JPL PUBLICATION 77-11

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

JPL PUBLICATION 77-11

Proposed Computer Model for Electric Discharge Atomic Vapor Lasers

Kenneth G. Harstad

March 15, 1977

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

Prepared Under Contract No NAS 7-100
National Aeronautics and Space Administration

PREFACE

The work described in this report was performed by the Control and Energy Conversion Division of the Jet Propulsion Laboratory.

CONTENTS

Appendixes

Figures

ABSTRACT

A detailed computer model for the rate kinetics of an atomic vapor laser excited by electrical discharge is proposed. The model equations are defined and the computer program structure is discussed.

PROPOSED COMPUTER MODEL FOR ELECTRIC DISCHARGE ATOMIC VAPOR LASERS

Introduction

It is the goal of the present program of computer modeling to numerically simulate the gross features of atomic vapor lasers excited by electrical discharge. Due to the inherent complexity of interactions in such a laser, the extent of attainment of this goal is uncertain. Nevertheless, much qualitative information on some of the processes taking place can be expected, leading to a better understanding of the physics of this laser type.

Limitations on computation time and computer program storage make inevitable considerable simplification of real processes or characteristics of the laser. As a first step, the proposed model focuses on rate processes, i.e., the change of electronic excited state populations through inelastic collisions and radiative interaction. Only temporal variations are included; the system is taken spatially homogeneous. In any particular atom or ion, levels of similar configuration (e.g. terms) are handled as a unit with populations assumed proportional to degeneracies. The set of level units of the atom or ion is split into three distinct groups: an upper subset in equilibrium with the free continuum states, a middle subset assumed quasistationary and a lower subset requiring numerical integration of rate equations to determine populations. The continuum plus the upper subset are treated together and will be called the "extended continuum" [1]. The relaxation times between states in the extended continuum and also in the quasistationary group are short relative to any other physical time of interest in the system. Populations in intermediate levels are small compared to the lower levels or the extended continuum. The occupation probability (based on equilibrium) is high for the low levels due to the Boltzmann factor and high for free or near free states due to large degeneracies. The short relaxation times and large possible population fluxes lead to equilibrium of extended continuum states among themselves. Fluxes in the quasistationary group are relatively small and the net rates into and out of a particular level are nearly the same;

populations change in a slow adiabatic manner so that they may be approximated by algebraically solving the rate equations assuming null rates [2]. It is assumed in the model that the degree of ionization of the laser gas is always sufficient (greater than 10^{-4}) to neglect the effects of atom-atom collisions on rates [1]; thus excited state equilibrium is characterized by the electron temperature without regard to heavy particle temperatures. This amount of ionization also assures a Maxwellian electron velocity distribution [2]. Neglect of atom-atom inelastic collisions in the rate equations does not mean that they are necessarily inconsequential in sublevel relaxation or relaxation between close levels in a unit. An equilibrium extended continuum also requires a sufficiently small charged particle mean free path.

Rate Equations

Let the population of level unit n, specie s with parent ion core charge ze be $N_n^{s,z}$. The extended continuum is denoted by n*; the corresponding reference number density is $N_{n*}^{s,z} \equiv N_1^{s,z+1}$. Note that the actual density of the extended continuum is

$$(1 + \varepsilon z^{3/2})\ N_1^{s,z+1}$$

where

$$\varepsilon = N_e(\pi e^2\ \ell_D/kT_e)^{3/2}/3 << 1\ [1].$$

This holds for an extended continuum lower principal quantum number of about 5 to 7 and is based on the Debye lowering of ionization potentials [3]. Here N_e is the electron density and ℓ_D is the Debye length. A Saha factor is defined as

$$\beta_n^{s,z}(T_e) = \frac{g_n^{s,z}}{2\ g_1^{s,z+1}} \left(\frac{h^2}{2\pi\ m_e kT_e}\right)^{3/2} \exp\ (I_n^{s,z}/kT_e)$$

where the g's are degeneracies, h the Planck constant, m_e the electron mass and $I_n^{s,z}$ is the ionization potential of level n. At equilibrium, $N_n^{s,z} = \beta_n^{s,z}\ N_e N_1^{s,z+1}$

with $\beta^{s,z}_{n*} N_e \equiv 1$. For hydrogenic levels, the preceding degeneracy factor equals the square of the principal quantum number. If the ionization rate coefficient from n is $K^{s,z}_{n\infty}(T_e)$, the total excitation rate coefficient into the extended continuum is

$$K^{s,z}_{nn*} = K^{s,z}_{n\infty} + \sum_{m \geq n*} K^{s,z}_{nm} \quad .$$

Excitation coefficients K_{nm} are discussed in Appendix D. The maximum principal quantum number is $(z\ell_D/2a_o)^{1/2}$. At equilibrium, $N_m K_{mn} = N_n K_{nm}$, giving deexcitation coefficients. The net radiative emission rate m → n is $A_{mn} E_{mn}$ where A is the Einstein spontaneous emission coefficient and E is the escape factor (Appendix B). If a particular transition is coupled into the laser cavity, E is replaced by a factor dependent on the laser intensity (Appendix C). In the following, E will denote a generalized emission factor. The emission rate depends on the transition line profile unless the transition is optically thin; line profiles and broadening for the model are the topics of Appendix A.

A normalized population $\overline{N}^{s,z}_n$ is defined by $N^{s,z}_n \equiv N_e \beta^{s,z}_n \overline{N}^{s,z}_n$ so that at equilibrium $\overline{N}^{s,z}_n = N^{s,z+1}_1$ (all n). Also, $R^{s,z}_n = (d/dt)\overline{N}^{s,z}_n$ and $R_e = (d/dt)\ell n\, N_e$ are net rate functions.

For n < n*, the basic rate equations for an atom or ion are

$$\begin{aligned} R^{s,z}_n = & \sum_{k \leq n*} N_e K^{s,z}_{nk} \left(\overline{N}^{s,z}_k - \overline{N}^{s,z}_n \right) \\ & + \sum_{\substack{k>n \\ k \leq n*}} \beta^{s,z}_k A^{s,z}_{kn} E^{s,z}_{kn} \overline{N}^{s,z}_k \Big/ \beta^{s,z}_n \\ & - \overline{N}^{s,z}_n \left(\sum_{k<n} A^{s,z}_{nk} E^{s,z}_{nk} + R_e - \left(\frac{3}{2} + \frac{I^{s,z}_n}{kT_e} \right) \frac{d\,\ell n\, T_e}{dt} \right) \quad . \end{aligned}$$

The quasistationary group is $n^* > n \geq \overline{n}$. The rate $(d/dt)N_n^{s,z} = 0$ for this group corresponds to setting R_n, R_e and $d\ell nT_e/dt$ null in the above. Rate equations for all atoms and ions, plus those for cavity intensities (Appendix C) and the electron energy equation for $d\ell nT_e/dt$ together define the model. An implicit assumption in the model is that for any given core, only one excited electron is in a discrete state, the rest being free. The radiative recombination term is given by $E_{n^*n} \equiv 1$,

$$A_{n^*n} = \frac{32}{3\sqrt{3}} \frac{\alpha^3 Ryd}{h} z^4 \frac{G_n}{p_n^3} N_e \left(\frac{h^2}{2\pi m_e kT_e}\right)^{3/2} \exp\left(\frac{I_{n^*}}{kT_e}\right) F\left(\frac{\Delta E_{n^*}}{kT_e}\right)$$

where α is the fine structure constant, p_n the principal quantum number, G_n the Gaunt factor, ΔE_{n^*n} the energy gap to the extended continuum, and $F(x) \equiv e^x E_1(x)$ [1].

Charge and Specie Conservation

The constraints produced by charge and specie conservation are given next. It is assumed that the highest stage of ionization of a specie atom is an ion ground state denoted by N_1^s. Total specie number density is N_s and maximum parent charge number is z_s. Given N_s, N_1^s is found

$$N_s = (1 + \varepsilon z_s^{3/2})\, N_1^s + N_e\left(S_a^s + \varepsilon S_b^s\right) ,$$

$$S_a^s \equiv \sum_{z=1}^{z_s} \sum_{n<n^*} \beta_n^{s,z}\, \overline{N}_n^{s,z} \quad ,$$

$$S_b^s \equiv \sum_{z} (z-1)^{3/2}\, \beta_1^{s,z}\, \overline{N}_1^{s,z} \quad .$$

The terms proportional to ε give the populations of the bound states parts of the extended continuum. The electron density is obtained using

$$C - \varepsilon A = N_e(1 + D - \varepsilon B) \ ,$$

$$S_c^s \equiv \sum_z \sum_n (z - 1)\, \beta_n^{s,z}\, \overline{N}_n^{s,z} \ ,$$

$$S_d^s \equiv \sum_z (z - 2)\, (z - 1)^{3/2}\, \beta_1^{s,z}\, \overline{N}_1^{s,z} \ ,$$

$$A = \sum_s z_s^{3/2}\, N_s \Big/ \left(1 + \varepsilon z_s^{3/2}\right) ,$$

$$B = \sum_s \left(S_d^s - z_s\, S_b^s + z_s^{3/2}\, \left(S_a^s + \varepsilon S_b^s\right) \Big/ \left(1 + \varepsilon z_s^{3/2}\right)\right) \ ,$$

$$C = \sum_s z_s N_s \ ,$$

$$D = \sum_s \left(z_s\, S_a^s - S_c^s\right)$$

In the above, ε is calculated with $N_e = C/(1 + D)$ and then the final value of N_e is found.

Neglecting terms of order ε, the conservation and rate equations may be combined to give

$$R_e = \sum_{s,z} (z_s + 1 - z) \left(N_e\, \alpha_I^{s,z} - N_1^{s,z+1}\, \alpha_R^{s,z}\right) .$$

$$\alpha_I^{s,z} = \sum_n \beta_n^{s,z} \overline{N}_n^{s,z} K_{nn*}^{s,z}$$

is an ionization coefficient, and

$$\alpha_R^{s,z} = \sum_n \beta_n^{s,z} \left(N_e K_{nn*}^{s,z} + \beta_{n*}^{s,z} A_{n*n}^{s,z} \Big/ \beta_n^{s,z} \right).$$

is a recombination coefficient. Similarly, "rate heating" of the free electron gas D is given as

$$D_e = N_e \sum_{s,z} \left(N_1^{s,z+1} d_R^{s,z} - N_e d_I^{s,z} \right) \quad ,$$

$$d_R^{s,z} = \sum_n \beta_n^{s,z} I_n^{s,z} \left(N_e K_{nn*}^{s,z} + \beta_{n*}^{s,z} A_{n*n}^{s,z} \Big/ \beta_n^{s,z} \right)$$

is a recombination heating coefficient and

$$d_I^{s,z} = \sum_n \beta_n^{s,z} I_n^{s,z} \left(\overline{N}_n^{s,z} K_{nn*}^{s,z} + \sum_{k<n*} K_{nk}^{s,z} \left(\overline{N}_n^{s,z} - \overline{N}_k^{s,z} \right) \right)$$

is a coefficient of energy absorption for inelastic collisions.

