                     DEKODE  43
NAME(NO) = NGAS                           DEKODE  44
LABEL(N,M) = NO                           DEKODE  45
7 M = M+MM                                     DEKODE  46
IF (MM.EQ.1) N = 0                         DEKODE  47
GO TO 2                                       DEKODE  48
99 NTYPE = NO                                DEKODE  49
DEKODE 50
DEKODE 51
C
C   GENERATE CHECKSUM IDENTIFIERS --
C
K1 = K2 = K1SQ = K2SQ = 0                     DEKODE 52
DO 8 L = 1,5                                    DEKODE 53
K1 = K1 + LABEL(L,1)                           DEKODE 54
K2 = K2 + LABEL(L,2)                           DEKODE 55
K1SQ = K1SQ + LABEL(L,1)*LABEL(L,1)           DEKODE 56
8 K2SQ = K2SQ + LABEL(L,2)*LABEL(L,2)           DEKODE 57
ENCODE (10,110,LHS) K1, K1SQ                 DEKODE 58
ENCODE (10,110,RHS) K2, K2SQ                 DEKODE 59
110 FORMAT (14,I6)                           DEKODE 60
DEKODE 61
DEKODE 62
DEKODE 63
DEKODE 64
DEKODE 65
DEKODE 66
C
RETURN
END

```

```

SUBROUTINE UPDATE (INFILE, NTAPE, NSCRTCH, LIST, DATE)          UPDATE    2
C ..... UPDATE 3
C ..... UPDATE 4
C ..... UPDATE 5
C ..... UPDATE 6
C ..... UPDATE 7
C ..... UPDATE 8
C ..... UPDATE 9
C ..... UPDATE 10
C ..... UPDATE 11
C ..... UPDATE 12
C ..... UPDATE 13
C ..... UPDATE 14
C ..... UPDATE 15
C ..... UPDATE 16
C ..... UPDATE 17
C ..... UPDATE 18
C ..... UPDATE 19
C ..... UPDATE 20
C ..... UPDATE 21
C ..... UPDATE 22
C ..... UPDATE 23
C ..... UPDATE 24
C ..... UPDATE 25
C ..... UPDATE 26
C ..... UPDATE 27
C ..... UPDATE 28
C ..... UPDATE 29
C ..... UPDATE 30
C ..... UPDATE 31
C ..... UPDATE 32
C ..... UPDATE 33
C ..... UPDATE 34
C ..... UPDATE 35
C ..... UPDATE 36
C ..... UPDATE 37
C ..... UPDATE 38
C ..... UPDATE 39
C ..... UPDATE 40
C ..... UPDATE 41
C ..... UPDATE 42
C ..... UPDATE 43
C ..... UPDATE 44
C ..... UPDATE 45
C ..... UPDATE 46
C ..... UPDATE 47
C ..... UPDATE 48
C ..... UPDATE 49
C ..... UPDATE 50
C ..... UPDATE 51
C ..... UPDATE 52
C ..... UPDATE 53
C ..... UPDATE 54
C ..... UPDATE 55
C ..... UPDATE 56
C ..... UPDATE 57
C ..... UPDATE 58

THIS SUBROUTINE SEARCHES TWO SOURCES--AN INPUT FILE TAPE *INFILE* AND/OR INPUT CARD DATA (IF MODIFY = TRUE)--TO GENERATE AN UPDATED FILE ON TAPE *NTAPE*, WHICH CONTAINS ALL OF THE DATA OF THE FILE *INFILE* MODIFIED WITH ADDITIONS OR REVISIONS DEFINED BY THE CARD DATA. THE FILE ON TAPE *NTAPE* CAN BE CATALOGUED AS A PERMANENT FILE, IF DESIRED, FOR FUTURE USE AS THE INPUT LIBRARY. THE FILE GENERATED ON NTAPE CONTAINS DATES OF ENTRY FOR ALL CROSS SECTIONS WHICH HAVE BEEN CATALOGUED. IF LIST = TRUE, THE CONTENTS OF THE UPDATED CROSS SECTION FILE *NTAPE* ARE PRINTED OUT.

DIMENSION IMAGE(8), KINETIC(60), NAME(100), LABEL(5,2), GAS(5,2)
INTEGER BLANK, LHS1, RHS1, LHS2, RHS2, DATE, GAS
LOGICAL LIST, MODIFY, ENDFILE

CALL SECOND (T0)
NTYPE = 0
NUMBERS = 10H1234567890
BLANK = 1H
REWIND NTAPE
ENDFILE = .FALSE.
INPUT = INFILE

GENERATE OR MODIFY ELECTRON CROSS SECTION DATA FILE --
MODIFY = .TRUE.
READ (5,100)
IF (EOF(5)) 10,20
10 MODIFY = .FALSE.
20 REWIND 5
REWIND INPUT
READ (INPUT,100)
IF (EOF(INPUT)) 1,2
1 INPUT = 0
GO TO 3
2 BACKSPACE INPUT
3 IF (MODIFY.AND.INPUT.EQ.INFILE) GO TO 33
NFILE = INPUT
IF (INPUT.EQ.INFILE) GO TO 46
NFILE = 5
IF (.NOT.MODIFY) RETURN

THE FOLLOWING SECTION IS USED WITH ONLY ONE INPUT DATA SOURCE --
46 READ (INFILE,120) IMAGE
IF (EOF(INFILE)) 45,4
4 IF (IMAGE(8).EQ.BLANK) IMAGE(8) = DATE
WRITE (NTAPE,120) IMAGE
READ (INFILE,120) IMAGE
WRITE (NTAPE,120) IMAGE
NREC = 0
47 READ (INFILE,120) IMAGE

```

```

        WRITE (NTAPE,120) IMAGE          UPDATE      59
        IF (IMAGE(1).NE.BLANK) GO TO 14   UPDATE      60
          IF (NREC.GT.0) GO TO 46       UPDATE      61
          BACKSPACE NTAPE             UPDATE      62
          BACKSPACE NTAPE             UPDATE      63
          BACKSPACE NTAPE             UPDATE      64
          GO TO 46                     UPDATE      65
14 NREC = 1                         UPDATE      66
        READ (INFILE,120) IMAGE         UPDATE      67
        WRITE (NTAPE,120) IMAGE         UPDATE      68
        GO TO 47                     UPDATE      69
C   THE FOLLOWING SECTION OCCURS WHEN DATA IS ASSEMBLED FROM BOTH A    UPDATE      70
C   TAPE AND CARD INPUT FILE --                                UPDATE      71
C   UPDATE      72
C   UPDATE      73
33 IF (ENDFILE) GO TO 44           UPDATE      74
        READ (INPUT,150) KINETIC       UPDATE      75
        IF (EOF(INPUT)) 6,7          UPDATE      76
7 CALL DEKODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60) UPDATE      77
        GO TO 37                     UPDATE      78
6 ENDFILE = .TRUE.                 UPDATE      79
        INPUT = 5                   UPDATE      80
        NFILE = 5                   UPDATE      81
        REWIND NFILE                UPDATE      82
44 READ (INFILE,120) IMAGE         UPDATE      83
        IF (EOF(INFILE)) 45,8        UPDATE      84
8 IMAGE(8) = DATE                 UPDATE      85
        BACKSPACE NFILE              UPDATE      86
        READ (INFILE,150) KINETIC       UPDATE      87
        CALL DEKODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60) UPDATE      88
C   CHECK TAPE3 TO DETERMINE WHETHER THE PROCESS ENCOUNTERED ON CARD    UPDATE      89
C   INPUT WAS PREVIOUSLY USED TO UPDATE TAPE FILE DATA --               UPDATE      90
C   UPDATE      91
C   UPDATE      92
        REWIND NSCRTCH                UPDATE      93
43 READ (NSCRTCH) LHS2, RHS2       UPDATE      94
        IF (EOF(NSCRTCH)) 34,5        UPDATE      95
5 IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 43                  UPDATE      96
        READ (INPUT,120)              UPDATE      97
        GO TO 41                     UPDATE      98
37 NFILE = INPUT                 UPDATE      99
        BACKSPACE NFILE              UPDATE     100
        READ (INFILE,120) IMAGE         UPDATE     101
C   CHECK CARDS TO SEE IF A CROSS SECTION PRESENTED ON THE TAPE FILE    UPDATE     102
C   SHOULD BE SUPERCEDED BY CARD INPUT DATA (UPDATE) --               UPDATE     103
C   UPDATE     104
C   UPDATE     105
        REWIND 5                      UPDATE     106
36 READ (5,150) KINETIC            UPDATE     107
        IF (EOF(5)) 34,9              UPDATE     108
9 CALL DEKODE (NAME, KINETIC, LHS2, RHS2, LABEL, GAS, 10, NTYPE, 60) UPDATE     109
        IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 35                  UPDATE     110
C   THE PROCESS DEFINED ON TAPE FILE HAS BEEN FOUND IN THE CARD INPUT. UPDATE     111
C   SO IT IS REPLACED --          UPDATE     112
C   UPDATE     113
C   UPDATE     114
        NFILE = 5                   UPDATE     115

```

BACKSPACE 5	UPDATE	116
READ (5,120) IMAGE	UPDATE	117
IMAGE(R) = DATE	UPDATE	118
WRITE (NSCRATCH) LHS2, RHS2	UPDATE	119
READ (INPUT,120)	UPDATE	120
GO TO 34	UPDATE	121
C	UPDATE	122
35 READ (5,120)	UPDATE	123
31 READ (5,120) KARD	UPDATE	124
IF (KARD.EQ.BLANK) GO TO 36	UPDATE	125
READ (5,120)	UPDATE	126
GO TO 31	UPDATE	127
C	UPDATE	128
COPY CROSS SECTION DATA ONTO TAPE NTAPE --	UPDATE	129
C	UPDATE	130
34 WRITE (NTAPE,120) IMAGE	UPDATE	131
READ (INFILE,120) IMAGE	UPDATE	132
WRITE (NTAPE,120) IMAGE	UPDATE	133
NREC = 0	UPDATE	134
32 READ (INFILE,120) IMAGE	UPDATE	135
WRITE (NTAPE,120) IMAGE	UPDATE	136
IF (IMAGE(1).NE.BLANK) GO TO 13	UPDATE	137
IF (NREC.GT.0) GO TO 39	UPDATE	138
BACKSPACE NTAPE	UPDATE	139
BACKSPACE NTAPE	UPDATE	140
BACKSPACE NTAPE	UPDATE	141
GO TO 39	UPDATE	142
13 NREC = 1	UPDATE	143
READ (INFILE,120) IMAGE	UPDATE	144
WRITE (NTAPE,120) IMAGE	UPDATE	145
GO TO 32	UPDATE	146
39 IF (INFILE.EQ.INPUT) GO TO 33	UPDATE	147
C	UPDATE	148
C EXHAUST OLD DATA FOR THIS PROCESS --	UPDATE	149
C	UPDATE	150
41 READ (INPUT,120) IMAGE	UPDATE	151
IF (IMAGE(1).EQ.BLANK) GO TO 33	UPDATE	152
READ (INPUT,120)	UPDATE	153
GO TO 41	UPDATE	154
C	UPDATE	155
C IF ELECTRUM CROSS SECTION DATA CONTAINED MODIFICATIONS BY CARD	UPDATE	156
C INPUT, A NEW FILE IS GENERATED. THE CONTENTS OF THE UPDATED FILE	UPDATE	157
C (WHICH MAY BE CATALOGUED FOR FUTURE USE) ARE COPIED ONTO OUTPUT	UPDATE	158
C IF LIST IS SPECIFIED TO BE TRUE.	UPDATE	159
C	UPDATE	160
45 ENDFILE NTAPE	UPDATE	161
REWIND NSCRATCH	UPDATE	162
CALL SECOND (TIME)	UPDATE	163
TIME = TIME-T0	UPDATE	164
IF (MODIFY) WRITE (6,160) TIME	UPDATE	165
IF (.NOT.LIST) GO TO 99	UPDATE	166
IF (INFILE.EQ.INFILE) GO TO 99	UPDATE	167
REWIND NTAPE	UPDATE	168
LL = LINE = 0	UPDATE	169
23 READ (NTAPE,120) IMAGE	UPDATE	170
IF (EOF(NTAPE)) 11,12	UPDATE	171
12 IF (LINE.NE.0) GO TO 25	UPDATE	172

```

IF (LL.EQ.0) GO TO 26          UPDATE    173
WRITE (6,190) (NUMBERS, I = 1,8), (I, I = 1,8)   UPDATE    174
WRITE (6,180)                   UPDATE    175
26 WRITE (6,170) NTAPE, (I, I = 1,8), (NUMBERS, I = 1,8) UPDATE    176
25 LINE = LINE+1               UPDATE    177
LL = LL+1                     UPDATE    178
WRITE (6,140) LL, IMAGE       UPDATE    179
IF (LINE.EQ.40) LINE = 0      UPDATE    180
GO TO 23                      UPDATE    181
11 WRITE (6,190) (NUMBERS, I = 1,8), (I, I = 1,8) UPDATE    182
C ----- FORMAT STATEMENTS -----
C
C 100 FORMAT (A10)             UPDATE    183
C 120 FORMAT (8A10)            UPDATE    184
C 140 FORMAT (20X,IS,5X+8A10)  UPDATE    185
C 150 FORMAT (80A1)            UPDATE    186
C
C 160 FORMAT (//33X+ELECTRON CROSS SECTION FILE WAS UPDATED. TIME REQU
1IRED WAS=F5.1* CP SEC.*//33X+CONSULT DAY FILE TO DETERMINE WHETHER
2THE UPDATED FILE WAS RECATALOGUED.*)
C
C 170 FORMAT (1H1,49X+ELECTRON CROSS SECTION DATA ON TAPE*.12//30X,
18I10/20X,* LINE*4X+8A10/20X+90(1H-)/) UPDATE    194
C
C 180 FORMAT (/20X+CONTINUED*)  UPDATE    195
C
C 190 FORMAT (/20X+90(1H-)/30X,8A10/30X+8I10) UPDATE    196
C
C ----- UPDATE    197
C 99 REWIND NTAPE              UPDATE    198
END                           UPDATE    199
                                UPDATE    200
                                UPDATE    201
                                UPDATE    202
                                UPDATE    203
                                UPDATE    204
C ----- UPDATE    205
C
C ----- UPDATE    206
C
C ----- UPDATE    207
C
C ----- UPDATE    208

```

```

SUBROUTINE PLASMA (NDATA, MAX, MESH, LHS, RHS, PROCESS, EV, F,
1 G, Q, U0, UM, NTYPE, NAME, MISSING, ERROR, OUTSIDE, IDEG, OUT)          PLASMA      2
C-----PLASMA      3
C-----PLASMA      4
C-----PLASMA      5
C-----PLASMA      6
C-----PLASMA      7
C-----PLASMA      8
C-----PLASMA      9
C-----PLASMA     10
C-----PLASMA     11
C-----PLASMA     12
C-----PLASMA     13
C-----PLASMA     14
C-----PLASMA     15
C-----PLASMA     16
C-----PLASMA     17
C-----PLASMA     18
C-----PLASMA     19
C-----PLASMA     20
C-----PLASMA     21
C-----PLASMA     22
C-----PLASMA     23
C-----PLASMA     24
C-----PLASMA     25
C-----PLASMA     26
C-----PLASMA     27
C-----PLASMA     28
C-----PLASMA     29
C-----PLASMA     30
C-----PLASMA     31
C-----PLASMA     32
C-----PLASMA     33
C-----PLASMA     34
C-----PLASMA     35
C-----PLASMA     36
C-----PLASMA     37
C-----PLASMA     38
C-----PLASMA     39
C-----PLASMA     40
C-----PLASMA     41
C-----PLASMA     42
C-----PLASMA     43
C-----PLASMA     44
C-----PLASMA     45
C-----PLASMA     46
C-----PLASMA     47
C-----PLASMA     48
C-----PLASMA     49
C-----PLASMA     50
C-----PLASMA     51
C-----PLASMA     52
C-----PLASMA     53
C-----PLASMA     54
C-----PLASMA     55
C-----PLASMA     56
C-----PLASMA     57
C-----PLASMA     58

C THIS SUBROUTINE SCANS THE ELECTRON CROSS SECTION FILE TO EXTRACT
C DATA FOR THE INPUT REACTION "PROCESS", DEFINED BY "LHS" AND "RHS".
C IF THE REACTION IS FOUND, THE RAW CROSS SECTION DATA IS EXAMINED
C FOR ERRORS, AND IF ACCEPTABLE, IS INTERPOLATED OVER THE INPUT EN-
C ERGY GRID DEFINED BY THE VECTOR EV(I), I = 1..MESH+1. THE EXTERNAL
C ELECTRON FILE CONSISTS OF (ARBITRARILY MANY) PACKAGES OF THE FORM
C
C   A) REACTION, UNITS, NPTS, MONTH (60A1,F7.3,I3,A10)
C   B) COMMENT (6A10)
C
C   1) ENERGY VALUES (EV)
C   2) CROSS SECTION VALUES
C
C ARBITRARY NUMBER OF CARD PAIRS (1) AND (2), TERMINATED
C BY THE BLANK CARD (C) BELOW. THE FORMAT IS VARIABLE?
C THERE ARE *NPTS* FIELDS FK.0, WHERE K = 180/NPTS).
C IF NPTS ≤ 0 OR NPTS > 10, PROGRAM DEFAULTS TO NPTS = 10.
C THE UNITS OF THE CROSS SECTION DATA ARE UNITS X 1.0E-16
C CM2 (DEFAULT: UNITS = 1).
C
C C) BLANK CARD
C
C ERROR CONDITIONS ENCOUNTERED ARE IDENTIFIED BY LOGICAL VARIABLES
C WHICH ARE RETURNED WITH THE VALUE .TRUE. TO THE CALLING PROGRAM.
C
C INPUT PARAMETERS --
C
C NDATA = LOGICAL FILE FOR ELECTRON CROSS SECTION DATA.          PLASMA      35
C
C MAX = DIMENSION DECLARATOR DEFINED FOR EV(I), F(I), G(I),
C       AND Q(I) IN THE CALLING PROGRAM.          PLASMA      36
C
C MESH = NUMBER OF SUBINTERVALS INTO WHICH THE ELECTRON ENERGY          PLASMA      37
C       RANGE IS DIVIDED (AND OVER WHICH THE CROSS SECTION DATA          PLASMA      38
C       IS TO BE INTERPOLATED). MESH+1 ≤ MAX.          PLASMA      39
C
C PROCESS = VECTOR (4A10) CONTAINING HOLLERITH NAME OF REACTION.          PLASMA      40
C
C LHS,RHS = (INTEGERS) ENCODED WITH UNIQUE IDENTIFIERS OF THE LEFT          PLASMA      41
C       AND RIGHT HAND SIDE OF THE REACTION (CF. SUBROUTINE          PLASMA      42
C       DEKODE).
C
C F, G = SCRATCH VECTORS (DIMENSIONED MAX IN CALLING PROGRAM.)          PLASMA      43
C
C OUTPUT PARAMETERS --
C
C Q(I) = CROSS SECTION VECTOR (UNITS OF CM2) DEFINED AT EV(I)          PLASMA      44
C
C U0 = CROSS SECTION THRESHOLD ENERGY (EV).          PLASMA      45

```

```

C          UM      = CROSS SECTION MAXIMUM ENERGY CUTOFF (EV).           PLASMA    59
C          MISSING = .TRUE., IF THE REACTION WAS NOT FOUND.             PLASMA    60
C          OUTSIDE = .TRUE., IF THE CROSS SECTIONS WERE DEFINED OVER AN PLASMA    61
C                      ENERGY RANGE [U0+UM] THAT DOES NOT SPAN THE ENERGY GRID PLASMA    62
C                      DEFINED BY THE INPUT ENERGY VECTOR EV(I).            PLASMA    63
C          ERROR   = .TRUE., IF ENERGY VALUES WERE NOT SEQUENCED IN MONO- PLASMA    64
C                      TONICALLY ASCENDING ORDER (ONLY DATA IN ASCENDING ORDER PLASMA    65
C                      IS PERMITTED.)                           PLASMA    66
C          -----
C          DIMENSION EV(1), F(1), G(1), Q(1), NAME(1), IMAGE(60), KOMMENT(6), PLASMA    67
C          1 PROCESS(4), Y(10), LABEL(5,2), GAS(5,2)                         PLASMA    68
C          LOGICAL FORWARD, REVERSE, ERROR, THRESH, OUT(2), MISSING, OUTSIDE PLASMA    69
C          INTEGER LHS, RHS, GAS                                         PLASMA    70
C          PLASMA    71
C          PLASMA    72
C          PLASMA    73
C          PLASMA    74
C          PLASMA    75
C          PLASMA    76
C          PLASMA    77
C          PLASMA    78
C          PLASMA    79
C          PLASMA    80
C          PLASMA    81
C          PLASMA    82
C          PLASMA    83
C          PLASMA    84
C          PLASMA    85
C          PLASMA    86
C          PLASMA    87
C          PLASMA    88
C          PLASMA    89
C          PLASMA    90
C          PLASMA    91
C          PLASMA    92
C          PLASMA    93
C          PLASMA    94
C          PLASMA    95
C          PLASMA    96
C          PLASMA    97
C          PLASMA    98
C          PLASMA    99
C          PLASMA   100
C          PLASMA   101
C          PLASMA   102
C          PLASMA   103
C          PLASMA   104
C          PLASMA   105
C          PLASMA   106
C          PLASMA   107
C          PLASMA   108
C          PLASMA   109
C          PLASMA   110
C          PLASMA   111
C          PLASMA   112
C          PLASMA   113
C          PLASMA   114
C          PLASMA   115
C
C          UNIT = 1.E-16
C          NPAGE = SH(IH1)
C          ERROR = OUTSIDE = .FALSE.
C          MESHP1 = MESH+1
C          DO 17 L = 1,MESHP1
C          17 Q(L) = 0.
C
C          REWIND NDATA
C          20 READ (NDATA,100) IMAGE, UNITS, NPTS, MONTH
C          100 FORMAT (60A1,F7.3,I3,A10)
C          IF (EOF(NDATA)) 99,1
C          1 CALL DEKODE (NAME, IMAGE, L1, L2, LABEL, GAS, 10, NTYPE, 60)
C          FORWARD = LHS.EQ.L1.AND.RHS.EQ.L2
C          REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1
C          MISSING = .NOT.(FORWARD.OR.REVERSE)
C          IF ((NPTS.LE.0).OR.(NPTS.GT.10)) NPTS = 10
C          INV = 80/NPTS
C          ENCODE (10,101,FORM) NPTS, INV
C          101 FORMAT (I1H,I2,I1H,I2,3H.0)
C          IF (UNITS.EQ.0,) UNITS = 1.
C          IF (MISSING) GO TO 3
C          THRESH = .FALSE.
C          THRESH = FALSE AT START OF CROSS SECTION DATA FOR PROCESS J, AND
C          BECOMES TRUE AS SOON AS THE FIRST NON-ZERO VALUE APPEARS.
C          DO 4 L = 1,MAX
C          4 F(L) = G(L) = 0.
C          LPTS = LTH = 1
C          F0 = U0 = 0.
C          X0 = - 1.0
C          LAST = MAX
C
C          3 READ (NDATA,112) KOMMENT
C          112 FORMAT (8A10)
C
C          ISW = 1
C          L = 0

```

```

R L2 = L          PLASMA 116
7 ISW = - ISW    PLASMA 117
READ (INDATA,FORM) (Y(N), N = 1,NPTS)  PLASMA 118
SUM = 0.          PLASMA 119
DO 16 N = 1,NPTS  PLASMA 120
16 SUM = SUM + Y(N)  PLASMA 121
C               PLASMA 122
C               BLANK CARD TERMINATES (U, SIGMA) DATA PACKAGE FOR THE INELASTIC  PLASMA 123
C               PROCESSES NK.  PLASMA 124
C               PLASMA 125
IF ((SUM.EQ.0.).AND.(ISW.LT.0)) GO TO 10  PLASMA 126
IF (MISSING) GO TO 7  PLASMA 127
IF (L2.EQ.LAST) GO TO 7  PLASMA 128
IF (ISW.GT.0) GO TO 9  PLASMA 129
C               PLASMA 130
L = L2          PLASMA 131
DO 12 N = 1,NPTS  PLASMA 132
IF (L.EQ.LAST) GO TO 7  PLASMA 133
IF (Y(N).GT.X0) GO TO 11  PLASMA 134
IF (Y(N).NE.0.) ERROR = .TRUE.  PLASMA 135
LAST = L          PLASMA 136
GO TO 7          PLASMA 137
11 L = L+1        PLASMA 138
12 X0 = F(L) = Y(N)  PLASMA 139
GO TO 7          PLASMA 140
C               PLASMA 141
9 L = L2          PLASMA 142
DO 13 N = 1,NPTS  PLASMA 143
IF (L.EQ.LAST) GO TO 8  PLASMA 144
L = L+1          PLASMA 145
G(L) = Y(N)*UNITS  PLASMA 146
IF (G(L).GT.0.) THRESH = .TRUE.  PLASMA 147
IF (THRESH) GO TO 5  PLASMA 148
U0 = F(L)          PLASMA 149
LTH = L          PLASMA 150
5 IF (F0.EQ.0.) GO TO 13  PLASMA 151
UM = F(L)          PLASMA 152
LPTS = L          PLASMA 153
13 F0 = G(L)        PLASMA 154
GO TO 8          PLASMA 155
C               PLASMA 156
10 IF (MISSING) GO TO 20  PLASMA 157
LPTS = LPTS - LTH + 1  PLASMA 158
ERROR = ERROR.OR.LPTS.LE.1  PLASMA 159
IF (ERROR) GO TO 6  PLASMA 160
DO 14 L = 1,LPTS  PLASMA 161
LL = L + LTH - 1  PLASMA 162
G(L) = G(LL)        PLASMA 163
14 F(L) = F(LL)        PLASMA 164
C               PLASMA 165
OUTSIDE = .TRUE.  PLASMA 166
DO 15 L = 1,MESHPI  PLASMA 167
SIGMA = 0.          PLASMA 168
X = EV(L)          PLASMA 169
IF (X.GT.UM) GO TO 15  PLASMA 170
IF (X.LT.U0) GO TO 15  PLASMA 171
OUTSIDE = .FALSE.  PLASMA 172

```

```

      CALL INTERP (IDEGR, X, SIGMA, F, G, 1, LPTS)          PLASMA    173
15 IF (SIGMA.GE.0.) Q(L) = SIGMA*UNIT                  PLASMA    174
C
6 IF (.NOT.OUT(1)) GO TO 18                          PLASMA    175
      WRITE (6,102) UNIT, PROCESS                      PLASMA    176
102 FORMAT (1H1,34X,*CROSS SECTION (UNITS OF*, 1PE10.3* CM2) VS ELECTR PLASMA    177
      ION ENERGY (EV; FOR*/57X,4A10//)
      IF (KOMMENT(1).NE.1H) WRITE (6,103) KOMMENT        PLASMA    178
103 FORMAT (35X,*REFERENCE -- *,6A10//)
      WRITE (6,104) UNIT                  PLASMA    179
104 FORMAT (56X*U*15X*SIGMA(U)*/54X*(EV) *10X*(1PE7.1* CM2)*)
      WRITE (6,105) UNIT                  PLASMA    180
105 FORMAT (/50X,33(1H-)/)
      LL = 0
      LINE = 1
      DO 33 L = 1,LPTS
      IF (LL.NE.LINE) GO TO 32
      WRITE (6,N*AGE)
      WRITE (6,104) UNIT
      WRITE (6,105)
32 LINE = LINE+1
      WRITE (6,106) F(L), G(L)
106 FORMAT (50X,F8.3,1PE22.3)
      LL = 40*(LINE/40)
33 IF (LL.EQ.LINE) WRITE (6,105)
      IF (LL.NE.LINE) WRITE (6,105)
      WRITE (6,108) MONTH
108 FORMAT (/50X,* (DATA WAS SUBMITTED ON*,A9,*))
C
C     IF IOUT(2) IS SPECIFIED, DATA IS PLOTTED --
C
18 IF (.NOT.OUT(2)) GO TO 99
      IF (ERROR) GO TO 99
      IPLOT = 0
      IF (F(LPTS).GT.50.*F(1)) IPLOT = -1
      WRITE (6,102) UNIT, PROCESS
      CALL PLOT (MAX, LPTS, 1, G, 0., 0., F, 0., 0., .TRUE., .TRUE.,
      1,.TRUE.,.TRUE.,.TRUE., TITLE, 1, IPLOT)
      IF (IPLOT.EQ.0) WRITE (6,109)
109 FORMAT (/64X,*ELECTRON ENERGY U (EV)*)
      IF (IPLOT.NE.0) WRITE (6,110)
110 FORMAT (/60X,*LOG OF ELECTRON ENERGY U (EV)*)
C
99 RETURN
END

```

```

FUNCTION SHAPE(T)
DIMENSION TIME(21), Y(21)
LOGICAL INTRP
COMMON / TIMES / TR, TF, TFAILL, TDROP, TIME, Y, INTRP, N, UNITS
TP = T/UNITS
IF (INTRP) GO TO 2
SHAPE = 0.
IF (T.LT.0.) RETURN
SHAPE = 1.0
IF (TR.EQ.0.) RETURN
X = TR/TF
FO = X*(1. + 1./X)**(1. + X)
IF (TP.GT.TDROP) GO TO 1
SHAPE = FO*(1. - EXP(-TP/TR))*EXP(-TP/TF)
RETURN
1 TP = TP-TDROP
FO = FO*(1. - EXP(-TDROP/TR))*EXP(-TDROP/TF)
SHAPE = FO*EXP(-TP/TFAILL)
RETURN
2 CALL INTERP (2, TP, SHAPE, TIME, Y, 1, N)
IF (SHAPE.LT.0.) SHAPE = 0.
RETURN
END

```

SHAPE	2
SHAPE	3
SHAPE	4
SHAPE	5
SHAPE	6
SHAPE	7
SHAPE	8
SHAPE	9
SHAPE	10
SHAPE	11
SHAPE	12
SHAPE	13
SHAPE	14
SHAPE	15
SHAPE	16
SHAPE	17
SHAPE	18
SHAPE	19
SHAPE	20
SHAPE	21
SHAPE	22
SHAPE	23
SHAPE	24

