

JAERI-M

6 4 0 1

A METHOD OF SOLVING THE RATE EQUATIONS

February 1976

Tsuneo AMANO\* and Masao OKAMOTO

日本原子力研究所  
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合せは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

JAERI - M 6401

A Method of Solving the Rate Equations

Tsuneo AMANO\* and Masao OKAMOTO

Division of Thermonuclear Fusion Research, Tokai, JAERI

(Received January 20, 1976)

A numerical method is presented of solving the rate equations of impurities. The rate equation is treated as an eigenvalue problem. Examples of the calculation are given for C, O, Fe and Mo impurities

---

\* Plasma Physics Laboratory, Faculty of Engineering, Osaka University,  
Suita, Osaka

JAERI - M 6401

A Method of Solving the Rate Equations

Tsuneo AMANO\* and Masao OKAMOTO

Division of Thermonuclear Fusion Research, Tokai, JAERI

(Received January 20, 1976)

A numerical method is presented of solving the rate equations of impurities. The rate equation is treated as an eigenvalue problem. Examples of the calculation are given for C, O, Fe and Mo impurities

---

\* Plasma Physics Laboratory, Faculty of Engineering, Osaka University,  
Suita, Osaka

rate equationの解法について

日本原子力研究所東海研究所核融合研究部

天野恒雄・岡本正雄\*

(1976年1月20日受理)

不純物の電離状態を表わす rate equation の数値解法について述べる。rate equation は固有値問題として扱った。炭素、酸素、鉄、モリブデンの不純物について数値計算例を示す。

---

\* 大阪大学工学部超高温理工学研究施設

# 目 次 な し

A numerical method is presented for the rate equation which is frequently met in the problem of impurities. Although the rate equations are solved in some articles<sup>1~3)</sup>, in the present paper, we present a rather simple method and show numerical results for carbon, oxygen, iron and molybdenum impurities, since the present method is quite useful when we incorporate the effect of heavy impurities in the one-dimensional Tokamak transport code

The ground-state population densities  $n_k$  ( $k$  denoting the ionization stage) of impurities in dilute plasmas (say, Tokamak plasmas) are determined by rate equations of the form:

$$\frac{dn_k}{dt} = n_e(\alpha_{k-1}n_{k-1} - \alpha_k n_k) - n_e(\beta_{k-1}n_k - \beta_k n_{k+1}), \quad (1)$$

$(k=1, 2, \dots, K)$

with

$$\alpha_0 = \beta_0 = \alpha_K = \beta_K = 0.$$

Here,  $n_e$  is the electron density,  $\alpha$  the collisional ionization rate coefficient,  $\beta$  the radiative recombination coefficient. The three-body recombination is not included, since we are considering a dilute plasma in a Tokamak. For example, for oxygen nine such rate equations exist ( $K=9$ )

The system of Eqs. (1) for  $k = 1, 2, \dots, K$  can be solved numerically by the Runge-Kutta method. However, if we adopt the Runge-Kutta method, we must choose a time mesh smaller than the minimum ionization time. The ionization time ( $\sim 1/n_e \alpha$ ) depends strongly on the electron temperature  $T_e$  and varies on a wide range. For example, for  $n_e = 10^{13} \text{ cm}^{-3}$  and  $T_e = 10 \text{ eV} \sim 1 \text{ keV}$ , the minimum ionization time of oxygen, which corresponds to the ionization from the stage  $k = 1$  to  $k = 2$  for  $T_e = 100 \text{ eV}$ , is about  $5 \times 10^{-7} \text{ sec}$ . It is clear that the Runge-Kutta method requires too much time. Then, we solve the eigenvalue equation of Eq. (1) instead of Runge-Kutta method to avoid a too much small time mesh.

The eigenvalue equation of Eq. (1) is written by

$$\lambda \vec{\xi} = A \vec{\xi} \quad (2)$$

Here the matrix  $A$  is of the form:

$$A = \begin{bmatrix} -\alpha_1 & \beta_1 & & & \\ \alpha_1 & -\alpha_2 - \beta_1 & \beta_2 & & 0 \\ & \alpha_2 & -\alpha_3 - \beta_2 & \beta_3 & \\ & & \ddots & & \\ & & \alpha_{k-1} & -\alpha_k + \beta_{k-1} & \beta_k \\ 0 & & & & \alpha_{k-1} - \beta_{k-1} \end{bmatrix} \quad (3)$$

which is a non-symmetric tridiagonal matrix.

It should be noted that A is the Jacobi matrix since  $\alpha_k \beta_k > 0$ . By Sturm's theorem, it is proved that all the eigenvalues of the Jacobi matrix are real and non-degenerate<sup>4)</sup>. Then, we get the eigenvalues by the bi-section method<sup>5)</sup>. We write eigen vectors  $\vec{\xi}_m$  in the form:

$$\vec{\xi}_m = e^{\lambda m t} \vec{x}_m. \quad (4)$$

We seek  $\vec{x}_m$  by the inverse iteration method<sup>6)</sup>. If eigenvalues and eigen-vectors of the system were found the solution could be written as

$$y_n(t) = \sum_{m=1}^K c_m x_m^{(n)} e^{\lambda m t}, \quad n=1, 2, \dots, K \quad (5)$$

where  $x_m^{(n)}$ 's are elements of the vector  $\vec{x}_m$ . The initial condition gives the coefficients  $c_m$ 's if we solve the system of linear equations.

In the present numerical calculation, we employ the approximate estimation given by Hinov<sup>7)</sup> for rate coefficients of ionization and recombination  $\alpha$ ,  $\beta$ . That is,

$$\left. \begin{aligned} \alpha_k &= 5.9 \times 10^{-8} q_k (E_k^{k+1})^{-3/2} \sqrt{x_k} K_1(x_k), & [\text{cm}^3/\text{sec}] \\ \beta_k &= 5.2 \times 10^{-14} k \cdot x_k^{3/2} \cdot K_1(x_k) e^{x_k}, & [\text{cm}^3/\text{sec}] \\ x_k &= E_k^{k+1} \cdot x_H/T_e, & (T_e : \text{in eV}) \\ K_1(z) &= \int_z^\infty \frac{e^{-t}}{t} dt, \end{aligned} \right\} \quad (6)$$

where  $q_k$  is the number of valence electron in the outermost shell,  $E_k^{k+1}$  is the ionization potential in rydberg unit to ionize the stage  $k$  to  $k+1$  and  $\chi_H = 13.59$  eV. Only ground states are considered and the dielectronic recombination<sup>8)</sup> has not been included in Eq. (6). Ionization potentials in Eq. (6) are referred to Lotz<sup>9)</sup> for carbon, oxygen and iron impurities. The data for heavier atoms, especially in multiple ionization states, are less adequately known, and in many cases may be only roughly estimated. Here, we use the estimation of ionization potential given by Hinnov<sup>7)</sup> for molybdenum impurities. Ionization potentials  $E_k^{k+1} \cdot \chi_H$  and the number of valence electrons  $q_k$  are listed in Table 1.

Numerical examples for carbon, oxygen, iron and molybdenum impurities are shown in Figs. 1, 2, 3 and 4, respectively. The abscissas in the figures are normalized time which is the real time multiplied by the electron density,  $n_e t$ . Impurity densities  $n_k$  are also normalized by the initial density of neutrals. At  $t=0$ , only neutrals (say, CI) have densities and the solutions of rate equations show time developments of impurities in various states for specified electron temperature.

The list of program used in the present calculation is appended to the paper. In the program code, the bi-section method was developed to the case when the tridiagonal matrix is nonsymmetric.

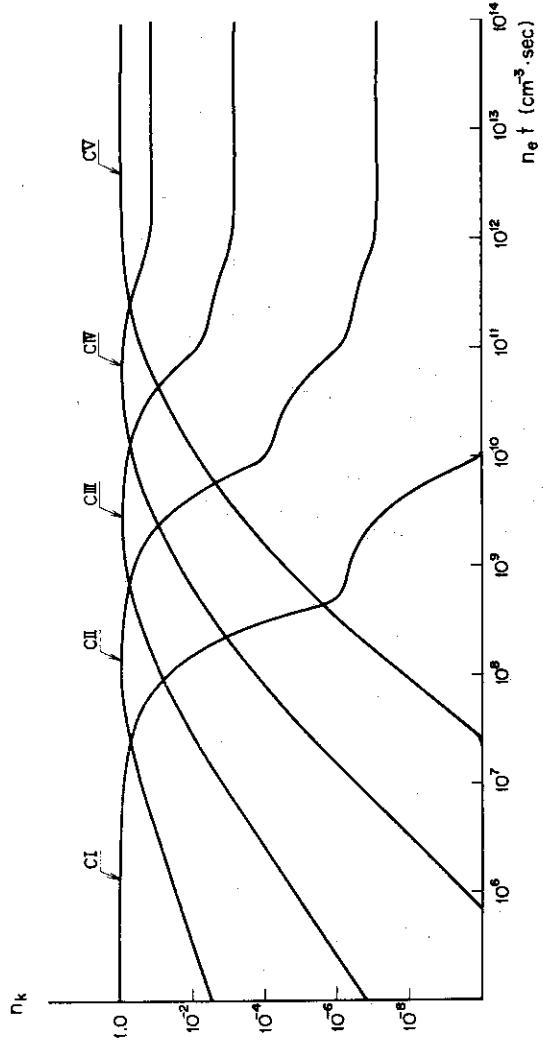
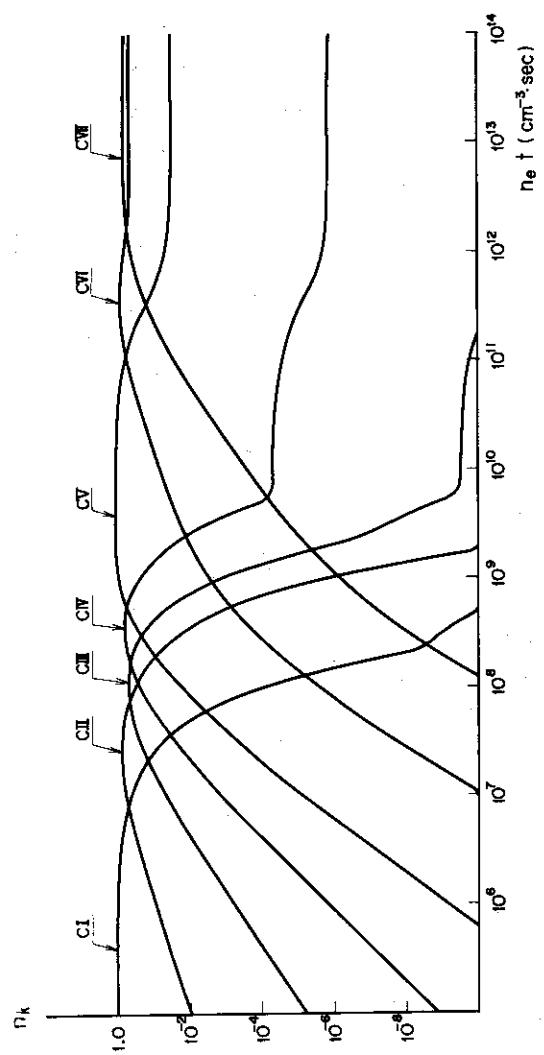
In conclusion, we thank the members of theoretical group of the division of thermonuclear fusion research in JAERI for their fruitful discussions.