Reduction of Rate Equations

The rate equations may be rewritten in the form (for given s,z)

$$R_n = \sum_{k<n*} M_{nk} \overline{N}_k + F_n \overline{N}_1^{+}$$

where the + superscript denotes z+1. Nulling R_e and $d\ell nT_e/dt$ in matrix M gives matrix M'. The rate equations are of the stiff type, i.e., characterized by multiple time scales. Integrators designed to handle this type of problem [4] require the Jacobian matrix $J_{nm} = \partial R_n/\partial \overline{N}_m$ to scale the numerical integration time step. For the quasistationary levels, $R_n = 0$ for M replaced by M', giving

$$\overline{N}_j = -\sum_{k,m} M'^{-1}_{jk}\, M'_{km}\, \overline{N}_m - \overline{N}_1^{+} \sum_{k} M'^{-1}_{jk}\, F_k$$

where $j,k \geq \overline{n}$ and $m < \overline{n}$. Substituting this expression for the quasistationary populations into the rate equations for the lower levels ($n < \overline{n}$) gives the reduced set of equations

$$R_n = \sum_{m<\overline{n}} H_{nm}\, \overline{N}_m + G_n\, \overline{N}_1^{+}$$

where

$$H_{nm} = M_{nm} - \sum_{j,k} M_{nj}\, M'^{-1}_{jk}\, M'_{km} \quad ,$$

$$G_n = F_n - \sum_{j,k} M_{nj}\, M'^{-1}_{jk}\, F_k \quad .$$

The Jacobian of the reduced set P may be found with the aid of the matrix

$$T_{jm} \equiv \partial\overline{N}_j \Big/ \partial\overline{N}_m \quad (j \geq \overline{n},\ m < \overline{n}) \quad .$$

Using the chain rule,

$$P_{nm} = J_{nm} + \sum_{j} J_{nj} \, T_{jm} \qquad (n < \overline{n})$$

and

$$0 = J'_{km} + \sum_{j} J'_{kj} \, T_{jm} \qquad (k \geq \overline{n}).$$

(Primes have the same significance for J as M.)

Combining,

$$P_{nm} = J_{nm} - \sum_{j,k} J_{nj} \, J'^{-1}_{jk} \, J'_{km}$$

where the matrix J is calculated for the quasistationary levels independent. In general, the reduction of the rate equations leads to two matrix inversions.

The dependence of the rates on N_e and $\overline{N}_1^+$ leads to cross terms in the reduced Jacobian for different s,z. Due to the algebraic complexity of the model, the Jacobians are formed assuming fixed transition line profiles. The error in this assumption should not be large since the short time scales are associated with upper states which are usually optically thin.

Energy Equation and the Penning Effect

The electron energy equation is taken from 13-moment solutions to the Boltzmann equation [1,5]. Included in the equation are the rate of change of enthalpy, elastic electron-heavy particle energy exchange, continuum radiation, collisional-radiative heating D_e, and joule dissipation. Provision is made in the model to input either the current density or an imposed electric field or to bypass the energy equation by providing the temporal variation of the electron temperature. Heavy particle temperatures are assumed equal and constant.

Due to the possible importance of Penning excitation exchange in lasers having noble gases [6], source code for one and two electron excitation and/or ionization of a receptor atom by a metastable donor is part of the model. This code is placed in blocks for easy removal. The probability of double ionization is apparently small [7].

Discussion

The model for the laser rate kinetics has been designed to be of a very general nature. The desire for flexibility and to test concepts led to the program form. Atom or ion level structure, oscillator strengths, atomic constants, etc. are input by distinct subprograms that may be changed at will to provide an arbitrary mix of elements and ionization stages. In order that this be done with maximum storage economy, this information is transferred to common blocks on an end-to-end basis. Further flexibility is obtained by the use of Univac Fortran V Parameter variables for dimension information, DO loop constants, index constants and the like. These variables are replaced by their assigned values at compilation. The specification statements for the Parameter variables, array dimensions, etc. are not part of subprogram source code but are placed in Fortran procedures (by the Procedure Definition Processor) for inclusion at compilation. Thus only a single set of source code defining the procedures need be changed before compilation to modify the storage requirements of the assembled program. It is thus possible to minimize storage for a given set of atoms and ions with ease and without error. The numbers of level units in the divers groups can also be readily changed.

The program is designed to treat lower level groups as stiff and Jacobian matrices are calculated. One purpose of the model is to determine if and/or when use of the quasistationary approximation for intermediate groups removes the stiffness so that simpler integrators may be used. Success of this tactic has been recently reported [8], but its use in cases of fast pumping is questionable [1]. Use of the quasistationary group reduces storage requirements; it may or may not help insofar as reducing the program running time. It is necessary to determine the effect on running times and also constraints (e.g. values of

$\bar{n}$ and n*) of the three group model. Another item of consequence is the conditions for an emission factor E of unity. Calculations related to this factor are relatively extensive. After initial calculations, simplifications may ensue.

A detailed listing of the model is provided in Appendix E.

The program is in the process of being debugged. A sample run utilizing a single argon atom only with twelve level units has been apparently successful. Use of multiple species has led to difficulties with the matrix inversion aspect of the program; this anomaly and the configuration of output data remain as program problems. Numerical results will be given in a later report.

References

1. K.G. Harstad, Plasma Phys. 14 857 (1972), Plasma Phys. 16 1177 (1974).
2. L.M. Biberman, I.T. Yakubov and V.S. Vorobev, Proc. IEEE 59 555 (1971).
3. H.R. Griem, Plasma Spectroscopy, McGraw-Hill Book Co., New York (1964).
4. F.T. Krogh, JPL Technical Memorandum 33-479 (1971).
5. K.G. Harstad, JPL Technical Report 32-1318 (1968).
6. E. Sovero, C.J. Chen and F.E.C. Culick, J. Appl. Phys. 47 4538 (1976).
7. K. Gerard and H. Hotop, Chem. Phys. Lett. 43 175 (1976).
8 J.W. Hiestand and A.R. George, AIAA J. 14 1153 (1976).
9. P.R. Berman and W.E. Lamb, Jr., Phys. Rev. A 2 2435 (1970), Phys. Rev. A 4 319 (1971); M. Borenstein and W.E. Lamb, Jr., Phys. Rev. A 5 1311 (1972); P.R. Berman, Phys. Rev A 5 927 (1972), Phys. Rev. A 6 2157 (1972).
10. V.A. Alekseev, T.L. Andreeva and I.I. Sobelman, Sov. Phys. JETP 35 325 (1972), 37 413 (1973).
11. G. Traving, Plasma Diagnostics, W. Lochte-Holtgreven Editor, North-Holland Pub. Co., Amsterdam (1968), pp. 66-134.
12. A.P. Kazantsev, Opt. Spect. 37 589 (1975).
13. R.R. Teachout and R.T. Pack, At. Data 3 195 (1971).
14. T. Holstein, Phys. Rev. 83 1159 (1951).
15. T. Hiramoto, J. Phys. Soc. Japan 26 785 (1969).
16. K.G. Harstad, JQSRT 9 1273 (1969).
17. C.E. Klots and V.E. Anderson, J. Chem. Phys. 56 120 (1972).
18. V.I. Sysun, Opt. Spect. 33 320 (1972).
19. Y.B. Golubovskii and R.I. Lyagushchenko, Opt. Spec. 38 628 (1975).
20. L.W. Casperson, J. Appl. Phys. 46 5194 (1975).
21. M. Gryzinski, Phys. Rev. 138 A336 (1965).
22. R. Goldstein, JPL Technical Report 32-1372 (1969).

23. R. Goldstein, Private Communication.

24. P. Mansbach and J. Keck, Phys. Rev. 181 275 (1969).

25. B.K. Thomas and J.D. Garcia, Phys. Rev. 179 94 (1969).

26. E.A. Mason, R.J. Munn and F.J. Smith, Phys. Fluids 10 1827 (1967).

APPENDIX A
LINE BROADENING

It is desired to have a complete, yet compact, description of the excited states phase shifts (homogeneous broadening) and velocity changes (inhomogeneous broadening). The most rigorous approach is that using the quantum density matrix [9,10]. The need for simplicity in the computer model leads to consideration of classical theory for the estimation of line profiles. In the model line center frequency shifts are neglected and the Voigt integral is used to enfold the Lorentz impact and Doppler profiles. The theory is presented in [11]; results of interest are summarized below.

For an inverse power law potential, $V = e^2 a_0^{m-1} C_m r^{-m}$, of emitter-perturber interaction the optical or phase shift cross section is πb_W^2 where the Weisskopf radius

$$b_W = \left(\frac{|C_m|}{p_m} \frac{\alpha c}{v} \right)^{\frac{1}{m-1}} a_0 \quad .$$

The symbols are: e-electron charge, a_o - Bohr radius, r - distance between interacting atoms, v - relative velocity of atoms, c - velocity of light, α - fine structure constant. Constant p_m is proportional to the critical phase shift that defines the effective interaction range: $p_3 = 0.318$, $p_4 = 0.411$, $p_6 = 0.533$, $p_{12} = 0.773$. This impact result is valid for angular frequencies in a region at the line center, $\Delta\omega < v/b_W$, and for perturber densities N_p smaller than b_W^{-3}. These limits are of small consequence for the problem under discussion. The optical collision frequency (line semi-half width) is $\pi N_p \left\langle v b_W^2 \right\rangle$ where the velocity distribution is taken as Maxwellian. For elastic collisions the phase shifts between the upper and lower states of a line subtract and $C_m = C_{m(\text{upper})} - C_{m(\text{lower})}$. In the case of an ion emitter and electron perturbers, a correction for the hyperbolic collision path may be necessary [11]. If b_C is the Coulomb collision radius, $b_C > b_W$, the cross section increases by a factor of approximately $2.3(b_C/b_W)^s$, $s = (2m - 4)/(2m - 3)$, giving an inverse square root dependence of the width on electron temperature. Interactions considered in the model are:

1) resonance broadening (ground state atom perturber and like atom emitter), $|C_3| = 6f_{abs}\mathrm{Ryd}(g_\ell/g_u)^{1/2}/(\pi\Delta E)$; (this interaction involves excitation exchange and is not elastic; the interaction of upper and lower states is added);

2) electron-quadrapole broadening, $|C_3| \simeq \langle r_e^2\rangle q_L/a_o^2$ where $q_L^2 = \frac{2}{15} L(L + 1)/(L(L + 1) - 3/4)$ for the emitter;

3) electron quadratic Stark broadening,

$$C_4 = 2\left(\sum_{\mathrm{lower}} f_{ems}(\mathrm{Ryd}/\Delta E)^2 - \sum_{\mathrm{upper}} f_{abs}(\mathrm{Ryd}/\Delta E)^2\right)$$

sums are over all emitter bound states;

4) Van der Waals broadening (ground state atom perturber and unlike atom emitter), $C_6 \simeq -\alpha_p\langle r_e^2\rangle/a_o^5$;

5) atom perturber and ion quadrapole emitter, $|C_6| \simeq Z\langle r_e^2\rangle\alpha_p q_L/a_0^5$. (The polarization interaction $C_4 = -Z\alpha_p/(2a_0^3)$ is state independent and gives no broadening.)