```

SUBROUTINE BOLTZ (MAX, MESH, NK, GAS, FRACT, MIX, NMOL, THOL,      BOLTZ    2
1 ITMAX, TMAX, EPS, KAPTION, DATE, OUT, EVCM, NE, PROCESS, U, N1,   BOLTZ    3
2 N2, NEL, S, SBEAM, SOURCE, X, XO, QM, F, G, A, B, VSIG, POWER,   BOLTZ    4
3 PCOLL, DISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE,   BOLTZ    5
4 ATTACH, VD, MU, D, EK, AMPS, UBAR, TE, CONVRGE, PBAL)           BOLTZ    6
C -----
C -----
C THIS SUBROUTINE PERFORMS A NUMERICAL SOLUTION OF THE BOLTZMANN TRANS- BOLTZ    7
C PORT EQUATION FOR A MULTICOMPONENT GAS, WITH THE INCLUSION OF INELAS- BOLTZ    8
C TIC E-MOLECULE COLLISIONS, ELASTIC MOMENTUM TRANSFER COLLISIONS (WITH BOLTZ    9
C RECOIL), SUPERELASTIC COLLISIONS, ELECTRON-ELECTRON (COULOMB) SCAT- BOLTZ   10
C TERING, AND EXTERNAL ELECTRON ENERGY DEPOSITION. THE ANALYSIS CON- BOLTZ   11
C SISTS OF CALCULATION OF THE ELECTRON ENERGY DISTRIBUTION F(U) FOR THE BOLTZ   12
C ELECTRONS IN A PLASMA WITH IONIZATION NE/NTOT (OR SINGLE ELECTRON IF BOLTZ   13
C NE = 0), SUBJECTED TO A (SPATIALLY AND TEMPORALLY) CONSTANT ELECTRIC BOLTZ   14
C FIELD. THIS SUBROUTINE CALCULATES THE ELECTRON ENERGY DISTRIBUTION BOLTZ   15
C FUNCTION, PLASMA PARAMETERS (MU, D, VD, EK, UBAR, TE, ...), ELECTRON BOLTZ   16
C EXCITATION RATES (CM3/SEC) FOR THE FORWARD (AND REVERSE) INELASTIC BOLTZ   17
C COLLISION PROCESSES J = 1,2,...,NK, AND THE (NET) ELECTRICAL POWER BOLTZ   18
C PARTITIONING FOR ALL OF THESE PROCESSES (AND FOR ELASTIC HEATING). BOLTZ   19
C
C INPUT PARAMETERS --
C
C MAX      = DIMENSION DECLARATOR DEFINED FOR VARIOUS VECTORS      BOLTZ   20
C AND ARRAYS IN THE CALLING PROGRAM.                                BOLTZ   21
C
C MESH     = NUMBER OF SUBDIVISIONS INTO WHICH THE ELECTRON          BOLTZ   22
C ENERGY RANGE (0,EMAX) IS PARTITIONED.                            BOLTZ   23
C
C NK       = NUMBER OF INELASTIC E(-) COLLISION PROCESSES          BOLTZ   24
C INCLUDED IN THE PLASMA KINETICS ANALYSIS.                         BOLTZ   25
C
C X(I)     = (I - 1)DX, ELECTRON ENERGY GRID (EV), WHERE            BOLTZ   26
C DX = EMAX/MESH.                                              BOLTZ   27
C
C XQ(I,J)  = X(I)Q(I,J), WHERE Q(I,J) = INELASTIC SCATTERING      BOLTZ   28
C CROSS SECTION (CM2) FOR THE JTH INELASTIC ELECTRON                BOLTZ   29
C COLLISION PROCESS, AT ENERGY X(I).                                BOLTZ   30
C
C QM(I,L)  = ARRAY CONTAINING TWO COLUMN VECTORS, CONVENIENT      BOLTZ   31
C FOR THE BOLTZMANN ANALYSIS --                                     BOLTZ   32
C
C QM(I+1) = X(I+.5)/NTOT SUMK(FI(K)*QMOM(I+.5,K))               BOLTZ   33
C
C QM(I+2) = X(I+.5)**2*2*ME*NTOT*                                     BOLTZ   34
C             SUMK(FI(K)*QMOM(I+.5,K)/MASS(K))                      BOLTZ   35
C
C WHERE QMOM(I,K) = MOMENTUM TRANSFER CROSS SECTION                 BOLTZ   36
C AT ENERGY X(I) FOR SPECIES K.                                    BOLTZ   37
C
C F(I)     = F(X(I)), INITIAL GUESS FOR THE ELECTRON ENERGY        BOLTZ   38
C DISTRIBUTION FUNCTION, DIMENSIONED F(MAX) IN THE                  BOLTZ   39
C CALLING PROGRAM.                                         BOLTZ   40
C
C U(J)     = ENERGY LOSS (EV) FOR THE JTH INELASTIC ELECTRON      BOLTZ   41
C PROCESS.                                                 BOLTZ   42
C
C

```

C	N1(J)	= NEUTRAL PARTICLE DENSITY (CM-3) OF THE LOWER STATE IN THE JTH INELASTIC COLLISION PROCESS.	BOLTZ	59
C			BOLTZ	60
C			BOLTZ	61
C	N2(J)	= NEUTRAL PARTICLE DENSITY (CM-3) OF THE UPPER STATE IN THE JTH INELASTIC COLLISION PROCESS.	BOLTZ	62
C			BOLTZ	63
C	NEL(J)	= NET NUMBER OF ELECTRONS (RIGHT HAND SIDE - LEFT HAND SIDE) FOR THE JTH ELECTRON COLLISION PROCESS.	BOLTZ	64
C			BOLTZ	65
C	NMOL	= TOTAL NEUTRAL PARTICLE DENSITY (CM-3).	BOLTZ	66
C			BOLTZ	67
C	NE	= ELECTRON DENSITY (CM-3).	BOLTZ	68
C			BOLTZ	69
C	DNEDT	= INPUT ESTIMATE OF D/DT(NE), THE RATE OF CHANGE OF THE SECONDARY ELECTRON DENSITY (CM-3/SEC). THE BOLTZMANN ANALYSIS DETERMINES D/DT(NE) SELF-CONSISTENTLY IN AN ITERATIVE LOOP, AND EXECUTION MAY BE OPTIMIZED BY A GOOD INITIAL ESTIMATE OF DNEDT. UPON OUTPUT, DNEDT IS THE ACTUAL (SELF-CONSISTENT) VALUE OF D/DT(NE).	BOLTZ	70
C			BOLTZ	71
C			BOLTZ	72
C	SOURCE	= TOTAL RATE OF EXTERNAL SOURCE CREATION OF SECONDARY ELECTRONS AT U = 0 (CM-3/SEC).	BOLTZ	73
C			BOLTZ	74
C			BOLTZ	75
C	SBEAM	= TOTAL RATE OF E-BEAM CREATION OF SECONDARY ELECTRONS (CM-3/SEC).	BOLTZ	76
C			BOLTZ	77
C			BOLTZ	78
C	S(I)	= NORMALIZED E-BEAM SOURCE FUNCTION: SBEAM*S(I) = RATE OF GENERATION OF ELECTRONS (/SEC/CM3/EV) IN THE ENERGY RANGE [U,U+DU], WHERE U = X(I). THE TOTAL EXTERNAL POWER DEPOSITION IS: DEPOSIT = SBEAM*INT(DU U S(U)) (EV/SEC/CM3).	BOLTZ	79
C			BOLTZ	80
C			BOLTZ	81
C			BOLTZ	82
C			BOLTZ	83
C	GAS(K)	= NAME OF THE KTH GAS IN THE MIXTURE.	BOLTZ	84
C			BOLTZ	85
C			BOLTZ	86
C	FRACT(K)	= RELATIVE FRACTION OF THE KTH GAS IN THE MIXTURE. SUMK(FRACT(K)) NEED NOT EQUAL 1; THE PROGRAM WILL AUTOMATICALLY NORMALIZE THE CONCENTRATIONS, AFTER SORTING THEM INTO DESCENDING NUMERICAL ORDER.	BOLTZ	87
C			BOLTZ	88
C			BOLTZ	89
C			BOLTZ	90
C			BOLTZ	91
C			BOLTZ	92
C	PROCESS	= ARRAY (DIMENSIONED (4,1)) CONTAINING 4 WORDS PER COLUMN TO PROVIDE A 40-BCD CHARACTER SPECIFICATION FOR EACH OF THE INELASTIC E- COLLISION PROCESSES.	BOLTZ	93
C			BOLTZ	94
C			BOLTZ	95
C	MIX	= NUMBER OF SPECIES IN THE MIXTURE.	BOLTZ	96
C			BOLTZ	97
C	EVCM	= ELECTRIC FIELD (V/CM).	BOLTZ	98
C			BOLTZ	99
C	TMOL	= MOLECULAR TEMPERATURE (DEG K).	BOLTZ	100
C			BOLTZ	101
C	A(I,J)	= SCRATCH ARRAY, DIMENSIONED A(MAX,3) IN THE CALLING PROGRAM, USED FOR STORING CERTAIN COEFFICIENTS THAT DEFINE THE FINITE DIFFERENCE FORM OF THE BOLTZMANN EQUATION (A(I,1), A(I,2), AND A(I,3) ARE USED TO STORE THE THREE MAJOR DIAGONALS DURING THE SOLUTION). LATER, A(I,J) IS USED AS A SCRATCH	BOLTZ	102
C			BOLTZ	103
C			BOLTZ	104
C			BOLTZ	105
C			BOLTZ	106
C			BOLTZ	107
C			BOLTZ	108
C			BOLTZ	109
C			BOLTZ	110
C			BOLTZ	111
C			BOLTZ	112
C			BOLTZ	113
C			BOLTZ	114
C			BOLTZ	115

	ARRAY FOR CALCULATING NUMEROUS PLASMA PARAMETERS.	BOLTZ	116
C		BOLTZ	117
C	B(I) = SCRATCH VECTOR, DIMENSIONED B(MAX) IN THE CALLING	BOLTZ	118
C	PROGRAM, USED FOR THE RIGHT HAND SIDE IN THE	BOLTZ	119
C	ITERATIVE SOLUTION OF THE BOLTZMANN EQUATION.	BOLTZ	120
C		BOLTZ	121
C	EPS = MAXIMUM CHANGE BETWEEN SUCCESSIVE ITERATIONS FOR	BOLTZ	122
C	DEFINITION OF CONVERGENCE FOR THE ELECTRON ENERGY	BOLTZ	123
C	DISTRIBUTION FUNCTION. CONVERGENCE IS DECLARED IF	BOLTZ	124
C	$\text{MAX}(I)(F(I(N)-F(I(N-1))/F(I,N-1)) < \text{EPS}$	BOLTZ	125
C		BOLTZ	126
C	ITMAX = MAXIMUM NUMBER OF ITERATIONS ALLOWED TO SOLVE THE	BOLTZ	127
C	BOLTZMANN EQUATION. (IF ITMAX ≤, PROGRAM SETS	BOLTZ	128
C	CONVRGE = TRUE, AND IMMEDIATELY TRANSFERS CONTROL	BOLTZ	129
C	TO CALCULATION OF PLASMA PARAMETERS FOR THE INPUT	BOLTZ	130
C	DISTRIBUTION FUNCTION F.	BOLTZ	131
C		BOLTZ	132
C	TMAX = MAXIMUM CP TIME ALLOWED FOR ATTAINMENT OF CONVER-	BOLTZ	133
C	GENCE OF THE ELECTRON ENERGY DISTRIBUTION FUNCTION	BOLTZ	134
C		BOLTZ	135
C	OUT(I) = LOGICAL VECTOR OF OUTPUT REQUESTS (I = 1,2,...,5):	BOLTZ	136
C		BOLTZ	137
C	1) TABLE OF PLASMA PARAMETERS	BOLTZ	138
C	2) TABLE OF ELECTRON ENERGY DISTRIBUTION	BOLTZ	139
C	3) LOGPLOT OF ENERGY DISTRIBUTION F(U)	BOLTZ	140
C	4) LOGPLOT OF RELATIVE FUNCTION F(U)/FB(U,TE)	BOLTZ	141
C	(FB(U,TE) = BOLTZMANN DISTRIBUTION AT TE.)	BOLTZ	142
C	5) TABLE OF EXCITATION RATES AND ELECTRICAL	BOLTZ	143
C	POWER TRANSFER FOR ALL E(-) PROCESSES.	BOLTZ	144
C		BOLTZ	145
C		BOLTZ	146
C	KAPTION(4) = VECTOR CONTAINING A 40 BCD CHARACTER TITLE	BOLTZ	147
C		BOLTZ	148
C	-----	BOLTZ	149
C	OUTPUT PARAMETERS --	BOLTZ	150
C		BOLTZ	151
C		BOLTZ	152
C	VSIG(K,J) = FORWARD (K = 1) AND REVERSE (K = 2) ELECTRON EX-	BOLTZ	153
C	CITATION RATES (CM <sup>3</sup> /SEC) FOR THE JTH INELASTIC	BOLTZ	154
C	COLLISION PROCESS.	BOLTZ	155
C		BOLTZ	156
C	VSIG(1,J) = $\langle V(U)Q(U, J) \rangle$	BOLTZ	157
C	VSIG(2,J) = $\langle V(U)Q(U,-J) \rangle$	BOLTZ	158
C		BOLTZ	159
C	WHERE Q(U,-J) IS THE CROSS SECTION FOR THE REVERSE	BOLTZ	160
C	PROCESS.	BOLTZ	161
C		BOLTZ	162
C	POWER(J) = NET POWER (ACCOUNTING FOR FORWARD AND REVERSE COL-	BOLTZ	163
C	LISION PROCESSES) PARTITIONED INTO THE JTH INELAS-	BOLTZ	164
C	TIC PROCESS. (WATT/ELECTRON)	BOLTZ	165
C		BOLTZ	166
C	VD = ELECTRON DRIFT VELOCITY (CM/S).	BOLTZ	167
C		BOLTZ	168
C	MU = ELECTRON MOBILITY (CM <sup>2</sup> /V/SEC).	BOLTZ	169
C		BOLTZ	170
C	D = ELECTRON DIFFUSION COEFFICIENT (CM <sup>2</sup> /SEC).	BOLTZ	171
C		BOLTZ	172

C	EK	= CHARACTERISTIC ELECTRON ENERGY = D/MU (EV).	BOLTZ	173
C	UBAR	= AVERAGE ELECTRON ENERGY (EV).	BOLTZ	174
C	TE	= EFFECTIVE ELECTRON TEMPERATURE (DEG K). DEFINED BY (3/2)KTE = UBAR	BOLTZ	175
C	DISCH	= E*NE*MU*EVCM**2, TOTAL ELECTRICAL DISCHARGE POWER DENSITY (W/CM3).	BOLTZ	176
C	DEPOSIT	= POWER DENSITY FROM EXTERNAL DEPOSITION (INTO ELECTRON KINETIC ENERGY) (W/CM3).	BOLTZ	177
C	PCOLL	= ELECTRICAL POWER DENSITY INTO INELASTIC COLLISIONS (W/CM3).	BOLTZ	178
C	ELASTIC	= ELECTRICAL POWER DENSITY INTO ELASTIC HEATING OF THE MOLECULAR GAS (W/CM3).	BOLTZ	179
C	DEDT	= UBAR*ONE/DT = RATE OF CHANGE OF ENERGY DENSITY STORED IN THE ELECTRON GAS (W/CM3).	BOLTZ	180
C	NOTE:	IF NE = 0, THE FIVE POWER DENSITIES (DISCH, DE- POSIT, PCOLL, ELASTIC, AND DEDT) ARE EVALUATED FOR UNIT ELECTRON DENSITY (I.E. AS IF NE = 1.0 CM-3). AND THUS, THE EFFECTIVE UNITS ARE W/ELECTRON.	BOLTZ	181
C	DNEDT	= D/DT(NE), RATE OF CHANGE OF SECONDARY ELECTRON DENSITY (CM-3/SEC).	BOLTZ	182
C	IONIZE	= TOTAL IONIZATION FREQUENCY (SEC-1).	BOLTZ	183
C	ATTACH	= TOTAL FREQUENCY FOR ATTACHMENT AND RECOMBINATION.	BOLTZ	184
C	DLNEDT	= (1/NE)*D/DT(NE), LOGARITHMIC DERIVATIVE OF NE.	BOLTZ	185
C	AMPS	= E*VD = ELECTRICAL CURRENT DENSITY PER UNIT ELECTRON DENSITY (A CM).	BOLTZ	186
C	F(I)	= NORMALIZED ELECTRON ENERGY DISTRIBUTION FUNCTION (UNITS OF EV**(-3/2)).	BOLTZ	187
C	G(I)	= F(I)/F(1)	BOLTZ	188
C	B(I)	= F(I)/FBOLTZ(TE,I)	BOLTZ	189
C	CONVRGE	= LOGICAL VARIABLE WHICH SPECIFIES SUCCESSFUL CON- VERGENCE OF THE ELECTRON DISTRIBUTION FUNCTION CALCULATIONS.	BOLTZ	190
C	-----			
C	1	DIMENSION XQ(MAX,1), QM(MAX,1), X(1), F(1), G(1), VSIG(2,1), U(1), N1(1), N2(1), NEL(1), POWER(1), A(MAX,3), PROCESS(4,1),	BOLTZ	191
C	2	B(1), OUT(1), GAS(1), FRACT(1), NAME(5), FI(5), FORM(15),	BOLTZ	192
C	3	KAPTION(4), NOUT(6), S(1), PCT(6)	BOLTZ	193
C			BOLTZ	194
C			BOLTZ	195
C			BOLTZ	196
C			BOLTZ	197
C			BOLTZ	198
C			BOLTZ	199
C			BOLTZ	200
C			BOLTZ	201
C			BOLTZ	202
C			BOLTZ	203
C			BOLTZ	204
C			BOLTZ	205
C			BOLTZ	206
C			BOLTZ	207
C			BOLTZ	208
C			BOLTZ	209
C			BOLTZ	210
C			BOLTZ	211
C			BOLTZ	212
C			BOLTZ	213
C			BOLTZ	214
C			BOLTZ	215
C			BOLTZ	216
C			BOLTZ	217
C			BOLTZ	218
C			BOLTZ	219
C			BOLTZ	220
C			BOLTZ	221
C			BOLTZ	222
C			BOLTZ	223
C			BOLTZ	224
C			BOLTZ	225
C			BOLTZ	226
C			BOLTZ	227
C			BOLTZ	228
C			BOLTZ	229



```

DLNEDT = 0.
IF (NE,NE.0.) DLNEDT = DNEDT/NE
C = 0.
IF (NE,NE.0.) C = 1./NE
SB = C*SBEAM
S0 = C*SOURCE
SEXT = S0 + SB
C = SB/EM
TEST = SB*NE.0.
CONVRGE = .TRUE.
DO 31 J = 1,NK
31 TEST = TEST.OR.(NEL(J).NE.0)
TEST = TEST.AND.(DLNEDT.EQ.0.)
ELECT = 0.
ITER = 0
TIME = 0.
C INITIAL (COARSE) SOLUTION CONVERGENCE PARAMETER --
EPSILON = 0.01
C
C NORMALIZE THE INPUT DISTRIBUTION FUNCTION --
DO 13 I = 1,M
13 B(I) = SQRT(X(I))*F(I)
CALL SIMPSON (B, MESH/2, DX, FNORM)
DO 14 I = 1,M
A(I,I) = X(I)*S(I)
G(I) = F(I)
14 F(I) = F(I)/FNORM
C
CALL SIMPSON (A, MESH/2, DX, UPLUS)
DEPOSIT = C0*SB*UPLUS
C
C IF ITMAX < 0, THE PROGRAM AUTOMATICALLY SUPPRESSES THE BOLTZMANN
C SOLUTION, AND TRANSFERS DIRECTLY TO CALCULATION OF PLASMA PARA-
C METERS AND EXCITATION RATES BASED UPON THE INPUT CUNCTION F(I).
C
C IF (ITMAX.LE.0) GO TO 65
C
C GO TO CALCULATE INITIAL APPROXIMATION TO D/DT(LN NE) --
IF (TEST) GO TO 65
C
C -----
C THE BOLTZMANN EQUATION IS REDUCED TO A FINITE DIFFERENCE SYSTEM
C DEFINED OVER AN ENERGY GRID SUBDIVIDED INTO *MESH* INTERVALS. IT
C BECOMES A MATRIX EQUATION AF = B(F). THE MATRIX A, WHICH IS TRI-
C DIAGONAL, INCLUDES ALL OF THE TERMS FROM THE LHS OF THE BOLTZMANN
C EQUATION, AS WELL AS THE DIAGONAL ELEMENTS OF THE RHS INELASTIC
C COLLISION AND ELECTRON-ELECTRON SCATTERING TERMS. AFTER REDUCTION
C OF THE ELECTRON-ELECTRON TERMS TO FINITE DIFFERENCES, ONLY TRI-
C DIAGONAL TERMS RESULT, WITH COEFFICIENTS WHICH ARE EVALUATED USING
C THE PREVIOUS DISTRIBUTION FUNCTION. THE VECTOR B(F) IS A LINEAR
C FUNCTION OF F, COMPOSED OF THE OFF-DIAGUNAL ELEMENTS OF THE IN-
C ELASTIC SCATTERING TERM, ALSO EVALUATED USING THE PREVIOUS DISTRI-
C BUTION FUNCTION.
C
C THE BOLTZMANN EQUATION IS WRITTEN

```

BOLTZ 287  
 BOLTZ 288  
 BOLTZ 289  
 BOLTZ 290  
 BOLTZ 291  
 BOLTZ 292  
 BOLTZ 293  
 BOLTZ 294  
 BOLTZ 295  
 BOLTZ 296  
 BOLTZ 297  
 BOLTZ 298  
 BOLTZ 299  
 BOLTZ 300  
 BOLTZ 301  
 BOLTZ 302  
 BOLTZ 303  
 BOLTZ 304  
 BOLTZ 305  
 BOLTZ 306  
 BOLTZ 307  
 BOLTZ 308  
 BOLTZ 309  
 BOLTZ 310  
 BOLTZ 311  
 BOLTZ 312  
 BOLTZ 313  
 BOLTZ 314  
 BOLTZ 315  
 BOLTZ 316  
 BOLTZ 317  
 BOLTZ 318  
 BOLTZ 319  
 BOLTZ 320  
 BOLTZ 321  
 BOLTZ 322  
 BOLTZ 323  
 BOLTZ 324  
 BOLTZ 325  
 BOLTZ 326  
 BOLTZ 327  
 BOLTZ 328  
 BOLTZ 329  
 BOLTZ 330  
 BOLTZ 331  
 BOLTZ 332  
 BOLTZ 333  
 BOLTZ 334  
 BOLTZ 335  
 BOLTZ 336  
 BOLTZ 337  
 BOLTZ 338  
 BOLTZ 339  
 BOLTZ 340  
 BOLTZ 341  
 BOLTZ 342  
 BOLTZ 343

```

C   1/(SQRT(2E/M)*NE)*(SQRT(U)*F(U)*DNE/DT + DJF(U)/DU + DJEL(U)/DU      BOLTZ 344
C   * DJEE(U)/DU - SBEAM*S(U)) = ((NU(ION) + SOURCE/NE)*DELTA(U)      BOLTZ 345
C   * SUMK[U*F(U)*NK*QK(U)]),      BOLTZ 346
C
C WHERE
C
C   J(F)    = -(1/3)*NE*SQRT(2E/M)*ESQ*U DF/DU /<N QM(U)>      BOLTZ 347
C   J(EL)   = - NE*SQRT(2E/M)*USQ<(2ME/M)NQM(U)>*(F + (KT/E)DF/DU)      BOLTZ 348
C   J(EE)   = - (2PI/3)*Q**4/E**2*LN(LAMBDA)*NE*NE*SQRT(2E/M)*      BOLTZ 349
C   (PINT(U)*DF/DU + QINT(U)*F(U))      BOLTZ 350
C
C WHERE
C
C   PINT(U) = 2*INT(DU*U**1.5*F(U)) + 2*U**1.5*INT(DU*F(U))      BOLTZ 351
C   QINT(U) = 3*INT(DU*SQRT(U)*F(U))      BOLTZ 352
C   DEBYE  = SQRT(KTE/(4*PI*NE*NE))      BOLTZ 353
C   LAMBDA = DEBYE/RMIN      BOLTZ 354
C   RMIN   = ESU**2/(E*UBAR)      BOLTZ 355
C
C DEFINE QUANTITIES RELATED TO THE DRIVING FIELD AND ELASTIC COLLISI-      BOLTZ 356
C  ONS --
C
C   P(U)   = (E**2/3)*U/<N QMOM(U)> + (KT/E)*U**2 <(2ME/M)N QMOM(U)>      BOLTZ 357
C   Q(U)   = U**2 <(1ME/M)N QMOM(U)>      BOLTZ 358
C
C   P(I+.5) = (E**2/3)*QM(I,1) + (KT/E)*QM(I,2)      BOLTZ 359
C   Q(I+.5) = QM(I,2)      BOLTZ 360
C
C DEFINE TERMS RELATED TO ELECTRON-ELECTRON SCATTERING --
C
C   ALPHA(U) = (2PI/3)*NE(Q**4/E**2)*LOG(LAMBDA)*      BOLTZ 361
C   (PINT(U)/DU + QINT(U)/2)      BOLTZ 362
C   BETA(U)  = (2PI/3)*NE(Q**4/E**2)*LOG(LAMBDA)*      BOLTZ 363
C   (PINT(U)/DU - QINT(U)/2)      BOLTZ 364
C
C   A(I,1)  = P(I-.5)/DU - Q(I-.5)/2 + BETA(I-.5)      BOLTZ 365
C   A(I,2)  = -(P(I-.5) * P(I-.5))/DU + (Q(I+.5) - Q(I-.5))/2      BOLTZ 366
C   -DU*SUMK[NK(U*QK(U) * NK*(U + UK)*QK(U + UK))]      BOLTZ 367
C   - ALPHA(I-.5) - BETA(I+.5) - DU*SQRT(U M/2E)*DNEDT/NE      BOLTZ 368
C   - ALPHA(I-.5) - BETA(I+.5)      BOLTZ 369
C   A(I,3)  = P(I+.5)/DU + Q(I+.5)/2 + ALPHA(I+.5)      BOLTZ 370
C
C   B(I)    = -DU*SUMK[NK(U+UK)*QK(U+UK)*F(U+UK) + NK*QK(U)*F(U-UK)]      BOLTZ 371
C   - SQRT(M/2E)*(DU*S(U)/NE + NU(ION)*DELTA(U))      BOLTZ 372
C
C THE REDUCTION OF THE BOLTZMANN EQUATION TO A TRIDIAGONAL FINITE      BOLTZ 373
C DIFFERENCE SYSTEM OF EQUATIONS --
C
C   A(I+1)*F(I-1) + A(I+2)*F(I) + A(I+3)*F(I+1) = B(I)      BOLTZ 374
C
C I = 2,3,...,M. THE SET IS COMPLETED BY THE CONDITION THAT F(1)      BOLTZ 375
C CAN BE APPROXIMATED AS (E.G.) F(1) = F(2).      BOLTZ 376
C
C IN GENERAL, DIRECT INVERSION OF THE LINEAR SYSTEM IS NOT PRACTICAL      BOLTZ 377
C EITHER FROM THE STANDPOINT OF CORE STORAGE OR MATRIX INVERSION.      BOLTZ 378
C INSTEAD, ADVANTAGE IS TAKEN OF THE FACT THAT A IS SPARSE (I.E. ALL      BOLTZ 379
C ELEMENTS EXCEPT FOR THREE DIAGONALS ARE ZERO), AND SOLUTION IS      BOLTZ 380
C

```

```

C      ATTEMPTED BY AN ITERATIVE APPROACH: A(F(N))*F(N+1) = B(F(N)).      BOLTZ    401
C      -----
C      40 TRANS = DLNEDT/EM      BOLTZ    402
C      THE FIRST EQUATION IS F(1) - F(2) = 0 --      BOLTZ    403
C      A(1,1) = 0.      BOLTZ    404
C      A(1,2) = 1.0      BOLTZ    405
C      A(1,3) = - 1.0      BOLTZ    406
C      Q1 = QM(1,2)/2.      BOLTZ    407
C      P1 = (KT*QM(1,2) + ESQ3*QM(1,1))/DX      BOLTZ    408
C      DO 1 I = 2,M      BOLTZ    409
C      Q2 = QM(I,2)/2.      BOLTZ    410
C      P2 = (KT*QM(I,2) + ESQ3*QM(I,1))/DX      BOLTZ    411
C      DIAG = 0.      BOLTZ    412
C      DO 4 J = 1,NK      BOLTZ    413
C      DIAG = DIAG + N1(J)*XQ(I,J)      BOLTZ    414
C      JU = U(J)/DX      BOLTZ    415
C      IJ = I*JU      BOLTZ    416
C      IF (IJ.GT.M) GO TO 4      BOLTZ    417
C      IF (IJ.LT.1) GO TO 4      BOLTZ    418
C      DIAG = DIAG + N2(J)*XQ(IJ,J)      BOLTZ    419
C      4 CONTINUE      BOLTZ    420
C      ROOT = SQRT(X(I))      BOLTZ    421
C      A(I,1) = P1 - Q1      BOLTZ    422
C      A(I,2) = - P2 - P1 + Q2 - Q1 - (DIAG + ROOT*TRANS)*DX      BOLTZ    423
C      A(I,3) = P2 + Q2      BOLTZ    424
C      P1 = P2      BOLTZ    425
C      1 Q1 = Q2      BOLTZ    426
C
C      IF (NE.EQ.0.) GO TO 10      BOLTZ    427
C      INTEGRAL OF F(U) (FOR E-E SCATTERING) --      BOLTZ    428
C      CALL SIMPSON (F, MESH/2, 1., FINT)      BOLTZ    429
C      -----
C      CALCULATE THE RHS VECTOR B, WHICH CONTAINS ALL OF THE OFF-DIAGONAL      BOLTZ    430
C      PARTS OF THE INELASTIC COLLISION TERMS, DEFINED IN TERMS OF THE      BOLTZ    431
C      PRESENT DISTRIBUTION VECTOR F(N):      BOLTZ    432
C
C      10 IF (ITER.EQ.ITMAX) GO TO 98      BOLTZ    433
C      CALL SECOND (TIME)      BOLTZ    434
C      TIME = TIME-T0      BOLTZ    435
C      IF (TIME.GT.TMAX) GO TO 98      BOLTZ    436
C      ITER = ITER+1      BOLTZ    437
C      B(1) = 0.      BOLTZ    438
C      DO 5 I = 2,M      BOLTZ    439
C      OFF = 0.      BOLTZ    440
C      DO 8 J = 1,NK      BOLTZ    441
C      JU = U(J)/DX      BOLTZ    442
C      IF (NEL(J).LT.0) GO TO 8      BOLTZ    443
C
C      COLLISIONS (OF THE FIRST KIND) IN WHICH ELECTRONS ARE      BOLTZ    444
C      SCATTERED INTO ENERGY X(I), LOSING ENERGY U(J) --      BOLTZ    445
C      IJ = I*JU      BOLTZ    446
C      IF (IJ.GT.M) GO TO 6      BOLTZ    447
C      IF (IJ.LT.1) GO TO 6      BOLTZ    448

```

```

      OFF = OFF + XQ(IJ,J)*F(IJ)*N1(J)          BOLTZ   458
C
C       6 IF (NEL(J).GT.0) GO TO 8               BOLTZ   459
C
C           COLLISIONS (OF THE SECOND KIND) IN WHICH ELECTRONS ARE
C           SCATTERED INTO ENERGY X(I), GAINING ENERGY U(J) --
C           IJ = I-JU                           BOLTZ   460
C           IF (IJ.LT.1) GO TO 8                 BOLTZ   461
C           IF (IJ.GT.M) GO TO 8                 BOLTZ   462
C           OFF = OFF + XQ(IJ,J)*F(IJ)*N2(J)     BOLTZ   463
C
C           R CONTINUE                         BOLTZ   464
C           5 B(I) = - (C+S(I) + OFF)*DX        BOLTZ   465
C
C           PUT THE DELTA-FUNCTION SINGULARITY ARISING FROM SECONDARY IONIZA-
C           TION AND EXTERNAL CREATION OF ELECTRONS AT ZERO ENERGY AT U = DX
C           (I.E., AT I = 2) --                   BOLTZ   466
C           B(2) = B(2) - (IONIZE + S0)/EM        BOLTZ   467
C
C-----                                     BOLTZ   468
C
C           SOLVE THE LINEAR EQUATION: A(F(N))*F(N+1) = B(F(N)) --
C
C           THIS LOOP FOR SOLVING A*F(N+1) = B WILL ULTIMATELY PLACE THE SOLU-
C           TION F(N+1) INTO THE VECTOR G, AND DOES NOT DESTROY THE MATRIX A
C           CONSTRUCTED ABOVE. THE SOLUTION WORKS DOWNWARD (ANNIHILATING THE
C           LOWER DIAGONAL) BY MEANS OF ELEMENTARY ROW OPERATIONS, FOLLOWED
C           WITH BACK SUBSTITUTION UPWARDS.          BOLTZ   469
C
C           IN THE DOWNWARD ANNIHILATION OF THE LOWER DIAGONAL, THE VECTOR
C           G(I) IS TEMPORARILY USED TO STORE THE COEFFICIENTS OF F(I+1) --  BOLTZ   470
C
C           G(1) = A(1,3)/A(1,2)                  BOLTZ   471
C           B(1) = B(1)/A(1,2)                  BOLTZ   472
C           F1 = -F(1)/2.                      BOLTZ   473
C           F2 = F3 = 0.                        BOLTZ   474
C           ELECT = PNEW = 0.                  BOLTZ   475
C           A2 = B2 = 0.                        BOLTZ   476
C           XHALF = DX/2.                      BOLTZ   477
C           IF (NE,EQ.0.) GO TO 35            BOLTZ   478
C           DEBYE LENGTH FOR PLASMA --       BOLTZ   479
C           DEBYE = SQRT(KTE/(4.*PI*NE*ESQ))  BOLTZ   480
C           CLASSICAL DISTANCE OF CLOSEST APPROACH -- BOLTZ   481
C           RMIN = 300.*ESU/UBAR              BOLTZ   482
C           LAMBDA = DEBYE/RMIN             BOLTZ   483
C           LOG = ALOG(LAMBDA)              BOLTZ   484
C
C           35 DO 26 I = 1,M                  BOLTZ   485
C           A1 = A2                          BOLTZ   486
C           B1 = B2                          BOLTZ   487
C           IF (NE,EQ.0.) GO TO 20          BOLTZ   488
C           XROOT = SQRT(X(I))
C           X32 = X(I)*XROOT              BOLTZ   489
C           F1 = F1 + F(I)                BOLTZ   490
C           F2 = F2 + XROOT*F(I)          BOLTZ   491
C           F3 = F3 + X32*F(I)           BOLTZ   492
C           X32 = XHALF**1.5             BOLTZ   493
C           XHALF = XHALF + DX           BOLTZ   494
C
C           26 CONTINUE
C           20 IF (NE,EQ.0.) GO TO 15
C           XHALF = XHALF + DX

```

```

C      PU = PINT(U)/DU, AND QU = QINT(U)/2.          BOLTZ    515
C      PU = F3 + X32*(FINT-F1)                      BOLTZ    516
C      PU = PU + PU                                  BOLTZ    517
C      QU = 1.5*F2*DX                                BOLTZ    518
C      A2 = CONST*(PU + QU)*LOG                     BOLTZ    519
C      B2 = CONST*(PU - QU)*LOG                     BOLTZ    520
C      DF = FAVG2 = 0.                               BOLTZ    521
C      IF (I.LT.M) DF = F(I+1) - F(I)              BOLTZ    522
C      IF (I.LT.M) FAVG2 = F(I+1) + F(I)            BOLTZ    523
C      POLD = PNEW                                    BOLTZ    524
C      PNEW = (PU*DF + QU*FAVG2)*LOG               BOLTZ    525
C      ELECT = ELECT + 0.5*(POLD + PNEW)           BOLTZ    526
C      20 IF (I.EQ.1) GO TO 26                      BOLTZ    527
C      B(I) = B(I) - (A(I,1) + B1)*B(I-1)         BOLTZ    528
C      G(I) = A(I,3) + A2                           BOLTZ    529
C      Q = (A(I,2) - A1 - B2) - (A(I,1) + B1)*G(I-1)
C      B(I) = B(I)/Q                                BOLTZ    530
C      G(I) = G(I)/Q                                BOLTZ    531
C      26 CONTINUE                                 BOLTZ    532
C
C      UPWARD BACK SUBSTITUTION, WITH NEW SOLUTION INTO G --
C
C      G(M) = B(M)                                BOLTZ    533
C      DO 3 J = 1,MESH                            BOLTZ    534
C      I = M-J                                  BOLTZ    535
C      3 G(I) = B(I) - G(I)*G(I+1)                BOLTZ    536
C
C      -----
C      RENORMALIZE THE NEW DISTRIBUTION FUNCTION, AND TEST FOR
C      CONVERGENCE --
C
C      DO 25 I = 1,M                            BOLTZ    537
C      25 B(I) = SQRT(X(I))*G(I)                  BOLTZ    538
C
C      CALL SIMPSON (B, MESH/2, DX, FNORM)        BOLTZ    539
C      CALL SIMPSON (G, MESH/2, 1.0, GINT)          BOLTZ    540
C      FINT = GINT/FNORM                          BOLTZ    541
C
C      DELTA = 0.                                BOLTZ    542
C      ERROR = .FALSE.                           BOLTZ    543
C      CONVRGE = .TRUE.                           BOLTZ    544
C      DO 11 I = 1,M                            BOLTZ    545
C      FOLD = F(I)                                BOLTZ    546
C      F(I) = G(I)/FNORM                         BOLTZ    547
C      FNEW = F(I)                                BOLTZ    548
C      B(I) = B(I)*X(I)                          BOLTZ    549
C      F1 = ABS(FOLD)                            BOLTZ    550
C      F2 = ABS(FNEW)                            BOLTZ    551
C      FMAX = F1                                 BOLTZ    552
C
C      IF (F2.GT.F1) FMAX = F2                  BOLTZ    553
C      ERROR = ERROR.OR.(FNEW.LT.0.)             BOLTZ    554
C      ETA = 0.                                 BOLTZ    555
C      IF ((FMAX.AND.MASK).NE.0) ETA = ABS(FNEW-FOLD)/FMAX
C      IF (ETA.GT.DELTA) DELTA = ETA            BOLTZ    556
C      TEST = ETA.LE.EPSILON                    BOLTZ    557
C      11 CONVRGE = CONVRGE.AND.TEST           BOLTZ    558

```