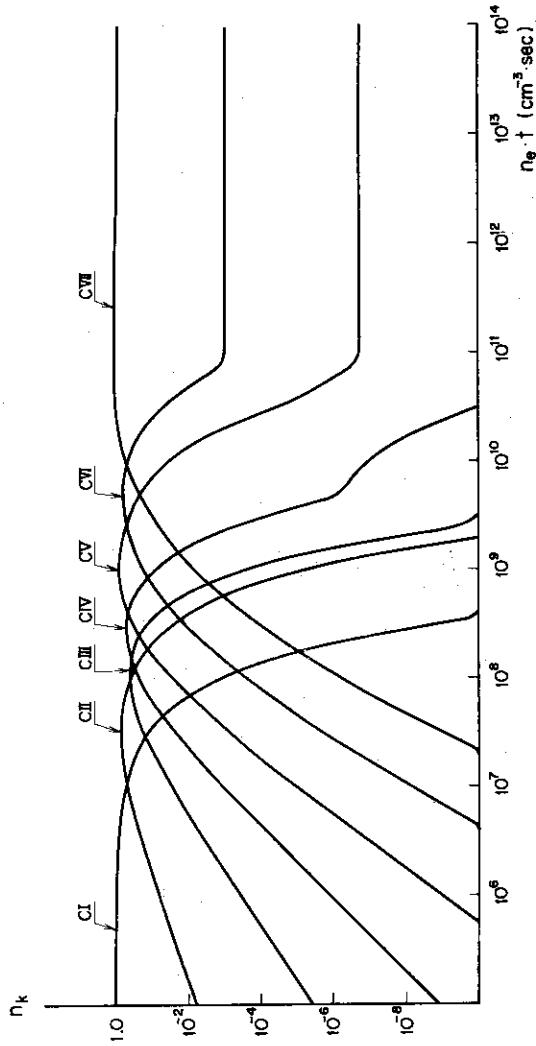
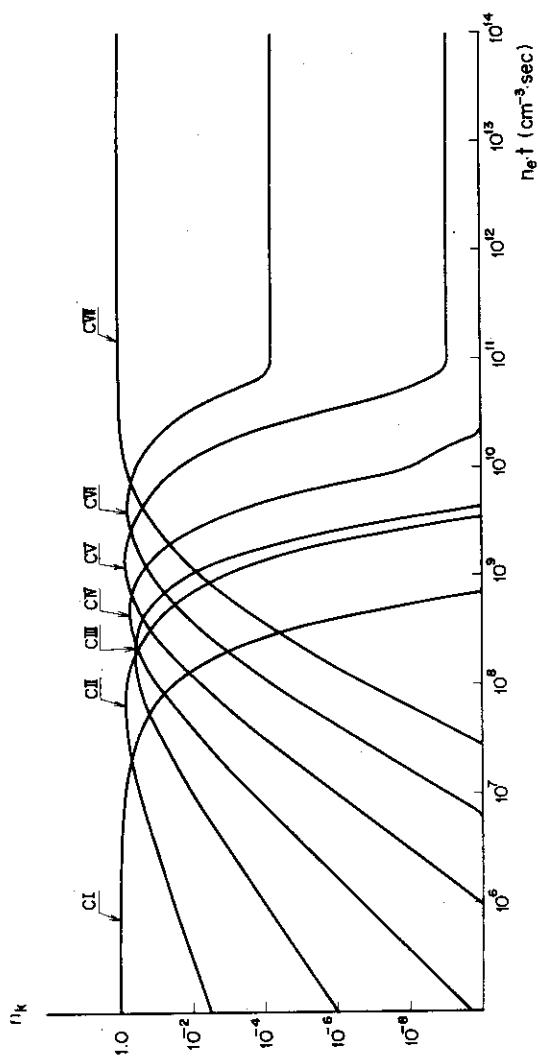
#### References

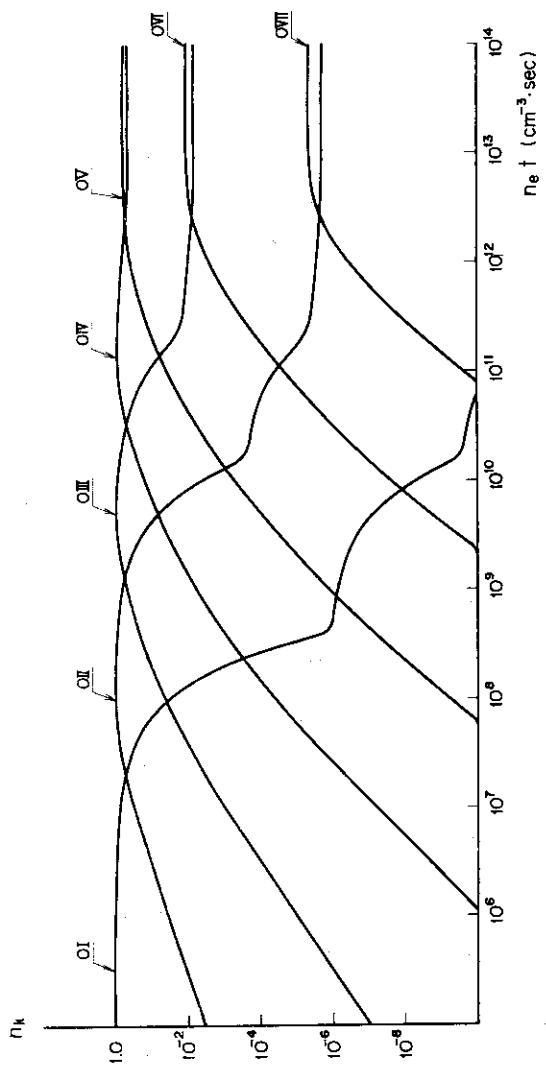
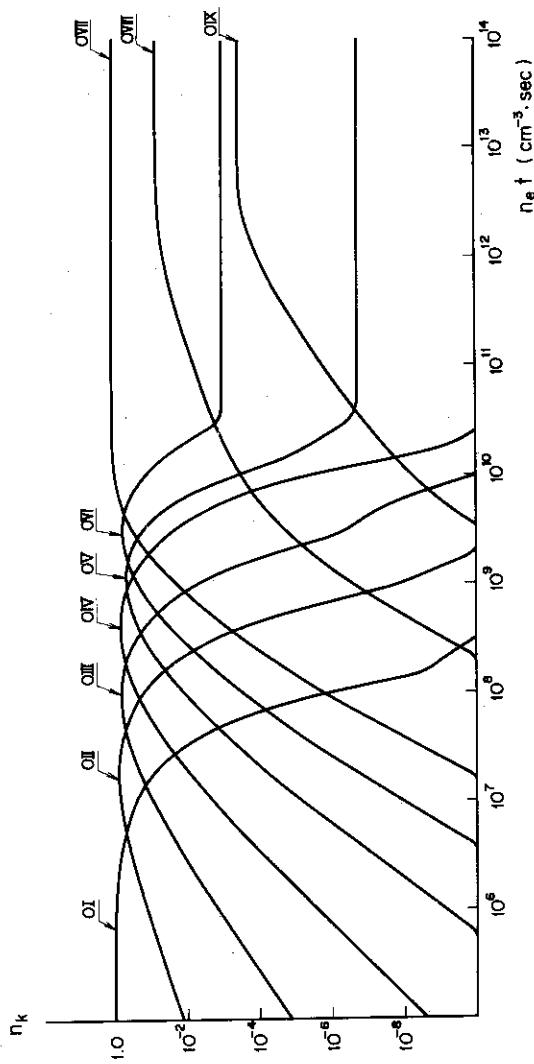
- 1) A. C. Kolb and R. W. P. McWhirter: Phys. of Fluids 7 (1964) 519.
- 2) D. Düchs and H. R. Griem: Phys. of Fluids 9 (1966) 1099.
- 3) D. Düchs, W. Engelhardt and W. Köppendörfer: Nucl. Fusion 14 (1974) 73.
- 4) D. K. Faddeeb and B. H. Faddeeba: Numerical Method of Linear Algebra, translated by S. Furuya and T. Oguni, Sangyo-Tosho, 1970 (in Japanese).
- 5) A. Ralston and H. S. Wilf: Mathematical Methods for Digital Computers, Wiley, New York and London (1960).
- 6) J. H. Wilkinson: The Algebraic Eigenvalue Problem, Clarendon Press, Oxford (1965).
- 7) E. Hinnov: Princeton Plasma Physics Laboratory Report, Matt-777 (1970).
- 8) A. Burgess: Astrophys. J. 139 (1964) 776 and ibid 141 (1965) 1588.
- 9) W. Lotz: J. Opt. Soc. Am. 57 (1967) 873.

Table 1 Ionization Potentials and Equivalent Electrons

	CARBON		OXYGEN		IRON		MOLIBDENUM	
	$E_k^{k+1}$ (eV)	$q_k$						
I	11.260	2	13.618	4	7.870	2	17.184	1
II	24.382	1	35.117	3	16.182	1	43.314	5
III	47.883	2	54.900	2	30.651	6	74.365	4
IV	64.492	1	77.413	1	54.8	5	109.141	3
V	392.084	2	113.902	2	75.0	4	146.962	2
VI	489.980	1	138.115	1	99.0	3	187.339	1
VII			739.327	2	125.0	2	230.163	6
VIII			871.390	1	151.18	1	275.027	5
IX					234.87	6	321.794	4
X					262.0	5	370.328	3
XI					290.3	4	420.493	2
XII					330.8	3	472.154	1
XIII					361.0	2	525.311	2
XIV					392.2	1	579.963	1
XV					457.0	2	635.838	10
XVI					490.0	1	692.937	9
XVII					1265.0	6	751.260	8
XVIII					1358.0	5	810.806	7
XIX					1456.0	4	871.430	6
XX					1582.0	3	933.025	5
XXI					1689.0	2	995.834	4
XXII					1799.0	1	1059.458	3
XXIII					1950.0	2	1124.171	2
XXIV					2045.0	1	1189.834	1
XXV					8828.0	2	1256.450	6
XXVI					9278.0	1	1323.881	5
XXVII							1392.128	4
XXVIII							1461.327	3
XXIX							1531.341	2
XXX							1602.171	1
XXXI							1673.816	2
XXXII							1746.142	1
XXXIII							3928.955	6
XXXIV							4163.469	5
XXXV							4404.780	4
XXXVI							4652.889	3
XXXVII							4907.795	2
XXXVIII							5169.499	1
XXXIX							5438.000	2
XXXX							5713.299	1
XXXXI							21752.000	2
XXXXII							22853.195	1

Fig. 1-(a) Carbon  $T_e = 10$  eVFig. 1-(b) Carbon  $T_e = 100$  eV

Fig. 1-(c) Carbon  $T_e = 1$  keVFig. 1-(d) Carbon  $T_e = 10$  keV

Fig. 2 - (a) Oxygen  $T_e = 10 \text{ eV}$ Fig. 2 - (b) Oxygen  $T_e = 100 \text{ eV}$