Broadening by ions is small and is neglected. Symbols are: f_{ems}, f_{abs} - emission and absorption oscillator strengths, Ryd - Rydberg unit of energy, ΔE - transition energy, g_ℓ, g_u - lower and upper degeneracies, $\langle r_e^2\rangle$ - average square of the active electron radius of a state, L - total angular momentum quantum number, α_p - perturber atom polarizability, Z - ion charge number.

The parameters in the resonance broadening constant C_3 refer to the transition between the emitter state and the (perturber) ground state. If the emitter state effective principal quantum number $n > 3$, the oscillator strength is small and the excitation exchange interaction is weak; further, the emitter radius is large. In this situation, repulsive electron exchange effects become dominant [12]. For $n \lesssim (\alpha c/v)^{1/4}$, these effects can be estimated by using $b_W \approx \langle r_e^2 \rangle^{1/2}$. Effects at larger n [12] are not relevant to the model. The maximum of the widths calculated using the two interactions is used.

Broadening by electrons is estimated as the largest of the quadrapole and quadratic Stark interactions. Inelastic collisions between sublevels are frequently important in Stark broadening. Provision is made in the model to calculate the Stark width from curve fits to the calculations of Griem [3], assuming additive upper and lower state interactions. Curve fits of the form $CN_e n^k T_e^{\,r}$ give good results for many levels (typically $k = 5$ and $r = 0.4$).

The form of the Van der Waals constant C_6 is an approximation valid in the limit of large energy level separations [3,11]. As with the case of resonance broadening, electron exchange effects can be important, especially for small α_p. Only the simple form above is used; more precise analysis requires involved calculation and more knowledge of potentials than is commonly available. The polarizability α_p can be found from data [13] or estimated as $4a_0^3\,(\mathrm{Ryd}/\Delta E_r)^2$ where the resonance level-ground energy gap pertains [3]. Also, $\langle r_e^2 \rangle / a_0^2 \simeq n^2(5n^2 + 1 - 3\ell(\ell + 1))/2Z^2$ where Z refers to the parent ion.

Finally, the total homogeneous line width is the sum of the widths obtained from the formulas above plus that due to the frequencies of quenching of the upper and lower states. The quenching mechanisms are that of spontaneous emission (natural broadening), induced emission or absorption (power broadening) and inelastic excitation and de-excitation between levels ("rate" broadening). Note that the quenching mechanisms give a Lorentz profile without constraint (this includes resonance broadening).

APPENDIX B

RADIATIVE ESCAPE FACTORS

An escape factor for a given transition gives the effect of induced radiative processes on the emission rate. The factors are calculated assuming a uniform medium and are averaged over the gas volume. For a non-inverted line of large optical depth, trapping gives an escape factor much less than unity. Conversely, amplified spontaneous emission from an inverted line gives an escape factor greater than unity.

The theory of line trapping is discussed in [14-19]. Radiation loss from any particular line may be written as $N_u \, h\nu_o \, A_{u\ell} \, E$ where N_u is the upper state density, $h\nu_o$ the line center energy, $A_{u\ell}$ the Einstein spontaneous emission coefficient and escape factor $E = \langle \exp(-\tau_\nu) \rangle$, τ_ν being the optical depth (negative for an inverted line). The average is taken over frequency and solid angle and for the model also over spatial position. A cylindrical geometry and a Doppler line profile gives $E = 0.90/(\tau_o \sqrt{\ell n \, \tau_o})$ where τ_o is the line center optical depth based on the radius. A Lorentz profile gives $E = 0.63/\sqrt{\tau_o}$. Trapping decreases as τ_o drops to order unity and for small depth $E = 1 - \langle \tau_\nu \rangle$. In the model, the maximum of the Doppler and Lorentz factors (large τ_o limit, each calculated using the appropriate limiting line profile) is used to approximate trapping for the Voigt profile; then a smooth interpolation to the small depth limit is performed so that the entire range of optical parameters is covered in a convenient yet reasonable manner. Note that at large depths, escape is determined by the line wings where the homogeneous broadening dominates since it is of distinct proportion. The approximation used gives neglible error at the larger optical depths where trapping is most pronounced. (Some error may occur due to non-impact optical interactions not properly accounted for in the model; the effect of this is considered slight since it can manifest itself only in cases of near complete trapping, i.e., at small optical transition rates.)

An interpolation scheme between small and large gain limits is also used for inverted transitions. A discussion on the calculation of $\langle \exp(|\tau_\nu|) \rangle$ for $|\tau_\nu| \gtrsim 1$ follows. Let the cylinder center line optical depth value for the transition line center be g, L the cylinder length (which defines g), R the cylinder radius, r the distance from an emission point to a cylinder boundary point, $\bar{p}(\nu)$ the line profile relative to the line center ν_o so that $\bar{p}(\nu_o) = 1$, and the cylinder aspect ratio $b = L/(2R)$. Then $E = \langle \exp(g\bar{p}r/L) \rangle$. Averaging over frequency, $E = \langle F(gr/L) \rangle$ where

$$F(s) = \frac{k}{\sqrt{\pi}} \int_{-\infty}^{\infty} \bar{p}\,(x)\ \exp\,(s\ \bar{p}(x))\ dx \quad ,$$

$k \equiv \sqrt{\pi}\, w_v p_o$: w_v the Voigt half-width and p_o the line center absorption coefficient, $x \equiv (\nu - \nu_o)/w_v$. For a Lorentz profile $\bar{p} = (1 + x^2)^{-1}$, $F(s) = \sqrt{\pi}\, k \exp\,(s/2)\ I_o\ (s/2) \approx k \exp\,(s)/\sqrt{s}$ for large s. For a Doppler profile, $\bar{p} = \exp(-x^2)$ and it can be easily shown that F approaches the same limit for large s. Since $F(0) = 1$, the large gain limit is calculated using $F(s) \simeq 1 + k(e^s - 1)/\sqrt{s}$. In the Lorentz limit, this approximation gives a relative error $\leq$20% for $0 < s \leq 1$, $\leq$5% for $s \geq 2$. In the Doppler limit, the error is as large as 34% at smaller s. The similar results for the profile limits is to be expected since it is the line center that controls the amplification process.

Let the cylinder center line be parallel to the z-axis in rectangular coordinates, offset in the x direction by amount ωR, $0 \leq \omega \leq 1$. The equations for the cylinder wall are

$$x = 2R\ q\ \cos\,\theta$$

$$y = 2R\ q\ \sin\,\theta$$

$$q \equiv \frac{1}{2}\left(\omega \cos\,\theta + (1 - \omega^2 \sin^2\,\theta)^{1/2}\right)$$

$$z = r\,\cos\,\phi, \qquad 2Rq = r\,\sin\,\phi$$

Consider $0 \leq z \leq \upsilon L$, $0 \leq \upsilon \leq 1$; then for the cylinder side $\phi_o \leq \phi \leq \frac{\pi}{2}$ where $q\ \text{ctn}\ \phi_o = b\upsilon$. The solid angle differential is $\sin\,\phi d\phi d\theta$. The emitter position is characterized by the variables υ and ω; the volume average is given by integration over 2 $\omega d\omega d\upsilon$. For $H(s) = (\exp(s) - 1)/\sqrt{s}$, the excess side loss $E_{es} = (E - 1)_{side}$ is approximately

$$E_{es} = \frac{2k}{\pi} \int_0^1 \omega d\omega \int_0^\pi d\theta \int_0^1 d\upsilon \int_{\phi_o}^{\pi/2} \sin \phi d \phi H \left(\frac{gq}{b \sin \phi} \right).$$

The integrals over ω and θ can be transformed to a single integral over q and the integrals over υ and ϕ can be transformed to a single integral by integrati by parts. Using $u \equiv gq/b$, the result is

$$E_{es} = \frac{4kb}{\pi g} \int_0^{g/b} u^2 du \sqrt{1 - (bu/g)^2} \int_u^{\sqrt{u^2+g^2}} H(s) \frac{ds}{s^2} \left(\frac{1}{\sqrt{s^2 - u^2}} - \frac{1}{g} \right).$$

After removing the singularity at s = u by integration by parts, the remaining integration is done numerically. The results are given in Figure B1. It is seen that E_{es} is significant only for $g \gtrsim b$, corresponding to large transverse gain. For the low gain limit $E_{es} = \langle|\tau_\nu|\rangle_{side}$, or $E_{es} = g \langle\bar{p}(\nu)\rangle\langle r/L\rangle_{side}$. Th last factor is

$$\langle r/L\rangle_{side} = \frac{4}{\pi b^2} \int_0^1 q^2 dq \sqrt{1 - q^2} \left(\frac{b}{q} \tan^{-1} \frac{b}{q} - \frac{1}{2} \ln \left[1 + \left(\frac{b}{q} \right)^2 \right] \right).$$

After numerical evaluation for b > 1, the following curve fit gives this factor with an error less than 1%:

$$\langle r/L\rangle_{side} \doteq 0.66/b - 1.09/b^2 + 2.30/b^3 - 1.60/b^4 .$$

The average solid angle that the side subtends is $4\pi S_s$ where

$$S_s = \frac{4}{\pi b} \int_0^1 dq \sqrt{1 - q^2} \left(\sqrt{q^2 + b^2} - q \right) \doteq 1 - 0.4244/b + 0.172/b^2$$

$$- 0.090/b^3 + 0.025/b^4.$$

If a is the ratio of the Lorentz half-width to Doppler width, then

$$k \doteq (1 + 0.7304a + 0.5811a^2)/(1 + 1.2946a + 1.0299a^2)$$

and

$$\langle \overline{p} \rangle \doteq (0.707 + 1.25a)/(1 + 2.5a) \text{ within an error of } 2.3\%.$$

Similar calculations apply to the cylinder ends:

$$E_{ee} = \frac{2kg}{\pi b^2} \int_0^1 qdq \; (\cos^{-1} q - q \sqrt{1 - q^2}) \int_u^{\sqrt{u^2+g^2}} H(s) \frac{ds}{s^2} \quad ,$$

and

$$\langle r/L \rangle_{ends} = \frac{1}{\pi b^2} \int_0^1 qdq \; (\cos^{-1} q - q\sqrt{1 - q^2}) \; \ln \left[1 + \left(\frac{b}{q} \right)^2 \right]$$

$$\doteq (0.9428 + \ln b)/(8b^2) + 0.0132/b^4 \quad .$$

Results of numerical calculation of the excess end loss E_{ee} are given in Figure B2.

