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)

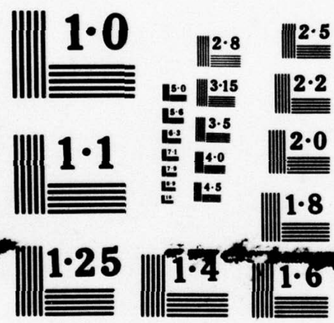
JAN 79 W B LACINA N00014-78-C-0499

UNCLASSIFIED NRTC-79-TR-VOL-2 NL

1 OF 2

ADA

066156


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

NRTC-79-7R

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

William B. Lacina

Sponsored by

OFFICE OF NAVAL RESEARCH
Department of the Navy

Contract No. N00014-78-C-0499

NORTHROP
Research and Technology Center

(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) Jan ~~uary,~~ 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) 131 p.

Sponsored by

OFFICE OF NAVAL RESEARCH
Department of the Navy

Contract No. N00014-78-C-0499 (15)

407 696

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 6. PERFORMING ORG. REPORT NUMBER NRTC-79-7R
7. AUTHOR(s) William B. Lacina (AD-A066155)		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0499
9. PERFORMING ORGANIZATION NAME AND ADDRESS NORTHROP Research and Technology Center, One Research Park, Palos Verdes Peninsula, Ca. 90274		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Attn.: 613C:MAK
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Dept. of the Navy, 800 North Quincy Street, Arlington, Virginia 22217		12. REPORT DATE January, 1979 13. NUMBER OF PAGES 126
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research, Pasadena, California		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

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.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

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:	NORTHROP 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

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

```
C
      REAL NO, N1, N2, NDOT, KF, KR, NTOT, NOUT, NMOL, MASS, MOLWT, KB,
     1   LDIDT, KTE, MU, ISUS, NE, IBEAM, INDUCT, KVOLT, LOSS, LENGTH,
     2   JBEAM, JSUS, KVCM, IONIZE, JB
C
      INTEGER GAS, TYPE, TITLE, RATE, LHS, RHS, TODAY, GENDATE, WARN,
     1   WORD
C
      LOGICAL CONVRGE, ERROR, REJECT, STOP, FATAL, LIST, OUT, ELECT,
     1   EXPAND, TEST, MISSING, OUTSIDE, STIM, END, MODIFY, ILLEGAL,
     2   ARC, FLAG, INTRP, PLOTS, REPEAT, ERRORS, FE
C
      EQUIVALENCE (B,QM)
C
C  -----------------------------------------------------------------
C
C     LABELED COMMON BLOCKS --
C
      COMMON / DATA / RATEK, KF, KR, VSIG, E
C
      COMMON / DISCH / INDUCT, CAPAC, RESIST, MU, AREA, DIST
C
      COMMON / GAINS / GNET, GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY
C
      COMMON / SOURCE / UPLUS, JBEAM, DVDX, FACTOR, ENERGY, SB, S0
C
      COMMON / CONST / NTOT, TMOL, FREQ, HNU
C
      COMMON / TIMES / TR, TF, TFALL, TC, TB, JB, INTRP, NPTS, UNITS
C
C  -----------------------------------------------------------------
C
C     DATA INITIALIZATION --
C
      DATA MESH, ITMAX, METHOD, MAXDER, IDEG, IO, NSIZE, LIMIT, NCYCLE,
     1   KOUNT / 500, 100, 1, 7, 2, 10*1, 10, 2*200, 20 /
C
      DATA EMAX, TMAX, EPS, ETA, RE, ELIMIT, PCT, PER, WARN / 20.0,
     1   50.0, 3*0.001, 30.0, 2*5.0, 8HWARNING: /
C
      DATA KB, EE, PI, EFMT, FFMT / 1.38E-23, 1.602E-19, 3.14159,
     1   9H(1PE10.2), 7H(F10.2) /
C
      DATA OUT, EXPAND, REPEAT, FATAL, FE, ERROR, ERRORS, PLOTS, FLAG,
     1   LIST / 64*.FALSE., 201*.TRUE. /
C
C  -----------------------------------------------------------------
C
C     NAMELIST PARAMETERS --
C
      NAMELIST / CONTROL / EMAX, MESH, EXPAND, ITMAX, TMAX, EPS, IDEG,
     1   PCT, KOUNT, NCYCLE, LIMIT, MAXDER, METHOD, ETA, FATAL, IO
C
      NAMELIST / PARAM / TPULSE, TMOL, TE, PTOT, ATM
C
      NAMELIST / RATES / KF, KR
C
```

```
      NAMELIST / EBEAM / JBEAM, ENERGY, FACTOR, TR, TF, TC, TFALL, UA,
     1   UB, JB, TB, UNITS
C
      NAMELIST / CIRCUIT / KVOLT, INDUCT, CAPAC, RESIST, AREA, DIST
C
      NAMELIST / OPTICAL / LOSS, REFLECT, GAMMA, AREA, LENGTH, CAVITY,
     1   OMEGA
C
C ------------------------------------------------------------------
C
C     ARRAY STORAGE IS DEFINED BY THE FOLLOWING DIMENSION DECLARATORS --
C
      MAX = 200
      M1 = MAX+1
C     (MAX = MAXIMUM VALUE OF NCYCLE, THE NUMBER OF INTERVALS INTO WHICH
C     THE TOTAL PULSE DURATION IS SUBDIVIDED FOR BOLTZMANN CALCULATIONS
C     AND OUTPUT GENERATION.)
      MGRID = 500
C     (MGRID = MAXIMUM NUMBER OF BINS INTO WHICH THE ELECTRON ENERGY
C     RANGE MAY BE PARTITIONED IN THE E- KINETICS ANALYSIS.)
      KMAX = 200
C     (KMAX = MAXIMUM NUMBER OF REACTIONS)
      NMAX = 30
C     (NMAX = MAXIMUM VALUE OF NTYPE, THE NUMBER OF SPECIES)
      NMAXP2 = NMAX+2
C     (NMAXP2 IS MAXIMUM NUMBER OF EQUATIONS FOR SPECIES AND CIRCUIT.)
      NKMAX = 25
C     (NKMAX = MAXIMUM VALUE OF NK = NUMBER OF SECONDARY E- REACTIONS)
C
C     IF IT IS DESIRED TO CHANGE DIMENSION STORAGE, THE ABOVE DECLARATOR
C     VALUES MUST BE MODIFIED, AND THE FOLLOWING MISCELLANEOUS ARRAYS
C     MUST BE CHANGED (AS SHOWN) IN THE DIMENSION STATEMENT --
C
C     KF(KMAX), KR(KMAX), GAS(KMAX), LEV1(NKMAX), LEV2(NKMAX), N1(NKMAX)
C     N2(NKMAX), VSIG(2*NKMAX), POWER(NKMAX), U(NKMAX), NEL(NKMAX),
C     PROCESS(4,NKMAX), Q(MGRID+1,NKMAX), QM(MGRID+1), QMOM(MGRID+1,2),
C     F(MGRID+1), G(MGRID+1), A(MGRID+1,3), B(MGRID+1), EV(MGRID+1),
C     FI(NMAX), NAME(NMAX), MASS(NMAX), E(NMAX), PLOTS(NMAX), PMAX(NMAX)
C     NTIME(NMAX), RPCT(NMAX), NO(NMAXP2), POP(NMAXP2), NDOT(NMAXP2),
C     DNYDTN(NMAXP2,8), NOUT(MAX+1,2,NMAX+2), YMAX(NMAXP2), ERR(NMAXP2),
C     SCRTCH(NMAXP2,12), PHI(NMAX+2,2*NMAX+5), S(MGRID+1), TIME(MAX+1),
C     ISUS(MAX+1,2), COND(MAX+1,2), IBEAM(MAX+1), RAD(MAX+1,3),
C     ALPHA(MAX+1,4), RATEK(KMAX)
C
C     NOTE ALSO THAT DIMENSIONS OF ARRAYS WHICH OCCUR IN LABELLED COMMON
C     BLOCKS MUST BE ACCORDINGLY MODIFIED IN ANY SUBROUTINE WHERE THEY
C     OCCUR.  IN PARTICULAR, COMMON / DATA / CONTAINS DIMENSIONS IN THE
C     SYNTHESIZED SUBROUTINES *JACOB* AND *DNOT*.
C
C     IF CERTAIN DIMENSION DECLARATORS DO NOT AGREE WITH THOSE ON AN IN-
C     PUT DATA FILE (NSCRTCH) ACCESSED DURING EXECUTION, AN EXIT OCCURS.
C
C ------------------------------------------------------------------
C
C     FILE USAGE --
C
C     INPUT CARD FILE --
```

```
      KARDS = 10                                                        LASER
C                                                                       LASER
C     SUBROUTINE *EDITOR*, WHICH EDITS THE INPUT CARD IMAGES, CREATES AN LASER
C     EFFECTIVE INPUT FILE ON TAPE 5, WHICH IS THEN SUBSEQUENTLY USED.  LASER
C                                                                       LASER
C     INPUT E- CROSS SECTION FILE --                                    LASER
      INPUT = 8                                                         LASER
      REWIND INPUT                                                      LASER
C                                                                       LASER
C     FILE OF UPDATED E- CROSS SECTIONS --                              LASER
      NDATA = 9                                                         LASER
      REWIND NDATA                                                      LASER
C                                                                       LASER
C     SCRATCH TAPE USED IN PROGRAM SETUP --                             LASER
      NSCRTCH = 4                                                       LASER
C                                                                       LASER
C     FILE FOR SYNTHESIS OF RATE SUBROUTINES --                         LASER
      NTAPE = 7                                                         LASER
      REWIND NTAPE                                                      LASER
      MTAPE = 3                                                         LASER
      REWIND MTAPE                                                      LASER
      LTAPE = 2                                                         LASER
      REWIND LTAPE                                                      LASER
C                                                                       LASER
C ----------------------------------------------------------------------LASER
C                                                                       LASER
      DO 7 NK = 1,NKMAX                                                 LASER
    7 LEV1(NK) = LEV2(NK) = 0                                           LASER
      DO 1 I = 1,NMAXP2                                                 LASER
      YMAX(I) = 1.0                                                     LASER
      DO 1 N = 1,8                                                      LASER
    1 DNYDTN(I,N) = 0.                                                  LASER
C                                                                       LASER
C     C IS JUST SOME ARBITRARY CONSTANT USED TO TAG RATES WHICH ARE NOT LASER
C     EXPLICITLY INITIALIZED LATER BY INPUT --                          LASER
      C = -SQRT(1.25963782)                                             LASER
      DO 62 K = 1,KMAX                                                  LASER
   62 KF(K) = KR(K) = C                                                 LASER
C                                                                       LASER
      KB = KB/EE                                                        LASER
C     ----------------                                                  LASER
      CALL DATE (TODAY)                                                 LASER
C     ----------------                                                  LASER
C                                                                       LASER
      REWIND NSCRTCH                                                    LASER
      READ (NSCRTCH) TITLE, GENDATE                                     LASER
      IF (EOF(NSCRTCH)) 4,3                                             LASER
C                                                                       LASER
C ----------------------------------------------------------------------LASER
C                   SYNTHESIS OF MOLECULAR KINETICS SUBROUTINES         LASER
C ----------------------------------------------------------------------LASER
C                                                                       LASER
C     SYNTHESIZE SUBROUTINE *DNDT* AND *JACOB* TO CALCULATE THE VALUES  LASER
C     OF DNI/DT AND D/DNJ(DNI/DT) AS A FUNCTION OF TIME, AND SUBROUTINE LASER
C     *LEVELS* TO DEFINE THE CORRESPONDENCE OF LOWER AND UPPER STATES IN LASER
C     THE E- KINETICS ANALYSIS WITH THE MOLECULAR SPECIES LABELS --     LASER
C                                                                       LASER
```

```
C     READ DATA BLOCK NO. 1 --
    4 CALL EDITOR (KARDS,LIST)
      GENDATE = TODAY
C
      CALL SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, INPUT, NSIZE, NMAX, GAS,
     1   KMAX, NKMAX, LEV1, LEV2, TODAY)
C
C     AT THE TERMINATION OF PROGRAM GENERATION, THE SYNTHESIZED SUBROU-
C     TINES (80 BCD CHARACTER RECORDS) ARE STORED ON FILE *MTAPE* = 3,
C     WHICH IS USED AS A SOURCE FILE TO BE COMPILED (AND CATALOGED AS A
C     BCD UPDATE FILE, IF DESIRED). NUMERICAL DATA ASSOCIATED WITH RATE
C     CONSTANTS AND INFORMATION CHARACTERIZING THE REACTION SCHEME EN-
C     COUNTERED (UPON WHICH THE SYNTHESIZED SUBROUTINES ARE BASED) IS
C     STORED ON FILE *NSCRTCH* = 4.  IF FUTURE EXECUTION OF THE ANALYSIS
C     IS INTENDED, *NSCRTCH* MUST ALSO BE CATALOGUED AND SAVED.
C
      CALL EXIT
C
C ----------------------------------------------------------------------
C
C     GENERATE AN UPDATED ELECTRON CROSS SECTION FILE --
C     READ DATA BLOCK NO. 2 --
    3 CALL EDITOR (KARDS,LIST)
      CALL UPDATE (INPUT, NDATA, NTAPE, .NOT.LIST, TODAY)
C
C     --------------------
      CALL COVER (TITLE,2)
C     --------------------
C
C     READ DATA BLOCK NO. 3 --
      CALL EDITOR (KARDS,.NOT.LIST)
C
C     ----------------------------------------
C     GENERATE COMMENT CARD INFORMATION, IF ANY --
C     ----------------------------------------
      LC = 0
   55 READ (5,101) IMAGE
      IF (EOF(5)) 56,59
   59 IF (LC.GT.0) GO TO 57
      WRITE (6,304)
      WRITE (6,302)
   57 LC = LC+1
      WRITE (6,303) IMAGE
      IF (LC.NE.30) GO TO 55
         WRITE (6,302)
         LC = 0
         GO TO 55
   56 IF (LC.NE.0) WRITE (6,302)
C
C     READ DATA BLOCK NO. 4 --
      CALL EDITOR (KARDS,LIST)
C
C     ------------------------------------------------------------------
C     READ GENERAL CONTROL PARAMETERS THAT DEFINE ACCURACIES, SCOPE OF
C     CALCULATIONS, ITERATION LIMITS, INTERPOLATION ORDER, METHOD OF
C     INTEGRATION, AND OUTPUT OPTIONS --
C     ------------------------------------------------------------------
```

```
C
C     -----------------
      READ (5,CONTROL)
C     -----------------
C
      IF (METHOD.NE.0.AND.METHOD.NE.2) METHOD = 1
      IF (MAXDER.GT.8) MAXDER = 8
      IF (MAXDER.EQ.8.AND.METHOD.NE.0) MAXDER = 7
      IF (NCYCLE.LE.0) NCYCLE = 1
      IF (NCYCLE.GT.MAX) NCYCLE = MAX
      IF (LIMIT.LT.0) LIMIT = 0
      IF (LIMIT.GT.NCYCLE) LIMIT = NCYCLE
C
      READ (NSCRTCH) MAXGAS, NTYPE, MAXK, KTYPE, MAXNK, NK, ERRORS
C
      ERROR(1) = ERRORS
      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
      ERROR(9) = (NTYPE.GT.NMAX).OR.(KTYPE.GT.KMAX).OR.(NK.GT.NKMAX)
C
C     TRUNCATE DATA FILE *NSCRTCH* IF NECESSARY --
      IF (NTYPE.GT.NMAX) NTYPE = NMAX
      IF (KTYPE.GT.KMAX) KTYPE = KMAX
      IF (NK.GT.NKMAX) NK = NKMAX
C
      READ (NSCRTCH) (GAS(N), N = 1,NTYPE)
      READ (NSCRTCH) (LEV1(N), LEV2(N), N = 1,NK)
C
      DO 72 I = 1,10
   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.
      TMOL = 300.
C
C     ---------------
      READ (5,PARAM)
C     ---------------
C
      IF (MESH.GT.MGRID) MESH = MGRID
      MESHP1 = MESH+1
      IF (TMOL.LE.0.) TMOL = 300.
      IF (TE.LE.0.) TE = TMOL
      IF (PTOT.EQ.0.) PTOT = 760.*ATM
      ATM = PTOT/760.
      UNIT = 1.0
      IF (TPULSE.LE.0.) GO TO 42
      TT = TPULSE
   43 IF (TT.GT.1) GO TO 42
      TT = 1000.*TT
      UNIT = UNIT/1000.
```

```
      GO TO 43
   42 TOUT = TPULSE/NCYCLE
      DTIME = TOUT/UNIT
C
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     ----------------
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     -----------------------------------------------------------------
C     READ EXPERIMENTAL ELECTRICAL AND CIRCUIT PARAMETERS --
C     -----------------------------------------------------------------
C
      TR = 0.
      TF = TC = 10000.
      TFALL = 1.0
      DO 517 I = 1,21
  517 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     ---------------
C
      IF (UB.GT.EMAX) UB = EMAX
C     FOR THE SQUARE WAVE S(U) = 1, UA ≤ U ≤ UB, THE AVERAGE ENERGY
C     UPLUS = <U+> IS GIVEN BY --
      UPLUS = (UA + UB)/2.
      SMAX = JB(1)
      IF (SMAX.LT.0.) GO TO 519
```

```
      NPTS = 1
      T0 = TB(1)
      DO 515 I = 2,21
      T1 = TB(I)
      IF (T1.LE.T0) GO TO 518
      NPTS = NPTS+1
      SI = JB(I)
      IF (SI.LT.0.) GO TO 519
      IF (SI.GT.SMAX) SMAX = SI
  515 T0 = T1
  518 INTRP = NPTS.GT.1
C     COMPUTE NORMALIZED E-BEAM CURRENT DENSITY SHAPE FUNCTION --
      DO 521 I = 1,NPTS
  521 JB(I) = JB(I)/SMAX
      GO TO 520
  519 INTRP = .FALSE.
C
  520 KVOLT = 0.
      POISCH = 0.
      AREA = DIST = 1.
      INDUCT = RESIST = 0.
      CAPAC = 1.0
C
C     ----------------
      READ (5,CIRCUIT)
C     ----------------
C
      VOLT = 1000.*KVOLT
      ELECT = KVOLT.NE.0.
      IF (DIST.EQ.0.) DIST = 1.0
C
      ELECT = ELECT.AND.NK.NE.0
      NEQ = NTYPE
      IF (ELECT) NEQ = NEQ+2
      NP1 = NTYPE+1
      NP2 = NTYPE+2
C
C     -----------------------------------------------------------------
C     READ UPDATED RATES FOR ALL PROCESSES --
C     -----------------------------------------------------------------
C
C     READ NUMERICAL RATE DATA PROVIDED AT THE TIME OF PROGRAM SYNTHESIS
C     (STORED ON TAPE NSCRTCH), AND (POSSIBLY) MODIFIED BY INPUT FROM
C     THE $RATES ... $ CARD.  NOTE THAT CERTAIN RATES ARE ABSENT IF THEY
C     DID NOT APPEAR AT PROGRAM SYNTHESIS.  THESE RATES ARE NOT ACCES-
C     SIBLE BY INPUT, AND ARE IGNORED IF AN ATTEMPT IS MADE TO SPECIFY
C     THEM ON THE $RATES ... $ CARD.  HOWEVER, FOR SECONDARY ELECTRON
C     COLLISIONS (FOR WHICH RATES ARE NORMALLY OBTAINED BY DEFAULT TO
C     E- KINETICS CALCULATIONS), FIXED INPUT VALUES FOR RATES MAY BE
C     ASSIGNED BY THE $RATES ... $ CARD IF THERE IS NO ELECTRIC FIELD
C     (I.E., EVCM = 0 SPECIFIED ON THE FOREGOING $CIRCUIT ... $ CARD).
C
C     --------------
      READ (5,RATES)
C     --------------
C
      REWIND MTAPE
```

```
      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 (NSCRTCH) LSUM, LSUM, LABEL, RATE, FK, RK, RNAME, COMMENT
      IF (EOF(NSCRTCH)) 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.I) NL = NL+1
  503 IF (LABEL(J,2).EQ.I) 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.1HX) 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
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.C) KF(K) = 0.
         VSIG(1,N) = KF(K)
   76 IF (R2.NE.1HV) GO TO 73
         IF (KR(K).EQ.C) KR(K) = 0.
         VSIG(2,N) = KR(K)
   73 IF (KF(K).EQ.C) GO TO 67
      IF (IMAGE(1).NE.1H ) GO TO 67
         KF(K) = C
         L = L+1
```

```
         J = 1HF                                                          LASER
         NFLAG = 2H**                                                     LASER
         TEST = ILLEGAL = .TRUE.                                          LASER
         ENCODE (10,305,LINE(L)) J, K                                     LASER
         STOP = .TRUE.                                                    LASER
   67 IF (KR(K).EQ.C) GO TO 68                                            LASER
      IF (IMAGE(2).NE.1H ) GO TO 68                                       LASER
         KR(K) = C                                                        LASER
         L = L+1                                                          LASER
         J = 1HR                                                          LASER
         NFLAG = 2H**                                                     LASER
         TEST = ILLEGAL = .TRUE.                                          LASER
         ENCODE (10,305,LINE(L)) J, K                                     LASER
         STOP = .TRUE.                                                    LASER
C                                                                         LASER
C     AT THIS POINT, WITH THE EXCEPTION OF CHANGES ENCOUNTERED AND PER-   LASER
C     PERMITTED BY INPUT, VECTORS KF AND KR CONTAIN ALL C-VALUES, AND     LASER
C     LINE(L) CONTAINS NAMES OF RATES INACCESSIBLE BY INPUT FOR THE PRO-  LASER
C     GRAM EXECUTION.                                                     LASER
C                                                                         LASER
   68 IF (KF(K).EQ.C) GO TO 63                                            LASER
      ENCODE (50,323,COMMENT)                                             LASER
         MODIFY = .TRUE.                                                  LASER
         IF (.NOT.TEST) NFLAG = 2H *                                      LASER
      ENCODE (10,301,RATE(1)) KF(K)                                       LASER
      IF (KF(K).EQ.0.) RATE(1) = 1H                                       LASER
      GO TO 64                                                            LASER
   63 KF(K) = FK                                                          LASER
   64 IF (KR(K).EQ.C) GO TO 69                                            LASER
      ENCODE (50,323,COMMENT)                                             LASER
         MODIFY = .TRUE.                                                  LASER
         IF (.NOT.TEST) NFLAG = 2H *                                      LASER
      ENCODE (10,301,RATE(2)) KR(K)                                       LASER
      IF (KR(K).EQ.0.) RATE(2) = 1H                                       LASER
      GO TO 10                                                            LASER
   69 KR(K) = RK                                                          LASER
   10 WRITE (MTAPE) (NTIME(I), I = 1,NTYPE), RATE, RNAME                  LASER
      IF (.NOT.OUT(10)) GO TO 71                                          LASER
      IF (LC.NE.0) GO TO 2                                                LASER
      IF (K.EQ.1) GO TO 83                                                LASER
         WRITE (6,102)                                                    LASER
         IF (MODIFY) WRITE (6,123)                                        LASER
         IF (ILLEGAL) WRITE (6,124)                                       LASER
   83 ILLEGAL = MODIFY = .FALSE.                                          LASER
      WRITE (6,PAGE)                                                      LASER
      WRITE (6,103) GENDATE                                               LASER
      IF (ELECT) WRITE (6,219)                                            LASER
      WRITE (6,105)                                                       LASER
    2 LC = LC+1                                                           LASER
      IF (LC.EQ.KOUNT) LC = 0                                             LASER
      WRITE (6,104) NFLAG, K, (RNAME(J), J = 1,5), RATE, COMMENT          LASER
      GO TO 71                                                            LASER
C                                                                         LASER
   70 IF (.NOT.OUT(10)) GO TO 93                                          LASER
      WRITE (6,102)                                                       LASER
      IF (MODIFY) WRITE (6,123)                                           LASER
      IF (ILLEGAL) WRITE (6,124)                                          LASER
```

```
   93 ERROR(7) = STOP
C
C     ---------------------------------------
C     BEGIN GENERATION OF ERROR DIAGNOSTICS --
C     ---------------------------------------
C
      LINES = 6
      WRITE (6,212)
C
      IF (.NOT.ERROR(1)) GO TO 514
         WORD = WARN
         IF (FATAL(1)) WORD = 6HFATAL:
         WRITE (6,201) WORD
         LINES = LINES+5
C
  514 IF (.NOT.ERROR(8)) GO TO 516
         WORD = WARN
         IF (FATAL(8)) WORD = 6HFATAL:
         WRITE (6,208) WORD, MAXGAS, MAXK, MAXNK
         LINES = LINES+5
C
  516 IF (.NOT.ERROR(9)) GO TO 525
         WORD = WARN
         IF (FATAL(9)) WORD = 6HFATAL:
         WRITE (6,209) WORD, NKMAX, KMAX, NMAX, NK, KTYPE, NTYPE
         LINES = LINES+5
C
  525 IF (.NOT.ERROR(7)) GO TO 75
         WORD = WARN
         IF (FATAL(7)) WORD = 6HFATAL:
         WRITE (6,207) WORD
         LINES = LINES + 5 + (L+7)/8
         WRITE (6,108) (LINE(I), I = 1,L)
C
C ---------------------------------------------------------------
C
   75 DO 5 I = 1,NTYPE
      MASS(I) = 0.
    5 E(I) = NO(I) = C
C
C     READ INPUT DATA: NAMES, CONCENTRATIONS, ENERGIES, MASSES --
C
      L = NGAS = 0
      PRESS = 0.
      IONIZE = 1.0E-12
   20 READ (5,101) TYPE, RATE, DUMMY, DUMMY, NPLOT
      IF (EOF(5)) 11,31
   31 BACKSPACE 5
      READ (5,112) P0, E0, MOLWT
      IF (TYPE.NE.3HRAD) GO TO 34
C     SPECIES *1* CORRESPONDS TO RADIATION --
      NO(1) = P0
      GO TO 20
   34 IF (TYPE.NE.4HE(-)) GO TO 39
C     SPECIES *2* CORRESPONDS TO ELECTRONS --
      NO(2) = P0
      PLOTS(2) = NPLOT.EQ.4HPLOT
```

```
      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
    8 CONTINUE
      L = L+1
      LINE(L) = TYPE
      ERROR(2) = .TRUE.
      GO TO 20
    9 IF (RATE(1).NE.1H ) NO(I) = P0*0.965E 19/TMOL
      IF (RATE(2).NE.1H ) E(I) = E0
      PLOTS(I) = NPLOT.EQ.4HPLOT
      GO TO 20
C
   11 IF (.NOT.ERROR(2)) GO TO 526
          WORD = WARN
          IF (FATAL(2)) WORD = 6HFATAL:
          WRITE (6,202) 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/TMOL
      IF (NO(2).LE.0.) NO(2) = IONIZE*NMOL
C
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
```

```
C     CHECK INITIALIZATION OF ENERGIES --
  527 L = 0
      E(1) = E(2) = 0.
      DO 15 I = 3,NTYPE
      IF (E(I).NE.0) GO TO 15
         E(I) = 0.
         ERROR(4) = .TRUE.
         L = L+1
         LINE(L) = GAS(I)
   15 CONTINUE
C
      IF (.NOT.ERROR(4)) GO TO 528
          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
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.
      IF (.NOT.REPEAT) GO TO 79
   95 IF (EMAX.GE.5.) GO TO 87
      EMAX = EMAX*2.
      IF (EMAX.GT.5.) EMAX = 5.
      GO TO 89
   87 EMAX = EMAX + 5.
   89 IF (EMAX.LE.ELIMIT) GO TO 79
         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,MESHP1
      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
      READ (NSCRTCH)
      READ (NSCRTCH)
      READ (NSCRTCH)
C
C  ------------------------------------------------------------------
C
C     IF THE SYNTHESIZED PROGRAM WAS CONSTRUCTED TO DEFAULT TO ELECTRON
C     KINETICS CALCULATIONS AS THE SOURCE FOR SECONDARY ELECTRON COLLI-
C     SION RATES, AND IF THE EXPERIMENTAL SITUATION CORRESPONDS TO AN
C     ELECTRIC DISCHARGE (EVCM ≠ 0), THE ELECTRON CROSS SECTION FILE IS
C     PROCESSED --
C
      CALL SECOND (T0)
      IN = NTYPE
      K = N = 0
      J = 1
      STOP = .FALSE.
   14 READ (NSCRTCH) LHS, RHS, LABEL, RATE, FK, RK, RNAME
      IF (EOF(NSCRTCH)) 60,505
  505 DECODE (1,100,RATE(1)) R1
      DECODE (1,100,RATE(2)) R2
      IF (R1.NE.1HV.AND.R2.NE.1HV) GO TO 14
      N = N+1
      ENCODE (40,101,PROCESS(1,N)) (RNAME(L), L = 1,4)
C
C     ---------------------------------------------------
C     PROCESS THE INELASTIC ELECTRON CROSS SECTION FILE --
C     ---------------------------------------------------
C
      CALL PLASMA (NDATA, MGRID+1, MESH, LHS, RHS, RNAME, EV, F, G,
     1 Q(1,N), U0, UM, IN, GAS, MISSING, REJECT, OUTSIDE, IDEG, OUT(8))
C
         TEST = MISSING.OR.REJECT.OR.OUTSIDE
         IF (.NOT.TEST) GO TO 32
         STOP = STOP.OR.TEST
         K = K+1
         IF (J.GT.231) GO TO 33
         ENCODE (50,322,LINE(J))
         J = J+5
         ENCODE (50,129,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 33
         ENCODE (100,128,LINE(J))
         J = J+10
         GO TO 33
C
   32 DO 24 L = 1,MESHP1
   24 Q(L,N) = EV(L)*Q(L,N)
   33 U(N) = 0.
      L1 = LEV1(N)
```

```
      IF (L1.EQ.0) GO TO 19
      U(N) = - E(L1)
      GO TO 13
   19 DO 21 L = 1,5
      I = LABEL(L,1)
      IF (I.EQ.0) GO TO 13
   21 U(N) = U(N)-E(I)
   13 L2 = LEV2(N)
      IF (L2.EQ.0) GO TO 23
      U(N) = U(N) + E(L2)
      GO TO 38
   23 DO 22 L = 1,5
      I = LABEL(L,2)
      IF (I.EQ.0) GO TO 38
   22 U(N) = U(N)+E(I)
   38 CONTINUE
      GO TO 14
C
   60 ERROR(5) = STOP
      FE = FATAL(5).AND.ERROR(5)
C
C     ----------------------------------------------
C     PROCESS THE MOMENTUM TRANSFER CROSS SECTIONS --
C     ----------------------------------------------
C
      IN = NTYPE
      TWOM = 2./1837.
      STOP = .FALSE.
      DO 17 I = 1,NGAS
      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 (NDATA, 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
      IF (MASS(I).LE.0.) ENCODE (50,119,LINE(J))
      IF (MASS(I).LE.0.) J = J+5
      IF (J.LE.225) GO TO 17
      ENCODE (100,128,LINE(J))
      J = J+10
      GO TO 17
C
C     ---------------------------------------------------------------
C     CONSTRUCT TWO MOMENTUM TRANSFER FUNCTIONS WHICH OCCUR IN THE
C     BOLTZMANN EQUATION --
C     ---------------------------------------------------------------
C
   36 DO 44 L = 1,MESHP1
      FQ = FRACT*QM(L)
      QMOM(L,1) = QMOM(L,1) + FQ
   44 QMOM(L,2) = QMOM(L,2) + FQ/MASS(I)
C
   17 STOP = STOP.OR.TEST
C
      ERROR(6) = STOP
      FE = FE.OR.(FATAL(6).AND.ERROR(6))
      ERRORS = ERROR(5).OR.ERROR(6)
C
C     ----------------------------------------------------
C     GENERATE ADDITIONAL WARNING DIAGNOSTICS FOR E- DATA
C     ----------------------------------------------------
C
      IF (.NOT.ERRORS) GO TO 98
         WORD = WARN
         IF (FE) WORD = 6HFATAL:
         J = J-1
         LINES = LINES + 5 + (J+4)/5
         IF (REPEAT.OR.OUT(8).OR.OUT(9).OR.(LINES.GT.55))
     1      WRITE (6,212)
         LINES = 6
         WRITE (6,205) WORD
         WRITE (6,206) (LINE(L), L = 1,J)
C
   98 FATAL(10) = .TRUE.
      DO 46 L = 1,MESHP1
      X = EV(L)
      XSQ = TWOM*X*X
      ERROR(10) = QMOM(L,1).LE.0.
      IF (ERROR(10)) GO TO 47
      A(L,1) = X/NMOL/QMOM(L,1)
   46 A(L,2) = XSQ*NMOL*QMOM(L,2)
C
      XBAR = DE/2.
      DO 41 I = 1,MESH
      CALL INTERP (2, XBAR, QMOM(I,1), EV, A(1,1), 1, MESHP1)
      CALL INTERP (2, XBAR, QMOM(I,2), EV, A(1,2), 1, MESHP1)
   41 XBAR = XBAR + DE
      QMOM(MESHP1,1) = QMOM(MESH,1)
      QMOM(MESHP1,2) = QMOM(MESH,2)
C
   47 WORD = WARN
```

```
      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,MESHP1
      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
      YO(1) = DY(1) = 0.
      CALL PLOT (1, MESH+1, 1, S, YO, 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
```

```
      ENCODE (40,322,KAPTION)
C
C   -----------------------------------------------
C   INTEGRATE EQUATIONS FROM T = 0 TO T = TPULSE --
C   -----------------------------------------------
C
   30 END = NP.EQ.LIMIT
      HOUT = TOUT - TT
      IF (H.LT.HOUT) GO TO 18
C
C     BEFORE INTEGRATING FROM T , (T + H), WHICH WILL PASS THE CYCLE
C     TIME, WE PAUSE TO GENERATE OUTPUT EXTRAPOLATED TO TIME NP*TOUT --
C
         DO 16 I = 1,9
         OUT(I) = .FALSE.
         IF (IO(I).EQ.0) GO TO 16
         OUT(I) = NP.EQ.IO(I)*(NP/IO(I))
   16    CONTINUE
         NP = NP+1
         TP = T + HOUT
         TIME(NP) = TP/UNIT
         IBEAM(NP) = JBEAM*SHAPE(TP)
         IF (LIMIT.NE.0) ENCODE (40,121,KAPTION) TP
C
C     CALCULATE EXTRAPOLATED VALUES OF THE POPULATION DENSITIES --
C
      SH = HOUT/H
      SJ = 1.
      JP1 = JSTART+1
      DO 25 I = 1,NP2
   25 POP(I) = 0.
      DO 65 J = 1,JP1
      DO 66 I = 1,NP2
   66 POP(I) = POP(I) + DNYDTN(I,J)*SJ
   65 SJ = SJ*SH
      NE = POP(2)
C
C   ------------------------------------------------------------------
C                         ELECTRON KINETICS ANALYSIS
C   ------------------------------------------------------------------
C
      IF (.NOT.ELECT) GO TO 48
C
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 (NEQ, TP, POP, NDOT)
      IF (DU.EQ.0.) S0 = S0 + SB
      IF (DU.EQ.0.) SB = 0.
C
      CHARGE = POP(NP1)
      CURRENT = -POP(NP2)
      RD = DIST/AREA/(EE*MU)
      IF (NE.GT.0.) RD = RD/NE
      VOLT = CURRENT*RD
      ITER = ITMAX
```

```
      IF (VOLT.EQ.0.) ITER = 0
      EVCM = ABS(VOLT/DIST)
      ESQ = EVCM*EVCM
      DNEDT = NDOT(2)
C
      NOUT(NP,1,NP1) = VC = CHARGE/CAPAC/1000.
      NOUT(NP,2,NP1) = KVOLT = VOLT/1000.
      NOUT(NP,1,NP2) = VR = CURRENT*RESIST/1000.
      NOUT(NP,2,NP2) = LDIDT = VC - VR - KVOLT
C
      CALL LEVELS (N1, N2, POP)
C
      IF (NP.EQ.1) GO TO 90
      DP = NE*ELASTIC + EE*(UBAR*DNEDT - NE*MU*ESQ - UPLUS*SB)
      DEPOSIT = IBEAM(NP)*DVDX
      PDISCH = NE*EE*MU*ESQ
      PCOLL = 0.
      DO 524 J = 1,NK
      IF (NEL(J).GE.0) GO TO 530
      PWR = N1(J)*POWER(J)
      GO TO 524
  530 PWR = EE*U(J)*(N1(J)*VSIG(1,J) - N2(J)*VSIG(2,J))
  524 PCOLL = PCOLL + PWR
      DP = DP + NE*PCOLL
      BEFORE = 100.*DP/(PDISCH + DEPOSIT)
C
   90 CALL SECOND (T0)
C
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, SO, EV, Q, QMOM, F, G, A, B, VSIG, POWER, PCOLL,
     3 PDISCH, DEPOSIT, DEDT, ELASTIC, DNEDT, DLNEDT, IONIZE, ATTACH,
     4 VD, MU, D, EK, AMPS, UBAR, TE, CONVRGE, PERCENT)
C
C     ----------------------------------------------------------------
C
      E(2) = UBAR
      CALL SECOND (T1)
      TBOLTZ = TBOLTZ + (T1 - T0)
      NBOLTZ = NBOLTZ + 1
C
      IF (ITER.LE.0) PERCENT = 0.
      REJECT = (.NOT.CONVRGE).OR.(PERCENT.GT.PCT)
      REPEAT = REJECT.AND.EXPAND
      IF (REPEAT) GO TO 95
      IF (REJECT) GO TO 99
C
C     OBTAIN ELECTRON PARAMETERS, NORMALIZED WITHOUT POPULATION DENSITY:
      IF (NE.NE.0.) ELASTIC = ELASTIC/NE
      DO 523 J = 1,NK
      IF (NEL(J).GE.0) GO TO 523
      IF (N1(J).NE.0.) POWER(J) = POWER(J)/N1(J)
  523 IF (NE.NE.0.) POWER(J) = POWER(J)/NE
C
C     ----------------------------------------------------------------
```

```
C
C     THE RATES PRODUCED BY SUBROUTINE RATES HERE ARE USED FOR OUTPUT
C     INFORMATION ONLY --
C
C     ------------------------------
   48 CALL DNDT (NEQ, TP, POP, NDOT)
C     ------------------------------
C
C     (NOTE: DO NOT INTERCHANGE POP AND NO.)
      RD = DIST/AREA/(EE*MU)
      IF (NE.GT.0.) RD = RD/NE
      COND(NP,1) = SIGMA = EE*MU*NE
      COND(NP,2) = RD
      JSUS = EE*NE*VD
      ISUS(NP,1) = SIGMA*EVCM
      ISUS(NP,2) = - NDOT(NP2)/AREA
C
      E(1) = HNU
C     CAVITY PHOTON DENSITY --
      PHOTON = POP(1)
      DNPHDT = NDOT(1)
C
C     EFFECTIVE PHOTON DENSITY IN MEDIUM --
      NDOT(1) = (CAVITY/LENGTH)*NDOT(1)
      POP(1) = (CAVITY/LENGTH)*POP(1)
C
      ETOT = DUDT = 0.
      DO 86 I = 1,NTYPE
      ITAU(I) = 1H
      TAU = 0.
      IF (NDOT(I).NE.0.) TAU = POP(I)/NDOT(I)
      TAU = ABS(TAU)
      TAU = TAU/UNIT
      IF (TAU.NE.0.) ENCODE (10,FFMT,ITAU(I)) TAU
      IF (TAU.GT.1.E 04) ENCODE (10,EFMT,ITAU(I)) TAU
      NOUT(NP,1,I) = POP(I)
      NOUT(NP,2,I) = NDOT(I)
      ETOT = ETOT + E(I)*POP(I)
   86 DUDT = DUDT + E(I)*NDOT(I)
      ETOT = EE*ETOT
      DUDT = EE*DUDT
C
      NDOT(1) = DNPHDT
      POP(1) = PHOTON
      RAD(NP,1) = RADIATE = 3.0E 10*HNU*PHOTON
      RAD(NP,2) = 3.0E 10*HNU*DNPHDT
      RAD(NP,3) = PBEAM = DVDX*IBEAM(NP)
      PSTIM = GAMMA*RADIATE
      HEAT = PDISCH + PBEAM - DUDT - PSTIM
      P = PBEAM
C
      ALPHA(NP,1) = GNET
      ALPHA(NP,2) = GAMMA
      ALPHA(NP,3) = GAIN
      ALPHA(NP,4) = ABSORB
C
      IF (.NOT.OUT(6)) GO TO 61
```

```
C
C  ---------------------------------------------------------------------
C      OUTPUT OF POPULATION DENSITIES AND THEIR RATES OF CHANGE, AND
C      MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS --
C  ---------------------------------------------------------------------
C
C      ENCODE OPTICAL AND ELECTRICAL PARAMETERS --
       L = 1
       ENCODE (120,307,LINE(L)) IBEAM(NP), FACTOR, ENERGY
       L = L+12
       IF (ENERGY.LE.0.) L = L-4
       DVDX = DVDX/1000.
       PBEAM = PBEAM/1000.
       ENCODE (80,325,LINE(L)) DVDX, PBEAM
       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
      L = L+12
      IF (ABS(EFF).GT.100.) L = L-4
C
   54 DEDT = DEDT/1000.
      DUDT = DUDT/1000.
      HEAT = HEAT/1000.
      ENCODE (120,316,LINE(L)) DEDT, DUDT, ETOT
      L = L+12
      ENCODE (40,327,LINE(L)) HEAT
      L = L+4
      IF (.NOT.ELECT) GO TO 529
      ENCODE (80,326,LINE(L)) BEFORE, AFTER
      L = L+8
  529 ENCODE (40,322,LINE(L))
      L = L+4
      ENCODE (120,320,LINE(L))
      LMAX = L+3
C
      L = 0
      WRITE (6,110) KAPTION, UNIT
      DO 35 I = 1,NTYPE
      WRITE (6,400) I, GAS(I), E(I), POP(I), NDOT(I), ITAU(I),
     1   (LINE(L+K), K = 1,4)
   35 IF (L.LT.LMAX) L = L+4
      LC = NTYPE+6
      WRITE (6,401) (LINE(L+K), K = 1,4)
      L = L+4
      STEP = H/UNIT
      WRITE (6,107) STEP, (LINE(L+K), K = 1,4)
      L = L+4
      WRITE (6,114) JSTART, (LINE(L+K), K = 1,4)
      L = L+4
      WRITE (6,401) (LINE(L+K), K = 1,4)
      L = L+4
      WRITE (6,111) (LINE(L+K), K = 1,4)
      L = L+4
   51 IF (L.GE.LMAX) GO TO 77
      WRITE (6,401) (LINE(L+K), K = 1,4)
      LC = LC+1
      L = L+4
      GO TO 51
   77 NSKIP = 43-LC
      IF (NSKIP.LT.1) NSKIP = 1
      ENCODE (80,500,IMAGE) NSKIP
      WRITE (6,IMAGE) TODAY
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)
C
      ARC = NE/NTOT.GT.RE
      END = END.OR.ARC
      IF (.NOT.END) GO TO 78
```

```
      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     ------------------------------------------------------------
C     CONTINUE TO INTEGRATE THE EQUATIONS FROM A DEAD START FROM THIS
C     CYCLE POINT --
C     ------------------------------------------------------------
C
   78 T = TP )
      DO 49 I = 1,NEQ
   49 DNYDTN(I,1) = POP(I)
      JSTART = NFLAG = 0
      DELTA = ETA
C
C     ------------------------------------------------------------
   18 CALL GEAR (NEQ, T, DNYDTN, SCRTCH, H, HMIN, HMAX, DELTA, METHOD,
     1 YMAX, ERR, KFLAG, JSTART, MAXDER, NMAXP2, PHI)
C     ------------------------------------------------------------
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 (NFLAG.LE.5) GO TO 18
      DELTA = 2.*DELTA
      IF (NFLAG.LE.10) GO TO 18
C
   97 IF (NP.LT.15) GO TO 99
C
C ------------------------------------------------------------------
C                         OUTPUT GENERATION
C ------------------------------------------------------------------
C
C     SUMMARY OF UNIMPORTANT REACTIONS --
C
      REWIND MTAPE
      KOUNT = 25
      LC = 0
      DO 507 K = 1,KTYPE
      READ (MTAPE) (NTIME(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
C     SUMMARY OF IMPORTANT REACTIONS --
C
      REWIND MTAPE
      LC = 0
      DO 501 K = 1,KTYPE
```

```
      READ (MTAPE) (NTIME(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 CONTINUE
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) = 5HDI/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) = 10HTHRESHHOLD
      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)
   82 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
   88 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
```

```
      RAD(I,NI) = RAD(I,3)                                                LASER
  512 ALPHA(I,3) = RAD(I,3)/1000.                                         LASER
   96 IF (NI.EQ.0) GO TO 508                                              LASER
      NPLOT = NI                                                          LASER
      WRITE (6,122)                                                       LASER
      CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,        LASER
     1 .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)                 LASER
      WRITE (6,306) UNIT, TODAY                                           LASER
C                                                                         LASER
      IF (.NOT.STIM) GO TO 509                                            LASER
C     COMPUTE POWER EFFICIENCY --                                         LASER
      DO 510 I = 1,NP                                                     LASER
      P = RAD(I,3)                                                        LASER
      IF (ELECT) P = P + RAD(I,2)                                         LASER
      IF (P.EQ.0.) P = 1.E 99                                             LASER
      ALPHA(I,2) = 100.*RAD(I,1)/P                                        LASER
  510 IF (ABS(ALPHA(I,2)).GT.100.) ALPHA(I,2) = 0.                        LASER
C                                                                         LASER
C     ----------------------------                                        LASER
C     INTEGRATE POWER DENSITIES --                                        LASER
C     ----------------------------                                        LASER
C                                                                         LASER
  509 DO 506 I = 1,3                                                      LASER
      B1 = RAD(1,I)                                                       LASER
      RAD(1,I) = 0.                                                       LASER
      DO 506 K = 2,NP                                                     LASER
      K1 = K-1                                                            LASER
      B2 = B1                                                             LASER
      B1 = RAD(K,I)                                                       LASER
  506 RAD(K,I) = RAD(K1,I) + 0.5E 03*TOUT*(B1 + B2)                       LASER
C                                                                         LASER
      WRITE (6,125)                                                       LASER
      CALL PLOT (M1, NP, 1, RAD, 0., 0., TIME, 0., DTIME, .FALSE.,        LASER
     1   .TRUE., .TRUE., .TRUE., .TRUE., KAPTION, NPLOT, 0)               LASER
      WRITE (6,306) UNIT, TODAY                                           LASER
C                                                                         LASER
C     ------------------------------------------------------------------- LASER
C     TABULAR SUMMARY OF MISCELLANEOUS ELECTRICAL AND OPTICAL PARAMETERS  LASER
C     ------------------------------------------------------------------- LASER
C                                                                         LASER
      I2 = 1                                                              LASER
  522 I1 = I2                                                             LASER
      I2 = I2+50                                                          LASER
      IF (I2.GT.NP) I2 = NP                                               LASER
      WRITE (6,223) UNIT                                                  LASER
      WRITE (6,226) (TIME(I), IBEAM(I), ALPHA(I,3), NOUT(I,2,NP1),        LASER
     1   ISUS(I,1), ALPHA(I,4), ALPHA(I,1), ALPHA(I,2), I = I1,I2)        LASER
      IF (I2.LT.NP) GO TO 522                                             LASER
C                                                                         LASER
      IF (.NOT.STIM) GO TO 508                                            LASER
      DO 513 I = 1,NP                                                     LASER
      ALPHA(I,3) = RAD(I,1)                                               LASER
      P = RAD(I,3)                                                        LASER
      IF (ELECT) P = P + RAD(I,2)                                         LASER
      IF (P.EQ.0.) P = 1.E 99                                             LASER
  513 ALPHA(I,4) = 100.*RAD(I,1)/P                                        LASER
C                                                                         LASER
```