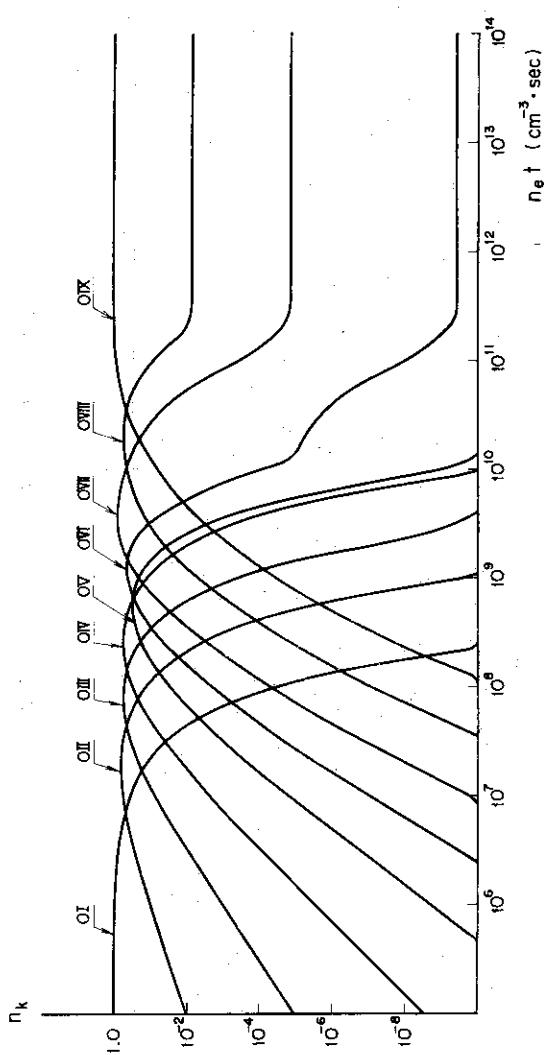


Fig. 2-(c) Oxygen  $T_e = 1$  keV

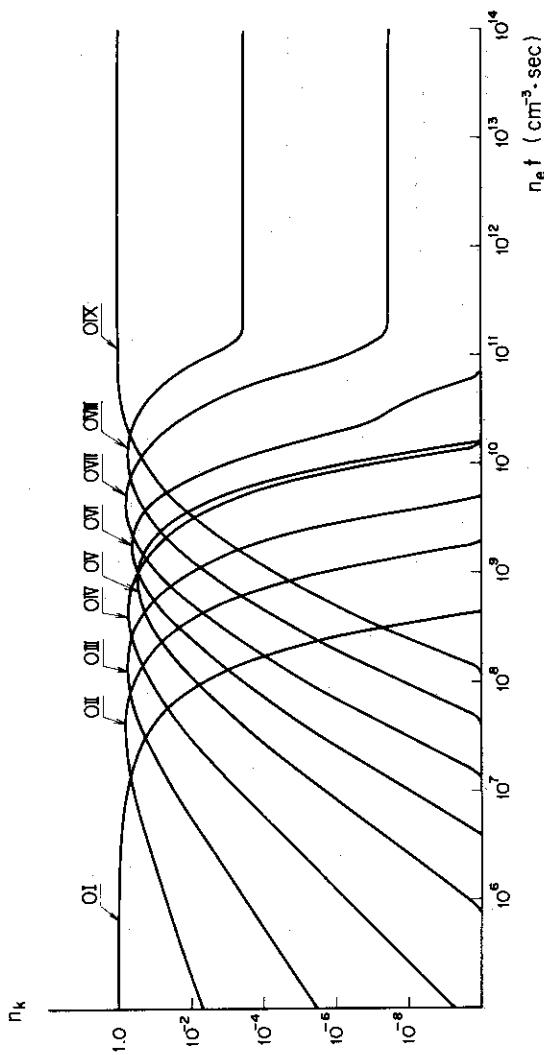
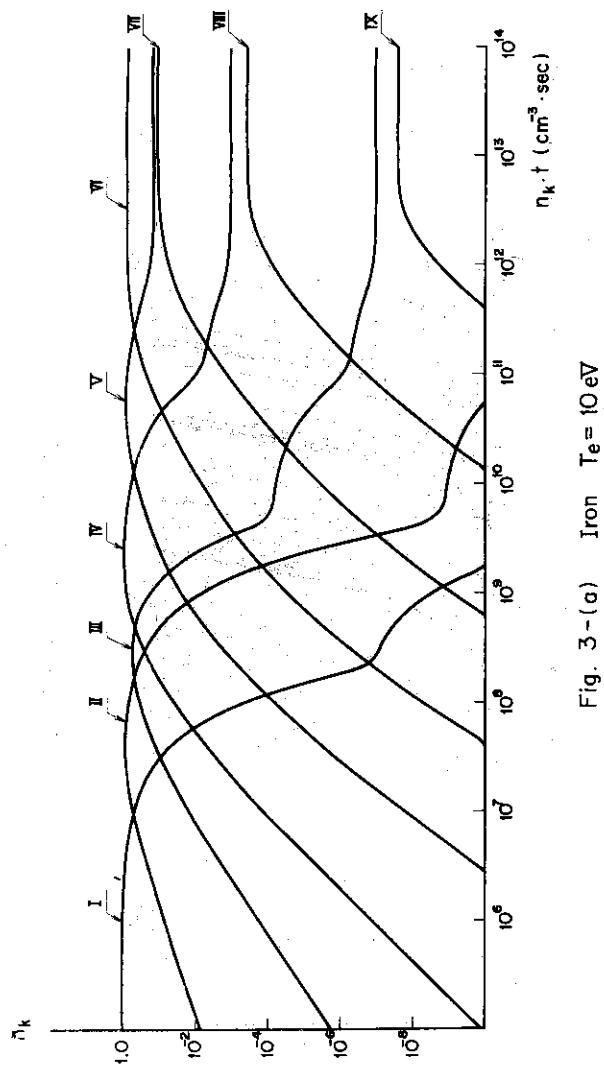
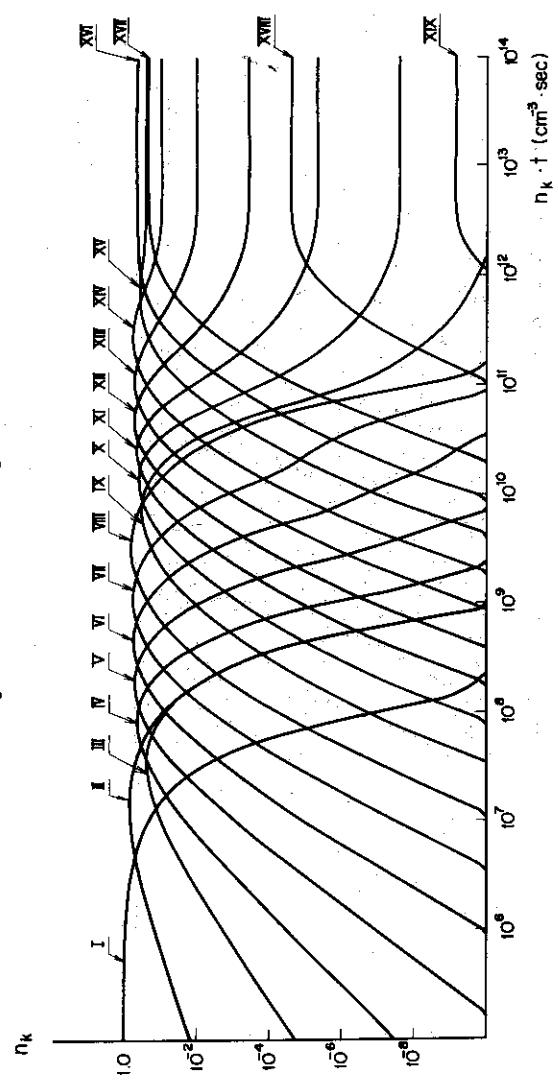
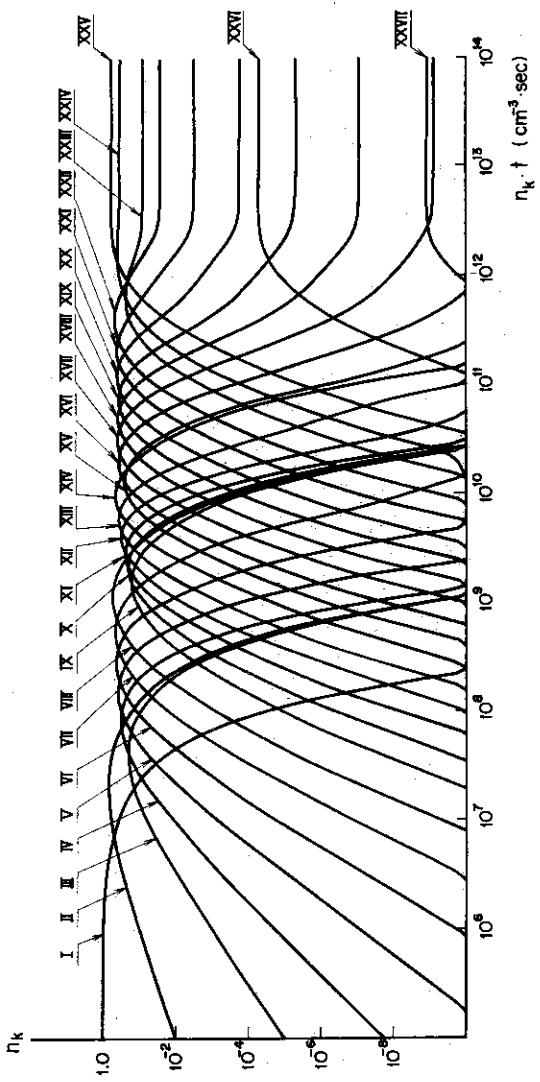
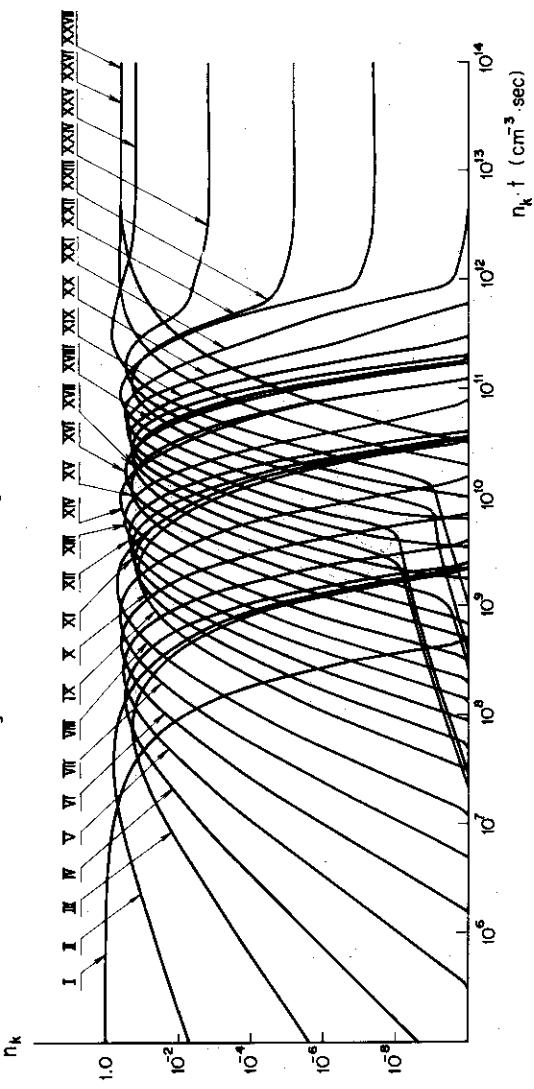
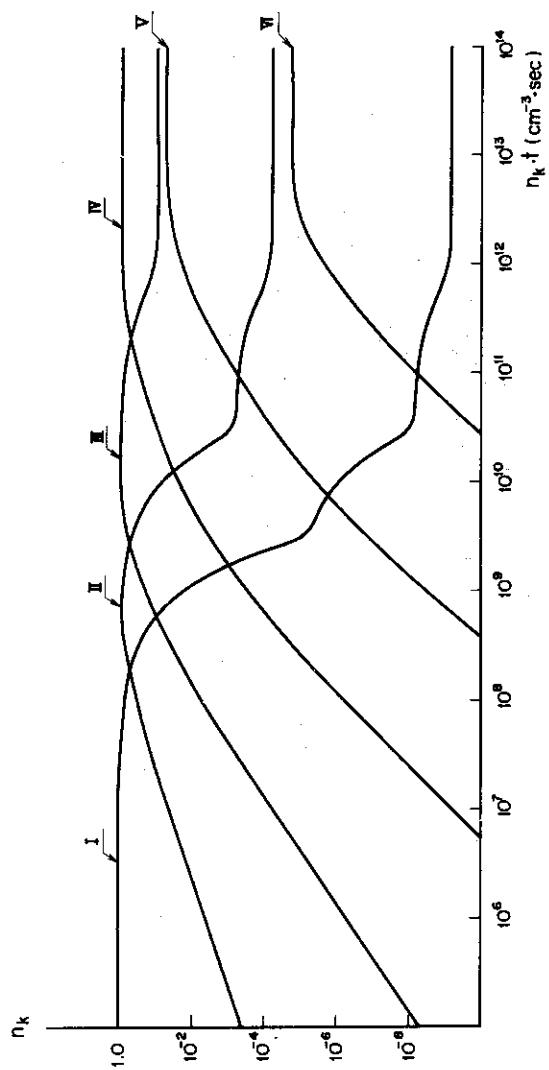
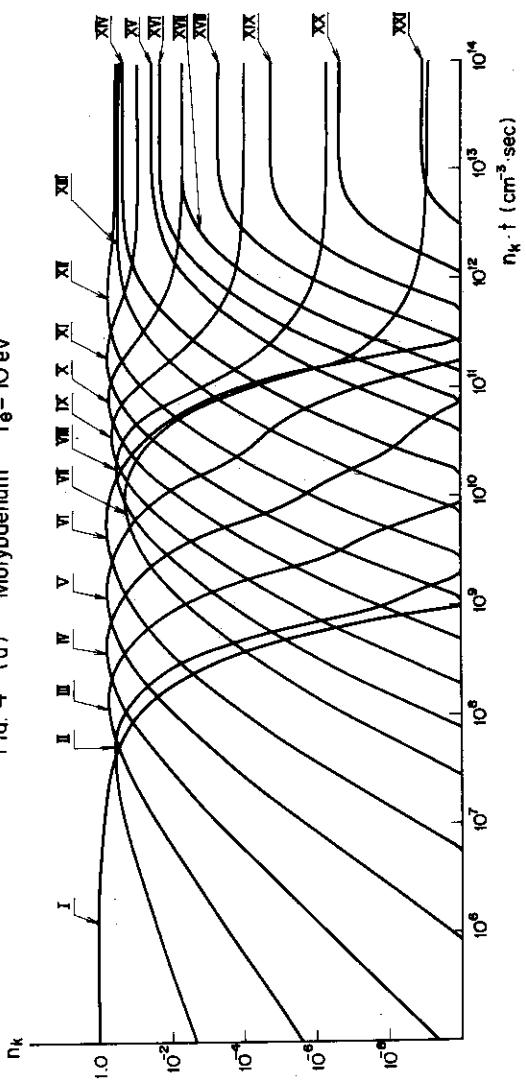
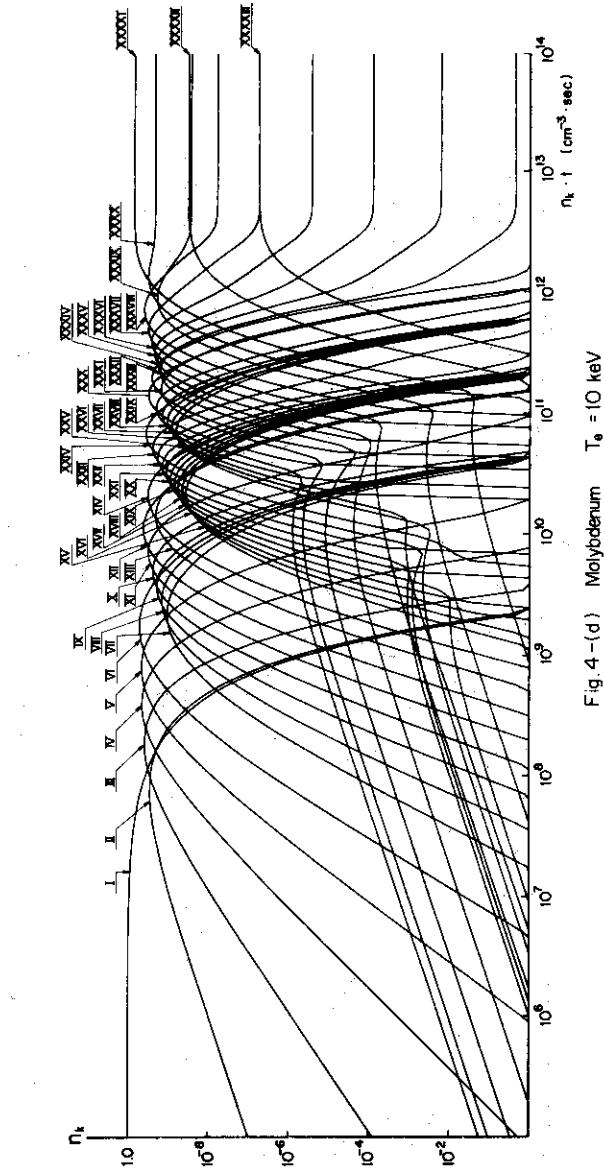
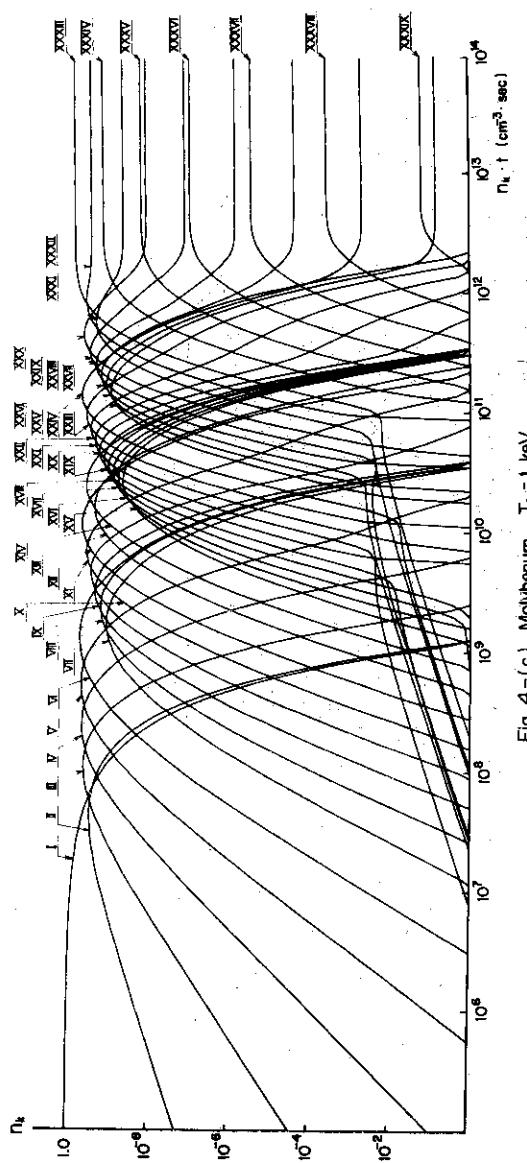