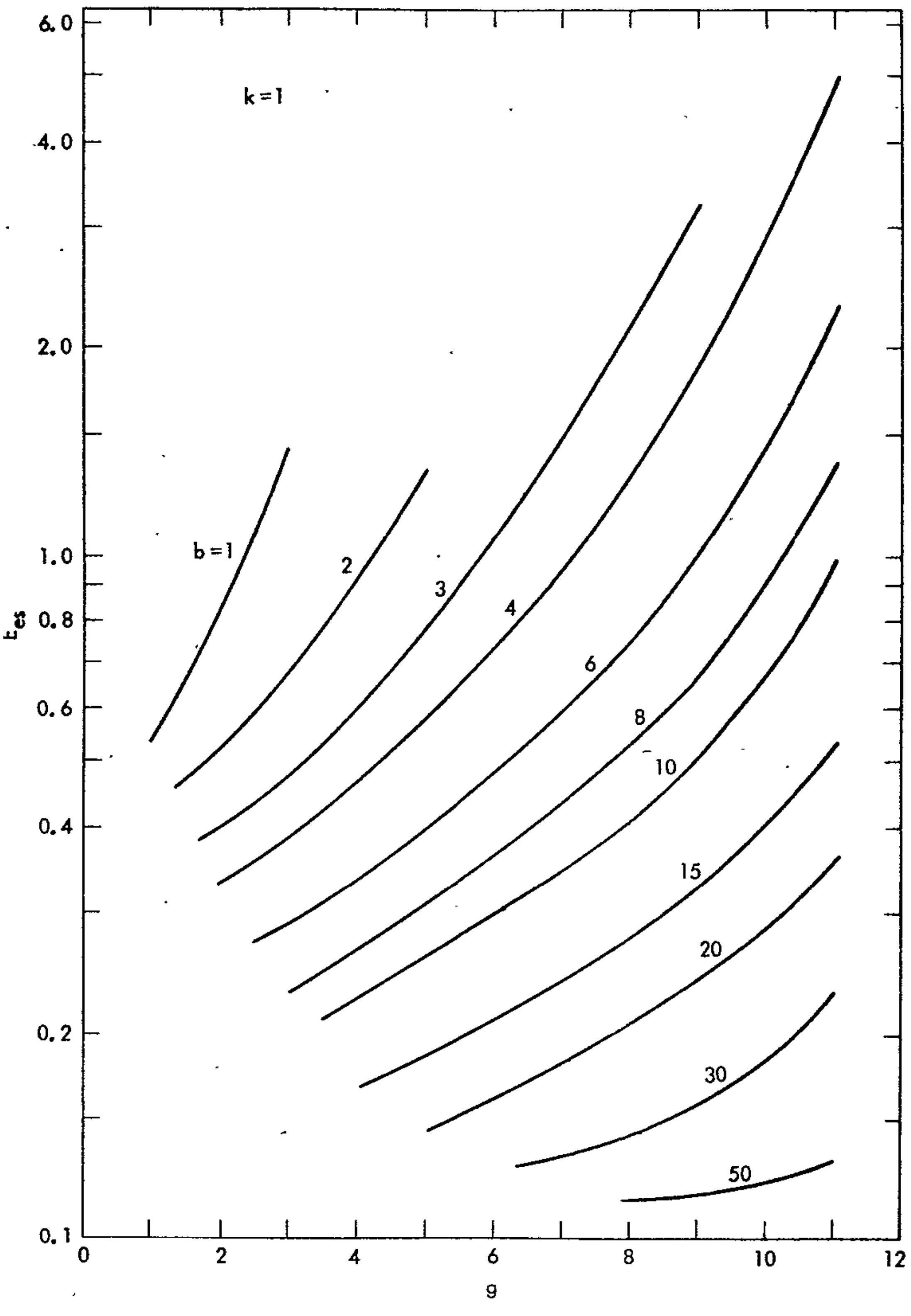

Fig. B1. Excess side emission factor for inverted transitions as a function of axial gain. Parameter b is the cavity aspect ratio.

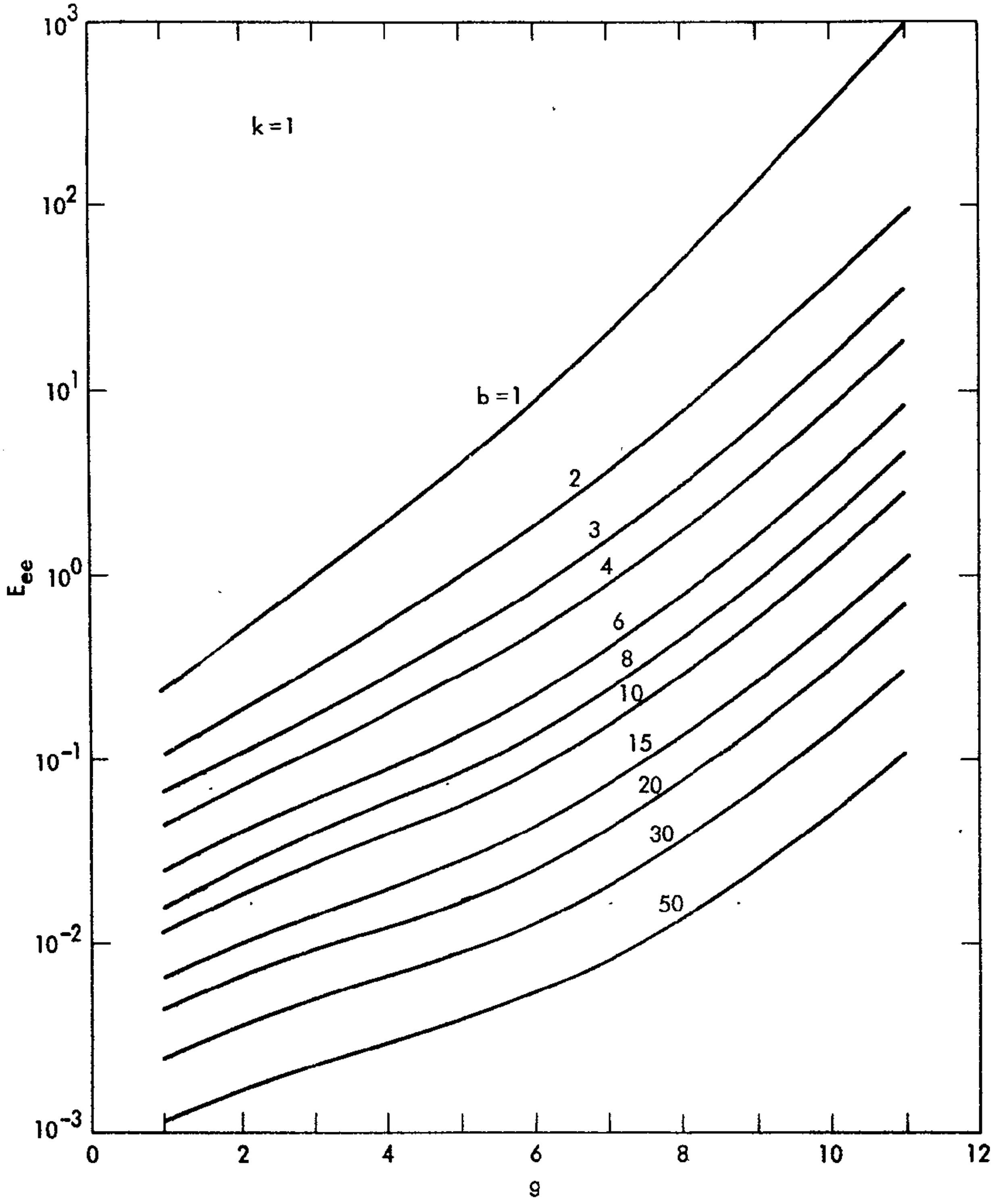

Fig. B2. Excess end emission factor for inverted transitions as a function of axial gain. Parameter b is the cavity aspect ratio.

APPENDIX C
CAVITY EQUATIONS

Quantum coherency effects are not considered in the modeling of the interaction of strong cavity electromagnetic fields with atomic excited state populations. Feedback from the optical cavity and saturation effects are estimated using rate type equations with steady state intensity profiles without velocity distribution or spatial hole burning. The present form of the cavity equations contain inherent inaccuracies that would be most pronounced for low pressure and/or very fast developing systems. This situation is dictated by the need for relative overall simplicity; future simplifications in other aspects of the model may lead to further development of the cavity equations.