```

CONVRGE = CONVRGE.AND.(.NOT.ERROR)
CALL SIMPSON (B, MESH/2, DX, UBAR)
UBAR = UBAR/FNORM
KTE = (2./3.)*E*UBAR
TE = KTE/KB
IF (.NOT.CONVRGE) GO TO 10
C
C -----
C   CALCULATE THE FORWARD AND REVERSE COLLISION RATES VSIG(1,J) AND
C   VSIG(2,J) (CM3/SEC) AND NET POWER BALANCE POWER(J) (WATT/ELECTRON)
C   FOR EACH OF THE INELASTIC PROCESSES J --
C
65 DLNEDT0 = DLNEDT
PCOLL = 0.
C   IONIZE AND ATTACH ARE THE IONIZATION AND ATTACHMENT FREQUENCIES--
IONIZE = ATTACH = 0.
DO 16 J = 1,NK
FORWARD = REVERSE = 0.
JU = UJ = U(J)/DX
DO 15 I = 1,M
A(I,1) = F(I)*XQ(I,J)
A(I,2) = 0.
IJ = I*JU
IF (IJ.LT.1) GO TO 15
IF (IJ.GT.M) GO TO 15
A(I,2) = F(I)*XQ(IJ,J)
15 A(I,3) = X(I)*A(I,1)
C
CALL SIMPSON (A(I,1), MPTS, DX, VSIG(1,J))
CALL SIMPSON (A(I,2), MPTS, DX, VSIG(2,J))
C
C   NOTE: EM = SQRT[2(1.602E-12)/9.31E-28] = 5.866E 07 EV**(-1/2)CM/S.
C   UNITS OF F(U)UQ(U)DU ARE EV**(1/2)CM2, SO RESULT FOR <VSIG> WILL
C   BE IN UNITS OF CM3/SEC.
C
VSIG(1,J) = EM*VSIG(1,J)
VSIG(2,J) = EM*VSIG(2,J)
C
IF (NEL(J).LT.0) GO TO 24
FORWARD = U(J)*VSIG(1,J)
REVERSE = U(J)*VSIG(2,J)
GO TO 27
C
24 CALL SIMPSON (A(I,3), MPTS, DX, FORWARD)
FORWARD = EM*FORWARD
C
27 POWER(J) = CO*(N1(J)*FORWARD - N2(J)*REVERSE)
C
FREQ = N1(J)*VSIG(1,J)
IF (NEL(J).EQ.+1) IONIZE = IONIZE + FREQ
IF (NEL(J).EQ.-1) ATTACH = ATTACH + FREQ
C
16 PCOLL = PCOLL + POWER(J)
DLNEDT = SEXY + IONIZE - ATTACH
IF (ITMAX.LE.0) GO TO 60
C

```

BOLTZ 572  
 BOLTZ 573  
 BOLTZ 574  
 BOLTZ 575  
 BOLTZ 576  
 BOLTZ 577  
 BOLTZ 578  
 BOLTZ 579  
 BOLTZ 580  
 BOLTZ 581  
 BOLTZ 582  
 BOLTZ 583  
 BOLTZ 584  
 BOLTZ 585  
 BOLTZ 586  
 BOLTZ 587  
 BOLTZ 588  
 BOLTZ 589  
 BOLTZ 590  
 BOLTZ 591  
 BOLTZ 592  
 BOLTZ 593  
 BOLTZ 594  
 BOLTZ 595  
 BOLTZ 596  
 BOLTZ 597  
 BOLTZ 598  
 BOLTZ 599  
 BOLTZ 600  
 BOLTZ 601  
 BOLTZ 602  
 BOLTZ 603  
 BOLTZ 604  
 BOLTZ 605  
 BOLTZ 606  
 BOLTZ 607  
 BOLTZ 608  
 BOLTZ 609  
 BOLTZ 610  
 BOLTZ 611  
 BOLTZ 612  
 BOLTZ 613  
 BOLTZ 614  
 BOLTZ 615  
 BOLTZ 616  
 BOLTZ 617  
 BOLTZ 618  
 BOLTZ 619  
 BOLTZ 620  
 BOLTZ 621  
 BOLTZ 622  
 BOLTZ 623  
 BOLTZ 624  
 BOLTZ 625  
 BOLTZ 626  
 BOLTZ 627  
 BOLTZ 628

```

C      D/DT(NE) IS DETERMINED SELF-CONSISTENTLY BY AN ITERATIVE LOOP. IF BOLTZ      629
C      THE CHANGE IN THE (LOGARITHMIC) DERIVATIVE OF NE BECOMES  $\leq$  2 %, NO BOLTZ      630
C      FURTHER ITERATION IS REQUIRED -- BOLTZ      631
C      BOLTZ      632
C      BOLTZ      633
C      BOLTZ      634
C      BOLTZ      635
C      BOLTZ      636
C      BOLTZ      637
C      BOLTZ      638
C      BOLTZ      639
C      BOLTZ      640
C      BOLTZ      641
C      BOLTZ      642
C      BOLTZ      643
C      BOLTZ      644
C      BOLTZ      645
C      BOLTZ      646
C      BOLTZ      647
C      BOLTZ      648
C      BOLTZ      649
C      BOLTZ      650
C      BOLTZ      651
C      BOLTZ      652
C      BOLTZ      653
C      BOLTZ      654
C      BOLTZ      655
C      BOLTZ      656
C      BOLTZ      657
C      BOLTZ      658
C      BOLTZ      659
C      BOLTZ      660
C      BOLTZ      661
C      BOLTZ      662
C      BOLTZ      663
C      BOLTZ      664
C      BOLTZ      665
C      BOLTZ      666
C      BOLTZ      667
C      BOLTZ      668
C      BOLTZ      669
C      BOLTZ      670
C      BOLTZ      671
C      BOLTZ      672
C      BOLTZ      673
C      BOLTZ      674
C      BOLTZ      675
C      BOLTZ      676
C      BOLTZ      677
C      BOLTZ      678
C      BOLTZ      679
C      BOLTZ      680
C      BOLTZ      681
C      BOLTZ      682
C      BOLTZ      683
C      BOLTZ      684
C      BOLTZ      685

C      IF (ABS(DLNEDT-DLNEDT0).GT.ABS(DLNEDT)/50.) GO TO 40
C      IF (EPS.GE.EPSILON) GO TO 60
C      FINE TUNING OF CALCULATION --
C          EPSILON = EPS
C          GO TO 40
C      DETERMINE CP TIME REQUIRED FOR OBTAINING F(U) --
C      60 CALL SECOND (TIME)
C          TIME = TIME-T0
C      -----
C      XBAR = -DX/2.
C      DO 9 I = 1,M
C          G(I) = F(I)/F(1)
C          B(I) = B(I)*X(I)
C          CALL INTERP (I, XBAR, QMAVG, X, QM(I,1), 1, M)
C          A(I,1) = F(I)*X(I)*X(I)/QMAVG
C          IF (F(I).GT.1.E-20*F(I)) MP = I
C      9 XBAR = XBAR + DX
C      F HAS BEEN NORMALIZED TO INT(DU*SQRT(U)*F(U)) = 1, AND G(I) =
C          F(I)/F(1).
C      ELASTIC COLLISION FREQUENCY NU:
C      CALL SIMPSON (A, MPTS, DX, NU)
C          NU = EM*NU
C      CALCULATE AVERAGE ENERGY AND EFFECTIVE TEMPERATURE --
C      V
C      IF (ITER.GT.0) GO TO 75
C      CALL SIMPSON (B, MESH/2, DX, UBAR)
C      UBAR = UBAR/FNORM
C      KTE = (2./3.)*E*UBAR
C      TE = KTE/KB
C      -----
C      CALCULATE DIFFUSION COEFFICIENT D, MOBILITY MU, CHARACTERISTIC
C      ENERGY EK = D/MU, DRIFT VELOCITY VD = MU*EVCM, AND THE FORWARD AND
C      REVERSE ELASTIC POWER TRANSFER --
C      75 F1 = F(1)
C      DO 12 I = 1,MESH
C          F0 = F1
C          F1 = F(I+1)
C          FAVG = (F0 + F1)/2.
C          DF = F1-F0
C          A(I,1) = QM(I,1)*FAVG
C          A(I,2) = QM(I,1)*DF
C          A(I,3) = QM(I,2)*FAVG

```

```

12 B(1) = QM(I,2)*DF          BOLTZ   686
A(M,1) = QM(M,1)*F1          BOLTZ   687
A(M,2) = QM(M,1)*DF          BOLTZ   688
A(M,3) = QM(M,2)*F1          BOLTZ   689
B(M)  = QM(M,2)*DF          BOLTZ   690
C
C -----
C DIFFUSION COEFFICIENT D (CM2/SEC) --
C
C CALL SIMPSON (A(1,1), MPTS, DX, D)
C     D = EM*D/3.
C
C MOBILITY MU (CM2/VOLT/SEC) --
C
C CALL SIMPSON (A(1,2), MPTS, 1.0, MU)
C     INTEGRATION-BY-PARTS CORRECTION TERM --
C     MU = MU - EMAX*A(M,2)/DX
C     MU = - EM*MU/3.
C
C EK = D/MU
C VD = MU*EVCM
C AMPS = E*1.0E-07*VD
C COND = E*1.0E-07*MU
C RHN = 1./COND
C CONDUCT = NE*COND
C DNEDT = NE*DLNEDT
C
C -----
C FORWARD (PF) AND REVERSE (PR) ELASTIC POWER --
C
C CALL SIMPSON (A(1,3), MPTS, DX, PF)
C     PF = C0*EM*PF
C
C CALL SIMPSON (B, MPTS, 1.0, PR)
C     PR = C0*EM*KT*PR
C
C ELASTIC = PF + PR
C
C ELECTRIC DISCHARGE POWER DENSITY --
C DISCH = C0*MU*EVCM*EVCM
C
C RATE OF CHANGE OF STORED ELECTRON KINETIC ENERGY --
C DEDT = C0*UBAR*DLNEDT
C
C PTOTAL = DISCH + DEPOSIT
C PWR = PCOLL + ELASTIC + DEDT
C ELECTRON-ELECTRON SCATTERING POWER DISCREPANCY --
C ELECT = C0*EM*CONST*DX*ELECT
C
C IF (PTOTAL.EQ.0.) GO TO 53
C     P = PTOTAL/100.
C     PCT(1) = ELASTIC/P
C     PCT(2) = PCOLL/P
C     PCT(3) = DEDT/P
C     PCT(4) = ELECT/P

```

BOLTZ 691  
 BOLTZ 692  
 BOLTZ 693  
 BOLTZ 694  
 BOLTZ 695  
 BOLTZ 696  
 BOLTZ 697  
 BOLTZ 698  
 BOLTZ 699  
 BOLTZ 700  
 BOLTZ 701  
 BOLTZ 702  
 BOLTZ 703  
 BOLTZ 704  
 BOLTZ 705  
 BOLTZ 706  
 BOLTZ 707  
 BOLTZ 708  
 BOLTZ 709  
 BOLTZ 710  
 BOLTZ 711  
 BOLTZ 712  
 BOLTZ 713  
 BOLTZ 714  
 BOLTZ 715  
 BOLTZ 716  
 BOLTZ 717  
 BOLTZ 718  
 BOLTZ 719  
 BOLTZ 720  
 BOLTZ 721  
 BOLTZ 722  
 BOLTZ 723  
 BOLTZ 724  
 BOLTZ 725  
 BOLTZ 726  
 BOLTZ 727  
 BOLTZ 728  
 BOLTZ 729  
 BOLTZ 730  
 BOLTZ 731  
 BOLTZ 732  
 BOLTZ 733  
 BOLTZ 734  
 BOLTZ 735  
 BOLTZ 736  
 BOLTZ 737  
 BOLTZ 738  
 BOLTZ 739  
 BOLTZ 740  
 BOLTZ 741  
 BOLTZ 742

```

PCT(5) = DISCH/P          BOLTZ    743
PCT(6) = DEPOSIT/P        BOLTZ    744
PERCENT = PCT(1) + PCT(2) + PCT(3) + PCT(4)  BOLTZ    745
PBAL = PERCENT - 100.     BOLTZ    746
GO TO 54                 BOLTZ    747
C
      53     PMAX = AMAX1(ABS(ELASTIC), ABS(PCOLL), ABS(DEDT))  BOLTZ    748
              P = ELASTIC + ELECT + PCOLL * DEDT               BOLTZ    749
              PBAL = 100.*P/PMAX                            BOLTZ    750
C
      54     FB = 2./(KTE/E)**1.5/SQRT(PI)                   BOLTZ    751
              EX = EXP(-E*DX/KTE)                         BOLTZ    752
              DU 19 I = 1,M                                BOLTZ    753
              B(I) = F(I)/FB                           BOLTZ    754
19   FB = FR*EX                         -----
C
C-----TABLE OF PLASMA PARAMETERS (UBAR, TE, VD, D, ... ETC.) --
C
      NG1 = NGAS-1           BOLTZ    755
      IF (.NOT.UUT(I)) GO TO 21             BOLTZ    756
      NX = 68-6*NGAS          BOLTZ    757
      ENCODE (150,250,FORM) NX, NG1, NG1  BOLTZ    758
      WRITE (6,350) KAPTION            BOLTZ    759
      IF (NGAS.EQ.1) WRITE (6,210) NAME(I)  BOLTZ    760
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I),I=1,NGAS), (FI(I),I=1,NGAS)  BOLTZ    761
C
      WRITE (6,100) EVCM, TMOL, PTOT, ATM, EN, EP, KVCMATH, UBAR,  BOLTZ    762
1     TE, D, MU, EK, VD, COND, CONDUCT, RHO, AMPS, DISCH, KAPT,  BOLTZ    763
2     DEPOSIT, KAPT, PTOTAL, KAPT, PCOLL, KAPT, ELASTIC, KAPT, DEDT,  BOLTZ    764
3     KAPT, PWR, KAPT, NU, NE, FE, SD, SB, IONIZE, ATTACH, DLNEDT,  BOLTZ    765
4     DNEDT                BOLTZ    766
C
      WRITE (6,380) MESH, EMAX, DX          BOLTZ    767
C
C-----TABULAR OUTPUT OF ELECTRON ENERGY DISTRIBUTION FUNCTION --
C
      21 NX = 58 - 6*NGAS           BOLTZ    768
      ENCODE (150,220,FORM) NX, NG1, NG1  BOLTZ    769
      IF (.NOT.OUT(2)) GO TO 22          BOLTZ    770
      MULT = (MESH-1)/100 + 1            BOLTZ    771
      DU = MULT*DX                      BOLTZ    772
      WRITE (6,350) KAPTION            BOLTZ    773
      WRITE (6,200) DU, X(M), MESH, DX  BOLTZ    774
      WRITE (6,330) EN, EP, KVCMATH    BOLTZ    775
      IF (FE.GT.0.) WRITE (6,320) FE, NE  BOLTZ    776
      IF (NGAS.EQ.1) GO TO 17          BOLTZ    777
      WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I),I=1,NGAS), TMOL  BOLTZ    778
      GO TO 7                          BOLTZ    779
17   WRITE (6,260) NAME(I), TMOL       BOLTZ    780
?    WRITE (6,290)                  BOLTZ    781
?    WRITE (6,240) (X(I), F(I), I = 1,M,MULT)  BOLTZ    782
      WRITE (6,160)                  BOLTZ    783
      IF (ITER.GT.0) WRITE (6,180) TIME, ITER, DELTA, PBAL      BOLTZ    784
                                         -----

```

```

      WRITE (6,280) DATE          BOLTZ    800
C   22 IF (.NOT.OUT(3)) GO TO 23 BOLTZ    801
C   -----
C   PLOT THE DISTRIBUTION FUNCTION F(U)/F(0) -- BOLTZ    802
C   -----
C   WRITE (6,110) F(1)          BOLTZ    803
C   WRITE (6,330) EN, EP, KVCMATM BOLTZ    804
C   IF (FE.GT.0.) WRITE (6,320) FE, NE BOLTZ    805
C   IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL BOLTZ    806
C   IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS),
C   1     (FI(I),I=1,NGAS), TMOL BOLTZ    807
C   -----
C   DF = 2.                      BOLTZ    808
C   IF (G(MP).GT.1.E-10) DF = 1. BOLTZ    809
C   F0 = -10.*DF                 BOLTZ    810
C   CALL PXLOGY (MAX, MP, 1, G, F0, DF, X, 0., 0., .TRUE., .FALSE.,
C   1.TRUE., .TRUE., .TRUE., TITLE, 1, 0) BOLTZ    811
C   WRITE (6,150) DATE          BOLTZ    812
C   -----
C   BOLTZMANN DISTRIBUTION FUNCTION WITH EFFECTIVE TEMPERATURE TE IS BOLTZ    813
C   GIVEN BY FB(U,TE) = 2(KTE/E)**(-3/2)/SQRT(Pi)*EXP(-E*U/KTE). PLOT BOLTZ    814
C   F(U)/FB(U,TE) --           BOLTZ    815
C   -----
C   23 IF (.NOT.OUT(4)) GO TO 30 BOLTZ    816
C   -----
C   WRITE (6,120) TE             BOLTZ    817
C   WRITE (6,330) EN, EP, KVCMATM BOLTZ    818
C   IF (FE.GT.0.) WRITE (6,320) FE, NE BOLTZ    819
C   IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL BOLTZ    820
C   IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS),
C   1     (FI(I), I = 1,NGAS), TMOL BOLTZ    821
C   -----
C   CALL PXLOGY (MAX, MP, 1, B, -4.0, .5, X, 0., 0., .TRUE., .FALSE.,
C   1.TRUE., .TRUE., .TRUE., TITLE, 1, 0) BOLTZ    822
C   WRITE (6,150) DATE          BOLTZ    823
C   -----
C   TABULAR PLOT OF COLLISION RATES, POWER TRANSFER -- BOLTZ    824
C   -----
C   30 IF (.NOT.OUT(5)) GO TO 99 BOLTZ    825
C   SKIP = SH(1H1)                BOLTZ    826
C   LSKIP = 1H                     BOLTZ    827
C   IF (SB.EQ.0.) LSKIP = 1H+       BOLTZ    828
C   L = 15                         BOLTZ    829
C   LINE = LL = 0                  BOLTZ    830
C   DO 18 J = 1,NK                 BOLTZ    831
C   IF (LL.NE.LINE) GO TO 28       BOLTZ    832
C   J1 = L - (NK-J+1)              BOLTZ    833
C   IF (J1.GT.7) J1 = 7            BOLTZ    834
C   IF (J1.GE.1) ENCODE (10,300,SKIP) J1 BOLTZ    835
C   -----
C   BOLTZ                         BOLTZ    836
C   BOLTZ                         BOLTZ    837
C   BOLTZ                         BOLTZ    838
C   BOLTZ                         BOLTZ    839
C   BOLTZ                         BOLTZ    840
C   BOLTZ                         BOLTZ    841
C   BOLTZ                         BOLTZ    842
C   BOLTZ                         BOLTZ    843
C   BOLTZ                         BOLTZ    844
C   BOLTZ                         BOLTZ    845
C   BOLTZ                         BOLTZ    846
C   BOLTZ                         BOLTZ    847
C   BOLTZ                         BOLTZ    848
C   BOLTZ                         BOLTZ    849
C   BOLTZ                         BOLTZ    850
C   BOLTZ                         BOLTZ    851
C   BOLTZ                         BOLTZ    852
C   BOLTZ                         BOLTZ    853
C   BOLTZ                         BOLTZ    854
C   BOLTZ                         BOLTZ    855
C   BOLTZ                         BOLTZ    856

```

```

        WRITE (6,SKIP)
        WRITE (6,270) KAPTION
        WRITE (6,330) EN, EP, KVCMATH
        IF (NE.GT.0.) WRITE (6,320) FE, NE
        IF (NGAS.EQ.1) WRITE (6,260) NAME(1), TMOL
        IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I),I = 1+NGAS),
1          (FI(I),I=1,NGAS)+ TMOL
        WRITE (6,140) UBAR, TE, VD, MU
        WRITE (6,230) KAPT
        IF (LINE.GT.0) GO TO 28
        WRITE (6,170) ELASTIC
        IF (PTOTAL.NE.0.) WRITE (6,420) PCT(1)
        IF (SB.NE.0.) WRITE (6,430) SB
        IF (SO.NE.0.) WRITE (6,390) LSKIP, SO
28 IF (N1(J)*N2(J).EQ.0.) GO TO 18
        LINE = LINE+1
        FREQ = NEL(J)*N1(J)*VSIG(1,J)
        IF (PTOTAL.NE.0.) PCTJ = POWER(J)/P
        DO 29 I = 1,6
29 NOUT(I) = 1H
        IF (N1(J).NE.0.) ENCODE (10+190,NOUT(1)) N1(J)
        IF (N2(J).NE.0.) ENCODE (10+190,NOUT(2)) N2(J)
        IF (NEL(J).GE.0) ENCODE (10+360,NOUT(3)) U(J)
        IF (NEL(J).NE.0) ENCODE (10+190,NOUT(4)) FREQ
        ENCODE (10+190,NOUT(5)) VSIG(1,J)
        IF (NEL(J).EQ.0) ENCODE (10+190,NOUT(6)) VSIG(2,J)
C      WRITE (6,130) LINE, (PROCESS(I,J), I = 1+4), NOUT, POWER(J)
C      IF (PTOTAL.NE.0.) WRITE (6,420) PCTJ
C      LL = L*(LINE/L)
C      IF (LL.EQ.LINE) WRITE (6,160)
18 CONTINUE
        IF (LL.NE.LINE) WRITE (6,160)
        PWR = PWR + ELECT
        IF (PTOTAL.EQ.0.) GO TO 55
        WRITE (6,370) DISCH, KAPT, PCT(5), PCOLL, PCT(2), DEPOSIT, KAPT,
1 PCT(6), ELASTIC, PCT(1), DEDT, PCT(3)
        IF (NE.GT.0.) WRITE (6,310) ELECT, PCT(4)
        WRITE (6,340) PTOTAL, KAPT, PWR, PERCENT
        GU TO 56
55 WRITE (6,400) PCOLL, KAPT, ELASTIC, KAPT, DEDT, KAPT
        IF (NE.GT.0.) WRITE (6,310) ELECT
        WRITE (6,410) PWR, KAPT
56 IF (ITER.GT.0) WRITE (6,180) TIME, ITER, DELTA, PBAL
        WRITE (6,280) DATE
C      GO TO 99
C      -----
C      ----- FORMAT STATEMENTS -----
C      100 FORMAT (/60X*PLASMA PARAMETERS//48X,42(IH-1)//50X*E (FIELD)*4X*= * BOLTZ 908
        11PE11.4,3X*VOLT/CM*/50X*TMOL*RX* = *0PF8.0,6X*DEG K*/50X*NMOL*8X BOLTZ 909
        2* = *1PE11.4,3X*CM-3*50X*PTOT*8X* = *0PF9.1,5X*TORR*/62X* = *F8.2 BOLTZ 910
        3,6X*ATM*/50X*E*N*9X* = *1PE11.4,3X*VOLT CM2*/50X*E/P*9X* = *0PF8. BOLTZ 911
        42*6X*V/CM*TORR*/62X* = *F8.2,6X*KV/CM/ATM*/50X*<U>*9X* = *0PF8.3, BOLTZ 912
        56X*EV*/50X*TE = 2<U>/3K = *F9.0,5X*DEG K*/50X*DIFFUSION D = * BOLTZ 913

```

$6F^9 \cdot 2 \cdot 5X \cdot CM^2 / SEC^* / 50X \cdot MOBILITY \cdot MU = "F8.1,6X \cdot CM^2 / VOLT / S^* / 50X \cdot EK = BOLTZ$  914  
 $7D / MU \cdot 3X^* = "OPF8.3,6X \cdot EV^* / 50X \cdot VDRIFT = MU \cdot E = #IPE11.4 \cdot 3X \cdot CM / SEC^* // BOLTZ$  915  
 $850X \cdot CONDUCT / NE = "E11.4 \cdot 3X \cdot CM^2 / OHM^* / 50X \cdot CONDUCT \cdot 5X^* = "E11.4,3X BOLTZ$  916  
 $9 \cdot OHM \cdot CM^* / 50X \cdot RHO \cdot NE \cdot 6X^* = "E11.4 \cdot 3X \cdot OHM / CM^2 \cdot / 50X \cdot J / NE = E \cdot VD = # BOLTZ$  917  
 $SE11.4 \cdot 3X \cdot AMP \cdot CM / EL^* // 50X \cdot DISCHARGE \cdot 3X^* = "E11.4 \cdot 3X, A10 / 50X \cdot DEPOSIT BOLTZ$  918  
 $110N = "E11.4 \cdot 3X, A10 / 50X \cdot TOTAL \cdot POWER = "E11.4 \cdot 3X, A10 / 50X \cdot INELAS BOLTZ$  919  
 $2TIC = "E11.4 \cdot 3X, A10 / 50X \cdot ELASTIC \cdot HEAT = "E11.4 \cdot 3X, A10 / 50X \cdot E < U > DN BOLTZ$  920  
 $3E / DT = "E11.4 \cdot 3X, A10 / 50X \cdot POWER \cdot DISS = "E11.4 \cdot 3X, A10 // / 50X \cdot NU(MO BOLTZ$  921  
 $4H \cdot 5X^* = "E11.4 \cdot 3X \cdot SEC-1^* // / 50X \cdot NE \cdot 10X^* = "E11.4 \cdot 3X \cdot CM-3^* / 50X \cdot NE / NM BOLTZ$  922  
 $5OL \cdot 5X^* = "E11.4 \cdot 5X \cdot S(U = 0) / NE = "E11.4 \cdot 3X \cdot SEC-1^* / 50X \cdot S(U > 0) / N BOLTZ$  923  
 $6E = "E11.4 \cdot 3X \cdot SEC-1^* / 50X \cdot NU(IONIZE) = "E11.4 \cdot 3X \cdot SEC-1^* / 50X \cdot NU(A BOLTZ$  924  
 $7TT, REC) = "E11.4 \cdot 3X \cdot SEC-1^* / 50X \cdot ONE / DT / NE = "E11.4 \cdot 3X \cdot SEC-1^* / 50 BOLTZ$  925  
 $8X \cdot DNE / DT \cdot 6X^* = "E11.4 \cdot 3X \cdot CM-3 \cdot SEC^* // 48X \cdot 42(1H-)) BOLTZ$  926  
  
C 110 FORMAT (1H1,9X, \*LOG PLOT OF ELECTRON DISTRIBUTION F(U)/F(0) AS A BOLTZ 927  
FUNCTION OF ELECTRON ENERGY U (EV), WHERE F(0) = \*, IPE10.3,\* EV\*, BOLTZ 928  
22H\*,\*(-3/2)\*/) BOLTZ 929  
C 120 FORMAT (1H1,17X\*LOG PLOT OF RELATIVE ELECTRON DISTRIBUTION FUNCTIO BOLTZ 931  
IN F(U)/FBOLTZ(U,TE), WHERE TE = (2/3)UAVG/KB = "F6.0\* DEG K\*) BOLTZ 932  
C 130 FORMAT (/I3,3X,5A10,A12,A8,3A12,1PE13.3\*OPF8.2) BOLTZ 933  
C 140 FORMAT (26X,\*U(AVG) = \*,F6.3,\* EV\*, TE = \*,F6.0,\* DEG K, VD = \*, BOLTZ 934  
1 IPE8.2\* CM/SEC, MU = "E8.2\* CM^2/VOLT/SEC\*) BOLTZ 935  
C 150 FORMAT (/62X\*ELECTRON ENERGY U (EV)\*16X\*DR. WILLIAM B. LACINA,\*A11 BOLTZ 936  
1/100X\*NORTHROP RESEARCH AND TECHNOLOGY\*) BOLTZ 937  
C 160 FORMAT (/IX,135(1H-)) BOLTZ 938  
C 170 FORMAT (95X\*MOMENTUM TRANSFER = \*1PE10.3) BOLTZ 939  
C 180 FORMAT (/IX\*F(U) CONVERGED IN=FS.1\* SEC, IN=I3\* ITERATIONS; MAXIMU BOLTZ 940  
SM RELATIVE CHANGE IN LAST ITERATION <"IPE10.3\*: POWER BALANCE ACCU BOLTZ 941  
SRACY = "OPF6.2\* %\*) BOLTZ 942  
C 190 FORMAT (IPE10.3) BOLTZ 943  
C 200 FORMAT (15X \*NORMALIZED ELECTRON DISTRIBUTION FUNCTION F(U), IN BOLTZ 944  
1 UNITS OF EV\*, 2H\*,\*(-3/2), WITH OUTPUT AT=OPF6.3\* EV INTERVALS\*)/ BOLTZ 945  
215X\*THE ELECTRON ENERGY RANGE [0, \*OPF6.2\*) EV WAS SUBDIVIDED INTO BOLTZ 946  
3\*I4\* INTERVALS, GIVING A RESOLUTION = "F5.3\* EV.\*/) BOLTZ 947  
C 210 FORMAT (48X\*GAS MIXTURE -- // / 65X\*PURE \*A3) BOLTZ 948  
C 220 FORMAT (1H1,I2,4HXA3,,I2,20H(\*/\*A3),\* = "2PF6.2,,I2,38H(\* /\*F6.2), BOLTZ 949  
1\* TMOL = \*OPF5.0\* DEG K\*) BOLTZ 950  
C 230 FORMAT (2X\*J\*14X\*REACTION(J)\*21X\*N(A)\*9X\*N(B)\*5X\*U(J)\*4X\*DNE/DT/NE BOLTZ 951  
1\*2X\*<VSIG(A>B)> <VSIG(B>A)> (NET) POWER PERCENT\* /13X\*A + E(-) + BOLTZ 952  
2B + E(-)\*16X\*(CM-3)\*7X\*(CM-3)\*4X\*(EV)\*5X\*(SEC-1)\*4X\*(CM3/SEC)\*3X BOLTZ 953  
3\*(CM3/SEC)\*3X\*(A9\*) POWER\*/IX,135(1H-)) BOLTZ 954  
C 240 FORMAT (5(OPF12.3\*1PE14.5)/) BOLTZ 955  
C 250 FORMAT (24H(48X,\*GAS MIXTURE -- \*//,I2,5HX,A3,I2,20H(\*/\*A3)\* = \*.2 BOLTZ 956

1PF6.2,,I2+1H1(* /*F6.2))	BOLTZ	971
C 260 FORMAT (55X,*PURE *,A3,*, TMOL =*,F5.0,* DEG K*)	BOLTZ	972
C 270 FORMAT (48X,4A10//30X*PLASMA KINETICS ANALYSIS WITH SUMMARY OF ELE ICTRON PARAMETERS, COLLISION RATES,*/31X*AND POWER BALANCE FOR ALL 2ELASTIC AND INELASTIC COLLISION PROCESSES INCLUDED*)	BOLTZ	973
C 280 FORMAT (/103X, *DR. WILLIAM B. LACINA,*A11/103X*NORTHROP RESEARCH I AND TECHNOLOGY*)	BOLTZ	974
C 290 FORMAT (2X+132(1H-)/5(7X*U(EV)*7X*F(U)*3X)/2X+132(1H-)/)	BOLTZ	975
C 300 FORMAT (5H(1H1,,[1,4H(/)]))	BOLTZ	976
C 310 FORMAT (82X*E-E POWER TRANSFER DISCREPANCY = *1PE10.3,0PF8.2* %*)	BOLTZ	977
C 320 FORMAT (34X*FRACTIONAL IONIZATION = NE/NTOT = *1PE10.3*, NE = * 1 E10.3* CM-3*)	BOLTZ	978
C 330 FORMAT (31X*E/NTOT = *1PE10.3* VOLT CM2 = *0PF6.3* VOLT/CM/TORR = 1*F6.3* KVOLT/CM/ATM*)	BOLTZ	979
C 340 FORMAT (* TOTAL ELECTRICAL = *1PE10.3,A11* 100.00 %,*15X*TOTAL PO WER INTO COLLISIONS, HEATING, STORAGE = *1PE10.3,0PF8.2* %*)	BOLTZ	980
C 350 FORMAT (1H1,48X,4A10/)	BOLTZ	981
C 360 FORMAT (F7.2)	BOLTZ	982
C 370 FORMAT (/1X*DISCHARGE POWER = *1PE10.3,A11,0PF7.2* %,*18X*POWER INTO INELASTIC E-MOLECULE COLLISIONS = *1PE10.3,0PF8.2* %*/1X*SOUR CE DEPOSITION = *1PE10.3,A11,0PF7.2* %,*34X*ELASTIC E-MOLECULE HEA TING = *1PE10.3,0PF8.2* %*/22X+-----*12X+-----*14X*D/DT(STORE ID ELECTRON KINETIC ENERGY) = E<U>ONE/DT = *1PE10.3,0PF8.2* %*)	BOLTZ	983
C 380 FORMAT (/48X*CALCULATION PARAMETERS USED:/*48X*MESH =*14*, EMAX =* 10PF6.2* EV, DE =*F5.3* EV.*)	BOLTZ	984
C 390 FORMAT (A1,64X*S(U = 0)/NE =*1PE10.3)	BOLTZ	985
C 400 FORMAT (/1X*DISCHARGE POWER = 0.0*46X*POWER INTO INELASTIC E-MOL ECULE COLLISIONS = *1PE10.3,A11/1X*SOURCE DEPOSITION = 0.0*62X*ELA STIC E-MOLECULE HEATING = *1PE10.3,A11/63X*D/DT(STORED ELECTRON KI NETIC ENERGY) = E<U>ONE/DT = *1PE10.3,A11)	BOLTZ	986
C 410 FORMAT (* TOTAL ELECTRICAL = 0.0*43X*TOTAL POWER INTO COLLISIONS, HEATING, STORAGE = *1PE10.3,A11)	BOLTZ	987
C 420 FORMAT (1H*,124X,F8.2)	BOLTZ	988
C 430 FORMAT (1H*64X,*S(U > 0)/NE =*1PE10.3)	BOLTZ	989
C 500 FORMAT (1H1,20(/)28X,*PROGRAM COMPLETED *,I3,* ITERATIONS IN *, IF4.0,* SECONDS WITHOUT SUCCESSFUL CONVERGENCE,*/28X,*REQUESTED LIM ITS WERE *,I3,* ITERATIONS AND *,F4.0,* SECONDS, WITH A CONVERGENC E CONDI-*/*28X,*ITION THAT THE CHANGE BETWEEN TWO ITERATIONS BE LESS	BOLTZ	990
	BOLTZ	991
	BOLTZ	992
	BOLTZ	993
	BOLTZ	994
	BOLTZ	995
	BOLTZ	996
	BOLTZ	997
	BOLTZ	998
	BOLTZ	999
	BOLTZ	1000
	BOLTZ	1001
	BOLTZ	1002
	BOLTZ	1003
	BOLTZ	1004
	BOLTZ	1005
	BOLTZ	1006
	BOLTZ	1007
	BOLTZ	1008
	BOLTZ	1009
	BOLTZ	1010
	BOLTZ	1011
	BOLTZ	1012
	BOLTZ	1013
	BOLTZ	1014
	BOLTZ	1015
	BOLTZ	1016
	BOLTZ	1017
	BOLTZ	1018
	BOLTZ	1019
	BOLTZ	1020
	BOLTZ	1021
	BOLTZ	1022
	BOLTZ	1023
	BOLTZ	1024
	BOLTZ	1025
	BOLTZ	1026
	BOLTZ	1027