Fig. 2-(d) Oxygen  $T_e = 10$  keV

Fig. 3-(a) Iron  $T_e = 10$  eVFig. 3-(b) Iron  $T_e = 100$  eV

Fig. 3-(c) Iron  $T_e = 1 \text{ keV}$ Fig. 3-(d) Iron  $T_e = 10 \text{ keV}$

Fig. 4-(a) Molybdenum  $T_e = 10$  eVFig. 4-(b) Molybdenum  $T_e = 100$  eV



FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 11

\* SOURCE STATEMENT \*

```

C ***** MAI00010
C ***** MAI00020
C ***** MAI00030
C ***** MAI00040
C ***** MAI00050
C ***** MAI00060
C ***** MAI00070
C ***** MAI00080
C ***** MAI00090
C ***** MAI00100
C ***** MAI00110
C ***** MAI00120
C ***** MAI00130
C ***** MAI00140
C ***** MAI00150
C ***** MAI00160
C ***** MAI00170
C ***** MAI00180
C ***** MAI00190
C ***** MAI00200
C ***** MAI00210
C ***** MAI00220
C ***** MAI00230
C ***** MAI00240
C ***** MAI00250
C ***** MAI00260
C ***** MAI00270
C ***** MAI00280
C ***** MAI00290
C ***** MAI00300
C ***** MAI00310
C ***** MAI00320
C ***** MAI00330
C ***** MAI00340
C ***** MAI00350
C ***** MAI00360
C ***** MAI00370
C ***** MAI00380
C ***** MAI00390
C ***** MAI00400
C ***** MAI00410
C ***** MAI00420
C ***** MAI00430
C ***** MAI00440
C ***** MAI00450
C ***** MAI00460
C ***** MAI00470
C ***** MAI00480
C ***** MAI00490
C ***** MAI00500
C ***** MAI00510
C ***** MAI00520
C ***** MAI00530
C ***** MAI00540
C ***** MAI00550
C ***** MAI00560
C ***** MAI00570
C ***** MAI00580
C ***** MAI00590
C
1   LOGICAL CARBON,OXYGEN,IRON,MOLIB
2   LOGICAL PLOTT
3   DOUBLE PRECISION A,B,YL,    OX,SOA
4   DOUBLE PRECISION E
5   COMMON/COTH/CARBON,OXYGEN,IRON,MOLIB
6   COMMON/EIV/E(50),YL(50,50),OX(50),SOA(50)
7   COMMON/CMF/A(50,50),B(50,50)
8   COMMON/OPA/ALPHA(50)+BETA(50)
9   COMMON/REPLT/TM(200)+YDX(50+200)
10  COMMON/PLT/NPLOT
11  NAMELIST/RATE/CARBON,OXYGEN,IRON,KMAX,KMIN,NPLOT,PLOTT
C
C ** AN = ELECTRON DENSITY. TE = ELECTRON TEMPERATURE
C
12  AN=1.0E13,
13  CALL PLTSRT
14  PLOTT=.FALSE.
15  KMAX=9
16  KMIN=1
17  NPLOT=100
18  B00 CONTINUE
19  CARBON=.FALSE.,
20  OXYGEN=.FALSE.,
21  IRON=.FALSE.,
22  MOLIB=.FALSE.
C
C ** INPUT DATA (NAMELIST IS USED)
C   CARBON --- LOGICAL   IF CARBON=.TRUE., RATE EQUATION FOR CARBON IS SOLVED.
C   OXYGEN --- LOGICAL   IF OXYGEN=.TRUE., RATE EQUATION FOR OXYGEN IS SOLVED.
C   IRON ---- LOGICAL    IF IRON=.TRUE., RATE EQUATION FOR IRON IS SOLVED.
C   MOLIB ---- LOGICAL   IF MOLIB=.TRUE., RATE EQUATION FOR MOLIB DENOM IS SOLVED.
C   KMAX ----- ELECTRIC CHARGE NUMBER + 1 (CARBON=.TRUE., KMAX=7) MAI00540
C   KMIN ----- MINIMUM IONIZATION STAGE WHICH IS CONSIDERED. MAI00550
C   (IF ALL IONIZATION STAGES ( FROM NEUTRALS TO FULLY IONIZED IONS) ARE CONSIDERED, KMIN=1.) MAI00560
C   NPLOT ---- NO. OF PLOTTER OUTPUT MAI00570
C   PLOTT --- LOGICAL   IF PLOTT=.TRUE., THE PLOTTING ROUTINES MAI00580
C

```

FACOM 230-75 (M7) FORTRAN-D -75U715- V06-L05 75.11.28 PAGE 12

\* SOURCE STATEMENT (RTMAIN) \*

```

C          WORK.
C** IONIZATION AND RECOMBINATION RATE COEFFICIENTS FOR CARBON
C
23      READ(5,RATE,ERR=4000,END=5000)                                MAI00600
24      IF(CARBON) GO TO 100                                         MAI00610
25      IF(OXYGEN)  GO TO 200                                         MAI00650
26      IF(IRON)    GO TO 300                                         MAI00660
27      IF(MOLIB)   GO TO 450                                         MAI00670
28      GO TO 4000                                         MAI00680
29      100 IF(OXYGEN) GO TO 4000                                     MAI00690
30      IF(IRON)    GO TO 4000                                     MAI00700
31      IF(MOLIB)   GO TO 4000                                     MAI00710
32      WRITE(6,501)                                              MAI00720
33      501 FORMAT(1H1,//////////,10X,'C A R B O N!',/////////)     MAI00730
34      GO TO 4000                                              MAI00740
35      200 IF(IRON) GO TO 4000                                     MAI00750
36      IF(MOLIB)   GO TO 4000                                     MAI00760
37      WRITE(6,502)                                              MAI00770
38      502 FORMAT(1H1,//////////,10X,'D X Y G E N!',/////////)     MAI00780
39      GO TO 4000                                              MAI00790
40      300 IF(MOLIB) GO TO 4000                                     MAI00800
41      WRITE(6,503)                                              MAI00810
42      503 FORMAT(1H1,//////////,10X,'I R O N!',/////////)        MAI00820
43      GO TO 4000                                              MAI00830
44      450 WRITE(6,504)                                         MAI00840
45      504 FORMAT(1H1,//////////,10X,'M O L I B !',/////////)     MAI00850
46      400 CONTINUE                                              MAI00860
47      WRITE(6,699) NPLOT                                         MAI00870
48      WRITE(6,700)                                              MAI00880
49      699 FORMAT(1H ,'* * PLUTTER NO **',I10)                   MAI00890
50      700 FORMAT(1H ,'* * INPUT DATA **')                         MAI00900
51      WRITE(6,RATE)                                            MAI00910
52      KMIN=KMIN+1                                             MAI00920
53      KMAX=KMAX-1                                             MAI00930
54      KM=KMAX-KMIN+1                                         MAI00940
55      N=KMAX                                                 MAI00950
56      M1=1                                                   MAI00960
57      M2=N                                                   MAI00970
58      AN=1.0E13                                              MAI00980
59      DTE=1.0                                                 MAI00990
60      DO 3000 MT1=1,4                                         MAI01000
61      DTE=0.0                                                 MAI01010
62      DO 3000 MT2=1,9                                         MAI01020
63      DTE=DTE+DDTE                                           MAI01030
64      TE=DTE*10.0**MT1                                         MAI01040
65      IF(TE.GE.30000) GO TO 3000                           MAI01050
66      WRITE(6,99)                                            MAI01060
67      99 FORMAT(1H1)                                         MAI01070
68      WRITE(6,98) TE                                         MAI01080
69      98 FORMAT(1H ,'* TE =',1PE12.3,' EV')                  MAI01090
C
C++ CALCULATE RATE COEFFICIENTS
C      ALFA = IONIZATION,  BETA = RECOMBINATION
C
70      DO 3 K=1,KMAX                                         MAI01100
71      3 DX(K)=0.00                                         MAI01120
72      DX(1)=1.0E10                                         MAI01130
73      IF(CARBON) CALL CRATE(TE)                            MAI01140
74      IF(OXYGEN) CALL ORATE(TE)                            MAI01150
                                         MAI01160
                                         MAI01170
                                         MAI01180

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 13

\* SOURCE STATEMENT (FTMAIN )\*