Denote the total intensity for a given transition as $I = ch\nu_o N_{ph}$ where N_{ph} is the photon density. The corresponding spectral quantities are $I_\nu = ch\nu N_\nu$. For each direction of propagation,

$$\frac{dN_\nu}{dt} = \frac{c^3}{8\pi\,\nu^2} A_{u\ell}\, p(\nu)\ (N_u - g_u N_\ell / g_\ell) N_\nu + \eta\, A_{u\ell}\, N_u - N_\nu / t_c ,$$

A is the Einstein emission coefficient, p the transition profile, η the solid angle and polarization factor, and the cavity decay time $t_c = L/(c\ \ell n(1/\sqrt{R_1 R_2}))$ for length L and mirror reflectivities R_1 and R_2. The intensity is taken unpolarized and ν replaced by ν_o except in the profile functions. (For plane polarized light, $\eta \rightarrow \eta/2$.) Let the line center gain logarithm be

$$g = \frac{A_{u\ell}}{8\pi} \left(\frac{c}{\nu_o}\right)^2 (N_u - g_u N_\ell / g_\ell)\ L\ p_o \quad ,$$

$\Delta\nu$ the mode spacing, and the saturation parameter $s \equiv g/\ell n\ (1/\sqrt{R_1 R_2})$. Then summing over modes (frequencies):

$$\frac{dN_{ph}}{dt} = (\omega s - 1)\ N_{ph} / t_c + \eta_e\, A_{u\ell}\, N_u$$

where for mode intensities I_m,

$$\omega = \sum_m I_m\, p\,(\nu_m)/I\, p_o$$

and

$$\eta_e = \sum_m \eta(\nu_m)\Delta\nu = \int \eta d\nu$$

defines the total spontaneous emission noise into all active modes. Note that

$$I = \sum_m I_m$$

and $\omega \leq 1$. The mode spacing $\Delta\nu = cf/(4L)$ where the cavity length and mirror radii define f. For the model, $f = (x/(2 - x))^{1/2}$, $0 < x < 2$, x is the ratio of length L to the effective mean mirror radius. (This equation defines the mean radius.) Only one mode exists if $p_o\ \Delta\nu \gtrsim 1$. The intensity profile-transition profile overlap integral ω is approximated by using steady state saturation intensity expressions. For homogeneous broadening a set of longitudinal modes gives [20]:

$$\omega(s) = \left[1 - \coth\left(\frac{1}{p_o\Delta\nu}\right) \tanh\left(\frac{\sqrt{1 - s}}{p_o\Delta\nu}\right) \sqrt{1 - s}\right] \Big/ s \quad .$$

As $s \to 1$ (or $p_o\Delta\nu \to \infty$), $\omega \to 1$. This narrowing effect of saturation can be generalized by considering an intensity profile of the form $\bar{p}(\nu)/(1 - s\bar{p}(\nu))$. Then in the limit $s \to 0$, $\omega \to \langle\bar{p}\rangle$ (a function of the ratio of Lorentz to Doppler widths, Appendix B). Numerical integration is performed to find ω in the limit $p_o\ \Delta\nu \to 0$, a curve fit to the results is modified for finite $p_o\ \Delta\nu$ to give

$$\omega(s) \simeq 1 - (1 - s)^{0.3}\ (1 - \langle\bar{p}\rangle)/(1 + p_o\Delta\nu) \quad .$$

This is the generalized saturation narrowing used in the model.

The present cavity model can be expected to be a valid approximation at higher pressures where impact broadening is significant and at inversions not many times that at threshold. Different transverse modes have different diffraction losses; the number of modes taken must be equal to those of relatively small loss. This number would define the effective emission solid angle η_e. It must depend on the number of photon passes in the cavity, cavity geometry, wavelength, and transition width. At present the parameter η_e is input into the program. The single mode minimum value is $(\lambda/(2\pi R))^2$; an upper bound is the average end solid angle (Appendix B). Spontaneous emission noise is most important at the start of the intensity buildup and the larger values of η_e are most pertinent.

The model presents the maximal effect of saturation narrowing within the rate approximation; calculations with $\omega(s \equiv 0)$ may be used to determine the minimal effect.

APPENDIX D

CLASSICAL EXCITATION RATES

Expressions are needed for excitation rates between many pairs of levels. The expressions must be general in nature and relatively simple, yet reasonably accurate. The use of classical theory produces expressions that satisfy these constraints [1]. Provision is made in the model to use empirical excitation rates from ground states.

The present model utilizes the cross sections of Gryzinski [21]. Excitation from a level "ℓ" to all levels "u" and above (including the continuum states) is given by the coefficient [22]:

$$\overline{K}_{\ell u}\ (T_e) = g_e\ R\ (y,A) \Big/ \Delta E_{\ell u}^{\ 3/2} \quad ,$$

where g_e is the number of equivalent electrons in the lower state, $\Delta E_{\ell u}$ is the energy gap, $y \equiv \Delta E_{\ell u}/kT_e$, $A \equiv I_\ell/\Delta E_{\ell u} \geq 1$ for lower state ionization energy I_ℓ. The rate coefficient to level "u" alone is obtained by subtracting a similar expression with $u \rightarrow u + 1$: $K_{\ell u} = \overline{K}_{\ell u} - \overline{K}_{\ell u+1}$. Ionization is given by $\overline{K}$ with $A \equiv 1$. For $A \leq 10$ and $0.01 \leq y \leq 10$, a good analytical approximation to numerical evaluation of rate coefficients using a Maxwellian velocity distribution is [22]:

$$R = (3.84 \times 10^{-6}\ \mathrm{cm}^3\ \mathrm{eV}^{3/2}\ \mathrm{sec}^{-1})\ y^t\ e^{-y} \Big/ \left(A^{1/4}\ (y^2 + 7y/4 + 1/9)\right) ,$$

$$t = (A + 30)/(10A + 25).$$

Asymptotic expansions for small and large y yield respectively

$$R\ (y\ \mathrm{small}) = 4.39 \times 10^{-6} \sqrt{y}\ \ e^{-y} \ \left(1 - A\ \ln\ (Ay)/3\right) ,$$

$$R\ (y\ \mathrm{large}) = 2.93 \times 10^{-6}\ e^{-y} \Big/ (\sqrt{A}\ y^S), \quad S = (3A + 1)/(2A + 2).$$

These functions are suitably matched in the intermediate regions. The above excitation function may be applied to allowed transitions. Gryzinski's exchange cross sections give coefficients for forbidden transitions; a curve fit is made to numerical calculations [23].

Another classical calculation is that of Mansbach and Keck [24], where Monte Carlo trajectories are used in a three body (free electron, active electron, parent ion) system. For principal quantum numbers n_ℓ and n_u, excitation rate coefficient

$$K_{\ell u}\ (T_e) = 3.73 \times 10^{-7}\ \mathrm{cm}^3\ \mathrm{sec}^{-1} \left(\frac{\mathrm{Ryd}}{kT_e}\right)^{0.17} \frac{n_\ell^{4.66}}{n_u^{3}}\ e^{-y}$$

and for ionization

$$K_{\ell\infty}\ (T_e) = 1.87 \times 10^{-7}\ \mathrm{cm}^3\ \mathrm{sec}^{-1} \left(\frac{kT_e}{\mathrm{Ryd}}\right)^{0.83} n_\ell^{4.66}\ e^{-Ay} \quad .$$

(Calculations are made for hydrogen-like atoms.) These rates are smaller than that of Gryzinski for small energy gaps, $\Delta E \lesssim kT_e$ [24]. Since coefficients are large under this condition and the levels involved tend to be quasi-stationary [1], the difference is probably of small significance. Within the classical approximation, the Mansbach-Keck rates may be regarded as theoretically more rigorous. However, collective interactions between free electrons and a highly excited active electron make the accuracy of any lone perturber collision theory doubtful for transitions between levels of large quantum number. The model may be easily modified to employ Mansbach-Keck or quantum Born excitation rates.

The form of the excitation coefficient as given by the R-function can be expected to be approximately correct for ions as well as atoms [25].

APPENDIX E

PROGRAM LISTING

The computer program is written in terms of the following units or reference values for physical quantities.

pressure	1 mbar = $10^2 \mathrm{Nm}^{-2} \approx 10^{-3}$ atm
density	10^{15} cm^{-3}
rate coefficient	10^{-7} cm^3 sec^{-1} (also Stark coefficient)
line width, relaxation time	10^8 sec^{-1}
temperature	10^4 °K
atomic mass	1 amu = 1.66053×10^{-27} kg
length (except wavelength)	1 cm
wavelength	1 μm
elastic cross section	10^{-15} cm^2
current density	1 abamp $\mathrm{cm}^{-2} = 10^5$ amp m^{-2}
electric conductivity	1 mho cm^{-1}
electric field	1 kV m^{-1}
magnetic field	1 web m^{-2}
intensity	1 W cm^{-2}

The following symbol conventions are used generally throughout the program. The few exceptions are noted later.

AN	vector of ground state populations $N_1^{s,z}$ of atom or ion type s,z
ANB	vector of normal excited state populations $\overline{N}_n^{s,z}$
ANC	vector of specie densities N_s
ANE	electron density N_e
ANP	vector of densities of maximum ionization N_1^s
CAR	laser cavity aspect ratio b
CL	cavity cylinder length
CMR	cavity mean mirror reflectivity $\sqrt{R_1R_2}$
NBAR	vector of level indices $\overline{n}$
NCM	maximum number of species (distinct nuclear cores)
NSP	number of atom and ion types

NSTAR	vector of level indices n*
T	heavy particle temperature
TE	electron temperature
Z or ZC	charge number z of parent ion
ZS	maximum charge number z_s of specie

In the Fortran procedure source deck SPECS, values of Parameter variables are to satisfy the constraints:

If

$NL_k \equiv$ maximum number of level units of atom or ion type k as given by the data input subprograms,

$KM \equiv$ maximum number of equations to be integrated,

Then

$$NLM \geq \max\ (NL_k) \quad ,$$

$$ND \geq \sum_k NL_k \quad ,$$

$$M \geq 1 + NSP + \sum_k (NL_k + 3)^2$$

$$NT \geq \max\ \left((NLM + 3)^2,\ NLM^2 + NLM + 20\right) \quad ,$$

$$KSV \geq 1 + 3\ NLM + 5\ NLM^2/4 \quad .$$

In the format specifications in procedure SPEC3, the second record of format 62 and the third record of format 63 refer to the vector ANC and must have data length equal to NCM; the third and fourth record of 62 and the fourth record of 63 refer to vectors NBAR, NSTAR with data lengths determined by NSP. The program is written for a maximum of four cavity equations. This is reflected in the last dimension of ICAV, the dimension of WL, the last record of format 62 and corresponding last record of 63 referring to ICAV, the last dimensions of GP and LU in SPEC4 and the dimension of ANPH in SPEC7. Symbols in the procedures are defined in this appendix as the need arises. Common blocks LEVELS, EESND and PENCOM are used for input data and calculated primary variables, TEMPS, INCR, STORE and MSTOR are used for scratch and data transfer between subroutines.