```

        4 THAN EPS = *,1PE10.3***)          BOLTZ 1028
C      600 FORMAT (/28X,*DISTRIBUTION FUNCTION BECAME NEGATIVE AT SOME POINTS  BOLTZ 1029
C           IN LAST ITERATION*)          BOLTZ 1030
C   -----
C   CONVERGENCE FAILED FOR LIMITS PRESCRIBED. ERROR MESSAGE --          BOLTZ 1031
C
C   98 WRITE (6,500) ITER, TIME, ITMAX, TMAX, EPS          BOLTZ 1032
C   IF (ERROR) WRITE (6,600)          BOLTZ 1033
C   -----
C   99 RETURN          BOLTZ 1034
C   END          BOLTZ 1035
C
C   -----
C   SUBROUTINE SIMEQ (A, M, N, NI, SING)          BOLTZ 1036
C   -----
C   THIS SUBROUTINE WILL SOLVE AN N X N SYSTEM OF SIMULTANEOUS EQUA-          BOLTZ 1037
C   TIONS OF THE FORM A(I,J)X(J) = B(I). A IS DIMENSIONED A(M,NN) IN          BOLTZ 1038
C   THE MAIN PROGRAM. AND INPUT CONSISTS OF PUTTING THE N X N MATRIX          BOLTZ 1039
C   IN THE UPPER LEFT HAND BOX OF A. NI DIFFERENT B VECTORS CAN BE          BOLTZ 1040
C   SPECIFIED AS INPUT STORED IN SUCCESSIVE COLUMNS TO THE RIGHT OF          BOLTZ 1041
C   THE N X N MATRIX (IN S M, N+NI$NN). UPON OUTPUT, THE SOLUTION VEC-          BOLTZ 1042
C   TORS REPLACE THE INPUT VECTORS B. IF B IS AN N X N UNIT MATRIX ON          BOLTZ 1043
C   INPUT, IT WILL CONTAIN A-INVERSE ON OUTPUT. SING = TRUE IF THE
C   MATRIX IS SINGULAR.
C
C   -----
C   DIMENSION A(M,1)
LOGICAL SING
SING = .FALSE.
NPN1 = N + NI
DO 9 I = 1,N
Z1 = 0.
DO 2 J = I,N
X1 = ABS(A(J,I))
IF (Z1-X1) 1,2,2
1 Z1 = X1
I1 = J
2 CONTINUE
IF (Z1) 3,11,3
3 Z = A(I1,I)
A(I1,I) = A(I,I)
IP1 = I + 1
DO 4 L = IP1,NPN1
X = A(I1,L)
A(I1,L) = A(I,L)
4 A(I,L) = X/Z
DO 8 J = 1,N
IF (J-I) 5,8,5
5 IF (A(J,I).EQ.0.) GO TO 8
Z = -A(J,I)
DO 7 L = IP1,NPN1
7 A(J,L) = A(J,L) + Z*A(I,L)
8 CONTINUE
9 CONTINUE
10 RETURN
11 WRITE (6,12)
12 FORMAT (1H1,20(/),40X,*COEFFICIENT MATRIX IS SINGULAR*)
SING = .TRUE.
RETURN
END

```

SUBROUTINE GEAR (N, T, Y, SAVE, H, HMIN, HMAX, EPS, MF, YMAX,  
 IERROR, KFLAG, JSTART, MAXDER, M, PW) GEAR 2  
 ..... GEAR 3  
 ..... GEAR 4  
 ..... GEAR 5  
 ..... GEAR 6  
 ..... GEAR 7  
 ..... GEAR 8  
 ..... GEAR 9  
 ..... GEAR 10  
 ..... GEAR 11  
 ..... GEAR 12  
 ..... GEAR 13  
 ..... GEAR 14  
 ..... GEAR 15  
 ..... GEAR 16  
 ..... GEAR 17  
 ..... GEAR 18  
 ..... GEAR 19  
 ..... GEAR 20  
 ..... GEAR 21  
 ..... GEAR 22  
 ..... GEAR 23  
 ..... GEAR 24  
 ..... GEAR 25  
 ..... GEAR 26  
 ..... GEAR 27  
 ..... GEAR 28  
 ..... GEAR 29  
 ..... GEAR 30  
 ..... GEAR 31  
 ..... GEAR 32  
 ..... GEAR 33  
 ..... GEAR 34  
 ..... GEAR 35  
 ..... GEAR 36  
 ..... GEAR 37  
 ..... GEAR 38  
 ..... GEAR 39  
 ..... GEAR 40  
 ..... GEAR 41  
 ..... GEAR 42  
 ..... GEAR 43  
 ..... GEAR 44  
 ..... GEAR 45  
 ..... GEAR 46  
 ..... GEAR 47  
 ..... GEAR 48  
 ..... GEAR 49  
 ..... GEAR 50  
 ..... GEAR 51  
 ..... GEAR 52  
 ..... GEAR 53  
 ..... GEAR 54  
 ..... GEAR 55  
 ..... GEAR 56  
 ..... GEAR 57  
 ..... GEAR 58

C THIS SUBROUTINE WAS TAKEN FROM THE BOOK, NUMERICAL INITIAL VALUE  
 C PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS, BY C. WILLIAM GEAR,  
 C PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N. J., 1971, PP. 158-166.  
 C  
 C THIS SUBROUTINE INTEGRATES A SET OF N ORDINARY DIFFERENTIAL FIRST  
 C ORDER EQUATIONS OVER ONE STEP OF LENGTH H AT EACH CALL. H CAN BE  
 C SPECIFIED BY THE USER AT EACH STEP, BUT IT MAY BE INCREASED OR  
 C DECREASED BY THE PRESENT SUBROUTINE WITHIN THE RANGE HMIN TO HMAX  
 C IN ORDER TO ACHIEVE AS LARGE A STEP AS POSSIBLE WHILE NOT COMMIT-  
 CTING A SINGLE STEP ERROR WHICH IS LARGE THAN EPS IN THE L-2 NORM,  
 C WHERE EACH COMPONENT OF THE ERROR IS DIVIDED BY THE COMPONENTS OF  
 C YMAX. THE PROGRAM REQUIRES THREE SUBROUTINES NAMED:  
 C  
 C RATES (N, T, Y, DY) GEAR 20  
 C SIMEQ (PW, M, N, I, SING) GEAR 21  
 C JACOB (M, T, Y, PW) GEAR 22  
 C  
 C THE FIRST, RATES, EVALUATES THE DERIVATIVES OF THE DEPENDENT VARI- GEAR 23  
 C ABLES STORED IN Y(1,I) FOR I = 1 TO N, AND STORES THE DERIVATIVES GEAR 24  
 C IN THE VECTOR DY. THE SECOND, SIMEQ, IS CALLED ONLY IF THE METHOD GEAR 25  
 C FLAG MF IS SET TO 1 OR 2 FOR STIFF METHODS. IT MUST INVERT THE GEAR 26  
 C N X N MATRIX STORED IN THE ARRAY PW(M,M). IF THE INVERSION IS GEAR 27  
 C SUCCESSFUL, SING (SINGULARITY) IS RETURNED FALSE. JACOB IS USED GEAR 28  
 C ONLY IF MF IS 1, AND COMPUTES THE PARTIAL DERIVATIVES OF THE DIF- GEAR 29  
 C FERENTIAL EQUATIONS AS DESCRIBED UNDER THE MF PARAMETER. GEAR 30  
 C  
 C THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER IN THE ARRAY GEAR 31  
 C PW AND THE ARRAY SAVE. THE ARRAY PW IS USED ONLY TO HOLD THE MA- GEAR 32  
 C TRIX OF THE SAME NAME, BUT SAVE IS USED TO HOLD SEVERAL ARRAYS. GEAR 33  
 C THE REGIONS USED ARE:  
 C  
 C SAVE(J,I) -- IS USED AND ISISN IS USED TO SAVE THE VALUES OF GEAR 34  
 C Y IN CASE A STEP HAS TO BE REPEATED. GEAR 35  
 C  
 C SAVE(9,I) -- IS USED MAINLY TO HOLD THE CORRECTION TERMS IN GEAR 36  
 C THE CORRECTOR LOOP. GEAR 37  
 C  
 C SAVE(10,I) -- IS USED TO SAVE THE VALUES OF THE SUMS OF ALL GEAR 38  
 C OF THE CORRECTION TERMS IN THE PREVIOUS STEP GEAR 39  
 C AFTER THEY HAVE BEEN ACCUMULATED IN THE ARRAY GEAR 40  
 C ERROR IN THE CURRENT STEP. THIS ENABLES THE GEAR 41  
 C BACKWARDS DIFFERENCE OF ERROR TO BE FORMED. IT GEAR 42  
 C IS USED TO ESTIMATE THE STEP SIZE FOR ONE ORDER GEAR 43  
 C HIGHER THAN CURRENT. GEAR 44  
 C  
 C SAVE(N1+I,1) -- IS USED TO STORE THE DERIVATIVES WHEN THEY ARE GEAR 45  
 C COMPUTED BY RATES. IT IS ALSO ACCESSED AS GEAR 46  
 C SAVE(N2,I) AS A COMPLETE ARRAY. GEAR 47  
 C  
 C SAVE(N5+I,1) -- HOLDS THE DERIVATIVES DURING JACOBIAN EVALUA- GEAR 48  
 C TIONS. IT IS REFERENCED AS SAVE(N6,I) AS A COM- GEAR 49  
 C PLETE ARRAY. GEAR 50  
 C

C	INPUT AND OUTPUT PARAMETERS HAVE THE FOLLOWING MEANING --	GEAR	59
C	N THE NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS. N MAY C BE DECREASED ON LATER CALLS IF THE NUMBER OF ACTIVE EQUA- C TIONS REDUCES, BUT IT MUST NOT BE INCREASED WITHOUT CALL- C ING JSTART = 0.	GEAR	60
C	M DIMENSION DECLARATOR FOR THE JACOBIAN PW AND THE ARRAY Y C CONTAINING THE DEPENDENT VARIABLES AND THEIR DERIVATIVES.	GEAR	61
C	T THE INDEPENDENT VARIABLE.	GEAR	62
C	Y AN M X 8 ARRAY CONTAINING THE DEPENDENT VARIABLES AND C THEIR SCALED DERIVATIVES. Y(I,J+1) CONTAINS THE JTH C DERIVATIVE OF Y(I) SCALED BY H**J/FACTORIAL J, WHERE H IS C THE CURRENT STEP SIZE. ONLY Y(I,1) NEED BE PROVIDED BY C THE CALLING PROGRAM ON THE FIRST ENTRY. IF IT IS DESIRED C TO INTERPOLATE TO NON-MESH POINTS, THESE VALUES CAN BE C USED. IF THE CURRENT STEP SIZE IS H AND THE VALUE AT T+E C IS NEEDED, FORM S = E/H, AND THEN COMPUTE	GEAR	63
C	NO Y(I)[T+E] = SUMJ Y(I,J+1)*S**J J=0	GEAR	64
C	SAVE A BLOCK OF AT LEAST 12*N SCRATCH LOCATIONS USED BY THE C SUBROUTINES.	GEAR	65
C	H THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP. H MAY BE C ADJUSTED UP OR DOWN BY THE PROGRAM IN ORDER TO ACHIEVE AN C ECONOMICAL INTEGRATION. HOWEVER, IF THE H PROVIDED BY THE C USER DOES NOT CAUSE A LARGER ERROR THAN REQUESTED, IT WILL C BE USED. TO SAVE COMPUTER TIME, THE USER IS ADVISED TO C USE A FAIRLY SMALL STEP FOR THE FIRST CALL. IT WILL BE C AUTOMATICALLY INCREASED LATER.	GEAR	66
C	HMIN THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE INTEGRA- C TION. NOTE THAT ON STARTING THIS MUST BE MUCH SMALLER C THAN THE AVERAGE H EXPECTED SINCE A FIRST ORDER METHOD IS C USED INITIALLY.	GEAR	67
C	HMAX THE MAXIMUM SIZE TO WHICH THE STEP SIZE WILL BE INCREASED.	GEAR	68
C	EPS THE ERROR TEST CONSTANT. SINGLE STEP ERROR ESTIMATES DI- C VIDED BY YMAX(I) MUST BE LESS THAN THIS IN THE EUCLIDEAN NORM. THE STEP AND/OR ORDER IS ADJUSTED TO ACHIEVE THIS.	GEAR	69
C	MF THE METHOD INDICATOR. THE FOLLOWING ARE AVAILABLE:	GEAR	70
C	0 AN ADAMS PREDICTOR-CORRECTOR IS USED.	GEAR	71
C	1 A MULTISTEP METHOD SUITABLE FOR STIFF SYSTEMS IS C USED. IT WILL ALSO WORK FOR NON-STIFF SYSTEMS. HOWEVER, THE USER MUST PROVIDE A SUBROUTINE JACOB C WHICH EVALUATES THE PARTIAL DERIVATIVES OF THE DIFFERENTIAL EQUATIONS WITH RESPECT TO THE Y-S.	GEAR	72
C		GEAR	73
C		GEAR	74
C		GEAR	75
C		GEAR	76
C		GEAR	77
C		GEAR	78
C		GEAR	79
C		GEAR	80
C		GEAR	81
C		GEAR	82
C		GEAR	83
C		GEAR	84
C		GEAR	85
C		GEAR	86
C		GEAR	87
C		GEAR	88
C		GEAR	89
C		GEAR	90
C		GEAR	91
C		GEAR	92
C		GEAR	93
C		GEAR	94
C		GEAR	95
C		GEAR	96
C		GEAR	97
C		GEAR	98
C		GEAR	99
C		GEAR	100
C		GEAR	101
C		GEAR	102
C		GEAR	103
C		GEAR	104
C		GEAR	105
C		GEAR	106
C		GEAR	107
C		GEAR	108
C		GEAR	109
C		GEAR	110
C		GEAR	111
C		GEAR	112
C		GEAR	113
C		GEAR	114
C		GEAR	115

C	THIS IS DONE BY CALL JACOB (M, T, Y, PW(1,N+2)).	GEAR	116
C	THE JACOBIAN PHI(I,J), WHICH REPRESENTS THE PAR-	GEAR	117
C	TIAL OF THE ITH EQUATION WITH RESPECT TO THE JTH	GEAR	118
C	DEPENDENT VARIABLE, IS STORED IN THE UPPER LEFT	GEAR	119
C	N X N CORNER OF THAT PART OF THE ARRAY PW BEGIN-	GEAR	120
C	NING AT COLUMN (N+2).	GEAR	121
C	2 THE SAME AS CASE 1, EXCEPT THAT THIS SUBROUTINE	GEAR	122
C	COMPUTES THE PARTIAL DERIVATIVES BY NUMERICAL DIF-	GEAR	123
C	FERENCEING OF THE DERIVATIVES. HENCE, JACOB IS NOT	GEAR	124
C	CALLED.	GEAR	125
C	YMAX AN ARRAY OF N LOCATIONS WHICH CONTAINS THE MAXIMUM OF EACH	GEAR	126
C	Y SEEN SO FAR, AUTOMATICALLY UPDATED AFTER EACH COMPLETED	GEAR	127
C	STEP (UNLESS THE USER OVERRIDES IT BY CHANGING YMAX BEFORE	GEAR	128
C	A SUBSEQUENT CALL). ALL OF THE COMPONENTS OF YMAX SHOULD	GEAR	129
C	BE INITIALIZED TO 1.0 BEFORE THE FIRST ENTRY. (CF. ALSO	GEAR	130
C	THE DESCRIPTION FOR EPS, GIVEN ABOVE.)	GEAR	131
C	ERROR AN ARRAY OF N ELEMENTS WHICH CONTAIN THE ESTIMATED ONE	GEAR	132
C	STEP ERROR IN EACH COMPONENT.	GEAR	133
C	KFLAG A COMPLETION CODE WITH THE FOLLOWING MEANINGS--	GEAR	134
C	+1 THE STEP WAS SUCCESSFUL; T IS ADVANCED TO (T+H)	GEAR	135
C	AND RESULTS OF INTEGRATION FROM T TO (T+H) ARE	GEAR	136
C	RETURNED TO THE CALLING PROGRAM.	GEAR	137
C	-1 THE STEP WAS TAKEN WITH H = HMIN, BUT THE	GEAR	138
C	REQUESTED ERROR WAS NOT ACHIEVED. T IS ADVANCED	GEAR	139
C	TO (T+DT). CALLING PROGRAM MUST APPROVE FAILURE	GEAR	140
C	OF ACCURACY.	GEAR	141
C	-2 THE MAXIMUM ORDER SPECIFIED WAS FOUND TO BE TOO	GEAR	142
C	LARGE.	GEAR	143
C	-3 CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED FOR	GEAR	144
C	H > HMIN. T IS NOT ADVANCED. CALLING PROGRAM	GEAR	145
C	MUST DECREASE H AND HMIN AND TRY AGAIN.	GEAR	146
C	-4 THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED	GEAR	147
C	FOR THIS PROBLEM. T IS NOT ADVANCED. ETA MUST	GEAR	148
C	BE INCREASED.	GEAR	149
C	JSTART INPUT INDICATOR WITH THE FOLLOWING MEANINGS --	GEAR	150
C	-1 REPEAT THE LAST STEP WITH A NEW H	GEAR	151
C	0 PERFORM THE FIRST STEP. THE FIRST STEP MUST BE	GEAR	152
C	DONE WITH THIS VALUE OF JSTART SO THAT THE SUBROU-	GEAR	153
C	TINE CAN INITIALIZE ITSELF.	GEAR	154
C	+1 TAKE A NEW STEP CONTINUING FROM THE LAST.	GEAR	155
C	JSTART IS SET TO NO. THE CURRENT ORDER OF THE METHOD AT	GEAR	156
C	EXIT. NO IS ALSO THE ORDER OF THE MAXIMUM DERIVATIVE	GEAR	157
C	AVAILABLE.	GEAR	158
C	MAXDER THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE METHOD.	GEAR	159
C	SINCE THE ORDER IS EQUAL TO THE HIGHEST DERIVATIVE USED.	GEAR	160
C	THIS RESTRICTS THE ORDER. IT MUST BE LESS THAN 8 FOR	GEAR	161
C	ADAMS AND 7 FOR THE STIFF METHODS.	GEAR	162
C		GEAR	163
C		GEAR	164
C		GEAR	165
C		GEAR	166
C		GEAR	167
C		GEAR	168
C		GEAR	169
C		GEAR	170
C		GEAR	171
C		GEAR	172

```

C      PW      A BLOCK OF AT LEAST M*(2M+1) SCRATCH LOCATIONS.      GEAR    173
C.....                                         GEAR    174
C.....                                         GEAR    175
C.....                                         GEAR    176
C.....                                         GEAR    177
C.....                                         GEAR    178
C.....                                         GEAR    179
C.....                                         GEAR    180
C.....                                         GEAR    181
C.....                                         GEAR    182
C.....                                         GEAR    183
C.....                                         GEAR    184
C.....                                         GEAR    185
C.....                                         GEAR    186
C.....                                         GEAR    187
C.....                                         GEAR    188
C.....                                         GEAR    189
C.....                                         GEAR    190
C.....                                         GEAR    191
C.....                                         GEAR    192
C.....                                         GEAR    193
C.....                                         GEAR    194
C.....                                         GEAR    195
C.....                                         GEAR    196
C.....                                         GEAR    197
C.....                                         GEAR    198
C.....                                         GEAR    199
C.....                                         GEAR    200
C.....                                         GEAR    201
C.....                                         GEAR    202
C.....                                         GEAR    203
C.....                                         GEAR    204
C.....                                         GEAR    205
C.....                                         GEAR    206
C.....                                         GEAR    207
C.....                                         GEAR    208
C.....                                         GEAR    209
C.....                                         GEAR    210
C.....                                         GEAR    211
C.....                                         GEAR    212
C.....                                         GEAR    213
C.....                                         GEAR    214
C.....                                         GEAR    215
C.....                                         GEAR    216
C.....                                         GEAR    217
C.....                                         GEAR    218
C.....                                         GEAR    219
C.....                                         GEAR    220
C.....                                         GEAR    221
C.....                                         GEAR    222
C.....                                         GEAR    223
C.....                                         GEAR    224
C.....                                         GEAR    225
C.....                                         GEAR    226
C.....                                         GEAR    227
C.....                                         GEAR    228
C.....                                         GEAR    229

C..... DIMENSION Y(M,8), YMAX(1), SAVE(10,1), ERROR(1), PW(M,1), A(8),
C..... 1 PERTST(7,2,3)
C..... LOGICAL SING
C..... THE COEFFICIENTS IN PERTST ARE USED IN SELECTING THE STEP AND
C..... ORDER. THEREFORE, ONLY ABOUT ONE PERCENT ACCURACY IS NEEDED.
C..... DATA PERTST / 2.0, 4.5, 7.333, 10.42, 13.7, 17.15, 1.0,
C..... 1   2.0, 12.0, 24.0, 37.89, 53.33, 70.08, 87.97,
C..... 2   3.0, 6.0, 9.167, 12.5, 15.98, 1.0, 1.0,
C..... 3   12.0, 24.0, 37.89, 53.33, 70.08, 87.97, 1.0,
C..... 4   1.0, 1.0, 0.5, 0.1667, 0.04133, 0.008267, 1.0,
C..... 5   1.0, 1.0, 2.0, 1.0, 0.3157, 0.07407, 0.0139 /
C..... DATA A(2) / -1.0 /
C..... IRET = 1
C..... KFLAG = 1
C..... METHOD = MF+1
C..... IF (JSTART.LE.0) GO TO 140
C..... BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING H
C..... BY THE FACTOR R IF THE CALLER HAS CHANGED H. ALL VARIABLES DEPEND-
C..... ON H MUST ALSO BE CHANGED. E IS A COMPARISON FOR ERRORS OF
C..... THE CURRENT ORDER. NO. EUP IS TO TEST FOR INCREASING THE ORDER.
C..... EDWN FOR DECREASING THE ORDER. HNEW IS THE STEP SIZE THAT WAS
C..... USED ON THE LAST CALL.
C..... 100 DO 110 I = 1,N
C..... 100 DO 110 J = 1,K
C..... 110 SAVE(J,I) = Y(I,J)
C..... HOLD = HNEW
C..... IF (H.EQ.HOLD) GO TO 130
C..... 120 RACUM = H/HOLD
C..... IRET1 = 1
C..... GO TO 750
C..... 130 NUOLD = NU
C..... TULD = T
C..... RACUM = 1.0
C..... IF (JSTART.GT.0) GO TO 250
C..... GU TO 170
C..... 140 IF (JSTART.EQ.-1) GO TO 160
C..... ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL DERIVA-
C..... TIVES ARE CALCULATED.
C..... NO = 1
C..... N1 = N*10
C..... N2 = N1 + 1
C..... N5 = N1 + N
C..... N6 = N5 + 1

```

```

C -----
C      CALL DNDT (N, T, Y, SAVE(N2,1))
C -----
C
C      DO 150 I = 1,N
150 Y(I,2) = SAVE(N1+I,1)*H
HNEW = H
K = 2
GO TO 100
C      REPEAT LAST STEP BY RESTORING SAVED INFORMATION --
160 IF (NQ.EQ.NQOLD) JSTART = 1
      T = TOLD
      NQ = NQOLD
      K = NQ+1
      GO TO 120
C
C      SET THE COEFFICIENTS THAT DETERMINE THE ORDER AND THE METHOD TYPE. GEAR 246
C      CHECK FOR EXCESSIVE ORDER. THE LAST TWO STATEMENTS OF THIS SEC- GEAR 247
C      TION SET IWEVAL > 0 IF PW IS TO BE REEVALUATED BECAUSE OF THE OR- GEAR 248
C      DER CHANGE, AND THEN REPEAT THE INTEGRATION STEP IF IT HAS NOT YET GEAR 249
C      BEEN DONE (IRET = 1) OR SKIP TO A FINAL SCALING BEFORE EXIT IF IT GEAR 250
C      HAS BEEN COMPLETED (IRET = 2). GEAR 251
C
C      170 IF (MF.EQ.0) GO TO 180
      IF (NQ.GT.6) GO TO 190
      GO TO (221,222,223,224,225,226), NO
180 IF (NQ.GT.7) GO TO 190
      GO TO (211,212,213,214,215,216,217), NO
190 KFLAG = -2
      RETURN
C
C      211 A(1) = -1.0
      GO TO 230
212 A(1) = A(3) = -0.5
      GO TO 230
213 A(1) = -0.41666666666667
C      A(1) = -5./12.
      A(3) = -0.75
      A(4) = -0.16666666666667
C      A(4) = -1./6.
      GO TO 230
214 A(1) = -.375
      A(3) = -0.91666666666667
C      A(3) = -11./12.
      A(4) = -0.33333333333333
C      A(4) = -1./3.
      A(5) = -0.41666666666667E-01
C      A(5) = -1./24.
      GO TO 230
215 A(1) = -0.3486111111111111
C      A(1) = -251./720.
      A(3) = -1.04166666666667
C      A(3) = -25./24.
      A(4) = -0.4861111111111111
C      A(4) = -35./72.
      A(5) = -0.10416666666667
C      A(5) = -5./48.

```

C	A(6) = -0.83333333333333E-02	GEAR	287
C	A(6) = -1./120.	GEAR	288
	GO TO 230	GEAR	289
216	A(1) = -0.32986111111111	GEAR	290
C	A(1) = -95./288.	GEAR	291
	A(3) = -1.1416666666667	GEAR	292
C	A(3) = -137./120.	GEAR	293
	A(4) = -0.625	GEAR	294
C	A(4) = -5./8.	GEAR	295
	A(5) = -0.17708333333333	GEAR	296
C	A(5) = -17./96.	GEAR	297
	A(6) = -0.025	GEAR	298
C	A(6) = -1./40.	GEAR	299
	A(7) = -0.13888888888889E-02	GEAR	300
C	A(7) = -1./720.	GEAR	301
	GO TO 230	GEAR	302
217	A(1) = -0.31559193121693	GEAR	303
C	A(1) = -19087./60480.	GEAR	304
	A(3) = -1.225	GEAR	305
C	A(3) = -49./40.	GEAR	306
	A(4) = -0.75185185185185	GEAR	307
C	A(4) = -203./270.	GEAR	308
	A(5) = -0.25520833333333	GEAR	309
C	A(5) = -49./192.	GEAR	310
	A(6) = -0.48611111111111E-01	GEAR	311
C	A(6) = -7./144.	GEAR	312
	A(7) = -0.48611111111111E-02	GEAR	313
C	A(7) = -7./1440.	GEAR	314
	A(8) = -0.19841269841270E-03	GEAR	315
C	A(8) = -1./5040.	GEAR	316
	GO TO 230	GEAR	317
221	A(1) = -1.	GEAR	318
	GO TO 230	GEAR	319
222	A(1) = -0.666666666666667	GEAR	320
C	A(1) = -2./3.	GEAR	321
	A(3) = -0.33333333333333	GEAR	322
C	A(3) = -1./3.	GEAR	323
	GO TO 230	GEAR	324
223	A(1) = -0.54545454545454	GEAR	325
	A(3) = A(1)	GEAR	326
C	A(1) = A(3) = -6./11.	GEAR	327
	A(4) = -0.90909090909091E-01	GEAR	328
C	A(4) = -1./11.	GEAR	329
	GO TO 230	GEAR	330
224	A(1) = -0.48	GEAR	331
	A(3) = -0.7	GEAR	332
	A(4) = -0.2	GEAR	333
	A(5) = -0.02	GEAR	334
	GO TO 230	GEAR	335
225	A(1) = -0.43795620437956	GEAR	336
C	A(1) = -120./274.	GEAR	337
	A(3) = -0.82116788321168	GEAR	338
C	A(3) = -225./274.	GEAR	339
	A(4) = -0.31021897810219	GEAR	340
C	A(4) = -85./274.	GEAR	341
	A(5) = -0.54744525547445E-01	GEAR	342
C	A(5) = -15./274.	GEAR	343

```

A(6) = -0.36496350364964E-02          GEAR    344
C   A(6) = -1./274.                      GEAR    345
C   GO TO 230                           GEAR    346
226 A(1) = -0.40816326530612          GEAR    347
C   A(1) = -180./441.                     GEAR    348
C   A(3) = -0.92063492063492          GEAR    349
C   A(3) = -58./63.                      GEAR    350
C   A(4) = -0.416666666666667         GEAR    351
C   A(4) = -15./36.                      GEAR    352
C   A(5) = -0.99206349206349E-01       GEAR    353
C   A(5) = -25./252.                     GEAR    354
C   A(6) = -0.11904761904762E-01       GEAR    355
C   A(6) = -3./252.                      GEAR    356
C   A(7) = -0.56689342403628E-03       GEAR    357
C   A(7) = -1./1764.                     GEAR    358
C
230 K = NQ+1                           GEAR    359
  IDOUB = K                            GEAR    360
  MTYP = (4-MF)/2                      GEAR    361
C   MTYP = 1 (STIFF, MF = 1 OR 2), MTYP = 2 (ADAMS-Moulton). GEAR    362
  ENQ2 = 0.5/(NQ+1)                    GEAR    363
  ENQ3 = 0.5/(NQ+2)                    GEAR    364
  ENQ1 = 0.5/N                         GEAR    365
  PEPSh = EPS                          GEAR    366
  EUP = (PERTST(NQ,MTYP,2)*PEPSh)**2   GEAR    367
  E = (PERTST(NQ,MTYP,1)*PEPSh)**2     GEAR    368
  EDWN = (PERTST(NQ,MTYP,3)*PEPSh)**2   GEAR    369
  IF (EDWN.EQ.0) GO TO 780            GEAR    370
  BND = EPS*ENQ3/N                    GEAR    371
  IWEVAL = MF                          GEAR    372
  GO TO (250+680), IRET              GEAR    373
C
C   THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY MULTI-
C   PLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE MATRIX. GEAR    374
C
250 T = T+H                           GEAR    375
  DO 260 J = 2*K                      GEAR    376
  DO 260 J1 = J,K                      GEAR    377
  J2 = K-J1+J-1                        GEAR    378
  DO 260 I = 1,N                      GEAR    379
  260 Y(I,J2) = Y(I,J2) + Y(I,J2+1)   GEAR    380
C
C   UP TO 3 CORRECTOR ITERATIONS ARE TAKEN. CONVERGENCE IS TESTED BY GEAR    381
C   REQUIRING CHANGES TO BE LESS THAN BND WHICH IS DEPENDENT ON THE GEAR    382
C   ERROR TEST CONSTANT. THE SUM OF THE CORRECTIONS IS ACCUMULATED IN GEAR    383
C   THE ARRAY ERROR(I). IT IS EQUAL TO THE KTH DERIVATIVE OF Y MULTI- GEAR    384
C   PLIED BY H**K/FACTORIAL(K-1)*A(K)), AND IS THEREFORE PROPOR- GEAR    385
C   TIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER OF H PRESENT (H**K GEAR    386
C
  DO 270 I = 1,N                      GEAR    387
  270 ERROR(I) = 0.                      GEAR    388
C
  DO 430 L = 1,3                      GEAR    389
C
C   -----
C   CALL DNOT (N, T, Y, SAVE(N2+1))      GEAR    390
C   -----

```

```

C      GO TO (280+300,320), METHOD          GEAR    401
C      THIS SECTION IS ENCOUNTERED ONLY FOR MF = 0 --  GEAR    402
C
C      280 DO 290 I = 1,N          GEAR    403
C      290 SAVE(9,I) = Y(I,2) - SAVE(N1+I,1)*H          GEAR    404
C      GO TO 410          GEAR    405
C
C      -----          GEAR    406
C      EVALUATE THE JACOBIAN AND PLACE IT IN AN N X N BOX IN THE UPPER  GEAR    407
C      LEFT HAND CORNER OF THAT PART OF PW BEGINNING WITH COLUMN (N+2).  GEAR    408
C
C      -----          GEAR    409
C      EVALUATE THE JACOBIAN BY A CALL TO AN EXTERNAL SUBROUTINE (THIS  GEAR    410
C      SECTION IS ENCOUNTERED ONLY IF MF = 1) --          GEAR    411
C
C      300 IF (IWEVAL.LT.1) GO TO 380          GEAR    412
C
C      -----          GEAR    413
C      CALL JACOB (M, T, Y, PW(1,N+2))          GEAR    414
C
C      -----          GEAR    415
C      R = A(1)*H          GEAR    416
C      DO 310 I = 1,N          GEAR    417
C      DO 310 J = 1,N          GEAR    418
C      JPN1 = J+N+1          GEAR    419
C      310 PW(I,JPN1) = PW(I,JPN1)*R          GEAR    420
C      GO TO 360          GEAR    421
C
C      -----          GEAR    422
C
C      EVALUATE THE JACOBIAN INTO PW BY NUMERICAL DIFFERENCING. R IS THE  GEAR    423
C      CHANGE MADE TO THE ELEMENT OF Y. IT IS EPS RELATIVE TO Y WITH A  GEAR    424
C      MINIMUM OF EPS**2. THIS SECTION IS ENCOUNTERED ONLY IF MF = 2 --  GEAR    425
C
C      320 IF (IWEVAL.LT.1) GO TO 380          GEAR    426
C      DO 330 I = 1,N          GEAR    427
C      330 SAVE(9,I) = Y(I,1)          GEAR    428
C      DO 350 J = 1,N          GEAR    429
C      R = EPS*AMAX1(EPS,ABS(SAVE(9,J)))          GEAR    430
C      Y(J,1) = Y(J,1) + R          GEAR    431
C      D = A(1)*H/R          GEAR    432
C
C      -----          GEAR    433
C      CALL DNDF (N, T, Y, SAVE(N6+1))          GEAR    434
C
C      -----          GEAR    435
C      JPN1 = J+N+1          GEAR    436
C      DO 340 I = 1,N          GEAR    437
C      340 PW(I,JPN1) = (SAVE(N5+I,1) - SAVE(N1+I,1))*D          GEAR    438
C      350 Y(J,1) = SAVE(9,J)          GEAR    439
C
C      -----          GEAR    440
C
C      IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE WITH  GEAR    441
C      CONVERGENCE, PW IS REEVALUATED PRIOR TO STARTING THE CORRECTOR          GEAR    442
C
C      -----          GEAR    443
C
C      -----          GEAR    444
C
C      -----          GEAR    445
C
C      -----          GEAR    446
C
C      -----          GEAR    447
C
C      -----          GEAR    448
C
C      -----          GEAR    449
C
C      -----          GEAR    450
C
C      -----          GEAR    451
C
C      -----          GEAR    452
C
C      -----          GEAR    453
C
C      -----          GEAR    454
C
C      -----          GEAR    455
C
C      -----          GEAR    456
C
C      -----          GEAR    457

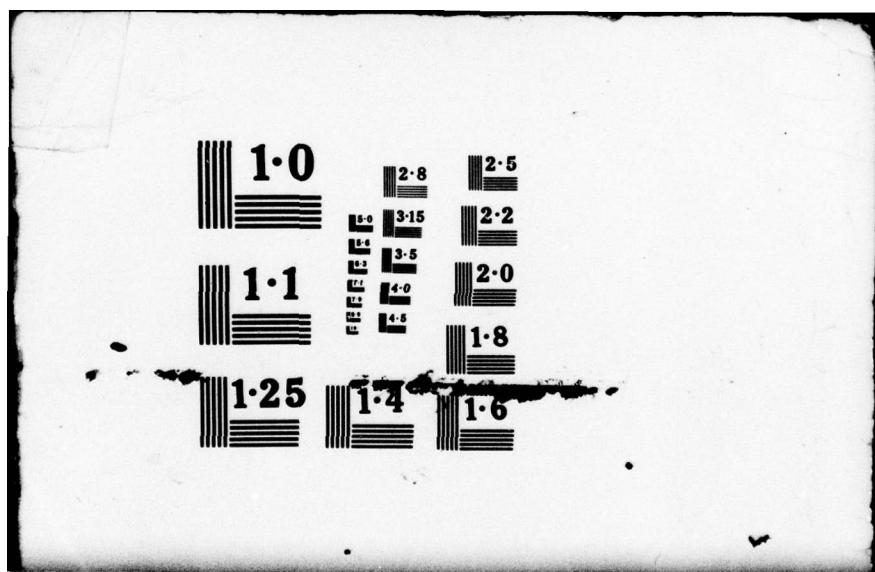
```

AD-A066 156 NORTHROP RESEARCH AND TECHNOLOGY CENTER PALOS VERDES --ETC F/G 20/5  
THEORETICAL MODELING OF MOLECULAR AND ELECTRON KINETIC PROCESSE--ETC(U)  
JAN 79 W B LACINA N00014-78-C-0499  
UNCLASSIFIED NRTC-79-TR-VOL-2 NL

2 of 2  
ADA  
066156



END  
DATE  
FILMED  
5 79  
DDC