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

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

```
C
  118 FORMAT (10X*CROSS SECTION DOES NOT SPAN [0,*F4.1*] EV.*)
C
  119 FORMAT (10X*THE MASS ENTERED FOR THIS SPECIES IS ≤ 0*)
C
  120 FORMAT (A10* + E → E + *A10)
C
  121 FORMAT (8X*TIME T = *,1PE10.3* SEC*)
C
  122 FORMAT (1H1,56X*POWER DENSITY (WATT/CM3)*//)
C
  123 FORMAT (45X≠* THE ORIGINAL RATE CONSTANT(S) HAVE BEEN MODIFIED≠)
C
  124 FORMAT (44X≠** ILLEGAL ATTEMPT TO MODIFY RATE CONSTANT(S) WAS REJ
     1ECTED≠)
C
  125 FORMAT (1H1,56X*ENERGY DENSITY (JOULE/LITER)*//)
C
  126 FORMAT (1H1,42X*LASER GAIN AND MEDIUM ABSORPTION COEFFICIENTS (CM-
     11)*//)
C
  127 FORMAT (40X*THE PROCESSING OF ELECTRON CROSS SECTIONS REQUIRED*,
     1F5.1* CP SEC,*/40X*AND*I3* BOLTZMANN ELECTRON CALCULATIONS CONSUME
     2D*,F6.1* CP SEC.*)
C
  128 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)
C
  129 FORMAT (I2*)  *4A10,5X)
C
  130 FORMAT (1H1,24X,*NORMALIZED EXTERNAL SOURCE FUNCTION FOR CREATION
     1OF ELECTRONS IN THE ENERGY RANGE (U, U + DU)*/43X*AVERAGE ENERGY O
     2F CREATED ELECTRONS = < U+ > = *F6.2* EV*/)
C
  131 FORMAT (/62X*ELECTRON ENERGY U (EV)*)
C
  201 FORMAT (//15X,A10*ERRORS WERE DETECTED IN PROCESSING THE INPUT REA
     1TION 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.*)
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:*/)
C
  203 FORMAT (//15X,A10*INITIAL POPULATION DENSITIES FOR THE FOLLOWING S
     1PECIES WERE NOT DEFINED.  IF THE PROGRAM IS EXE-*/25X*CUTED, NO(I)
     2 = 0 WILL BE ASSUMED:*/)
C
  204 FORMAT (//15X,A10*ENERGIES FOR THE FOLLOWING SPECIES WERE NOT DEFI
     1NED.  IF PROGRAM IS EXECUTED, E(I) = 0 WILL BE*/25X*ASSUMED:*/)
C
  205 FORMAT (//15X,A10*ERRORS OCCURRED FOR THE INPUT CROSS SECTIONS FOR
     1 THE FOLLOWING ELECTRON COLLISION PROCESSES.*/25X*IF THE PROGRAM I
     2S EXECUTED, A ZERO CROSS SECTION OVER ALL ENERGY WILL BE ASSUMED:*
```

```
     3/)
C
  206 FORMAT ((40X,5A10))
C
  207 FORMAT (//15X,A10*AN ATTEMPT WAS MADE TO ENTER, BY $RATES ... $ IN
     1PUT, THE FOLLOWING RATES, WHICH ARE NOT ACCES-*/25X*SIBLE BY INPUT
     2 FOR THE SYNTHETIC PROGRAM THAT WAS GENERATED.  IF THE PROGRAM EXE
     3CUTES, THE AT-*/25X*TEMPTED MODIFICATIONS WILL BE IGNORED, AND THE
     4 ORIGINAL RATES USED:*/)
C
  208 FORMAT (//15X,A10*SYNTHESIZED SUBROUTINES AND DATA FILE WERE GENER
     1ATED WITH DIMENSION DECLARATORS NMAX =*I4*,*/25X*KMAX =*I4*, AND N
     2KMAX =*I4*.  THESE MUST AGREE WITH THE CORRESPONDING DIMENSION DEC
     3LARATORS */25X*IN THE MAIN PROGRAM (LASER).  EXECUTION MAY BE POSS
     4IBLE IF PRESENT STORAGE EXCEEDS ORIGINAL, BUT*/25X*CAUTION IS ADVI
     5SED TO INSURE THAT LABELED COMMON BLOCKS AGREE WITH THOSE IN THE
     6 SYNTHETIC*/25X*SUBROUTINES ASSOCIATED WITH THE DATA FILE GENERATE
     7D ON TAPE4.*)
C
  209 FORMAT (//15X,A10*TAPE4 DATA FILE VECTORS EXCEEDED DIMENSION STORA
     1GE IN MAIN PROGRAM, AND WERE TRUNCATED DURING*/25X*READ.  NKMAX =*
     2I4*, KMAX =*I4*, AND NMAX =*I4*.  TAPE4 CONTAINS NK =*I4*, KTYPE =
     3*I4*, AND*/25X*NTYPE =*I4*.  CAUTION IS ADVISED.*)
C
  210 FORMAT (//15X,A10*MOMENTUM TRANSFER COLLISION FREQUENCY IS ZERO AT
     1 SOME POINT.  E- ANALYSIS CONTAINS 1/QM TERMS.*)
C
  211 FORMAT (2HN(,I2,1H),5X,3HDN(,I2,4H)/DT,1X)
C
  212 FORMAT (1H1/55X*SUMMARY OF POSSIBLE ERROR CONDITIONS*////15X*SEVER
     1ITY*45X*DESCRIPTION */15X,105(1H-))
C
  214 FORMAT (1H1,25X*DISCHARGE CURRENT DENSITY JSUS (AMP/CM2), AND ITS
     1RATE OF CHANGE D/DT(JSUS) (A/CM2/S)*//)
C
  215 FORMAT (1H1,20X≠PLASMA CONDUCTIVITY SIGMA = E*NE*MU (/CM/OHM), AND
     1 DISCHARGE IMPEDANCE RD = D/AREA/SIGMA (OHM)≠//)
C
  216 FORMAT (1H1/40X*E-BEAM CURRENT DENSITY (AMP/CM2) AS A FUNCTION OF
     1TIME*//)
C
  217 FORMAT (1H1/56X*CIRCUIT VOLTAGES (KVOLT)*//)
C
  218 FORMAT (1H1/27X*INTRACAVITY RADIATION INTENSITY I (W/CM2) AND ITS
     1RATE OF CHANGE DI/DT (W/CM2/SEC)*//)
C
  219 FORMAT (/13X*(VSIG(K,I) ARE FORWARD (K=1) OR REVERSE (K=2) RATES F
     1OR THE ITH INELASTIC PROCESS IN THE E- KINETICS ANALYSIS)*)
C
  220 FORMAT (1H1,20(/),40X*ELECTRON ENERGY RANGE EMAX =*,0PF6.2* EXCEED
     1S MAXIMUM VALUE =*0PF6.2)
C
  221 FORMAT (1H1,40X*OPTICALLY EXTRACTED POWER DENSITY (KW/CM3) AND EFF
     1ICIENCY (%)*//)
C
  222 FORMAT (1H1,35X*OPTICALLY EXTRACTED ENERGY DENSITY (JOULE/LITER) A
     1ND EFFICIENCY (%)*//)
```

```
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-)/)
C
  224 FORMAT (40X*ELECTRON DENSITY NE =*1PE10.3* EXCEEDS LIMIT NE/NMOL ≤
     1 *1PE10.3)
C
  225 FORMAT (/40X*CALCULATION IS TERMINATED AT CYCLE NP =*I3*, T = *1PE
     110.3,* SEC.*)
C
  226 FORMAT (18X,F9.2,5F12.2,F13.2,F11.2)
C
  227 FORMAT (1H1/32X,*POPULATION DENSITY N (CM-3) AND RATE OF CHANGE DN
     1/DT (CM-3/S) FOR *,A10/)
C
  300 FORMAT (///25X*PROGRAM IS TERMINATED FOR ERRORS SPECIFIED FATAL.*)
C
  301 FORMAT (1PE10.4)
C
  302 FORMAT (///25X,86(1H*)///)
C
  303 FORMAT (28X,8A10)
C
  304 FORMAT (1H1,5(/))
C
  305 FORMAT (*K*,A1,*(*,I3,*)*)
C
  306 FORMAT (/62X*TIME (*1PE9.3* SEC)*/100X*DR. WILLIAM B. LACINA,*A11/
     1100X*NORTHROP RESEARCH AND TECHNOLOGY*)
C
  307 FORMAT (3X*JBEAM*5X* = *0PF10.2*  AMP/CM2*,8X*DEPOSITION = *F10.2,
     117X*ENERGY*4X* = *F10.0*  KEV*,9X)
C
  308 FORMAT (3X*S(U > 0)    = *1PE10.3*  CM-3/SEC*7X*S(U = 0)    = *
     1 1PE10.3*  CM-3/SEC*4X)
C
  309 FORMAT (3X*R(DISCH)*2X* = *1PE10.3*  OHM*)
C
  310 FORMAT (43X*Q*9X* = *,1PE10.3*  COULOMB*8X*I(DISCH)*2X* = *,1PE10.
     13*  AMP*9X)
C
  311 FORMAT (3X*J(DISCH)*2X* = *1PE10.3*  AMP/CM2*8X*ESUS*6X* = *0PF10.
     13*  KVOLT/CM*7X≠ESUS*JSUS  = ≠0PF10.2*  KW/CM3*)
C
  312 FORMAT (43X*Q/C*7X* = *0PF10.3*  KVOLT*10X*V(DISCH)*2X* = *0PF10.3
     1*  KVOLT*)
C
  313 FORMAT (3X*V(RESIST)   = *0PF10.3*  KVOLT*)
C
  314 FORMAT (3X≠L*DI/DT≠3X* = *0PF10.3*  KVOLT*)
C
  315 FORMAT (3X*INTENSITY   = *1PE10.3*  WATT/CM2*,7X*OPTICAL*3X* = *,
     10PF10.2*  KW/CM3*9X*EFFICIENCY = *,0PF10.2*  %*)
C
  316 FORMAT (3X≠E<U>DNE/DT = ≠1PE10.3*  KW/CM3*9X*DE(TOT)/DT = *E10.3,
```

```
     1*  KW/CM3*9X*E(TOT)*4X* = *E10.3*  J/CM3*7X)                    LASER
C                                                                     LASER
  317 FORMAT (3X*AREA*6X* = *0PF10.2*  CM2*12X*DIST*6X* = *0PF10.2*  CM* LASER
     113X*CONDUCT*3X* = *1PE10.3*  /OHM/CM*)                          LASER
C                                                                     LASER
  318 FORMAT (43X*REFLECT*3X* = *0PF10.2*  %*14X*LOSS*6X* = *0PF10.2*  % LASER
     1/PASS*)                                                         LASER
C                                                                     LASER
  319 FORMAT (3X*LENGTH*4X* = *0PF10.2*  CM*13X*OMEGA/4/PI = *1PE10.3,   LASER
     117X*THRESH*4X* = *1PE10.3*  CM-1*)                              LASER
C                                                                     LASER
  320 FORMAT (40(1H-),80X)                                            LASER
C                                                                     LASER
  321 FORMAT (3X*NET GAIN    = *1PE10.3,*  CM-1*11X*LASER GAIN = *1PE10.3 LASER
     1*  CM-1*11X*ABSORPTION = *,1PE10.3*  CM-1*)                     LASER
C                                                                     LASER
  322 FORMAT (10X)                                                    LASER
C                                                                     LASER
  323 FORMAT (≠***** THE ORIGINAL RATE HAS BEEN MODIFIED *****≠)      LASER
C                                                                     LASER
  324 FORMAT (3X*T(CAVITY)   = *F10.1*  NS*13X*L(CAVITY)   = *F10.2*  CM*) LASER
C                                                                     LASER
  325 FORMAT (3X*DVDX*6X* = *F10.2*  KV/CM*10X*P(BEAM)*3X* = *F10.2,    LASER
     1 *  KW/CM3*6X)                                                  LASER
C                                                                     LASER
  326 FORMAT (3X*DP(BEFORE) = *F10.2*  %*14X*DP(AFTER)  =*F11.2*  %*11X) LASER
C                                                                     LASER
  327 FORMAT (3X*HEAT + SP   = *1PE10.3*  KW/CM3*6X)                  LASER
C                                                                     LASER
  400 FORMAT (9X,I5,5X,A10,0PF7.2,1P2E14.3,2X,A10,6X,4A10)            LASER
C                                                                     LASER
  401 FORMAT (82X,4A10)                                               LASER
C                                                                     LASER
  402 FORMAT (1H1,36X*THE FOLLOWING REACTIONS CONTRIBUTE LESS THAN *F3.0 LASER
     1* %*//29X*K*8X*KF(K)*10X*KR(K)*15X*REACTION(K)*/27X,75(1H-)/)   LASER
C                                                                     LASER
  403 FORMAT (25X,I5,5X,A10,5X,A10,5X,4A10/)                          LASER
C                                                                     LASER
  404 FORMAT (1H1,40X*THE FOLLOWING REACTIONS WERE IMPORTANT (> *F3.0,  LASER
     1* %)*//29X*K*8X*KF(K)*10X*KR(K)*15X*REACTION(K)*/27X,75(1H-)/)  LASER
C                                                                     LASER
  500 FORMAT (1H(,I2,≠(/)90X*DR. WILLIAM B. LACINA,*A11/90X*NORTHROP RES LASER
     1EARCH AND TECHNOLOGY*)≠)                                        LASER
C                                                                     LASER
C-------------------------------------------------------------------- LASER
C                                                                     LASER
   99 CALL EXIT                                                       LASER
      END                                                             LASER
```

```
      SUBROUTINE SYNTH (LTAPE, MTAPE, NTAPE, NSCRTCH, NDATA, NSIZE,         SYNTH
     1   MAXGAS, GAS, KMAX, NKMAX, LEV1, LEV2, DATE)                        SYNTH
C                                                                           SYNTH
C .......................................................................   SYNTH
C                                                                           SYNTH
C     THIS SUBROUTINE WILL EDIT THE INPUT FILE OF KINETIC REACTIONS AND     SYNTH
C     AUTOMATICALLY GENERATE SUBROUTINES REQUIRED FOR THE MOLECULAR KI-     SYNTH
C     NETICS ANALYSIS AND ITS LINKAGE TO COUPLED ELECTRON KINETICS CAL-     SYNTH
C     CULATIONS.  SYMBOLIC REACTIONS ARE TRANSLATED INTO COMPUTER CODED     SYNTH
C     EQUATIONS. SUBROUTINES SYNTHESIZED ARE: (1) RATES OF CHANGE DNI/DT    SYNTH
C     OF POPULATION DENSITIES, (2) THE JACOBIAN D(DNI/DT)/D(NJ), AND (3)    SYNTH
C     DEFINITION OF THE CORRESPONDENCE OF MOLECULAR STATES WHICH OCCUR      SYNTH
C     IN THE ELECTRON-MOLECULE SCATTERING PROCESSES FOR THE E- KINETICS.    SYNTH
C     INPUT CONSISTS OF AN ARBITRARILY LONG SEQUENCE OF CARD PAIRS OF       SYNTH
C     THE FORM                                                              SYNTH
C                                                                           SYNTH
C         1)  A1 + A2 + A3 + ...  =  B1 + B2 + B3 + ...                     SYNTH
C         2)  KF, KR, KOMMENT  (2E10.3,5X,5A10)                             SYNTH
C                                                                           SYNTH
C     WHERE A1, A2, ..., B1, B2, ... ARE THE PHYSICAL NAMES OF THE RE-      SYNTH
C     ACTANT SPECIES (FOR ANY SPECIES NAME, THE FIRST *NSIZE* CHARACTERS    SYNTH
C     ARE RECOGNIZED, AND OTHERS IGNORED). EACH SIDE OF THE REACTION MAY    SYNTH
C     CONTAIN UP TO *MAX* SPECIES.  THERE ARE NO RESTRICTIONS ON THE        SYNTH
C     REACTION FORMAT (WHICH MAY INCLUDE EMBEDDED BLANKS) EXCEPT FOR        SYNTH
C     THE FOLLOWING --                                                      SYNTH
C                                                                           SYNTH
C         1)  ELECTRONS ARE DENOTED BY EITHER E, E-, OR E(-).               SYNTH
C                                                                           SYNTH
C         2)  HIGH ENERGY ELECTRONS (E-BEAM) ARE DENOTED BY HE-.            SYNTH
C                                                                           SYNTH
C         3)  BUFFER GASES ARE DENOTED BY M.                                SYNTH
C                                                                           SYNTH
C         4)  NOISE PHOTONS ARE DENOTED BY HNU.  IF HNU APPEARS ON THE      SYNTH
C             LEFT HAND SIDE, IT IS REJECTED.  ALL PROCESSES WHICH CON-     SYNTH
C             TRIBUTE NOISE TO THE BUILDUP OF STIMULATED EMISSION ARE       SYNTH
C             TO BE DESCRIBED WITH HNU ON THE RIGHT HAND SIDE OF THE        SYNTH
C             REACTION.                                                     SYNTH
C                                                                           SYNTH
C         5)  STIMULATED EMISSION OR ABSORPTION PROCESSES ARE DESCRIBED     SYNTH
C             IN TERMS OF PHOTON NUMBER DENSITY, AND ARE RECOGNIZED BY      SYNTH
C             THE APPEARANCE OF RAD.                                        SYNTH
C                                                                           SYNTH
C         6)  THERE ARE NO SPECIAL RESTRICTIONS ON SPECIES NAMES EXCEPT     SYNTH
C             THAT IONS MUST EXPLICITLY EXHIBIT THEIR CHARGE IN THEIR       SYNTH
C             NAME (E.G. F-, KR2(+), AR(++), CO-, ETC.).  IF AN ION         SYNTH
C             IS POSITIVE, THE + SYMBOL MUST BE IMMEDIATELY FOLLOWED BY     SYNTH
C             EITHER ANOTHER + OR ), IN ORDER TO AVOID CONFUSION WITH       SYNTH
C             THE NORMAL USAGE OF + IN WRITING THE REACTION AS ABOVE.       SYNTH
C                                                                           SYNTH
C     IF KF = 0 (KR = 0), NO TRANSLATION OF THE FORWARD (REVERSE) RE-       SYNTH
C     ACTION TERM OCCURS (THUS, NULL OPERATIONS CONTAINING UNNECESSARY      SYNTH
C     MULTIPLICATIONS BY ZERO) ARE ELIMINATED.  RATE CONSTANTS USED FOR     SYNTH
C     SYNTHESIS CAN BE CHANGED IN SUBSEQUENT EXECUTION, HOWEVER, SO A       SYNTH
C     NONZERO VALUE SHOULD BE USED DURING SYNTHESIS FOR ANY REACTION        SYNTH
C     PROCESS WHICH IS NOT TO BE PERMANENTLY NEGLECTED.                     SYNTH
C                                                                           SYNTH
C     REACTIONS ARE SUBJECTED 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:
C
C           1)  HIGH ENERGY ELECTRONS MUST BALANCE ON LHS AND RHS.
C
C           2)  BUFFER GAS M MUST BALANCE ON BOTH SIDES OF EQUATION.
C
C           3)  CHARGE CONSERVATION MUST NOT BE VIOLATED.
C
C           4)  NO REVERSE PROCESS FOR SPONTANEOUS RADIATION ALLOWED.
C
C           5)  DUPLICATE REACTIONS (EVEN WRITTEN BACKWARDS) ARE IGNORED
C
C           6)  DETAIL BALANCE FOR BINARY MOLECULAR COLLISIONS ENFORCED.
C
C           7)  SECONDARY E- COLLISION PROCESSES MAY HAVE FIXED RATE CON-
C               STANTS, OR ZERO MAY BE ENTERED, IN WHICH CASE THEY WILL BE
C               AUTOMATICALLY LINKED BY DEFAULT TO AN E- KINETICS ANALY-
C               SIS.  DURING SYNTHESIS, THE ELECTRON CROSS SECTION FILE
C               WILL BE SCANNED TO DETERMINE WHETHER DATA FOR THE PROCESS
C               IS AVAILABLE, AND IF NOT, AN INFORMATIVE DIAGNOSTIC ISSUED
C
C           8)  STIMULATED EMISSION (AND ABSORPTION) ARE DENOTED BY THE
C               SYNTAX: B + RAD = A + RAD. (IT IS ASSUMED THAT THE FOR-
C               WARD REACTION DENOTES EMISSION.)  ABSORPTION PROCESSES CAN
C               BE ENTERED AS: X + RAD = Y.  (ONLY THE FORWARD REACTION IS
C               RECOGNIZED, AND CORRESPONDS TO ABSORPTION.)
C
C     THERE ARE A VARIETY OF ERRORS RELATED TO EXCEEDED LIMITS, BAD
C     SYNTAX, BAD PHYSICS, ETC. WHICH ARE RECOGNIZED AND FLAGGED.  BOTH
C     FATAL AND NON-FATAL WARNING CONDITIONS ARE GENERATED DURING SYNTH-
C     SIS, AND ARE PROVIDED IN AN EDITED OUTPUT SUMMARY OF THE REACTION
C     SCHEME WHICH WAS PROCESSED.  A CROSS-REFERENCE LISTING OF THE
C     OCCURRENCE OF SPECIES IN THE REACTION SCHEME IS ALSO GENERATED.
C     THE SYNTHESIZED SUBROUTINES ARE DOCUMENTED WITH COMMENT CARDS.
C
C ...............................................................
C
      DIMENSION IMAGE(80), KAR(10), REFER(40), LINE(100), NBUFF(2),
     1 NHE(2), GAS(1), KINETIC(10), LABEL(5,2), LL(160), FORM(5),
     2 LEV1(1), LEV2(1), VSIG(2), TITLE(3), NAME(100), COMM(5), KODE(8),
     3 LDUM(10)
C
      REAL KF, KR, KB
C
      INTEGER GAS, SIGN, RATE(2), E, HNU, RAD, LHS, RHS, HE, TITLE, DATE
C
      LOGICAL ELECT(2), RADIATE, REJECT, REVERSE, FORWARD, DETAIL, EXIT,
     1   SOURCE, BUFFER, PHOTO(2), OPTICAL, LASER, TEST
C
      DATA E, HE, HNU, RAD, F, R, SKIP / 4HE(-), 5HHE(-), 3HHNU, 3HRAD,
     1     7HFORWARD, 7HREVERSE, 5H(1H1) /
C
      DATA KB, E0, H, C / 1.38 E-23, 1.602 E-19, 6.625 E-34, 3.0 E 10 /
C
C     NTAPE   =  FILE FOR GENERATION OF SUBROUTINES RATES, LEVELS
C     MTAPE   =  FILE FOR GENERATION OF SUBROUTINE JACOB
```