```

75      TECIRON) CALL FERATE(E)
76      TECMOLIR) CALL MORATE(E)
C
C** FORM A TRIDIAGONAL MATRIX
C
77      DO 10 I=1,KMAX
78      DO 10 J=1,KMAX
79      10 A(I,J)=0,
80      DO 1  I=2,KMAX1
81      A(I,1)=(ALFA(I)+BETA(I-1))
82      A(I,I-1)=ALFA(I-1)
83      A(I,I+1)=BETA(I)
84      1 CONTINUE
85      A(1,1)=-ALFA(1)
86      A(1,2)= BETA(1)
87      A(KMAX,KMAX)=ALFA(KMAX1)
88      A(KMAX,KMAX)=BETA(KMAX1)
89      DO 2  I=1,KMAX
90      DO 2  J=1,KMAX
91      2 A(I,J)=A(I,J)*AN
C
C** EIGTRI SOLVES THE ABOVE TRIDIAGONAL MATRIX.
C  FUC11 IS ENTRY IN EIGTRI
C
92      CALL EIGTRI(KM,IE)
93      DT=0.0
94      CALL FUC11(DT,M1,M2)
95      DO 5 I=1,KMAX
96      5 IF(DABS(OX(I)),LT,1.0) OX(I)=0.0
97      WRITE(6,430) DT,(OX(I),I=1,N)
98      DT=0.05
99      TT=-9.05
100     J1MAX=181
101     DO 305 J1=1,J1MAX
102     TT=TT+DT
103     S1=10.0**TT
104     CALL FUC11(DT,M1,M2)
105     DO 4 I=1,KMAX
106     4 IF(DAHS(OX(I)),LT,1.0) OX(I)=0.0
107     TM(J1)=DT
108     DO 51 K=1,KMAX
109     51 YOX(K,J1)=SNGL(OX(K))
110     IF(PLOTT) GO TO 1110
111     WRITE(6,430) DT,(OX(I),I=1,N)
112     1110 CONTINUE
113     430 FORMAT(1H ,1PE10.3,1P9E12.3,4(/1H ,10X,1P9E12.3))
114     305 CONTINUE
115     CALL PLTREC(E,KMAX,J1MAX)
116     3000 CONTINUE
117     GO TO 800
118     4000 *RITE(6,4001)
119     4001 FORMAT(//,.1H ,'* *** DATA NOIREKATA NI AYAMARI GA ARU ***')
120     5000 CALL PLTEND
121     STOP
122     END

```

MA101190  
MA101200  
MA101210  
MA101220  
MA101230  
MA101240  
MA101250  
MA101260  
MA101270  
MA101280  
MA101290  
MA101300  
MA101310  
MA101320  
MA101330  
MA101340  
MA101350  
MA101360  
MA101370  
MA101380  
MA101390  
MA101400  
MA101410  
MA101420  
MA101430  
MA101440  
MA101450  
MA101460  
MA101470  
MA101480  
MA101490  
MA101500  
MA101510  
MA101520  
MA101530  
MA101540  
MA101550  
MA101560  
MA101570  
MA101580  
MA101590  
MA101600  
MA101610  
MA101620  
MA101630  
MA101640  
MA101650  
MA101660  
MA101670  
MA101680  
MA101690  
MA101700  
MA101710  
MA101720  
MA101730

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 1

\* SOURCE STATEMENT \*

```

1      SUBROUTINE CRATE(TE)
C
C      TE = ELECTRON TEMPERATURE (EV)
C      Y = IONIZATION (CM**3/SEC), Z = RECOMBINATION (CM**3/SEC)
C      FROM E.HINNUV, MATT-77(1970).
C
C
2      COMMON/YRA/Y(50),Z(50)
3      DIMENSION OJ(6),E1(6),E2(6)
4      DATA KMAX,XMAX/7.6/
5      DATA OJ/2.0,1.0,2.0,1.0,2.0,1.0/
6      DATA E1/ 0.829, 1.793, 3.521, 4.743, 28.833, 36.031/
7      DATA E2/ 11.264, 24.376, 47.864, 64.476, 391.906, 489.840/
8      DO 6000 K=1,KMAX1
9      X=E2(K)/TE
10     IF(X>4.0) 10,10,20
11     10 ARG=4.0/X
12     RES=7*((((0.0009442714*ARG-0.0049362007)*ARG+0.011723273)
1     *ARG-0.017555779)*ARG+0.020412099)*ARG-0.022951979)
2     *ARG+0.031205561)*ARG-0.062498588)*ARG+0.24999999)*ARG
13     X2=X
14     IF(X2.GT.72.0) X2=72.0
15     Y(K)=5.9*E8*0.5*(1.0/(E1(K)*SQR(E1(K)))*SQR(X)*RES*EXP(-X2)
16     Z(K)=5.2E-14*FLOAT(K)**2*SQR(X)*RES
17     GO TO 6000
18     10 RES=-ALOG(ABS(X))-((((((0.10317602E-11*X-0.15798675E-10)*X
1     +0.16826592E-9)*X+0.21415699E-8)*X+0.27635830E-7)*X
2     -0.30726221E-6)*X+0.30996040E-5)*X+0.28337590E-4)*X
3     +0.23148392E-3)*X+0.001666906 )*X+0.010416662 )*X
4     -0.055555520)*X+0.25)*X-1.0)*X-0.57721566
19     Y(K)=5.9E-8*0.5*(1.0/(E1(K)*SQR(E1(K)))*SQR(X)*RES
20     Z(K)=5.2E-14*FLOAT(K)*X*SQR(X)*RES*EXP(X)
21     6000 CONTINUE
22     RETURN
23     END

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 15

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE ORATE(TE)                               ORT00010
C
C** IONIZATION AND RECOMBINATION RATE COEFFICIENTS FOR OXYGEN   ORT00020
C   TE = ELECTRON TEMPERATURE (EV)                           ORT00030
C   Y = IONIZATION (CM**3/SEC), Z = RECOMBINATION (CM**3/SEC)   ORT00040
C   FROM E.HINNOV, MATT-777(1970).                         ORT00050
C
C   COMMON/YR/A/Y(50),Z(50)                                ORT00060
C   DIMENSION Y(J),E1(J),E2(J)                            ORT00070
C   DATA KMAX,KMAA1/9,0/                                ORT00080
C   DATA WJ/4,0,3,0,2,0,1,0,2,0,1,0,2,0,1,0/             ORT00090
C   DATA E1/ 1.001, 2.585, 4.041, 5.693, 8.376,10.157,54.367,64.075/ ORT00100
C   DATA E2/13.614,35.146,54.934,77.394,113.573,138.08,739.114,871.1/ ORT00110
C   DO 6000 K=1,KMAX1                                ORT00120
C   X=E2(K)/TE                                         ORT00130
C   IF(X<-4.0) 10,10,20                                ORT00140
C   10 ARG=4.0/X
C   RES=(((((0.00094427614*ARG-0.0049362007)*ARG+0.011723273) ORT00150
C   1 *ARG-0.017555779)*ARG+0.020412099)*ARG-0.022951979) ORT00160
C   2 *ARG+0.031208561)*ARG-0.062498588)*ARG+0.24999999)*ARG ORT00170
C   13 X2=X                                         ORT00180
C   14 IF(X2.GT.72.0) X2=72.0                         ORT00190
C   15 Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQRT(E1(K))))*SQRT(X)*RES*EXP(-X2) ORT00200
C   16 Z(K)=5.2E-14*FLOAT(K)*X*SQRT(X)*RES             ORT00210
C   17 GO TO 6000                                     ORT00220
C   18 10 RES=-ALOG(ABS(X))-((((((0.10317602E-11*X-0.15798675E-10)*X ORT00230
C   1 +0.16826592E-9)*X-0.21915699E-8)*X+0.27635830E-7)*X ORT00240
C   2 -0.30726221E-6)*X+0.30996040E-5)*X-0.28337590E-4)**X ORT00250
C   3 +0.23148392E-3)*X-0.0016666906 )*X+0.010416662 )**X ORT00260
C   4 -0.055555520)*X+0.25)*X-1.0)*X-0.57721566             ORT00270
C   19 Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQRT(E1(K))))*SQRT(X)*RES ORT00280
C   20 Z(K)=5.2E-14*FLOAT(K)*X*SQRT(X)*RES*EXP(X)           ORT00290
C   21 6000 CONTINUE                                    ORT00300
C   22 RETURN                                         ORT00310
C   23 END                                            ORT00320

```

FACOM 120-75 (M7) FORTRAN-D 75U715+ V06-L05 75.11.28 PAGE 9

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE PERATE(E)
2
3      C** IONIZATION AND RECOMBINATION RATE COEFFICIENTS FOR IRON
4      C   TE = ELECTRON TEMPERATURE (EV)
5      C   Y = IONIZATION (CM**3/SEC), Z = RECOMBINATION (CM**3/SEC)
6      C   FROM F.HINNOV, MATT-77/1970.
7
8      COMMON/ORA/Y(50),Z(50)
9      DIMENSION WJ(26),E1(26),E2(26)
10     DATA KMAX,KMAX1/27,26/
11     DATA WJ/2.0,1.0,6.0,5.0,4.0,3.0,2.0,1.0,6.0,5.0,4.0,3.0,2.0,1.0,
12          *1.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,1.0,7.0,1.0/
13     DATA E1/0.579,1.190,2.255,4.031,5.317,7.282,9.195,11.120,17.276,
14          19.272,21.353,24.332,26.554,28.849,33.615,36.043,39.049,
15          99.890,107.098,116.366,124.237,132.328,143.435,150.423,
16          649.356,682.457/
17     DATA E2/7.870,16.189,30.651,54.800,75.000,99.000,125.000,151.180,
18          177.1234,234.871,262.00,290.30,330.80,361.00,392.20,457.00,490.00,
19          1265.0,1358.0,1456.0,1582.0,1689.0,1799.0,1950.0,2045.0,
20          8828.0,9276.0/
21     DO 6000 K=1,KMAX1
22     X=E2(K)/TE
23     1FX=4.0*10.10*20
24     20 ARG=4.0/X
25     RES=((((0.00094427614*ARG-0.0049362007)*ARG+0.011723273),
26          *ARG-0.017555779)*ARG+0.020412099)*ARG-0.022951979)
27          +ARG+0.031208561)*ARG-0.062498588)*ARG+0.249999999)*ARG
28     X2=X
29     1TF(X2,GT,72.0), X2=72.0
30     Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQR(E1(K))))*SQR(X)*RES*EXP(-X2)
31     Z(K)=5.2E-14*FLOAT(K)*X*SQR(X)*RES
32     GO TO 6000
33     10 RES=-ALOG(ABS(X))-(((((0.10317602E-11*X-0.15798675E-10)*X
34          +0.16626592E-9)*X-0.21915699E-8)*X+0.27635830E-7)*X
35          -0.30726221E-6)*X+0.30996040E-5)*X-0.28337590E-4)*X
36          +0.123148392E-32*X-0.601666690)*X+0.10416662)*X
37          -0.055555520)*X+0.25)*X-1.0)*X-0.57721566
38     Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQR(E1(K))))*SQR(X)*RES
39     Z(K)=5.2E-14*FLOAT(K)*X*SQR(X)*RES*EXP(X)
40     6000 CONTINUE
41     RETURN
42     END

```

FRT0010  
FRT0020  
FRT0030  
FRT0040  
FRT0050  
FRT0060  
FRT0070  
FRT0080  
FRT0090  
FRT0100  
FRT0110  
FRT0120  
FRT0130  
FRT0140  
FRT0150  
FRT0160  
FRT0170  
FRT0180  
FRT0190  
FRT0200  
FRT0210  
FRT0220  
FRT0230  
FRT0240  
FRT0250  
FRT0260  
FRT0270  
FRT0280  
FRT0290  
FRT0300  
FRT0310  
FRT0320  
FRT0330  
FRT0340  
FRT0350  
FRT0360  
FRT0370  
FRT0380  
FRT0390  
FRT0400  
FRT0410  
FRT0420