```

C      ITERATION IN THE CASE OF STIFF METHODS. IWEVAL IS THEN SET TO -1   GEAR    458
C      AS AN INDICATOR THAT IT HAS BEEN DONE.                                GEAR    459
C
C      360 IWEVAL = -1                                                     GEAR    460
C
C      ADD THE IDENTITY MATRIX TO THE JACOBIAN. THIS SECTION IS ENCOUN- GEAR    461
C      TERED ONLY FOR MF = 1 OR 2 --                                         GEAR    462
C
C      DO 370 I = 1,N                                                       GEAR    463
C      IPN1 = I+N+1                                                       GEAR    464
C      370 PW(I,IPN1) = 1.0 + PW(I,IPN1)                                    GEAR    465
C
C      FETCH (I + PHI), SHIFT IT TO THE LEFT, CALCULATE RHS VECTOR -- GEAR    466
C      380 DO 390 I = 1,N                                                   GEAR    467
C      PW(I,N+1) = SAVE(NS+I+1) = Y(I,2) - SAVE(N+I+1)*H                 GEAR    468
C      DO 390 J = 1,N                                                       GEAR    469
C      JPN1 = J+N+1                                                       GEAR    470
C      390 PW(I,J) = PW(I,JPN1)                                           GEAR    471
C
C      -----
C      CALL SIMEQ (PW, M, N, 1, SING)                                     GEAR    472
C      -----
C
C      IF (SING) GO TO 440                                              GEAR    473
C
C      DO 400 I = 1,N                                                       GEAR    474
C      400 SAVE(9,I) = PW(I,N+1)                                           GEAR    475
C
C      *****
C      CORRECT AND SEE IF ALL OF THE CHANGES ARE LESS THAN BND RELATIVE GEAR    476
C      TO YMAX. IF SO, THE CORRECTOR IS SAID TO HAVE CONVERGED.          GEAR    477
C
C      410 NT = N                                                       GEAR    478
C      DO 420 I = 1,N                                                       GEAR    479
C      Y(I,1) = Y(I,1) + A(I)*SAVE(9,I)                                     GEAR    480
C      Y(I,2) = Y(I,2) - SAVE(9,I)                                         GEAR    481
C      ERROR(I) = ERROR(I) + SAVE(9,I)                                     GEAR    482
C      IF (ABS(SAVE(9,I)).LE.(BND*YMAX(I))) NT = NT-1                  GEAR    483
C      420 CONTINUE                                              GEAR    484
C      IF (NT.LE.0) GO TO 490                                            GEAR    485
C      430 CONTINUE                                              GEAR    486
C
C      *****
C      THE CORRECTOR ITERATION FAILED TO CONVERGE IN THREE TRIES. VARI- GEAR    487
C      OUS POSSIBILITIES ARE CHECKED FOR. IF M IS ALREADY HMIN AND THIS GEAR    488
C      IS EITHER ADAMS METHOD OR THE STIFF METHOD IN WHICH THE MATRIX PW GEAR    489
C      HAS ALREADY BEEN REEVALUATED, A NO-CONVERGENCE EXIT IS TAKEN.      GEAR    490
C      OTHERWISE, THE MATRIX PW IS REEVALUATED AND/OR THE STEP IS RE- GEAR    491
C      DUCED TO TRY AND GET CONVERGENCE.                                     GEAR    492
C
C      *****
C
C      440 T = TOLD
C      IF ((H.LE.(HMIN*.00001)).AND.((IWEVAL-MTYP).LT.-1)) GO TO 460
C      IF ((MF.EQ.0).OR.(IWEVAL.NE.0)) RACUM = RACUM*.25

```

```

IWEVAL = MF          GEAR      515
IRET1 = 2           GEAR      516
GO TO 750           GEAR      517
460 KFLAG = -3      GEAR      518
470 DO 480 I = 1,N   GEAR      519
    DO 480 J = 1,K   GEAR      520
480 Y(I,J) = SAVE(J,I) GEAR      521
    H = HOLD         GEAR      522
    NU = NQOLD        GEAR      523
    JSTART = NU       GEAR      524
    RETURN            GEAR      525
C .....             GEAR      526
C .....             GEAR      527
C .....             GEAR      528
C THE CORRECTOR CONVERGED AND CONTROL IS PASSED TO STATEMENT 520
C IF THE ERROR TEST IS O.K. (OR TO 540 OTHERWISE.) IF THE STEP IS
C O.K., IT IS ACCEPTED. IF IDOUB HAS BEEN REDUCED TO ONE, A TEST
C IS MADE TO SEE IF THE STEP CAN BE INCREASED AT THE CURRENT ORDER
C OR BY GOING TO ONE HIGHER OR ONE LOWER. SUCH A CHANGE IS ONLY
C MADE IF THE STEP CAN BE INCREASED BY AT LEAST 1.1. IF NO CHANGE
C IS POSSIBLE, IDOUB IS SET TO 10 TO PREVENT FURTHER TESTING FOR 10
C STEPS. IF A CHANGE IS POSSIBLE, IT IS MADE, AND IDOUB IS SET TO
C NQ+1 TO PREVENT FURTHER TESTING FOR THAT NUMBER OF STEPS. IF THE
C ERROR WAS TOO LARGE, THE OPTIMUM STEP SIZE FOR THIS OR LOWER
C ORDER IS COMPUTED, AND THE STEP RETRIED. IF IT SHOULD FAIL TWICE
C MORE, IT IS AN INDICATION THAT THE DERIVATIVES THAT HAVE ACCUMU-
C LATED IN THE Y ARRAY HAVE ERRORS OF THE WRONG ORDER SO THE FIRST
C DERIVATIVES ARE RECOMPUTED AND THE ORDER IS SET TO 1.
C .....             GEAR      529
C .....             GEAR      530
C .....             GEAR      531
C .....             GEAR      532
C .....             GEAR      533
C .....             GEAR      534
C .....             GEAR      535
C .....             GEAR      536
C .....             GEAR      537
C .....             GEAR      538
C .....             GEAR      539
C .....             GEAR      540
C .....             GEAR      541
C .....             GEAR      542
C .....             GEAR      543
C .....             GEAR      544
C .....             GEAR      545
490 D = 0.           GEAR      546
    DO 500 I = 1,N   GEAR      547
500 D = D + (ERROR(I)/YMAX(I))**2 GEAR      548
    IWEVAL = 0        GEAR      549
    IF (D.GT.E) GO TO 540 GEAR      550
    IF (K.LT.3) GO TO 520 GEAR      551
C COMPLETE THE CORRECTION OF THE HIGHER ORDER DERIVATIVES AFTER A
C SUCCESSFUL STEP. GEAR      552
    DO 510 J = 3,K   GEAR      553
    DO 510 I = 1,N   GEAR      554
510 Y(I,J) = Y(I,J) + A(J)*ERROR(I) GEAR      555
520 KFLAG = 1        GEAR      556
    HNEW = H          GEAR      557
    IF (IDOUB.LE.1) GO TO 550 GEAR      558
    IDOUB = IDOUB-1   GEAR      559
    IF (IDOUB.GT.1) GO TO 700 GEAR      560
    DO 530 I = 1,N   GEAR      561
530 SAVE(10+I) = ERROR(I) GEAR      562
    GO TO 700         GEAR      563
C .....             GEAR      564
C .....             GEAR      565
C REDUCE THE FAILURE FLAG COUNT TO CHECK FOR MULTIPLE FAILURES. GEAR      566
C RESTORE T TO ITS ORIGINAL VALUE AND TRY AGAIN UNLESS THERE HAVE GEAR      567
C BEEN THREE FAILURES. IN THAT CASE, THE DERIVATIVES ARE ASSUMED GEAR      568
C TO HAVE ACCUMULATED ERRORS SO A RESTART FROM THE CURRENT VALUE OF GEAR      569
C Y IS TRIED.        GEAR      570
C .....             GEAR      571

```

```

540 KFLAG = KFLAG-2 GEAR 572
  IF (H.LE.(HMIN*1.00001)) GO TO 740 GEAR 573
  T = TOLD GEAR 574
  IF (KFLAG.LE.-5) GO TO 720 GEAR 575
C GEAR 576
C PR1, PR2, AND PR3 WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE GEAR 577
C SHOULD BE DIVIDED AT ORDER ONE LOWER, AT THIS ORDER, AND AT ORDER GEAR 578
C ONE HIGHER, RESPECTIVELY. GEAR 579
C GEAR 580
C 550 PR2 = (D/E)**ENQ2*1.2 GEAR 581
  PR3 = 1.E 20 GEAR 582
  IF ((NQ.GE.MAXDER).OR.(KFLAG.LE.-1)) GO TO 570 GEAR 583
  D = 0. GEAR 584
  DO 560 I = 1,N GEAR 585
  560 D = D + ((ERROR(I) - SAVE(10+I))/YMAX(I))**2 GEAR 586
  PR3 = (D/EUP)**ENQ3*1.4 GEAR 587
  570 PR1 = 1.E 20 GEAR 588
  IF (NQ.LE.1) GO TO 590 GEAR 589
  D = 0. GEAR 590
  DO 580 I = 1,N GEAR 591
  580 D = D + (Y(I,K)/YMAX(I))**2 GEAR 592
  PR1 = (D/EDWN)**ENQ1*1.3 GEAR 593
  590 CONTINUE GEAR 594
  IF (PR2.LE.PR3) GO TO 650 GEAR 595
  IF (PR3.LT.PR1) GO TO 660 GEAR 596
  600 R = 1.0/AMAX1(PR1+1.E-04) GEAR 597
  NEWQ = NQ-1 GEAR 598
  610 IDOUB = 10 GEAR 599
  IF ((KFLAG.EQ.1).AND.(R.LT.1.1)) GO TO 700 GEAR 600
  IF (NEWQ.LE.NQ) GO TO 630 GEAR 601
C COMPUTE ONE ADDITIONAL SCALED DERIVATIVE IF ORDER IS INCREASED. GEAR 602
  DO 620 I = 1,N GEAR 603
  620 Y(I,NEWQ+1) = ERROR(I)*A(K)/K GEAR 604
  630 K = NEWQ+1 GEAR 605
  IF (KFLAG.EQ.1) GO TO 670 GEAR 606
  RACUM = RACUM*R GEAR 607
  IRET1 = 3 GEAR 608
  GO TO 750 GEAR 609
  640 IF (NEWQ.EQ.NQ) GO TO 250 GEAR 610
  NQ = NEWQ GEAR 611
  GO TO 170 GEAR 612
  650 IF (PR2.GT.PR1) GO TO 600 GEAR 613
  NEWQ = NQ GEAR 614
  R = 1.0/AMAX1(PR2+1.E-04) GEAR 615
  GO TO 610 GEAR 616
  660 R = 1.0/AMAX1(PR3+1.E-04) GEAR 617
  NEWQ = NQ+1 GEAR 618
  GO TO 610 GEAR 619
  670 IRET = 2 GEAR 620
  R = AMIN1(R+HMAX/ABS(H)) GEAR 621
  H = H*R GEAR 622
  HNEW = H GEAR 623
  IF (NQ.EQ.NEWQ) GO TO 680 GEAR 624
  NQ = NEWQ GEAR 625
  GO TO 170 GEAR 626
  680 R1 = 1. GEAR 627
  DO 690 J = 2,K GEAR 628

```

```

R1 = R1*R
DO 690 I = 1,N
690 Y(I,J) = Y(I,J)*R1
IDOUB = K
DO 710 I = 1,N
710 YMAX(I) = AMAX1(YMAX(I), ABS(Y(I,I)))
JSTART = NO
RETURN
720 IF (NO.EQ.1) GO TO 780
C
C -----
CALL DNDT (N, T, Y, SAVE(N2+1))
C -----
C
R = H/HOLD
DO 730 I = 1,N
Y(I,1) = SAVE(I,1)
SAVE(2,I) = HOLD*SAVE(N1+I,1)
730 Y(I,2) = SAVE(2,I)*R
NU = 1
KFLAG = 1
GO TO 170
740 KFLAG = -1
HNEW = H
JSTART = NO
RETURN
C
C THIS SECTION SCALES ALL VARIABLES CONNECTED WITH H AND RETURNS
C TO THE ENTERING SECTION.
C
750 RACUM = AMAX1(ABS(HMIN/HOLD)*RACUM)
RACUM = AMIN1(RACUM,ABS(HMAX/HOLD))
R1 = 1.0
DO 760 J = 2,K
R1 = R1*RACUM
DO 760 I = 1,N
760 Y(I,J) = SAVE(J,I)*R1
H = HOLD*RACUM
DO 770 I = 1,N
770 Y(I,1) = SAVE(I,1)
IDOUB = K
GO TO (130+250,640)+IRET1
780 KFLAG = -4
GO TO 470
RETURN
END
      GEAR 629
      GEAR 630
      GEAR 631
      GEAR 632
      GEAR 633
      GEAR 634
      GEAR 635
      GEAR 636
      GEAR 637
      GEAR 638
      GEAR 639
      GEAR 640
      GEAR 641
      GEAR 642
      GEAR 643
      GEAR 644
      GEAR 645
      GEAR 646
      GEAR 647
      GEAR 648
      GEAR 649
      GEAR 650
      GEAR 651
      GEAR 652
      GEAR 653
      GEAR 654
      GEAR 655
      GEAR 656
      GEAR 657
      GEAR 658
      GEAR 659
      GEAR 660
      GEAR 661
      GEAR 662
      GEAR 663
      GEAR 664
      GEAR 665
      GEAR 666
      GEAR 667
      GEAR 668
      GEAR 669
      GEAR 670
      GEAR 671
      GEAR 672
      GEAR 673
      GEAR 674

```

```

SUBROUTINE PLOT (MM, MP, MULT, Y, Y0, DY, X, X0, DX, SCALEX,
1 SCALEY, SAME, CLEAR, CENTER, NAME, NP, IP)          PLOT    2
C
C .....                                         PLOT    3
C
C THIS PROGRAM GENERATES LINEAR, SEMILOG, OR LOGLOG PLOTS FOR UP TO 10 PLOT    4
C VECTORS Y(I,J), J = 1,2,...,NP, PROVIDED BY AN ARRAY DIMENSIONED PLOT    5
C Y(MM,...) IN THE CALLING PROGRAM. EACH VECTOR IS CONSIDERED TO BE A PLOT    6
C FUNCTION OF AN INDEPENDENT VARIABLE X(I), DEFINED BELOW. ALL PLOTS PLOT    7
C ARE GENERATED FOR EACH VECTOR BY SAMPLING MP POINTS, TAKEN WITH A RE- PLOT    8
C PETITION INDEX MULT (I.E., I = 1, (1*MULT), ..., [1 + (MP-1)*MULT]). PLOT    9
C HORIZONTALLY, THE PLOTTING RESOLUTION CONSISTS OF A MAXIMUM OF 50 PLOT   10
C SUBINTERVALS, CORRESPONDING TO 51 POINTS. IF MP ≤ 51, MP POINTS ARE PLOT   11
C PLOTTED. THERE ARE OPTIONS (AS WELL AS PROGRAM DEFAULTS) FOR SPECI- PLOT   12
C FFYING INITIAL VALUES AND TICK MARK INTERVALS EITHER BY DIRECT INPUT, PLOT   13
C OR BY INTERNAL AUTOMATIC SCALING. AUTOMATIC SCALING, WHICH IS ACCOM- PLOT   14
C PLISHED BY SUBROUTINES "AXIS" AND "INTERP", GENERATES CONVENIENT IN- PLOT   15
C TEGER VALUES FOR INITIAL VALUES AND TICK MARK INTERVALS. IF SCALEX PLOT   16
C = TRUE IS SPECIFIED, AUTOMATIC SCALING OF THE HORIZONTAL AXIS OCCURS. PLOT   17
C IF MP > 51, THE PROGRAM DEFAULTS TO AUTOMATIC SCALING FOR THE X-AXIS, PLOT   18
C AND INTERNALLY GENERATES, BY INTERPOLATION, 51 POINTS TO BE PLOTTED. PLOT   19
C IT IS ASSUMED THAT THE VECTOR X IS DEFINED BY X(I) = X0 + (I-1)*DX, PLOT   20
C IN WHICH CASE, THE X-ORIGIN IS X0 AND THE TICK MARK INTERVAL IS 5*DX. PLOT   21
C HOWEVER, THERE ARE TWO EXCEPTIONS IF DX = 0: 1) FOR AUTOMATIC X-AXIS PLOT   22
C SCALING, THE INPUT VECTOR X(I) IS USED, AND 2) IF MP ≤ 51, IT IS AS- PLOT   23
C SUMED THAT THE INDEPENDENT VARIABLE IS JUST THE INTEGER I. (THUS, IF PLOT   24
C DATA IS DEFINED OVER A NONUNIFORM GRID X(I), SCALEX = TRUE AND DX = 0 PLOT   25
C SHOULD BE SPECIFIED BY THE CALLING PROGRAM.) IF SCALEY = TRUE, AUTO- PLOT   26
C MATIC SCALING OF THE VERTICAL AXIS OCCURS FOR THE PLOT(S). IF SCALEY PLOT   27
C = FALSE, THE ORIGIN(S) AND TICK MARK INTERVAL(S) FOR THE VERTICAL PLOT   28
C AXIS (DIVIDED INTO TEN TICKS) ARE TAKEN TO BE THOSE SPECIFIED BY THE PLOT   29
C VECTORS Y0(2) AND DY(2). UNLESS DY(1) OR DY(2) = 0, IN WHICH CASE PLOT   30
C ONE, BOTH, OR ALL VECTORS ARE AUTOMATICALLY SCALED BY DEFAULT. IF PLOT   31
C SAME = TRUE, PLOT(S) ARE SCALED TOGETHER USING SPECIFIED VALUES Y0(1) PLOT   32
C AND DY(1) IF DY(2) ≠ 0, OR AUTOMATIC SCALE VALUES IF DY(1) = 0. IN PLOT   33
C THE CASE THAT SAME = TRUE, A SINGLE (COMMON) VERTICAL SCALE FOR THE PLOT   34
C PLOT(S) WILL APPEAR AT THE LEFT. IF SAME = FALSE AND TWO PLOTS (NP = PLOT   35
C 2) ARE REQUESTED, DIFFERENT LEFT AND RIGHT HAND SCALING WILL APPEAR. PLOT   36
C PLOTS WILL BE SMOOTHED BY INTERPOLATION, AND WILL BE CENTERED ON THE PLOT   37
C PAGE (IF REQUESTED).
C
C MISCELLANEOUS CONDITIONS --
C
C 1) IF THE DATA HAS A TOTAL RANGE THAT IS LESS THAN 1.E-04 TIMES ITS PLOT   38
C AVERAGE VALUE, THE SMALL (AC) VARIATIONS WILL BE PLOTTED WITH A PLOT   39
C (DC) BASELINE VALUE SPECIFIED.
C
C 2) PROGRAM DEFAULTS TO FIRST 10 VECTORS IF NP > 10. PLOT   40
C
C 3) PROGRAM DEFAULTS TO PLOT OF FIRST VECTOR ONLY IF HISTOGRAM IS PLOT   41
C REQUESTED (CLEAR = FALSE).
C
C 4) IF NP = 2 AND SAME = TRUE, PROGRAM DEFAULTS TO SAME = FALSE IF PLOT   42
C THE TOTAL RANGE OF THE SMALLER OF THE TWO VECTORS IS LESS THAN PLOT   43
C FIVE TIMES THE TOTAL RANGE OF THE TWO VECTORS SCALED TOGETHER.
C
C 5) IF NP ≠ 2, ONLY LEFT-HAND TICK MARKS ARE GENERATED ON THE PLOT   44
C
C

```

C	VERTICAL SIDE.	PLOT	59
C	6) PAGE SKIP AND HEADING TITLES MUST BE GENERATED IN THE CALLING PROGRAM.	PLOT	60
C	7) IF ANY LOG OPTION IS CALLED, INITIAL AND INCREMENTAL VALUES EX- PLICITLY SPECIFIED WITHOUT AUTOMATIC SCALING REQUEST (E.G., X0, DX, Y0, DY) ARE UNDERSTOOD TO BE ACTUAL LOG QUANTITIES.	PLOT	61
C	8) IF NP DATA POINTS SAMPLED WITH A REPETITION MULTIPLE MULT EX- CEEDS MM, MULT IS REDUCED TO ITS MAXIMUM ALLOWED VALUE.	PLOT	62
C	INPUT PARAMETERS --	PLOT	63
C	X0 = INITIAL VALUE FOR INDEPENDENT VARIABLE X.	PLOT	64
C	DX = INCREMENTAL VALUE FOR THE INDEPENDENT VARIABLE X, CORRESPONDING TO A RESOLUTION OF THE X-AXIS INTO 50 INTERVALS, SPANNED BY A MAXIMUM OF 51 POINTS. TICK MARK INTERVAL = 5*MULT*DX	PLOT	65
C	Y0(I) = INITIAL VALUE FOR LEFT (I=1) AND RIGHT (I=2) VERTI- CAL SCALES.	PLOT	66
C	DY(I) = INCREMENTAL TICK MARK INTERVAL FOR THE LEFT (I=1) AND RIGHT HAND (I=2) VERTICAL SCALES.	PLOT	67
C	X(I) = INDEPENDENT VARIABLE PROVIDED BY THE CALLING PRO- GRAM, USED ONLY WHEN DX = 0 AND AUTOMATIC X-AXIS SCALING IS REQUIRED. THE VECTOR X IS DESTROYED IN SOME SITUATIONS.	PLOT	68
C	MM = DIMENSION DECLARATOR FOR THE ARRAY Y IN THE CALL- ING PROGRAM: Y(MM,...).	PLOT	69
C	NP = NUMBER OF DATA POINTS TO BE SAMPLED FOR PLOT GEN- ERATION.	PLOT	70
C	MULT = REPETITION FACTOR.	PLOT	71
C	Y(I,J) = ARRAY DIMENSIONED Y(MM,...) IN THE CALLING PRO- GRAM. PLOTS OF THE VECTORS J = 1,2,...,NP ARE GENERATED FROM DATA POINTS Y(1,J), Y(1+MULT,J), Y(1+2*MULT,J), ... AND EACH POINT Y(I,J) IS AS- SUMED TO CORRESPOND TO THE DEPENDENT VARIABLE X(I) = X0 + (I-1)*MULT*DX. IF DX = 0 AND AUTOMA- TIC X-AXIS SCALING IS REQUIRED. THE VECTOR X PRO- VIDED BY INPUT IS USED.	PLOT	72
C	CLEAR = LOGICAL VARIABLE	PLOT	73
C	CLEAR = TRUE : NORMAL PLOT	PLOT	74
C	CLEAR = FALSE: HISTOGRAM	PLOT	75
C	CENTER = LOGICAL VARIABLE TO CENTER PLOT ON PAGE.	PLOT	76
C	SCALEX = LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF	PLOT	77
C		PLOT	78
C		PLOT	79
C		PLOT	80
C		PLOT	81
C		PLOT	82
C		PLOT	83
C		PLOT	84
C		PLOT	85
C		PLOT	86
C		PLOT	87
C		PLOT	88
C		PLOT	89
C		PLOT	90
C		PLOT	91
C		PLOT	92
C		PLOT	93
C		PLOT	94
C		PLOT	95
C		PLOT	96
C		PLOT	97
C		PLOT	98
C		PLOT	99
C		PLOT	100
C		PLOT	101
C		PLOT	102
C		PLOT	103
C		PLOT	104
C		PLOT	105
C		PLOT	106
C		PLOT	107
C		PLOT	108
C		PLOT	109
C		PLOT	110
C		PLOT	111
C		PLOT	112
C		PLOT	113
C		PLOT	114
C		PLOT	115

C	HORIZONTAL AXIS.	PLOT	116
C	SCALEY = LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF VERTICAL AXIS.	PLOT	117
C	SAME = LOGICAL VARIABLE TO SPECIFY SAME SCALE ON THE LEFT AND RIGHT HAND VERTICAL AXES.	PLOT	118
C	NAME(J) = VECTOR OF WORDS (10 BCD CHARACTERS) TO LABEL THE JTH VECTOR PLOTTED.	PLOT	119
C	NP = NUMBER OF VECTORS TO BE PLOTTED (IF NP > 10, DEFAULT TO ONLY 10 PLOTS OCCURS.)	PLOT	120
C	IP = PLOT OPTION, FUNCTIONALLY SIMILAR TO THE USE OF THE FOUR ALTERNATE ENTRY POINTS:	PLOT	121
C	PLOT (IP = 0) -- Y VS X. PYLOGX (IP = -1) -- LOGY VS X. PXLOGY (IP = 2) -- Y VS LOGX. PLOGLOG (IP = 1) -- LOGY VS LOGX.	PLOT	122
C	.....	PLOT	123
C	DIMENSION Y(MULT,MM,1), YP(51,10), K(2,51), TICK(11), FORM(10), 1 DY(1), X1(2), KX(2), DOT(10), XA(10), XB(10), CAPTION(10), ZD(3), 2 DZ(3), X(MULT,1), Y0(1), NAME(1)	PLOT	134
C	LOGICAL T1, T2, TEST, SCALEX, SCALEY, UNSCALE, CENTER, CLEAR, 1 SAME, DIFF, SCALE, XLOG, YLOG	PLOT	135
C	INTEGER DOT DATA DOT / 1H+, 1H+, 1H+, 1H-, 1H+, 1H0, 1MS, 1ME, 1M+, 1H, /	PLOT	136
C	SET UP GENERAL CONTROL CONDITIONS --	PLOT	137
C	ISW = 0 TEST = (IP-1)*(IP+1)*(IP-2).E0.0 IF (TEST) ISW = IP GO TO 40 ENTRY PYLOGX ISW = -1 GO TO 40 ENTRY PXLOGY ISW = 2 GO TO 40 ENTRY PLOGLOG ISW = 1	PLOT	138
40	NVEC = 10 DXMULT = DX*MULT XLOG = IABS(ISW).E0.1 YLOG = ISW.GE.1 NPLOT = NP IF (NPLOT.EQ.0) GO TO 99 IF (NP.GT.NVEC) NPLOT = NVEC IF (.NOT.CLEAR) NPLOT = 1 DIFF = .FALSE.	PLOT	139
		PLOT	140
		PLOT	141
		PLOT	142
		PLOT	143
		PLOT	144
		PLOT	145
		PLOT	146
		PLOT	147
		PLOT	148
		PLOT	149
		PLOT	150
		PLOT	151
		PLOT	152
		PLOT	153
		PLOT	154
		PLOT	155
		PLOT	156
		PLOT	157
		PLOT	158
		PLOT	159
		PLOT	160
		PLOT	161
		PLOT	162
		PLOT	163
		PLOT	164
		PLOT	165
		PLOT	166
		PLOT	167
		PLOT	168
		PLOT	169
		PLOT	170
		PLOT	171
		PLOT	172

```

IF (NPLOT.EQ.2) DIFF = .NOT.SAME
MPLOT = MP
SCALE = SCALEX.OR.(MPLOT.GT.51)
IF (SCALE) MPLOT = 51
MA = M1 = 1
MB = M2 = MPLOT
DO 50 I=1, NVEC
50 XA(I) = XB(I) = 0.0
DZ(1) = DY(1)
Z0(1) = Y0(1)
Z0(2) = DZ(2) = 0.0
Z0(3) = X0
DZ(3) = 5.*DXMULT
C
C      SET UP X-AXIS CONTROL --
C
UNSCALE = .NOT.SCALE
XMAX = XMIN = 0.
IF (UNSCALE) GO TO 61
IF (DX.EQ.0.0) GO TO 63
X(1,1) = X0
DO 54 I=2, MP
54 X(1,I) = X(1,I-1) + DXMULT
63 XMIN = X(1,1)
XMAX = X(1,MP)
EPS = 1.E-05*ABS(XMAX)
XMAX = XMAX - EPS
EPS = 1.E-05*ABS(XMIN)
XMIN = XMIN + EPS
IF (.NOT.XLOG) GO TO 61
IF (XMIN.EQ.0.) XMIN = X(1,2)/10.
IF (XMIN.LE.0.) GO TO 98
XMIN = ALOG10(XMIN)
XMAX = ALOG10(XMAX)
61 CALL AXIS (SCALE, XMAX, XMIN, Z0(3), DZ(3), XI)
.C
.C      SET UP PLOTTING ARRAYS YP --
C
XI = Z0(3)
DXI = DZ(3)/5.0
DO 37 M=1, MPLOT
XP = XI
IF (XLOG) XP = 10.0**XI
IF (UNSCALE) GO TO 53
IF (XP.GT.X(1,1)) GO TO 55
M1 = M
MA = M1+1
55 IF (XP.LT.X(1,MP)) MB = M
M2 = MB+1
53 DO 33 I=1, NPLOT
YP(M,I) = Y(1,M,I)
IF (UNSCALE) GO TO 33
CALL INTERP (2, XP, YP(M,I), X, Y(1,1+I), MULT, MP)
33 CONTINUE
37 XI = XI + DXI
IF (M2.GT.MPLOT) M2 = MPLOT
IF (.NOT.YLOG) GO TO 45
PLOT   173
PLOT   174
PLOT   175
PLOT   176
PLOT   177
PLOT   178
PLOT   179
PLOT   180
PLOT   181
PLOT   182
PLOT   183
PLOT   184
PLOT   185
PLOT   186
PLOT   187
PLOT   188
PLOT   189
PLOT   190
PLOT   191
PLOT   192
PLOT   193
PLOT   194
PLOT   195
PLOT   196
PLOT   197
PLOT   198
PLOT   199
PLOT   200
PLOT   201
PLOT   202
PLOT   203
PLOT   204
PLOT   205
PLOT   206
PLOT   207
PLOT   208
PLOT   209
PLOT   210
PLOT   211
PLOT   212
PLOT   213
PLOT   214
PLOT   215
PLOT   216
PLOT   217
PLOT   218
PLOT   219
PLOT   220
PLOT   221
PLOT   222
PLOT   223
PLOT   224
PLOT   225
PLOT   226
PLOT   227
PLOT   228
PLOT   229

```

```

C
      M1 = MA          PLOT    230
      M2 = MB          PLOT    231
      DO 34 I=1, NPLLOT PLOT    232
      DO 34 M=M1, M2   PLOT    233
      IF (YP(M,I).LE.0.) GO TO 98 PLOT    234
      34 YP(M,I) = ALOG10(YP(M,I)) PLOT    235
C
C   ESTABLISH MAXIMUM AND MINIMUM VALUES FOR EACH VECTOR --
C
      45 DO 1 I=1, NPLLOT PLOT    236
          XA(I) = XB(I) = YP(MA+I) PLOT    237
          DO 2 M = MA,MB PLOT    238
          YM1 = YP(M,I) PLOT    239
          IF(XA(I).GT.YM1) XA(I) = YM1 PLOT    240
          2 IF(XB(I).LT.YM1) XB(I) = YM1 PLOT    241
          IF (XA(I).GE.0.) GO TO 1 PLOT    242
          IF (XB(I).LE.0.) GO TO 1 PLOT    243
          A = ABS(XA(I) + XB(I)) PLOT    244
          B = ABS(XA(I) - XB(I)) PLOT    245
          IF ((B-A).GT.(A+B)/100.) GO TO 1 PLOT    246
          B = XB(I) PLOT    247
          IF (B.GT.A) XA(I) = 0. PLOT    248
          IF (B.LT.A) XB(I) = 0. PLOT    249
          1 CONTINUE PLOT    250
          NSCALE = 1 PLOT    251
C
C   FIND LARGEST XB(I) AND SMALLEST XA(I) --
C
          IF (NPLLOT.EQ.1) GO TO 18 PLOT    252
          NSCALE = 2 PLOT    253
          XMIN = XA(1) PLOT    254
          XMAX = XB(1) PLOT    255
          DO 23 I = 2,NPLLOT PLOT    256
          IF(XA(I).LT.XMIN) XMIN = XA(I) PLOT    257
          23 IF(XB(I).GT.XMAX) XMAX = XB(I) PLOT    258
          EPS = 1.0E-09*ABS(XMAX-XMIN) PLOT    259
          XMIN = XMIN + EPS PLOT    260
          XMAX = XMAX - EPS PLOT    261
C
C   SET UP Y-AXIS CONTROL (FOR SINGLE OR DOUBLE SCALING) --
C
          IF (I.NOT.DIFF) GO TO 10 PLOT    262
          Z0(2) = Y0(2) PLOT    263
          DZ(2) = DY(2) PLOT    264
          10 A = XB(1) - XA(1) PLOT    265
          B = XB(2) - XA(2) PLOT    266
          A = AMAX1(XB(1),A) PLOT    267
          B = AMAX1(XB(2),B) PLOT    268
          DS = AMIN1(A,B) PLOT    269
          XMULT = 10. PLOT    270
          IF (NPLLOT.EQ.2) DIFF = DIFF.OR.((XMAX-XMIN).GT.XMULT*DS) PLOT    271
          IF(DIFF) GO TO 18 PLOT    272
          NSCALE = 1 PLOT    273
          DZ(2) = DZ(1) PLOT    274
          Z0(2) = Z0(1) PLOT    275
          XA(1) = XMIN PLOT    276
          PLOT    277
          PLOT    278
          PLOT    279
          PLOT    280
          PLOT    281
          PLOT    282
          PLOT    283
          PLOT    284
          PLOT    285
          PLOT    286

```

```

XB(1) = XMAX PLOT 287
XA(2) = XB(2) = 0.0 PLOT 288
C PLOT 289
18 DO 15 I = 1,NSCALE PLOT 290
SCALE = SCALEY.OR.(DZ(I),EQ.0.0)
IF (.NOT.CLEAR) XA(I) = 0. PLOT 291
CALL AXIS (SCALE, XB(I), XA(I), Z0(I), DZ(I), XI)
M0 = 0 PLOT 292
IF (.NOT.SCALE) GO TO 29 PLOT 293
XMAX = Z0(I) + 10.*DZ(I)
M0 = 0.5*((XMAX-XB(I))-(XA(I)-Z0(I)))/DZ(I)
IF ((XI.EQ.0.),AND.(Z0(I).EQ.0.)) M0 = 0 PLOT 294
29 DS = M0*DZ(I) PLOT 295
IF (XI.EQ.0.) GO TO 31 PLOT 296
XI = XI - DS PLOT 297
Z0(I) = Z0(I) + DS PLOT 298
31 Z0(I) = Z0(I) - DS PLOT 299
15 XA(I) = XI PLOT 300
IF (.NOT.DIFF) XA(2) = XA(1) PLOT 301
C XA(1), XA(2) ARE DC BASELINE VALUES AT THIS POINT. PLOT 302
C PLOT 303
DO 38 I=1, NPLLOT PLOT 304
XI = XA(1) PLOT 305
IF (I.EQ.2) XI = XA(2) PLOT 306
DO 38 M=M1,M2 PLOT 307
38 YP(M,I) = YP(M,I) - XI PLOT 308
C PLOT 309
DO 11 I=1, NSCALE PLOT 310
11 XI(I) = Z0(I) + 11.0*DZ(I) PLOT 311
KX(1) = KX(2) = -1 PLOT 312
EPS = 0.02 PLOT 313
ETA = 0.5 * EPS PLOT 314
C DETERMINE LOCATION OF HORIZONTAL AXES --
C PLOT 315
DO 9 I = 1,NSCALE PLOT 316
IF(DZ(I).EQ.0.0) GO TO 9 PLOT 317
KX(I) = -5.0*Z0(I)/DZ(I)*1.00001 PLOT 318
9 CONTINUE PLOT 319
C CONVERT PLOTTING ARRAY YP TO NORMALIZE VALUES BETWEEN 0 AND 50 PLOT 320
C (CORRESPONDING TO LINES 1+2+3+...+51 FOR EACH PLOTTED PAGE) --
C PLOT 321
DO 4 I = 1,NPLLOT PLOT 322
DXI = DZ(I) PLOT 323
X0I = Z0(I) PLOT 324
IF (I.NE.2) GO TO 88 PLOT 325
IF (.NOT.DIFF) GO TO 88 PLOT 326
X0I = Z0(2) PLOT 327
DXI = DZ(2) PLOT 328
4 CONTINUE PLOT 329
DO 4 M = M1,M2 PLOT 330
4 YP(M,I) = ((YP(M,I)-X0I)/DXI)*5.0 PLOT 331
C DETERMINE NUMBER OF HORIZONTAL ELEMENTS PER LINE OF PLOT --
C PLOT 332
C IMAX = (M2 + 3)/5 PLOT 333
PLOT 334
PLOT 335
PLOT 336
PLOT 337
PLOT 338
PLOT 339
PLOT 340
PLOT 341
PLOT 342
PLOT 343

```