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

```
      WRITE (NTAPE,422)
      WRITE (NTAPE,406)
      WRITE (NTAPE,105)
      ENCODE (80,208,KODE)
      WRITE (NTAPE,102) KODE
      WRITE (NTAPE,105)
C
      WRITE (MTAPE,403)
      WRITE (MTAPE,507)
      WRITE (MTAPE,105)
      WRITE (MTAPE,422)
      WRITE (MTAPE,406)
      WRITE (MTAPE,105)
      WRITE (MTAPE,102) KODE
      WRITE (MTAPE,105)
C
C     FORWARD DEFAULT FOR SECONDARY ELECTRON COLLISIONS --
      M1 = 1
      M2 = 2
      INTEGER = 4H(I1)
      INITIAL = 7HR = KF(
      NTOT = 5H*NTOT
      MULT = 4H*NO(
      LAST = 6H - KR(
      IF (NSIZE.GT.10) NSIZE = 10
      MAX = 5
      M0 = 5
      M10 = 10*M0
      M20 = 2*M10
      LC = 0
      NK = 0
      INPUT = 0
      K = 1
    1 K = K-1
C     READ HOLERITH STATEMENT OF REACTION NUMBER K --
   10 K = K+1
      READ (5,100) IMAGE
      IF (EOF(5)) 5,76
   76 READ (5,101) VSIG, COMM
      DO 61 L = 1,M0
   61 REFER(L) = COMM(L)
      INPUT = INPUT+1
      EXIT = EXIT.OR.REJECT
      REWIND LTAPE
      ENCODE (10,209,NUMBER) K
      REJECT = DETAIL = .FALSE.
      RATE(1) = RATE(2) = 1H
      MESS = M0 + 1
      IF (REFER(1).EQ.1H ) MESS = 1
      BUFFER = SOURCE = RADIATE = ELECT(1) = ELECT(2) = .FALSE.
      PHOTO(1) = PHOTO(2) = .FALSE.
      DO 9 L = 1,MAX
      DO 9 M = 1,2
      NBUFF(M) = NHE(M) = 0
    9 LABEL(L,M) = 0
      DO 22 L = 1,100
   22 LINE(L) = 1H
```

```
      DO 21 L = 1,160
   21 LL(L) = 1H
      NKAR = 6
      DECODE (7,100,INITIAL) (LL(NKAR+L), L = 1,7)
      NKAR = NKAR+7
      K2 = K/10
      K3 = K - 10*K2
      K1 = K2/10
      K2 = K2 - 10*K1
      IF (K1.EQ.0) GO TO 11
      NKAR = NKAR+1
      ENCODE (10,INTEGER,LL(NKAR)) K1
   11 IF (K1+K2.EQ.0) GO TO 29
      NKAR = NKAR+1
      ENCODE (10,INTEGER,LL(NKAR)) K2
   29 NKAR = NKAR+1
      ENCODE (10,INTEGER,LL(NKAR)) K3
      NKAR = NKAR+1
      LL(NKAR) = 1H)
      M = 1
      NO = NTYPE
      NE = NCH = 0
      NN = 1
      LP = I = J = N = 0
C     SCAN THE 80 BCD CHARACTERS TO DETERMINE SPECIES --
    2 IF (I.EQ.80) GO TO 4
      I = I+1
C     IMBEDDED BLANKS ARE IGNORED.
      IF (IMAGE(I).EQ.1H ) GO TO 2
      IF (IMAGE(I).NE.1H*) GO TO 3
         MM = 1
         GO TO 4
C
    3 IF (IMAGE(I).NE.1H+) GO TO 6
      IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H+) GO TO 6
         MM = 0
         GO TO 4
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
      J = J+1
      LP = LP+1
      LINE(LP) = KAR(J) = IMAGE(I)
      IF (KAR(J).EQ.1H+) NCH = NCH + NN
      IF (KAR(J).EQ.1H-) NCH = NCH - NN
      GO TO 2
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
      IF (M.GT.2) GO TO 60
      IF (I.EQ.80.AND.M.NE.2) GO TO 60
      IF (N.LT.MAX) GO TO 38
         ENCODE (M10,214,REFER(MESS)) MAX
```

```
      MESS = MESS + M0
   60 REJECT = .TRUE.
      ENCODE (M10,216,REFER(MESS))
      MESS = MESS+M0
      LB = 0
      DO 53 L = 1,80
      IF (IMAGE(L).EQ.1H ) GO TO 53
      LB = LB+1
      IMAGE(LB) = IMAGE(L)
   53 CONTINUE
      LB = LB+1
      IF (LB.GT.80) LB = 80
      DO 55 L = LB,80
   55 IMAGE(L) = 1H
      NUMBER = 1H
C     ENCODE UNRECOGNIZABLE REACTION --
      ENCODE (100,100,KINETIC) (IMAGE(L), L = 1,80)
      GO TO 26
   38 ENCODE (10,100,NGAS) (KAR(L), L = 1,J)
      IF (NGAS.EQ.1HE) NCH = NCH - NN
      IF (NGAS.EQ.1HE.OR.NGAS.EQ.2HE-) NGAS = E
      IF (NGAS.EQ.3HHE-) NGAS = HE
      IF (NGAS.EQ.HE) NCH = NCH + NN
      J = 0
      IF (NGAS.EQ.1HM) GO TO 8
      IF (NGAS.EQ.HE) GO TO 8
      IF (NGAS.NE.HNU) GO TO 30
      IF (M.EQ.1) GO TO 67
      IF (RADIATE) GO TO 67
      IF (PHOTO(1).OR.PHOTO(2)) GO TO 67
      RADIATE = .TRUE.
      VSIG(2) = 0.
      GO TO 16
   30 IF (NGAS.NE.RAD) GO TO 45
      IF (PHOTO(M)) GO TO 67
      PHOTO(M) = .TRUE.
      IF (RADIATE.AND.(PHOTO(1).OR.PHOTO(2))) GO TO 67
      GO TO 45
   67 REJECT = .TRUE.
      ENCODE (M10,227,REFER(MESS))
      MESS = MESS + M0
      GO TO 16
   45 N = N+1
      IF (NGAS.EQ.E) NE = NE+1
      DO 15 L = 1,N0
      IF (GAS(L).NE.NGAS) GO TO 15
      LABEL(N,M) = L
      GO TO 8
   15 CONTINUE
      IF (N0.NE.MAXGAS) GO TO 39
      IF (REJECT) GO TO 16
      REJECT = .TRUE.
      ENCODE (M20,213,REFER(MESS)) MAXGAS
      MESS = MESS + 2*M0
      GO TO 16
   39 N0 = N0 + 1
      GAS(N0) = NGAS
```

```
      LABEL(N,M) = NO                                                   SYNTH
    8 NEW = NKAR - 80*(NKAR/80) + 7                                     SYNTH
      IF (NEW.LE.72) GO TO 17                                           SYNTH
      NEW = NEW+8                                                       SYNTH
      NKAR = 80*(NEW/80) + 6                                            SYNTH
      LL(NKAR) = 1H$                                                    SYNTH
   17 DECODE (4,100,MULT) (LL(NKAR+L), L = 1,4)                         SYNTH
      IF (NGAS.NE.1HM) GO TO 70                                         SYNTH
      NBUFF(M) = NBUFF(M) + 1                                           SYNTH
      BUFFER = .TRUE.                                                   SYNTH
      DECODE (5,100,NTOT) (LL(NKAR+L), L = 1,5)                         SYNTH
      NKAR = NKAR+5                                                     SYNTH
      GO TO 16                                                          SYNTH
   70 IF (NGAS.NE.HE) GO TO 20                                          SYNTH
         NHE(M) = NHE(M) + 1                                            SYNTH
         SOURCE = .TRUE.                                                SYNTH
         GO TO 16                                                       SYNTH
   20 NKAR = NKAR+4                                                     SYNTH
      N1 = LABEL(N,M)/10                                                SYNTH
      N2 = LABEL(N,M) - 10*N1                                           SYNTH
      IF (N1.EQ.0) GO TO 13                                             SYNTH
      NKAR = NKAR+1                                                     SYNTH
      ENCODE (10,INTEGER,LL(NKAR)) N1                                   SYNTH
   13 NKAR = NKAR+1                                                     SYNTH
      ENCODE (10,INTEGER,LL(NKAR)) N2                                   SYNTH
      NKAR = NKAR+1                                                     SYNTH
      LL(NKAR) = 1H)                                                    SYNTH
   16 IF (I.EQ.80) GO TO 27                                             SYNTH
C                                                                       SYNTH
C     DELIMITERS (+, →) ENCODED INTO HOLERITH LINE TO DEFINE REACTION -- SYNTH
C                                                                       SYNTH
      LP = LP+1                                                         SYNTH
      LINE(LP) = 1H                                                     SYNTH
      LP = LP+1                                                         SYNTH
      LINE(LP) = IMAGE(I)                                               SYNTH
      LP = LP+1                                                         SYNTH
      LINE(LP) = 1H                                                     SYNTH
      M = M + MM                                                        SYNTH
      IF (MM.EQ.0) GO TO 2                                              SYNTH
         NL = NE                                                        SYNTH
         IF (.NOT.SOURCE) GO TO 62                                      SYNTH
         IF (NHE(1).EQ.1) GO TO 62                                      SYNTH
         REJECT = .TRUE.                                                SYNTH
         ENCODE (M10,226,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
   62    IF (.NOT.(PHOTO(1).AND.NE.NE.0)) GO TO 64                      SYNTH
         REJECT = .TRUE.                                                SYNTH
         ENCODE (M10,227,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
   64    ELECT(1) = .NOT.(PHOTO(1).OR.SOURCE.OR.BUFFER)                 SYNTH
         DETAIL = .NOT.(PHOTO(1).OR.SOURCE)                             SYNTH
         DETAIL = DETAIL.AND.(NE.EQ.0).AND.(N+NBUFF(1).EQ.2)            SYNTH
         ELECT(1) = (NE.EQ.1).AND.(N.EQ.2).AND.ELECT(1)                 SYNTH
         IF (VSIG(1).NE.0.) ENCODE (10,103,RATE(1)) VSIG(1)             SYNTH
         FORWARD = (VSIG(1).NE.0.).AND.(NE.LE.1)                        SYNTH
         FORWARD = FORWARD.OR.ELECT(1)                                  SYNTH
         LEFT = NKAR                                                    SYNTH
```

```
      NN = - 1                                                          SYNTH
      NE = N = 0                                                        SYNTH
      IF (.NOT.FORWARD) NKAR = 9                                        SYNTH
C                                                                       SYNTH
C     CONSTRUCT REVERSE REACTION TERM --                                SYNTH
C                                                                       SYNTH
      IF (REJECT) GO TO 2                                               SYNTH
      DECODE (6,100,LAST) (LL(NKAR+L), L = 1,6)                         SYNTH
      NKAR = NKAR+6                                                     SYNTH
      IF (K1.EQ.0) GO TO 24                                             SYNTH
      NKAR = NKAR+1                                                     SYNTH
      ENCODE (10,INTEGER,LL(NKAR)) K1                                   SYNTH
   24 IF (K1+K2.EQ.0) GO TO 28                                          SYNTH
      NKAR = NKAR+1                                                     SYNTH
      ENCODE (10,INTEGER,LL(NKAR)) K2                                   SYNTH
   28 NKAR = NKAR+1                                                     SYNTH
      ENCODE (10,INTEGER,LL(NKAR)) K3                                   SYNTH
      NKAR = NKAR+1                                                     SYNTH
      LL(NKAR) = 1H)                                                    SYNTH
      GO TO 2                                                           SYNTH
C                                                                       SYNTH
   27 IF (LP.GT.100) LP = 100                                           SYNTH
      ENCODE (100,100,KINETIC) (LINE(L), L = 1,LP)                      SYNTH
      OPTICAL = PHOTO(1).OR.PHOTO(2)                                    SYNTH
C                                                                       SYNTH
      IF (NBUFF(1).EQ.NBUFF(2).AND.NBUFF(1).LE.1) GO TO 56              SYNTH
         ENCODE (M10,224,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
         ENCODE (M10,216,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
         REJECT = .TRUE.                                                SYNTH
         GO TO 23                                                       SYNTH
C                                                                       SYNTH
   56 IF (NHE(1).EQ.NHE(2).AND.NHE(1).LE.1) GO TO 57                    SYNTH
         ENCODE (M10,225,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
         ENCODE (M10,216,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
         REJECT = .TRUE.                                                SYNTH
         GO TO 23                                                       SYNTH
C                                                                       SYNTH
   57 IF (.NOT.((SOURCE.AND.(RADIATE.OR.OPTICAL)).OR.(.NOT.PHOTO(1).    SYNTH
     1   AND.PHOTO(2)))) GO TO 66                                       SYNTH
         ENCODE (M10,227,REFER(MESS))                                   SYNTH
         MESS = MESS + M0                                               SYNTH
         REJECT = .TRUE.                                                SYNTH
C                                                                       SYNTH
C     GENERATE CHECKSUM IDENTIFIER --                                   SYNTH
C                                                                       SYNTH
   66 IF (REJECT) GO TO 23                                              SYNTH
      K1 = K2 = NBUFF(1)*(MAXGAS+1) + NHE(1)*(MAXGAS+2)                 SYNTH
      K1SQ = K2SQ = K1*K1 - 2*NBUFF(1)*(MAXGAS+1)*NHE(1)*(MAXGAS+2)     SYNTH
      DO 31 L = 1,5                                                     SYNTH
      K1 = K1 + LABEL(L,1)                                              SYNTH
      K2 = K2 + LABEL(L,2)                                              SYNTH
      K1SQ = K1SQ + LABEL(L,1)*LABEL(L,1)                               SYNTH
   31 K2SQ = K2SQ + LABEL(L,2)*LABEL(L,2)                               SYNTH
```

```
      ENCODE (10,107,LHS) K1, K1SQ
      ENCODE (10,107,RHS) K2, K2SQ
C
C     CHECK FOR DUPLICATION --
C
      IF (K.EQ.1) GO TO 23
      KM1 = K-1
      DO 25 L = 1,KM1
      READ (LTAPE) L1, L2
      IF (REJECT) GO TO 23
      REJECT = LHS.EQ.L1.AND.RHS.EQ.L2
      REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1
      IF (RADIATE) REVERSE = .FALSE.
      REJECT = REJECT.OR.REVERSE
      IF (.NOT.REJECT) GO TO 25
      ENCODE (M10,210,REFER(MESS)) L
      IF (REVERSE) ENCODE (M10,211,REFER(MESS)) L
      MESS = MESS + M0
   25 CONTINUE
C
   23 ELECT(2) = (NE.EQ.1).AND.(N.EQ.2)
      NR = NE
      REVERSE = .NOT.(RADIATE.OR.(PHOTO(1).AND.VSIG(1).EQ.0.))
      IF (.NOT.REVERSE) VSIG(2) = 0.
      IF (PHOTO(1).AND..NOT.PHOTO(2)) VSIG(2) = 0.
      ELECT(2) = .NOT.(BUFFER.OR.RADIATE.OR.OPTICAL.
     1           OR.SOURCE).AND.ELECT(2)
      IF (VSIG(2).NE.0.) ENCODE (10,103,RATE(2)) VSIG(2)
      IF (VSIG(1)+VSIG(2).NE.0.) ELECT(1) = ELECT(2) = .FALSE.
      IF (ELECT(1).AND.ELECT(2)) VSIG(1) = VSIG(2) = 0.
      ELECT(1) = ELECT(1).AND.VSIG(1).EQ.0.
      ELECT(2) = ELECT(2).AND.VSIG(2).EQ.0.
      DETAIL = (VSIG(1).NE.0.).AND.(VSIG(2).EQ.0.).AND.(NE.EQ.0).AND.
     1             (N+NBUFF(2).EQ.2).AND.DETAIL.AND.REVERSE
      REVERSE = REVERSE.AND.(DETAIL.OR.(VSIG(2).NE.0.).OR.ELECT(2))
      REJECT = .NOT.(FORWARD.OR.REVERSE).OR.REJECT
      REJECT = REJECT.OR.(K.GT.KMAX)
      IF (K.LE.KMAX) GO TO 43
      ENCODE (M10,215,REFER(MESS)) KMAX
      MESS = MESS + M0
C
C     TEST FOR CHARGE CONSERVATION --
C
   43 IF (NCH.EQ.0) GO TO 85
         REJECT = .TRUE.
         ENCODE (M10,204,REFER(MESS))
         MESS = MESS + M0
   85 IF (REJECT) GO TO 51
      IF (.NOT.(ELECT(1).OR.ELECT(2))) GO TO 51
      IF (NK.LT.NKMAX) GO TO 48
      REJECT = .TRUE.
      ENCODE (M20,218,REFER(MESS)) NKMAX
      MESS = MESS + 2*M0
      GO TO 51
C
C     SEARCH FILE OF ELECTRON CROSS SECTIONS --
C
```

```
   48 CALL DEKODE (NAME, IMAGE, L1, L2, LDUM, KAR, 10, IN, 60)
      REWIND NDATA
   74 READ (NDATA,100) IMAGE
      IF (EOF(NDATA)) 79,72
   72 CALL DEKODE (NAME, IMAGE, LL1, LL2, LDUM, KAR, 10, IN, 60)
      IF (L1.NE.LL1.OR.L2.NE.LL2) GO TO 77
         M1 = 1
         M2 = 2
         GO TO 52
C
   77 IF (L1.NE.LL2.OR.L2.NE.LL1) GO TO 78
         M1 = 2
         M2 = 1
         GO TO 52
C
C     EXHAUST NUMERICAL DATA FOR THIS PROCESS --
C
   78 READ (NDATA,100)
   73 READ (NDATA,102) LETTERS
      IF (LETTERS.EQ.1H ) GO TO 74
      READ (NDATA,102)
      GO TO 73
C
   79 ENCODE (M10,219,REFER(MESS))
      MESS = MESS + M0
      GO TO 52
C     IF DESIRED, LACK OF E- CROSS SECTIONS CAN BE MADE SUFFICIENT FOR
C     REJECTION OF THE REACTION BY THE REMOVAL OF THE ABOVE CARD, AND
C     THE REMOVAL OF *C* ON THE FOLLOWING THREE CARDS --
C     ELECT(1) = ELECT(2) = .FALSE.
C     REJECT = .TRUE.
C     GO TO 51
   52 READ (NDATA,102) (REFER(L), L = 1,M0)
      NK = NK+1
C     KOLL(NK) = K
      IF (.NOT.ELECT(1)) GO TO 54
         RATE(1) = 10H(COMPUTED)
         LEV1(NK) = K1-2
         ENCODE (M10,223,REFER(MESS)) F
         MESS = MESS + M0
   54 IF (.NOT.ELECT(2)) GO TO 51
         RATE(2) = 10H(COMPUTED)
         LEV2(NK) = K2-2
         ENCODE (M10,223,REFER(MESS)) R
         MESS = MESS + M0
C
   51 IF (NL.EQ.NR) GO TO 98
      NE = NR-NL
      IF (ELECT(1).AND.NE.LT.0) GO TO 98
      IF (SOURCE) ENCODE (M10,312,REFER(MESS))
      IF (.NOT.SOURCE) ENCODE (M10,313,REFER(MESS))
      MESS = MESS + M0
   98 IF (REJECT) NUMBER = 1H
      IF (FORWARD) GO TO 46
         ENCODE (M10,220,REFER(MESS))
         MESS = MESS + M0
   46 IF (REVERSE) GO TO 47
```

```
      IF (.NOT.FORWARD) GO TO 49
      ENCODE (M10,222,REFER(MESS))
      MESS = MESS + M0
      IF (.NOT.RADIATE) GO TO 26
      ENCODE (M10,221,REFER(MESS))
      MESS = MESS + M0
      GO TO 26
   49 MESS = MESS-M0
      ENCODE (M10,212,REFER(MESS))
      MESS = MESS + M0
      IF (.NOT.RADIATE) GO TO 47
      ENCODE (M10,221,REFER(MESS))
      MESS = MESS + M0
      GO TO 26
   47 IF (.NOT.DETAIL) GO TO 26
      RATE(2) = 10HX E(-E/KT)
      ENCODE (M10,203,REFER(MESS))
      MESS = MESS + M0
   26 IF (LC.GT.0) GO TO 7
      IF (K.GT.1) WRITE (6,202)
      IF (K.GT.1) WRITE (6,470) DATE
      WRITE (6,200)
    7 IF (MESS.EQ.1) MESS = M0+1
      REFER(MESS) = 1H
      LC = LC+1+(MESS-1)/M0
      IF (LC.GT.40) LC = 0
      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 --
      WRITE (6,201) NUMBER, (LINE(L), L = 1,45), RATE, (REFER(L), L =
     1 1,MESS)
      IF (REJECT) GO TO 1
      NTYPE = N0
      KTYPE = K
      IF (ELECT(1)) ENCODE (10,110,RATE(1)) M1,NK
      IF (ELECT(2)) ENCODE (10,110,RATE(2)) M2,NK
      KF = KR = UNDEF
      IF (FORWARD.AND.VSIG(1).NE.0.) KF = VSIG(1)
      IF (REVERSE.AND.VSIG(2).NE.0.) KR = VSIG(2)
      WRITE (LTAPE) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, (REFER(L),
     1   L = 1,5)
      PHOTO(2) = PHOTO(2).AND.PHOTO(1)
C
C     GENERATE RATE EXPRESSIONS --
C
      IF (.NOT.REVERSE) NKAR = LEFT
      WRITE (NTAPE,404) K, (KINETIC(L), L = 1,6)
      WRITE (NTAPE,105)
      WRITE (MTAPE,404) K, (KINETIC(L), L = 1,6)
      WRITE (MTAPE,105)
      IF (ELECT(1)) WRITE (NTAPE,405) K, M1, NK
      IF (ELECT(1)) WRITE (MTAPE,405) K, M1, NK
      IF (ELECT(2)) WRITE (NTAPE,409) K, M2, NK
      IF (ELECT(2)) WRITE (MTAPE,409) K, M2, NK
      IF (ELECT(1).OR.ELECT(2)) WRITE (NTAPE,105)
      IF (ELECT(1).OR.ELECT(2)) WRITE (MTAPE,105)
      IF (SOURCE) GO TO 65
```

```
      IF (.NOT.DETAIL) GO TO 44
C     WRITE EXPRESSIONS FOR DETAILED BALANCE RELATION --
      WRITE (NTAPE,408)
      WRITE (NTAPE,105)
      WRITE (MTAPE,408)
      WRITE (MTAPE,105)
      GO TO 65
   44 IF (.NOT.PHOTO(1)) GO TO 35
      IF (PHOTO(2)) WRITE (NTAPE,412) NPHOTON
      IF (.NOT.PHOTO(2)) WRITE (NTAPE,413) NPHOTON
      IF (PHOTO(2)) WRITE (MTAPE,412) NPHOTON
      IF (.NOT.PHOTO(2)) WRITE (MTAPE,413) NPHOTON
      IF (.NOT.PHOTO(2)) GO TO 35
      IF (LASER) GO TO 35
      LASER = .TRUE.
   65 DO 41 M = 1,2
      ENCODE (10,106,LINE) M
      DO 42 L = 1,5
      J = LABEL(L,M)
      IF (J.EQ.0) GO TO 92
      L1 = L+1
      ENCODE (10,108,LINE(L1)) J
   42 CONTINUE
   92 IF (DETAIL) WRITE (MTAPE,109) (LINE(N), N = 1,L)
   41 WRITE (NTAPE,109) (LINE(N), N = 1,L)
      IF (SOURCE) GO TO 35
      IF (DETAIL) GO TO 58
      WRITE (NTAPE,417)
      WRITE (NTAPE,105)
      GO TO 35
   58 WRITE (NTAPE,410) K, K
      WRITE (NTAPE,105)
      WRITE (MTAPE,410) K, K
      WRITE (MTAPE,105)
   35 WRITE (NTAPE,100) (LL(L), L = 1,NKAR)
      IF (SOURCE) WRITE (NTAPE,311)
      IF (PHOTO(1)) WRITE (NTAPE,416)
      WRITE (NTAPE,105)
      WRITE (NTAPE,217) K
      DO 18 I = 1,NTYPE
      N1 = N2 = 0
      DO 14 L = 1,MAX
      IF (LABEL(L,1).EQ.I) N1 = N1+1
   14 IF (LABEL(L,2).EQ.I) N2 = N2+1
      N = N2-N1
C
C     IT IS UNDERSTOOD THAT, IF RAD APPEARS ON BOTH SIDES OF THE EQUA-
C     TION, A NET INCREASE IN RADIATION RESULTS.)
C
      IF (I.EQ.NPHOTON.AND.N1.EQ.1.AND.N2.EQ.1) N = 1
      IF (N.EQ.0) GO TO 18
      IF (N.GT.0) SIGN = 1H+
      IF (N.LT.0) SIGN = 1H-
      ENCODE (10,205,NSIGN) SIGN
      N = IABS(N)
      IND = 1
      IF (I.GT.9) IND = 2
```

```
      IF (I.GT.99) IND = 3
      ENCODE (50,500,FORM) IND, IND, MAX
      WRITE (NTAPE,FORM) I, I, (NSIGN, J = 1,N)
   18 CONTINUE
C
      IF (.NOT.SOURCE) GO TO 95
      WRITE (NTAPE,305)
C     DETERMINE IF SOURCE TERM CORRESPONDS TO IONIZATION --
      DO 94 L = 1,MAX
      J = LABEL(L,2)
      IF (J.EQ.0) GO TO 97
      IF (J.EQ.2) GO TO 96
   94 CONTINUE
C     SOURCE TERM WAS IONIZATION --
   96 WRITE (NTAPE,310)
   97 WRITE (NTAPE,105)
      WRITE (NTAPE,309)
      WRITE (NTAPE,304)
      GO TO 90
   95 NE = NR-NL
      IF (ELECT(1).OR.ELECT(2).OR.(NE.EQ.0)) GO TO 90
      WRITE (NTAPE,105)
      IF (NE.GT.0) SIGN = 1H+
      IF (NE.LT.0) SIGN = 1H-
      IF (NE.GT.0) WRITE (NTAPE,424)
      IF (NE.LT.0) WRITE (NTAPE,425)
      WRITE (NTAPE,105)
      NE = IABS(NE)
      WRITE (NTAPE,426) (SIGN, L = 1,NE)
   90 IF (RADIATE) WRITE (NTAPE,418)
C
C ...............................................................
C
C     GENERATE JACOBIAN --
C
C     THE RATE FOR REACTION K, ENCODED INTO THE VECTOR LL ABOVE, IS R.
C     IF SPECIES I OCCURS NR TIMES ON THE RIGHT, NL TIMES ON THE LEFT,
C     OF REACTION K, THEN THE CONTRIBUTION OF REACTION K TO NDOT(I) IS
C     NI*R, WHERE N = (NR - NL).  THE FOLLOWING LOOP CALCULATES THE
C     DERIVATIVE S = DS/DN(J) FOR EVERY SPECIES J WHICH OCCURS.  FOR
C     EVERY SPECIES I, REACTION K MAKES A CONTRIBUTION OF NI*S TO THE
C     JACOBIAN PHI(I,J) = DNDOT(I)/DN(J).
C
      DO 89 J = 1,NTYPE
      IP = JP = 0
      LINE(1) = 1H(
      NKAR = 1
      M = JN = NJ = 0
   83 M = M+1
      IF (M.GT.2) GO TO 82
      IF (M.EQ.2) GO TO 88
      IF (.NOT.FORWARD) GO TO 83
      NSIGN = 1H
      KJ = 1HF
      GO TO 84
   88 IF (.NOT.REVERSE) GO TO 82
      NSIGN = 1H-
```

```
      KJ = 1HR
C     NJ IS THE NUMBER OF TIMES SPECIES J APPEARS (ON LHS OR RHS) --
   84 NJ = 0
      DO 63 L = 1,MAX
      JL = LABEL(L,M)
      IF (JL.NE.J) GO TO 63
      NJ = NJ+1
      LJ = L
   63 CONTINUE
      JN = JN + NJ
      IF (NJ.EQ.0) GO TO 83
      IP = IP+1
      NKAR = NKAR+1
      LINE(NKAR) = NSIGN
      NKAR = NKAR+1
      ENCODE (10,306,LINE(NKAR)) KJ, K
      IF (NJ.GT.1) ENCODE (10,307,LINE(NKAR)) NJ, KJ, K
      DO 91 L = 1,MAX
      IF (L.EQ.LJ) GO TO 91
      JL = LABEL(L,M)
      IF (JL.EQ.0) GO TO 83
      NKAR = NKAR+1
      ENCODE (10,308,LINE(NKAR)) JL
   91 CONTINUE
      GO TO 83
   82 IF (JN.EQ.0) GO TO 89
      LKAR = NKAR = NKAR+1
      LINE(NKAR) = 1H)
      IF (NHE(1).EQ.0) GO TO 68
      NKAR = NKAR+1
      LINE(NKAR) = 9H*IBEAM/E0
      JP = IP
   68 IF (NBUFF(1).EQ.0) GO TO 69
      NKAR = NKAR+1
      LINE(NKAR) = NTOT
      JP = IP
   69 IF (JP.NE.2) LINE(1) = LINE(LKAR) = 1H
      LKAR = 0
      DO 81 L = 1,NKAR
      DECODE (10,100,LINE(L)) (KAR(L1), L1 = 1,10)
      DO 93 L2 = 1,10
      IF (KAR(L2).EQ.1H ) GO TO 93
      LKAR = LKAR+1
      LL(LKAR) = KAR(L2)
   93 CONTINUE
   81 CONTINUE
      WRITE (MTAPE,503) (LL(L), L = 1,LKAR)
      IF (PHOTO(1)) WRITE (MTAPE,416)
      WRITE (MTAPE,105)
C
      DO 86 I = 1,NTYPE
      N1 = N2 = 0
      DO 87 L = 1,MAX
      IF (LABEL(L,1).EQ.I) N1 = N1+1
   87 IF (LABEL(L,2).EQ.I) N2 = N2+1
      N = N2-N1
      IF (I.EQ.NPHOTON.AND.N1.EQ.1.AND.N2.EQ.1) N = 1
```

```
      IF (N.EQ.0) GO TO 86
      IF (N.GT.0) SIGN = 1H+
      IF (N.LT.0) SIGN = 1H-
      ENCODE (10,205,NSIGN) SIGN
      N = IABS(N)
      WRITE (MTAPE,502) I, J, I, J, (NSIGN, L = 1,N)
      IF (I.NE.1.OR.J.NE.1) GO TO 86
         WRITE (NTAPE,105)
         WRITE (NTAPE,503) (LL(L), L = 1,LKAR)
         IF (PHOTO(2)) WRITE (NTAPE,411)
         WRITE (NTAPE,420) NSIGN
   86 CONTINUE
      IF (RADIATE) WRITE (MTAPE,419) J, J
      WRITE (MTAPE,105)
C
   89 CONTINUE
C
      WRITE (NTAPE,105)
      GO TO 10
    5 WRITE (MTAPE,423)
      WRITE (6,202)
      WRITE (6,470) DATE
      NREJ = INPUT-KTYPE
      IF (EXIT.AND.LC.GT.30) WRITE (6,SKIP)
      WRITE (6,302) INPUT, KTYPE, KMAX, NREJ, NK, NTYPE, MAXGAS
      IF (EXIT) WRITE (6,303)
C
C     EDIT THE REACTIONS TO DETERMINE WHERE EACH SPECIES OCCURS --
C
      LC = 0
      KOUNT = 40
      TEST = .FALSE.
      DO 50 I = 1,NTYPE
      N = 0
      REWIND LTAPE
      DO 33 K = 1,KTYPE
      READ (LTAPE) LHS, RHS, LABEL
      DO 34 M = 1,2
      DO 34 L = 1,5
      IF (LABEL(L,M).EQ.I) GO TO 36
   34 CONTINUE
      GO TO 37
   36 N = N+1
      LINE(N) = K
   37 IF (N.LT.160.AND.K.LT.KTYPE) GO TO 33
      IF (N.EQ.0) GO TO 33
      IF (LC.NE.0) GO TO 32
      IF (TEST) WRITE (6,202)
      IF (TEST) WRITE (6,470) DATE
      TEST = .TRUE.
      WRITE (6,300) NTYPE
      WRITE (6,202)
   32 LC = LC + 2 + N/20
      WRITE (6,301) I, GAS(I), (LINE(L), L = 1,N)
      IF (LC.GT.KOUNT) LC = 0
      N = 0
   33 CONTINUE
```

```
   50 CONTINUE
      WRITE (6,202)
      WRITE (6,470) DATE
C
      WRITE (NSCRTCH) MAXGAS, NTYPE, KMAX, KTYPE, NKMAX, NK, EXIT
      WRITE (NSCRTCH) (GAS(I), I = 1,NTYPE)
      WRITE (NSCRTCH) (LEV1(I), LEV2(I), I = 1,NK)
C
C     COPY LTAPE ONTO NSCRTCH --
      REWIND LTAPE
      DO 59 K = 1,KTYPE
      READ (LTAPE) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM
      WRITE (NSCRTCH) LHS, RHS, LABEL, RATE, KF, KR, KINETIC, COMM
   59 CONTINUE
C
      NP1 = NTYPE + 1
      NP2 = NTYPE + 2
C
      WRITE (NTAPE,414)
      WRITE (NTAPE,105)
      WRITE (NTAPE,406)
      WRITE (MTAPE,406)
      WRITE (NTAPE,105)
      WRITE (MTAPE,105)
      ENCODE (80,506,KODE) NP1, NP2
      WRITE (NTAPE,102) KODE
      WRITE (MTAPE,102) KODE
      WRITE (NTAPE,105)
      WRITE (MTAPE,105)
      WRITE (NTAPE,508)
      WRITE (MTAPE,508)
      ENCODE (80,509,KODE) NP1, NP2
      WRITE (NTAPE,102) KODE
      ENCODE (80,513,KODE) NP2, NP1, NP2
      WRITE (NTAPE,102) KODE
      WRITE (NTAPE,440)
      WRITE (NTAPE,105)
      ENCODE (80,514,KODE) NP1, NP1
      WRITE (NTAPE,102) KODE
      ENCODE (80,515,KODE) NP2, NP1
      WRITE (NTAPE,102) KODE
      WRITE (NTAPE,105)
C
      ENCODE (80,511,KODE) NP1, NP1
      WRITE (MTAPE,102) KODE
      ENCODE (80,516,KODE) NP1, NP2
      WRITE (MTAPE,102) KODE
      ENCODE (80,517,KODE) NP2, NP1
      WRITE (MTAPE,102) KODE
      ENCODE (80,518,KODE) NP2, NP2
      WRITE (MTAPE,102) KODE
      ENCODE (80,519,KODE) NP2, NP2
      WRITE (MTAPE,102) KODE
      WRITE (MTAPE,440)
      WRITE (MTAPE,105)
      ENCODE (80,521,KODE) NP1, NP1
      WRITE (MTAPE,102) KODE
```

```
      ENCODE (80,520,KODE) NP1, NP1
      WRITE (MTAPE,102) KODE
      WRITE (MTAPE,105)
C
      WRITE (NTAPE,406)
      WRITE (NTAPE,105)
      WRITE (MTAPE,406)
      WRITE (MTAPE,105)
      ENCODE (80,207,KODE)
      WRITE (NTAPE,102) KODE
      WRITE (NTAPE,105)
      WRITE (MTAPE,102) KODE
      WRITE (MTAPE,105)
      N2 = 0
   12 N1 = N2+1
      IF (N1.GT.NTYPE) GO TO 19
      N2 = N1+3
      IF (N2.GT.NTYPE) N2 = NTYPE
      ENCODE (80,206,KODE) (L, GAS(L), L = N1,N2)
      WRITE (NTAPE,102) KODE
      WRITE (MTAPE,102) KODE
      GO TO 12
   19 WRITE (NTAPE,105)
      WRITE (NTAPE,406)
      WRITE (NTAPE,105)
      WRITE (NTAPE,440)
      WRITE (NTAPE,450)
      WRITE (MTAPE,105)
      WRITE (MTAPE,406)
      WRITE (MTAPE,105)
      WRITE (MTAPE,440)
      WRITE (MTAPE,450)
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
      WRITE (NTAPE,600)
      WRITE (NTAPE,105)
      WRITE (NTAPE,406)
      WRITE (NTAPE,105)
      WRITE (NTAPE,604)
      WRITE (NTAPE,105)
      WRITE (NTAPE,460)
      WRITE (NTAPE,406)
      WRITE (NTAPE,105)
      ENCODE (80,601,KODE)
      WRITE (NTAPE,102) KODE
      WRITE (NTAPE,105)
      DO 80 I = 1,NK
      ENCODE (80,602,KODE) M1, I, LEV1(I)
      IF (LEV1(I).EQ.0) ENCODE (80,603,KODE) M1, I
      WRITE (NTAPE,102) KODE
      ENCODE (80,602,KODE) M2, I, LEV2(I)
      IF (LEV2(I).EQ.0) ENCODE (80,603,KODE) M2, I
      WRITE (NTAPE,102) KODE
   80 WRITE (NTAPE,105)
```

```
      WRITE (NTAPE,440)
      WRITE (NTAPE,450)
C
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
C-----------------------  FORMAT STATEMENTS  -----------------------
C
  100 FORMAT (80A1)
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 (S
     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
     5 A COUPLED ELECTRON ANALYSIS.)*//4X,*I*,12X,*REACTION(I)*,34X,
     6*RATE CONSTANTS*14X*RATE REFERENCES AND/OR COMMENTS*/8X*(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))
C
  207 FORMAT (*C  THE FOLLOWING MOLECULAR SPECIES (WITH LABELS) WERE INC
     1LUDED --*)
C
  208 FORMAT (*C  THE FOLLOWING REACTIONS DEFINE THE KINETICS --*)
C
  209 FORMAT (I5)
C
  210 FORMAT (*REACTION IGNORED -- SAME AS NUMBER*,I4,*.*)
C
  211 FORMAT (*REACTION IGNORED -- REVERSE OF NO.*,I4,*.*)
C
  212 FORMAT (*REACTION IS IGNORED -- KF = KR = 0.*)
C
  213 FORMAT (*REACTION IS IGNORED -- MORE THAN*,I3,* DIFFERENT GAS SPEC
     1IES ARE NOT PERMITTED WITH PRESENT DIMENSION.*)
C
  214 FORMAT (*> *,I1,* SPECIES ON LHS OR RHS NOT PERMITTED.*)
C
  215 FORMAT (*MORE THAN *,I3,* REACTIONS ARE IGNORED.*)
C
  216 FORMAT (*BAD SYNTAX -- UNRECOGNIZABLE REACTION IS IGNORED.*)
C
  217 FORMAT (11X*RATE(*I3*) = R*56X)
C
  218 FORMAT (*REACTION REJECTED -- NO MORE THAN*,I3,* REACTIONS AREALLO
     1WED FOR THE COUPLED E- KINETICS ANALYSIS.*)
C
  219 FORMAT (*WARNING -- NO E- CROSS SECTION DATA WAS FOUND.*)
C
  220 FORMAT (*FORWARD REACTION IS IGNORED -- KF = 0.*)
C
  221 FORMAT (*NO REVERSE REACTION ALLOWED FOR RADIATIVE DECAY.*)
C
  222 FORMAT (*REVERSE REACTION IS IGNORED -- KR = 0.*)
C
  223 FORMAT (A7,* RATE IS OBTAINED FROM E- KINETICS ANALYSIS*)
C
  224 FORMAT (*IMPROPER BUFFER GAS SPECIFICATION.*)
C
  225 FORMAT (*IMPROPER HIGH ENERGY ELECTRON TERMS.*)
C
  226 FORMAT (*THREE-BODY HE- COLLISION NOT ALLOWED.*)
C
  227 FORMAT (*THIS RADIATIVE PROCESS NOT ALLOWED.*)
C
  300 FORMAT (1H1/35X,*SUMMARY OF REACTIONS FOR WHICH EACH SPECIES OCCUR
     1S: NTYPE = *,I3/31X*(THIS EDIT PERMITS RAPID DELETION OF ANY SPECI
     2ES FROM THE KINETIC SYSTEM)*///7X,*I*5X*GAS(I)*40X*REACTIONS CONTA
     3INING GAS(I)*)
C
  301 FORMAT (/I8,5X,A10,5X,I4,19(*,*I4)/(28X,I4,19(*,*I4)))
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
```

```
     3 AN E- KINETICS ANALYSIS. *,I3,* SEPARATE SPECIES WERE ENCOUNTERED SYNTH
     4 (MAXIMUM ALLOWED =*,I3,*).*/)                                     SYNTH
C                                                                        SYNTH
  303 FORMAT (4X*ERRORS WHICH WERE DETECTED IN PROCESSING THE INPUT REAC SYNTH
     1TION SCHEME MAY CAUSE PROGRAM TERMINATION IF THEY HAVE BEEN SPECIF SYNTH
     2IED TO*/4X,*BE TREATED AS FATAL.  MODIFICATIONS OF THE REACTION SC SYNTH
     3HEME, CORRECTIONS IN REACTION SYNTAX, CHANGES IN DIMENSION STORAGE SYNTH
     4, OR*/4X,*ADDITIONS TO THE E- CROSS SECTION FILE MAY BE REQUIRED T SYNTH
     5O REMOVE ALL OF THE ERROR DECLARATIONS.*)                          SYNTH
C                                                                        SYNTH
  304 FORMAT (11X≠DVDX = DVDX + U*S≠52X)                                 SYNTH
C                                                                        SYNTH
  305 FORMAT (11X*U = (E2 - E1)*,56X)                                    SYNTH
C                                                                        SYNTH
  306 FORMAT (*K*,A1,*(*,I3,*)*)                                         SYNTH
C                                                                        SYNTH
  307 FORMAT (I1,3H.*K,A1,*(*,I3,*)*)                                    SYNTH
C                                                                        SYNTH
  308 FORMAT (4H*NO(,I2,*)*)                                             SYNTH
C                                                                        SYNTH
  309 FORMAT (*C*5X*E-BEAM ENERGY DEPOSITION --*47X)                     SYNTH
C                                                                        SYNTH
  310 FORMAT (*C*79X/*C*5X*SECONDARY ELECTRON CREATION--*45X/11X*SB = SB SYNTH
     1 + R*58X/11X*U = U + UPLUS*56X)                                    SYNTH
C                                                                        SYNTH
  311 FORMAT (6X*S = R*69X/6X≠R = R*IBEAM/E0≠60X)                        SYNTH
C                                                                        SYNTH
  312 FORMAT (*E- CREATION ASSUMED TO BE OVER ENERGY DISTRIBUTION*)      SYNTH
C                                                                        SYNTH
  313 FORMAT (*E- CREATED (OR LOST) ASSUMED TO BE AT ZERO ENERGY*)       SYNTH
C                                                                        SYNTH
  400 FORMAT (≠*DECK,DNDT≠,70X/6X*SUBROUTINE DNDT (N, T, NO, NDOT)*42X)  SYNTH
C                                                                        SYNTH
  401 FORMAT (6X*DIMENSION NO(1), NDOT(1)*50X)                           SYNTH
C                                                                        SYNTH
  402 FORMAT (6X*REAL NO, NTOT, NDOT, NOISE, NE, KF, KR, KB, KT, MU, LO, SYNTH
     1 IBEAM,*12X/5X*1 JBEAM, LENGTH*60X)                                SYNTH
C                                                                        SYNTH
  403 FORMAT (6X≠IBEAM = JBEAM*DEPOSIT*SHAPE(T)≠44X/6X≠KT = KB*TMOL≠62X) SYNTH
C                                                                        SYNTH
  404 FORMAT (1HC,I3,2X,6A10,5X)                                         SYNTH
C                                                                        SYNTH
  405 FORMAT (*C      FORWARD RATE IS OBTAINED FROM E(-) KINETICS ANALYSI SYNTH
     1S*,22X/11X,*KF(*,I3,*) = VSIG(*,I1,*,*,I2,*)*,48X)                 SYNTH
C                                                                        SYNTH
  406 FORMAT (*C  *,69(1H.),8X)                                          SYNTH
C                                                                        SYNTH
  407 FORMAT (6X*DO 1 I = 1,N*62X/4X*1 NDOT(I) = 0.*62X/6X≠NDOT(1) = - C SYNTH
     1*GAMMA*NO(1)≠49X/6X*ALPHA = GAIN = HNU = FREQ = NOISE = DVDX = 0.* SYNTH
     229X/6X*SB = S0 = 0.0*61X)                                          SYNTH
C                                                                        SYNTH
  408 FORMAT (*C*,5X,*REVERSE RATE IS OBTAINED FROM DETAIL BALANCE --*,  SYNTH
     127X)                                                               SYNTH
C                                                                        SYNTH
  409 FORMAT (*C      REVERSE RATE IS OBTAINED FROM E(-) KINETICS ANALYSI SYNTH
     1S*,22X/11X,*KR(*,I3,*) = VSIG(*,I1,*,*,I2,*)*,48X)                 SYNTH
C                                                                        SYNTH
```

```
  410 FORMAT (*      KR(*,I3,*) = KF(*,I3,18H)*EXP(-(E1-E2)/KT),40X)     SYNTH
C                                                                       SYNTH
  411 FORMAT (11X*GAIN = GAIN + R*54X)                                  SYNTH
C                                                                       SYNTH
  412 FORMAT (*C*5X*(STIMULATED EMISSION PROCESS, WITH NO(*I2*) = INTEN/ SYNTH
     1C/HNU)*17X/*C*79X)                                                SYNTH
C                                                                       SYNTH
  413 FORMAT (*C*5X*(RADIATIVE ABSORPTION PROCESS, WITH NO(*I2*) = INTEN SYNTH
     1/C/HNU)*16X/*C*79X)                                               SYNTH
C                                                                       SYNTH
  414 FORMAT (*C*5X*COMPUTE FINAL EXPRESSION FOR D/DT(PHOTON DENSITY) -- SYNTH
     1*22X/*C*79X/6X≠NDOT(1) = (LENGTH/CAVITY)*(NDOT(1) + NOISE)≠31X/*C* SYNTH
     279X/6X*ABSORB = GAIN - ALPHA*53X/6X,≠DVDX = DEPOSIT*DVDX≠55X)     SYNTH
C                                                                       SYNTH
  415 FORMAT (*C*5X≠GAIN    = SIGMA*(N2-N1) IS THE LASER TRANSITION GAIN≠ SYNTH
     1 23X/*C*5X≠ABSORB = SUMK[SIGMA(K)*NK] IS THE TOTAL ABSORPTION OF T SYNTH
     2HE MEDIUM≠10X/*C*5X*ALPHA  = (GAIN-ABSORB) IS THE NET GAIN IN THE  SYNTH
     3MEDIUM*22X/*C*5X*GAMMA  = THRESHHOLD GAIN COEFFICIENT (CM-1)*31X)  SYNTH
C                                                                       SYNTH
  416 FORMAT (6X≠R = R*C≠67X)                                           SYNTH
C                                                                       SYNTH
  417 FORMAT (6X≠HNU = E0*(E1 - E2)≠56X/6X*FREQ = HNU/H*62X)            SYNTH
C                                                                       SYNTH
  418 FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE --* SYNTH
     131X/*C*79X/11X≠NOISE = NOISE + R*OMEGA/4./PI≠40X)                 SYNTH
C                                                                       SYNTH
  419 FORMAT (*C*79X/*C*5X*PHOTON NUMBER DENSITY INCREASED BY NOISE --* SYNTH
     131X/*C*79X/11X≠R = R*OMEGA/4./PI≠51X/11X*PHI(1,*I2*) = PHI(1,*I2 SYNTH
     2*) + R*44X)                                                       SYNTH
C                                                                       SYNTH
  420 FORMAT (11X*ALPHA = ALPHA*A4,52X)                                 SYNTH
C                                                                       SYNTH
  421 FORMAT (6X*DATA KB, E0, H, C, PI /*1PE10.3*,*1PE10.3*,*1PE10.3*,*  SYNTH
     118X/5X*1     *1PE10.3*, 3.14159 /*49X/*C*79X)                     SYNTH
C                                                                       SYNTH
  422 FORMAT (*C*5X*CAVITY = MIRROR SEPARATION (CM)*44X/*C*5X*LENGTH = L SYNTH
     1ENGTH OF ACTIVE MEDIUM (CM)*37X/*C*5X≠OMEGA  = AREA/CAVITY**2≠51X/ SYNTH
     2*C*5X*AREA  = AREA OF OPTICS (CM2)*45X/*C*5X*GAMMA  = [LOSS + LN( SYNTH
     31/R)/2]/LENGTH*40X/*C*79X)                                        SYNTH
C                                                                       SYNTH
  423 FORMAT (6X*RATIO = LENGTH/CAVITY*53X/6X*DO 3 I = 1,N*62X/4X≠3 PHI( SYNTH
     11,I) = RATIO*PHI(1,I)≠49X/*C*79X)                                 SYNTH
C                                                                       SYNTH
  424 FORMAT (*C*5X*CREATION OF (ZERO ENERGY) SECONDARY ELECTRONS--*27X) SYNTH
C                                                                       SYNTH
  425 FORMAT (*C*5X*LOSS OF (ZERO ENERGY) SECONDARY ELECTRONS --*30X)   SYNTH
C                                                                       SYNTH
  426 FORMAT (11X*S0 = S0 *A1* R*58X)                                   SYNTH
C                                                                       SYNTH
  440 FORMAT (6X*RETURN*68X)                                            SYNTH
C                                                                       SYNTH
  450 FORMAT (6X*END*71X)                                               SYNTH
C                                                                       SYNTH
  460 FORMAT (*C*5X*THE GENERAL KINETICS SYNTHESIS PROGRAM WHICH AUTOMAT SYNTH
     1ICALLY GEN-*11X/*C*5X*ERATED THIS SUBROUTINE WAS DEVELOPED BY --* SYNTH
     232X/*C*79X/*C*15X,40(1H-),24X/*C*15X*I*38X*I*24X/*C*15X*I   DR. WI SYNTH
     3LLIAM B. LACINA*14X*I*24X/*C*15X*I   NORTHROP RESEARCH AND TECHNOL SYNTH
```

```
      40GY    I*24X/*C*15X*I    ONE RESEARCH PARK*18X*I*24X/*C*15X*I    PALO SYNTH
      5S VERDES PENINSULA. CA 90274    I*24X/*C*15X*I    TEL: (213) 377-481 SYNTH
      61, EXT. 322*6X*I*24X/*C*15X*I*38X*I*24X/*C*15X,40(1H-),24X/*C*79X) SYNTH
C                                                                          SYNTH
  470 FORMAT (/12X*GENERALIZED KINETICS SYNTHESIS CODE: DR. WILLIAM B. L SYNTH
     1ACINA, NORTHROP RESEARCH AND TECHNOLOGY.  DATE:*A11)                SYNTH
C                                                                          SYNTH
  500 FORMAT (*(11X,5HNDOT(,I*,I3,*,9H) = NDOT(,I*,I3,*,1H),*,I2,*A4)*)  SYNTH
C                                                                          SYNTH
  502 FORMAT (11X*PHI(*,I2,*,*,I2,*) = PHI(*,I2,*,*,I2,*)*,10A4,6X)       SYNTH
C                                                                          SYNTH
  503 FORMAT (6X,*R = *,62A1,8X/(5X,*$*10X,56A1,8X))                      SYNTH
C                                                                          SYNTH
  504 FORMAT (6X*DIMENSION PHI(N,1), NO(1)*49X)                           SYNTH
C                                                                          SYNTH
  505 FORMAT (*C*5X*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT  SYNTH
     1FILE OF SYM-*10X/*C*5X*BOLIC REACTIONS WHICH DEFINE A COUPLED SYST SYNTH
     2EM OF ELECTRON AND*13X/*C*5X*MOLECULAR KINETICS EQUATIONS.  IT RET SYNTH
     3URNS THE JACOBIAN MATRIX,*11X/*C*79X/*C*20X,*PHI(I,J) = D[NDOT(I)] SYNTH
     4/D[NO(J)]*29X/*C*79X/*C*5X*WHERE I,J = 1,2,3,...NTYPE.  N IS THE D SYNTH
     5IMENSION DECLARATOR FOR PHI*8X/*C*5X*IN THE CALLING PROGRAM.  THE  SYNTH
     6RATE CONSTANTS KF AND KR HAVE UNITS*10X/*C*5X*OF CM2, SEC-1, CM3/S SYNTH
     7EC, CM6/SEC, ... AS APPROPRIATE.*,22X)                              SYNTH
C                                                                          SYNTH
  506 FORMAT (*C*5X*EXTERNAL CIRCUIT EQUATIONS (Q → NO(*,I2,*), AND I →  SYNTH
     1NO(*,I2,*)) -- *15X)                                                 SYNTH
C                                                                          SYNTH
  507 FORMAT (6X*DO 1 I = 1,N*,62X/6X*DO 1 J = 1,N*,62X/4X*1 PHI(I,J) =  SYNTH
     10.*61X/6X≠PHI(1,1) = - C*GAMMA≠54X)                                  SYNTH
C                                                                          SYNTH
  508 FORMAT (6X*NE = NO(2)*64X/6X*IF (NE.EQ.0.) NE = 1.0*52X/6X≠CONDUCT SYNTH
     1 = NE*E0*MU≠56X/6X*RD = D/AREA/CONDUCT*55X/*C*79X/6X*IF (L0.EQ.0.) SYNTH
     2 GO TO 2*53X)                                                        SYNTH
C                                                                          SYNTH
  509 FORMAT (6X*DQDT = NDOT(*I2*) = NO(*I2*)*)                           SYNTH
C                                                                          SYNTH
  510 FORMAT (≠*DECK,JACOB≠69X/6X*SUBROUTINE JACOB (N, T, NO, PHI)*42X)  SYNTH
C                                                                          SYNTH
  511 FORMAT (6X*PHI(*,I2*,*I2*) = 0.*59X)                                SYNTH
C                                                                          SYNTH
  512 FORMAT (6X*COMMON / CONST / NTOT, TMOL, FREQ, HNU*36X/6X*COMMON /  SYNTH
     1DISCH / L0, C0, R0, MU, AREA, D*34X/6X*COMMON / SOURCE / UPLUS, JB SYNTH
     2EAM, DVDX, DEPOSIT, ENERGY, SB, S0*13X/6X*COMMON / GAINS / ALPHA,  SYNTH
     3GAMMA, GAIN, ABSORB, OMEGA, LENGTH, CAVITY*8X/*C*79X)               SYNTH
C                                                                          SYNTH
  513 FORMAT (6X*DIDT = NDOT(*I2*) = (-NO(*I2*)/C0 - (R0 + RD)*1H*,*NO(* SYNTH
     1I2*))/L0*22X)                                                        SYNTH
C                                                                          SYNTH
  514 FORMAT (4X*2 DQDT = NDOT(*I2*) = - NO(*I2*)/C0/(R0 + RD)*)          SYNTH
C                                                                          SYNTH
  515 FORMAT (6X*CURRENT = NO(*I2*) = NDOT(*I2*)*)                        SYNTH
C                                                                          SYNTH
  516 FORMAT (6X*PHI(*I2*,*I2*) = 1.0*)                                   SYNTH
C                                                                          SYNTH
  517 FORMAT (6X*PHI(*I2*,*I2*) = -1./L0/C0*52X)                          SYNTH
C                                                                          SYNTH
  518 FORMAT (6X*PHI(*I2*,*I2*) = -(R0 + RD)/L0*48X)                      SYNTH
```

```
C
  519 FORMAT (6X*PHI(*I2*, 2) = NO(*I2*)*1H*,*RD/NE/L0*46X)
C
  520 FORMAT (6X*PHI(*I2*, 2) = NO(*I2≠)*DIDQ*RD/NE/(R0 + RD)≠)
C
  521 FORMAT (4X*2 PHI(*I2*,*I2*) = DIDQ = - 1./C0/(R0 + RD)*)
C
  600 FORMAT (≠*DECK,LEVELS≠68X/6X*SUBROUTINE LEVELS (N1, N2, NO)*44X)
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 N1(I), N2(I)*10X/*C*5X*OF THE (LOWER AND UPPER) LEVELS INVOLVED I
     2N THE ITH INELASTIC*13X/*C*5X*SCATTERING PROCESS INCLUDED IN THE C
     2OUPLED E- KINETICS ANALYSIS.*10X)
C
  605 FORMAT (*C*5X*THIS SUBROUTINE WAS SYNTHESIZED BY EDITING AN INPUT
     1FILE OF SYM-*10X/*C*5X*BOLIC REACTIONS WHICH DEFINE A COUPLED SYST
     2EM OF ELECTRON AND*13X/*C*5X*MOLECULAR KINETICS. IT RETURNS THE RA
     3TES NDOT(I) = (D/DT)NO(I),*11X/*C*5X*I = 1,2,...NTYPE (CM-3/SEC).
     4RATE CONSTANTS KF AND KR HAVE UNITS*10X/*C*5X*OF CM2, SEC-1, CM3/S
     5EC, CM6/SEC, ... AS APPROPRIATE.*,22X)
C
C ---------------------------------------------------------------------
C
      RETURN
      END


      SUBROUTINE ANALYZE (NTYPE, KTYPE, RATE, NTIME, RPCT, FLAG, PMAX,
     1 GAS, PCT, KAPTION, LTAPE, MTAPE, NTAPE)
C
C ......................................................................
C
C     THIS SUBROUTINE ANALYZES THE CONTRIBUTIONS OF ALL REACTIONS TO
C     EVERY SPECIES, AND PRINTS OUT DIAGNOSTICS SUMMARIZING THE SENSI-
C     TIVITY OF EACH REACTION TO THE TOTAL CALCULATION.
C
C ......................................................................
C
      DIMENSION RATE(1), NTIME(1), KAPTION(4), KODE(10), PMAX(1),
     1   RPCT(1), FLAG(1), GAS(1)
C
      LOGICAL FLAG, TEST
C
      REWIND LTAPE
      REWIND MTAPE
      REWIND NTAPE
      DO 8 I = 1,NTYPE
    8 PMAX(I) = 0.
      DO 1 K = 1,KTYPE
      R = RATE(K)
      READ (MTAPE) (NTIME(L), L = 1,NTYPE)
      DO 2 I = 1,NTYPE
      RPCT(I) = 0.
      IF (R.EQ.0.) GO TO 2
      NI = NTIME(I)
      IF (NI.EQ.0) GO TO 2
      RPCT(I) = NI*R
      PABS = ABS(RPCT(I))
      IF (PABS.GT.PMAX(I)) PMAX(I) = PABS
    2 CONTINUE
    1 WRITE (NTAPE) (RPCT(L), L = 1,NTYPE)
      REWIND NTAPE
```

```
C                                                                        ANALYZE
      DO 3 K = 1,KTYPE                                                   ANALYZE
      PABS = 0.                                                          ANALYZE
      READ (NTAPE) (RPCT(L), L = 1,NTYPE)                                ANALYZE
      DO 4 I = 1,NTYPE                                                   ANALYZE
      IF (PMAX(I).EQ.0.) GO TO 4                                         ANALYZE
      PERCENT = RPCT(I) = 100.*RPCT(I)/PMAX(I)                           ANALYZE
      PERCENT = ABS(PERCENT)                                             ANALYZE
      IF (PERCENT.GT.PCT) FLAG(K) = .FALSE.                              ANALYZE
      IF (PERCENT.GT.PABS) PABS = PERCENT                                ANALYZE
    4 CONTINUE                                                           ANALYZE
    3 WRITE (LTAPE) (RPCT(L), L = 1,NTYPE), PABS                         ANALYZE
C                                                                        ANALYZE
      NA = 1                                                             ANALYZE
    5 NB = NA+9                                                          ANALYZE
      IF (NB.GT.NTYPE) NB = NTYPE                                        ANALYZE
      NDASH = 37 + 11*(NB-NA)                                            ANALYZE
      NX = (138-NDASH)/2                                                 ANALYZE
      REWIND LTAPE                                                       ANALYZE
      TEST = .FALSE.                                                     ANALYZE
      DO 6 K = 1,KTYPE                                                   ANALYZE
      K1 = K-1                                                           ANALYZE
      IF (K1.NE.50*(K1/50)) GO TO 9                                      ANALYZE
      IF (K1.EQ.0) GO TO 7                                               ANALYZE
      WRITE (6,100) NX, NDASH                                            ANALYZE
      IF (TEST) WRITE (6,101) PCT                                        ANALYZE
    7 WRITE (6,102) KAPTION, NX, (GAS(I), I = NA,NB)                     ANALYZE
      WRITE (6,100) NX, NDASH                                            ANALYZE
      TEST = .FALSE.                                                     ANALYZE
    9 READ (LTAPE) (RPCT(L), L = 1,NTYPE), PCTMAX                        ANALYZE
      R = RATE(K)                                                        ANALYZE
      NFLAG = 1H                                                         ANALYZE
      IF (FLAG(K)) NFLAG = 1H*                                           ANALYZE
      IF (FLAG(K)) TEST = .TRUE.                                         ANALYZE
      DO 10 I = NA,NB                                                    ANALYZE
      I1 = I-NA+1                                                        ANALYZE
      KODE(I1) = 1H                                                      ANALYZE
      IF (RPCT(I).EQ.0.) GO TO 10                                        ANALYZE
      ENCODE (10,104,KODE(I1)) RPCT(I)                                   ANALYZE
   10 CONTINUE                                                           ANALYZE
    6 WRITE (6,103) NX, NFLAG, K, R, PCTMAX, (KODE(I), I = 1,I1)         ANALYZE
      WRITE (6,100) NX, NDASH                                            ANALYZE
      IF (TEST) WRITE (6,101) PCT                                        ANALYZE
      NA = NB+1                                                          ANALYZE
      IF (NA.LE.NTYPE) GO TO 5                                           ANALYZE
C                                                                        ANALYZE
C     ----------------------- FORMAT STATEMENTS ----------------------- ANALYZE
C                                                                        ANALYZE
  100 FORMAT (/=X,=(1H-)/)                                               ANALYZE
C                                                                        ANALYZE
  101 FORMAT (18X#* THIS REACTION CONTRIBUTES LESS THAN #F3.0* % TO ALL  ANALYZE
     1SPECIES THROUGHOUT THE ENTIRE CALCULATION SO FAR*)                 ANALYZE
C                                                                        ANALYZE
  102 FORMAT (1H1,47X,4A10,/29X*PERCENTAGE CONTRIBUTION OF REACTION K TO ANALYZE
     1 DN(I)/DT, EXPRESSED (FOR EACH SPECIES)*/33X*AS A PERCENTAGE OF TH ANALYZE
     2E MAXIMUM RATE OCCURING FOR ALL REACTIONS INCLUDED*//=X,4X*K*4X,   ANALYZE
     3*RATE(K)*4X*MAX %*5X,8(1X,A10),A10,A7)                             ANALYZE
C                                                                        ANALYZE
  103 FORMAT (=X,A1,I4,1PE12.3,0PF8.1,10(1X,A10))                        ANALYZE
C                                                                        ANALYZE
  104 FORMAT (F10.3)                                                     ANALYZE
C                                                                        ANALYZE
C     --------------------------------------------------------------- ANALYZE
C                                                                        ANALYZE
      RETURN                                                             ANALYZE
      END                                                                ANALYZE
```

```
      SUBROUTINE DEKODE (NAME, IMAGE, LHS, RHS, LABEL, GAS, NSIZE,
     1   NTYPE, LONG)
C
      DIMENSION NAME(1), IMAGE(1), GAS(5,2), LABEL(5,2), KAR(10)
      INTEGER LHS, RHS, GAS, E, HNU
      E = 4HE(-)
      HNU = 3HHNU
      NO = NTYPE
      DO 1 L = 1,5
      DO 1 M = 1,2
      GAS(L,M) = 1H
    1 LABEL(L,M) = 0
      M = 1
      I = J = N = MM = 0
    2 IF (I.EQ.LONG) GO TO 4
      I = I+1
      IF (IMAGE(I).EQ.1H ) GO TO 2
      IF (IMAGE(I).NE.1H+) GO TO 3
         MM = 1
         GO TO 4
    3 IF (IMAGE(I).NE.1H+) GO TO 6
      IF (IMAGE(I+1).EQ.1H).OR.IMAGE(I+1).EQ.1H+) GO TO 6
         MM = 0
         GO TO 4
    6 IF (J.EQ.NSIZE) GO TO 2
      J = J+1
      KAR(J) = IMAGE(I)
      GO TO 2
    4 IF (J.EQ.0) GO TO 99
      IF (M.GT.2) GO TO 99
      ENCODE (10,100,NGAS) (KAR(L), L = 1,J)
  100 FORMAT (10A1)
      IF (NGAS.EQ.1HE.OR.NGAS.EQ.2HE-) NGAS = E
      J = 0
      N = N+1
      GAS(N,M) = NGAS
      IF (NGAS.EQ.HNU) GO TO 7
      IF (NO.EQ.0) GO TO 9
      DO 5 L = 1,NO
      IF (NGAS.NE.NAME(L)) GO TO 5
      LABEL(N,M) = L
      GO TO 7
    5 CONTINUE
    9 NO = NO+1
      NAME(NO) = NGAS
      LABEL(N,M) = NO
    7 M = M+MM
      IF (MM.EQ.1) N = 0
      GO TO 2
   99 NTYPE = NO
C
C     GENERATE CHECKSUM IDENTIFIERS --
C
      K1 = K2 = K1SQ = K2SQ = 0
      DO 8 L = 1,5
      K1 = K1 + LABEL(L,1)
      K2 = K2 + LABEL(L,2)
      K1SQ = K1SQ + LABEL(L,1)*LABEL(L,1)
    8 K2SQ = K2SQ + LABEL(L,2)*LABEL(L,2)
      ENCODE (10,110,LHS) K1, K1SQ
      ENCODE (10,110,RHS) K2, K2SQ
  110 FORMAT (I4,I6)
C
      RETURN
      END
```

```
      SUBROUTINE UPDATE (INFILE, NTAPE, NSCRTCH, LIST, DATE)
C
C     ..............................................................
C
C     THIS SUBROUTINE SEARCHES TWO SOURCES--AN INPUT FILE TAPE *INFILE*
C     AND/OR INPUT CARD DATA (IF MODIFY = TRUE)--TO GENERATE AN UPDATED
C     FILE ON TAPE *NTAPE*, WHICH CONTAINS ALL OF THE DATA OF THE FILE
C     *INFILE* MODIFIED WITH ADDITIONS OR REVISIONS DEFINED BY THE CARD
C     DATA.  THE FILE ON TAPE *NTAPE* CAN BE CATALOGUED AS A PERMANENT
C     FILE, IF DESIRED, FOR FUTURE USE AS THE INPUT LIBRARY.  THE FILE
C     GENERATED ON NTAPE CONTAINS DATES OF ENTRY FOR ALL CROSS SECTIONS
C     WHICH HAVE BEEN CATALOGUED.  IF LIST = TRUE, THE CONTENTS OF THE
C     UPDATED CROSS SECTION FILE *NTAPE* ARE PRINTED OUT.
C
C     ..............................................................
C
      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
C
      CALL SECOND (T0)
      NTYPE = 0
      NUMBERS = 10H1234567890
      BLANK = 1H
      REWIND NTAPE
      ENDFILE = .FALSE.
      INPUT = INFILE
C
C     GENERATE OR MODIFY ELECTRON CROSS SECTION DATA FILE --
C
      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
C
C     THE FOLLOWING SECTION IS USED WITH ONLY ONE INPUT DATA SOURCE --
C
   46 READ (NFILE,120) IMAGE
      IF (EOF(NFILE)) 45,4
    4 IF (IMAGE(8).EQ.BLANK) IMAGE(8) = DATE
      WRITE (NTAPE,120) IMAGE
      READ (NFILE,120) IMAGE
      WRITE (NTAPE,120) IMAGE
      NREC = 0
   47 READ (NFILE,120) IMAGE
```

```
      WRITE (NTAPE,120) IMAGE
      IF (IMAGE(1).NE.BLANK) GO TO 14
         IF (NREC.GT.0) GO TO 46
         BACKSPACE NTAPE
         BACKSPACE NTAPE
         BACKSPACE NTAPE
         GO TO 46
   14 NREC = 1
      READ (NFILE,120) IMAGE
      WRITE (NTAPE,120) IMAGE
      GO TO 47
C
C     THE FOLLOWING SECTION OCCURS WHEN DATA IS ASSEMBLED FROM BOTH A
C     TAPE AND CARD INPUT FILE --
C
   33 IF (ENDFILE) GO TO 44
      READ (INPUT,150) KINETIC
      IF (EOF(INPUT)) 6,7
    7 CALL DEKODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60)
      GO TO 37
    6 ENDFILE = .TRUE.
      INPUT = 5
      NFILE = 5
      REWIND NFILE
   44 READ (NFILE,120) IMAGE
      IF (EOF(NFILE)) 45,8
    8 IMAGE(8) = DATE
      BACKSPACE NFILE
      READ (NFILE,150) KINETIC
      CALL DEKODE (NAME, KINETIC, LHS1, RHS1, LABEL, GAS, 10, NTYPE, 60)
C
C     CHECK TAPE3 TO DETERMINE WHETHER THE PROCESS ENCOUNTERED ON CARD
C     INPUT WAS PREVIOUSLY USED TO UPDATE TAPE FILE DATA --
C
      REWIND NSCRTCH
   43 READ (NSCRTCH) LHS2, RHS2
      IF (EOF(NSCRTCH)) 34,5
    5 IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 43
      READ (INPUT,120)
      GO TO 41
   37 NFILE = INPUT
      BACKSPACE NFILE
      READ (NFILE,120) IMAGE
C
C     CHECK CARDS TO SEE IF A CROSS SECTION PRESENTED ON THE TAPE FILE
C     SHOULD BE SUPERCEDED BY CARD INPUT DATA (UPDATE) --
C
      REWIND 5
   36 READ (5,150) KINETIC
      IF (EOF(5)) 34,9
    9 CALL DEKODE (NAME, KINETIC, LHS2, RHS2, LABEL, GAS, 10, NTYPE, 60)
      IF (LHS2.NE.LHS1.OR.RHS2.NE.RHS1) GO TO 35
C
C     THE PROCESS DEFINED ON TAPE FILE HAS BEEN FOUND IN THE CARD INPUT,
C     SO IT IS REPLACED --
C
      NFILE = 5
```

```
      BACKSPACE 5
      READ (5,120) IMAGE
      IMAGE(8) = DATE
      WRITE (NSCRTCH) LHS2, RHS2
      READ (INPUT,120)
      GO TO 34
C
   35 READ (5,120)
   31 READ (5,120) KARD
      IF (KARD.EQ.BLANK) GO TO 36
      READ (5,120)
      GO TO 31
C
C     COPY CROSS SECTION DATA ONTO TAPE NTAPE --
C
   34 WRITE (NTAPE,120) IMAGE
      READ (NFILE,120) IMAGE
      WRITE (NTAPE,120) IMAGE
      NREC = 0
   32 READ (NFILE,120) IMAGE
      WRITE (NTAPE,120) IMAGE
      IF (IMAGE(1).NE.BLANK) GO TO 13
         IF (NREC.GT.0) GO TO 39
         BACKSPACE NTAPE
         BACKSPACE NTAPE
         BACKSPACE NTAPE
         GO TO 39
   13 NREC = 1
      READ (NFILE,120) IMAGE
      WRITE (NTAPE,120) IMAGE
      GO TO 32
   39 IF (NFILE.EQ.INPUT) GO TO 33
C
C     EXHAUST OLD DATA FOR THIS PROCESS --
C
   41 READ (INPUT,120) IMAGE
      IF (IMAGE(1).EQ.BLANK) GO TO 33
      READ (INPUT,120)
      GO TO 41
C
C     IF ELECTRON CROSS SECTION DATA CONTAINED MODIFICATIONS BY CARD
C     INPUT, A NEW FILE IS GENERATED.  THE CONTENTS OF THE UPDATED FILE
C     (WHICH MAY BE CATALOGUED FOR FUTURE USE) ARE COPIED ONTO OUTPUT
C     IF LIST IS SPECIFIED TO BE TRUE.
C
   45 ENDFILE NTAPE
      REWIND NSCRTCH
      CALL SECOND (TIME)
      TIME = TIME-TO
      IF (MODIFY) WRITE (6,160) TIME
      IF (.NOT.LIST) GO TO 99
      IF (NFILE.EQ.INFILE) GO TO 99
      REWIND NTAPE
      LL = LINE = 0
   23 READ (NTAPE,120) IMAGE
      IF (EOF(NTAPE)) 11,12
   12 IF (LINE.NE.0) GO TO 25
```

```
      IF (LL.EQ.0) GO TO 26                                             UPDATE
      WRITE (6,190) (NUMBERS, I = 1,8), (I, I = 1,8)                    UPDATE
      WRITE (6,180)                                                     UPDATE
   26 WRITE (6,170) NTAPE, (I, I = 1,8), (NUMBERS, I = 1,8)             UPDATE
   25 LINE = LINE+1                                                     UPDATE
      LL = LL+1                                                         UPDATE
      WRITE (6,140) LL, IMAGE                                           UPDATE
      IF (LINE.EQ.40) LINE = 0                                          UPDATE
      GO TO 23                                                          UPDATE
   11 WRITE (6,190) (NUMBERS, I = 1,8), (I, I = 1,8)                    UPDATE
C                                                                       UPDATE
C  -------------------  FORMAT STATEMENTS  -------------------------    UPDATE
C                                                                       UPDATE
  100 FORMAT (A10)                                                      UPDATE
C                                                                       UPDATE
  120 FORMAT (8A10)                                                     UPDATE
C                                                                       UPDATE
  140 FORMAT (20X,I5,5X,8A10)                                           UPDATE
C                                                                       UPDATE
  150 FORMAT (80A1)                                                     UPDATE
C                                                                       UPDATE
  160 FORMAT (///33X*ELECTRON CROSS SECTION FILE WAS UPDATED.  TIME REQU UPDATE
     1IRED WAS*F5.1* CP SEC.*/33X*CONSULT DAY FILE TO DETERMINE WHETHER UPDATE
     2THE UPDATED FILE WAS RECATALOGUED.*)                              UPDATE
C                                                                       UPDATE
  170 FORMAT (1H1,49X,*ELECTRON CROSS SECTION DATA ON TAPE*,I2///30X,   UPDATE
     18I10/20X,*  LINE*4X,8A10/20X,90(1H-)/)                            UPDATE
C                                                                       UPDATE
  180 FORMAT (/20X*CONTINUED*)                                          UPDATE
C                                                                       UPDATE
  190 FORMAT (/20X,90(1H-)/30X,8A10/30X,8I10)                           UPDATE
C                                                                       UPDATE
C  ---------------------------------------------------------------     UPDATE
C                                                                       UPDATE
   99 REWIND NTAPE                                                      UPDATE
      END                                                               UPDATE
```

```
      SUBROUTINE PLASMA (NDATA, MAX, MESH, LHS, RHS, PROCESS, EV, F,
     1 G, Q, U0, UM, NTYPE, NAME, MISSING, ERROR, OUTSIDE, IDEG, OUT)
C
C     ------------------------------------------------------------------
C
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
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 = [80/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
C     INPUT PARAMETERS --
C
C     NDATA    =  LOGICAL FILE FOR ELECTRON CROSS SECTION DATA.
C
C     MAX      =  DIMENSION DECLARATOR DEFINED FOR EV(I), F(I), G(I),
C                 AND Q(I) IN THE CALLING PROGRAM.
C
C     MESH     =  NUMBER OF SUBINTERVALS INTO WHICH THE ELECTRON ENERGY
C                 RANGE IS DIVIDED (AND OVER WHICH THE CROSS SECTION DATA
C                 IS TO BE INTERPOLATED).  MESH+1 ≤ MAX.
C
C     PROCESS  =  VECTOR (4A10) CONTAINING HOLLERITH NAME OF REACTION.
C
C     LHS,RHS  =  (INTEGERS) ENCODED WITH UNIQUE IDENTIFIERS OF THE LEFT
C                 AND RIGHT HAND SIDE OF THE REACTION (CF. SUBROUTINE
C                 DEKODE).
C
C     F, G     =  SCRATCH VECTORS (DIMENSIONED MAX IN CALLING PROGRAM.)
C
C
C     OUTPUT PARAMETERS --
C
C     Q(I)     =  CROSS SECTION VECTOR (UNITS OF CM2) DEFINED AT EV(I)
C
C     U0       =  CROSS SECTION THRESHOLD ENERGY (EV).
```

```
C
C     UM       =  CROSS SECTION MAXIMUM ENERGY CUTOFF (EV).
C
C     MISSING =   .TRUE., IF THE REACTION WAS NOT FOUND.
C
C     OUTSIDE =   .TRUE., IF THE CROSS SECTIONS WERE DEFINED OVER AN
C                 ENERGY RANGE [U0,UM] THAT DOES NOT SPAN THE ENERGY GRID
C                 DEFINED BY THE INPUT ENERGY VECTOR EV(I).
C
C     ERROR    =  .TRUE., IF ENERGY VALUES WERE NOT SEQUENCED IN MONO-
C                 TONICALLY ASCENDING ORDER (ONLY DATA IN ASCENDING ORDER
C                 IS PERMITTED.)
C
C  ---------------------------------------------------------------------
C
      DIMENSION EV(1), F(1), G(1), Q(1), NAME(1), IMAGE(60), KOMMENT(6),
     1 PROCESS(4), Y(10), LABEL(5,2), GAS(5,2)
C
      LOGICAL FORWARD, REVERSE, ERROR, THRESH, OUT(2), MISSING, OUTSIDE
      INTEGER LHS, RHS, GAS
C
      UNIT = 1.E-16
      NPAGE = 5H(1H1)
      ERROR = OUTSIDE = .FALSE.
      MESHP1 = MESH+1
      DO 17 L = 1,MESHP1
   17 Q(L) = 0.
C
      REWIND NDATA
   20 READ (NDATA,100) IMAGE, UNITS, NPTS, MONTH
  100 FORMAT (60A1,F7.3,I3,A10)
      IF (EOF(NDATA)) 99,1
    1 CALL DEKODE (NAME, IMAGE, L1, L2, LABEL, GAS, 10, NTYPE, 60)
      FORWARD = LHS.EQ.L1.AND.RHS.EQ.L2
      REVERSE = LHS.EQ.L2.AND.RHS.EQ.L1
      MISSING = .NOT.(FORWARD.OR.REVERSE)
         IF ((NPTS.LE.0).OR.(NPTS.GT.10)) NPTS = 10
         INV = 80/NPTS
         ENCODE (10,101,FORM) NPTS, INV
  101 FORMAT (1H(,I2,1HE,I2,3H.0))
      IF (UNITS.EQ.0.) UNITS = 1.
      IF (MISSING) GO TO 3
      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.
      DO 4 L = 1,MAX
    4 F(L) = G(L) = 0.
      LPTS = LTH = 1
      F0 = U0 = 0.
      X0 = - 1.0
      LAST = MAX
C
    3 READ (NDATA,112) KOMMENT
  112 FORMAT (8A10)
C
      ISW = 1
      L = 0
```

```
    8 L2 = L
    7 ISW = - ISW
      READ (NDATA,FORM) (Y(N), N = 1,NPTS)
      SUM = 0.
      DO 16 N = 1,NPTS
   16 SUM = SUM + Y(N)
C
C     BLANK CARD TERMINATES (U, SIGMA) DATA PACKAGE FOR THE INELASTIC
C     PROCESSES NK.
C
      IF ((SUM.EQ.0.).AND.(ISW.LT.0)) GO TO 10
      IF (MISSING) GO TO 7
      IF (L2.EQ.LAST) GO TO 7
      IF (ISW.GT.0) GO TO 9
C
      L = L2
      DO 12 N = 1,NPTS
      IF (L.EQ.LAST) GO TO 7
      IF (Y(N).GT.X0) GO TO 11
      IF (Y(N).NE.0.) ERROR = .TRUE.
      LAST = L
      GO TO 7
   11 L = L+1
   12 X0 = F(L) = Y(N)
      GO TO 7
C
    9 L = L2
      DO 13 N = 1,NPTS
      IF (L.EQ.LAST) GO TO 8
      L = L+1
      G(L) = Y(N)*UNITS
      IF (G(L).GT.0.) THRESH = .TRUE.
      IF (THRESH) GO TO 5
         U0 = F(L)
         LTH = L
    5 IF (F0.EQ.0.) GO TO 13
         UM = F(L)
         LPTS = L
   13 F0 = G(L)
      GO TO 8
C
   10 IF (MISSING) GO TO 20
      LPTS = LPTS - LTH + 1
      ERROR = ERROR.OR.LPTS.LE.1
      IF (ERROR) GO TO 6
      DO 14 L = 1,LPTS
      LL = L + LTH - 1
      G(L) = G(LL)
   14 F(L) = F(LL)
C
      OUTSIDE = .TRUE.
      DO 15 L = 1,MESHP1
      SIGMA = 0.
      X = EV(L)
      IF (X.GT.UM) GO TO 15
      IF (X.LT.U0) GO TO 15
      OUTSIDE = .FALSE.
```

```
      CALL INTERP (IDEG, X, SIGMA, F, G, 1, LPTS)
   15 IF (SIGMA.GE.0.) Q(L) = SIGMA*UNIT
C
    6 IF (.NOT.OUT(1)) GO TO 18
      WRITE (6,102) UNIT, PROCESS
  102 FORMAT (1H1,34X,*CROSS SECTION (UNITS OF*, 1PE10.3* CM2) VS ELECTR
     1ON ENERGY (EV) FOR*/57X,4A10//)
      IF (KOMMENT(1).NE.1H ) WRITE (6,103) KOMMENT
  103 FORMAT (35X,*REFERENCE -- *,6A10//)
      WRITE (6,104) UNIT
  104 FORMAT (56X*U*15X*SIGMA(U)*/54X*(EV) *10X*(*1PE7.1* CM2)*)
      WRITE (6,105)
  105 FORMAT (/50X,33(1H-)/)
      LL = 0
      LINE = 1
      DO 33 L = 1,LPTS
      IF (LL.NE.LINE) GO TO 32
      WRITE (6,NPAGE)
      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, TFALL, 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
      F0 = X*(1. + 1./X)**(1. + X)
      IF (TP.GT.TDROP) GO TO 1
      SHAPE = F0*(1. - EXP(-TP/TR))*EXP(-TP/TF)
      RETURN
    1 TP = TP-TDROP
      F0 = F0*(1. - EXP(-TDROP/TR))*EXP(-TDROP/TF)
      SHAPE = F0*EXP(-TP/TFALL)
      RETURN
    2 CALL INTERP (2, TP, SHAPE, TIME, Y, 1, N)
      IF (SHAPE.LT.0.) SHAPE = 0.
      RETURN
      END
```