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 14

\* SOURCE STATEMENT \*

```

1      SUBROUTINE MORTATE(TE)                                MRT00010
C
C** IONIZATION AND RECOMBINATION RATE COEFFICIENTS FOR MOLTBENUM   MRT00020
C   TE = ELECTRON TEMPERATURE (EV)                                MRT00030
C   Y = IONIZATION (CM**3/SEC), Z = RECOMBINATION (CM**3/SEC)        MRT00040
C   FROM EMINOV, MATT-777(1970).                                     MRT00050
C
C
2      COMMON/YR/A/Y(50),Z(50)                                MRT00060
3      DIMENSION Y(42),E1(42),E2(42)                          MRT00070
4      DATA KMAX1,KMAX2/43,42/                                MRT00080
5      DATA B/1.0,5.0,4.0,3.0,2.0,1.0,6.0,5.0,4.0,3.0,2.0,1.0,0,2.0,1.0,   MRT00110
1          10.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0,0,5.0,4.0,3.0,   MRT00120
2          2.0,1.0,2.0,1.0,6.0,5.0,4.0,3.0,2.0,1.0,2.0,1.0,2.0,1.0,   MRT00130
6      DATA E1/1.264,3.185,5.470,8.028,10.81,13.78,16.93,20.23,23.67,   MRT00140
1          27.24,30.93,34.73,38.64,42.66,46.77,50.97,55.26,59.64,   MRT00150
2          64.10,68.63,73.25,77.93,82.69,87.52,92.42,97.38,102.40,   MRT00160
3          107.49,112.64,117.55,123.12,128.44,239.00,306.25,324.00,   MRT00170
4          342.25,361.00,380.25,400.00,420.25,1600.00,1681.00/   MRT00180
7      DATA E2/17.184,43.314,74.365,109.141,146.962,187.339,230.163,   MRT00190
1          279.027,321.794,370.328,420.493,472.154,525.311,579.963,   MRT00200
2          635.836,692.937,751.260,810.806,871.430,933.025,995.834,   MRT00210
3          1059.455,1124.171,1189.834,1256.450,1323.881,1392.128,   MRT00220
3          1461.327,1531.341,1602.171,1673.816,1746.142,1928.955,   MRT00230
4          4163.469,4404.780,4652.869,4907.795,5169.499,5438.000,   MRT00240
5          5713.299,21752.00,22853.195/   MRT00250
8      DO 6000 K=1,KMAX1                                MRT00260
9      X=E2(K)/TE                                MRT00270
10     IF(X>4.0) 10,10,20                                MRT00280
11     20 ARG=4.0/X                                MRT00290
12     RES=((((0.00094427614*ARG-0.0049362007)*ARG+0.011723273)   MRT00300
1       *ARG-0.017555779)*ARG+0.020412099)*ARG-0.022951979)   MRT00310
2       *ARG+0.031208561)*ARG-0.062498588)*ARG+0.24999999)*ARG   MRT00320
13     X2=X                                MRT00330
14     IF(X2.GT.72.0) X2=72.0                                MRT00340
15     Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQR(E1(K))))*SQR(X)*RES*EXP(-X2)   MRT00350
16     Z(K)=5.2E-14*FLOAT(K)*X*SQR(X)*RES                MRT00360
17     GO TO 6000                                MRT00370
18     10 RES=-ALOG(ABS(X))-((((0.10317602E-11*X-0.15798675E-10)*X   MRT00380
1       +0.16826592E-9)*X+0.21915699E-8)*X+0.27635830E-7)*X   MRT00390
2       +0.30726221E-6)*X+0.30996040E-5)*X-0.28337590E-4)*X   MRT00400
3       +0.23148392E-3)*X+0.001666690E-1)*X+0.010416662E-2)*X   MRT00410
4       -0.055555520)*X+0.25)*X-1.0)*X-0.57721566   MRT00420
19     Y(K)=5.9E-8*WJ(K)*(1.0/(E1(K)*SQR(E1(K))))*SQR(X)*RES   MRT00430
20     Z(K)=5.2E-14*FLOAT(K)*X*SQR(X)*RES*EXP(X)            MRT00440
21     6000 CONTINUE                                MRT00450
22     RETURN                                MRT00460
23     END                                MRT00470

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 6

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE EIGTRI(NN,IERO)
2      LOGICAL FIRST,IN
3      INTEGER AG
4      DOUBLE PRECISION PKT
5      DOUBLE PRECISION A,UL,V,OX,SOX
6      DOUBLE PRECISION E
7      DOUBLE PRECISION LAMBDA,NORM,L,T,U,S,MULT
8      DOUBLE PRECISION ERR,Y1(50),YMAX,S(50)
9      DOUBLE PRECISION P(50),Y(52),B(50),C(50),W(50),R(50)
10     DIMENSION IN(50)
11     COMMON/EIV/E(50),V(50,50),UX(50),SOX(50)
12     COMMON/CMF/A(50,50),UL(50,50)
13     NN=NN
14     IER=0
15     NORM=DABS(A(1,1))+DABS(A(1,2))
16     N1=NN-1
17     DO 2010 I=2,N1
18     T=DABS(A(I,1))+DABS(A(I,1+1))+DABS(A(I,I-1))
19     2010 NORM=DMAX1(NORM,T)
20     T=DABS(A(N,N))+DABS(A(N,N-1))
21     NORM=DMAX1(NORM,T)
22     DO 2011 I=2,N
23     2011 W(I)=A(I-1,1)*A(I,I-1)
24     K=1
25     U=1.0
26     DO 2012 I=1,N
27     2012 F(I)=NORM
28     ITER=0
29     2013 L=E(K)
30     2014 LAMBDA=0.5D0*(L+U)
31     IF(DABS(LAMBDA-L).LE.1.D-8,OR,DABS(LAMBDA-U).LE.1.D-8) GO TO 2030
32     AG=0
33     I=1
34     2016 S=A(I,1)-LAMBDA
35     2018 IF(S.GE.0.0) AG=AG+J
36     IF(S.EQ.0.0) GO TO 2020
37     I=I+1
38     IF(I.GT.N) GO TO 2020
39     S=A(I,1)-LAMBDA-W(I)/S
40     GO TO 2016
41     2020 I=I+2
42     IF(I.LE.N) GO TO 2016
43     2022 IF(AG.GE.K) GO TO 2024
44     U=LAMBDA
45     GO TO 2014
46     2024 L=LAMBDA
47     M=MINU(AG,N)
48     DO 2026 I=K,M
49     2026 F(I)=LAMBDA
50     IITER=ITER+1
51     IF(IITER.GT.2000) GO TO 2040
52     GO TO 2014
53     2030 E(K)=LAMBDA
54     K=K+1
55     IF(K.LE.N) GO TO 2013
56     2040 CONTINUE
57     DO 2042 I=1,N
58     DO 2044 J=1,N
59     P(IJ)=0.0

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 7

\* SOURCE STATEMENT (EIGTR1 )\*

```

60      R(J)=A(J,J+1)          E1G00600
61      R(JJ)=A(J,J)-E(1)       E1G00610
62      R(J)=A(J+1,J)          E1G00620
63 2044 Y(J)=1.00             E1G00630
64      Y(N)=1.00             E1G00640
65      R(N)=0.                E1G00650
66      R(N)=0.                E1G00660
67      R(N)=A(N,N)-E(1)       E1G00670
68      Y(N+1)=0.0             E1G00680
69      Y(N+2)=0.0             E1G00690
70      FIRST=.TRUE.           E1G00700
71      DO 2050 J=1,N1         E1G00710
72      IF(DABS(R(JJ)),LT,DABS(E(J))) GO TO 2046
73      MULT=B(J)/R(J)         E1G00720
74      INC(J)=.FALSE.         E1G00730
75      GO TO 2048             E1G00740
76 2046 MULT=R(J)/B(J)        E1G00750
77      INC(J)=.TRUE.          E1G00760
78      R(J)=B(J)              E1G00770
79      T=R(J+1)               E1G00780
80      R(J+1)=0(J)            E1G00790
81      R(J)=T                 E1G00800
82      P(J)=G(J+1)            E1G00810
83      G(J+1)=0.0              E1G00820
84 2048 W(J)=MULT             E1G00830
85      G(J+1)=G(J+1)-MULT*T*P(J) E1G00840
86      R(J+1)=R(J+1)-MULT*W(J) E1G00850
87      IF(R(J),EQ,0.0) R(J)=1.0E-30 E1G00860
88 2050 CONTINUE               E1G00870
89      IF(R(N),EQ,0.0) R(N)=1.0E-30 E1G00880
90 2054 CONTINUE               E1G00890
91      DO 2066 J=1,4             E1G00900
92      DO 2066 JJ=1,N             E1G00910
93      K=JJ-J+1                  E1G00920
94      T=Y(K)                   E1G00930
95 2062 Y(K)=(T-Y(K+1)*R(K)-Y(K+2)*P(K))/R(K) E1G00940
96 2066 CONTINUE               E1G00950
97      FRR=0.                    E1G00960
98      T=DARS(Y(1))             E1G00970
99      K=1                      E1G00980
100     DO 100 J=2,N             E1G00990
101     S=DARS(Y(J))             E1G01000
102     IF(S,LE,T) GO TO 100      E1G01010
103     T=S                      E1G01020
104     K=J                      E1G01030
105 100 CONTINUE               E1G01040
106      T=1.0G/Y(K)             E1G01050
107      DO 101 J=1,N             E1G01060
108      Y(J)=Y(J)*T             E1G01070
109      IF(T,LE,1.0) GO TO 201   E1G01080
110      DO 102 K=1,N             E1G01090
111      FRR=FRR+DARS(Y(K)-Y1(K)) E1G01100
112      FRR=FRR/N               E1G01110
113      IF(FRR,LE,1.E-5) GO TO 2074 E1G01120
114      IF(T,LE,4.0) GO TO 200    E1G01130
115 201 DO 223 K=1,N             E1G01140
116      Y(K)=Y(K)               E1G01150
117      DO 2076 J=1,N1            E1G01160
118      IF(T,J) GO TO 2066       E1G01170
                                         E1G01180

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 8

\* SOURCE STATEMENT (EIGINT )\*

```

119      Y(J+1)=Y(J+1)-e(J)*Y(J)          EIGO1190
120      GO TO 2070                      EIGO1200
121      2068 T=Y(J)                      EIGO1210
122      Y(J)=Y(J+1)                      EIGO1220
123      Y(J+1)=T-k(J)*Y(J+1)            EIGO1230
124      2070 CONTINUE                  EIGO1240
125      2060 CONTINUE                  EIGO1250
126      WRITE(6,600) ERR,I+E(I)        EIGO1260
127      IER=1                         EIGO1270
128      2074 CONTINUE                  EIGO1280
129      DO 2082 J=1,N                  EIGO1290
130      2082 V(J+1)=Y(J)              EIGO1300
131      DO 2083 I=1,N                  EIGO1310
132      DO 2065 J=1,N                  EIGO1320
133      A(I,J)=V(I,J)                EIGO1330
134      2063 CONTINUE                  EIGO1340
135      601 FORMAT(1PE12.3)             EIGO1350
136      CALL DECOMP(N)                EIGO1360
137      CALL SOLVE(N,OX,C,IE)         EIGO1370
138      IF(IE.EQ.1) GO TO 5000       EIGO1380
139      CALL IMPRUV(N,OX,C,DIGITS)   EIGO1390
140      RETURN                         EIGO1400
141      5000 CONTINUE                  EIGO1410
142      DO 1001 I=1,N                  EIGO1420
143      305 C(I)=OX(I)                EIGO1430
144      DO 1001 J=1,N                  EIGO1440
145      A(I,J)=V(I,J)                EIGO1450
146      1001 CONTINUE                  EIGO1460
147      CALL DMATI(N+1,C)             EIGO1470
148      RETURN                         EIGO1480
149      ENTRY FUC11(DT,M1,M2)        EIGO1490
150      DO 303 I=M1,M2                EIGO1500
151      OX(I)=0.0                     EIGO1510
152      DO 304 K=1,N                  EIGO1520
153      21 PKT=E(K)*DT                EIGO1530
154      IF(PKT.LE.(-70.0)) GO TO 304  EIGO1540
155      OX(I)=OX(I)+C(K)*V(I,K)*DEXP(PKT) EIGO1550
156      304 CONTINUE                  EIGO1560
157      303 CONTINUE                  EIGO1570
158      RETURN                         EIGO1580
159      600 FORMAT(1H /'! EIGEN VECTOR DOES NOT CONVERGE!', ERR =',
160      *     1PE12.3,' I =',I5,' E(I) =',1PE12.3/) EIGO1590
160      END                           EIGO1600
                                         EIGO1610

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 2

\* SOURCE STATEMENT \*