Along with the proper specification statements, a particular problem requires that appropriate data and function subroutines be included in the program assembly. These subroutines are described in succeeding pages. Their specific names are required by the Collector for substitution in dummy subroutine calls. The proper Collector directives are essential to a problem.

```
-PDP,IFL SPECS
SPEC1 PROC
      PARAMETER NSP=3,M=1591,ND= 60,NLM=20,NT=529,NPL=NLM+1,NSQ=NLM*NLM,
     C    MB=4*ND
      DIMENSION DS(M),INDS(M),INX(NSP,8),BP(MB),TMP(NT)
      COMMON/LEVELS/INX,DS,BP/TEMPS/TMP
      EQUIVALENCE (DS,INDS)
 END
SPEC2 PROC
      PARAMETER NCM=3
      DIMENSION IC(NCM),ANC(NCM),ANB(ND),NBAR(NSP),NSTAR(NSP),ICAV(3,4),
     C    CFST(4),BT(ND),WL(4)
      COMMON/EESND/IC,NC,NBAR,NSTAR,ICAV,NCAV,CL,CAR,CMR,ETA,FCAV,CGM,
     C PS,CFST,TE,T,ANE,ANC,ANB,BT,WL
 END
SPEC3 PROC
 62   FORMAT(1P8E10.3/1P3E10.3/3I3/3I3/12I3)
 63   FORMAT(1H1,2X14HCAVITY LENGTH=1PE10.3,2X16HLENGTH/DIAMETER=
     C    1PE10.3,2X13HMIRROR REFL.=1PE10.3,2X11HMODE RATIO=1PE10.3,
     C    2X11HRES. RATIO=1PE10.3/
     C    3X3HTE=1PE10.3,2X2HT=1PE10.3,2X3HNE=1PE10.3/
     C    3X3HNC=3(2X1PE10.3)/
     C    3X5HNBAR=3(2XI3),2X6HNSTAR=3(2XI3)/
     C    3X16HCAVITY INDICES- 4(2X3I3) )
 END
SPEC4 PROC
      PARAMETER NHSQ=(NLM+NSQ)/2,NP4=NHSQ+21
      DIMENSION AN(NSP),W(NHSQ),WD(NHSQ),PHI(NHSQ),R(ND),
     C    GP(3,4),LU(2,4)
      COMMON/INCR/R
      EQUIVALENCE (WD(1),TMP(21)),(WD(1),PHI(1)),(W(1),TMP(NP4)),
     C    (GP(1,1),TMP(1)),(LU(1,1),TMP(13))
      IFN(I,L)=((I-1)*(2*L-I))/2
 END
SPEC5 PROC
      LOGICAL LPN
      COMMON/PENCOM/LPN,RPEN,JRC,JRL,JDC,JDL
 END
SPEC6 PROC
      PARAMETER KSV= 561
      DIMENSION SV(KSV)
      COMMON/STORE/SV
 END
SPEC7 PROC
      PARAMETER KM=50,NRSTM=2*NLM*NLM+KM*KM+3*(NLM+KM)
      DIMENSION ANR(KM),ANPH(4),STRMTX(NRSTM)
      COMMON/MSTOR/STRMTX
 END
```

Subroutine DATAIN is used to input and collect the data defining the particular problem at hand. The dummy calls LDUMYn are made equivalent to the names of the atom and ion data subroutines used in the problem by the Collector processor which generates the executable absolute program. It is assumed NSP ≤ 6. The argument list in the data subroutines is as follows:

PQ	vector effective principal quantum numbers based on term values
AQ	vector angular momentum quantum numbers (active electron)
AQT	vector total angular momentum quantum numbers
DG	vector degeneracies
ID	integer vector with the properties: ID(1) is the number of ground level equivalent electrons; otherwise, nominal ID(j) = 1, ID(j) < 0 if the Stark broadening of level j is to be calculated by the curve fit to Griem's tables (Appendix A), $\lvert$ID(j)$\rvert$ = 2 if empirical excitation rates from the the ground level to j are to be used
W	vector Gaunt factor for radiative recombination into levels [1]
FR	array of transition oscillator strengths, these are input $\leq$ 0 for nonallowed transitions (array is two dimensional)
NL	dimension of arrays, i.e., maximum number of levels
GNLI	degeneracy of ground level of next ionization stage
GFF	overall Gaunt factor for recombination radiation,

$$= \frac{5}{6} \sum_{j<n^*} \omega_j G_j p_j^{-3}$$

for $p_j \equiv$ PQ(j),

ω_j is the relative weight of level j in its shell, and for hydrogenic levels the shell Gaunt factor

$$G_j = 1 - 0.173 p_j^{-2/3} - 0.050 p_j^{-4/3}$$

IS	integer parent ion charge number
AM	atomic mass
ALPHA	polarizibility (applies to atoms only, null for ions)
CS, RS, KS	Stark curve fit parameters (Appendix A)

The calls place data into temporary storage TMP, then a call to TRANS shifts the data into the array DS (INDS). DS(1) = NSP, DS(k + 1) = NL_k ($1 \leq k \leq$ NSP). The array INX(k, i) contains storage locations of the first elements of arrays for type k according to i=1: array is (IS, AM, GNLI, GFF, ALPHA, CS, RS, KS), i=2: PQ, i=3: AQ, i=4: AQT, i=5: DG, i=6: ID, i=7: W and i=8: FR. The oscillator strength for a transition m → n

$$|FR(m,n)| = |DS(m - 1 + NL_k \cdot (n - 1) + INX(k, 8))| \; .$$

For level units made from levels of different momentum quantum numbers ℓ, the average of the product $\ell(\ell + 1)$ is used to determine AQ(j) For each type k, a set of four vectors of broadening parameters is placed sequentially into the array BP. The first vector is the resonance broadening constant C_3, the second vector $<r_e^2>/a_o^2$, the third vector Stark constant C_4 and the fourth vector the sum of A-values to lower levels, i.e., the natural line width (Appendix A). Functions F1 and F2 are used to complete the sums over levels in C_4 for $j \geq n^*$. These functions are based on hydrogenic oscillator strengths. The vector IC gives the type index for atoms of the species and NC is the number of species.

Array ICAV inputs the transitions that are to be considered in the cavity equations. The first index of ICAV indicates the following: 1 for type index (or null), 2 for lower level index and 3 for upper level index. NCAV is the number of cavity equations. Symbol ETA is the effective noise emission factor for the cavity and CGM is the ratio of the length to effective mean mirror radius (Appendix C). The subroutine changes CGM to the mode spacing and calculates the cavity decay rate as FCAV, the number of photon passes as PS and the wavelengths as WL.

The dummy call to SDUMY is to be replaced by the Collector by EQUIL or QSTAT which initialize the level populations according to equilibrium or quasistationary calculations respectively. These subroutines are discussed later; they place the Saha factors in the vector BT.

The Penning effect inputs into PENCOM are logical constant LPN - true if the effect is to be considered, RPEN - Penning rate constant (negative for two electron excitation), JRC - receptor specie index, JRL - receptor level index, JDC - donor specie index and JDL - donor level index. The donor specie must follow the receptor specie at data input.