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

```
C
C     N1(J)      =  NEUTRAL PARTICLE DENSITY (CM-3) OF THE LOWER STATE
C                   IN THE JTH INELASTIC COLLISION PROCESS.
C
C     N2(J)      =  NEUTRAL PARTICLE DENSITY (CM-3) OF THE UPPER
C                   STATE IN THE JTH INELASTIC COLLISION PROCESS.
C
C     NEL(J)     =  NET NUMBER OF ELECTRONS (RIGHT HAND SIDE - LEFT
C                   HAND SIDE) FOR THE JTH ELECTRON COLLISION PROCESS.
C
C     NMOL       =  TOTAL NEUTRAL PARTICLE DENSITY (CM-3).
C
C     NE         =  ELECTRON DENSITY (CM-3).
C
C     DNEDT      =  INPUT ESTIMATE OF D/DT(NE), THE RATE OF CHANGE OF
C                   THE SECONDARY ELECTRON DENSITY (CM-3/SEC).  THE
C                   BOLTZMANN ANALYSIS DETERMINES D/DT(NE) SELF-CON-
C                   SISTENTLY IN AN ITERATIVE LOOP, AND EXECUTION MAY
C                   BE OPTIMIZED BY A GOOD INITIAL ESTIMATE OF DNEDT.
C                   UPON OUTPUT, DNEDT IS THE ACTUAL (SELF-CONSISTENT)
C                   VALUE OF D/DT(NE).
C
C     SOURCE     =  TOTAL RATE OF EXTERNAL SOURCE CREATION OF SECON-
C                   DARY ELECTRONS AT U = 0 (CM-3/SEC).
C
C     SBEAM      =  TOTAL RATE OF E-BEAM CREATION OF SECONDARY ELEC-
C                   TRONS (CM-3/SEC).
C
C     S(I)       =  NORMALIZED E-BEAM SOURCE FUNCTION: SBEAM*S(I) =
C                   RATE OF GENERATION OF ELECTRONS (/SEC/CM3/EV) IN
C                   THE ENERGY RANGE [U,U+DU], WHERE U = X(I).  THE
C                   TOTAL EXTERNAL POWER DEPOSITION IS: DEPOSIT =
C                   SBEAM*INT[DU U S(U)] (EV/SEC/CM3).
C
C     GAS(K)     =  NAME OF THE KTH GAS IN THE MIXTURE.
C
C     FRACT(K)   =  RELATIVE FRACTION OF THE KTH GAS IN THE MIXTURE.
C                   SUMK[FRACT(K)] NEED NOT EQUAL 1; THE PROGRAM WILL
C                   AUTOMATICALLY NORMALIZE THE CONCENTRATIONS, AFTER
C                   SORTING THEM INTO DESCENDING NUMERICAL ORDER.
C
C     PROCESS    =  ARRAY (DIMENSIONED (4,1)) CONTAINING 4 WORDS PER
C                   COLUMN TO PROVIDE A 40-BCD CHARACTER SPECIFICATION
C                   FOR EACH OF THE INELASTIC E- COLLISION PROCESSES.
C
C     MIX        =  NUMBER OF SPECIES IN THE MIXTURE.
C
C     EVCM       =  ELECTRIC FIELD (V/CM).
C
C     TMOL       =  MOLECULAR TEMPERATURE (DEG K).
C
C     A(I,J)     =  SCRATCH ARRAY, DIMENSIONED A(MAX,3) IN THE CALLING
C                   PROGRAM, USED FOR STORING CERTAIN COEFFICIENTS
C                   THAT DEFINE THE FINITE DIFFERENCE FORM OF THE
C                   BOLTZMANN EQUATION (A(I,1), A(I,2), AND A(I,3) ARE
C                   USED TO STORE THE THREE MAJOR DIAGONALS DURING THE
C                   SOLUTION).  LATER, A(I,J) IS USED AS A SCRATCH
```

```
C                       ARRAY FOR CALCULATING NUMEROUS PLASMA PARAMETERS.
C
C        B(I)        =  SCRATCH VECTOR, DIMENSIONED B(MAX) IN THE CALLING
C                       PROGRAM, USED FOR THE RIGHT HAND SIDE IN THE
C                       ITERATIVE SOLUTION OF THE BOLTZMANN EQUATION.
C
C        EPS         =  MAXIMUM CHANGE BETWEEN SUCCESSIVE ITERATIONS FOR
C                       DEFINITION OF CONVERGENCE FOR THE ELECTRON ENERGY
C                       DISTRIBUTION FUNCTION.  CONVERGENCE IS DECLARED IF
C
C                       MAX(I)[F(I(N)-F(I(N-1)]/F(I,N-1) < EPS
C
C        ITMAX       =  MAXIMUM NUMBER OF ITERATIONS ALLOWED TO SOLVE THE
C                       BOLTZMANN EQUATION.  (IF ITMAX ≤, PROGRAM SETS
C                       CONVRGE = TRUE, AND IMMEDIATELY TRANSFERS CONTROL
C                       TO CALCULATION OF PLASMA PARAMETERS FOR THE INPUT
C                       DISTRIBUTION FUNCTION F.
C
C        TMAX        =  MAXIMUM CP TIME ALLOWED FOR ATTAINMENT OF CONVER-
C                       GENCE OF THE ELECTRON ENERGY DISTRIBUTION FUNCTION
C
C        OUT(I)      =  LOGICAL VECTOR OF OUTPUT REQUESTS (I = 1,2,...5):
C
C                       1)  TABLE OF PLASMA PARAMETERS
C                       2)  TABLE OF ELECTRON ENERGY DISTRIBUTION
C                       3)  LOGPLOT OF ENERGY DISTRIBUTION F(U)
C                       4)  LOGPLOT OF RELATIVE FUNCTION F(U)/FB(U,TE)
C                           (FB(U,TE) = BOLTZMANN DISTRIBUTION AT TE.)
C                       5)  TABLE OF EXCITATION RATES AND ELECTRICAL
C                           POWER TRANSFER FOR ALL E(-) PROCESSES.
C
C        KAPTION(4)  =  VECTOR CONTAINING A 40 BCD CHARACTER TITLE
C
C  ------------------------------------------------------------------
C
C  OUTPUT PARAMETERS --
C
C        VSIG(K,J)   =  FORWARD (K = 1) AND REVERSE (K = 2) ELECTRON EX-
C                       CITATION RATES (CM3/SEC) FOR THE JTH INELASTIC
C                       COLLISION PROCESS.
C
C                       VSIG(1,J)  =  <V(U)Q(U, J)>
C                       VSIG(2,J)  =  <V(U)Q(U,-J)>
C
C                       WHERE Q(U,-J) IS THE CROSS SECTION FOR THE REVERSE
C                       PROCESS.
C
C        POWER(J)    =  NET POWER (ACCOUNTING FOR FORWARD AND REVERSE COL-
C                       LISION PROCESSES) PARTITIONED INTO THE JTH INELAS-
C                       TIC PROCESS. (WATT/ELECTRON)
C
C        VD          =  ELECTRON DRIFT VELOCITY (CM/S).
C
C        MU          =  ELECTRON MOBILITY (CM2/V/SEC).
C
C        D           =  ELECTRON DIFFUSION COEFFICIENT (CM2/SEC).
C
```

```
C        EK          =  CHARACTERISTIC ELECTRON ENERGY = D/MU (EV).
C
C        UBAR        =  AVERAGE ELECTRON ENERGY (EV).
C
C        TE          =  EFFECTIVE ELECTRON TEMPERATURE (DEG K). DEFINED BY
C                       (3/2)KTE  =  UBAR
C
C        DISCH       =  E*NE*MU*EVCM**2. TOTAL ELECTRICAL DISCHARGE POWER
C                       DENSITY (W/CM3).
C
C        DEPOSIT     =  POWER DENSITY FROM EXTERNAL DEPOSITION (INTO ELEC-
C                       TRON KINETIC ENERGY) (W/CM3).
C
C        PCOLL       =  ELECTRICAL POWER DENSITY INTO INELASTIC COLLISIONS
C                       (W/CM3).
C
C        ELASTIC     =  ELECTRICAL POWER DENSITY INTO ELASTIC HEATING OF
C                       THE MOLECULAR GAS (W/CM3).
C
C        DEDT        =  UBAR*DNE/DT = RATE OF CHANGE OF ENERGY DENSITY
C                       STORED IN THE ELECTRON GAS (W/CM3).
C
C        NOTE:          IF NE = 0. THE FIVE POWER DENSITIES (DISCH, DE-
C                       POSIT, PCOLL, ELASTIC, AND DEDT) ARE EVALUATED FOR
C                       UNIT ELECTRON DENSITY (I.E. AS IF NE = 1.0 CM-3),
C                       AND THUS, THE EFFECTIVE UNITS ARE W/ELECTRON.
C
C        DNEDT       =  D/DT(NE), RATE OF CHANGE OF SECONDARY ELECTRON
C                       DENSITY (CM-3/SEC).
C
C        IONIZE      =  TOTAL IONIZATION FREQUENCY (SEC-1).
C
C        ATTACH      =  TOTAL FREQUENCY FOR ATTACHMENT AND RECOMBINATION.
C
C        DLNEDT      =  (1/NE)*D/DT(NE), LOGARITHMIC DERIVATIVE OF NE.
C
C        AMPS        =  E*VD = ELECTRICAL CURRENT DENSITY PER UNIT ELEC-
C                       TRON DENSITY (A CM).
C
C        F(I)        =  NORMALIZED ELECTRON ENERGY DISTRIBUTION FUNCTION
C                       (UNITS OF EV**(-3/2).
C
C        G(I)        =  F(I)/F(1)
C
C        B(I)        =  F(I)/FBOLTZ(TE,I)
C
C        CONVRGE     =  LOGICAL VARIABLE WHICH SPECIFIES SUCCESSFUL CON-
C                       VERGENCE OF THE ELECTRON DISTRIBUTION FUNCTION
C                       CALCULATIONS.
C
C -------------------------------------------------------------------------
C
      DIMENSION XQ(MAX,1), QM(MAX,1), X(1), F(1), G(1), VSIG(2,1),
     1 U(1), N1(1), N2(1), NEL(1), POWER(1), A(MAX,3), PROCESS(4,1),
     2 B(1), OUT(1), GAS(1), FRACT(1), NAME(5), FI(5), FORM(15),
     3 KAPTION(4), NOUT(6), S(1), PCT(6)
C
```

```
      REAL NMOL, MU, ME, KT, KB, N1, N2, KTE, NU, NE, LAMBDA, IONIZE,
     1 LOG, KVCMATM
      LOGICAL CONVRGE, TEST, ERROR, OUT
      INTEGER DATE, SKIP, GAS
C
      DATA E, PI, KB, ME / 1.602E-12, 3.1415927, 1.38E-16, 9.31E-28 /
      DATA NMAX, MASK / 5, 77770000000000000000B /
      CALL SECOND(T0)
C
C     SORT SPECIES IN DESCENDING ORDER OF CONCENTRATION --
      SUM = 0.
      DO 37 N = 1,MIX
   37 SUM = SUM + FRACT(N)
      NAME(1) = NAME(2) = NAME(3) = NAME(4) = NAME(5) = 1H
      NGAS = 0
   38 FMAX = 0.
      DO 2 N = 1,MIX
         DO 33 K = 1,NGAS
         IF (GAS(N).EQ.NAME(K)) GO TO 2
   33    CONTINUE
         IF (FRACT(N).LE.FMAX) GO TO 2
         FMAX = FRACT(N)
         N0 = N
    2 CONTINUE
      IF (FMAX.LT.1.0E-04*SUM) GO TO 45
      NGAS = NGAS+1
      NAME(NGAS) = GAS(N0)
      FI(NGAS) = FRACT(N0)
      IF (NGAS.EQ.NMAX) GO TO 45
      GO TO 38
   45 PTOT = NMOL*TMOL/0.965E 19
      ATM = PTOT/760.
      DO 36 N = 1,NGAS
   36 FI(N) = FI(N)/SUM
C
      FE = NE/NMOL
      ESU = 300.*E
      ESQ = ESU*ESU
      IF (TE.LE.0.) TE = TMOL
      KTE = KB*TE
      UBAR = 1.5*KTE/E
      KT = KB*TMOL/E
      CONST = (2.*PI/3.)*NE*ESQ*9.0E 04
      DX = X(2)-X(1)
      EM = SQRT(2.*E/ME)
      M = MESH + 1
      MPTS = MESH/2
      EMAX = X(M)
      C0 = E*1.0E-07
      IF (NE.NE.0.) C0 = NE*C0
      KAPT = 10HWATT/ELECT
      IF (NE.NE.0.) KAPT = 10HWATT/CM3
      EN = EVCM/NMOL
      ESQ3 = EVCM*EVCM/3.
      EP = EVCM/PTOT
      KVCMATM = .760*EP
      IONIZE = ATTACH = 0.
```

```
      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,1) = 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
      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
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-DIAGONAL ELEMENTS OF THE IN-
C     ELASTIC SCATTERING TERM, ALSO EVALUATED USING THE PREVIOUS DISTRI-
C     BUTION FUNCTION.
C
C     THE BOLTZMANN EQUATION IS WRITTEN
C
```

```
C      1/(SQRT(2E/M)*NE)*(SQRT(U)F(U)DNE/DT + DJF(U)/DU + DJEL(U)/DU
C        + DJEE(U)/DU - SBEAM*S(U)) = ((NU(ION) + SOURCE/NE)*DELTA(U)
C        + SUMK[U*F(U)*NK*QK(U)]],
C
C      WHERE
C
C         J(F)   =  -(1/3)NE*SQRT(2E/M)*ESQ*U DF/DU /<N QM(U)>
C         J(EL)  =  - NE*SQRT(2E/M)*USQ<(2ME/M)NQM(U)>*[F + (KT/E)DF/DU]
C         J(EE)  =  - (2PI/3)*Q**4/E**2*LN(LAMBDA)*NE*NE*SQRT(2E/M)*
C                          [PINT(U)*DF/DU + QINT(U)*F(U)]
C
C      WHERE
C
C         PINT(U)  =  2*INT[DU*U**1.5*F(U)] + 2*U**1.5*INT[DU*F(U)]
C         QINT(U)  =  3*INT[DU*SQRT(U)*F(U)]
C         DEBYE    =  SQRT(KTE/(4*PI*NE*NE))
C         LAMBDA   =  DEBYE/RMIN
C         RMIN     =  ESU**2/(E*UBAR)
C
C      DEFINE QUANTITIES RELATED TO THE DRIVING FIELD AND ELASTIC COLLI-
C      SIONS --
C
C      P(U)  =  (E**2/3)*U/<N QMOM(U)> + (KT/E)*U**2 <(2ME/M)N QMOM(U)>
C      Q(U)  =  U**2 <(IME/M)N QMOM(U)>
C
C         P(I+.5)  =  (E**2/3)*QM(I,1) + (KT/E)*QM(I,2)
C         Q(I+.5)  =  QM(I,2)
C
C      DEFINE TERMS RELATED TO ELECTRON-ELECTRON SCATTERING --
C
C      ALPHA(U)  =  (2PI/3)NE(Q**4/E**2)LOG(LAMBDA)*
C                                       [PINT(U)/DU + QINT(U)/2]
C      BETA(U)   =  (2PI/3)NE(Q**4/E**2)LOG(LAMBDA)*
C                                       [PINT(U)/DU - QINT(U)/2]
C
C         A(I,1)  =  P(I-.5)/DU  -  Q(I-.5)/2  +  BETA(I-.5)
C         A(I,2)  =  -[P(I+.5) + P(I-.5)]/DU  +  [Q(I+.5) - Q(I-.5)]/2
C                      -DU*SUMK[NK U QK(U) + NK*(U + UK)QK(U + UK)]
C                   - ALPHA(I-.5) - BETA(I+.5) - DU*SQRT(U M/2E)DNEDT/NE
C                     - ALPHA(I-.5)  -  BETA(I+.5)
C         A(I,3)  =  P(I+.5)/DU  +  Q(I+.5)/2  +  ALPHA(I+.5)
C
C         B(I)    =  -DU*SUMK[NK(U+UK)QK(U+UK)F(U+UK) +NK*UQK(U)F(U-UK)]
C                      - SQRT(M/2E)[DU*S(U)/NE + NU(ION)DELTA(U)]
C
C      THE REDUCTION OF THE BOLTZMANN EQUATION TO A TRIDIAGONAL FINITE
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)
C
C      I = 2,3,...,M.  THE SET IS COMPLETED BY THE CONDITION THAT F(1)
C      CAN BE APPROXIMATED AS (E.G.) F(1) = F(2).
C
C      IN GENERAL, DIRECT INVERSION OF THE LINEAR SYSTEM IS NOT PRACTICAL
C      EITHER FROM THE STANDPOINT OF CORE STORAGE OR MATRIX INVERSION.
C      INSTEAD, ADVANTAGE IS TAKEN OF THE FACT THAT A IS SPARSE (I.E. ALL
C      ELEMENTS EXCEPT FOR THREE DIAGONALS ARE ZERO), AND SOLUTION IS
```

```
C     ATTEMPTED BY AN ITERATIVE APPROACH:    A[F(N)]*F(N+1) = B[F(N)].      BOLTZ
C                                                                           BOLTZ
C     --------------------------------------------------------------------- BOLTZ
C                                                                           BOLTZ
   40 TRANS = DLNEDT/EM                                                     BOLTZ
C     THE FIRST EQUATION IS F(1) - F(2) = 0 --                              BOLTZ
      A(1,1) = 0.                                                           BOLTZ
      A(1,2) = 1.0                                                          BOLTZ
      A(1,3) = - 1.0                                                        BOLTZ
      Q1 = QM(1,2)/2.                                                       BOLTZ
      P1 = (KT*QM(1,2) + ESQ3*QM(1,1))/DX                                   BOLTZ
      DO 1 I = 2,M                                                          BOLTZ
      Q2 = QM(I,2)/2.                                                       BOLTZ
      P2 = (KT*QM(I,2) + ESQ3*QM(I,1))/DX                                   BOLTZ
      DIAG = 0.                                                             BOLTZ
      DO 4 J = 1,NK                                                         BOLTZ
      DIAG = DIAG + N1(J)*XQ(I,J)                                           BOLTZ
      JU = UJ = U(J)/DX                                                     BOLTZ
      IJ = I+JU                                                             BOLTZ
      IF (IJ.GT.M) GO TO 4                                                  BOLTZ
      IF (IJ.LT.1) GO TO 4                                                  BOLTZ
      DIAG = DIAG + N2(J)*XQ(IJ,J)                                          BOLTZ
    4 CONTINUE                                                              BOLTZ
      ROOT = SQRT(X(I))                                                     BOLTZ
      A(I,1) = P1 - Q1                                                      BOLTZ
      A(I,2) = - P2 - P1 + Q2 - Q1 - (DIAG + ROOT*TRANS)*DX                 BOLTZ
      A(I,3) = P2 + Q2                                                      BOLTZ
      P1 = P2                                                               BOLTZ
    1 Q1 = Q2                                                               BOLTZ
C                                                                           BOLTZ
      IF (NE.EQ.0.) GO TO 10                                                BOLTZ
C     INTEGRAL OF F(U) (FOR E-E SCATTERING) --                              BOLTZ
      CALL SIMPSON (F, MESH/2, 1., FINT)                                    BOLTZ
C                                                                           BOLTZ
C     --------------------------------------------------------------------- BOLTZ
C                                                                           BOLTZ
C     CALCULATE THE RHS VECTOR B, WHICH CONTAINS ALL OF THE OFF-DIAGONAL    BOLTZ
C     PARTS OF THE INELASTIC COLLISION TERMS, DEFINED IN TERMS OF THE       BOLTZ
C     PRESENT DISTRIBUTION VECTOR F(N):                                     BOLTZ
C                                                                           BOLTZ
   10 IF (ITER.EQ.ITMAX) GO TO 98                                           BOLTZ
      CALL SECOND (TIME)                                                    BOLTZ
      TIME = TIME-T0                                                        BOLTZ
      IF (TIME.GT.TMAX) GO TO 98                                            BOLTZ
      ITER = ITER+1                                                         BOLTZ
      B(1) = 0.                                                             BOLTZ
      DO 5 I = 2,M                                                          BOLTZ
      OFF = 0.                                                              BOLTZ
      DO 8 J = 1,NK                                                         BOLTZ
      JU = U(J)/DX                                                          BOLTZ
      IF (NEL(J).LT.0) GO TO 8                                              BOLTZ
C                                                                           BOLTZ
C            COLLISIONS (OF THE FIRST KIND) IN WHICH ELECTRONS ARE          BOLTZ
C            SCATTERED INTO ENERGY X(I), LOSING ENERGY U(J) --              BOLTZ
             IJ = I+JU                                                      BOLTZ
             IF (IJ.GT.M) GO TO 6                                           BOLTZ
             IF (IJ.LT.1) GO TO 6                                           BOLTZ
```

```
          OFF = OFF + XQ(IJ,J)*F(IJ)*N1(J)
C
    6 IF (NEL(J).GT.0) GO TO 8
C
C         COLLISIONS (OF THE SECOND KIND) IN WHICH ELECTRONS ARE
C         SCATTERED INTO ENERGY X(I), GAINING ENERGY U(J) --
          IJ = I-JU
          IF (IJ.LT.1) GO TO 8
          IF (IJ.GT.M) GO TO 8
          OFF = OFF + XQ(I,J)*F(IJ)*N2(J)
C
    8 CONTINUE
    5 B(I) = - (C*S(I) + OFF)*DX
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) --
      B(2) = B(2) - (IONIZE + SO)/EM
C
C ----------------------------------------------------------------------
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.
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) --
C
      G(1) = A(1,3)/A(1,2)
      B(1) = B(1)/A(1,2)
      F1 = -F(1)/2.
      F2 = F3 = 0.
      ELECT = PNEW = 0.
      A2 = B2 = 0.
      XHALF = DX/2.
      IF (NE.EQ.0.) GO TO 35
C     DEBYE LENGTH FOR PLASMA --
      DEBYE = SQRT(KTE/(4.*PI*NE*ESQ))
C     CLASSICAL DISTANCE OF CLOSEST APPROACH --
      RMIN = 300.*ESU/UBAR
      LAMBDA = DEBYE/RMIN
      LOG = ALOG(LAMBDA)
   35 DO 26 I = 1,M
      A1 = A2
      B1 = B2
      IF (NE.EQ.0.) GO TO 20
      XROOT = SQRT(X(I))
      X32 = X(I)*XROOT
      F1 = F1 + F(I)
      F2 = F2 + XROOT*F(I)
      F3 = F3 + X32*F(I)
      X32 = XHALF**1.5
      XHALF = XHALF + DX
```

```
C     PU = PINT(U)/DU, AND QU = QINT(U)/2.
      PU = F3 + X32*(FINT-F1)
      PU = PU + PU
      QU = 1.5*F2*DX
      A2 = CONST*(PU + QU)*LOG
      B2 = CONST*(PU - QU)*LOG
      DF = FAVG2 = 0.
      IF (I.LT.M) DF = F(I+1) - F(I)
      IF (I.LT.M) FAVG2 = F(I+1) + F(I)
      POLD = PNEW
      PNEW = (PU*DF + QU*FAVG2)*LOG
      ELECT = ELECT + 0.5*(POLD + PNEW)
   20 IF (I.EQ.1) GO TO 26
      B(I) = B(I) - (A(I,1) + B1)*B(I-1)
      G(I) = A(I,3) + A2
      Q = (A(I,2) - A1 - B2) - (A(I,1) + B1)*G(I-1)
      B(I) = B(I)/Q
      G(I) = G(I)/Q
   26 CONTINUE
C
C     UPWARD BACK SUBSTITUTION, WITH NEW SOLUTION INTO G --
C
      G(M) = B(M)
      DO 3 J = 1,MESH
      I = M-J
    3 G(I) = B(I) - G(I)*G(I+1)
C
C ---------------------------------------------------------------
C
C     RENORMALIZE THE NEW DISTRIBUTION FUNCTION, AND TEST FOR
C     CONVERGENCE --
C
      DO 25 I = 1,M
   25 B(I) = SQRT(X(I))*G(I)
C
      CALL SIMPSON (B, MESH/2, DX, FNORM)
      CALL SIMPSON (G, MESH/2, 1.0, GINT)
      FINT = GINT/FNORM
C
      DELTA = 0.
      ERROR = .FALSE.
      CONVRGE = .TRUE.
      DO 11 I = 1,M
      FOLD = F(I)
      F(I) = G(I)/FNORM
      FNEW = F(I)
      B(I) = B(I)*X(I)
      F1 = ABS(FOLD)
      F2 = ABS(FNEW)
      FMAX = F1
      IF (F2.GT.F1) FMAX = F2
      ERROR = ERROR.OR.(FNEW.LT.0.)
      ETA = 0.
      IF ((FMAX.AND.MASK).NE.0) ETA = ABS(FNEW-FOLD)/FMAX
      IF (ETA.GT.DELTA) DELTA = ETA
      TEST = ETA.LE.EPSILON
   11 CONVRGE = CONVRGE.AND.TEST
```

```
      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
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 DLNEDTO = 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(1,1), MPTS, DX, VSIG(1,J))
      CALL SIMPSON (A(1,2), 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(1,3), 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 = SEXT + IONIZE - ATTACH
      IF (ITMAX.LE.0) GO TO 60
C
```

```
C     D/DT(NE) IS DETERMINED SELF-CONSISTENTLY BY AN ITERATIVE LOOP.  IF
C     THE CHANGE IN THE (LOGARITHMIC) DERIVATIVE OF NE BECOMES ≤ 2 %, NO
C     FURTHER ITERATION IS REQUIRED --
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 --
            EPSILON = EPS
            GO TO 40
C
C     DETERMINE CP TIME REQUIRED FOR OBTAINING F(U) --
   60 CALL SECOND (TIME)
      TIME = TIME-T0
C
C ----------------------------------------------------------------------
C
      XBAR = -DX/2.
      DO 9 I = 1,M
      G(I) = F(I)/F(1)
      B(I) = B(I)*X(I)
      CALL INTERP (1, XBAR, QMAVG, X, QM(1,1), 1, M)
      A(I,1) = F(I)*X(I)*X(I)/QMAVG
      IF (F(I).GT.1.E-20*F(1)) MP = I
    9 XBAR = XBAR + DX
C
C     F HAS BEEN NORMALIZED TO INT[DU*SQRT(U)*F(U)] = 1, AND G(I) =
C     F(I)/F(1).
C
C     ELASTIC COLLISION FREQUENCY NU:
C
      CALL SIMPSON (A, MPTS, DX, NU)
            NU = EM*NU
C
C     CALCULATE AVERAGE ENERGY AND EFFECTIVE TEMPERATURE --
V
      IF (ITER.GT.0) GO TO 75
      CALL SIMPSON (B, MESH/2, DX, UBAR)
      UBAR = UBAR/FNORM
      KTE = (2./3.)*E*UBAR
      TE = KTE/KB
C
C ----------------------------------------------------------------------
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)
      DO 12 I = 1,MESH
      F0 = F1
      F1 = F(I+1)
      FAVG = (F0 + F1)/2.
      DF = F1-F0
      A(I,1) = QM(I,1)*FAVG
      A(I,2) = QM(I,1)*DF
      A(I,3) = QM(I,2)*FAVG
```