```

IMAX1 = IMAX+1 PLOT 344
MAX = 5*IMAX + 1 PLOT 345
MAX2 = 2*MAX PLOT 346
N = 68 - MAX PLOT 347
IF(.NOT.DIFF) N = N+5 PLOT 348
IF(.NOT.CENTER) N = 9 PLOT 349
N2 = N-2 PLOT 350
ENCODE(70,100,FORM) N, N2, MAX2, N PLOT 351
100 FORMAT (1H(I2,7HX102A1),7H(1H+1PGI2,3H.4+I3,10HX,1PG14.4)5X,4H(1H+
1 I2, 20HX,7X,A1,SH --- ,A10)) PLOT 352
    ENCODE(90,101,CAPTION) N, N, N PLOT 353
101 FORMAT(4H(1H+,I2,?4HX4X17H * --- LHS AXIS),5H(1H+,,I2, 23HX,22X,
* * *,E12.4,*)), 4H(1H+,I2,24HX4X17H * --- RHS AXIS)) PLOT 354
    PLOT 355
    PLOT 356
    PLOT 357
    PLOT 358
    PLOT 359
C C PLOT S1 LINES -- PLOT 360
C C
LINE1 = 5 PLOT 361
DO 6 L=1, 51 PLOT 362
LL = L-1 PLOT 363
XLINE = LINE = S1-L PLOT 364
T1 = (LINE.EQ.KX(1)).OR.(LINE.EQ.KX(2)) PLOT 365
T2 = (LL/50)*50.EQ.LL PLOT 366
TEST = T1.OR.T2 PLOT 367
DO 5 M=1, MAX PLOT 368
K(2,M) = 1H PLOT 369
DO 42 I=1, NPLOT PLOT 370
IF(K(1,M).EQ.DOT(I)) GO TO 44 PLOT 371
42 CONTINUE PLOT 372
K(1,M) = 1H PLOT 373
44 IF(CLEAR) K(1,M) = 1H PLOT 374
5 IF(TEST) K(1,M) = K(2,M) = 1H- PLOT 375
IF(.NOT.TEST) GO TO 14 PLOT 376
DO 8 I=1, 11 PLOT 377
I1 = 1 + 5*(I-1) PLOT 378
8 K(1,I1) = 1HI PLOT 379
14 K(1,1) = K(1,MAX) = 1HI PLOT 380
IF(L.EQ.S1) GO TO 22 PLOT 381
T2 = 5*(LL/51).EQ.LL PLOT 382
IF(.NOT.(T1.OR.T2)) GO TO 24 PLOT 383
K(2,1) = K(2,MAX-1) = 1H- PLOT 384
DO 62 I=1, NPLOT PLOT 385
IF(K(1,M).EQ.DOT(I)) GO TO 64 PLOT 386
62 CONTINUE PLOT 387
K(1,2) = 1H- PLOT 388
64 DO 66 I=1, NPLOT PLOT 389
IF(K(1,MAX-1).EQ.DOT(I)) GO TO 24 PLOT 390
66 CONTINUE PLOT 391
K(1,MAX-1) = 1H- PLOT 392
24 DO 3 I=1, NPLOT PLOT 393
DO 3 M=M1, M2 PLOT 394
IF(ABS(XLINE-YP(M+I)).LT.ETA) K(1,M) = DOT(I) PLOT 395
IF((M.EQ.M2).OR.(.NOT.CLEAR)) GO TO 3 PLOT 396
XAVG = (YP(M,I)+YP(M+1,I))/2.0 PLOT 397
IF(ABS(XLINE-XAVG).LT.ETA) K(2,M) = DOT(I) PLOT 398
X2AVG = 0.5*(XAVG + YP(M+1,I)) PLOT 399
IF(ABS(XLINE-X2AVG).LT.ETA) K(2,M) = DOT(I) PLOT 400

```

```

IF(ABS(XLINE-X2AVG).LT.ETA) K(2,M) = DOT(I) PLOT 401
XMAX = AMAX1(X1AVG,X2AVG) PLOT 402
XMIN = AMIN1(X1AVG,X2AVG) PLOT 403
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(2,M) = DOT(I) PLOT 404
XMAX = AMAX1(X1AVG,YP(M,I)) PLOT 405
XMIN = AMIN1(X1AVG,YP(M,I)) PLOT 406
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M) = DOT(I) PLOT 407
XMAX = AMAX1(X2AVG,YP(M+1,I)) PLOT 408
XMIN = AMIN1(X2AVG,YP(M+1,I)) PLOT 409
IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M+1) = DOT(I) PLOT 410
3 CONTINUE PLOT 411
22 K(2,MAX) = 1H PLOT 412
M0 = 12 PLOT 413
TEST = (L.LT.49).AND.(L.GT.44) PLOT 414
IF(.NOT.TEST) GO TO 20 PLOT 415
IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 20 PLOT 416
IF((XA(1).NE.0.0).OR.(XA(2).NE.0.0)) M0 = 21 PLOT 417
GO TO 21 PLOT 418
20 IF(NPLOT.EQ.1) GO TO 13 PLOT 419
IF(LINE1.NE.L) GO TO 13 PLOT 420
L1 = L-4 PLOT 421
IF(L1.GT.NPLOT) GO TO 13 PLOT 422
LINE1 = LINE1 + 1 PLOT 423
21 DO 19 M=3, M0 PLOT 424
19 K(1,M) = K(2,M) = 1H PLOT 425
13 WRITE(6,FORM(1)) (K(1,M), K(2,M), M=1, MAX) PLOT 426
IF(L.LE.4) GO TO 16 PLOT 427
IF(LINE1.EQ.(L+1)) WRITE(6,FORM(5)) DOT(L1), NAME(L1) PLOT 428
16 IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 25 PLOT 429
IF(L.NE.46) GO TO 28 PLOT 430
WRITE(6,CAPTION(1)) PLOT 431
IF(XA(1).NE.0.0) WRITE(6,CAPTION(4)) XA(1) PLOT 432
28 IF(L.NE.47) GO TO 25 PLOT 433
IF(.NOT.DIFF) GO TO 25 PLOT 434
WRITE(6,CAPTION(7)) PLOT 435
IF(XA(2).NE.0.0) WRITE(6,CAPTION(4)) XA(2) PLOT 436
25 IF((LL/5)*5.NE.LL) GO TO 6 PLOT 437
DO 17 I = 1,NSCALE PLOT 438
X1(I) = X1(I) - DZ(I) PLOT 439
IF(ABS(X1(I)).LT.0.001*DZ(I)) X1(I) = 0.0 PLOT 440
17 XB(I) = X1(I) PLOT 441
WRITE(6,FORM(2)) (XB(I), I=1, NSCALE) PLOT 442
6 CONTINUE PLOT 443
N8 = N-8 PLOT 444
IF(UNSCALE.AND.(DX.EQ.0.0)) GO TO 30 PLOT 445
AXMAX = ABS(Z0(3) + IMAX*DZ(3)) PLOT 446
IF(AXMAX.EQ.0.0) AXMAX = 10.0*ABS(DZ(3)) PLOT 447
MX = A = ALOG10(AXMAX) PLOT 448
MX = MX+1 PLOT 449
IF(A.LT.0.) MX = MX-1 PLOT 450
SX = 10.0**MX PLOT 451
IF((MX.LT.4).AND.(MX.GE.0)) SX = 1.0 PLOT 452
DO 7 I=1, IMAX1 PLOT 453
TICK(I) = Z0(3) + (I-1)*DZ(3) PLOT 454
7 TICK(I) = TICK(I)/SX PLOT 455
SA = TICK(2) PLOT 456
IF(SX.EQ.0.) SX = TICK(3) PLOT 457

```

```

SX = ABS(SX)
M0 = ALOG10(SX)
IF(M0.GE.0) N8 = N8 + 1
ENCODE(30,102,FORM) N8
102 FORMAT(2H(/,I2,2Hx,.8H11F10.2))
WRITE(6,FORM) (TICK(I), I=1,IMAX)
IF((MX.GE.4).OR.(MX.LT.0)) WRITE(6,103) MX
103 FORMAT (/68x,7H(X 10**,[3,*1])
GO TO 99
30 ENCODE(20,300,FORM) N
300 FORMAT(2H(/,I2,11Hx.*0*10I10))
WRITE(6,FORM) (M, M=5, MAX, 5)
GO TO 99
98 WRITE(6,104)
104 FORMAT(10(/).40X,*AN ATTEMPT WAS MADE TO PLOT VALUES < 0 WITH LOG
1 OPTION*10(/))
99 RETURN
END

```

```

SUBROUTINE AXIS (SCALE, XMAX, XMIN, X0, DX, XDC)          AXIS 2
LOGICAL SCALE                                              AXIS 3
C .....                                                       AXIS 4
C .....                                                       AXIS 5
C .....                                                       AXIS 6
C XMAX AND XMIN ARE THE LARGEST AND SMALLEST VALUES OF THE VECTOR TO   AXIS 7
C BE SCALED WITH CONVENIENT ORIGIN X0 AND TICK MARK SPACING DX. XI   AXIS 8
C IS SET EQUAL TO ZERO UNLESS THE AC RANGE 10*DX < X0/1000. IN THAT   AXIS 9
C CASE XDC IS SET EQUAL TO X0. X0 IS SET EQUAL TO ZERO, AND XMAX AND   AXIS 10
C XMIN ARE REDUCED BY AN AMOUNT XDC. THUS, XDC CORRESPONDS TO A DC   AXIS 11
C BASELINE THAT IS RETURNED NONZERO ONLY IF THE RANGE OF THE PLOT IS   AXIS 12
C VERY SMALL RELATIVE TO THE ABSOLUTE MAGNITUDE OF PLOTTED VALUES.   AXIS 13
C .....                                                       AXIS 14
C .....                                                       AXIS 15
C .....                                                       AXIS 16
C IF (.NOT.SCALE) GO TO 3                                  AXIS 17
C X0 IS CONVENIENT ORIGIN, AND RANGE 10*DX IS 1, 2, OR 5 TIMES SOME   AXIS 18
C POWER OF 10 --                                         AXIS 19
C AXIS 20
C IF (ABS(XMAX-XMIN).GT.1.0E-10*(XMAX+XMIN)) GO TO 1      AXIS 21
XMIN = 0.                                                 AXIS 22
IF (XMAX.EQ.0.) XMAX = 1.0                               AXIS 23
1 XA = XMIN                                              AXIS 24
4 B = XMAX - XA                                         AXIS 25
M = A = ALOG10(B)                                         AXIS 26
IF (A.LT.0.) M = M-1                                     AXIS 27
DS = 10.**M                                              AXIS 28
M = B/DS + 1.                                            AXIS 29
B = 2.                                                   AXIS 30
IF (M.GT.2) B = 5.                                         AXIS 31
IF (M.GT.5) B = 10.                                         AXIS 32
B = B**DS                                              AXIS 33
DS = B/10.                                               AXIS 34
M = A = XA/DS                                           AXIS 35
IF (A.LT.0.) M = M-1                                     AXIS 36
XA = M*DS                                              AXIS 37
IF ((XA+B).LT.XMAX) GO TO 4                            AXIS 38
DX = DS                                                 AXIS 39
X0 = XA                                                 AXIS 40
C CALCULATE DC BASELINE VALUE XDC --
3 XDC = 0.                                                 AXIS 41
IF (ABS(X0).LT.1.E 04*DX) GO TO 2                      AXIS 42
XMIN = XMIN - X0                                         AXIS 43
XMAX = XMAX - X0                                         AXIS 44
XDC = X0                                                 AXIS 45
X0 = 0.                                                   AXIS 46
2 IF (.NOT.SCALE) GO TO 99                             AXIS 47
C ALLOW DATA TO BE PLOTTED WITH ZERO ORIGIN (AT TOP OR BOTTOM) IF   AXIS 48
C POSSIBLE --
X1 = X0 + 10.*DX                                         AXIS 49
IF ((XMAX.LT.10.*DX).AND.(X0.GT.0.)) X0 = 0.           AXIS 50
IF ((XMIN.GT.-10.*DX).AND.(X1.LT.0.)) X0 = -10.*DX     AXIS 51
99 RETURN
END

```

```

SUBROUTINE INTERP (IDEG, XP, YP, X, Y, MULT, N)           INTERP    2
C .....                                                 INTERP    3
C THIS SUBROUTINE WILL INTERPOLATE A VECTOR Y, DEFINED AS A FUNCTION   INTERP    4
C OF THE VECTOR X, TO PRODUCE THE VALUE YP THAT CORRESPONDS TO XP.   INTERP    5
C IDEG = 1, 2, ... SPECIFIES LINEAR, QUADRATIC, ... ETC. INTERPOLA-   INTERP    6
C TION. THE VECTORS X AND Y PROVIDED BY THE CALLING PROGRAM ARE   INTERP    7
C SAMPLED WITH A REPETITION INDEX = MULT, AND N POINTS DEFINED BY   INTERP    8
C I = 1, (1+MULT), ..., (1 + (N-1)*MULT) WILL BE UTILIZED. THUS,   INTERP    9
C THE VECTORS X, Y MUST BE DIMENSIONED AT LEAST (1+(N-1)*MULT)   INTERP   10
C IN THE CALLING PROGRAM. VALUES FOR THE INDEPENDENT VECTOR X MUST   INTERP   11
C BE IN ASCENDING ORDER. XP NEED NOT LIE WITHIN THE RANGE X(1),...X(1+(N-1)*MULT), BUT IF IT FALLS OUTSIDE, LINEAR INTERPOLATION IS   INTERP   12
C GIVEN. (THE DEFAULT TO LINEAR INTERPOLATION CAN BE REMOVED BY   INTERP   13
C DELETION OF ONE CARD BELOW.) NORMAL USAGE IS MULT = 1.   INTERP   14
C .....                                                 INTERP   15
C .....                                                 INTERP   16
C .....                                                 INTERP   17
C .....                                                 INTERP   18
C .....                                                 INTERP   19
C .....                                                 INTERP   20
C .....                                                 INTERP   21
C .....                                                 INTERP   22
C .....                                                 INTERP   23
C .....                                                 INTERP   24
C .....                                                 INTERP   25
C .....                                                 INTERP   26
C .....                                                 INTERP   27
C .....                                                 INTERP   28
C .....                                                 INTERP   29
C .....                                                 INTERP   30
C .....                                                 INTERP   31
C .....                                                 INTERP   32
C .....                                                 INTERP   33
C .....                                                 INTERP   34
C .....                                                 INTERP   35
C .....                                                 INTERP   36
C .....                                                 INTERP   37
C .....                                                 INTERP   38
C .....                                                 INTERP   39
C .....                                                 INTERP   40
C .....                                                 INTERP   41
C .....                                                 INTERP   42
C .....                                                 INTERP   43
C .....                                                 INTERP   44
C .....                                                 INTERP   45
C .....                                                 INTERP   46
C .....                                                 INTERP   47
C .....                                                 INTERP   48
C .....                                                 INTERP   49
C .....                                                 INTERP   50
C .....                                                 INTERP   51

DIMENSION T(20), X(MULT,1), Y(MULT,1)
INT = IDEG
IF (INT.LE.0) INT = 1
IF (INT.GE.N) INT = N-1
IF (N.LT.2) GO TO 10
DO 4 I = 1:N
J = I
IF (XP.LE.X(1,I)) GO TO 1
4 CONTINUE
1 CONTINUE
C DEFAULT TO LINEAR INTERPOLATION IF XP LIES OUTSIDE RANGE OF X --
IF ((J.EQ.1).OR.(J.EQ.N)) INT = 1
K = INT
INT = INT + 1
J = J - INT/2
J = MAX0(J,1)
J = MIN0(J,N-K)
JK = J*K
DO 2 I = J,JK
KK = I-J+1
T(KK) = Y(I,I)
2 T(KK+INT) = X(I+I) - XP
DO 3 I = 1*K
IP1 = I + 1
DO 3 JJ = IP1,INT
3 T(JJ) = (T(I)*T(JJ+INT)-T(JJ)*T(I+INT))/(X(I,JJ+J-1)-X(I+I+J-1))
YP = T(INT)
RETURN
10 YP = Y(1,1)
RETURN
END

SUBROUTINE SIMPSON (F, M, H, ANS)                         SIMPSON   2
C .....                                                 SIMPSON   3
C THIS SUBROUTINE INTEGRATES A FUNCTION F DEFINED AS A VECTOR OVER N =   SIMPSON   4
C 2M INTERVALS: F(1), F(2), F(3), ..., F(2M+1). THE WIDTH OF THE SUB-   SIMPSON   5
C INTERVALS IS H, AND THE ANSWER IS RETURNED IN ANS.   SIMPSON   6
C .....                                                 SIMPSON   7
C .....                                                 SIMPSON   8
C .....                                                 SIMPSON   9
C .....                                                 SIMPSON  10
C .....                                                 SIMPSON  11
C .....                                                 SIMPSON  12
C .....                                                 SIMPSON  13
C .....                                                 SIMPSON  14
C .....                                                 SIMPSON  15
C .....                                                 SIMPSON  16
C .....                                                 SIMPSON  17
C .....                                                 SIMPSON  18
C .....                                                 SIMPSON  19
C .....                                                 SIMPSON  20
C .....                                                 SIMPSON  21

DIMENSION F(1)
N = M * M
SUM = F(N+1) - F(1)
DO 1 I = 1:N+2
II = I+1
Y = F(I) + F(II) + F(III)
1 SUM = SUM + Y + Y
ANS = H*SUM/3.
RETURN
END

```

```

SUBROUTINE EDITOR (INPUT, LIST)           EDITOR    2
C .....                                     EDITOR    3
C .....                                     EDITOR    4
C THIS SUBROUTINE READS AN INPUT CARD FILE TO THE EOF, AND WRITES IT   EDITOR    5
C ONTO TAPE 5. IF LIST = .TRUE. IS SPECIFIED, IT ALSO PRODUCES A     EDITOR    6
C WRITTEN OUTPUT LISTING OF THE INPUT CARD IMAGES.                      EDITOR    7
C .....                                     EDITOR    8
C .....                                     EDITOR    9
C .....                                     EDITOR   10
C .....                                     EDITOR   11
C DIMENSION IMAGE(8)                  EDITOR   12
C LOGICAL LIST                      EDITOR   13
C DATA SKIP, K / 5H(IH), 1234567890 /  EDITOR   14
C REWIND 5                           EDITOR   15
C KARD = 0                           EDITOR   16
C CALL DATE (MONTH)                 EDITOR   17
1 READ (INPUT,150) IMAGE             EDITOR   18
  IF (EOF(INPUT)) 5,3               EDITOR   19
3 IF (MOD(KARD+40).NE.0) GO TO 2   EDITOR   20
  IF (.NOT.LIST) GO TO 2          EDITOR   21
  IF (KARD.EQ.0) GO TO 4          EDITOR   22
  WRITE (6,110) (K, I = 1,8), (I, I = 1,8)
  WRITE (6,120)
120 FORMAT (//21X,*CONTINUED*)
  4 WRITE (6,SKIP)
  WRITE (6,300) MONTH
300 FORMAT (//47X,*SUMMARY OF CARD IMAGES FOR INPUT DATA DECK*/60X,*ID
  DATE:*,A9,*)//)
  WRITE (6,100) (I, I = 1,8), (K, I = 1,8)
100 FORMAT (22X,*CARD*,13X,8(1I,9X)/23X,*NO.*,4X,8I10//21X,92(IH-1))
  2 KARD = KARD + 1
  WRITE (5,150) IMAGE
150 FORMAT (8A10)
  IF (LIST) WRITE (6,200) KARD, IMAGE
200 FORMAT (22X,I3,* ...,*,8A10)
  GO TO 1
C
5 ENDFILE 5                         EDITOR   39
  REWIND 5                           EDITOR   40
  IF (KARD.EQ.0) GO TO 99          EDITOR   41
  IF (.NOT.LIST) GO TO 99          EDITOR   42
  WRITE (6,110) (K, I = 1,8), (I, I = 1,8)
110 FORMAT (/21X,92(IH-1)//30X,8I10/39X,8(1I,9X))
99 RETURN                            EDITOR   45
END                                  EDITOR   46

```

```

SUBROUTINE COVER (TITLE, NPAGE)                                COVER   2
DIMENSION TITLE(3), MESSAGE(10,3)                            COVER   3
INTEGER BLANK, TITLE                                         COVER   4
C .....                                                       COVER   5
C THIS SUBROUTINE WILL GENERATE NPAGE SEQUENTIAL TITLE PAGES FOR THE COVER   6
C OUTPUT PRINTOUT.                                            COVER   7
C .....                                                       COVER   8
C .....                                                       COVER   9
C .....                                                       COVER  10
C .....                                                       COVER  11
C .....                                                       COVER  12
C .....                                                       COVER  13
C BLANK = 1H                                                 COVER  14
C N = 0                                                       COVER  15
C DO 2 I = 1,3                                              COVER  16
C IF (TITLE(I).EQ.BLANK) GO TO 2                            COVER  17
C N = N+1                                                    COVER  18
C TITLE(N) = TITLE(I)                                       COVER  19
C DECODE (10,100,TITLE(N)) (MESSAGE(J,N), J = 1,10)        COVER  20
100 FORMAT (10A1)                                           COVER  21
2 CONTINUE                                                 COVER  22
C NSKIP = 3 + 6*(3-N)                                      COVER  23
ENCODE (10,150,KONTROL) NSKIP                            COVER  24
150 FORMAT (1H1,I2)                                         COVER  25
NSKIP = KONTROL                                           COVER  26
C .....                                                       COVER  27
C DO 1 K = 1,NPAGE                                         COVER  28
C J = 0                                                       COVER  29
C KONTROL = NSKIP                                           COVER  30
3 IF (J.EQ.N) GO TO 1                                     COVER  31
C J = J+1                                                    COVER  32
CALL HEADINX (IHX, KONTROL, MESSAGE(1,J))                 COVER  33
C KONTROL = BLANK                                           COVER  34
GO TO 3                                                    COVER  35
1 WRITE (6,200)                                           COVER  36
C .....                                                       COVER  37
C 200 FORMAT (////)
1/39X,52H      ***** THIS CODE WAS DEVELOPED BY           COVER  38
2/39X,52H      *                                              COVER  39
3/39X,52H      *                                              COVER  40
4/39X,52H      *          DR. WILLIAM B. LACINA           COVER  41
5/39X,52H      *          NORTHROP RESEARCH AND TECHNOLOGY    COVER  42
6/39X,52H      *          ONE RESEARCH PARK              COVER  43
7/39X,52H      *          PALOS VERDES PENINSULA, CA 90274    COVER  44
8/39X,52H      *          TEL: (213) 377-4811, EXT. 322     COVER  45
9/39X,52H      *                                              COVER  46
S/39X,52H      *                                              COVER  47
1/39X,52H      *                                              COVER  48
2/39X,52H      *                                              COVER  49
3/39X,52H      *****                                         COVER  50
C .....                                                       COVER  51
C RETURN                                                 COVER  52
END                                                       COVER  53
C .....                                                       COVER  54

```

```

SUBROUTINE HEADINX (JSYMB, JPAGE, MESSAGE)          HEADINX  2
C ..... HEADINX  3
C ..... HEADINX  4
C ..... HEADINX  5
C ..... HEADINX  6
C THE CALL SEQUENCE FOR THIS SUBROUTINE IS EQUIVALENT TO THAT SUPPLIED HEADINX  7
C BY THE CDC CYBERNET SYSTEM (WITH THE SAME NAME) FOR THE GENERATION OF HEADINX  8
C BLOCK HEADINGS. HOWEVER, THE PRESENT VERSION HAS TWO ADDITIONAL AD- HEADINX  9
C VANTAGES:                                         HEADINX 10
C ..... HEADINX 11
C 1) THE 10-BCD CHARACTER WORD STORED IN THE VECTOR MESSAGE(I) HEADINX 12
C (I = 1,2,...,10) IS AUTOMATICALLY CENTERED ON THE PRINTED HEADINX 13
C LINE, AND                                         HEADINX 14
C ..... HEADINX 15
C 2) THE CARRIAGE CONTROL SYMBOL *JPAGE* CAN ACCEPT MORE GENERAL HEADINX 16
C SPECIFICATIONS TO CONTROL THE SPACING OF THE PRINTED LINE. HEADINX 17
C JPAGE CAN HAVE THE USUAL FORMAT IHS (WHERE S = 0, 1, 2, .., HEADINX 18
C ETC). OR IT CAN HAVE A MORE GENERAL FORMAT 3HSNN, WHERE NN HEADINX 19
C IS A TWO-DIGIT NUMBER. AFTER THE PAGE CONTROL S IS EXECUTED, HEADINX 20
C TWO LINES ARE AUTOMATICALLY SPACED, FOLLOWED BY AN ADDITIONAL HEADINX 21
C NN LINES. THE REMAINING SEVEN BITS OF *JPAGE* ARE IGNORED. HEADINX 22
C ..... HEADINX 23
C (UNLIKE THE CDC SUBROUTINE, THE PLOTTING SYMBOL *JSYMB* IS IGNORED, HEADINX 24
C AND THE CHARACTER X IS ALWAYS USED.               HEADINX 25
C ..... HEADINX 26
C ..... HEADINX 27
C DIMENSION KAR(10,50), FORM(3), MESSAGE(10), LETTER(50), NUM(10), HEADINX 28
C 1 IMAGE(10), KK(500)                           HEADINX 29
C ..... HEADINX 30
C ..... HEADINX 31
C ..... HEADINX 32
C ..... HEADINX 33
C ..... HEADINX 34
C ..... HEADINX 35
C ..... HEADINX 36
C ..... HEADINX 37
C ..... HEADINX 38
C ..... HEADINX 39
C ..... HEADINX 40
C ..... HEADINX 41
C ..... HEADINX 42
C ..... HEADINX 43
C ..... HEADINX 44
C ..... HEADINX 45
C ..... HEADINX 46
C ..... HEADINX 47
C ..... HEADINX 48
C ..... HEADINX 49
C ..... HEADINX 50
C ..... HEADINX 51
C ..... HEADINX 52
C ..... HEADINX 53
C ..... HEADINX 54
C ..... HEADINX 55
C ..... HEADINX 56
C ..... HEADINX 57
C ..... HEADINX 58
C ..... HEADINX 59
C ..... HEADINX 60
C ..... HEADINX 61
C ..... HEADINX 62
C ..... HEADINX 63
C ..... HEADINX 64
C ..... HEADINX 65
C ..... HEADINX 66
C ..... HEADINX 67
C ..... HEADINX 68
C ..... HEADINX 69
C ..... HEADINX 70
C ..... HEADINX 71
C ..... HEADINX 72
C ..... HEADINX 73
C ..... HEADINX 74
C ..... HEADINX 75
C ..... HEADINX 76
C ..... HEADINX 77
C ..... HEADINX 78
C ..... HEADINX 79
C ..... HEADINX 80
C ..... HEADINX 81
C ..... HEADINX 82
C ..... HEADINX 83
C ..... HEADINX 84
C ..... HEADINX 85
C ..... HEADINX 86
C ..... HEADINX 87
C ..... HEADINX 88
C ..... HEADINX 89
C ..... HEADINX 90
C ..... HEADINX 91
C ..... HEADINX 92
C ..... HEADINX 93
C ..... HEADINX 94
C ..... HEADINX 95
C ..... HEADINX 96
C ..... HEADINX 97
C ..... HEADINX 98
C ..... HEADINX 99
C ..... HEADINX 100
C ..... HEADINX 101
C ..... HEADINX 102
C ..... HEADINX 103
C ..... HEADINX 104
C ..... HEADINX 105
C ..... HEADINX 106
C ..... HEADINX 107
C ..... HEADINX 108
C ..... HEADINX 109
C ..... HEADINX 110
C ..... HEADINX 111
C ..... HEADINX 112
C ..... HEADINX 113
C ..... HEADINX 114
C ..... HEADINX 115
C ..... HEADINX 116
C ..... HEADINX 117
C ..... HEADINX 118
C ..... HEADINX 119
C ..... HEADINX 120
C ..... HEADINX 121
C ..... HEADINX 122
C ..... HEADINX 123
C ..... HEADINX 124
C ..... HEADINX 125
C ..... HEADINX 126
C ..... HEADINX 127
C ..... HEADINX 128
C ..... HEADINX 129
C ..... HEADINX 130
C ..... HEADINX 131
C ..... HEADINX 132
C ..... HEADINX 133
C ..... HEADINX 134
C ..... HEADINX 135
C ..... HEADINX 136
C ..... HEADINX 137
C ..... HEADINX 138
C ..... HEADINX 139
C ..... HEADINX 140
C ..... HEADINX 141
C ..... HEADINX 142
C ..... HEADINX 143
C ..... HEADINX 144
C ..... HEADINX 145
C ..... HEADINX 146
C ..... HEADINX 147
C ..... HEADINX 148
C ..... HEADINX 149
C ..... HEADINX 150
C ..... HEADINX 151
C ..... HEADINX 152
C ..... HEADINX 153
C ..... HEADINX 154
C ..... HEADINX 155
C ..... HEADINX 156
C ..... HEADINX 157
C ..... HEADINX 158
C ..... HEADINX 159
C ..... HEADINX 160
C ..... HEADINX 161
C ..... HEADINX 162
C ..... HEADINX 163
C ..... HEADINX 164
C ..... HEADINX 165
C ..... HEADINX 166
C ..... HEADINX 167
C ..... HEADINX 168
C ..... HEADINX 169
C ..... HEADINX 170
C ..... HEADINX 171
C ..... HEADINX 172
C ..... HEADINX 173
C ..... HEADINX 174
C ..... HEADINX 175
C ..... HEADINX 176
C ..... HEADINX 177
C ..... HEADINX 178
C ..... HEADINX 179
C ..... HEADINX 180
C ..... HEADINX 181
C ..... HEADINX 182
C ..... HEADINX 183
C ..... HEADINX 184
C ..... HEADINX 185
C ..... HEADINX 186
C ..... HEADINX 187
C ..... HEADINX 188
C ..... HEADINX 189
C ..... HEADINX 190
C ..... HEADINX 191
C ..... HEADINX 192
C ..... HEADINX 193
C ..... HEADINX 194
C ..... HEADINX 195
C ..... HEADINX 196
C ..... HEADINX 197
C ..... HEADINX 198
C ..... HEADINX 199
C ..... HEADINX 200
C ..... HEADINX 201
C ..... HEADINX 202
C ..... HEADINX 203
C ..... HEADINX 204
C ..... HEADINX 205
C ..... HEADINX 206
C ..... HEADINX 207
C ..... HEADINX 208
C ..... HEADINX 209
C ..... HEADINX 210
C ..... HEADINX 211
C ..... HEADINX 212
C ..... HEADINX 213
C ..... HEADINX 214
C ..... HEADINX 215
C ..... HEADINX 216
C ..... HEADINX 217
C ..... HEADINX 218
C ..... HEADINX 219
C ..... HEADINX 220
C ..... HEADINX 221
C ..... HEADINX 222
C ..... HEADINX 223
C ..... HEADINX 224
C ..... HEADINX 225
C ..... HEADINX 226
C ..... HEADINX 227
C ..... HEADINX 228
C ..... HEADINX 229
C ..... HEADINX 230
C ..... HEADINX 231
C ..... HEADINX 232
C ..... HEADINX 233
C ..... HEADINX 234
C ..... HEADINX 235
C ..... HEADINX 236
C ..... HEADINX 237
C ..... HEADINX 238
C ..... HEADINX 239
C ..... HEADINX 240
C ..... HEADINX 241
C ..... HEADINX 242
C ..... HEADINX 243
C ..... HEADINX 244
C ..... HEADINX 245
C ..... HEADINX 246
C ..... HEADINX 247
C ..... HEADINX 248
C ..... HEADINX 249
C ..... HEADINX 250
C ..... HEADINX 251
C ..... HEADINX 252
C ..... HEADINX 253
C ..... HEADINX 254
C ..... HEADINX 255
C ..... HEADINX 256
C ..... HEADINX 257
C ..... HEADINX 258
C ..... HEADINX 259
C ..... HEADINX 260
C ..... HEADINX 261
C ..... HEADINX 262
C ..... HEADINX 263
C ..... HEADINX 264
C ..... HEADINX 265
C ..... HEADINX 266
C ..... HEADINX 267
C ..... HEADINX 268
C ..... HEADINX 269
C ..... HEADINX 270
C ..... HEADINX 271
C ..... HEADINX 272
C ..... HEADINX 273
C ..... HEADINX 274
C ..... HEADINX 275
C ..... HEADINX 276
C ..... HEADINX 277
C ..... HEADINX 278
C ..... HEADINX 279
C ..... HEADINX 280
C ..... HEADINX 281
C ..... HEADINX 282
C ..... HEADINX 283
C ..... HEADINX 284
C ..... HEADINX 285
C ..... HEADINX 286
C ..... HEADINX 287
C ..... HEADINX 288
C ..... HEADINX 289
C ..... HEADINX 290
C ..... HEADINX 291
C ..... HEADINX 292
C ..... HEADINX 293
C ..... HEADINX 294
C ..... HEADINX 295
C ..... HEADINX 296
C ..... HEADINX 297
C ..... HEADINX 298
C ..... HEADINX 299
C ..... HEADINX 300
C ..... HEADINX 301
C ..... HEADINX 302
C ..... HEADINX 303
C ..... HEADINX 304
C ..... HEADINX 305
C ..... HEADINX 306
C ..... HEADINX 307
C ..... HEADINX 308
C ..... HEADINX 309
C ..... HEADINX 310
C ..... HEADINX 311
C ..... HEADINX 312
C ..... HEADINX 313
C ..... HEADINX 314
C ..... HEADINX 315
C ..... HEADINX 316
C ..... HEADINX 317
C ..... HEADINX 318
C ..... HEADINX 319
C ..... HEADINX 320
C ..... HEADINX 321
C ..... HEADINX 322
C ..... HEADINX 323
C ..... HEADINX 324
C ..... HEADINX 325
C ..... HEADINX 326
C ..... HEADINX 327
C ..... HEADINX 328
C ..... HEADINX 329
C ..... HEADINX 330
C ..... HEADINX 331
C ..... HEADINX 332
C ..... HEADINX 333
C ..... HEADINX 334
C ..... HEADINX 335
C ..... HEADINX 336
C ..... HEADINX 337
C ..... HEADINX 338
C ..... HEADINX 339
C ..... HEADINX 340
C ..... HEADINX 341
C ..... HEADINX 342
C ..... HEADINX 343
C ..... HEADINX 344
C ..... HEADINX 345
C ..... HEADINX 346
C ..... HEADINX 347
C ..... HEADINX 348
C ..... HEADINX 349
C ..... HEADINX 350
C ..... HEADINX 351
C ..... HEADINX 352
C ..... HEADINX 353
C ..... HEADINX 354
C ..... HEADINX 355
C ..... HEADINX 356
C ..... HEADINX 357
C ..... HEADINX 358
C ..... HEADINX 359
C ..... HEADINX 360
C ..... HEADINX 361
C ..... HEADINX 362
C ..... HEADINX 363
C ..... HEADINX 364
C ..... HEADINX 365
C ..... HEADINX 366
C ..... HEADINX 367
C ..... HEADINX 368
C ..... HEADINX 369
C ..... HEADINX 370
C ..... HEADINX 371
C ..... HEADINX 372
C ..... HEADINX 373
C ..... HEADINX 374
C ..... HEADINX 375
C ..... HEADINX 376
C ..... HEADINX 377
C ..... HEADINX 378
C ..... HEADINX 379
C ..... HEADINX 380
C ..... HEADINX 381
C ..... HEADINX 382
C ..... HEADINX 383
C ..... HEADINX 384
C ..... HEADINX 385
C ..... HEADINX 386
C ..... HEADINX 387
C ..... HEADINX 388
C ..... HEADINX 389
C ..... HEADINX 390
C ..... HEADINX 391
C ..... HEADINX 392
C ..... HEADINX 393
C ..... HEADINX 394
C ..... HEADINX 395
C ..... HEADINX 396
C ..... HEADINX 397
C ..... HEADINX 398
C ..... HEADINX 399
C ..... HEADINX 400
C ..... HEADINX 401
C ..... HEADINX 402
C ..... HEADINX 403
C ..... HEADINX 404
C ..... HEADINX 405
C ..... HEADINX 406
C ..... HEADINX 407
C ..... HEADINX 408
C ..... HEADINX 409
C ..... HEADINX 410
C ..... HEADINX 411
C ..... HEADINX 412
C ..... HEADINX 413
C ..... HEADINX 414
C ..... HEADINX 415
C ..... HEADINX 416
C ..... HEADINX 417
C ..... HEADINX 418
C ..... HEADINX 419
C ..... HEADINX 420
C ..... HEADINX 421
C ..... HEADINX 422
C ..... HEADINX 423
C ..... HEADINX 424
C ..... HEADINX 425
C ..... HEADINX 426
C ..... HEADINX 427
C ..... HEADINX 428
C ..... HEADINX 429
C ..... HEADINX 430
C ..... HEADINX 431
C ..... HEADINX 432
C ..... HEADINX 433
C ..... HEADINX 434
C ..... HEADINX 435
C ..... HEADINX 436
C ..... HEADINX 437
C ..... HEADINX 438
C ..... HEADINX 439
C ..... HEADINX 440
C ..... HEADINX 441
C ..... HEADINX 442
C ..... HEADINX 443
C ..... HEADINX 444
C ..... HEADINX 445
C ..... HEADINX 446
C ..... HEADINX 447
C ..... HEADINX 448
C ..... HEADINX 449
C ..... HEADINX 450
C ..... HEADINX 451
C ..... HEADINX 452
C ..... HEADINX 453
C ..... HEADINX 454
C ..... HEADINX 455
C ..... HEADINX 456
C ..... HEADINX 457
C ..... HEADINX 458
C ..... HEADINX 459
C ..... HEADINX 460
C ..... HEADINX 461
C ..... HEADINX 462
C ..... HEADINX 463
C ..... HEADINX 464
C ..... HEADINX 465
C ..... HEADINX 466
C ..... HEADINX 467
C ..... HEADINX 468
C ..... HEADINX 469
C ..... HEADINX 470
C ..... HEADINX 471
C ..... HEADINX 472
C ..... HEADINX 473
C ..... HEADINX 474
C ..... HEADINX 475
C ..... HEADINX 476
C ..... HEADINX 477
C ..... HEADINX 478
C ..... HEADINX 479
C ..... HEADINX 480
C ..... HEADINX 481
C ..... HEADINX 482
C ..... HEADINX 483
C ..... HEADINX 484
C ..... HEADINX 485
C ..... HEADINX 486
C ..... HEADINX 487
C ..... HEADINX 488
C ..... HEADINX 489
C ..... HEADINX 490
C ..... HEADINX 491
C ..... HEADINX 492
C ..... HEADINX 493
C ..... HEADINX 494
C ..... HEADINX 495
C ..... HEADINX 496
C ..... HEADINX 497
C ..... HEADINX 498
C ..... HEADINX 499
C ..... HEADINX 500

```

8 10HXXXXXX. 9H XXXXXXXX / HEADINX 59  
 C DATA (KK(I), I = 151,280) HEADINX 60  
 S / 9HXXXXXXXX. 10HXXXXXXXX. 2\*10HXX XX. HEADINX 61  
 1 10HXXXXXXXX. 9HXXXXXXXX. 4\*2HXX. 9H XXXXXXXX. 10HXXXXXXXX. HEADINX 62  
 2 4\*10HXX XX. 10HXX XXX XX. 10HXX XXXXX. 9HXXXXXXXX. HEADINX 63  
 3 10H XXXXXX XX. 9HXXXXXXXX. 10HXXXXXXXX. 2\*10HXX XX. HEADINX 64  
 4 10HXXXXXX. 9HXXXXXXXX. 7HXX XX. 8HXX XX. 9HXX XX. HEADINX 65  
 5 10HXX XX. 9H XXXXXXXX. 10HXXXXXXXX. 10HXX XX. 2HXX. HEADINX 66  
 6 9HXXXXXXXX. 10H XXXXXXXX. 10H XX. 10HXX XX. HEADINX 67  
 7 10HXXXXXX. 9H XXXXXXXX. 2\*10HXXXXXXXX. 8\*6H XX. HEADINX 68  
 8 8\*10HXX XX. 10HXXXXXX. 9H XXXXXXXX. 2\*10HXX XX. HEADINX 69  
 9 4\*10H XX XX. 2\*8H XX XX. 7H XXXX. 6H XX. HEADINX 70  
 S 4\*10HXX XX. 3\*10HXX XX XX. 10HXX XXXX XX. 10H XXX XXX. HEADINX 71  
 1 9H XX XX. 10HXX XX. 9H XX XX. 8H XX XX. 7H XXXX. HEADINX 72  
 2 6H XX. 7H XXXX. 8H XX XX. 9H XX XX. 3\*10HXX XX. HEADINX 73  
 3 9H XX XX. 8H XX XX. 7H XXXX. 6\*6H XX. 10HXXXXXXXX. HEADINX 74  
 4 9HXXXXXXXX. 8H XX. 7H XX. 2\*6H XX. 5H XX. 4H XX. HEADINX 75  
 5 10H XXXXXXXX. 10HXXXXXX. 6H XX. 6H XXXX. 6H XX XX. HEADINX 76  
 6 5\*6H XX. 2\*8H XXXXXX. 9H XXXXXXXX. 10HXXXXXX. HEADINX 77  
 7 10HXX XX. 10H XXXX. 9H XXXX. 7H XX. 5H XX. HEADINX 78  
 8 4H XXX. 2\*10HXXXXXX / HEADINX 79  
 HEADINX 80  
 HEADINX 81  
 C DATA (KK(I), I = 281,410) HEADINX 82  
 S / 10HXXXXXXXX. 9HXXXXXXXX. 8H XX. 7H XXX. HEADINX 83  
 1 7H XXXXX. 9H XXXX. 10H XXX. 10HXX XXX. 9HXXXXXXXX. HEADINX 84  
 2 8H XXXXXX. 8H XXX. 8H XXXX. 8H XX XX. 8H XX XX. HEADINX 85  
 3 8H XX XX. 2\*10HXXXXXXXX. 3\*8H XX. 2\*10HXXXXXXXX. HEADINX 86  
 4 2\*2HXX. 9HXXXXXXXX. 10HXXXXXXXX. 10H XX. 10HXX XX. HEADINX 87  
 5 10HXXXXXXXX. 2\*9H XXXXXXXX. 10HXXXXXXXX. 10HXX XX. 2HXX. HEADINX 88  
 6 9HXXXXXXXX. 10HXXXXXXXX. 2\*10HXX XX. 10HXXXXXXXX. HEADINX 89  
 7 9H XXXXXXXX. 2\*10HXXXXXXXX. 9H XXX. 7H XXX. 6H XXX. HEADINX 90  
 8 5H XXX. 4H XXX. 3\*3HXX. 9H XXXXXXXX. 10HXXXXXXXX. HEADINX 91  
 9 2\*10HXX XX. 2\*9H XXXXXXXX. 2\*10HXX XX. 10HXXXXXXXX. HEADINX 92  
 S 2\*9H XXXXXXXX. 10HXXXXXXXX. 2\*10HXX XX. 10HXXXXXXXX. HEADINX 93  
 1 10H XXXXXXXX. 10H XX. 10HXX XX. 10HXXXXXXXX. HEADINX 94  
 2 9H XXXXXXXX. 7H XXXX. 9H XXXXXXXX. 9H XX XX. 4\*10HXX XX. HEADINX 95  
 3 9H XX XX. 9H XXXXXXXX. 7H XXXX. 1H +3\*6H XX. 2\*9H XXXXXXXX. HEADINX 96  
 4 3\*6H XX. 5\*1H .2\*9H XXXXXXXX. 5\*1H . 9H XX. 8H XX. HEADINX 97  
 5 7H XX. 6H XX. 5H XX. 4H XX. 3H XX. 2HXX. 2\*1H . 6H XX. HEADINX 98  
 6 9H XX XX XX. 8H XXXXXX. 2\*7H XXXX. 8H XXXXXX. 9H XX XX XX. HEADINX 99  
 7 6H XX. 3\*1H . 2\*6H XX. 2\*1H . 2\*6H XX. 2\*1H / HEADINX 100  
 HEADINX 101  
 C DATA (KK(I), I = 411,500) HEADINX 102  
 S / 6H XX. 9H XXXXXXXX. 10HXX XX XX. 6HXX XX. HEADINX 103  
 1 9HXXXXXXXX. 10H XXXXXXXX. 10H XX XX. 10HXX XX XX. HEADINX 104  
 2 9HXXXXXXXX. 6H XX. 2\*1H . 2\*9H XXXXXXXX. 2\*1H . 2\*9H XXXXXXXX. HEADINX 105  
 3 2\*1H . 7H XX. 7H XXX. 5H XX. 4\*4H XX. 5H XX. 6H XXX. HEADINX 106  
 4 7H XX. 5H XX. 6H XXX. 7H XX. 4\*8H XX. 7H XX. HEADINX 107  
 5 6H XXX. 5H XX. 8H XX. 7H XX. 6H XX. 5H XX. HEADINX 108  
 6 2\*4H XX. 5H XX. 6H XX. 7H XX. 8H XX. 4H XX. HEADINX 109  
 7 5H XX. 6H XX. 7H XX. 2\*8H XX. 7H XX. 6H XX. HEADINX 110  
 8 5H XX. 4H XX. 7H XXX. 8\*5H XX. 2\*7H XXXX. 8\*7H XX. HEADINX 111  
 9 7H XXXX. 10\*1H / HEADINX 112  
 HEADINX 113  
 HEADINX 114  
 HEADINX 115