Subroutine DATAIN and corollary subroutines may be placed in a separate segment in the absolute program since they are needed only at the program start. Thus they do not represent any added storage in the program.

```
-TFOR,SI  SUB1
      SUBROUTINE DATAIN
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      INCLUDE SPEC3,LIST
      INCLUDE SPEC5,LIST
      INCLUDE SPEC7,LIST
      PARAMETER N1S1=NPL+9,N1S2=N1S1+NLM,N1S3=N1S2+NLM,N1S4=N1S3+NLM,
     C    N1S5=N1S4+NLM,N1S6=N1S5+NLM
      DIMENSION PQ(NPL),AQ(NLM),AQT(NLM),DG(NLM),ID(NLM),W(NLM),FR(NSQ)
      DIMENSION R(ND)
      COMMON/INCR/R
      EQUIVALENCE (IS, TMP(1)),(AM,TMP(2)),(GNLI,TMP(3)),(GFF,TMP(4)),
     C     (ALPHA,TMP(5)),(CS,TMP(6)),(RS,TMP(7)),(KS,TMP(8)),(PQ(1),
     C     TMP(9)),(AQ(1),TMP(N1S1)),(AQT(1),TMP(N1S2)),(DG(1),
     C     TMP(N1S3)),(ID(1),TMP(N1S4)),(W(1),TMP(N1S5)),
     C     (FR(1),TMP(N1S6)),     (JS,NSTRT),(ANR(1),STRMTX(1)),
     C     (ANPH(1),STRMTX(1))
110   FORMAT(L7,1PE10.3,4I3)
111   FORMAT(1H0,2X23HPENNING RATE CONSTANT =1PE10.3,5X10HRCPT CORE I3,
     C 5X11HRCPT LEVEL I3,5X9HDNR CORE I3,5X10HDNR LEVEL I3)
      F1(X,Y)=1.0/(1.0-X*X/(Y*(Y-1.0)))**4-1.0
      F2(X,Y)=0.49*X**3*(1.0-2.8*EXP(-1.4*AMAX1(Y-X,0.75)))*
     C        F1(X,X+AMAX1(Y-X,0.75))
      INDS(1)=NSP
      NSTRT=NSP+2
      L=1
C     DUMMY SUBR CALLS ARE TO BE CONVERTED BY COLLECTOR
      CALL LDUMY1(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
      IF(NSP.EQ.1) GO TO 5
      CALL LDUMY2(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
      IF(NSP.EQ.2) GO TO 5
      CALL LDUMY3(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
      IF(NSP.EQ.3) GO TO 5
      CALL LDUMY4(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
      IF(NSP.EQ.4) GO TO 5
      CALL LDUMY5(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
      IF(NSP.EQ.5) GO TO 5
      CALL LDUMY6(PQ,AQ,AQT,DG,ID,W,FR,NL,GNLI,GFF,IS,AM,ALPHA,CS,RS,KS)
      IF(L.GT.NSP.OR.NSTRT.GT.M-8-6*NL-NL*NL) GO TO 40
      CALL TRANS
  5   JS=0
      DO 30 K=1,NSP
      L=INDS(K+1)
      L1=INX(K,3)-1
      L2=INX(K,2)-1
      L3=INX(K,5)
      L4=INX(K,8)-L
      Z=INDS(L2-7)**2
```

```
      DO 20 I=1,L
      A=  Z/DS(L2+1)**2-  Z/DS(I+L2)**2
      IF(I.EQ.1) A=1.0E9
      B=SQRT(DS(L3)/DS(I+L3-1))
      BP(I+JS)=1.91*B*ABS(DS(I*L+L4))/A
      A=DS(I+L2)**2
      B=1.0-3.0*DS(I+L1)*(1.0+DS(I+L1))
      BP(I+JS+L)=A*(5.0*A+B)/(2.0*Z)
      B=0.0
      C=0.0
      DO 10 J=1,L
      A=  Z/DS(J+L2)**2-  Z/DS(I+L2)**2
      IF(I.EQ.J) A=1.0E5
      A=SIGN(A*A,A)
      IF(I.GT.J) C=C+A*ABS(DS(I+L*J+L4-1))
   10 B=B+2.0*ABS(DS(I+L*J+L4-1))/A
      B=B-F2(DS(I+L2),DS(L1))
      BP(I+JS+2*L)=B
   20 BP(I+JS+3*L)=6.39*C
      JS=JS+4*L
   30 CONTINUE
      GO TO 50
   40 WRITE(6,42)
   42 FORMAT(1H0,3X20HINCORRECT INPUT DATA)
      STOP
   50 JS=0
      DO 60 K=1,NSP
      L=INX(K,1)
      L=INDS(L)
      IF(L.NE.1) GO TO 60
      JS=JS+1
      IF(JS.GT.NCM) GO TO 40
      IC(JS)=K
   60 CONTINUE
      NC=JS
      READ(5,62) CL,CAR,CMR,ETA,CGM,TE,T,ANE,ANC,NBAR,NSTAR,ICAV
      DO 66 K=1,ND
   66 R(K)=0.0
      NCAV=0
      DO 70 K=1,4
   70 IF(ICAV(1,K).GT.0) NCAV=NCAV+1
      DO 80 K=1,NSP
      IF(NSTAR(K).GT.(INDS(K+1)+1)) NSTAR(K)=INDS(K+1)+1
   80 IF(NBAR(K).GT.NSTAR(K)) NBAR(K)=NSTAR(K)
C    DUMMY SUBR CALLS ARE TO BE CONVERTED BY COLLECTOR
      CALL SDUMY
      WRITE(6,63) CL,CAR,CMR,ETA,CGM,TE,T,ANE,ANC,NBAR,NSTAR,ICAV
      A=TVEF(1.0,CAR,1.0)
      IF(ABS(CGM-1.0).GE.1.0.OR.CMR.GT.1.0.OR.CMR.LE.0.0) GO TO 40
      CFST(1)=CSA(CAR,0)
      CFST(2)=CSA(CAR,1)
      CFST(3)=CSA(CAR,2)
      CFST(4)=CSA(CAR,3)
      READ(5,110) LPN,RPEN,JRC,JRL,JDC,JDL
      IF(.NOT.LPN) GO TO 96
      IF(ABS(RPEN).GT.0.0) GO TO 90
      LPN=.FALSE.
      GO TO 96
   90 K=IC(JRC)
```

```
      IF(RPEN.LT.0.0) GO TO 92
      IF(JRL.GT.NSTAR(K)) JRL=NSTAR(K)
      GO TO 94
   92 IF(K.EQ.NSP) GO TO 94
      IF(JRL.GT.NSTAR(K+1)) JRL=NSTAR(K+1)
   94 WRITE(6,111) RPEN,JRC,JRL,JDC,JDL
      IF(JRC.GE.JDC) GO TO 40
      IF(JDC.GT.NC)  GO TO 40
      K=IC(JDC)
      IF(JDL.GE.NSTAR(K)) GO TO 40
   96 FCAV=299.7925*ALOG(1.0/CMR)/CL
      CGM=SQRT(CGM/(2.0-CGM))
      CGM=74.948*CGM/CL
      PS=1.0/ALOG(1.0/CMR)
      IF(NCAV.LE.0) GO TO 98
      DO 97  J=1,NCAV
      K=ICAV(1,J)
      L=INX(K,1)
      C=INDS(L)**2
      L=INX(K,2)-1
      I=ICAV(2,J)
      K=ICAV(3,J)
      A=C/DS(I+L)**2-C/DS(K+L)**2
   97 WL(J)=9.11267E-2/A
   98 L=0
      DO 100 K=1,NSP
  100 L=L+NBAR(K)-1
      L=L+NCAV+1
      IF(L.GT.KM) GO TO 40
      RETURN
      SUBROUTINE TRANS
      INX(L,1)=NSTRT
      INX(L,2)=NSTRT+8
      INX(L,3)=NSTRT+NL+9
      DO 4 K=4,8
    4 INX(L,K)=INX(L,K-1)+NL
      L=L+1
      INDS(L)=NL
      INDS(NSTRT)=IS
      DO 11 K=2,7
   11 DS(K+NSTRT-1)=TMP(K)
      INDS(NSTRT+7)=KS
      DS(NSTRT+NL+8)=PQ(NL+1)
      DO 21 K=1,NL
      DS(K+NSTRT+7)=PQ(K)
      DS(K+NSTRT+NL+8)=AQ(K)
      DS(K+NSTRT+2*NL+8)=AQT(K)
      DS(K+NSTRT+3*NL+8)=DG(K)
      INDS(K+NSTRT+4*NL+8)=ID(K)
      DS(K+NSTRT+5*NL+8)=W(K)
      DO 21 J=1,NL
   21 DS(J+NL*K+NSTRT+5*NL+8)=FR(J+NL*K-NL)
      NSTRT=NSTRT+(NL+3)**2
      RETURN
      END
```

Function ERFGS collects empirical ground state excitation rates for atom or ion types k. These rates are placed in function routines and depend on the electron temperature and level index n. The functions are to return a negative value (e.g., -1.0) for levels whose rates are not calculated. The Collector replaces the dummy names EDUMYi by the functions which are associated with the replacements for LDUMYi.

Function ERFGRY is used to calculate excitation coefficients based on the classical Gryzinski cross sections (Appendix D). For nonallowed transitions, function EREG based on Gryzinski exchange cross sections is used.

```
-TFOR,SI  SUB2
      FUNCTION ERFGS(TE,N,K)
      GO TO (1,2,3,4,5,6),K
C     DUMMY FUNCTION REFS ARE TO BE CONVERTED BY COLLECTOR
 1    ERFGS=EDUMY1(N,TE)
      RETURN
 2    ERFGS=EDUMY2(N,TE)
      RETURN
 3    ERFGS=EDUMY3(N,TE)
      RETURN
 4    ERFGS=EDUMY4(N,TE)
      RETURN
 5    ERFGS=EDUMY5(N,TE)
      RETURN
 6    ERFGS=EDUMY6(N,TE)
      RETURN
      END
```

```
-TFOR,SI  SUB3
      FUNCTION ERFGRY(Y,A)
      REAL Y,A
      IND=0
      IF(Y.LE.1.0E-3) GO TO 100
      IF(Y.GE.30.0)   GO TO 300
      IF(Y.GE.0.1.AND.Y.LE.10.0) GO TO 200
      IF(Y.GT.10.0)   GO TO 50
      X=ALOG10(Y)
      IF(X.LE.-0.5-0.25*A.AND.X.LE.-2.0) GO TO 100
      IF(X.GE. 0.5-0.25*A.AND.X.GE.-2.0) GO TO 200
      IND=1
      IF(A.LE.6.0)  GO TO 10
      IF(A.GE.10.0) GO TO 20
      WS=X+0.5+0.25*A
      GO TO 100
 10   WS=X+2.0
      GO TO 100
 20   WS=X+3.0
      GO TO 100
 50   IND=2
      WS=0.05*Y-0.5
      GO TO 200
100   F=SQRT(Y)*(1.0-A*ALOG(A*Y)/3.0)
      IF(F.LT.0.0) F=0.0
      IF(IND.EQ.0) GO TO 500
      FT=F
200   T=(A+30.0)/(10.0*A+25.0)
      F=A**0.25*(1.143*Y*Y+2.0*Y+0.127)
      F=Y**T/F
      IF(IND.EQ.0)  GO TO 500
      IF(IND.EQ.2)  GO TO 250
210   F=WS*(F-FT)+FT
      GO TO 500
250   FT=F
300   T=1.5-1.0/(A+1.0)
      F=1.5*SQRT(A)*Y**T
      F=1.0/F
      IF(IND.EQ.0)  GO TO 500
      GO TO 210
500   ERFGRY= 0.875*F*EXP(-Y)
      RETURN
      END
```

```
-TFOR,SI  SUB4
      FUNCTION EREG(Y,A,B)
      X=A/100.0
      IF(Y.GT.29.0) GO TO 10
      C=0.8665+(4.634-2.834*Y)*Y*EXP(-3.0*Y)+1.288*Y**3*EXP(-Y)
      G=5.97*EXP(-Y)-10.65*EXP(-2.0*Y)+7.85*EXP(-3.0*Y)
      GO TO 20
 10   C=0.8665
      G=0.0
 20   X=0.524/X-2.195+2.493*X-0.397*X*X
      W=2.0*C+2.0E-2*G*X
      X=B/10.0
      FX=X-0.1
      FX=6.5*FX+10.0*FX*FX
      IF(FX.GT.88.0) FX=88.0
      C =(252.26+3.713/X+0.0307/X**2)*(1.0-EXP(-FX))
      IF(X.GT.3.1) GO TO 30
      G =4.0+0.0541*X+0.247*X*X*EXP(-4.0*X*X)+184.0*X**4*EXP(-9.0*X*X)
      GO TO 40
 30   G =4.0+0.0541*X
 40   FX=2.013+0.2098*X-0.0802*X*X
      G = G*(Y-1.0)
      IF( G.GT.88.0)  G=88.0
      EREG=2.0E-3* C*Y**FX*EXP(-G )/A**W
      RETURN
      END
```

Subroutine NWTH calculates the line width per unit perturber number density for atom perturbers. The perturber type index is KP and the radiator index is KR, NL denotes the number of radiator levels calculated. If the perturber is an ion, the subroutine sets logical constant TEST to true and returns. The width of the transition between lower level i and upper level j is placed in the vector W residing in TEMPS common. The storage is arranged in terms of a half-matrix (null below the diagonal). Subroutine EWTH performs a similar calculation for electron perturbers. The basic broadening theory is outlined in Appendix A. The results of these subroutines along with natural, power and rate broadening are combined in LINWID. Power and rate quenching frequencies are obtained from INCR common. The total homogeneous broadening W and Doppler broadening WD reside in TEMPS.