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

```
            PCT(5) = DISCH/P
            PCT(6) = DEPOSIT/P
            PERCENT = PCT(1) + PCT(2) + PCT(3) + PCT(4)
            PBAL = PERCENT - 100.
            GO TO 54
C
   53       PMAX = AMAX1(ABS(ELASTIC), ABS(PCOLL), ABS(DEDT))
            P = ELASTIC + ELECT + PCOLL + DEDT
            PBAL = 100.*P/PMAX
C
   54 FB = 2./(KTE/E)**1.5/SQRT(PI)
      EX = EXP(-E*DX/KTE)
      DO 19 I = 1,M
      B(I) = F(I)/FB
   19 FB = FB*EX
C
C ------------------------------------------------------------------------
C
C     TABLE OF PLASMA PARAMETERS (UBAR, TE, VD, D, ... ETC.) --
C
      NG1 = NGAS-1
      IF (.NOT.OUT(1)) GO TO 21
      NX = 68-6*NGAS
      ENCODE (150,250,FORM) NX, NG1, NG1
      WRITE (6,350) KAPTION
      IF (NGAS.EQ.1) WRITE (6,210) NAME(1)
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I),I=1,NGAS),(FI(I),I=1,NGAS)
C
      WRITE (6,100) EVCM, TMOL, NMOL, PTOT, ATM, EN, EP, KVCMATM, UBAR,
     1   TE, D, MU, EK, VD, COND, CONDUCT, RHO, AMPS, DISCH, KAPT,
     2   DEPOSIT, KAPT, PTOTAL, KAPT, PCOLL, KAPT, ELASTIC, KAPT, DEDT,
     3   KAPT, PWR, KAPT, NU, NE, FE, S0, SB, IONIZE, ATTACH, DLNEDT,
     4   DNEDT
C
      WRITE (6,380) MESH, EMAX, DX
C
C     -----------------
C
C
C     TABULAR OUTPUT OF ELECTRON ENERGY DISTRIBUTION FUNCTION --
C
   21 NX = 58 - 6*NGAS
      ENCODE (150,220,FORM) NX, NG1, NG1
      IF (.NOT.OUT(2)) GO TO 22
      MULT = (MESH-1)/100 + 1
      DU = MULT*DX
      WRITE (6,350) KAPTION
      WRITE (6,200) DU, X(M), MESH, DX
      WRITE (6,330) EN, EP, KVCMATM
      IF (FE.GT.0.) WRITE (6,320) FE, NE
      IF (NGAS.EQ.1) GO TO 17
      WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I),I=1,NGAS), TMOL
      GO TO 7
   17 WRITE (6,260) NAME(1), TMOL
    7 WRITE (6,290)
      WRITE (6,240) (X(I), F(I), I = 1,M,MULT)
      WRITE (6,160)
      IF (ITER.GT.0) WRITE (6,180) TIME, ITER, DELTA, PBAL
```

```
      WRITE (6,280) DATE
C
   22 IF (.NOT.OUT(3)) GO TO 23
C
C     ------------------
C
C     PLOT THE DISTRIBUTION FUNCTION F(U)/F(0) --
C
      WRITE (6,110) F(1)
      WRITE (6,330) EN, EP, KVCMATM
      IF (FE.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
C
      DF = 2.
      IF (G(MP).GT.1.E-10) DF = 1.
      F0 = -10.*DF
      CALL PXLOGY (MAX, MP, 1, G, F0, DF, X, 0., 0., .TRUE., .FALSE.,
     1.TRUE., .TRUE., .TRUE., TITLE, 1, 0)
      WRITE (6,150) DATE
C
C     -------------------
C
C     BOLTZMANN DISTRIBUTION FUNCTION WITH EFFECTIVE TEMPERATURE TE IS
C     GIVEN BY FB(U,TE) = 2(KTE/E)**(-3/2)/SQRT(PI)*EXP(-E*U/KTE).  PLOT
C     F(U)/FB(U,TE) --
C
   23 IF (.NOT.OUT(4)) GO TO 30
C
      WRITE (6,120) TE
      WRITE (6,330) EN, EP, KVCMATM
      IF (FE.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
C
      CALL PXLOGY (MAX, MP, 1, B, -4.0, .5, X, 0., 0., .TRUE., .FALSE.,
     1 .TRUE., .TRUE., .TRUE., TITLE, 1, 0)
C
      WRITE (6,150) DATE
C
C     ------------------
C
C     TABULAR PLOT OF COLLISION RATES, POWER TRANSFER --
C
   30 IF (.NOT.OUT(5)) GO TO 99
      SKIP = SH(1H1)
      LSKIP = 1H
      IF (SB.EQ.0.) LSKIP = 1H+
      L = 15
      LINE = LL = 0
      DO 18 J = 1,NK
      IF (LL.NE.LINE) GO TO 28
         J1 = L - (NK-J+1)
         IF (J1.GT.7) J1 = 7
         IF (J1.GE.1) ENCODE (10,300,SKIP) J1
```

```
      WRITE (6,SKIP)
      WRITE (6,270) KAPTION
      WRITE (6,330) EN, EP, KVCMATM
      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 (S0.NE.0.) WRITE (6,390) LSKIP, S0
   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)
      IF (PTOTAL.NE.0.) WRITE (6,420) PCTJ
C
      LL = L*(LINE/L)
      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
      GO 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(1H-)//50X*E (FIELD)*4X*= *
     11PE11.4,3X*VOLT/CM*//50X*TMOL*8X* = *0PF8.0,6X*DEG K*/50X*NMOL*8X
     2* = *1PE11.4,3X*CM-3*/50X*PTOT*8X* = *0PF9.1,5X*TORR*/62X* = *F8.2
     3,6X*ATM*//50X*E/N*9X* = *1PE11.4,3X*VOLT CM2*/50X*E/P*9X* = *0PF8.
     42,6X*V/CM/TORR*/62X* = *F8.2,6X*KV/CM/ATM*//50X*<U>*9X* = *0PF8.3,
     56X*EV*/50X*TE = 2<U>/3K = *F9.0,5X*DEG K*//50X*DIFFUSION D  = *
```

```
     6F9.2,5X*CM2/SEC*/50X*MOBILITY MU  = *F8.1,6X*CM2/VOLT/S*/50X*EK =  BOLTZ
     7D/MU*3X* = *0PF8.3,6X*EV*/50X≠VDRIFT =MU*E = ≠1PE11.4,3X*CM/SEC*// BOLTZ
     850X*CONDUCT/NE   = *E11.4,3X*CM2/OHM*/50X*CONDUCT*5X* = *E11.4,3X  BOLTZ
     9*/OHM-CM*/50X≠RHO*NE≠6X* = *E11.4,3X*OHM/CM2*/50X≠J/NE = E*VD  = ≠ BOLTZ
     $E11.4,3X*AMP CM/EL*//50X*DISCHARGE*3X* = *E11.4,3X,A10/50X*DEPOSIT BOLTZ
     1ION   = *E11.4,3X,A10/50X*TOTAL POWER  = *E11.4,3X,A10//50X*INELAS BOLTZ
     2TIC     = *E11.4,3X,A10/50X*ELASTIC HEAT = *E11.4,3X,A10/50X*E<U>DN BOLTZ
     3E/DT    = *E11.4,3X,A10/50X*POWER DISS    = *E11.4,3X,A10//50X*NU(MO BOLTZ
     4M)*5X* = *E11.4,3X*SEC-1*//50X*NE*10X* = *E11.4,3X*CM-3*/50X*NE/NM BOLTZ
     50L*5X* = *E11.4/50X*S(U = 0)/NE  = *E11.4,3X*SEC-1*/50X*S(U > 0)/N BOLTZ
     6E  = *E11.4,3X*SEC-1*/50X*NU(IONIZE)    = *E11.4,3X*SEC-1*/50X*NU(A BOLTZ
     7TT,REC)  = *E11.4,3X*SEC-1*/50X*DNE/DT/NE     = *E11.4,3X*SEC-1*/50 BOLTZ
     8X*DNE/DT*6X* = *E11.4,3X*CM-3/SEC*//48X,42(1H-))                   BOLTZ
C                                                                        BOLTZ
  110 FORMAT (1H1,9X, *LOG PLOT OF ELECTRON DISTRIBUTION F(U)/F(0) AS A  BOLTZ
     1FUNCTION OF ELECTRON ENERGY U (EV), WHERE F(0) = *,1PE10.3,* EV*,  BOLTZ
     22H**,*(-3/2)*/)                                                    BOLTZ
C                                                                        BOLTZ
  120 FORMAT (1H1,17X*LOG PLOT OF RELATIVE ELECTRON DISTRIBUTION FUNCTIO BOLTZ
     1N F(U)/FBOLTZ(U,TE), WHERE TE = (2/3)UAVG/KB = *F6.0* DEG K*/)     BOLTZ
C                                                                        BOLTZ
  130 FORMAT (/I3,3X,5A10,A12,A8,3A12,1PE13.3,0PF8.2)                    BOLTZ
C                                                                        BOLTZ
  140 FORMAT (26X,*U(AVG) = *,F6.3,* EV, TE = *,F6.0,* DEG K, VD = *,    BOLTZ
     1 1PE8.2* CM/SEC, MU = *E8.2* CM2/VOLT/SEC*/)                       BOLTZ
C                                                                        BOLTZ
  150 FORMAT (/62X*ELECTRON ENERGY U (EV)*16X*DR. WILLIAM B. LACINA,*A11 BOLTZ
     1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)                           BOLTZ
C                                                                        BOLTZ
  160 FORMAT (/1X,135(1H-))                                              BOLTZ
C                                                                        BOLTZ
  170 FORMAT (95X*MOMENTUM TRANSFER = *1PE10.3)                          BOLTZ
C                                                                        BOLTZ
  180 FORMAT (/1X*F(U) CONVERGED IN*F5.1* SEC, IN*I3* ITERATIONS; MAXIMU BOLTZ
     $M RELATIVE CHANGE IN LAST ITERATION <*1PE10.3*; POWER BALANCE ACCU BOLTZ
     $RACY = *0PF6.2* %.*)                                               BOLTZ
C                                                                        BOLTZ
  190 FORMAT (1PE10.3)                                                   BOLTZ
C                                                                        BOLTZ
  200 FORMAT (15X    *NORMALIZED ELECTRON DISTRIBUTION FUNCTION F(U), IN BOLTZ
     1 UNITS OF EV*,2H**,*(-3/2), WITH OUTPUT AT*0PF6.3* EV INTERVALS.*/ BOLTZ
     215X*THE ELECTRON ENERGY RANGE [0,*0PF6.2*] EV WAS SUBDIVIDED INTO  BOLTZ
     3*I4* INTERVALS, GIVING A RESOLUTION = *F5.3* EV.*/)                BOLTZ
C                                                                        BOLTZ
  210 FORMAT (48X*GAS MIXTURE --*//65X*PURE *A3)                         BOLTZ
C                                                                        BOLTZ
  220 FORMAT (1H(,I2,4HXA3,,I2,20H(*/*A3),* = *2PF6.2,,I2,38H(* /*F6.2), BOLTZ
     1*, TMOL =*,0PF5.0* DEG K*/))                                       BOLTZ
C                                                                        BOLTZ
  230 FORMAT (2X*J*14X*REACTION(J)*21X*N(A)*9X*N(B)*5X*U(J)*4X*DNE/DT/NE BOLTZ
     1*2X*<VSIG(A→B)> <VSIG(B→A)>  (NET) POWER PERCENT* /13X*A + E(-) → BOLTZ
     2B + E(-)*16X*(CM-3)*7X*(CM-3)*4X*(EV)*5X*(SEC-1)*4X*(CM3/SEC)*3X   BOLTZ
     3*(CM3/SEC)*3X*(*A9*)  POWER*/1X,135(1H-)/)                         BOLTZ
C                                                                        BOLTZ
  240 FORMAT (5(0PF12.3,1PE14.5)/)                                       BOLTZ
C                                                                        BOLTZ
  250 FORMAT (24H(48X,*GAS MIXTURE -- *//,I2,5HX,A3,I2,20H(*/*A3)* = *,2 BOLTZ
```

```
     1PF6.2,,I2,11H(* /*F6.2)))                                        BOLTZ
C                                                                      BOLTZ
  260 FORMAT (55X,*PURE *,A3,*  TMOL =*,F5.0,* DEG K*/)               BOLTZ
C                                                                      BOLTZ
  270 FORMAT (48X,4A10//30X*PLASMA KINETICS ANALYSIS WITH SUMMARY OF ELE BOLTZ
     1CTRON PARAMETERS, COLLISION RATES,*/31X*AND POWER BALANCE FOR ALL BOLTZ
     2ELASTIC AND INELASTIC COLLISION PROCESSES INCLUDED*/)            BOLTZ
C                                                                      BOLTZ
  280 FORMAT (/103X,  *DR. WILLIAM B. LACINA,*A11/103X*NORTHROP RESEARCH BOLTZ
     1 AND TECHNOLOGY*)                                                BOLTZ
C                                                                      BOLTZ
  290 FORMAT (2X,132(1H-)/5(7X*U(EV)*7X*F(U)*3X)/2X,132(1H-)/)         BOLTZ
C                                                                      BOLTZ
  300 FORMAT (5H(1H1,,I1,4H(/)))                                       BOLTZ
C                                                                      BOLTZ
  310 FORMAT (82X*E-E POWER TRANSFER DISCREPANCY = *1PE10.3,0PF8.2* %*) BOLTZ
C                                                                      BOLTZ
  320 FORMAT (34X*FRACTIONAL IONIZATION = NE/NTOT = *1PE10.3*, NE = *  BOLTZ
     1 E10.3* CM-3*)                                                   BOLTZ
C                                                                      BOLTZ
  330 FORMAT (31X*E/NTOT = *1PE10.3* VOLT CM2 = *0PF6.3* VOLT/CM/TORR = BOLTZ
     1*F6.3* KVOLT/CM/ATM*)                                            BOLTZ
C                                                                      BOLTZ
  340 FORMAT (* TOTAL ELECTRICAL  = *1PE10.3,A11* 100.00 %,*15X*TOTAL PO BOLTZ
     1WER INTO COLLISIONS, HEATING, STORAGE = *1PE10.3,0PF8.2* %*)     BOLTZ
C                                                                      BOLTZ
  350 FORMAT (1H1,48X,4A10/)                                           BOLTZ
C                                                                      BOLTZ
  360 FORMAT (F7.2)                                                    BOLTZ
C                                                                      BOLTZ
  370 FORMAT (/1X*DISCHARGE POWER   = *1PE10.3,A11,0PF7.2* %,*18X*POWER BOLTZ
     1INTO INELASTIC E-MOLECULE COLLISIONS = *1PE10.3,0PF8.2* %*/1X*SOUR BOLTZ
     2CE DEPOSITION = *1PE10.3,A11,0PF7.2* %,*34X*ELASTIC E-MOLECULE HEA BOLTZ
     3TING = *1PE10.3,0PF8.2* %*/22X*---------*12X*-------*14X*D/DT(STORE BOLTZ
     1D ELECTRON KINETIC ENERGY) = E<U>DNE/DT = *1PE10.3,0PF8.2* %*)   BOLTZ
C                                                                      BOLTZ
  380 FORMAT (/48X*CALCULATION PARAMETERS USED:*/48X*MESH =*I4*, EMAX =* BOLTZ
     10PF6.2* EV, DE =*F5.3* EV.*)                                     BOLTZ
C                                                                      BOLTZ
  390 FORMAT (A1,64X*S(U = 0)/NE =*1PE10.3)                            BOLTZ
C                                                                      BOLTZ
  400 FORMAT (/1X*DISCHARGE POWER   = 0.0*46X*POWER INTO INELASTIC E-MOL BOLTZ
     1ECULE COLLISIONS = *1PE10.3,A11/1X*SOURCE DEPOSITION = 0.0*62X*ELA BOLTZ
     2STIC E-MOLECULE HEATING = *1PE10.3,A11/63X*D/DT(STORED ELECTRON KI BOLTZ
     3NETIC ENERGY) = E<U>DNE/DT = *1PE10.3,A11)                       BOLTZ
C                                                                      BOLTZ
  410 FORMAT (* TOTAL ELECTRICAL  = 0.0*43X*TOTAL POWER INTO COLLISIONS, BOLTZ
     1 HEATING, STORAGE = *1PE10.3,A11)                                BOLTZ
C                                                                      BOLTZ
  420 FORMAT (1H+,124X,F8.2)                                           BOLTZ
C                                                                      BOLTZ
  430 FORMAT (1H+64X,*S(U > 0)/NE =*1PE10.3)                           BOLTZ
C                                                                      BOLTZ
  500 FORMAT (1H1,20(/)28X,*PROGRAM COMPLETED *,I3,* ITERATIONS IN *,  BOLTZ
     1F4.0,* SECONDS WITHOUT SUCCESSFUL CONVERGENCE.*/28X,*REQUESTED LIM BOLTZ
     2ITS WERE *,I3,* ITERATIONS AND *,F4.0,* SECONDS, WITH A CONVERGENC BOLTZ
     3E CONDI-*/28X,*TION THAT THE CHANGE BETWEEN TWO ITERATIONS BE LESS BOLTZ
```

```
     4 THAN EPS = *,1PE10.3,*.*)
C
  600 FORMAT (/28X,*DISTRIBUTION FUNCTION BECAME NEGATIVE AT SOME POINTS
     1 IN LAST ITERATION*)
C
C ---------------------------------------------------------------------
C
C     CONVERGENCE FAILED FOR LIMITS PRESCRIBED.  ERROR MESSAGE --
C
   98 WRITE (6,500) ITER, TIME, ITMAX, TMAX, EPS
      IF (ERROR) WRITE (6,600)
C
C ---------------------------------------------------------------------
C
   99 RETURN
      END


      SUBROUTINE SIMEQ (A, M, N, N1, SING)
C
C .....................................................................
C
C     THIS SUBROUTINE WILL SOLVE AN N X N SYSTEM OF SIMULTANEOUS EQUA-
C     TIONS OF THE FORM A(I,J)X(J) = B(I).  A IS DIMENSIONED A(M,NN) IN
C     THE MAIN PROGRAM, AND INPUT CONSISTS OF PUTTING THE N X N MATRIX
C     IN THE UPPER LEFT HAND BOX OF A.  N1 DIFFERENT B VECTORS CAN BE
C     SPECIFIED AS INPUT STORED IN SUCCESSIVE COLUMNS TO THE RIGHT OF
C     THE N X N MATRIX (N ≤ M, N+N1≤NN).  UPON OUTPUT, THE SOLUTION VEC-
C     TORS REPLACE THE INPUT VECTORS B.  IF B IS AN N X N UNIT MATRIX ON
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 + N1
      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,
     1 ERROR, KFLAG, JSTART, MAXDER, M, PW)
C
C.......................................................................
C
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-
C     TING 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)
C                                SIMEQ (PW, M, N, 1, SING)
C                                JACOB (M, T, Y, PW)
C
C     THE FIRST, RATES, EVALUATES THE DERIVATIVES OF THE DEPENDENT VARI-
C     ABLES STORED IN Y(1,I) FOR I = 1 TO N, AND STORES THE DERIVATIVES
C     IN THE VECTOR DY.  THE SECOND, SIMEQ, IS CALLED ONLY IF THE METHOD
C     FLAG MF IS SET TO 1 OR 2 FOR STIFF METHODS.  IT MUST INVERT THE
C     N X N MATRIX STORED IN THE ARRAY PW(M,M).  IF THE INVERSION IS
C     SUCCESSFUL, SING (SINGULARITY) IS RETURNED FALSE.  JACOB IS USED
C     ONLY IF MF IS 1, AND COMPUTES THE PARTIAL DERIVATIVES OF THE DIF-
C     FERENTIAL EQUATIONS AS DESCRIBED UNDER THE MF PARAMETER.
C
C     THE TEMPORARY STORAGE SPACE IS PROVIDED BY THE CALLER IN THE ARRAY
C     PW AND THE ARRAY SAVE.  THE ARRAY PW IS USED ONLY TO HOLD THE MA-
C     TRIX OF THE SAME NAME, BUT SAVE IS USED TO HOLD SEVERAL ARRAYS.
C     THE REGIONS USED ARE:
C
C        SAVE(J,I)      -- 1≤J≤8 AND 1≤I≤N IS USED TO SAVE THE VALUES OF
C                          Y IN CASE A STEP HAS TO BE REPEATED.
C
C        SAVE(9,I)      -- IS USED MAINLY TO HOLD THE CORRECTION TERMS IN
C                          THE CORRECTOR LOOP.
C
C        SAVE(10,I)     -- IS USED TO SAVE THE VALUES OF THE SUMS OF ALL
C                          OF THE CORRECTION TERMS IN THE PREVIOUS STEP
C                          AFTER THEY HAVE BEEN ACCUMULATED IN THE ARRAY
C                          ERROR IN THE CURRENT STEP.  THIS ENABLES THE
C                          BACKWARDS DIFFERENCE OF ERROR TO BE FORMED.  IT
C                          IS USED TO ESTIMATE THE STEP SIZE FOR ONE ORDER
C                          HIGHER THAN CURRENT.
C
C        SAVE(N1+I,1) -- IS USED TO STORE THE DERIVATIVES WHEN THEY ARE
C                          COMPUTED BY RATES.  IT IS ALSO ACCESSED AS
C                          SAVE(N2,1) AS A COMPLETE ARRAY.
C
C        SAVE(N5+I,1) -- HOLDS THE DERIVATIVES DURING JACOBIAN EVALUA-
C                          TIONS.  IT IS REFERENCED AS SAVE(N6,1) AS A COM-
C                          PLETE ARRAY.
```

```
C
C      INPUT AND OUTPUT PARAMETERS HAVE THE FOLLOWING MEANING --
C
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.
C
C      M         DIMENSION DECLARATOR FOR THE JACOBIAN PW AND THE ARRAY Y
C                CONTAINING THE DEPENDENT VARIABLES AND THEIR DERIVATIVES.
C
C      T         THE INDEPENDENT VARIABLE.
C
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
C
C                                    NQ
C                      Y(I)(T+E)  =  SUMJ  Y(I,J+1)*S**J
C                                    J=0
C
C      SAVE      A BLOCK OF AT LEAST 12*N SCRATCH LOCATIONS USED BY THE
C                SUBROUTINES.
C
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.
C
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.
C
C      HMAX      THE MAXIMUM SIZE TO WHICH THE STEP SIZE WILL BE INCREASED.
C
C      EPS       THE ERROR TEST CONSTANT.  SINGLE STEP ERROR ESTIMATES DI-
C                VIDED BY YMAX(I) MUST BE LESS THAN THIS IN THE EUCLIDEAN
C                NORM.  THE STEP AND/OR ORDER IS ADJUSTED TO ACHIEVE THIS.
C
C      MF        THE METHOD INDICATOR.  THE FOLLOWING ARE AVAILABLE:
C
C                      0  AN ADAMS PREDICTOR-CORRECTOR IS USED.
C
C                      1  A MULTISTEP METHOD SUITABLE FOR STIFF SYSTEMS IS
C                         USED.  IT WILL ALSO WORK FOR NON-STIFF SYSTEMS.
C                         HOWEVER, THE USER MUST PROVIDE A SUBROUTINE JACOB
C                         WHICH EVALUATES THE PARTIAL DERIVATIVES OF THE
C                         DIFFERENTIAL EQUATIONS WITH RESPECT TO THE Y-S.
```

```
C                    THIS IS DONE BY CALL JACOB (M, T, Y, PW(1,N+2)).
C                    THE JACOBIAN PHI(I,J), WHICH REPRESENTS THE PAR-
C                    TIAL OF THE ITH EQUATION WITH RESPECT TO THE JTH
C                    DEPENDENT VARIABLE, IS STORED IN THE UPPER LEFT
C                    N X N CORNER OF THAT PART OF THE ARRAY PW BEGIN-
C                    NING AT COLUMN (N+2).
C
C                 2  THE SAME AS CASE 1, EXCEPT THAT THIS SUBROUTINE
C                    COMPUTES THE PARTIAL DERIVATIVES BY NUMERICAL DIF-
C                    FERENCING OF THE DERIVATIVES.  HENCE, JACOB IS NOT
C                    CALLED.
C
C     YMAX    AN ARRAY OF N LOCATIONS WHICH CONTAINS THE MAXIMUM OF EACH
C             Y SEEN SO FAR, AUTOMATICALLY UPDATED AFTER EACH COMPLETED
C             STEP (UNLESS THE USER OVERRIDES IT BY CHANGING YMAX BEFORE
C             A SUBSEQUENT CALL).  ALL OF THE COMPONENTS OF YMAX SHOULD
C             BE INITIALIZED TO 1.0 BEFORE THE FIRST ENTRY.  (CF. ALSO
C             THE DESCRIPTION FOR EPS, GIVEN ABOVE.)
C
C     ERROR   AN ARRAY OF N ELEMENTS WHICH CONTAIN THE ESTIMATED ONE
C             STEP ERROR IN EACH COMPONENT.
C
C     KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS--
C
C                +1  THE STEP WAS SUCCESSFUL; T IS ADVANCED TO (T+H)
C                    AND RESULTS OF INTEGRATION FROM T TO (T+H) ARE
C                    RETURNED TO THE CALLING PROGRAM.
C                -1  THE STEP WAS TAKEN WITH H = HMIN, BUT THE
C                    REQUESTED ERROR WAS NOT ACHIEVED.  T IS ADVANCED
C                    TO (T+DT).  CALLING PROGRAM MUST APPROVE FAILURE
C                    OF ACCURACY.
C                -2  THE MAXIMUM ORDER SPECIFIED WAS FOUND TO BE TOO
C                    LARGE.
C                -3  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED FOR
C                    H > HMIN.  T IS NOT ADVANCED.  CALLING PROGRAM
C                    MUST DECREASE H AND HMIN AND TRY AGAIN.
C                -4  THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED
C                    FOR THIS PROBLEM.  T IS NOT ADVANCED.  ETA MUST
C                    BE INCREASED.
C
C     JSTART  INPUT INDICATOR WITH THE FOLLOWING MEANINGS --
C
C                -1  REPEAT THE LAST STEP WITH A NEW H
C                 0  PERFORM THE FIRST STEP.  THE FIRST STEP MUST BE
C                    DONE WITH THIS VALUE OF JSTART SO THAT THE SUBROU-
C                    TINE CAN INITIALIZE ITSELF.
C                +1  TAKE A NEW STEP CONTINUING FROM THE LAST.
C
C             JSTART IS SET TO NQ, THE CURRENT ORDER OF THE METHOD AT
C             EXIT.  NQ IS ALSO THE ORDER OF THE MAXIMUM DERIVATIVE
C             AVAILABLE.
C
C     MAXDER  THE MAXIMUM DERIVATIVE THAT SHOULD BE USED IN THE METHOD.
C             SINCE THE ORDER IS EQUAL TO THE HIGHEST DERIVATIVE USED,
C             THIS RESTRICTS THE ORDER.  IT MUST BE LESS THAN 8 FOR
C             ADAMS AND 7 FOR THE STIFF METHODS.
C
```

```
C     PW       A BLOCK OF AT LEAST M*(2M+1) SCRATCH LOCATIONS.
C
C...................................................................
C
      DIMENSION Y(M,8), YMAX(1), SAVE(10,1), ERROR(1), PW(M,1), A(8),
     1 PERTST(7,2,3)
C
      LOGICAL SING
C
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,
     1              2.0, 12.0, 24.0, 37.89, 53.33, 70.08, 87.97,
     2              3.0, 6.0, 9.167, 12.5, 15.98, 1.0, 1.0,
     3              12.0, 24.0, 37.89, 53.33, 70.08, 87.97, 1.0,
     4              1.0, 1.0, 0.5, 0.1667, 0.04133, 0.008267, 1.0,
     5              1.0, 1.0, 2.0, 1.0, 0.3157, 0.07407, 0.0139 /
C
      DATA A(2) / -1.0 /
      IRET = 1
      KFLAG = 1
      METHOD = MF+1
      IF (JSTART.LE.0) GO TO 140
C
C     BEGIN BY SAVING INFORMATION FOR POSSIBLE RESTARTS AND CHANGING H
C     BY THE FACTOR R IF THE CALLER HAS CHANGED H.  ALL VARIABLES DEPEN-
C     DENT ON H MUST ALSO BE CHANGED.  E IS A COMPARISON FOR ERRORS OF
C     THE CURRENT ORDER, NQ.  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
      DO 110 J = 1,K
  110 SAVE(J,I) = Y(I,J)
      HOLD = HNEW
      IF (H.EQ.HOLD) GO TO 130
  120 RACUM = H/HOLD
      IRET1 = 1
      GO TO 750
  130 NQOLD = NQ
      TOLD = T
      RACUM = 1.0
      IF (JSTART.GT.0) GO TO 250
      GO TO 170
  140 IF (JSTART.EQ.-1) GO TO 160
C
C     ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL DERIVA-
C     TIVES ARE CALCULATED.
C
      NQ = 1
      N1 = N*10
      N2 = N1 + 1
      N5 = N1 + N
      N6 = N5 + 1
C
C
```

```
C     --------------------------------
      CALL DNDT (N, T, Y, SAVE(N2,1))
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.
C     CHECK FOR EXCESSIVE ORDER.  THE LAST TWO STATEMENTS OF THIS SEC-
C     TION SET IWEVAL > 0 IF PW IS TO BE REEVALUATED BECAUSE OF THE OR-
C     DER CHANGE, AND THEN REPEAT THE INTEGRATION STEP IF IT HAS NOT YET
C     BEEN DONE (IRET = 1) OR SKIP TO A FINAL SCALING BEFORE EXIT IF IT
C     HAS BEEN COMPLETED (IRET = 2).
C
  170 IF (MF.EQ.0) GO TO 180
      IF (NQ.GT.6) GO TO 190
      GO TO (221,222,223,224,225,226), NQ
  180 IF (NQ.GT.7) GO TO 190
      GO TO (211,212,213,214,215,216,217), NQ
  190 KFLAG = -2
      RETURN
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) = -0.34861111111111
C     A(1) = -251./720.
      A(3) = -1.0416666666667
C     A(3) = -25./24.
      A(4) = -0.48611111111111
C     A(4) = -35./72.
      A(5) = -0.10416666666667
C     A(5) = -5./48.
```

```
      A(6) = -0.83333333333333E-02
C     A(6) = -1./120.
      GO TO 230
  216 A(1) = -0.32986111111111
C     A(1) = -95./288.
      A(3) = -1.1416666666667
C     A(3) = -137./120.
      A(4) = -0.625
C     A(4) = -5./8.
      A(5) = -0.17708333333333
C     A(5) = -17./96.
      A(6) = -0.025
C     A(6) = -1./40.
      A(7) = -0.13888888888889E-02
C     A(7) = -1./720.
      GO TO 230
  217 A(1) = -0.31559193121693
C     A(1) = -19087./60480.
      A(3) = -1.225
C     A(3) = -49./40.
      A(4) = -0.75185185185185
C     A(4) = -203./270.
      A(5) = -0.25520833333333
C     A(5) = -49./192.
      A(6) = -0.48611111111111E-01
C     A(6) = -7./144.
      A(7) = -0.48611111111111E-02
C     A(7) = - 7./1440.
      A(8) = -0.19841269841270E-03
C     A(8) = -1./5040.
      GO TO 230
  221 A(1) = -1.
      GO TO 230
  222 A(1) = -0.66666666666667
C     A(1) = -2./3.
      A(3) = -0.33333333333333
C     A(3) = -1./3.
      GO TO 230
  223 A(1) = -0.54545454545454
      A(3) = A(1)
C     A(1) = A(3) = -6./11.
      A(4) = -0.90909090909091E-01
C     A(4) = -1./11.
      GO TO 230
  224 A(1) = -0.48
      A(3) = -0.7
      A(4) = -0.2
      A(5) = -0.02
      GO TO 230
  225 A(1) = -0.43795620437956
C     A(1) = -120./274.
      A(3) = -0.82116788321168
C     A(3) = -225./274.
      A(4) = -0.31021897810219
C     A(4) = -85./274.
      A(5) = -0.54744525547445E-01
C     A(5) = -15./274.
```