```

1      SUBROUTINE DECOMP(NN)                                     DECO0010
C** SUB. DECOMP, SOLVE, IMPRUV, SING, SOLVE THE SYSTEMS OF LINEAR DECO0020
C EQUATIONS. (COPIED FROM G.E.FORSYTHE AND C.B.MOLER, COMPUTER DECO0030
C SOLUTION OF LINEAR ALGEBRAIC SYSTEMS.) DECO0040
C DECO0050
C DECO0060
C DECO0070
2      DOUBLE PRECISION A+UL DECO0080
3      DOUBLE PRECISION SCALES, ROWNRM, BIG, SIZE, PIVOT, EM DECO0090
4      DIMENSION SCALES(50) DECO0100
5      COMMON/CMFE/A(50,50),UL(50,50) DECO0110
6      COMMON/DES/IPS(50) DECO0120
7      N=NN DECO0130
8      DO 5 I=1,N DECO0140
9      IPS(I)=I DECO0150
10     ROWNRM=0.0 DECO0160
11     DO 2 J=1,N DECO0170
12     UL(I,J)=A(I,J) DECO0180
13     IF (ROWNRM>DABS(UL(I,J))) 1,2,2 DECO0190
14     1 ROWNRM=DABS(UL(I,J)) DECO0200
15     2 CONTINUE DECO0210
16     IF (ROWNRM) 3,4,3 DECO0220
17     3 SCALES(I)=1.0D0/ROWNRM DECO0230
18     GO TO 5 DECO0240
19     4 CALL SING(1) DECO0250
20     SCALES(I) = 0.0 DECO0260
21     5 CONTINUE DECO0270
22     NM1=N-1 DECO0280
23     DO 17 K=1,NM1 DECO0290
24     BIG=0.0 DECO0300
25     DO 11 I=K,N DECO0310
26     IP=IPS(I) DECO0320
27     SIZE=DABS(UL(IP,K))*SCALES(IP) DECO0330
28     IF (SIZE>BIG) 11,11,10 DECO0340
29     10 BIG=SIZE DECO0350
30     IDXPIV=I DECO0360
31     11 CONTINUE DECO0370
32     IF (BIG) 13,12,13 DECO0380
33     12 CALL SING(2) DECO0390
34     GO TO 17 DECO0400
35     13 IF (IDXPIV=K) 14,15,14 DECO0410
36     14 J=IPS(K) DECO0420
37     IPS(K)=IPS(IDXPIV) DECO0430
38     IPS(IDXPIV)=J DECO0440
39     15 KP=IPS(K) DECO0450
40     PIVOT=UL(KP,K) DECO0460
41     KP1=K+1 DECO0470
42     DO 16 I=KP1,N DECO0480
43     IP=IPS(I) DECO0490
44     EM=UL(IP,K)/PIVOT DECO0500
45     UL(IP,K)=-EM DECO0510
46     DO 16 J=KP1,N DECO0520
47     IF (DABS(EM),LE,1.0E-20) EM=0.0 DECO0530
48     IF (DABS(UL(KP,J)),LE,1.0E-20) UL(KP,J)=0.0 DECO0540
49     UL(IP,J)=UL(IP,J)+EM*UL(KP,J) DECO0550
50     16 CONTINUE DECO0560
51     17 CONTINUE DECO0570
52     KP=IPS(N) DECO0580
53     IF (UL(KP,N)) 19,18,19 DECO0590
54     16 UL(KP,N)=1.0D-20 DECO0600
* SOURCE STATEMENT (DECOMP )* DECO0610
55     19 RETURN
56     END

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 22

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE SOLVE(NN,B,X,IE)          SLV00010
2      DOUBLE PRECISION A+UL               SLV00020
3      DOUBLE PRECISION B+X               SLV00030
4      DOUBLE PRECISION SUM              SLV00040
5      DIMENSION B(1),X(1)                SLV00050
6      COMMON/CMP/A(50,50),UL(50,50)       SLV00060
7      COMMON/DES/IPS(50)                 SLV00070
8      IE=0                                SLV00080
9      N=NN                                SLV00090
10     NP1=N+1                            SLV00100
11     IP=IPS(1)                           SLV00110
12     X(1)=B(IP)                          SLV00120
13     DO 2 I=2,N                           SLV00130
14     IP=IPS(I)                           SLV00140
15     IM1=I-1                            SLV00150
16     SUM=0.0                             SLV00160
17     DO 1 J=1,IM1                         SLV00170
18     1 SUM=SUM+UL(IP,J)*X(J)           SLV00180
19     2 X(I)=B(IP)-SUM                  SLV00190
20     10 IP=IPS(N)                         SLV00200
21     IF(UL(IP,N),EQ,0.0D0) GO TO 100    SLV00210
22     X(N)=X(N)/UL(IP,N)                 SLV00220
23     DO 4 IBACK=2,N                      SLV00230
24     I=NP1-IBACK                         SLV00240
25     IP=IPS(I)                           SLV00250
26     IP1=I+1                            SLV00260
27     SUM=0.0                             SLV00270
28     DO 3 J=IP1,N                        SLV00280
29     3 SUM=SUM+UL(IP,J)*X(J)           SLV00290
30     IF(UL(IP,I),EQ,0.0D0) GO TO 100    SLV00300
31     4 X(I)=(X(I)-SUM)/UL(IP,I)        SLV00310
32     RETURN                               SLV00320
33     100 IE=1                            SLV00330
34     RETURN                               SLV00340
35     END                                  SLV00350

```

FACOM 230-75 (M7) FORTRAN-D -750715~ V06-L05 75.11.28 PAGE 10

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE IMPRUV(NN,B,X,DIGITS)           [MP00010
2      DOUBLE PRECISION A,UL                   [MP00020
3      DOUBLE PRECISION B,X,R,DX,T             [MP00030
4      DOUBLE PRECISION SUM,ATJ,XJ             [MP00040
5      DIMENSION B(1),X(1),R(50),DX(50)        [MP00050
6      COMMON/CMP/A(50,50),UL(50,50)          [MP00060
7      N=NN                                     [MP00070
8      EPS=1.E-7                                [MP00080
9      ITMAX=3                                 [MP00090
10     C                                         [MP00100
11     XNORM=0.0                               [MP00110
12     DO 1 I=1,N                             [MP00120
13     XI=DABS(X(I))                         [MP00130
14     1 XNORM=AMAX1(XNORM,XI)                [MP00140
15     IF (XNORM) 3.2.3                      [MP00150
16     2 CONTINUE                                [MP00160
17     GO TO 10                                [MP00170
18     C                                         [MP00180
19     3 DO 9 ITER=1,ITMAX                  [MP00190
20     DO 5 I=1,N                           [MP00200
21     SUM=0.0                                 [MP00210
22     DO 4 J=1,N                           [MP00220
23     AIJ=A(I,J)                            [MP00230
24     XJ=X(J)                                [MP00240
25     4 SUM=SUM+AIJ*XJ                      [MP00250
26     SUM=S(1)-SUM                          [MP00260
27     5 R(I)=SUM                            [MP00270
28     CALL SOLVE(N,R,DX,1E)                 [MP00280
29     DXNORM=0.0                            [MP00290
30     DO 6 I=1,N                           [MP00300
31     T=X(I)                                [MP00310
32     X(I)=X(I)+DX(I)                      [MP00320
33     X1=DABS(X(I)-T)                      [MP00330
34     6 CONTINUE                                [MP00340
35     IF (ITER-1) 8.7.8                    [MP00350
36     7 DIGITS=- ALOG10(AMAX1(DXNORM/XNORM+EPS)) [MP00360
37     8 IF (DXNORM-EPS*XNORM) 10.10.9       [MP00370
38     9 CONTINUE                                [MP00380
39     C                                         [MP00390
40     10 RETURN                                [MP00400
41     END                                     [MP00410
42                                         [MP00420

```

FACOM 230-75 (NY) FORTRAN-D -750715- V06-L05 75.11.26 PAGE 21

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE SING(IWHY)                               SNG00010
2      11 FORMAT(54H MATRIX WITH ZERO ROW IN DECOMPOSE.    ) SNG00020
3      12 FORMAT(54H SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE. ) SNG00030
4      13 FORMAT(54H NO CONVERGENCE IN IMPROV. MATRIX IS NEARLY SINGULAR. ) SNG00040
5      IOUT=6                                         SNG00050
6      GO TO (1,2,3) ,IWHY                            SNG00060
7      1 *WRITE(100,11)                                SNG00070
8      GO TO 10                                     SNG00080
9      2 *WRITE(100,12)                                SNG00090
10     GO TO 10                                    SNG00100
11     3 *WRITE(100,13)                                SNG00110
12     10 RETURN                                     SNG00120
13     END                                         SNG00130

```

PACOM 250-75 CRAY FORTRAN-D 4750715- V06-L05 75.11.28 PAGE 4

\* SOURCE STATEMENT \*

```

1      SUBROUTINE DMATINV(M,N)
C
C** CALCULATE INVERSE MATRIX OF A + AND BY USE OF THIS INVERSE MATRIX
C   SOLVE THE SYSTEM OF LINEAR EQUATION. (AX = B + X = INVERSE(A)*B )
C** INPUT AND OUTPUT
C   M --- DIMENSION OF MATRIX
C   N --- TWO CALCULATES THE INVERSE MATRIX BUT DOES NOT SOLVE THE
C         SYSTEM OF LINEAR EQUATION.
C   A --- SOLVES AX=B USING THE INVERSE MATRIX OF A.
C   A --- MATRIX (INPUT) AND ALSO INVERSE OF A (OUTPUT).
C   B --- VECTOR (INV)
C
C
2      DOUBLE PRECISION A(1),B(1),DETERM,A(1),T,SWAP,PIVOT
3      DOUBLE PRECISION B(1),DETERM,A(1),T,SWAP,PIVOT
4      DIMENSION B(1)
5      DIMENSION PIVOT(500),INDEX1(500),INDEX2(500),PIVOT(500)
6      COMMON/CMTA/AL(50,50),BL(50,50)
7      EQUIVALENCE(1ROW,JROW),(1COLUMN,JCOLUMN),(A(1),T,SWAP)
8      INITIALZATION
9      N=NN
10     DO 20 J=1,N
11     20 PIVOT(J)=0
12     DO 555 I=1,N
13     C   SEARCH FOR PIVOT ELEMENT
14     A(1)=0.0
15     DO 105 J=1,N
16     IF((PIVOT(J),EN,1)) GO TO 105
17     DO 100 K=1,N
18     IF((PIVOT(K)-1,80+100,740)
19     80 IF(DABS(A(1)),GE,DABS(A(J,K))) GO TO 100
20     IROW=J
21     JCOLUMN=K
22     A(1)=A(J,K)
23     CONTINUE
24     105 CONTINUE
25     PIVOT(1COLUMN)=PIVOT(1COLUMN)+1
26     C   INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
27     IF(IROW,EN,1COLUMN) GO TO 260
28     DO 200 L=1,N
29     SWAP=A(1ROW,L)
30     A(1COLUMN,L)=SWAP
31     SWAP=A(1COLUMN)
32     A(1COLUMN)=B(1COLUMN)
33     B(1COLUMN)=SWAP
34     260 INDEX1(I)=IROW
35     INDEX2(I)=1COLUMN
36     PIVOT(I)=A(1COLUMN)*(1COLUMN)
37     C   DIVIDE PIVOT ROW BY PIVOT ELEMENT
38     A(1COLUMN,1COLUMN)=1.0
39     DO 350 L=1,N
40     IF(DABS(A(1COLUMN,L)),LT,1,E-300) A(1COLUMN,L)=0.0
41     350 A(1COLUMN,L)=A(1COLUMN,L)/PIVOT(I)
42     IF(N,EN,0) GO TO 380
43     A(1COLUMN)=B(1COLUMN)/PIVOT(I)
44     C   REDUCE NON-PIVOT ROWS
45     380 DO 550 L=1,N
46     IF(L,1,EN,1COLUMN) GO TO 550

```

DMT00010  
DMT00020  
DMT00030  
DMT00040  
DMT00050  
DMT00060  
DMT00070  
DMT00080  
DMT00090  
DMT00100  
DMT00110  
DMT00120  
DMT00130  
DMT00140  
DMT00150  
DMT00160  
DMT00170  
DMT00180  
DMT00190  
DMT00200  
DMT00210  
DMT00220  
DMT00230  
DMT00240  
DMT00250  
DMT00260  
DMT00270  
DMT00280  
DMT00290  
DMT00300  
DMT00310  
DMT00320  
DMT00330  
DMT00340  
DMT00350  
DMT00360  
DMT00370  
DMT00380  
DMT00390  
DMT00400  
DMT00410  
DMT00420  
DMT00430  
DMT00440  
DMT00450  
DMT00460  
DMT00470  
DMT00480  
DMT00490  
DMT00500  
DMT00510  
DMT00520  
DMT00530  
DMT00540  
DMT00550  
DMT00560  
DMT00570  
DMT00580  
DMT00590

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 5

\* SOURCE STATEMENT (DMAT) \*