```

L1 = L2 = 0                                HEADINX 116
DO 5 L = 1,10                               HEADINX 117
IF (MESSAGE(L),NE,BLANK) GO TO 6          HEADINX 118
5 L1 = L1+1                                 HEADINX 119
6 DO 7 I = 1,10                               HEADINX 120
L = 11-I                                    HEADINX 121
IF (MESSAGE(L),NE,BLANK) GO TO 8          HEADINX 122
7 L2 = L2+1                                 HEADINX 123
8 LB = L1 + L2                               HEADINX 124
IF (LB,GE,10) L1 = LB = 0                  HEADINX 125
LIP1 = L1 + 1                               HEADINX 126
NK = 10 - LB                                HEADINX 127
LB = (13*LB)/2                             HEADINX 128
NKPL1 = NK + L1                            HEADINX 129
DECODE (10,102,JPAGE) KSKIP               HEADINX 130
102 FORMAT (1X,A2,7X)                      HEADINX 131
IF (KSKIP,EQ,BLANK) GO TO 9                HEADINX 132
ENCODE (10,101,KONTROL) JPAGE              HEADINX 133
101 FORMAT (A3,7H(/)//1X)                   HEADINX 134
GO TO 10                                    HEADINX 135
9 ENCODE (10,103,KONTROL) JPAGE              HEADINX 136
103 FORMAT (A1,9H//1X)                      HEADINX 137
10 DO 1 L = LIP1,NKPL1                     HEADINX 138
DO 2 I = 1,NSYMBOL                         HEADINX 139
IF (MESSAGE(L),EQ,LETTER(I)) GO TO 1       HEADINX 140
2 CONTINUE                                   HEADINX 141
I = NSYMBOL                                  HEADINX 142
1 NUM(L) = I                                HEADINX 143
DO 3 LINE = 1,10                           HEADINX 144
DO 4 L = LIP1, NKPL1                        HEADINX 145
N = NUM(L)                                  HEADINX 146
4 IMAGE(L) = KAR(LINE,N)                   HEADINX 147
ENCODE (30,100,FORM) KONTROL, LB, NK        HEADINX 148
100 FORMAT (3H(1H,A10,I2,1HX,I2,9H(3X,A10))) HEADINX 149
WRITE (6,FORM) (IMAGE(K), K = LIP1, NKPL1)   HEADINX 150
3 KONTROL = 1H                                HEADINX 151
RETURN                                     HEADINX 152
END                                         HEADINX 153

```

```

PROGRAM ELECT (INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT,TAPE7,TAPE8,TAPE9,
1 TAPE10=INPUT)          ELECT    2
C-----ELECT    3
C-----ELECT    4
C-----ELECT    5
C-----ELECT    6
C-----ELECT    7
C-----ELECT    8
C-----ELECT    9
C-----ELECT   10
C-----ELECT   11
C-----ELECT   12
C-----ELECT   13
C-----ELECT   14
C-----ELECT   15
C-----ELECT   16
C-----ELECT   17
C-----ELECT   18
C-----ELECT   19
C-----ELECT   20
C-----ELECT   21
C-----ELECT   22
C-----ELECT   23
C-----ELECT   24
C-----ELECT   25
C-----ELECT   26
C-----ELECT   27
C-----ELECT   28
C-----ELECT   29
C-----ELECT   30
C-----ELECT   31
C-----ELECT   32
C-----ELECT   33
C-----ELECT   34
C-----ELECT   35
C-----ELECT   36
C-----ELECT   37
C-----ELECT   38
C-----ELECT   39
C-----ELECT   40
C-----ELECT   41
C-----ELECT   42
C-----ELECT   43
C-----ELECT   44
C-----ELECT   45
C-----ELECT   46
C-----ELECT   47
C-----ELECT   48
C-----ELECT   49
C-----ELECT   50
C-----ELECT   51
C-----ELECT   52
C-----ELECT   53
C-----ELECT   54
C-----ELECT   55
C-----ELECT   56
C-----ELECT   57
C-----ELECT   58

C THIS CODE PERFORMS A NUMERICAL SOLUTION OF THE BOLTZMANN TRANS-
C PORT EQUATION FOR THE ELECTRON ENERGY DISTRIBUTION IN A WEAKLY
C IONIZED PLASMA IN THE PRESENCE OF AN ELECTRIC FIELD. MOMENTUM
C TRANSFER (WITH RECOIL), INELASTIC BINARY E-NEUTRAL PROCESSES (WITH
C SUPERELASTIC COLLISIONS), AND ELECTRON-ELECTRON SCATTERING ARE ALL
C INCLUDED, WITH CROSS SECTIONS PROVIDED BY AN ARBITRARILY LARGE EX-
C TERNAL FILE OF DATA. INPUT PARAMETERS ARE GAS MIXTURE, TEMPERA-
C TURE, PRESSURE, EXCITED STATE POPULATION DENSITIES, AND A SEQUENCE
C OF F/N VALUES. OUTPUT CONSISTS OF TABLES AND PLOTS OF PLASMA PA-
C RAMETERS, POWER PARTITIONING, <VSIG> EXCITATION AND DE-EXCITATION
C METERS, POWER PARTITIONING, EXCITATION AND DE-EXCITATION <VSIG>
C RATES, ETC.

C *****
C * THIS CODE WAS DEVELOPED BY
C *
C * DR. WILLIAM B. LACINA
C * NORTHRUP RESEARCH AND TECHNOLOGY
C * ONE RESEARCH PARK
C * PALOS VERDES PENINSULA, CA 90274
C * TEL: (213) 377-4811, EXT. 322
C *
C *****
C COMPLETE DOCUMENTATION OF THE PRESENT ANALYSIS (INCLUDING DISCUS-
C SION OF THE MATHEMATICAL FORMULATION, TECHNIQUES OF NUMERICAL SO-
C LUTION, DESCRIPTION OF SUBROUTINES, AND INSTRUCTION FOR USAGE)
C IS AVAILABLE IN PUBLISHED REPORTS BY W. B. LACINA. A COMPREHEN-
C SIVE AND GENERAL LASER KINETICS CODE IS ALSO AVAILABLE, AND MAKES
C USE OF THE SAME SUBROUTINES FOR THE ELECTRON KINETICS ANALYSIS.

C -----
C DIMENSION Q(1001,30), QMOM(1001,2), A(1001,3), F(1001),
C 1 EV(1001), G(1001), B(1001), POWER(30), N1(30), N2(30), P(21,31),
C 2 U(30), NEL(30), RATE(21,2,30), VSIG(2,30), PROCESS(4,31), FI(5),
C 3 IDENT(5), NAME(5), MASS(5), GAS(100), NU(100), E(100), TITLE(3),
C 4 Y(2), DY(2), TABLE(21,9), FORM(15), KUDE(8), HEAD(9), IMAGE(60),
C 5 EOVERN(21), EN(21), IOUT(10), OUT(10), KAPT(5), LABEL(5,2),
C 6 KINETIC(6,50), NUMBER(30), GMOLE(100), LINE(250), S(1001)
C
C REAL NO, KB, KTE, MU, MASS, NMOL, N1, N2, NE, MOLWT, IONIZE
C
C INTEGER GAS, TYPE, TITLE, TODAY, LHS, RMS
C
C LOGICAL CONVRGE, OUT, FATAL, LIST, ERROR, STOP, TEST, MISSING,
C 1 OUTSIDE, REJECT, EXPAND, SEARCH, LIBRARY
C
C EQUIVALENCE (A,QM)

```

C		ELECT	59
C	-----	ELECT	60
C	-----	ELECT	61
C	NAMELIST / CONTROL / MESH, IOUT, FATAL, EMAX, ITMAX, TMAX, EPS, 1 TE, IDEG, PCT, SEARCH	ELECT	62
C	NAMELIST / PARAM / TMOL, PTOT, ATM, EUVERN	ELECT	63
C	NAMELIST / SOURCE / UNEDT, BEAM, CREATE, UA, UB, S	ELECT	64
C	-----	ELECT	65
C	-----	ELECT	66
C	-----	ELECT	67
C	-----	ELECT	68
C	-----	ELECT	69
C	DATA KR, EE / 1.38E-23, 1.602E-19 /	ELECT	70
C	DATA EM / .1, .2, .3, .5, .8, 1, 2, 3, 5, 12*0, /	ELECT	71
C	DATA HEAD / 6HE/NTOT, 10H<UP>=3TE/2, 9HEK = D/MU, 10HTE (DEG K), 1 9HV0 (CM/S), 10HMU, CM2/V/S, 9HD, CM2/S, 9HP/NE/NMOL, 2 9HJ/NE=E*VD /	ELECT	72
C	DATA TITLE, KAPT / BHELECTRON, BHKINETICS, BHANALYSIS, 5*1H /	ELECT	73
C	-----	ELECT	74
C	-----	ELECT	75
C	-----	ELECT	76
C	-----	ELECT	77
C	-----	ELECT	78
C	-----	ELECT	79
C	-----	ELECT	80
C	-----	ELECT	81
C	-----	ELECT	82
C	-----	ELECT	83
C	-----	ELECT	84
C	-----	ELECT	85
C	-----	ELECT	86
C	IOUT(10) IS A VECTOR OF OUTPUT OPTIONS -- OPTION I IS SUP- CRESSED IF IOUT(I) = 0, OTHERWISE PROVIDED --	ELECT	87
C	IOUT(1) -- INDIVIDUAL SUMMARY OF PLASMA PARAMETERS FOR C EACH E/N VALUE	ELECT	88
C	IOUT(2) -- TABLE OF ELECTRON DISTRIBUTION FUNCTION FOR C EACH VALUE OF E/N	ELECT	89
C	IOUT(3) -- PLOT OF ELECTRON DISTRIBUTION FUNCTION FOR C EACH VALUE OF E/N	ELECT	90
C	IOUT(4) -- PLOT OF F(U)/FBOLTZ(U,TE) FOR EACH E/N	ELECT	91
C	IOUT(5) -- SUMMARY OF PLASMA PARAMETERS, COLLISION C RATES, AND POWER BALANCE FOR EACH E/N	ELECT	92
C	IOUT(6) -- TABULAR SUMMARY OF INPUT CROSS SECTION DATA	ELECT	93
C	IOUT(7) -- PLOTS OF INPUT CROSS SECTIONS (IOUT(6) MUST C BE SIMULTANEOUSLY SPECIFIED)	ELECT	94
C	IOUT(8) -- TABULAR SUMMARY OF PLASMA PARAMETERS AS A C FUNCTION OF E/N FOR THE GIVEN GAS MIXTURE; TABLES OF POWER PARTITIONING AS A FUNCTION OF C E/N FOR ELASTIC AND INELASTIC PROCESSES.	ELECT	95
C	IOUT(9) -- PLOTS OF PLASMA PARAMETERS AS A FUNCTION OF C E/N FOR A GIVEN GAS MIXTURE (IF AT LEAST 5 CASES HAVE BEEN COMPUTED)	ELECT	96
C	IOUT(10) -- LOG PLOTS OF FORWARD (AND REVERSE) ELECTRON C RATES AS A FUNCTION OF E/N, FOR EACH OF THE INELASTIC COLLISION PROCESSES INCLUDED	ELECT	97
C	MISCELLANEOUS PARAMETERS DEFINED --	ELECT	98
C	EMAX -- MAXIMUM ELECTRON ENERGY (EV)	ELECT	99
C	MESH -- NUMBER OF INTERVALS INTO WHICH THE ENERGY RANGE	ELECT	100
C		ELECT	101
C		ELECT	102
C		ELECT	103
C		ELECT	104
C		ELECT	105
C		ELECT	106
C		ELECT	107
C		ELECT	108
C		ELECT	109
C		ELECT	110
C		ELECT	111
C		ELECT	112
C		ELECT	113
C		ELECT	114
C		ELECT	115

C IS DIVIDED. (MESH ≤ 1000 IS INSURED BY PROGRAM.) ELECT 116  
 C EVCM -- ELECTRIC FIELD (VOLT/CM) ELECT 117  
 C EOVERN -- E/N (1.E-16 VOLT CM<sup>2</sup>) ELECT 118  
 C TE -- INITIAL GUESS FOR ELECTRON TEMPERATURE (DEG K) ELECT 119  
 C ITMAX -- MAXIMUM NUMBER OF ITERATIONS PERMITTED. ELECT 120  
 C TMAX -- MAXIMUM CP TIME FOR OBTAINING CONVERGENCE. ELECT 121  
 C EPS -- CONVERGENCE CRITERION FOR MAXIMUM RELATIVE ELECT 122  
 C CHANGE FOR ALL VALUES OF THE ELECTRON DISTRIBUTION FUNCTION BETWEEN SUCCESSIVE ITERATIONS. ELECT 123  
 C IDEG -- DEGREE OF INTERPOLATION FOR CROSS SECTIONS TO ELECT 124  
 C GENERATE UNIFORM GRID OF VALUES FROM RAW DATA. ELECT 125  
 C  
 C INPUT CONSISTS OF THE FOLLOWING -- ELECT 126  
 C  
 C A) DECK OF ELECTRON CROSS SECTION PACKAGES (ARBITRARILY MANY) ELECT 127  
 C FOR UPDATE OF ELECTRON CROSS SECTION DATA FILE, TERMINATED ELECT 128  
 C BY AN EOF (7/8/9) CARD. ELECT 129  
 C  
 C B) LIST OF (UP TO LIMIT) ELECTRON KINETIC REACTIONS, TERMINATED ELECT 130  
 C BY AN EOF (7/8/9) CARD. THESE REACTIONS FORM THE ELECT 131  
 C BASIS FOR ALL SUBSEQUENT ELECTRON KINETICS CALCULATIONS. ELECT 132  
 C HOWEVER, IF NO REACTIONS ARE ENTERED IN THIS FILE, THE PROGRAM ELECT 133  
 C WILL SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE ELECT 134  
 C AND RETAIN ALL RELEVANT REACTIONS (I.E. WHICH CONTAIN GASES ELECT 135  
 C IN THE SPECIFIED MIXTURE). REGARDLESS OF WHERE THE KINETIC ELECT 136  
 C SCHEME IS FOUND, ONLY THE FIRST NKMAX LEGAL REACTIONS ARE ELECT 137  
 C RETAINED. ELECT 138  
 C  
 C THESE ARE FOLLOWED BY ARBITRARILY MANY OF THE FOLLOWING PACKAGES, ELECT 139  
 C EACH OF WHICH IS TERMINATED BY AN EOF (7/8/9) CARD -- ELECT 140  
 C  
 C 1) SCUNTRL ... S ELECT 141  
 C 2) SPARAM ... S ELECT 142  
 C 3) PACKAGE OF SPECIES CARDS, CONTAINING: NAME, FRACTION P<sub>0</sub>, ELECT 143  
 C ENERGY (EV), MOLECULAR WEIGHT (GM/MOLE). (A10.3E10.3) ELECT 144  
 C  
 C THE FRACTIONAL GAS COMPOSITION IS DEFINED BY THE VALUES ENTERED ELECT 145  
 C FOR P<sub>0</sub> ON THE SPECIES CARDS. IF THE TOTAL OF THESE VALUES IS MORE ELECT 146  
 C THAN 1.0E 08, IT IS ASSUMED THAT THE P<sub>0</sub> VALUES REPRESENT POPULATION ELECT 147  
 C DENSITIES (CM<sup>-3</sup>), AND ANY VALUES ENTERED FOR PTOT OR ATM ARE ELECT 148  
 C IGNORED. IF THE TOTAL IS LESS THAN 1.0E 08, IT IS ASSUMED THAT ELECT 149  
 C THE P<sub>0</sub> VALUES REPRESENT PARTIAL PRESSURES (IN TORR), UNLESS EITHER ELECT 150  
 C PTOT OR ATM IS SPECIFIED ON THE SPARAM ... S CARD, IN WHICH CASE ELECT 151  
 C THEY REPRESENT ONLY FRACTIONAL CONCENTRATIONS. IF BOTH PTOT (IN ELECT 152  
 C TORR) AND ATM (IN ATMOSPHERES) IS SPECIFIED, THE VALUE FOR PTOT ELECT 153  
 C IS ACCEPTED, AND ATM IS IGNORED. ELECT 154  
 C  
 C -----  
 C CALL DATE (TUDAY) ELECT 155  
 C  
 C DIMENSION DECLARATORS -- ELECT 156  
 C MGRID = 1000 ELECT 157  
 C NMAX = 5 ELECT 158  
 C NKMAX = 30 ELECT 159  
 C LIMIT = 50 ELECT 160  
 C MAX = 100 ELECT 161  
 C  
 C ELECT 162  
 C ELECT 163  
 C ELECT 164  
 C ELECT 165  
 C ELECT 166  
 C ELECT 167  
 C ELECT 168  
 C ELECT 169  
 C ELECT 170  
 C ELECT 171  
 C ELECT 172

```

C      SCRATCH FILE FOR E- UPDATE --
C      NTAPE = 7
C      INPUT E- CROSS SECTION TAPE FILE --
C      INPUT = 8
C      REWIND INPUT
C      FILE OF UPDATED E- CROSS SECTIONS --
C      NDATA = 9
C      INPUT CARD FILE --
C      KARDS = 10
C
C      KB = KB/EE
C      LIST = .TRUE.
C      CALL EDITUR (KARDS, LIST)
C      CALL UPDATE (INPUT, NDATA, NTAPE, .FALSE., TODAY)
C      CALL COVER (TITLE,2)
C
C      -----
C      ENTER AN INITIAL REACTION SCHEME, IF DESIRED --
C      -----
C
C      UNLESS SEARCH = TRUE IS SPECIFIED ON SCONTROL...S CARD BELOW, IT
C      IS ASSUMED THAT USER WISHES TO RESTRICT CALCULATIONS TO NO MORE
C      THAN THE FOLLOWING PROCESSES. IF NONE ARE ENTERED, PROGRAM WILL
C      AUTOMATICALLY SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND
C      USE ALL RELEVANT KINETIC PROCESSES INVOLVING SPECIES FOUND IN THE
C      MIXTURE SPECIFIED.
C
C      CALL EDITUR (KARDS,LIST)
C      DO 28 N = 1,LIMIT
C      DO 29 I = 1,6
C      28 KINETIC(I,N) = 1H
C      N = 0
C      LIBRARY = .TRUE.
C      53 READ (5,250) KODE
C      IF (EOF(5)) 50,33
C      33 IF (N.EQ.LIMIT) GO TO 53
C      N = N+1
C      LIBRARY = .FALSE.
C      DO 51 I = 1,6
C      51 KINETIC(I,N) = KODE(I)
C      GO TO 53
C      ALL KINETIC REACTIONS HAVE BEEN STORED.
C
C      -----
C      READ IN PACKAGE OF INPUT PARAMETERS FOR EXECUTION OF ANALYSIS --
C      -----
C
C      50 CALL EDITUR (KARDS,LIST)
C
C      ENTER CALCULATION CONTROL PARAMETERS --
C
C      MESH = 1000
C      EMAX = 5.0
C
C      ELECT    173
C      ELECT    174
C      ELECT    175
C      ELECT    176
C      ELECT    177
C      ELECT    178
C      ELECT    179
C      ELECT    180
C      ELECT    181
C      ELECT    182
C      ELECT    183
C      ELECT    184
C      ELECT    185
C      ELECT    186
C      ELECT    187
C      ELECT    188
C      ELECT    189
C      ELECT    190
C      ELECT    191
C      ELECT    192
C      ELECT    193
C      ELECT    194
C      ELECT    195
C      ELECT    196
C      ELECT    197
C      ELECT    198
C      ELECT    199
C      ELECT    200
C      ELECT    201
C      ELECT    202
C      ELECT    203
C      ELECT    204
C      ELECT    205
C      ELECT    206
C      ELECT    207
C      ELECT    208
C      ELECT    209
C      ELECT    210
C      ELECT    211
C      ELECT    212
C      ELECT    213
C      ELECT    214
C      ELECT    215
C      ELECT    216
C      ELECT    217
C      ELECT    218
C      ELECT    219
C      ELECT    220
C      ELECT    221
C      ELECT    222
C      ELECT    223
C      ELECT    224
C      ELECT    225
C      ELECT    226
C      ELECT    227
C      ELECT    228
C      ELECT    229

```

```

EXPAND = .FALSE.
PCT = 10.
PERCENT = PCT + 1.
TMAX = 50.
ITMAX = 100
EPS = .001
IUEG = 2
NPTS = 21
DO 3 I = 1,10
3 IOUT(I) = 1
DO 31 N = 1,NMAX
NAME(N) = 1H
IDENT(N) = 0
31 FI(N) = MASS(N) = 0.
DO 13 N = 1,MAX
GMOLE(N) = 0.
GAS(N) = 1H
13 E(N) = NO(N) = 0.
DO 18 N = 1,NPTS
18 EUVERN(N) = -1.0
TE = 0.
SEARCH = FATAL = .FALSE.
C
C -----
READ (5,CONTROL)
C
C -----
ENTER EXPERIMENTAL CONDITIONS --
IF (EOF(5)) 99,72
72 DO 42 I = 1,10
42 OUT(I) = IOUT(I).NE.0
TMOL = 300.
PTOT = ATM = 0.
C
C -----
READ (5,PARAM)
C
C -----
IF (EOF(5)) 99,71
71 IF (PTOT.EU.0.) PTOT = 760.*ATM
IF (TMOL.LE.0.) TMOL = 300.
IF (TE.EQ.0.) TE = TMOL
KTE = KB*TE
IF (MESH.GT.MGRID) MESH = MGRID
M = MESH + 1
C
SEARCH = SEARCH.OR.LIBRARY
C
NE = IONIZE = 0.
NTYPE = 0
C
ENTER EXTERNAL DEPOSITION SOURCE DATA --
DNEDT = BEAM = CREATE = 0.
UA = UB = 0.
C
C -----
READ (5,SOURCE)
C
ELECT 230
ELECT 231
ELECT 232
ELECT 233
ELECT 234
ELECT 235
ELECT 236
ELECT 237
ELECT 238
ELECT 239
ELECT 240
ELECT 241
ELECT 242
ELECT 243
ELECT 244
ELECT 245
ELECT 246
ELECT 247
ELECT 248
ELECT 249
ELECT 250
ELECT 251
ELECT 252
ELECT 253
ELECT 254
ELECT 255
ELECT 256
ELECT 257
ELECT 258
ELECT 259
ELECT 260
ELECT 261
ELECT 262
ELECT 263
ELECT 264
ELECT 265
ELECT 266
ELECT 267
ELECT 268
ELECT 269
ELECT 270
ELECT 271
ELECT 272
ELECT 273
ELECT 274
ELECT 275
ELECT 276
ELECT 277
ELECT 278
ELECT 279
ELECT 280
ELECT 281
ELECT 282
ELECT 283
ELECT 284
ELECT 285
ELECT 286

```

```

C           IF (EOF(5)) 99,73          ELECT 287
73  IF (UB.GT.EMAX) UB = EMAX      ELECT 288
     UA = UA*1.000001               ELECT 289
     UB = UB*1.000001               ELECT 290
C           ENTER SPECIES PARAMETERS -- ELECT 291
C
C           -----
C           20 READ (5,110) TYPE, P0, E0, MOLWT   ELECT 292
C           -----
C           IF (EOF(5)) 11,6          ELECT 293
C           REJECT MORE THAN MAX DIFFERENT SPECIES -- ELECT 294
6  IF (INTYPE.EQ.MAX) GF TO 20    ELECT 295
C           PROGRAM USES THE LATEST VALUES READ FOR E(-) OR IONIZE -- ELECT 296
C           IF (TYPE.NE.4HE(-).AND.TYPE.NE.2HE-.AND.TYPE.NE.1HE) GO TO 34 ELECT 297
     NE = P0                      ELECT 298
     GO TO 20                     ELECT 299
34  IF (TYPE.NE.6HIONIZE) GO TO 39 ELECT 300
     IONIZE = P0                  ELECT 301
     GO TO 20                     ELECT 302
C           REJECT DUPLICATION OF SPECIES -- ELECT 303
39  IF (INTYPE.EQ.0) GO TO 64    ELECT 304
DO 62 I = 1,NTYPE                ELECT 305
     IF (TYPE.NE.GAS(I)) GO TO 62   ELECT 306
C           PREVIOUS VALUES FOR GAS(I) ARE REPLACED BY MOST RECENT -- ELECT 307
     E(I) = E0                    ELECT 308
     NO(I) = P0                   ELECT 309
     GMOLE(I) = MOLWT            ELECT 310
     GO TO 20                     ELECT 311
62  CONTINUE                     ELECT 312
C           STORE PARAMETERS FOR THE NEW SPECIES CALLED *TYPE* -- ELECT 313
64  NTYPE = NTYPE+1              ELECT 314
     GAS(INTYPE) = TYPE          ELECT 315
     E(INTYPE) = E0              ELECT 316
     NO(INTYPE) = P0              ELECT 317
     GMOLE(INTYPE) = MOLWT       ELECT 318
     GO TO 20                     ELECT 319
C           PARAMETERS USED FOR MOMENTUM TRANSFER -- ELECT 320
C
11  PRESS = NMOL = NGAS = 0        ELECT 321
DO 69 I = 1,NTYPE                ELECT 322
69  PRESS = PRESS + NO(I)         ELECT 323
C           SORT SPECIES IN DESCENDING ORDER OF COMPOSITION -- ELECT 324
68  PMAX = 0.                     ELECT 325
DO 1 I = 1,NTYPE                ELECT 326
     DO 65 K = 1,NGAS            ELECT 327
     IF (I.EQ.IDENT(K)) GO TO 1  ELECT 328
65  CONTINUE                     ELECT 329
     IF (NO(I).LE.PMAX) GO TO 1  ELECT 330
     PMAX = NO(I)                ELECT 331
     NG = I                       ELECT 332
1  CONTINUE                     ELECT 333
C           REJECT COMPONENTS < 0.01 % FROM MOMENTUM TRANSFER CALCULATIONS -- ELECT 334

```