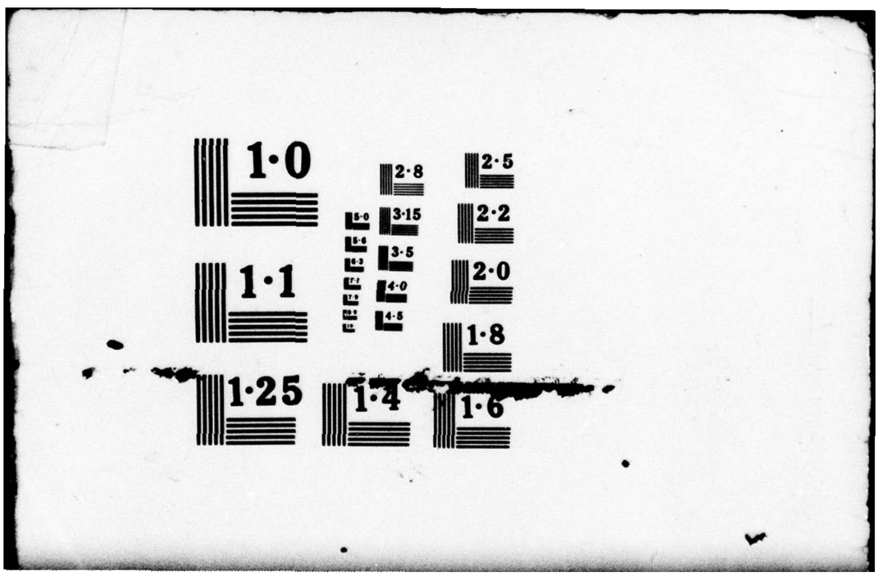
```
      A(6) = -0.36496350364964E-02
C     A(6) = -1./274.
      GO TO 230
  226 A(1) = -0.40816326530612
C     A(1) = -180./441.
      A(3) = -0.92063492063492
C     A(3) = -58./63.
      A(4) = -0.41666666666667
C     A(4) = -15./36.
      A(5) = -0.99206349206349E-01
C     A(5) = -25./252.
      A(6) = -0.11904761904762E-01
C     A(6) = -3./252.
      A(7) = -0.56689342403628E-03
C     A(7) = -1./1764.
C
  230 K = NQ+1
      IDOUB = K
      MTYP = (4-MF)/2
C     MTYP = 1 (STIFF, MF = 1 OR 2), MTYP = 2 (ADAMS-MOULTON).
      ENQ2 = 0.5/(NQ+1)
      ENQ3 = 0.5/(NQ+2)
      ENQ1 = 0.5/NQ
      PEPSH = EPS
      EUP = (PERTST(NQ,MTYP,2)*PEPSH)**2
      E = (PERTST(NQ,MTYP,1)*PEPSH)**2
      EDWN = (PERTST(NQ,MTYP,3)*PEPSH)**2
      IF (EDWN.EQ.0) GO TO 780
      BND = EPS*ENQ3/N
      IWEVAL = MF
      GO TO (250,680), IRET
C
C     THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY MULTI-
C     PLYING THE SAVED INFORMATION BY THE PASCAL TRIANGLE MATRIX.
C
  250 T = T+H
      DO 260 J = 2,K
      DO 260 J1 = J,K
      J2 = K-J1+J-1
      DO 260 I = 1,N
  260 Y(I,J2) = Y(I,J2) + Y(I,J2+1)
C
C     UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  CONVERGENCE IS TESTED BY
C     REQUIRING CHANGES TO BE LESS THAN BND WHICH IS DEPENDENT ON THE
C     ERROR TEST CONSTANT.  THE SUM OF THE CORRECTIONS IS ACCUMULATED IN
C     THE ARRAY ERROR(I).  IT IS EQUAL TO THE KTH DERIVATIVE OF Y MULTI-
C     PLIED BY  H**K/(FACTORIAL(K-1)*A(K)), AND IS THEREFORE PROPOR-
C     TIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER OF H PRESENT (H**K
C
      DO 270 I = 1,N
  270 ERROR(I) = 0.
C
      DO 430 L = 1,3
C
C     --------------------------------
      CALL DNDT (N, T, Y, SAVE(N2,1))
C     --------------------------------
```

```
C                                                                       GEAR
      GO TO (280,300,320), METHOD                                       GEAR
C                                                                       GEAR
C     THIS SECTION IS ENCOUNTERED ONLY FOR MF = 0 --                    GEAR
C                                                                       GEAR
  280 DO 290 I = 1,N                                                    GEAR
  290 SAVE(9,I) = Y(I,2) - SAVE(N1+I,1)*H                               GEAR
      GO TO 410                                                         GEAR
C                                                                       GEAR
C  ------------------------------------------------------------------- GEAR
C     EVALUATE THE JACOBIAN AND PLACE IT IN AN N X N BOX IN THE UPPER   GEAR
C     LEFT HAND CORNER OF THAT PART OF PW BEGINNING WITH COLUMN (N+2).  GEAR
C  ------------------------------------------------------------------- GEAR
C                                                                       GEAR
C     EVALUATE THE JACOBIAN BY A CALL TO AN EXTERNAL SUBROUTINE (THIS   GEAR
C     SECTION IS ENCOUNTERED ONLY IF MF = 1) --                         GEAR
C                                                                       GEAR
  300 IF (IWEVAL.LT.1) GO TO 380                                        GEAR
C                                                                       GEAR
C     ------------------------------                                    GEAR
      CALL JACOB (M, T, Y, PW(1,N+2))                                   GEAR
C     ------------------------------                                    GEAR
C                                                                       GEAR
      R = A(1)*H                                                        GEAR
      DO 310 I = 1,N                                                    GEAR
      DO 310 J = 1,N                                                    GEAR
      JPN1 = J+N+1                                                      GEAR
  310 PW(I,JPN1) = PW(I,JPN1)*R                                         GEAR
      GO TO 360                                                         GEAR
C                                                                       GEAR
C  ....................................                                 GEAR
C                                                                       GEAR
C     EVALUATE THE JACOBIAN INTO PW BY NUMERICAL DIFFERENCING.  R IS THE GEAR
C     CHANGE MADE TO THE ELEMENT OF Y.  IT IS EPS RELATIVE TO Y WITH A  GEAR
C     MINIMUM OF EPS**2.  THIS SECTION IS ENCOUNTERED ONLY IF MF = 2 -- GEAR
C                                                                       GEAR
  320 IF (IWEVAL.LT.1) GO TO 380                                        GEAR
      DO 330 I = 1,N                                                    GEAR
  330 SAVE(9,I) = Y(I,1)                                                GEAR
      DO 350 J = 1,N                                                    GEAR
      R = EPS*AMAX1(EPS,ABS(SAVE(9,J)))                                 GEAR
      Y(J,1) = Y(J,1) + R                                               GEAR
      D = A(1)*H/R                                                      GEAR
C                                                                       GEAR
C     ------------------------------                                    GEAR
      CALL DNDT (N, T, Y, SAVE(N6,1))                                   GEAR
C     ------------------------------                                    GEAR
C                                                                       GEAR
      JPN1 = J+N+1                                                      GEAR
      DO 340 I = 1,N                                                    GEAR
  340 PW(I,JPN1) = (SAVE(N5+I,1) - SAVE(N1+I,1))*D                      GEAR
  350 Y(J,1) = SAVE(9,J)                                                GEAR
C                                                                       GEAR
C  ....................................                                 GEAR
C                                                                       GEAR
C     IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE WITH GEAR
C     CONVERGENCE, PW IS REEVALUATED PRIOR TO STARTING THE CORRECTOR    GEAR
```