```

44      T=A(L1+ICOLUMN)
45      A(L1+ICOLUMN)=0.0
46      DO 450 L=1,N
47      IF(DABS(A(ICOLUMN+L)),LT,1.E-300) A(ICOLUMN+L)=0.0
48      IF(DABS(T),LT,1.E-300) T=0.00
49      450 A(L1+L)=A(L1+L)-A(ICOLUMN+L)*T
50      IF(M,EW,0) GO TO 550
51      R(L1)=B(L1)-R(ICOLUMN)*T
52      550 CONTINUE
53      555 CONTINUE
C     INTERCHANGE COLUMNS
54      DO 710 I=1,N
55      L=N+I-1
56      IF(INDEX1(L),EQ,INDEX2(L)) GO TO 710
57      JROW=INDEX1(L)
58      JCOLUMN=INDEX2(L)
59      DO 700 K=1,N-
60      SWAP=A(K,JROW)
61      ACK,A(JROW)=ACK,JCOLUMN
62      700 ACK,JCOLUMN=SWAP
63      710 CONTINUE
64      740 RETURN
65      END

```

DMT00600  
DMT00610  
DMT00620  
DMT00630  
DMT00640  
DMT00650  
DMT00660  
DMT00670  
DMT00680  
DMT00690  
DMT00700  
DMT00710  
DMT00720  
DMT00730  
DMT00740  
DMT00750  
DMT00760  
DMT00770  
DMT00780  
DMT00790  
DMT00800  
DMT00810  
DMT00820

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 16

\* SOURCE STATEMENT \*

```

1      SUBROUTINE PLTSRT
2      DIMENSION BUFFER(1024)
3      CALL PLOTS(BUFFER(1),1024)
4      CALL PLOT(100.0,15.0,-3)
5      RETURN
6      END

```

PLT00010  
PLT00020  
PLT00030  
PLT00040  
PLT00050  
PLT00060

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 17

\* SOURCE STATEMENT \*

```

1      SUBROUTINE PLTEND
2      CALL PLOT(0,0.0,0,999)
3      RETURN
4      END

```

PLT00070  
PLT00080  
PLT00090  
PLT00100

EACOM 230-75 (M7) FORTRAN-D -750/1D- V6a-LU5 75.11.26 PAGE 18

## \* SOURCE STATEMENT \*

```

1      SUBROUTINE PLTREC(TE,XMAX,JIMAX)          PLT00110
2      LOGICAL CARBON,OXYGEN,IRON,MOLIB          PLT00120
3      COMMON/COLM/CHRHOL,OXYGEN,IRON,MOLIB       PLT00130
4      DIMENSION XT(200),YU(50,200)              PLT00140
5      DIMENSION YMAX(50),MUL(50),FFN(50)          PLT00150
6      COMMON/REPL/TM(200),YLX(50,200)           PLT00160
7      COMMON/PLT/PLT
8      DATA WIDTH,HEIGHT/270.0,120.0/
9      DATA YFP2,YFP4,YFP6,YFP8,YFP10/2.0,4.0,6.0,8.0,10.0/
10     DATA AFP0,AFP1,AFP2,AFP3,AFP4,AFP5,AFP7/0.0,-1.0,-2.0,-3.0,
11        -4.0,-5.0,-6.0,-7.0/                      PLT00200
12     *                                             PLT00210
13     DATA XFP10/1.0/                           PLT00220
14     IF(TE.LT.950.0) RETURN                   PLT00230
15     CALL PLOT(350.0,0.0,-3)                  PLT00240
16     IF(IRON) CALL PLOT(350.0,0.0,-3)          PLT00250
17     IF(MOLIB) CALL PLOT(350.0,0.0,-3)          PLT00260
18     IF(MOLIB) GO TO 330                     PLT00270
19     CALL PLOT(270.0,0.0,2)                  PLT00280
20     IF(IRON) CALL PLOT(540.0,0.0,2)          PLT00290
21     330 CONTINUE                            PLT00300
22     IF(MOLIB) CALL PLOT(540.0,0.0,2)          PLT00310
23     CALL PLOT(270.0,-2.0,2)                 PLT00320
24     CALL PLOT(240.0,0.0,3)                  PLT00330
25     CALL PLOT(240.0,-2.0,2)                 PLT00340
26     CALL PLOT(210.0,0.0,3)                  PLT00350
27     CALL PLOT(210.0,-2.0,2)                 PLT00360
28     CALL PLOT(180.0,0.0,3)                  PLT00370
29     CALL PLOT(180.0,-2.0,2)                 PLT00380
30     CALL PLOT(150.0,0.0,3)                  PLT00390
31     CALL PLOT(150.0,-2.0,2)                 PLT00400
32     CALL PLOT(120.0,0.0,3)                  PLT00410
33     CALL PLOT(120.0,-2.0,2)                 PLT00420
34     CALL PLOT(90.0,0.0,3)                   PLT00430
35     CALL PLOT(90.0,-2.0,2)                  PLT00440
36     CALL PLOT(60.0,0.0,3)                   PLT00450
37     CALL PLOT(60.0,-2.0,2)                  PLT00460
38     CALL PLOT(30.0,0.0,3)                   PLT00470
39     CALL PLOT(30.0,-2.0,2)                  PLT00480
40     IF(MOLIB) GO TO 300                     PLT00490
41     CALL NUMBERC(28.0,-5.0,2.0,XFP7,0.0,-1)   PLT00500
42     CALL NUMBERC(28.0,-5.0,2.0,XFP6,0.0,-1)   PLT00510
43     CALL NUMBERC(28.0,-5.0,2.0,XFP5,0.0,-1)   PLT00520
44     CALL NUMBERC(118.0,-5.0,2.0,XFP4,0.0,-1)  PLT00530
45     CALL NUMBERC(148.0,-5.0,2.0,XFP3,0.0,-1)  PLT00540
46     CALL NUMBERC(178.0,-5.0,2.0,XFP2,0.0,-1)  PLT00550
47     CALL NUMBERC(208.0,-5.0,2.0,XFP1,0.0,-1)  PLT00560
48     CALL NUMBERC(238.0,-5.0,2.0,XFP0,0.0,-1)  PLT00570
49     CALL NUMBERC(268.0,-5.0,2.0,XFP10,0.0,-1) PLT00580
50     CALL SYMRCL(100.0,-20.0,4.0,6HLOG(T),0.0,6) PLT00590
51     300 CONTINUE                            PLT00600
52     CALL PLOT(0.0,0.0,3)                   PLT00610
53     IF(MOLIB) GO TO 333                   PLT00620
54     CALL PLOT(0.0,120.0,2)                PLT00630
55     333 CONTINUE                            PLT00640
56     IF(MOLIB) CALL PLOT(0.0,240.0,2)          PLT00650
57     CALL PLOT(0.0,100.0,3)                PLT00660
58     CALL PLOT(-2.0,80.0,3)                PLT00670
59     CALL PLOT(-2.0,80.0,2)                PLT00680
60

```

FACOM 230-75 (M7) FORTRAN-D -750715- V96-L05 75.11.28 PAGE 19

\* SOURCE STATEMENT (PLTRE) \*\*

```

59      CALL PLOT(0, 60, 0,3)          PLT00700
60      CALL PLOT(-2,0,60,0,2)        PLT00710
61      CALL PLOT(0,0, 40,0,3)        PLT00720
62      CALL PLOT(-2,0,40,0,2)        PLT00730
63      CALL PLOT(0,0, 20,0,3)        PLT00740
64      CALL PLOT(-2,0,20,0,2)        PLT00750
65      IF(MOLIB) GO TO 350          PLT00760
66      CALL NUMBER(-5.0, 18.0,2.0,YFP2, 0.0,-1)    PLT00770
67      CALL NUMBER(-5.0, 38.0,2.0,YFP4, 0.0,-1)    PLT00780
68      CALL NUMBER(-5.0, 58.0,2.0,YFP6, 0.0,-1)    PLT00790
69      CALL NUMBER(-5.0, 78.0,2.0,YFP8, 0.0,-1)    PLT00800
70      CALL NUMBER(-5.0, 98.0,2.0,YFP10,0.0,-1)    PLT00810
71      CALL SYMBOL(-10.0,120.0,14.0,0.6,MLOG(J),0.0,6)  PLT00820
72      CALL SYMBOL(70.0,-40.0,4.0,17HTE = EV+0.0+17)  PLT00830
73      CALL NUMBER(93.0,-40.0,4.0,TE +0.0,2)        PLT00840
74      FNPL01=FLDAT(NPLOT)          PLT00850
75      CALL NUMBER(0.0,-80.0,5.0,FNPL01,0.0,-1)       PLT00860
76      350 CONTINUE                  PLT00870
77      IF(MOLIB) CALL NUMBER(93.0,-15.0,4.0,TE+0.0,2)   PLT00880
78      DO 100 K=1,KMAX             PLT00890
79      DO 100 J1=1,J1MAX           PLT00900
80      XT(J1)=0.0                 PLT00910
81      YO(K,J1)=0.0               PLT00920
82      100 CONTINUE                  PLT00930
83      DO 10 J1=1,J1MAX           PLT00940
84      10 XT(J1)=ALOG10(TM(J1))+6.0      PLT00950
85      DO 11 J1=1,J1MAX           PLT00960
86      XT(J1)=XT(J1)/XT(J1MAX)     PLT00970
87      XT(J1)=XT(J1)*WJDTH       PLT00980
88      IF(IRUN) XT(J1)=XT(J1)+2.0  PLT00990
89      IF(MOLIB) XT(J1)=XT(J1)*2.0  PLT01000
90      11 CONTINUE                  PLT01010
91      DO 12 K=1,KMAX             PLT01020
92      DO 12 J1=1,J1MAX           PLT01030
93      IF(YO(K,J1).LE.0.0) GO TO 120  PLT01040
94      YO(K,J1)=ALOG10(YO(K,J1))  PLT01050
95      GO TO 12                      PLT01060
96      120 YO(K,J1)=0.0            PLT01070
97      12 CONTINUE                  PLT01080
98      DO 13 K=1,KMAX             PLT01090
99      DO 13 J1=1,J1MAX           PLT01100
100     YO(K,J1)=YO(K,J1)/12.0     PLT01110
101     YO(K,J1)=YO(K,J1)*WTGH1  PLT01120
102     IF(MOLIB) YO(K,J1)=2.0*YO(K,J1)  PLT01130
103     13 CONTINUE                  PLT01140
104     DO 14 K=1,KMAX             PLT01150
105     YMAX(K)=ANMAX1(YO(K+1)+YO(K+2))  PLT01160
106     DO 14 J1=2,J1MAX           PLT01170
107     YMAX(J1)=ANMAX1(YMAX(K)+YO(K+J1))  PLT01180
108     14 CONTINUE                  PLT01190
109     DO 15 K=1,KMAX             PLT01200
110     DO 15 J1=1,J1MAX           PLT01210
111     YMP=YMAX(K)+0.001         PLT01220
112     YM0=YMAX(K)-0.001         PLT01230
113     IF(YO(K,J1).LE.YMP.AND.YO(K,J1).GE.YM0) M01(K)=J1  PLT01240
114     15 CONTINUE                  PLT01250
115     DO 17 K=1,KMAX             PLT01260
116     CALL PLOT(XA(1),YO(K,1)+3)  PLT01270
117     DO 16 J1=2,J1MAX           PLT01280

```

FACOM 230-75 (M7) FORTRAN-D -750715- V06-L05 75.11.28 PAGE 20

\* SOURCE STATEMENT (PLTRE)\*

118	16 CALL PLOT(XT(J1),YO(K,J1),2)	PLT01290
119	FPN(K)=FLOAT(K)	PLT01300
120	IX=MJ1(K)	PLT01310
121	XN=XT(IX)+3.0	PLT01320
122	YN=YO(K,IX)-3.0	PLT01330
123	CALL NUMBER(XN,YN,4.0,FPN(K),0.0,-1)	PLT01340
124	17 CONTINUE	PLT01350
125	RETURN	PLT01360
126	END	PLT01370