```

TEST = (NGAS.NF .1).AND.(PMAX.LT.1.0E-04*FI(1))
IF (TEST) GO TO 66                           ELECT 344
NGAS = NGAS + 1                            ELECT 345
IDENT(NGAS) = NG                           ELECT 346
NAME(NGAS) = GAS(NG)                         ELECT 347
MASS(NGAS) = GMOLE(NG)                      ELECT 348
FI(NGAS) = NO(NG)                           ELECT 349
IF (NGAS.LT.NMAX) GO TO 68                  ELECT 350
C                                         ELECT 351
66 NMOL = PRESS                           ELECT 352
DO 67 I = 1,NGAS                          ELECT 353
67 FI(I) = FI(I)/PRESS                     ELECT 354
IF (NMOL.GT.1.0E 08) GO TO 54              ELECT 355
IF (PTOT.EU.0.) PTOT = PRESS               ELECT 356
NMOL = 0.965E 19*PTOT/THOL                ELECT 357
FRACT = NMOL/PRESS                        ELECT 358
DO 9 I = 1+NTYPE                          ELECT 359
9 NO(I) = NO(I)*FRACT                     ELECT 360
54 IF (NE.LE.0.) NE = IONIZE*NMOL          ELECT 361
NP1 = NTYPE+1                            ELECT 362
GAS(NP1) = 4HE(-)                         ELECT 363
E(NP1) = 0.                                ELECT 364
KOUNT = NP1                               ELECT 365
DO 7 N = 1+NKMAX                          ELECT 366
7 U(N) = NI(N) = N2(N) = 0.                 ELECT 367
ELECT 368
C                                         ELECT 369
N = NK = 0                                ELECT 370
C                                         ELECT 371
C -----  

C   SELECT, FROM THE INPUT REACTION QUEUE, ALL (LEGAL) REACTIONS WHICH ELECT 372
C   INVOLVE SPECIES WHICH HAVE BEEN ENTERED --  

C -----  

C   35 IF (SEARCH) GO TO 70                  ELECT 373
IF (N.EQ.LIMIT) GO TO 5                   ELECT 374
N = N+1                                  ELECT 375
DECODE (60,260,KINETIC(1,N)) IMAGE        ELECT 376
GO TO 61                                  ELECT 377
C                                         ELECT 378
C -----  

C   SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND RETAIN ALL ELECT 379
C   RELEVANT REACTIONS (UP TO NKMAX) FOR THE INPUT GAS MIXTURE --  

C -----  

C   70 READ (INDATA,260) (IMAGE(L), L = 1+60) ELECT 380
IF (EOF(INDATA)) 5,63                      ELECT 381
63 N = NK+1                                ELECT 382
C                                         ELECT 383
C   DUMP NUMERICAL DATA --
READ (INDATA)                                ELECT 384
59 READ (INDATA,250) KODE                  ELECT 385
IF (KODE(1).EQ.1H .AND.KODE(2).EQ.1H ) GO TO 61 ELECT 386
GO TO 59                                    ELECT 387
C                                         ELECT 388
61 IF (NK.EQ.NKMAX) GO TO 5                ELECT 389
CALL DEKODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60) ELECT 390
C   RECALL THAT KOUNT IS AUTOMATICALLY UPDATED UPON RETURN FROM DEKODE ELECT 391
C                                         ELECT 392
C                                         ELECT 393
C                                         ELECT 394
C                                         ELECT 395
C                                         ELECT 396
C                                         ELECT 397
C                                         ELECT 398
C                                         ELECT 399
C                                         ELECT 400

```

```

C      ELIMINATE MOMENTUM TRANSFER PROCESSES --
C      IF (LHS.EQ.RHS) GO TO 35
C      IF REACTION CONTAINS NONE OF THE SPECIES ENTERED, REJECT IT --
C
C      DO 55 L = 1,5
C      I = LABEL(L,1)
C      IF (I.LE.0) GO TO 58
C      IF (I.LE.NTYPE1) GO TO 56
C      SR I = LABEL(L,2)
C      IF (I.LE.0) GO TO 55
C      IF (I.LE.NTYPE1) GO TO 56
C      55 CONTINUE
C      GO TO 35
C
C      56 NL = NR = 0
C      UK = 0.
C      L1 = L2 = 0
C      LEVEL = 0
C      DO 48 L = 1,5
C      I = LABEL(L,1)
C      IF (I.EQ.0) GO TO 43
C      UK = UK - E(I)
C      IF (I.NE.NP1) LEVEL = I
C      48 IF (I.EQ.NP1) NL = NL + 1
C      43 IF (NL.EQ.1.AND.L.EQ.3) L1 = LEVEL
C      LEVEL = 0
C      DO 49 L = 1,5
C      I = LABEL(L,2)
C      IF (I.EQ.0) GO TO 38
C      UK = UK + E(I)
C      IF (I.NE.NP1) LEVEL = 1
C      49 IF (I.EQ.NP1) NR = NR + 1
C      38 IF (NR.EQ.1.AND.L.EQ.3) L2 = LEVEL
C      IF (L1.L2.EQ.0) GO TO 35
C
C      NK = NK+1
C      IF (L1.NE.0) N1(NK) = NO(L1)
C      IF (L2.NE.0) N2(NK) = NO(L2)
C      IF (N1(NK)*N2(NK).NE.0.) GO TO 47
C      NK = NK-1
C      GO TO 35
C      47 U(NK) = UK
C      NEL(NK) = NR - NL
C      ENCODE (40+260,PROCESS(1,NK)) (IMAGE(L), L = 1,40)
C      NUMBER(NK) = N
C      ENCODE (60+260,KINETIC(1,N)) (IMAGE(L), L = 1,60)
C      GO TO 35
C
C      5 NKP1 = NK+1
C      REWIND NDATA
C      ENCODE (40+320,PROCESS(1,NKP1))
C
C      IF (.NOT.EXPAND) GO TO 22
C
C      -----

```

ELECT 401  
 ELECT 402  
 ELECT 403  
 ELECT 404  
 ELECT 405  
 ELECT 406  
 ELECT 407  
 ELECT 408  
 ELECT 409  
 ELECT 410  
 ELECT 411  
 ELECT 412  
 ELECT 413  
 ELECT 414  
 ELECT 415  
 ELECT 416  
 ELECT 417  
 ELECT 418  
 ELECT 419  
 ELECT 420  
 ELECT 421  
 ELECT 422  
 ELECT 423  
 ELECT 424  
 ELECT 425  
 ELECT 426  
 ELECT 427  
 ELECT 428  
 ELECT 429  
 ELECT 430  
 ELECT 431  
 ELECT 432  
 ELECT 433  
 ELECT 434  
 ELECT 435  
 ELECT 436  
 ELECT 437  
 ELECT 438  
 ELECT 439  
 ELECT 440  
 ELECT 441  
 ELECT 442  
 ELECT 443  
 ELECT 444  
 ELECT 445  
 ELECT 446  
 ELECT 447  
 ELECT 448  
 ELECT 449  
 ELECT 450  
 ELECT 451  
 ELECT 452  
 ELECT 453  
 ELECT 454  
 ELECT 455  
 ELECT 456  
 ELECT 457

```

C      IF THE BOLTZMANN ANALYSIS BELOW DID NOT CONVERGE, CONTROL RETURNS      ELECT    458
C      TO THIS POINT TO EXPAND THE ELECTRON ENERGY RANGE --                  ELECT    459
C -----
C
C      40 IF (EMAX.GE.5.) GO TO 23                                         ELECT    460
C          EMAX = EMAX + EMAX                                           ELECT    461
C          IF (EMAX.GT.5.) EMAX = 5.                                         ELECT    462
C          GO TO 27                                                 ELECT    463
C      23 EMAX = EMAX + 5.0                                              ELECT    464
C      27 IF (EMAX.GT.25.) GO TO 50                                         ELECT    465
C
C      22 E0 = 0.                                                       ELECT    466
C          DE = EMAX/MESH                                             ELECT    467
C          NA = UA/DE+1                                              ELECT    468
C          NB = UB/DE+1                                              ELECT    469
C          IF ((UA.NE.UB).AND.(NA.EQ.NB)) NB = NA+1                ELECT    470
C          DU = (NB-NA)*DE                                           ELECT    471
C          SI = 0.                                                    ELECT    472
C          IF (DU.GT.0.) SI = 1./DU                                 ELECT    473
C          DO 29 I = 1..M                                         ELECT    474
C              QMOM(I,1) = QMOM(I,2) = 0.                           ELECT    475
C              EV(I) = E0                                         ELECT    476
C              E0 = E0 + DE                                         ELECT    477
C              S(I) = 0.                                              ELECT    478
C              IF (I.LT.NA.OR.I.GT.NB) GO TO 29                      ELECT    479
C              S(I) = SI                                         ELECT    480
C      29 QM(I) = EV(I)*S(I)                                         ELECT    481
C          BEAM = BEAM*DU*SI                                         ELECT    482
C
C      PLOT SOURCE FUNCTION S(U) FOR EXTERNAL DEPOSITION --             ELECT    483
C      IF (BEAM.EQ.0.) GO TO 45                                         ELECT    484
C      CALL SIMPSON (QM, MESH/2, DE, DEPOSIT)
C          DEPOSIT = 1.602E-19*BEAM*DEPOSIT
C          WRITE (6,360) BEAM, DEPOSIT
C          Y0(1) = DY(1) = 0.
C          CALL PLOT (1, M, 1, S, Y0, DY, EV, 0., 0., .TRUE., .TRUE., .TRUE., 1.*TRUE., .TRUE., TITLE, 1, 0)
C
C      WRITE (6,370) TODAY
C
C -----
C      PROCESS THE ELECTRON CROSS SECTION ARRAYS --
C -----
C
C      45 K = 0.                                                       ELECT    498
C          J = 1.                                                       ELECT    499
C          ERROR = .FALSE.                                         ELECT    500
C          DO 57 N = 1,NK                                         ELECT    501
C          I = NUMBER(N)
C          DECODE (60,260,KINETIC(I,I)) IMAGE
C          CALL DECODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60)
C
C      PROCESS THE NTH INELASTIC ELECTRON CROSS SECTION --
C
C          CALL PLASMA (INDATA, MGRID+1, MESH, LHS, RHS, PROCESS, EV, F, G,
C          1 Q(1,N), UTH, UM, KOUNT, GAS, MISSING, REJECT, OUTSIDE, IDEG,
C          2 OUT(6))

```

```

C
STOP = MISSING.OR.REJECT.OR.OUTSIDE
IF (.NOT.STOP) GO TO 32
IF (J.GT.231) GO TO 57
ENCODE (50,340,LINE(J))
J = J+5
K = K+1
ENCODE (50,111,LINE(J)) K, (PROCESS(L,N), L = 1,4)
J = J+5
IF (MISSING) ENCODE (50,115,LINE(J))
IF (MISSING) J = J+5
IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX
IF (OUTSIDE) J = J+5
IF (REJECT) ENCODE (50,117,LINE(J)) EMAX
IF (REJECT) J = J+5
IF (J.LE.225) GO TO 57
ENCODE (100,330,LINE(J))
J = J+10
GO TO 57
C
32 DO 24 L = 1,M
24 Q(L,N) = EV(L)*Q(L,N)
57 ERROR = ERROR.OR.STOP
C
C      PROCESS THE MOMENTUM TRANSFER CROSS SECTIONS --
C
TWHM = 2./1837.
DO 17 I = 1,NGAS
MISSING = REJECT = OUTSIDE = .FALSE.
ENCODE (40,113,KAPT) NAME(I)
ID = IDENT(I)
ID1 = ID + NP1
ID2 = ID*ID + NP1*NP1
ENCODE (10,114,LHS) ID1, ID2
RHS = LHS
C
CALL PLASMA (INDATA, MGRID+1, MESH, RHS, KAPT, EV, F, G, QM,
1 UTH, UM, KOUNT, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(6))
C
STOP = MISSING.OR.REJECT.OR.OUTSIDE.OR.(MASS(I).LE.0.).OR.
1
     (UM.LT.EMAX)
IF (.NOT.STOP) GO TO 36
IF (J.GT.221) GO TO 17
ENCODE (50,340,LINE(J))
J = J+5
K = K+1
ENCODE (50,111,LINE(J)) K, (KAPT(L), L = 1,4)
J = J+5
IF (MISSING) ENCODE (50,115,LINE(J))
IF (MISSING) J = J+5
IF (OUTSIDE) ENCODE (50,116,LINE(J)) EMAX
IF (OUTSIDE) J = J+5
IF (REJECT) ENCODE (50,117,LINE(J)) EMAX
IF (REJECT) J = J+5
IF (UM.LT.EMAX) ENCODE (50,118,LINE(J)) EMAX
IF (UM.LT.EMAX) J = J+5
IF (MASS(I).LE.0.) ENCODE (50,119,LINE(J))

```

	ELECT	515
	ELECT	516
	ELECT	517
	ELECT	518
	ELECT	519
	ELECT	520
	ELECT	521
	ELECT	522
	ELECT	523
	ELECT	524
	ELECT	525
	ELECT	526
	ELECT	527
	ELECT	528
	ELECT	529
	ELECT	530
	ELECT	531
	ELECT	532
	ELECT	533
	ELECT	534
	ELECT	535
	ELECT	536
	ELECT	537
	ELECT	538
	ELECT	539
	ELECT	540
	ELECT	541
	ELECT	542
	ELECT	543
	ELECT	544
	ELECT	545
	ELECT	546
	ELECT	547
	ELECT	548
	ELECT	549
	ELECT	550
	ELECT	551
	ELECT	552
	ELECT	553
	ELECT	554
	ELECT	555
	ELECT	556
	ELECT	557
	ELECT	558
	ELECT	559
	ELECT	560
	ELECT	561
	ELECT	562
	ELECT	563
	ELECT	564
	ELECT	565
	ELECT	566
	ELECT	567
	ELECT	568
	ELECT	569
	ELECT	570
	ELECT	571

```

IF (MASS(I).LE.0.) J = J+5          ELECT 572
IF (J.LE.225) GO TO 17             ELECT 573
ENCODE (100,330,LINE(J))          ELECT 574
J = J+10                          ELECT 575
GO TO 17                           ELECT 576
C                                     ELECT 577
C                                     ELECT 578
C                                     CONSTRUCT TWO MOMENTUM TRANSFER FUNCTIONS WHICH OCCUR IN THE ELECT 579
C                                     BOLTZMANN EQUATION --          ELECT 580
C                                     ELECT 581
C                                     ELECT 582
C                                     ELECT 583
36 FRACT = FI(I)                 ELECT 584
DO 44 L = 1,M                     ELECT 585
FQ = FRACT*QM(L)                 ELECT 586
QMOM(L,1) = QMOM(L,1) + FO       ELECT 587
44 QMOM(L,2) = QMOM(L,2) + FO/MASS(I) ELECT 588
C                                     17 ERROR = ERROR.OR.STOP          ELECT 589
C                                     ELECT 590
C                                     GENERATE SUMMARY OF WARNING DIAGNOSTICS, IF ANY --          ELECT 591
C                                     ELECT 592
C                                     ELECT 593
C                                     ELECT 594
C                                     J = J-1
C                                     IF (J.GT.0) WRITE (6,112) (LINE(L), L = 1,J)          ELECT 595
C                                     DO 46 L = 1,M          ELECT 596
C                                     X = EV(L)          ELECT 597
C                                     XSQ = TWOM*X*X          ELECT 598
C                                     STOP = QMOM(L,1).EQ.0.          ELECT 599
C                                     IF (STOP) GO TO 8          ELECT 600
C                                     A(L,1) = X/NMOL/QMOM(L,1)          ELECT 601
C                                     46 A(L,2) = XSQ/NMOL*QMOM(L,2)          ELECT 602
C                                     XBAR = DE/2.          ELECT 603
C                                     DO 41 I = 1,MESH          ELECT 604
C                                     CALL INTERP (2, XBAR, QMOM(I,1), EV, A(I,1), 1, M)          ELECT 605
C                                     CALL INTERP (2, XBAR, QMOM(I,2), EV, A(I,2), 1, M)          ELECT 606
41 XBAR = XBAR + DE          ELECT 607
QMOM(M,1) = QMOM(MESH,1)          ELECT 608
QMOM(M,2) = QMOM(MESH,2)          ELECT 609
C                                     8 ERROR = ERROR.OR.STOP          ELECT 610
C                                     FATAL = FATAL.AND.ERROR          ELECT 611
C                                     IF (STOP) WRITE (6,350)          ELECT 612
C                                     IF (FATAL) GO TO 50          ELECT 613
C                                     EXPON = EXP(-DE/KTE)          ELECT 614
C                                     FB = 1.          ELECT 615
C                                     DO 37 I = 1,M          ELECT 616
C                                     F(I) = FB          ELECT 617
C                                     37 FB = FB*EXPON          ELECT 618
C                                     ENCODE (40,500,KAPT)          ELECT 619
C                                     PROHIBIT FURTHER PLOTS OR TABULATIONS OF E- CROSS SECTION DATA--          ELECT 620
C                                     OUT(6) = OUT(7) = .FALSE.          ELECT 621

```

```

          ELECT 622
          ELECT 623
          ELECT 624
          ELECT 625
          ELECT 626
          ELECT 627
          ELECT 628

```

```

IF (EXPAND) GO TO 90
C
C -----
C
N = 0
DO 12 I = 1,NPTS
IF (Eovern(I).EQ.-1.0) GO TO 12
N = N+1
Eovern(N) = Eovern(I)
12 CONTINUE
C
C IF NO VALUES OF E/N (UNITS OF 1.0E-16 VOLT CM2) WERE SPECIFIED
C BY INPUT. DEFAULT IS TO THE TABLE VALUES *EN* DEFINED ABOVE --
C
C IF (N.NE.0) GO TO 25
DO 26 I = 1,NPTS
IF (EN(I).EQ.0.) GO TO 25
N = I
26 Eovern(I) = EN(I)
25 NPTS = N
C
NN = 0
15 IF (NN.EQ.NPTS) GO TO 60
NN = NN+1
EVCM = Eovern(NN)*1.0E-16*NMOL
90 CONVRGE = .FALSE.
C
C -----
C
CALL BOLTZ (MGRID=1, MESH, NK, NAME, F1, NGAS, NMOL, TMOL, ITMAX,
1 TMAX, EPS, KAPT, TODAY, OUT, EVCM, NE, PROCESS, U, N1, N2, NEL,
2 S, BEAM, CREATE, EV, Q, QHOM, F, G, A, B, VSIG, POWER, PCOLL,
3 DISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE, ATTACH, VD,
4 MU, D, EK, AMPS, UBAR, TE, CONVRGE, PERCENT)
C
C -----
C
EXPAND = (.NOT.CONVRGE).OR.(ABS(PERCENT).GT.PCT)
EXPAND = EXPAND.AND.(EMAX.LE.25.)
EXPAND = .FALSE.
IF (EXPAND) GO TO 40
IF (.NOT.CONVRGE) GO TO 60
C
PWR = DISCH + DEPOSIT
TABLE(NN,1) = Eovern(NN)
TABLE(NN,2) = UBAR
TABLE(NN,3) = EK
TABLE(NN,4) = TE
TABLE(NN,5) = VD
TABLE(NN,6) = MU
TABLE(NN,7) = D
TABLE(NN,8) = PWR/NMOL
TABLE(NN,9) = AMPS
C
IF (PWR.EU.0.) PWR = AMAX1(ABS(ELASTIC)*ABS(PCOLL)*ABS(DEDT))
P(NN,NKP1) = ELASTIC*100./PWR
DO 4 J = 1:NK
      ELECT 629
      ELECT 630
      ELECT 631
      ELECT 632
      ELECT 633
      ELECT 634
      ELECT 635
      ELECT 636
      ELECT 637
      ELECT 638
      ELECT 639
      ELECT 640
      ELECT 641
      ELECT 642
      ELECT 643
      ELECT 644
      ELECT 645
      ELECT 646
      ELECT 647
      ELECT 648
      ELECT 649
      ELECT 650
      ELECT 651
      ELECT 652
      ELECT 653
      ELECT 654
      ELECT 655
      ELECT 656
      ELECT 657
      ELECT 658
      ELECT 659
      ELECT 660
      ELECT 661
      ELECT 662
      ELECT 663
      ELECT 664
      ELECT 665
      ELECT 666
      ELECT 667
      ELECT 668
      ELECT 669
      ELECT 670
      ELECT 671
      ELECT 672
      ELECT 673
      ELECT 674
      ELECT 675
      ELECT 676
      ELECT 677
      ELECT 678
      ELECT 679
      ELECT 680
      ELECT 681
      ELECT 682
      ELECT 683
      ELECT 684
      ELECT 685

```

```

P(NN,J) = POWER(J)/PWR*100.
IF (VSIG(1,J).GT.0.) RATE(NN,1,J) = ALOG10(VSIG(1,J))
4 IF (VSIG(2,J).GT.0.) RATE(NN,2,J) = ALOG10(VSIG(2,J))

C      GO TO 15
60 IF (NN.EQ.1) GO TO 50
NPTS = NN
NG1 = NGAS-1
NSPACE = 58 - 6*NGAS
ENCODE (150,230,FORM) NSPACE, NG1, NG1

C      -----
C      TABULAR OUTPUT OF PLASMA PARAMETERS FOR THE SPECIFIED GAS MIXTURE
C      FOR SEVERAL VALUES OF E/N --
C      IF (.NOT.OUT(8)) GO TO 2
L = 22-NPTS
IF (L.GT.9) L = 9
ENCODE (10,100,SKIP) L
WRITE (6,SKIP)
WRITE (6,120)
IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL
IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
1,1,NGAS), TMOL
WRITE (6,130)
DO 52 N = 1,NPTS
52 WRITE (6,140) (TABLE(N,I), I = 1,9)
WRITE (6,150) EMAX, MESH, DE

C      TABULAR OUTPUT OF FRACTIONAL POWER PARTITIONING FOR EACH OF THE
C      ELASTIC AND INELASTIC COLLISION PROCESSES AS A FUNCTION OF E/N --
C      M1 = 1
14 IF (M1.GT.NKPI) GO TO 2
M2 = M1+20
IF (NKPI.LT.M2) M2 = NKPI
LSKIP = 20-M2+M1
IF (LSKIP.LE.0) LSKIP = 1
ENCODE (20,310,DY) LSKIP
L1 = 1
10 IF (L1.GT.NPTS) GO TO 21
L2 = L1+5
IF (L2.GT.NPTS) L2 = NPTS
NSPACE = (5-L2+L1)*15/2 + 5
L21 = L2-L1
NM2 = NSPACE-2
NDASH = 54 + 15*L21
ENCODE (100,280+LABEL) NSPACE, L21, NSPACE
ENCODE (20,290,Y0) NM2, NDASH
ENCODE (50,300,KAPT) NSPACE, L21
WRITE (6,UY)
WRITE (6,270)
IF (NGAS.EQ.1) WRITE (6,220) NAME(1), THUL
IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
1,1,NGAS), THUL
WRITE (6+LABEL) (EOVERN(L), L = L1..L2)

```

```

      WRITE (6,Y0)
      DO 19 N = M1,M2
19   WRITE (6,KAPT) (PROCESS(L,N), L = 1,4), (P(L,N), L = L1,L2)
      WRITE (6,Y0)
      L1 = L2+1
      GO TO 10
21   M1 = M2+1
      GO TO 14
C
C-----  

C C PLOTS OF PLASMA PARAMETERS FOR THE SPECIFIED GAS MIXTURE AS A  

C C FUNCTION OF E/NTOT --
C
2   IF (NPTS.LT.5) GO TO 50
Y0(1) = Y0(2) = DY(1) = DY(2) = 0.
IF (.NOT.OUT(9)) GO TO 30
C
      WRITE (6,170)
      IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
I 1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,2), Y0, DY, EOVERN, 0., 0.,
I .TRUE., .TRUE., .TRUE., .TRUE., .TRUE., HEAD(2), 2, 0)
      WRITE (6,200) TODAY
C
      WRITE (6,180)
      IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
I 1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,4), Y0, DY, EOVERN, 0., 0.,
I .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(4), 2, 0)
      WRITE (6,200) TODAY
C
      WRITE (6,190)
      IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
I 1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,6), Y0, DY, EOVERN, 0., 0.,
I .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(6), 2, 0)
      WRITE (6,200) TODAY
C
      WRITE (6,210)
      IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =
I 1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,8), Y0, DY, EOVERN, 0., 0.,
I .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(8), 2, 0)
      WRITE (6,200) TODAY
C
C PLOTS OF ENDOOTHERMIC ELECTRON COLLISION RATES <VSIG> (CM3/SEC) FOR
C ALL OF THE INELASTIC COLLISION PROCESSES, AS A FUNCTION OF E/NTOT
C
30  IF (.NOT.OUT(10)) GO TO 50
      DO 16 J = 1,NK
      NPLOT = 0
      IF (VSIG(1,J).GT.0.) NPLOT = 1
      ELECT    743
      ELECT    744
      ELECT    745
      ELECT    746
      ELECT    747
      ELECT    748
      ELECT    749
      ELECT    750
      ELECT    751
      ELECT    752
      ELECT    753
      ELECT    754
      ELECT    755
      ELECT    756
      ELECT    757
      ELECT    758
      ELECT    759
      ELECT    760
      ELECT    761
      ELECT    762
      ELECT    763
      ELECT    764
      ELECT    765
      ELECT    766
      ELECT    767
      ELECT    768
      ELECT    769
      ELECT    770
      ELECT    771
      ELECT    772
      ELECT    773
      ELECT    774
      ELECT    775
      ELECT    776
      ELECT    777
      ELECT    778
      ELECT    779
      ELECT    780
      ELECT    781
      ELECT    782
      ELECT    783
      ELECT    784
      ELECT    785
      ELECT    786
      ELECT    787
      ELECT    788
      ELECT    789
      ELECT    790
      ELECT    791
      ELECT    792
      ELECT    793
      ELECT    794
      ELECT    795
      ELECT    796
      ELECT    797
      ELECT    798
      ELECT    799

```

```

IF (NPLT, EQ.0) GO TO 16          ELECT 800
IF (NEL(J), EQ.0) NPLT = NPLT+1   ELECT 801
ENCODE (20,160,KAPT) J, J        ELECT 802
WRITE (6,240) J, (PROCESS(I,J), I=1,4)    ELECT 803
IF (NGAS.EU,1) WRITE (6,220) NAME(1), TMOL  ELECT 804
IF (NGAS.GT,1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (F(I), I =
1,1,NGAS), TMOL                ELECT 805
CALL PLOT (21, NPTS, 1, RATE(1,1,J), 0., 0., EOVERN, 0., 0.,
1, .TRUE., .TRUE., .TRUE., .TRUE., .TRUE., KAPT, NPLT, 0)  ELECT 806
WRITE (6,200) TODAY               ELECT 807
16 CONTINUE                      ELECT 808
C                                     ELECT 809
C                                     ELECT 810
C                                     GO TO 50  ELECT 811
C ----- FORMAT STATEMENTS -----  ELECT 812
C 100 FORMAT (5H(1H),,I1,4H(/))    ELECT 813
C 110 FORMAT (A10,3E10.3)         ELECT 814
C 111 FORMAT (I2*) *4A10.5X)      ELECT 815
C 112 FORMAT (1H1//20X*WARNING --//25X*ERRORS OCCURRED FOR THE INPUT E-
1CROSS SECTIONS FOR THE FOLLOWING ELECTRON COLLISION //25X*PROCESSE ELECT 816
2S. IF THE PROGRAM IS EXECUTED, A ZERO CROSS SECTION WILL BE ASSUM ELECT 817
3ED:////20X,90(1H-)//(40X,5A10))  ELECT 818
C 113 FORMAT (*MOMENTUM TRANSFER FOR *,A10)  ELECT 819
C 114 FORMAT (I4,I6)              ELECT 820
C 115 FORMAT (10(1H )*NO E- CROSS SECTION DATA WAS FOUND.)  ELECT 821
C 116 FORMAT (10(1H )*SIGMA = 0 IN THE RANGE (0.*F4.1*) EV.**)  ELECT 822
C 117 FORMAT (10(1H )*ERRORS OCCURRED IN CROSS SECTION DATA.)  ELECT 823
C 118 FORMAT (10(1H )*CROSS SECTION DOES NOT SPAN (0.*F4.1*) EV.**)  ELECT 824
C 119 FORMAT (10(1H )*THE MASS ENTERED FOR THIS SPECIES IS S 0*)  ELECT 825
C 120 FORMAT (42X*SUMMARY OF PLASMA PARAMETERS AS A FUNCTION OF E/NTOT*)  ELECT 826
C 130 FORMAT (/14X,*E/NTOT*,7X,*UBAR*,3X,*EK = D/MU*,5X,*TE*,10X,*VD*,,
1 12X,*MU*,13X,*D*9X*P/NMOL/NE*3X,11HJ/NE = E*VD/11X,* (E-16 V CM2)*
2 24X*(EV)*,5X,* (EV)*,6X,* (DEG K)*,5X,* (CM/S)*,7X,* (CM2/V/S)*,6X,
3 *(CM2/S)*,4X,* (WCM3/ELMOL)*,1X,* (AMP CM/ELECT)*,10X,116(1H-))  ELECT 827
C 140 FORMAT (/10X,F10.3,F11.3,F10.3,F11.0,1P5E14.3)  ELECT 828
C 150 FORMAT (/10X,116(1H-)//36X*(THE ELECTRON ENERGY RANGE [0.*F5.1*] E ELECT 829
1 V WAS DIVIDED INTO *,14/36X, *INTERVALS, GIVING AN ENERGY RESOLU ELECT 830
2 TION OF DU *=*,1P6E9.2,* EV.1*)  ELECT 831
C 160 FORMAT (6HVSIGF(,I2+2H) ,6HVSIGR(,I2+2H) )  ELECT 832
C 170 FORMAT (1H1/44X*AVVERAGE AND CHARACTERISTIC ELECTRON ENERGY (EV)*)  ELECT 833
C                                     ELECT 834
C                                     ELECT 835
C                                     ELECT 836
C                                     ELECT 837
C                                     ELECT 838
C                                     ELECT 839
C                                     ELECT 840
C                                     ELECT 841
C                                     ELECT 842
C                                     ELECT 843
C                                     ELECT 844
C                                     ELECT 845
C                                     ELECT 846
C                                     ELECT 847
C                                     ELECT 848
C                                     ELECT 849
C                                     ELECT 850
C                                     ELECT 851
C                                     ELECT 852
C                                     ELECT 853
C                                     ELECT 854
C                                     ELECT 855
C                                     ELECT 856

```

C	180 FORMAT (1H142X*EFFECTIVE ELECTRON TEMPERATURE AND DRIFT VELOCITY*)	ELECT	857
C	190 FORMAT (1H1/46X*ELECTRON MOBILITY AND DIFFUSION COEFFICIENT*)	ELECT	858
C	200 FORMAT (/56X*E/NTOT (1.0 E-16 VOLT CM2)*18X*DR. WILLIAM B. LACINA, 1*A1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)	ELECT	859
C	210 FORMAT (1H142X*TOTAL ELECTRICAL POWER AND CURRENT DENSITY PER NE*)	ELECT	860
C	220 FORMAT (55X,*PURE *,A3,*, TMOL =*,F5.0,*, DEG K*)	ELECT	861
C	230 FORMAT (1H(*I2.6HXA3,,I2,20H(*/*A3),* = *2PF6.2*,I2,38H(* /*F6.2), 1*, TMOL =*,OPF5.0* DEG K*)	ELECT	862
C	240 FORMAT (1H(.33X,*LOG PLOT OF ELECTRON COLLISION RATES <VSIG> (CM3/ 1SEC) FOR REACTION (*I2*)*/55X,4A10/)	ELECT	863
C	250 FORMAT (8A10)	ELECT	864
C	260 FORMAT (80A1)	ELECT	865
C	270 FORMAT (18X*SUMMARY OF FRACTIONAL % POWER PARTITION FOR ALL ELECTR ION COLLISION PROCESSES AS A FUNCTION OF E/NTOT*)	ELECT	866
C	280 FORMAT (*(*I2*X*4X*REACTION=15X*E/NTOT --> *F7.2,*I1*F15.2/*, 1I2*X*24X*(1.0E-16 V CM2)*)*)	ELECT	867
C	290 FORMAT (1H(*I2*X,*I3*(1H-1)/*)	ELECT	868
C	300 FORMAT (1H(*I2*X,4A10,F8.3,*I1*F15.3/*)	ELECT	869
C	310 FORMAT (5H(1H1,,I2*(/))*)	ELECT	870
C	320 FORMAT (*MOMENTUM TRANSFER COLLISIONS*)	ELECT	871
C	330 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)	ELECT	872
C	340 FORMAT (50X)	ELECT	873
C	350 FORMAT (/30X*FATAL ERROR -- MOMENTUM TRANSFER COLLISION FREQUENCY* 1/30X*BECAME ZERO AT SOME POINT; ANALYSIS HAS 1/QM TERMS.*)	ELECT	874
C	360 FORMAT (1H1.22X, *NORMALIZED EXTERNAL SOURCE FUNCTION S(U) FOR C REACTION OF ELECTRONS IN THE ENERGY RANGE [U,U+DU]*/25X*TOTAL ELECT 2RON CREATION RATE = *1PE10.3* CM-3/SEC. POWER DEPOSITION = *E10.3* 3* WATT/CM3*)	ELECT	875
C	370 FORMAT (/62X*ELECTRON ENERGY U (EV)*16X*DR. WILLIAM B. LACINA,*A1/ 1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)	ELECT	876
C	500 FORMAT (7X*ELECTRON KINETICS ANALYSIS*7X)	ELECT	877
C	-----	ELECT	878
C	99 CALL EXIT END	ELECT	879
		ELECT	880
		ELECT	881
		ELECT	882
		ELECT	883
		ELECT	884
		ELECT	885
		ELECT	886
		ELECT	887
		ELECT	888
		ELECT	889
		ELECT	890
		ELECT	891
		ELECT	892
		ELECT	893
		ELECT	894
		ELECT	895
		ELECT	896
		ELECT	897
		ELECT	898
		ELECT	899
		ELECT	900
		ELECT	901
		ELECT	902
		ELECT	903
		ELECT	904
		ELECT	905
		ELECT	906
		ELECT	907
		ELECT	908
		ELECT	909
		ELECT	910
		ELECT	911
		ELECT	912
		ELECT	913