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


1·0
2·8
2·5
5·0
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

```
C     ITERATION IN THE CASE OF STIFF METHODS.  IWEVAL IS THEN SET TO -1
C     AS AN INDICATOR THAT IT HAS BEEN DONE.
C
  360 IWEVAL = -1
C
C     ADD THE IDENTITY MATRIX TO THE JACOBIAN.  THIS SECTION IS ENCOUN-
C     TERED ONLY FOR MF = 1 OR 2 --
C
      DO 370 I = 1,N
      IPN1 = I+N+1
  370 PW(I,IPN1) = 1.0 + PW(I,IPN1)
C
C     FETCH (1 + PHI), SHIFT IT TO THE LEFT, CALCULATE RHS VECTOR --
  380 DO 390 I = 1,N
      PW(I,N+1) = SAVE(N5+I,1) = Y(I,2) - SAVE(N1+I,1)*H
      DO 390 J = 1,N
      JPN1 = J+N+1
  390 PW(I,J) = PW(I,JPN1)
C
C     ----------------------------
      CALL SIMEQ (PW, M, N, 1, SING)
C     ----------------------------
C
      IF (SING) GO TO 440
C
      DO 400 I = 1,N
  400 SAVE(9,I) = PW(I,N+1)
C
C  ....................................
C
C     CORRECT AND SEE IF ALL OF THE CHANGES ARE LESS THAN BND RELATIVE
C     TO YMAX.  IF SO, THE CORRECTOR IS SAID TO HAVE CONVERGED.
C
  410 NT = N
      DO 420 I = 1,N
      Y(I,1) = Y(I,1) + A(1)*SAVE(9,I)
      Y(I,2) = Y(I,2) - SAVE(9,I)
      ERROR(I) = ERROR(I) + SAVE(9,I)
      IF (ABS(SAVE(9,I)).LE.(BND*YMAX(I))) NT = NT-1
  420 CONTINUE
      IF (NT.LE.0) GO TO 490
  430 CONTINUE
C
C  ....................................
C
C     THE CORRECTOR ITERATION FAILED TO CONVERGE IN THREE TRIES.  VARI-
C     OUS POSSIBILITIES ARE CHECKED FOR.  IF H IS ALREADY HMIN AND THIS
C     IS EITHER ADAMS METHOD OR THE STIFF METHOD IN WHICH THE MATRIX PW
C     HAS ALREADY BEEN REEVALUATED, A NO-CONVERGENCE EXIT IS TAKEN.
C     OTHERWISE, THE MATRIX PW IS REEVALUATED AND/OR THE STEP IS RE-
C     DUCED TO TRY AND GET CONVERGENCE.
C
C  ....................................
C
  440 T = TOLD
      IF ((H.LE.(HMIN*1.00001)).AND.((IWEVAL-MTYP).LT.-1)) GO TO 460
      IF ((MF.EQ.0).OR.(IWEVAL.NE.0)) RACUM = RACUM*0.25
```

```
      IWEVAL = MF
      IRET1 = 2
      GO TO 750
  460 KFLAG = -3
  470 DO 480 I = 1,N
      DO 480 J = 1,K
  480 Y(I,J) = SAVE(J,I)
      H = HOLD
      NQ = NQOLD
      JSTART = NQ
      RETURN
C
C  .....................................
C
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
C  .....................................
C
  490 D = 0.
      DO 500 I = 1,N
  500 D = D + (ERROR(I)/YMAX(I))**2
      IWEVAL = 0
      IF (D.GT.E) GO TO 540
      IF (K.LT.3) GO TO 520
C     COMPLETE THE CORRECTION OF THE HIGHER ORDER DERIVATIVES AFTER A
C     SUCCESSFUL STEP.
      DO 510 J = 3,K
      DO 510 I = 1,N
  510 Y(I,J) = Y(I,J) + A(J)*ERROR(I)
  520 KFLAG = 1
      HNEW = H
      IF (IDOUB.LE.1) GO TO 550
      IDOUB = IDOUB-1
      IF (IDOUB.GT.1) GO TO 700
      DO 530 I = 1,N
  530 SAVE(10,I) = ERROR(I)
      GO TO 700
C
C     REDUCE THE FAILURE FLAG COUNT TO CHECK FOR MULTIPLE FAILURES.
C     RESTORE T TO ITS ORIGINAL VALUE AND TRY AGAIN UNLESS THERE HAVE
C     BEEN THREE FAILURES.  IN THAT CASE, THE DERIVATIVES ARE ASSUMED
C     TO HAVE ACCUMULATED ERRORS SO A RESTART FROM THE CURRENT VALUE OF
C     Y IS TRIED.
C
```

```
  540 KFLAG = KFLAG-2
      IF (H.LE.(HMIN*1.00001)) GO TO 740
      T = TOLD
      IF (KFLAG.LE.-5) GO TO 720
C
C     PR1, PR2, AND PR3 WILL CONTAIN THE AMOUNTS BY WHICH THE STEP SIZE
C     SHOULD BE DIVIDED AT ORDER ONE LOWER, AT THIS ORDER, AND AT ORDER
C     ONE HIGHER, RESPECTIVELY.
C
  550 PR2 = (D/E)**ENQ2*1.2
      PR3 = 1.E 20
      IF ((NQ.GE.MAXDER).OR.(KFLAG.LE.-1)) GO TO 570
      D = 0.
      DO 560 I = 1,N
  560 D = D + ((ERROR(I) - SAVE(10,I))/YMAX(I))**2
      PR3 = (D/EUP)**ENQ3*1.4
  570 PR1 = 1.E 20
      IF (NQ.LE.1) GO TO 590
      D = 0.
      DO 580 I = 1,N
  580 D = D + (Y(I,K)/YMAX(I))**2
      PR1 = (D/EDWN)**ENQ1*1.3
  590 CONTINUE
      IF (PR2.LE.PR3) GO TO 650
      IF (PR3.LT.PR1) GO TO 660
  600 R = 1.0/AMAX1(PR1,1.E-04)
      NEWQ = NQ-1
  610 IDOUB = 10
      IF ((KFLAG.EQ.1).AND.(R.LT.1.1)) GO TO 700
      IF (NEWQ.LE.NQ) GO TO 630
C     COMPUTE ONE ADDITIONAL SCALED DERIVATIVE IF ORDER IS INCREASED.
      DO 620 I = 1,N
  620 Y(I,NEWQ+1) = ERROR(I)*A(K)/K
  630 K = NEWQ+1
      IF (KFLAG.EQ.1) GO TO 670
      RACUM = RACUM*R
      IRET1 = 3
      GO TO 750
  640 IF (NEWQ.EQ.NQ) GO TO 250
      NQ = NEWQ
      GO TO 170
  650 IF (PR2.GT.PR1) GO TO 600
      NEWQ = NQ
      R = 1.0/AMAX1(PR2,1.E-04)
      GO TO 610
  660 R = 1.0/AMAX1(PR3,1.E-04)
      NEWQ = NQ+1
      GO TO 610
  670 IRET = 2
      R = AMIN1(R,HMAX/ABS(H))
      H = H*R
      HNEW = H
      IF (NQ.EQ.NEWQ) GO TO 680
      NQ = NEWQ
      GO TO 170
  680 R1 = 1.
      DO 690 J = 2,K
```

```
      R1 = R1*R
      DO 690 I = 1,N
  690 Y(I,J) = Y(I,J)*R1
      IDOUB = K
  700 DO 710 I = 1,N
  710 YMAX(I) = AMAX1(YMAX(I), ABS(Y(I,1)))
      JSTART = NQ
      RETURN
  720 IF (NQ.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(1,I)
      SAVE(2,I) = HOLD*SAVE(N1+I,1)
  730 Y(I,2) = SAVE(2,I)*R
      NQ = 1
      KFLAG = 1
      GO TO 170
  740 KFLAG = -1
      HNEW = H
      JSTART = NQ
      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(1,I)
      IDOUB = K
      GO TO (130,250,640), IRET1
  780 KFLAG = -4
      GO TO 470
      RETURN
      END
```

```
      SUBROUTINE PLOT (MM, MP, MULT, Y, YO, DY, X, XO, DX, SCALEX,
     1 SCALEY, SAME, CLEAR, CENTER, NAME, NP, IP)
C
C ..........................................................................
C
C THIS PROGRAM GENERATES LINEAR, SEMILOG, OR LOGLOG PLOTS FOR UP TO 10
C VECTORS Y(I,J), J = 1,2,...,NP, PROVIDED BY AN ARRAY DIMENSIONED
C Y(MM,.) IN THE CALLING PROGRAM.  EACH VECTOR IS CONSIDERED TO BE A
C FUNCTION OF AN INDEPENDENT VARIABLE X(I), DEFINED BELOW.  ALL PLOTS
C ARE GENERATED FOR EACH VECTOR BY SAMPLING MP POINTS, TAKEN WITH A RE-
C PETITION INDEX MULT (I.E., I = 1, (1+MULT), ...., [1 + (MP-1)*MULT]).
C HORIZONTALLY, THE PLOTTING RESOLUTION CONSISTS OF A MAXIMUM OF 50
C SUBINTERVALS, CORRESPONDING TO 51 POINTS.  IF MP ≤ 51, MP POINTS ARE
C PLOTTED.  THERE ARE OPTIONS (AS WELL AS PROGRAM DEFAULTS) FOR SPECI-
C FYING INITIAL VALUES AND TICK MARK INTERVALS EITHER BY DIRECT INPUT,
C OR BY INTERNAL AUTOMATIC SCALING.  AUTOMATIC SCALING, WHICH IS ACCOM-
C PLISHED BY SUBROUTINES *AXIS* AND *INTERP*, GENERATES CONVENIENT IN-
C TEGER VALUES FOR INITIAL VALUES AND TICK MARK INTERVALS.  IF SCALEX
C = TRUE IS SPECIFIED, AUTOMATIC SCALING OF THE HORIZONTAL AXIS OCCURS.
C IF MP > 51, THE PROGRAM DEFAULTS TO AUTOMATIC SCALING FOR THE X-AXIS,
C AND INTERNALLY GENERATES, BY INTERPOLATION, 51 POINTS TO BE PLOTTED.
C IT IS ASSUMED THAT THE VECTOR X IS DEFINED BY X(I) = XO + (I-1)*DX,
C IN WHICH CASE, THE X-ORIGIN IS XO AND THE TICK MARK INTERVAL IS 5*DX.
C HOWEVER, THERE ARE TWO EXCEPTIONS IF DX = 0: 1) FOR AUTOMATIC X-AXIS
C SCALING, THE INPUT VECTOR X(I) IS USED, AND 2) IF MP ≤ 51, IT IS AS-
C SUMED THAT THE INDEPENDENT VARIABLE IS JUST THE INTEGER I. (THUS, IF
C DATA IS DEFINED OVER A NONUNIFORM GRID X(I), SCALEX = TRUE AND DX = 0
C SHOULD BE SPECIFIED BY THE CALLING PROGRAM.)  IF SCALEY = TRUE, AUTO-
C MATIC SCALING OF THE VERTICAL AXIS OCCURS FOR THE PLOT(S).  IF SCALEY
C = FALSE, THE ORIGIN(S) AND TICK MARK INTERVAL(S) FOR THE VERTICAL
C AXIS (DIVIDED INTO TEN TICKS) ARE TAKEN TO BE THOSE SPECIFIED BY THE
C VECTORS YO(2) AND DY(2), UNLESS DY(1) OR DY(2) = 0, IN WHICH CASE
C ONE, BOTH, OR ALL VECTORS ARE AUTOMATICALLY SCALED BY DEFAULT.  IF
C SAME = TRUE, PLOT(S) ARE SCALED TOGETHER USING SPECIFIED VALUES YO(1)
C AND DY(1) IF DY(1) ≠ 0, OR AUTOMATIC SCALE VALUES IF DY(1) = 0.  IN
C THE CASE THAT SAME = TRUE, A SINGLE (COMMON) VERTICAL SCALE FOR THE
C PLOT(S) WILL APPEAR AT THE LEFT.  IF SAME = FALSE AND TWO PLOTS (NP =
C 2) ARE REQUESTED, DIFFERENT LEFT AND RIGHT HAND SCALING WILL APPEAR.
C PLOTS WILL BE SMOOTHED BY INTERPOLATION, AND WILL BE CENTERED ON THE
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
C     AVERAGE VALUE, THE SMALL (AC) VARIATIONS WILL BE PLOTTED WITH A
C     (DC) BASELINE VALUE SPECIFIED.
C
C 2)  PROGRAM DEFAULTS TO FIRST 10 VECTORS IF NP > 10.
C
C 3)  PROGRAM DEFAULTS TO PLOT OF FIRST VECTOR ONLY IF HISTOGRAM IS
C     REQUESTED (CLEAR = FALSE).
C
C 4)  IF NP = 2 AND SAME = TRUE, PROGRAM DEFAULTS TO SAME = FALSE IF
C     THE TOTAL RANGE OF THE SMALLER OF THE TWO VECTORS IS LESS THAN
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
```

```
C        VERTICAL SIDE.
C
C   6)   PAGE SKIP AND HEADING TITLES MUST BE GENERATED IN THE CALLING
C        PROGRAM.
C
C   7)   IF ANY LOG OPTION IS CALLED, INITIAL AND INCREMENTAL VALUES EX-
C        PLICITLY SPECIFIED WITHOUT AUTOMATIC SCALING REQUEST (E.G., X0,
C        DX, Y0, DY) ARE UNDERSTOOD TO BE ACTUAL LOG QUANTITIES.
C
C   8)   IF MP DATA POINTS SAMPLED WITH A REPETITION MULTIPLE MULT EX-
C        CEEDS MM, MULT IS REDUCED TO ITS MAXIMUM ALLOWED VALUE.
C
C   INPUT PARAMETERS --
C
C         X0          =  INITIAL VALUE FOR INDEPENDENT VARIABLE X.
C
C         DX          =  INCREMENTAL VALUE FOR THE INDEPENDENT VARIABLE X,
C                        CORRESPONDING TO A RESOLUTION OF THE X-AXIS INTO
C                        50 INTERVALS, SPANNED BY A MAXIMUM OF 51 POINTS.
C                        TICK MARK INTERVAL = 5*MULT*DX
C
C         Y0(I)       =  INITIAL VALUE FOR LEFT (I=1) AND RIGHT (I=2) VER-
C                        TICAL SCALES.
C
C         DY(I)       =  INCREMENTAL TICK MARK INTERVAL FOR THE LEFT (I=1)
C                        AND RIGHT HAND (I=2) VERTICAL SCALES.
C
C         X(I)        =  INDEPENDENT VARIABLE PROVIDED BY THE CALLING PRO-
C                        GRAM, USED ONLY WHEN DX = 0 AND AUTOMATIC X-AXIS
C                        SCALING IS REQUIRED.  THE VECTOR X IS DESTROYED
C                        IN SOME SITUATIONS.
C
C         MM          =  DIMENSION DECLARATOR FOR THE ARRAY Y IN THE CALL-
C                        ING PROGRAM: Y(MM,..).
C
C         MP          =  NUMBER OF DATA POINTS TO BE SAMPLED FOR PLOT GEN-
C                        GENERATION.
C
C         MULT        =  REPETITION FACTOR.
C
C         Y(I,J)      =  ARRAY, DIMENSIONED Y(MM,..) IN THE CALLING PRO-
C                        GRAM.  PLOTS OF THE VECTORS J = 1,2,...,NP ARE
C                        GENERATED FROM DATA POINTS Y(1,J), Y(1+MULT,J),
C                        Y(1+2*MULT,J), ...., AND EACH POINT Y(I,J) IS AS-
C                        SUMED TO CORRESPOND TO THE DEPENDENT VARIABLE
C                        X(I) = X0 + (I-1)*MULT*DX.  IF DX = 0 AND AUTOMA-
C                        TIC X-AXIS SCALING IS REQUIRED, THE VECTOR X PRO-
C                        VIDED BY INPUT IS USED.
C
C         CLEAR       =  LOGICAL VARIABLE
C
C                              CLEAR = TRUE :  NORMAL PLOT
C                              CLEAR = FALSE:  HISTOGRAM
C
C         CENTER      =  LOGICAL VARIABLE TO CENTER PLOT ON PAGE.
C
C         SCALEX      =  LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF
```

```
C                        HORIZONTAL AXIS.
C
C       SCALEY      =  LOGICAL VARIABLE TO SPECIFY AUTOMATIC SCALING OF
C                      VERTICAL AXIS.
C
C       SAME        =  LOGICAL VARIABLE TO SPECIFY SAME SCALE ON THE LEFT
C                      AND RIGHT HAND VERTICAL AXES.
C
C       NAME(J)     =  VECTOR OF WORDS (10 BCD CHARACTERS) TO LABEL THE
C                      JTH VECTOR PLOTTED.
C
C       NP          =  NUMBER OF VECTORS TO BE PLOTTED (IF NP > 10, DE-
C                      FAULT TO ONLY 10 PLOTS OCCURS.)
C
C       IP          =  PLOT OPTION, FUNCTIONALLY SIMILAR TO THE USE OF
C                      THE FOUR ALTERNATE ENTRY POINTS:
C
C                      PLOT (IP = 0) -- Y VS X.
C                      PYLOGX (IP = -1) -- LOGY VS X.
C                      PXLOGY (IP = 2) -- Y VS LOGX.
C                      PLOGLOG (IP = 1) -- LOGY VS LOGX.
C
C ..................................................................
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), Z0(3),
     2 DZ(3), X(MULT,1), Y0(1), NAME(1)
C
      LOGICAL T1, T2, TEST, SCALEX, SCALEY, UNSCALE, CENTER, CLEAR,
     1 SAME, DIFF, SCALE, XLOG, YLOG
C
      INTEGER DOT
      DATA DOT / 1H*, 1H+, 1HX, 1H-, 1H., 1HO, 1HS, 1HE, 1H=, 1H  /
C
C     SET UP GENERAL CONTROL CONDITIONS --
C
      ISW = 0
      TEST = (IP-1)*(IP+1)*(IP-2).EQ.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
   40 NVEC = 10
      DXMULT = DX*MULT
      XLOG = IABS(ISW).EQ.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.
```

```
      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
```

```
C
      M1 = MA
      M2 = MB
      DO 34 I=1, NPLOT
      DO 34 M=M1, M2
      IF (YP(M,I).LE.0.) GO TO 98
   34 YP(M,I) = ALOG10(YP(M,I))
C
C     ESTABLISH MAXIMUM AND MINIMUM VALUES FOR EACH VECTOR --
C
   45 DO 1 I=1, NPLOT
      XA(I) = XB(I) = YP(MA,I)
      DO 2 M = MA,MB
      YMI = YP(M,I)
      IF(XA(I).GT.YMI) XA(I) = YMI
    2 IF(XB(I).LT.YMI) XB(I) = YMI
      IF (XA(I).GE.0.) GO TO 1
      IF (XB(I).LE.0.) GO TO 1
      A = ABS(XA(I) + XB(I))
      B = ABS(XA(I) - XB(I))
      IF ((B-A).GT.(A+B)/100.) GO TO 1
      B = XB(I)
      IF (B.GT.A) XA(I) = 0.
      IF (B.LT.A) XB(I) = 0.
    1 CONTINUE
      NSCALE = 1
C
C     FIND LARGEST XB(I) AND SMALLEST XA(I) --
C
      IF (NPLOT.EQ.1) GO TO 18
      NSCALE = 2
      XMIN = XA(1)
      XMAX = XB(1)
      DO 23 I = 2,NPLOT
      IF(XA(I).LT.XMIN) XMIN = XA(I)
   23 IF(XB(I).GT.XMAX) XMAX = XB(I)
      EPS = 1.0E-05*ABS(XMAX-XMIN)
      XMIN = XMIN + EPS
      XMAX = XMAX - EPS
C
C     SET UP Y-AXIS CONTROL (FOR SINGLE OR DOUBLE SCALING) --
C
      IF (.NOT.DIFF) GO TO 10
      Z0(2) = Y0(2)
      DZ(2) = DY(2)
   10 A = XB(1) - XA(1)
      B = XB(2) - XA(2)
      A = AMAX1(XB(1),A)
      B = AMAX1(XB(2),B)
      DS = AMIN1(A,B)
      XMULT = 10.
      IF (NPLOT.EQ.2) DIFF = DIFF.OR.((XMAX-XMIN).GT.XMULT*DS)
      IF(DIFF) GO TO 18
      NSCALE = 1
      DZ(2) = DZ(1)
      Z0(2) = Z0(1)
      XA(1) = XMIN
```

```
      XB(1) = XMAX
      XA(2) = XB(2) = 0.0
C
   18 DO 15 I = 1,NSCALE
      SCALE = SCALEY.OR.(DZ(I).EQ.0.0)
      IF (.NOT.CLEAR) XA(I) = 0.
      CALL AXIS (SCALE, XB(I), XA(I), Z0(I), DZ(I), XI)
      M0 = 0
      IF (.NOT.SCALE) GO TO 29
      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
   29 DS = M0*DZ(I)
      IF (XI.EQ.0.) GO TO 31
      XI = XI - DS
      Z0(I) = Z0(I) + DS
   31 Z0(I) = Z0(I) - DS
   15 XA(I) = XI
      IF(.NOT.DIFF) XA(2) = XA(1)
C     XA(1), XA(2) ARE DC BASELINE VALUES AT THIS POINT.
C
      DO 38 I=1, NPLOT
      XI = XA(1)
      IF(I.EQ.2) XI = XA(2)
      DO 38 M=M1,M2
   38 YP(M,I) = YP(M,I) - XI
C
      DO 11 I=1, NSCALE
   11 X1(I) = Z0(I) + 11.0*DZ(I)
      KX(1) = KX(2) = -1
      EPS = 0.02
      ETA = 0.5 + EPS
C
C     DETERMINE LOCATION OF HORIZONTAL AXES --
C
      DO 9 I = 1,NSCALE
      IF(DZ(I).EQ.0.0) GO TO 9
      KX(I) = -5.0*Z0(I)/DZ(I)*1.00001
    9 CONTINUE
C
C     CONVERT PLOTTING ARRAY YP TO NORMALIZE VALUES BETWEEN 0 AND 50
C     (CORRESPONDING TO LINES 1,2,3,...,51 FOR EACH PLOTTED PAGE) --
C
      DO 4 I = 1,NPLOT
      DXI = DZ(1)
      X0I = Z0(1)
      IF (I.NE.2) GO TO 88
      IF (.NOT.DIFF) GO TO 88
      X0I = Z0(2)
      DXI = DZ(2)
   88 CONTINUE
      DO 4 M = M1,M2
    4 YP(M,I) = ((YP(M,I)-X0I)/DXI)*5.0
C
C     DETERMINE NUMBER OF HORIZONTAL ELEMENTS PER LINE OF PLOT --
C
      IMAX = (M2 + 3)/5
```

```
      IMAX1 = IMAX+1
      MAX = 5*IMAX + 1
      MAX2 = 2*MAX
      N = 68 - MAX
      IF(.NOT.DIFF) N = N+5
      IF(.NOT.CENTER) N = 9
      N2 = N-2
      ENCODE(70,100,FORM) N, N2, MAX2, N
  100 FORMAT (1H(I2,7HX102A1),7H(1H+1PGI2,3H.4,I3,10HX,1PG14.4)5X,4H(1H+
     1 I2, 20HX,7X,A1,5H --- ,A10))
      ENCODE(90,101,CAPTION) N, N, N
  101 FORMAT(4H(1H+,I2,24HX4X17H    * --- LHS AXIS),5H(1H+,,I2, 23HX,22X,
     +* + (*,E12.4,*)*), 4H(1H+,I2,24HX4X17H    + --- RHS AXIS))
C
C     PLOT 51 LINES --
C
      LINE1 = 5
      DO 6 L=1, 51
      LL = L-1
      XLINE = LINE = 51-L
      T1 = (LINE.EQ.KX(1)).OR.(LINE.EQ.KX(2))
      T2 = (LL/50)*50.EQ.LL
      TEST = T1.OR.T2
      DO 5 M=1, MAX
      K(2,M) = 1H
      DO 42 I=1, NPLOT
      IF(K(1,M).EQ.DOT(I)) GO TO 44
   42 CONTINUE
      K(1,M) = 1H
   44 IF(CLEAR) K(1,M) = 1H
    5 IF(TEST) K(1,M) = K(2,M) = 1H-
      IF(.NOT.TEST) GO TO 14
      DO 8 I=1, 11
      I1 = 1 + 5*(I-1)
    8 K(1,I1) = 1HI
   14 K(1,1) = K(1,MAX) = 1HI
      IF(L.EQ.51) GO TO 22
      T2 = 5*(LL/5).EQ.LL
      IF(.NOT.(T1.OR.T2)) GO TO 24
      K(2,1) = K(2,MAX-1) = 1H-
      DO 62 I=1, NPLOT
      IF(K(1,M).EQ.DOT(I)) GO TO 64
   62 CONTINUE
      K(1,2) = 1H-
   64 DO 66 I=1, NPLOT
      IF(K(1,MAX-1).EQ.DOT(I)) GO TO 24
   66 CONTINUE
      K(1,MAX-1) = 1H-
   24 DO 3 I=1, NPLOT
      DO 3 M=M1, M2
      IF(ABS(XLINE-YP(M,I)).LT.ETA) K(1,M) = DOT(I)
      IF((M.EQ.M2).OR.(.NOT.CLEAR)) GO TO 3
      XAVG = (YP(M,I)+YP(M+1,I))/2.0
      IF(ABS(XLINE-XAVG).LT.ETA) K(2,M) = DOT(I)
      X1AVG = 0.5*(XAVG + YP(M,I))
      X2AVG = 0.5*(XAVG + YP(M+1,I))
      IF(ABS(XLINE-X1AVG).LT.ETA) K(2,M) = DOT(I)
```

```
      IF(ABS(XLINE-X2AVG).LT.ETA) K(2,M) = DOT(I)                      PLOT
      XMAX = AMAX1(X1AVG,X2AVG)                                        PLOT
      XMIN = AMIN1(X1AVG,X2AVG)                                        PLOT
      IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(2,M) = DOT(I)          PLOT
      XMAX = AMAX1(X1AVG,YP(M,I))                                      PLOT
      XMIN = AMIN1(X1AVG,YP(M,I))                                      PLOT
      IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M) = DOT(I)          PLOT
      XMAX = AMAX1(X2AVG,YP(M+1,I))                                    PLOT
      XMIN = AMIN1(X2AVG,YP(M+1,I))                                    PLOT
      IF((XLINE.GE.XMIN).AND.(XLINE.LE.XMAX)) K(1,M+1) = DOT(I)        PLOT
    3 CONTINUE                                                         PLOT
   22 K(2,MAX) = 1H                                                    PLOT
      M0 = 12                                                          PLOT
      TEST = (L.LT.49).AND.(L.GT.44)                                   PLOT
      IF(.NOT.TEST) GO TO 20                                           PLOT
      IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 20                          PLOT
      IF((XA(1).NE.0.0).OR.(XA(2).NE.0.0)) M0 = 21                     PLOT
      GO TO 21                                                         PLOT
   20 IF (NPLOT.EQ.1) GO TO 13                                         PLOT
      IF (LINE1.NE.L) GO TO 13                                         PLOT
      L1 = L-4                                                         PLOT
      IF (L1.GT.NPLOT) GO TO 13                                        PLOT
      LINE1 = LINE1 + 1                                                PLOT
   21 DO 19 M=3, M0                                                    PLOT
   19 K(1,M) = K(2,M) = 1H                                             PLOT
   13 WRITE(6,FORM(1)) (K(1,M), K(2,M), M=1, MAX)                      PLOT
      IF (L.LE.4) GO TO 16                                             PLOT
      IF (LINE1.EQ.(L+1)) WRITE (6,FORM(5)) DOT(L1), NAME(L1)          PLOT
   16 IF(.NOT.DIFF.AND.XA(1).EQ.0.0) GO TO 25                          PLOT
      IF(L.NE.46) GO TO 28                                             PLOT
      WRITE(6,CAPTION(1))                                              PLOT
      IF(XA(1).NE.0.0) WRITE(6,CAPTION(4)) XA(1)                       PLOT
   28 IF(L.NE.47) GO TO 25                                             PLOT
      IF(.NOT.DIFF) GO TO 25                                           PLOT
      WRITE(6,CAPTION(7))                                              PLOT
      IF(XA(2).NE.0.0) WRITE(6,CAPTION(4)) XA(2)                       PLOT
   25 IF((LL/5)*5.NE.LL) GO TO 6                                       PLOT
      DO 17 I = 1,NSCALE                                               PLOT
      X1(I) = X1(I) - DZ(I)                                            PLOT
      IF(ABS(X1(I)).LT.0.001*DZ(I)) X1(I) = 0.0                        PLOT
   17 XB(I) = X1(I)                                                    PLOT
      WRITE(6,FORM(2)) (XB(I), I=1, NSCALE)                            PLOT
    6 CONTINUE                                                         PLOT
      N8 = N-8                                                         PLOT
      IF(UNSCALE.AND.(DX.EQ.0.0)) GO TO 30                             PLOT
      AXMAX = ABS(Z0(3) + IMAX*DZ(3))                                  PLOT
      IF(AXMAX.EQ.0.0) AXMAX = 10.0*ABS(DZ(3))                         PLOT
      MX = A = ALOG10(AXMAX)                                           PLOT
      MX = MX+1                                                        PLOT
      IF (A.LT.0.) MX = MX-1                                           PLOT
      SX = 10.0**MX                                                    PLOT
      IF((MX.LT.4).AND.(MX.GE.0)) SX = 1.0                             PLOT
      DO 7 I=1, IMAX1                                                  PLOT
      TICK(I) = Z0(3) + (I-1)*DZ(3)                                    PLOT
    7 TICK(I) = TICK(I)/SX                                             PLOT
      SX = TICK(2)                                                     PLOT
      IF (SX.EQ.0.) SX = TICK(3)                                       PLOT
```

```
      SX = ABS(SX)
      MO = ALOG10(SX)
      IF(MO.GE.0) N8 = N8 + 1
      ENCODE(30,102,FORM) N8
  102 FORMAT(2H(/,I2,2HX,,8H11F10.2))
      WRITE (6,FORM) (TICK(I), I=1,IMAX1)
      IF((MX.GE.4).OR.(MX.LT.0)) WRITE(6,103) MX
  103 FORMAT (/68X,7H(X 10**,I3,*)*)
      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)
      LOGICAL SCALE
C
C  ..................................................................
C
C     XMAX AND XMIN ARE THE LARGEST AND SMALLEST VALUES OF THE VECTOR TO
C     BE SCALED WITH CONVENIENT ORIGIN X0 AND TICK MARK SPACING DX.  XI
C     IS SET EQUAL TO ZERO UNLESS THE AC RANGE 10*DX ≤ X0/1000.  IN THAT
C     CASE XDC IS SET EQUAL TO X0, X0 IS SET EQUAL TO ZERO, AND XMAX AND
C     XMIN ARE REDUCED BY AN AMOUNT XDC.  THUS, XDC CORRESPONDS TO A DC
C     BASELINE THAT IS RETURNED NONZERO ONLY IF THE RANGE OF THE PLOT IS
C     VERY SMALL RELATIVE TO THE ABSOLUTE MAGNITUDE OF PLOTTED VALUES.
C
C  ..................................................................
C
      IF (.NOT.SCALE) GO TO 3
C
C     X0 IS CONVENIENT ORIGIN, AND RANGE 10*DX IS 1, 2, OR 5 TIMES SOME
C     POWER OF 10 --
C
      IF (ABS(XMAX-XMIN).GT.1.0E-10*(XMAX+XMIN)) GO TO 1
      XMIN = 0.
      IF (XMAX.EQ.0.) XMAX = 1.0
    1 XA = XMIN
    4 B = XMAX - XA
      M = A = ALOG10(B)
      IF (A.LT.0.) M = M-1
      DS = 10.**M
      M = B/DS + 1.
      B = 2.
      IF (M.GT.2) B = 5.
      IF (M.GT.5) B = 10.
      B = B*DS
      DS = B/10.
      M = A = XA/DS
      IF (A.LT.0.) M = M-1
      XA = M*DS
      IF ((XA+B).LT.XMAX) GO TO 4
      DX = DS
      X0 = XA
C     CALCULATE DC BASELINE VALUE XDC --
    3 XDC = 0.
      IF (ABS(X0).LT.1.E 04*DX) GO TO 2
      XMIN = XMIN - X0
      XMAX = XMAX - X0
      XDC = X0
      X0 = 0.
    2 IF (.NOT.SCALE) GO TO 99
C     ALLOW DATA TO BE PLOTTED WITH ZERO ORIGIN (AT TOP OR BOTTOM) IF
C     POSSIBLE --
      X1 = X0 + 10.*DX
      IF ((XMAX.LT.10.*DX).AND.(X0.GT.0.)) X0 = 0.
      IF ((XMIN.GT.-10.*DX).AND.(X1.LT.0.)) X0 = -10.*DX
   99 RETURN
      END
```

```
      SUBROUTINE INTERP (IDEG, XP, YP, X, Y, MULT, N)                   INTERP
C                                                                       INTERP
C     ..................................................................INTERP
C                                                                       INTERP
C     THIS SUBROUTINE WILL INTERPOLATE A VECTOR Y, DEFINED AS A FUNCTION INTERP
C     OF THE VECTOR X, TO PRODUCE THE VALUE YP THAT CORRESPONDS TO XP.   INTERP
C     IDEG = 1, 2, ... SPECIFIES LINEAR, QUADRATIC, ... ETC. INTERPOLA-  INTERP
C     TION.  THE VECTORS X AND Y PROVIDED BY THE CALLING PROGRAM ARE     INTERP
C     SAMPLED WITH A REPETITION INDEX = MULT, AND N POINTS DEFINED BY    INTERP
C     I = 1, (1+MULT), ..., (1 + (N-1)*MULT) WILL BE UTILIZED.  THUS,    INTERP
C     THE VECTORS X, Y MUST BE DIMENSIONED AT LEAST (1+(N-1)*MULT)       INTERP
C     IN THE CALLING PROGRAM.  VALUES FOR THE INDEPENDENT VECTOR X MUST  INTERP
C     BE IN ASCENDING ORDER.  XP NEED NOT LIE WITHIN THE RANGE (X(1),... INTERP
C     X(1+(N-1)*MULT), BUT IF IT FALLS OUTSIDE, LINEAR INTERPOLATION IS  INTERP
C     GIVEN. (THE DEFAULT TO LINEAR INTERPOLATION CAN BE REMOVED BY      INTERP
C     DELETION OF ONE CARD BELOW.) NORMAL USAGE IS MULT = 1.             INTERP
C                                                                       INTERP
C     ..................................................................INTERP
C                                                                       INTERP
      DIMENSION T(20), X(MULT,1), Y(MULT,1)                             INTERP
      INT = IDEG                                                        INTERP
      IF (INT.LE.0) INT = 1                                             INTERP
      IF (INT.GE.N) INT = N-1                                           INTERP
      IF (N.LT.2) GO TO 10                                              INTERP
      DO 4 I = 1,N                                                      INTERP
      J = I                                                             INTERP
      IF (XP.LE.X(1,I)) GO TO 1                                         INTERP
    4 CONTINUE                                                          INTERP
    1 CONTINUE                                                          INTERP
C     DEFAULT TO LINEAR INTERPOLATION IF XP LIES OUTSIDE RANGE OF X --  INTERP
      IF ((J.EQ.1).OR.(J.EQ.N)) INT = 1                                 INTERP
      K = INT                                                           INTERP
      INT = INT + 1                                                     INTERP
      J = J - INT/2                                                     INTERP
      J = MAX0(J,1)                                                     INTERP
      J = MIN0(J,N-K)                                                   INTERP
      JK = J+K                                                          INTERP
      DO 2 I = J,JK                                                     INTERP
      KK = I-J+1                                                        INTERP
      T(KK) = Y(1,I)                                                    INTERP
    2 T(KK+INT) = X(1,I) - XP                                           INTERP
      DO 3 I = 1,K                                                      INTERP
      IP1 = I + 1                                                       INTERP
      DO 3 JJ = IP1,INT                                                 INTERP
    3 T(JJ) = (T(I)*T(JJ+INT)-T(JJ)*T(I+INT))/(X(1,JJ+J-1)-X(1,I+J-1))  INTERP
      YP = T(INT)                                                       INTERP
      RETURN                                                            INTERP
   10 YP = Y(1,1)                                                       INTERP
      RETURN                                                            INTERP
      END                                                               INTERP
```

```
      SUBROUTINE SIMPSON (F, M, H, ANS)                                 SIMPSON
C                                                                       SIMPSON
C     ..................................................................SIMPSON
C                                                                       SIMPSON
C  THIS SUBROUTINE INTEGRATES A FUNCTION F DEFINED AS A VECTOR OVER N = SIMPSON
C  2M INTERVALS, F(1), F(2), F(3), ..., F(2M+1).  THE WIDTH OF THE SUB- SIMPSON
C  INTERVALS IS H, AND THE ANSWER IS RETURNED IN ANS.                   SIMPSON
C                                                                       SIMPSON
C     ..................................................................SIMPSON
C                                                                       SIMPSON
      DIMENSION F(1)                                                    SIMPSON
      N = M + M                                                         SIMPSON
      SUM = F(N+1) - F(1)                                               SIMPSON
      DO 1 I = 1,N,2                                                    SIMPSON
      I1 = I+1                                                          SIMPSON
      Y = F(I) + F(I1) + F(I1)                                          SIMPSON
    1 SUM = SUM + Y + Y                                                 SIMPSON
      ANS = H*SUM/3.                                                    SIMPSON
      RETURN                                                            SIMPSON
      END                                                               SIMPSON
```

```
      SUBROUTINE EDITOR (INPUT, LIST)
C
C  ..............................................................
C
C  THIS SUBROUTINE READS AN INPUT CARD FILE TO THE EOF, AND WRITES IT
C  ONTO TAPE 5.  IF LIST = .TRUE. IS SPECIFIED, IT ALSO PRODUCES A
C  WRITTEN OUTPUT LISTING OF THE INPUT CARD IMAGES.
C
C  ..............................................................
C
      DIMENSION IMAGE(8)
      LOGICAL LIST
      DATA SKIP, K / 5H(1H1), 1234567890 /
      REWIND 5
      KARD = 0
      CALL DATE (MONTH)
    1 READ (INPUT,150) IMAGE
      IF (EOF(INPUT)) 5,3
    3 IF (MOD(KARD,40).NE.0) GO TO 2
      IF (.NOT.LIST) GO TO 2
      IF (KARD.EQ.0) GO TO 4
      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,*(D
     1ATE:*,A9,*)*//)
      WRITE (6,100) (I, I = 1,8), (K, I = 1,8)
  100 FORMAT (22X,*CARD*,13X,8(I1,9X)/23X,*NO.*,4X,8I10//21X,92(1H-)/)
    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
      REWIND 5
      IF (KARD.EQ.0) GO TO 99
      IF (.NOT.LIST) GO TO 99
      WRITE (6,110) (K, I = 1,8), (I, I = 1,8)
  110 FORMAT (/21X,92(1H-)//30X,8I10/39X,8(I1,9X))
   99 RETURN
      END
```

```
      SUBROUTINE COVER (TITLE, NPAGE)
      DIMENSION TITLE(3), MESSAGE(10,3)
      INTEGER BLANK, TITLE
C
C  .................................................................
C
C     THIS SUBROUTINE WILL GENERATE NPAGE SEQUENTIAL TITLE PAGES FOR THE
C     OUTPUT PRINTOUT.
C
C  .................................................................
C
      BLANK = 1H
      N = 0
      DO 2 I = 1,3
      IF (TITLE(I).EQ.BLANK) GO TO 2
      N = N+1
      TITLE(N) = TITLE(I)
      DECODE (10,100,TITLE(N)) (MESSAGE(J,N), J = 1,10)
  100 FORMAT (10A1)
    2 CONTINUE
C
      NSKIP = 3 + 6*(3-N)
      ENCODE (10,150,KONTROL) NSKIP
  150 FORMAT (1H1,I2)
      NSKIP = KONTROL
C
      DO 1 K = 1,NPAGE
      J = 0
      KONTROL = NSKIP
    3 IF (J.EQ.N) GO TO 1
      J = J+1
      CALL HEADINX (1HX, KONTROL, MESSAGE(1,J))
      KONTROL = BLANK
      GO TO 3
    1 WRITE (6,200)
C
  200 FORMAT (////
     1/39X,52H     ****************************************************
     2/39X,52H     *                                                  *
     3/39X,52H     *                                                  *
     4/39X,52H     *    THIS CODE WAS DEVELOPED BY                    *
     5/39X,52H     *                                                  *
     6/39X,52H     *        DR. WILLIAM B. LACINA                     *
     7/39X,52H     *        NORTHROP RESEARCH AND TECHNOLOGY          *
     8/39X,52H     *        ONE RESEARCH PARK                         *
     9/39X,52H     *        PALOS VERDES PENINSULA, CA 90274          *
     $/39X,52H     *        TEL: (213) 377-4811, EXT. 322             *
     1/39X,52H     *                                                  *
     2/39X,52H     *                                                  *
     3/39X,52H     ****************************************************)
C
      RETURN
      END
```

```
      SUBROUTINE HEADINX (JSYMB, JPAGE, MESSAGE)
C
C     ..................................................................
C
C     THE CALL SEQUENCE FOR THIS SUBROUTINE IS EQUIVALENT TO THAT SUPPLIED
C     BY THE CDC CYBERNET SYSTEM (WITH THE SAME NAME) FOR THE GENERATION OF
C     BLOCK HEADINGS.  HOWEVER, THE PRESENT VERSION HAS TWO ADDITIONAL AD-
C     VANTAGES:
C
C         1) THE 10-BCD CHARACTER WORD STORED IN THE VECTOR MESSAGE(I)
C            (I = 1,2,...,10) IS AUTOMATICALLY CENTERED ON THE PRINTED
C            LINE, AND
C
C         2) THE CARRIAGE CONTROL SYMBOL *JPAGE* CAN ACCEPT MORE GENERAL
C            SPECIFICATIONS TO CONTROL THE SPACING OF THE PRINTED LINE.
C            JPAGE CAN HAVE THE USUAL FORMAT 1HS (WHERE S = 0, 1, 2, +,
C            ETC), OR IT CAN HAVE A MORE GENERAL FORMAT 3HSNN, WHERE NN
C            IS A TWO-DIGIT NUMBER.  AFTER THE PAGE CONTROL S IS EXECUTED,
C            TWO LINES ARE AUTOMATICALLY SPACED, FOLLOWED BY AN ADDITIONAL
C            NN LINES.  THE REMAINING SEVEN BITS OF *JPAGE* ARE IGNORED.
C
C     UNLIKE THE CDC SUBROUTINE, THE PLOTTING SYMBOL *JSYMB* IS IGNORED,
C     AND THE CHARACTER X IS ALWAYS USED.
C
C     ..................................................................
C
      DIMENSION KAR(10,50), FORM(3), MESSAGE(10), LETTER(50), NUM(10),
     1 IMAGE(10), KK(500)
C
      INTEGER BLANK
      EQUIVALENCE (KAR, KK)
C
      DATA LETTER / 1HA, 1HB, 1HC, 1HD, 1HE, 1HF, 1HG, 1HH, 1HI, 1HJ,
     1 1HK, 1HL, 1HM, 1HN, 1HO, 1HP, 1HQ, 1HR, 1HS, 1HT, 1HU, 1HV, 1HW,
     2 1HX, 1HY, 1HZ, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0,
     3 1H+, 1H-, 1H/, 1H*, 1H:, 1H$, 1H=, 1H(, 1H), 1H<, 1H>, 1H[, 1H],
     4 1H /
C
      DATA (KK(I), I = 1,150)
     $            / 10H XXXXXXXX , 10HXXXXXXXXXX, 3*10HXX      XX,
     1 2*10HXXXXXXXXXX, 3*10HXX      XX, 10HXXXXXXXXX , 10HXXXXXXXXXX,
     2 2*10HXX      XX, 2*10HXXXXXXXXX , 2*10HXX      XX, 10HXXXXXXXXXX,
     3 10HXXXXXXXXX , 10H XXXXXXXX , 10HXXXXXXXXXX, 10HXX      XX,
     4 4*2HXX, 10HXX      XX, 10HXXXXXXXXXX, 10H XXXXXXXX ,9HXXXXXXXXX,
     5 10HXXXXXXXXXX, 6*10HXX      XX, 10HXXXXXXXXXX, 9HXXXXXXXXX,
     6 2*10HXXXXXXXXXX, 2*2HXX, 2*6HXXXXXX, 2*2HXX, 4*10HXXXXXXXXXX,
     7 2*2HXX,2*6HXXXXXX,4*2HXX,9H XXXXXXXX,10HXXXXXXXXXX,10HXX      XX,
     8 2HXX,2*10HXX   XXXXX,2*10HXX      XX,10HXXXXXXXXXX,9H XXXXXXXX,
     9 4*10HXX      XX, 2*10HXXXXXXXXXX, 4*10HXX      XX, 2*9H XXXXXXXX,
     $ 6*6H    XX,2*9H XXXXXXXX,2*10H   XXXXXXX,4*8H      XX, 2*8HXX
     1XX,8HXXXXXXXX, 7H XXXXXX, 10HXX      XX, 9HXX     XX, 7HXX   XX,
     2 5HXX XX, 5HXXXXX, 6HXX  XX, 7HXX   XX, 8HXX    XX, 9HXX     XX,
     3 10HXX      XX, 8*2HXX, 2*10HXXXXXXXXXX, 10HXX      XX,
     4 10HXXXX   XXX, 10HXXXX  XXXX, 2*10HXX XXXX XX, 2*10HXX  XX  XX,
     5 4*10HXX      XX, 10HXXX     XX, 10HXXXX    XX, 10HXX XX   XX,
     6 2*10HXX  XX  XX, 10HXX   XX XX, 10HXX    XXXX, 10HXX     XXX,
     7 10HXX      XX, 9H XXXXXXXX, 10HXXXXXXXXXX, 6*10HXX      XX,
```

```
      8 10HXXXXXXXXXX, 9H XXXXXXXX /                                HEADINX
C                                                                   HEADINX
       DATA (KK(I), I = 151,280)                                    HEADINX
      $              /  9HXXXXXXXXX, 10HXXXXXXXXXX, 2*10HXX       XX, HEADINX
      1 10HXXXXXXXXXX, 9HXXXXXXXXX, 4*2HXX, 9H XXXXXXXX, 10HXXXXXXXXXX, HEADINX
      2 4*10HXX        XX, 10HXX  XXX XX, 10HXX    XXXX, 9HXXXXXXXXX, HEADINX
      3 10H XXXXXX XX, 9HXXXXXXXXX, 10HXXXXXXXXXX, 2*10HXX       XX, HEADINX
      4 10HXXXXXXXXXX, 9HXXXXXXXXX, 7HXX   XX, 8HXX    XX, 9HXX     XX, HEADINX
      5 10HXX        XX, 9H XXXXXXXX, 10HXXXXXXXXXX, 10HXX      XX, 2HXX, HEADINX
      6 9HXXXXXXXXX, 10H XXXXXXXXX, 10H         XX, 10HXX      XX,  HEADINX
      7 10HXXXXXXXXXX, 9H XXXXXXXX, 2*10HXXXXXXXXXX, 8*6H    XX,    HEADINX
      8 8*10HXX        XX, 10HXXXXXXXXXX, 9H XXXXXXXX, 2*10HXX      XX, HEADINX
      9 4*10H XX     XX , 2*8H  XX  XX, 7H   XXXX, 6H    XX,        HEADINX
      $ 4*10HXX        XX, 3*10HXX  XX  XX, 10HXX XXXX XX, 10H XXX  XXX , HEADINX
      1 9H XX     XX, 10HXX        XX, 9H XX     XX, 8H  XX  XX, 7H   XXXX, HEADINX
      2 6H    XX, 7H   XXXX, 8H  XX  XX, 9H XX     XX, 3*10HXX       XX, HEADINX
      3 9H XX     XX, 8H  XX  XX, 7H   XXXX, 6*6H    XX, 10HXXXXXXXXXX, HEADINX
      4 9HXXXXXXXXX, 8H        XX, 7H     XX, 2*6H    XX, 5H   XX, 4H  XX, HEADINX
      5 10H XXXXXXXXX, 10HXXXXXXXXXX, 6H    XX, 6H  XXXX, 6H XX XX, HEADINX
      6 5*6H    XX, 2*8H  XXXXXX, 9H XXXXXXXX, 10HXXXXXXXXXX,       HEADINX
      7 10HXX        XX, 10H        XXX, 9H      XXX, 7H     XX, 5H   XX, HEADINX
      8 4H XXX, 2*10HXXXXXXXXXX /                                   HEADINX
C                                                                   HEADINX
       DATA (KK(I), I = 281,410)                                    HEADINX
      $              /  10HXXXXXXXXXX, 9HXXXXXXXXX, 8H      XX, 7H    XXX, HEADINX
      1 7H  XXXXX, 9H     XXXX, 10H        XXX, 10HXX      XXX,9HXXXXXXXXX, HEADINX
      2 8H XXXXXXX, 8H     XXX, 8H    XXXX, 8H   XX XX, 8H  XX  XX, HEADINX
      3 8H XX   XX, 2*10HXXXXXXXXXX, 3*8H      XX, 2*10HXXXXXXXXXX, HEADINX
      4 2*2HXX, 9HXXXXXXXXX, 10HXXXXXXXXXX, 10H        XX, 10HXX       XX, HEADINX
      5 10HXXXXXXXXXX, 2*9H XXXXXXXX, 10HXXXXXXXXXX, 10HXX       XX, 2HXX, HEADINX
      6 9HXXXXXXXXX, 10HXXXXXXXXXX, 2*10HXX       XX, 10HXXXXXXXXXX, HEADINX
      7 9H XXXXXXXX, 2*10HXXXXXXXXXX, 9H       XXX, 7H    XXX, 6H   XXX, HEADINX
      8 5H  XXX, 4H XXX, 3*3HXXX, 9H XXXXXXXX, 10HXXXXXXXXXX,        HEADINX
      9 2*10HXX        XX, 2*9H XXXXXXXX, 2*10HXX       XX, 10HXXXXXXXXXX, HEADINX
      $ 2*9H XXXXXXXX, 10HXXXXXXXXXX, 2*10HXX       XX, 10HXXXXXXXXXX, HEADINX
      1 10H XXXXXXXXX, 10H        XX, 10HXX       XX, 10HXXXXXXXXXX, HEADINX
      2 9H XXXXXXXX, 7H   XXXX, 9H XXXXXXXX, 9H XX     XX,4*10HXX       XX, HEADINX
      3 9H XX     XX, 9H XXXXXXXX, 7H   XXXX,1H ,3*6H    XX,2*9H XXXXXXXX, HEADINX
      4 3*6H    XX, 5*1H ,2*9H XXXXXXXX, 5*1H , 9H       XX, 8H      XX, HEADINX
      5 7H     XX,6H    XX,5H   XX, 4H  XX, 3H XX, 2HXX, 2*1H , 6H    XX, HEADINX
      6 9H XX XX XX, 8H  XXXXXX, 2*7H   XXXX, 8H  XXXXXX, 9H XX XX XX, HEADINX
      7 6H    XX, 3*1H , 2*6H    XX, 2*1H , 2*6H    XX, 2*1H  /     HEADINX
C                                                                   HEADINX
       DATA (KK(I), I = 411,500)                                    HEADINX
      $              / 6H    XX, 9H XXXXXXXX, 10HXX  XX  XX, 6HXX  XX, HEADINX
      1 9HXXXXXXXXX, 10H XXXXXXXXX,10H    XX  XX, 10HXX  XX  XX,    HEADINX
      2 9HXXXXXXXXX,6H    XX, 2*1H , 2*9H XXXXXXXX, 2*1H , 2*9H XXXXXXXX, HEADINX
      3 2*1H , 7H     XX,7H    XXX, 5H   XX, 4*4H  XX, 5H   XX, 6H   XXX, HEADINX
      4 7H     XX, 5H   XX, 6H   XXX, 7H     XX, 4*8H      XX, 7H     XX, HEADINX
      5 6H   XXX, 5H   XX,8H      XX, 7H     XX, 6H    XX, 5H   XX, HEADINX
      6 2*4H  XX, 5H   XX, 6H    XX, 7H     XX, 8H      XX, 4H  XX, HEADINX
      7 5H   XX, 6H    XX, 7H     XX, 2*8H      XX, 7H     XX, 6H    XX, HEADINX
      8 5H   XX, 4H  XX, 7H   XXXX, 8*5H   XX, 2*7H   XXXX, 8*7H     XX, HEADINX
      9 7H   XXXX, 10*1H  /                                         HEADINX
C                                                                   HEADINX
       NSYMBOL = 50                                                 HEADINX
       BLANK = 1H                                                   HEADINX
```

```
      L1 = L2 = 0
      DO 5 L = 1,10
      IF (MESSAGE(L).NE.BLANK) GO TO 6
    5 L1 = L1+1
    6 DO 7 I = 1,10
      L = 11-I
      IF (MESSAGE(L).NE.BLANK) GO TO 8
    7 L2 = L2+1
    8 LB = L1 + L2
      IF (LB.GE.10) L1 = LB = 0
      L1P1 = L1 + 1
      NK = 10 - LB
      LB = (13*LB)/2
      NKPL1 = NK + L1
      DECODE (10,102,JPAGE) KSKIP
  102 FORMAT (1X,A2,7X)
      IF (KSKIP.EQ.BLANK) GO TO 9
      ENCODE (10,101,KONTROL) JPAGE
  101 FORMAT (A3,7H(/)//1X)
      GO TO 10
    9 ENCODE (10,103,KONTROL) JPAGE
  103 FORMAT (A1,9H//1X      )
   10 DO 1 L = L1P1,NKPL1
      DO 2 I = 1,NSYMBOL
      IF (MESSAGE(L).EQ.LETTER(I)) GO TO 1
    2 CONTINUE
      I = NSYMBOL
    1 NUM(L) = I
      DO 3 LINE = 1,10
      DO 4 L = L1P1, NKPL1
      N = NUM(L)
    4 IMAGE(L) = KAR(LINE,N)
      ENCODE (30,100,FORM) KONTROL, LB, NK
  100 FORMAT (3H(1H,A10,I2,1HX,I2,9H(3X,A10)))
      WRITE (6,FORM) (IMAGE(K), K = L1P1, NKPL1)
    3 KONTROL = 1H
      RETURN
      END
```

```
      PROGRAM ELECT (INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT,TAPE7,TAPE8,TAPE9,
     1 TAPE10=INPUT)
C
C     ------------------------------------------------------------------
C
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 E/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               ******************************************
C               *                                        *
C               *                                        *
C               *      THIS CODE WAS DEVELOPED BY        *
C               *                                        *
C               *           DR. WILLIAM B. LACINA        *
C               *           NORTHROP RESEARCH AND TECHNOLOGY *
C               *           ONE RESEARCH PARK            *
C               *           PALOS VERDES PENINSULA, CA 90274 *
C               *           TEL: (213) 377-4811, EXT. 322 *
C               *                                        *
C               *                                        *
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 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     ------------------------------------------------------------------
C
      DIMENSION Q(1001,30), QM(1001), QMOM(1001,2), A(1001,3), F(1001),
     1 EV(1001), G(1001), B(1001), POWER(30), N1(30), N2(30), P(21,31),
     2 U(30), NEL(30), RATE(21,2,30), VSIG(2,30), PROCESS(4,31), FI(5),
     3 IDENT(5), NAME(5), MASS(5), GAS(100), NO(100), E(100), TITLE(3),
     4 YO(2), DY(2), TABLE(21,9), FORM(15), KODE(8), HEAD(9), IMAGE(60),
     5 EOVERN(21), EN(21), IOUT(10), OUT(10), KAPT(5), LABEL(5,2),
     6 KINETIC(6,50), NUMBER(30), GMOLE(100), LINE(250), S(1001)
C
      REAL NO, KB, KTE, MU, MASS, NMOL, N1, N2, NE, MOLWT, IONIZE
C
      INTEGER GAS, TYPE, TITLE, TODAY, LHS, RHS
C
      LOGICAL CONVRGE, OUT, FATAL, LIST, ERROR, STOP, TEST, MISSING,
     1    OUTSIDE, REJECT, EXPAND, SEARCH, LIBRARY
C
      EQUIVALENCE (B,QM)
```

```
C
C ----------------------------------------------------------------------
C
      NAMELIST / CONTROL / MESH, IOUT, FATAL, EMAX, ITMAX, TMAX, EPS,
     1   TE, IDEG, PCT, SEARCH
C
      NAMELIST / PARAM / THOL, PTOT, ATM, EOVERN
C
      NAMELIST / SOURCE / DNEDT, BEAM, CREATE, UA, UB, S
C
C ----------------------------------------------------------------------
C
      DATA KB, EE / 1.38E-23, 1.602E-19 /
C
      DATA EN / .1, .2, .3, .5, .8, 1., 2., 3., 5., 12*0. /
C
      DATA HEAD / 6HE/NTOT, 10H<U>= 3TE/2, 9HEK = D/MU, 10HTE (DEG K),
     1            9HVD (CM/S), 10HMU,CM2/V/S, 9HD,  CM2/S, 9HP/NE/NMOL,
     2            9HJ/NE=E*VD /
C
      DATA TITLE, KAPT / 8HELECTRON, 8HKINETICS, 8HANALYSIS, 5*1H /
C
C ----------------------------------------------------------------------
C
C     DATA INPUT SEQUENCE --
C
C
C         IOUT(10) IS A VECTOR OF OUTPUT OPTIONS  --  OPTION I IS SUP-
C         PRESSED IF IOUT(I) = 0, OTHERWISE PROVIDED --
C
C             IOUT(1) --  INDIVIDUAL SUMMARY OF PLASMA PARAMETERS FOR
C                         EACH E/N VALUE
C             IOUT(2) --  TABLE OF ELECTRON DISTRIBUTION FUNCTION FOR
C                         EACH VALUE OF E/N
C             IOUT(3) --  PLOT OF ELECTRON DISTRIBUTION FUNCTION FOR
C                         EACH VALUE OF E/N
C             IOUT(4) --  PLOT OF F(U)/FBOLTZ(U,TE) FOR EACH E/N
C             IOUT(5) --  SUMMARY OF PLASMA PARAMETERS, COLLISION
C                         RATES, AND POWER BALANCE FOR EACH E/N
C             IOUT(6) --  TABULAR SUMMARY OF INPUT CROSS SECTION DATA
C             IOUT(7) --  PLOTS OF INPUT CROSS SECTIONS (IOUT(6) MUST
C                         BE SIMULTANEOUSLY SPECIFIED)
C             IOUT(8) --  TABULAR SUMMARY OF PLASMA PARAMETERS AS A
C                         FUNCTION OF E/N FOR THE GIVEN GAS MIXTURE;
C                         TABLES OF POWER PARTITIONING AS A FUNCTION OF
C                         E/N FOR ELASTIC AND INELASTIC PROCESSES.
C             IOUT(9) --  PLOTS OF PLASMA PARAMETERS AS A FUNCTION OF
C                         E/N FOR A GIVEN GAS MIXTURE (IF AT LEAST 5
C                         CASES HAVE BEEN COMPUTED)
C             IOUT(10)--  LOG PLOTS OF FORWARD (AND REVERSE) ELECTRON
C                         RATES AS A FUNCTION OF E/N, FOR EACH OF THE
C                         INELASTIC COLLISION PROCESSES INCLUDED
C
C     MISCELLANEOUS PARAMETERS DEFINED --
C
C         EMAX    --  MAXIMUM ELECTRON ENERGY (EV)
C         MESH    --  NUMBER OF INTERVALS INTO WHICH THE ENERGY RANGE
```

```
C                 IS DIVIDED. (MESH ≤ 1000 IS INSURED BY PROGRAM.)  ELECT
C        EVCM    --  ELECTRIC FIELD (VOLT/CM)                         ELECT
C        EOVERN  --  E/N (1.E-16 VOLT CM2)                            ELECT
C        TE      --  INITIAL GUESS FOR ELECTRON TEMPERATURE (DEG K)   ELECT
C        ITMAX   --  MAXIMUM NUMBER OF ITERATIONS PERMITTED.          ELECT
C        TMAX    --  MAXIMUM CP TIME FOR OBTAINING CONVERGENCE.       ELECT
C        EPS     --  CONVERGENCE CRITERION FOR MAXIMUM RELATIVE       ELECT
C                    CHANGE FOR ALL VALUES OF THE ELECTRON DISTRIBU-  ELECT
C                    TION FUNCTION BETWEEN SUCCESSIVE ITERATIONS.     ELECT
C        IDEG    --  DEGREE OF INTERPOLATION FOR CROSS SECTIONS TO    ELECT
C                    GENERATE UNIFORM GRID OF VALUES FROM RAW DATA.   ELECT
C                                                                     ELECT
C     INPUT CONSISTS OF THE FOLLOWING --                              ELECT
C                                                                     ELECT
C        A)  DECK OF ELECTRON CROSS SECTION PACKAGES (ARBITRARILY MANY)  ELECT
C            FOR UPDATE OF ELECTRON CROSS SECTION DATA FILE, TERMINATED  ELECT
C            BY AN EOF (7/8/9) CARD.                                  ELECT
C                                                                     ELECT
C        B)  LIST OF (UP TO LIMIT) ELECTRON KINETIC REACTIONS, TERMI-  ELECT
C            NATED BY AN EOF (7/8/9) CARD.  THESE REACTIONS FORM THE  ELECT
C            BASIS FOR ALL SUBSEQUENT ELECTRON KINETICS CALCULATIONS.  ELECT
C            HOWEVER, IF NO REACTIONS ARE ENTERED IN THIS FILE, THE PRO-  ELECT
C            GRAM WILL SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE  ELECT
C            AND RETAIL ALL RELEVANT REACTIONS (I.E. WHICH CONTAIN GASES  ELECT
C            IN THE SPECIFIED MIXTURE).  REGARDLESS OF WHERE THE KINETIC  ELECT
C            SCHEME IS FOUND, ONLY THE FIRST NKMAX LEGAL REACTIONS ARE  ELECT
C            RETAINED.                                                ELECT
C                                                                     ELECT
C     THESE ARE FOLLOWED BY ARBITRARILY MANY OF THE FOLLOWING PACKAGES,  ELECT
C     EACH OF WHICH IS TERMINATED BY AN EOF (7/8/9) CARD --           ELECT
C                                                                     ELECT
C        1)  $CONTROL ... $                                           ELECT
C        2)  $ PARAM ... $                                            ELECT
C        3)  PACKAGE OF SPECIES CARDS, CONTAINING: NAME, FRACTION PO,  ELECT
C            ENERGY (EV), MOLECULAR WEIGHT (GM/MOLE).  (A10,3E10.3)   ELECT
C                                                                     ELECT
C     THE FRACTIONAL GAS COMPOSITION IS DEFINED BY THE VALUES ENTERED  ELECT
C     FOR PO ON THE SPECIES CARDS.  IF THE TOTAL OF THESE VALUES IS MORE  ELECT
C     THAN 1.0E 08, IT IS ASSUMED THAT THE PO VALUES REPRESENT POPULA-  ELECT
C     TION DENSITIES (CM-3), AND ANY VALUES ENTERED FOR PTOT OR ATM ARE  ELECT
C     IGNORED.  IF THE TOTAL IS LESS THAN 1.0E 08, IT IS ASSUMED THAT  ELECT
C     THE PO VALUES REPRESENT PARTIAL PRESSURES (IN TORR), UNLESS EITHER  ELECT
C     PTOT OR ATM IS SPECIFIED ON THE $PARAM ...$ CARD, IN WHICH CASE  ELECT
C     THEY REPRESENT ONLY FRACTIONAL CONCENTRATIONS.  IF BOTH PTOT (IN  ELECT
C     TORR) AND ATM (IN ATMOSPHERES) IS SPECIFIED, THE VALUE FOR PTOT  ELECT
C     IS ACCEPTED, AND ATM IS IGNORED.                                ELECT
C                                                                     ELECT
C ------------------------------------------------------------------- ELECT
C                                                                     ELECT
      CALL DATE (TODAY)                                               ELECT
C                                                                     ELECT
C     DIMENSION DECLARATORS --                                        ELECT
      MGRID = 1000                                                    ELECT
      NMAX = 5                                                        ELECT
      NKMAX = 30                                                      ELECT
      LIMIT = 50                                                      ELECT
      MAX = 100                                                       ELECT
```

```
C
C     SCRATCH FILE FOR E- UPDATE --
      NTAPE = 7
C
C     INPUT E- CROSS SECTION TAPE FILE --
      INPUT = 8
      REWIND INPUT
C
C     FILE OF UPDATED E- CROSS SECTIONS --
      NDATA = 9
C
C     INPUT CARD FILE --
      KARDS = 10
C
      KB = KB/EE
      LIST = .TRUE.
      CALL EDITOR (KARDS, LIST)
      CALL UPDATE (INPUT, NDATA, NTAPE, .FALSE., TODAY)
      CALL COVER (TITLE,2)
C
C     -------------------------------------------
C     ENTER AN INITIAL REACTION SCHEME, IF DESIRED --
C     -------------------------------------------
C
C     UNLESS SEARCH = TRUE IS SPECIFIED ON $CONTROL...$ 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
      CALL EDITOR (KARDS,LIST)
      DO 28 N = 1,LIMIT
      DO 28 I = 1,6
   28 KINETIC(I,N) = 1H
      N = 0
      LIBRARY = .TRUE.
   53 READ (5,250) KODE
      IF (EOF(5)) 50,33
   33 IF (N.EQ.LIMIT) GO TO 53
      N = N+1
      LIBRARY = .FALSE.
      DO 51 I = 1,6
   51 KINETIC(I,N) = KODE(I)
      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
   50 CALL EDITOR (KARDS,LIST)
C
C     ENTER CALCULATION CONTROL PARAMETERS --
C
      MESH = 1000
      EMAX = 5.0
```

```
      EXPAND = .FALSE.
      PCT = 10.
      PERCENT = PCT + 1.
      TMAX = 50.
      ITMAX = 100
      EPS = .001
      IDEG = 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 EOVERN(N) = -1.0
      TE = 0.
      SEARCH = FATAL = .FALSE.
C
C     -----------------
      READ (5,CONTROL)
C     -----------------
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.EQ.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     ---------------
```

```
C
      IF (EOF(5)) 99,73
   73 IF (UB.GT.EMAX) UB = EMAX
      UA = UA*1.000001
      UB = UB*1.000001
C
C     ENTER SPECIES PARAMETERS --
C
C     --------------------------------
   20 READ (5,110) TYPE, P0, E0, MOLWT
C     --------------------------------
C
      IF (EOF(5)) 11,6
C     REJECT MORE THAN MAX DIFFERENT SPECIES --
    6 IF (NTYPE.EQ.MAX) GF TO 20
C     PROGRAM USES THE LATEST VALUES READ FOR E(-) OR IONIZE --
      IF (TYPE.NE.4HE(-).AND.TYPE.NE.2HE-.AND.TYPE.NE.1HE) GO TO 34
         NE = P0
         GO TO 20
   34 IF (TYPE.NE.6HIONIZE) GO TO 39
         IONIZE = P0
         GO TO 20
C     REJECT DUPLICATION OF SPECIES --
   39 IF (NTYPE.EQ.0) GO TO 64
      DO 62 I = 1,NTYPE
      IF (TYPE.NE.GAS(I)) GO TO 62
C        PREVIOUS VALUES FOR GAS(I) ARE REPLACED BY MOST RECENT --
         E(I) = E0
         NO(I) = P0
         GMOLE(I) = MOLWT
         GO TO 20
   62 CONTINUE
C
C     STORE PARAMETERS FOR THE NEW SPECIES CALLED *TYPE* --
   64 NTYPE = NTYPE+1
      GAS(NTYPE) = TYPE
      E(NTYPE) = E0
      NO(NTYPE) = P0
      GMOLE(NTYPE) = MOLWT
      GO TO 20
C
C     PARAMETERS USED FOR MOMENTUM TRANSFER --
C
   11 PRESS = NMOL = NGAS = 0
      DO 69 I = 1,NTYPE
   69 PRESS = PRESS + NO(I)
C     SORT SPECIES IN DESCENDING ORDER OF COMPOSITION --
   68 PMAX = 0.
      DO 1 I = 1,NTYPE
         DO 65 K = 1,NGAS
         IF (I.EQ.IDENT(K)) GO TO 1
   65    CONTINUE
      IF (NO(I).LE.PMAX) GO TO 1
         PMAX = NO(I)
         NG = I
    1 CONTINUE
C     REJECT COMPONENTS < 0.01 % FROM MOMENTUM TRANSFER CALCULATIONS --
```

```
      TEST = (NGAS.NE.1).AND.(PMAX.LT.1.0E-04*FI(1))                    ELECT
      IF (TEST) GO TO 66                                                ELECT
      NGAS = NGAS + 1                                                   ELECT
      IDENT(NGAS) = NG                                                  ELECT
      NAME(NGAS) = GAS(NG)                                              ELECT
      MASS(NGAS) = GMOLE(NG)                                            ELECT
      FI(NGAS) = NO(NG)                                                 ELECT
      IF (NGAS.LT.NMAX) GO TO 68                                        ELECT
C                                                                       ELECT
   66 NMOL = PRESS                                                      ELECT
      DO 67 I = 1,NGAS                                                  ELECT
   67 FI(I) = FI(I)/PRESS                                               ELECT
      IF (NMOL.GT.1.0E 08) GO TO 54                                     ELECT
      IF (PTOT.EQ.0.) PTOT = PRESS                                      ELECT
      NMOL = 0.965E 19*PTOT/TMOL                                        ELECT
      FRACT = NMOL/PRESS                                                ELECT
      DO 9 I = 1,NTYPE                                                  ELECT
    9 NO(I) = NO(I)*FRACT                                               ELECT
   54 IF (NE.LE.0.) NE = IONIZE*NMOL                                    ELECT
      NP1 = NTYPE+1                                                     ELECT
      GAS(NP1) = 4HE(-)                                                 ELECT
      E(NP1) = 0.                                                       ELECT
      KOUNT = NP1                                                       ELECT
      DO 7 N = 1,NKMAX                                                  ELECT
    7 U(N) = N1(N) = N2(N) = 0.                                         ELECT
C                                                                       ELECT
      N = NK = 0                                                        ELECT
C                                                                       ELECT
C     ----------------------------------------------------------------- ELECT
C     SELECT, FROM THE INPUT REACTION QUEUE, ALL (LEGAL) REACTIONS WHICH ELECT
C     INVOLVE SPECIES WHICH HAVE BEEN ENTERED --                        ELECT
C     ----------------------------------------------------------------- ELECT
C                                                                       ELECT
   35 IF (SEARCH) GO TO 70                                              ELECT
      IF (N.EQ.LIMIT) GO TO 5                                           ELECT
      N = N+1                                                           ELECT
      DECODE (60,260,KINETIC(1,N)) IMAGE                                ELECT
      GO TO 61                                                          ELECT
C                                                                       ELECT
C     ----------------------------------------------------------------- ELECT
C     SEARCH THE EXTERNAL ELECTRON CROSS SECTION FILE AND RETAIN ALL    ELECT
C     RELEVANT REACTIONS (UP TO NKMAX) FOR THE INPUT GAS MIXTURE --     ELECT
C     ----------------------------------------------------------------- ELECT
C                                                                       ELECT
   70 READ (NDATA,260) (IMAGE(L), L = 1,60)                             ELECT
      IF (EOF(NDATA)) 5,63                                              ELECT
   63 N = NK+1                                                          ELECT
C                                                                       ELECT
C     DUMP NUMERICAL DATA --                                            ELECT
      READ (NDATA)                                                      ELECT
   59 READ (NDATA,250) KODE                                             ELECT
      IF (KODE(1).EQ.1H .AND.KODE(2).EQ.1H ) GO TO 61                   ELECT
      GO TO 59                                                          ELECT
C                                                                       ELECT
   61 IF (NK.EQ.NKMAX) GO TO 5                                          ELECT
      CALL DEKODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60)    ELECT
C     RECALL THAT KOUNT IS AUTOMATICALLY UPDATED UPON RETURN FROM DEKODE ELECT
```

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

```
C     IF THE BOLTZMANN ANALYSIS BELOW DID NOT CONVERGE, CONTROL RETURNS  ELECT
C     TO THIS POINT TO EXPAND THE ELECTRON ENERGY RANGE --               ELECT
C     -----------------------------------------------------------------  ELECT
C                                                                        ELECT
   40 IF (EMAX.GE.5.) GO TO 23                                           ELECT
      EMAX = EMAX + EMAX                                                 ELECT
      IF (EMAX.GT.5.) EMAX = 5.                                          ELECT
      GO TO 27                                                           ELECT
   23 EMAX = EMAX + 5.0                                                  ELECT
   27 IF (EMAX.GT.25.) GO TO 50                                          ELECT
C                                                                        ELECT
   22 E0 = 0.                                                            ELECT
      DE = EMAX/MESH                                                     ELECT
      NA = UA/DE+1                                                       ELECT
      NB = UB/DE+1                                                       ELECT
      IF ((UA.NE.UB).AND.(NA.EQ.NB)) NB = NA+1                           ELECT
      DU = (NB-NA)*DE                                                    ELECT
      SI = 0.                                                            ELECT
      IF (DU.GT.0.) SI = 1./DU                                           ELECT
      DO 29 I = 1,M                                                      ELECT
      QMOM(I,1) = QMOM(I,2) = 0.                                         ELECT
      EV(I) = E0                                                         ELECT
      E0 = E0 + DE                                                       ELECT
      S(I) = 0.                                                          ELECT
      IF (I.LT.NA.OR.I.GT.NB) GO TO 29                                   ELECT
      S(I) = SI                                                          ELECT
   29 QM(I) = EV(I)*S(I)                                                 ELECT
      BEAM = BEAM*DU*SI                                                  ELECT
C                                                                        ELECT
C     PLOT SOURCE FUNCTION S(U) FOR EXTERNAL DEPOSITION --               ELECT
      IF (BEAM.EQ.0.) GO TO 45                                           ELECT
      CALL SIMPSON (QM, MESH/2, DE, DEPOSIT)                             ELECT
            DEPOSIT = 1.602E-19*BEAM*DEPOSIT                             ELECT
      WRITE (6,360) BEAM, DEPOSIT                                        ELECT
      Y0(1) = DY(1) = 0.                                                 ELECT
      CALL PLOT (1, M, 1, S, Y0, DY, EV, 0., 0., .TRUE., .TRUE., .TRUE., ELECT
     1 .TRUE., .TRUE., TITLE, 1, 0)                                      ELECT
C                                                                        ELECT
      WRITE (6,370) TODAY                                                ELECT
C                                                                        ELECT
C     -------------------------------------------                        ELECT
C     PROCESS THE ELECTRON CROSS SECTION ARRAYS --                       ELECT
C     -------------------------------------------                        ELECT
C                                                                        ELECT
   45 K = 0                                                              ELECT
      J = 1                                                              ELECT
      ERROR = .FALSE.                                                    ELECT
      DO 57 N = 1,NK                                                     ELECT
      I = NUMBER(N)                                                      ELECT
      DECODE (60,260,KINETIC(1,I)) IMAGE                                 ELECT
      CALL DEKODE (GAS, IMAGE, LHS, RHS, LABEL, IOUT, 10, KOUNT, 60)     ELECT
C                                                                        ELECT
C     PROCESS THE NTH INELASTIC ELECTRON CROSS SECTION --                ELECT
C                                                                        ELECT
      CALL PLASMA (NDATA, MGRID+1, MESH, LHS, RHS, PROCESS, EV, F, G,    ELECT
     1 Q(1,N), UTH, UM, KOUNT, GAS, MISSING, REJECT, OUTSIDE, IDEG,      ELECT
     2 OUT(6))                                                           ELECT
```

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

```
      IF (MASS(I).LE.0.) J = J+5
      IF (J.LE.225) GO TO 17
      ENCODE (100,330,LINE(J))
      J = J+10
      GO TO 17
C
C     ----------------------------------------------------------------
C     CONSTRUCT TWO MOMENTUM TRANSFER FUNCTIONS WHICH OCCUR IN THE
C     BOLTZMANN EQUATION --
C     ----------------------------------------------------------------
C
   36 FRACT = FI(I)
      DO 44 L = 1,M
      FQ = FRACT*QM(L)
      QMOM(L,1) = QMOM(L,1) + FQ
   44 QMOM(L,2) = QMOM(L,2) + FQ/MASS(I)
C
   17 ERROR = ERROR.OR.STOP
C
C     -------------------------------------------------------
C     GENERATE SUMMARY OF WARNING DIAGNOSTICS, IF ANY --
C     -------------------------------------------------------
C
      J = J-1
      IF (J.GT.0) WRITE (6,112) (LINE(L), L = 1,J)
C
      DO 46 L = 1,M
      X = EV(L)
      XSQ = TWOM*X*X
      STOP = QMOM(L,1).EQ.0.
      IF (STOP) GO TO 8
      A(L,1) = X/NMOL/QMOM(L,1)
   46 A(L,2) = XSQ*NMOL*QMOM(L,2)
C
      XBAR = DE/2.
      DO 41 I = 1,MESH
      CALL INTERP (2, XBAR, QMOM(I,1), EV, A(1,1), 1, M)
      CALL INTERP (2, XBAR, QMOM(I,2), EV, A(1,2), 1, M)
   41 XBAR = XBAR + DE
      QMOM(M,1) = QMOM(MESH,1)
      QMOM(M,2) = QMOM(MESH,2)
C
    8 ERROR = ERROR.OR.STOP
      FATAL = FATAL.AND.ERROR
         IF (STOP) WRITE (6,350)
         IF (FATAL) GO TO 50
      EXPON = EXP(-DE/KTE)
      FB = 1.
      DO 37 I = 1,M
      F(I) = FB
   37 FB = FB*EXPON
C
      ENCODE (40,500,KAPT)
C
C     PROHIBIT FURTHER PLOTS OR TABULATIONS OF E- CROSS SECTION DATA--
C
      OUT(6) = OUT(7) = .FALSE.
```

```
      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
      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, FI, NGAS, NMOL, TMOL, ITMAX,
     1 TMAX, EPS, KAPT, TODAY, OUT, EVCM, NE, PROCESS, U, N1, N2, NEL,
     2 S, BEAM, CREATE, EV, Q, QMOM, 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.EQ.0.) PWR = AMAX1(ABS(ELASTIC),ABS(PCOLL),ABS(DEDT))
      P(NN,NKP1) = ELASTIC*100./PWR
      DO 4 J = 1,NK
```

```
      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     ------------------------------------------------------------
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
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.NKP1) GO TO 2
      M2 = M1+20
      IF (NKP1.LT.M2) M2 = NKP1
      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,YO) NM2, NDASH
      ENCODE (50,300,KAPT) NSPACE, L21
      WRITE (6,DY)
      WRITE (6,270)
      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,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     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 =
     1   1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,2), Y0, DY, EOVERN, 0., 0.,
     1 .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 =
     1   1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,4), Y0, DY, EOVERN, 0., 0.,
     1 .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 =
     1   1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,6), Y0, DY, EOVERN, 0., 0.,
     1 .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 =
     1   1,NGAS), TMOL
      CALL PLOT (21, NPTS, 1, TABLE(1,8), Y0, DY, EOVERN, 0., 0.,
     1 .TRUE., .TRUE., .FALSE., .TRUE., .TRUE., HEAD(8), 2, 0)
      WRITE (6,200) TODAY
C
C     PLOTS OF ENDOTHERMIC 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
```

```
      IF (NPLOT.EQ.0) GO TO 16                                          ELECT
      IF (NEL(J).EQ.0) NPLOT = NPLOT+1                                  ELECT
      ENCODE (20,160,KAPT) J, J                                         ELECT
      WRITE (6,240) J, (PROCESS(I,J), I=1,4)                            ELECT
      IF (NGAS.EQ.1) WRITE (6,220) NAME(1), TMOL                        ELECT
      IF (NGAS.GT.1) WRITE (6,FORM) (NAME(I), I = 1,NGAS), (FI(I), I =  ELECT
     1   1,NGAS), TMOL                                                  ELECT
      CALL PLOT (21, NPTS, 1, RATE(1,1,J), 0., 0., EOVERN, 0., 0.,      ELECT
     1   .TRUE., .TRUE., .TRUE., .TRUE., .TRUE., KAPT, NPLOT, 0)        ELECT
      WRITE (6,200) TODAY                                               ELECT
   16 CONTINUE                                                          ELECT
C                                                                       ELECT
      GO TO 50                                                          ELECT
C                                                                       ELECT
C  -----------------------  FORMAT STATEMENTS  -------------------- ELECT
C                                                                       ELECT
  100 FORMAT (5H(1H1,,I1,4H(/)))                                        ELECT
C                                                                       ELECT
  110 FORMAT (A10,3E10.3)                                               ELECT
C                                                                       ELECT
  111 FORMAT (I2*)  *4A10,5X)                                           ELECT
C                                                                       ELECT
  112 FORMAT (1H1/20X*WARNING --*//25X*ERRORS OCCURRED FOR THE INPUT E- ELECT
     1CROSS SECTIONS FOR THE FOLLOWING ELECTRON COLLISION */25X*PROCESSE ELECT
     2S.  IF THE PROGRAM IS EXECUTED, A ZERO CROSS SECTION WILL BE ASSUM ELECT
     3ED:*//20X,90(1H-)//(40X,5A10))                                    ELECT
C                                                                       ELECT
  113 FORMAT (*MOMENTUM TRANSFER FOR *,A10)                             ELECT
C                                                                       ELECT
  114 FORMAT (I4,I6)                                                    ELECT
C                                                                       ELECT
  115 FORMAT (10(1H )*NO E- CROSS SECTION DATA WAS FOUND.*)             ELECT
C                                                                       ELECT
  116 FORMAT (10(1H )*SIGMA = 0 IN THE RANGE [0,*F4.1*] EV.*)           ELECT
C                                                                       ELECT
  117 FORMAT (10(1H )*ERRORS OCCURRED IN CROSS SECTION DATA.*)          ELECT
C                                                                       ELECT
  118 FORMAT (10(1H )*CROSS SECTION DOES NOT SPAN [0,*F4.1*] EV.*)      ELECT
C                                                                       ELECT
  119 FORMAT (10(1H )*THE MASS ENTERED FOR THIS SPECIES IS ≤ 0*)        ELECT
C                                                                       ELECT
  120 FORMAT (42X*SUMMARY OF PLASMA PARAMETERS AS A FUNCTION OF E/NTOT*) ELECT
C                                                                       ELECT
  130 FORMAT (/14X,*E/NTOT*,7X,*UBAR*,3X,*EK = D/MU*,5X,*TE*,10X,*VD*,  ELECT
     1 12X,*MU*,13X,*D*9X*P/NMOL/NE*3X,11HJ/NE = E*VD/11X,*(E-16 V CM2)* ELECT
     24X*(EV)*,5X,*(EV)*,6X,*(DEG K)*,5X,*(CM/S)*,7X,*(CM2/V/S)*,6X,    ELECT
     3*(CM2/S)*,4X,*(WCM3/ELMOL)*,1X,*(AMP CM/ELECT)*/10X,116(1H-))    ELECT
C                                                                       ELECT
  140 FORMAT (/10X,F10.3,F11.3,F10.3,F11.0,1P5E14.3)                    ELECT
C                                                                       ELECT
  150 FORMAT (/10X,116(1H-)//36X*(THE ELECTRON ENERGY RANGE [0,*F5.1*] E ELECT
     1V WAS DIVIDED INTO *,I4/36X,    *INTERVALS, GIVING AN ENERGY RESOLU ELECT
     2TION OF DU =*,1PE9.2,* EV.)*)                                     ELECT
C                                                                       ELECT
  160 FORMAT (6HVSIGF(,I2,2H) ,6HVSIGR(,I2,2H) )                        ELECT
C                                                                       ELECT
  170 FORMAT (1H1/44X*AVERAGE AND CHARACTERISTIC ELECTRON ENERGY (EV)*)  ELECT
```

```
C
  180 FORMAT (1H142X*EFFECTIVE ELECTRON TEMPERATURE AND DRIFT VELOCITY*)
C
  190 FORMAT (1H1/46X*ELECTRON MOBILITY AND DIFFUSION COEFFICIENT*)
C
  200 FORMAT (/56X*E/NTOT (1.0 E-16 VOLT CM2)*18X*DR. WILLIAM B. LACINA,
     1*A11/100X*NORTHROP RESEARCH AND TECHNOLOGY*)
C
  210 FORMAT (1H142X*TOTAL ELECTRICAL POWER AND CURRENT DENSITY PER NE*)
C
  220 FORMAT (55X,*PURE *,A3,*  TMOL =*,F5.0,* DEG K*/)
C
  230 FORMAT (1H(,I2,4HXA3,,I2,20H(*/*A3),* = *2PF6.2,,I2,38H(* /*F6.2),
     1*, TMOL =*,0PF5.0* DEG K*/))
C
  240 FORMAT (1H1,33X,*LOG PLOT OF ELECTRON COLLISION RATES <VSIG> (CM3/
     1SEC) FOR REACTION (*I2*)*/55X,4A10/)
C
  250 FORMAT (8A10)
C
  260 FORMAT (80A1)
C
  270 FORMAT (18X*SUMMARY OF FRACTIONAL % POWER PARTITION FOR ALL ELECTR
     1ON COLLISION PROCESSES AS A FUNCTION OF E/NTOT*)
C
  280 FORMAT (*(/*I2≠X,4X*REACTION*15X*E/NTOT -->    *F7.2,≠I1*F15.2/*,
     1I2,≠X,24X*(1.0E-16 V CM2)*)≠)
C
  290 FORMAT (1H(,I2≠X,≠I3*(1H-)/)*)
C
  300 FORMAT (1H(,I2*X,4A10,F8.3,*I1*F15.3/)*)
C
  310 FORMAT (5H(1H1,,I2*(/))*)

  320 FORMAT (*MOMENTUM TRANSFER COLLISIONS*)
C
  330 FORMAT (50X*NO FURTHER WARNING DIAGNOSTICS WILL BE ISSUED.*)
C
  340 FORMAT (50X)
C
  350 FORMAT (/30X*FATAL ERROR -- MOMENTUM TRANSFER COLLISION FREQUENCY*
     1/30X*BECAME ZERO AT SOME POINT; ANALYSIS HAS 1/OM TERMS.*)
C
  360 FORMAT (1H1,22X,   *NORMALIZED EXTERNAL SOURCE FUNCTION S(U) FOR C
     1REATION 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*/)
C
  370 FORMAT (/62X*ELECTRON ENERGY U (EV)*16X*DR. WILLIAM B. LACINA,*A11
     1/100X*NORTHROP RESEARCH AND TECHNOLOGY*)
C
  500 FORMAT (7X*ELECTRON KINETICS ANALYSIS*7X)
C
C -----------------------------------------------------------------
C
   99 CALL EXIT
      END
```