Subroutine CRATE calculates the collisional rate coefficient matrix K_{ij} for type KR. The matrix resides in TEMPS as the symbol RM. It is assumed that the equivalent number of ground state electrons (ANG) is contained in empirical excitation rates. Ionization rates are the matrix elements with j = NPL.

```
-TFOR,SI  SUB5
      SUBROUTINE NWTH(T,KR,KP,NL,TEST,
      INCLUDE SPEC1,LIST
      PARAMETER NHSQ=(NLM+NSQ)/2
      DIMENSION W(NHSQ)
      EQUIVALENCE (IS,TMP(1)),(L,TMP(2)),(AMR,TMP(3)),(AMP,TMP(4)),
     C    (ALPH,TMP(5)),(RM,TMP(6)),(K,TMP(7)),(C6,TMP(8)),(C6P,TMP(9)),
     C    (C3,TMP(10)),(C3P,TMP(11)),(CU,TMP(12)),(CUP,TMP(13)),
     C    (W(1),TMP(21))
      LOGICAL TEST
      F(X)=X*(X+1.0)/(7.5*X*(X+1.0)-5.625)
      TEST=.FALSE.
      K=INX(KR,1)
      IZR=INDS(K)
      J=INX(KP,1)
      L=INDS(J)
      IF(L.NE.1) GO TO 70
      IS=0
      IF(KR.EQ.1) GO TO 6
      L=KR-1
      DO 4 I=1,L
 4    IS=IS+4*INDS(I+1)
 6    L=INDS(KR+1)
      AMR=DS(K+1)
      IF(KR.EQ.KP) GO TO 40
      AMP=DS(J+1)
      ALPH=DS(J+4)
      RM=1.0/AMR+1.0/AMP
      RM=(T*RM)**0.3
      LS=INX(KR,4)-1
      CU=(IZR-1)
      DO 30 I=1,NL
      K=((I-1)*(2*L-I))/2
      C6=BP(I+IS+L)
      IF(IZR.EQ.1) GO TO 16
      C3=DS(I+LS)
      C3=F(C3)
      C6=C6*CU*SQRT(C3)
 16   DO 30 J=I,NL
      IF(J.EQ.I) GO TO 20
      C6P=BP(J+IS+L)
      IF(IZR.EQ.1) GO TO 18
      C3=DS(J+LS)
      C3=F(C3)
      C6P=C6P*CU*SQRT(C3)
 18   C6P=ALPH*ABS(C6-C6P)
      W(J+K)=1.9E-3*C6P**0.4*RM
      GO TO 30
 20   W(J+K)=0.0
 30   CONTINUE
      GO TO 100
 40   RM=SQRT(2.0*T/AMR)
      DO 60 I=1,NL
      K=((I-1)*(2*L-I))/2
      C3=BP(I+IS)
      CU=BP(I+IS+L)
      DO 60 J=I,NL
      IF(J.EQ.I) GO TO 50
      C3P=C3+BP(J+IS)
```

```
      CUP=CU+BP(J+IS+L)
      C3P=0.0969*C3P
      CUP=2.04E-4*CUP*RM
      W(J+K)=AMAX1(C3P,CUP)
      GO TO 60
   50 W(J+K)=0.0
   60 CONTINUE
      GO TO 100
   70 TEST=.TRUE.
  100 RETURN
      END
```

```
TFOR,SI   SUB6
      SUBROUTINE EWTH(TE,KR,NL)
      INCLUDE SPEC1,LIST
      PARAMETER NHSQ=(NLM+NSQ)/2
      DIMENSION W(NHSQ)
      EQUIVALENCE (IS,TMP(1)),(L,TMP(2)),(L1,TMP(3)),(L2,TMP(4)),(L3,
     C     TMP(5)),(Z,TMP(6)),(CS,TMP(7)),(RS,TMP(8)),(KS,TMP(9)),(S,
     C     TMP(10)),(TD,TMP(11)),(TR,TMP(12)),(K,TMP(13)),(C1,TMP(14)),
     C     (C2,TMP(15)),(C3,TMP(16)),(C4,TMP(17)),(C1P,TMP(18)),(C3P,
     C     TMP(19)),(C4P,TMP(20)),(W(1),TMP(21))
      DATA A1,A2,A3,A4/0.029,12.2,5.625,7.5/
      IS=0
      IF(KR.EQ.1) GO TO 6
      L=KR-1
      DO 4 I=1,L
    4 IS=IS+4*INDS(I+1)
    6 L=INDS(KR+1)
      IS=IS+L
      L1=INX(KR,1)
      L2=INX(KR,4)-1
      L3=INX(KR,6)-1
      Z=INDS(L1)-1
      Z=Z*Z
      CS=DS(L1+5)
      RS=DS(L1+6)
      KS=INDS(L1+7)
      S=RS+0.5
      TD=TE**S
      TR=SQRT(TE)
      S=S/(1.0+S)
      RS=TE**RS
      DO 50 I=1,NL
      K=((I-1)*(2*L-I))/2
      C3=DS(I+L2)
      C3=C3*(C3+1.0)
      C3=C3/(C3*A4-A3)
      C3=SQRT(C3)*BP(I+IS)
      C4=BP(I+IS+L)
      IF(INDS(I+L3).GT.0.OR.I.EQ.1) GO TO 10
      C1=CS*DS(I+L1+7)**KS
      C2=A2*(A1*Z/C1)**S/TD
      C1=RS*C1*AMAX1(1.0,C2)
      GO TO 12
   10 C1=0.0
   12 CONTINUE
      DO 50 J=I,NL
      IF(J.EQ.I) GO TO 40
      C3P=DS(J+L2)
      C3P=C3P*(C3P+1.0)
      C3P=C3P/(C3P*A4-A3)
      C3P=SQRT(C3P)*BP(J+IS)
      C3P=ABS(C3-C3P)
      C4P=ABS(C4-BP(J+IS+L))
      IF(INDS(J+L3).GT.0.OR.J.EQ.1) GO TO 20
      C1P=CS*DS(J+L1+7)**KS
      C2=A2*(A1*Z/C1P)**S/TD
      C1P=RS*C1P*AMAX1(1.0,C2)
      GO TO 22
   20 C1P=0.0
```

```
 22     CONTINUE
        C1P=C1+C1P
        C2=CBRT(TR*C4P*C4P)
        C4P=11.14*(Z*C4P)**0.4/TR
        C4P=0.0362*AMAX1(C4P,C2)
        C4P=AMAX1(C4P,C1P)
        C2=8.18*CBRT(Z*C3P*C3P)/TR
        C3P=0.0969*AMAX1(C3P,C2)
        W(J+K)=AMAX1(C3P,C4P)
        GO TO 50
 40     W(J+K)=0.0
 50     CONTINUE
        RETURN
        END
```

```
-TFOR,SI  SUB7
      SUBROUTINE LINWID(TE,T,ANE,AN,KR,NL)
      INCLUDE SPEC1,LIST
      INCLUDE SPEC4,LIST
      LOGICAL TEST
      IS=0
      IR=0
      IF(KR.EQ.1) GO TO 6
      L=KR-1
      DO 4 I=1,L
      IS=IS+4*INDS(I+1)
  4   IR=IR+INDS(I+1)
  6   L=INDS(KR+1)
      IS=IS+3*L
      CALL EWTH(TE,KR,NL)
      DO 20 I=1,NL
      K=IFN(I,L)
      DO 20 J=I,NL
 20   W(J+K)=ANE*WD(J+K)
      DO 30 K=1,NSP
      CALL NWTH(T,KR,K,NL,TEST)
      IF(TEST) GO TO 30
      DO 25 I=1,NL
      N=IFN(I,L)
      DO 25 J=I,NL
      W(J+N)=W(J+N)+AN(K)*WD(J+N)
 25   CONTINUE
 30   CONTINUE
      N=INX(KR,2)-1
      A=INDS(N-7)**2
      A=1.415E3*SQRT(T/DS(N-6))*A
      DO 50 I=1,NL
      K=IFN(I,L)
      B=BP(I+IS)+R(I+IR)
      C=1.0/DS(I+N)**2
      DO 50 J=I,NL
      W(J+K)=W(J+K)+B+BP(J+IS)+R(J+IR)
 50   WD(J+K)=A*(C-1.0/DS(J+N)**2)
      RETURN
      END
```

```
-TFOR,SI  SUB8
      SUBROUTINE CRATE(TE,KR,NL)
      INCLUDE SPEC1,LIST
      PARAMETER N2=NSQ+NLM+1
      DIMENSION RM(NLM,NPL),STP(15)
      EQUIVALENCE (RM(1,1),TMP(1)),(STP(1),TMP(N2)),(L,STP(1)),(L1,
     C    STP(2)),(L2,STP(3)),(L3,STP(4)),(L4,STP(5)),(ANG,STP(6)),
     C    (U,STP(7)),(V,STP(8)),(VP,STP(9)),(GL,STP(10)),(GU,STP(11)),
     C    (A,STP(12)),(B,STP(13)),(Y,STP(14)),(R,STP(15))
      DATA PC/15.79/
      L=INDS(KR+1)
      L1=INX(KR,2)-1
      L2=INX(KR,5)-1
      L3=INX(KR,6)-1
      L4=INX(KR,8)-L-1
      ANG=INDS(L3+1)
      Z=INDS(L1-7)**2
      DO 60 I=1,NL
      U=  Z/DS(I+L1)**2
      GL=DS(I+L2)
      DO 50 J=I,NL
      IF(J.EQ.I) GO TO 40
      V=VP
      VP=  Z/DS(J+L1+1)**2
      GU=DS(J+L2)
      A=U-V
      Y=A*PC/TE
      R=GU*EXP(-Y)/GL
      IF(I.EQ.1.AND.IABS(INDS(J+L3)).EQ.2) GO TO 30
  6   B=DS(I+L*J+L4)
      IF(B.LE.0.0) GO TO 20
      B=A*SQRT(A)
      A=U/A
      RM(I,J)=ERFGRY(Y,A)/B
      A=U-VP
      Y=A*PC/TE
      B=A*SQRT(A)
      A=U/A
      RM(I,J)=RM(I,J)-ERFGRY(Y,A)/B
 10   IF(I.EQ.1) RM(I,J)=ANG*RM(I,J)
 12   RM(J,I)=RM(I,J)/R
      GO TO 50
 20   B=(U-VP)/A
      A=U/A
      RM(I,J)=(A/U)*SQRT(A/U)*EREG(Y,A,B)
      GO TO 10
 30   B=ERFGS(TE,J,KR)
      IF(B.LT.0.0) GO TO 6
      RM(I,J)=B
      GO TO 12
 40   RM(I,J)=0.0
      VP=  Z/DS(J+L1+1)**2
 50   CONTINUE
      V=  Z/DS(NL+L1+1)**2
      A=U-V
      Y=A*PC/TE
      B=A*SQRT(A)
      A=U/A
      RM(I,NPL)=ERFGRY(Y,A)/B
```

```
      IF(I.EQ.1) GO TO 54
      GO TO 60
54    VP=ERFGS(TE,L+1,KR)
      IF(VP.GT.0.0) GO TO 56
      RM(I,NPL)=ANG*RM(I,NPL)
      GO TO 60
56    Y=U*PC/TE
      B=U*SQRT(U)
      RM(I,NPL)=ANG*(RM(I,NPL)-ERFGRY(Y,1.0)/B)+VP
60    CONTINUE
      RETURN
      END
```

Functions TVEF and AXEF give transverse or side and axial or end escape factors for inverted transitions, respectively. Least square curve fits to numerical calculations are used (see Appendix B). The arguments are C - axial gain logarithm, R - cavity aspect ratio and P - ratio of Lorentz to Doppler line widths. To avoid possible overflow problems, values of gain are limited to certain bounds in the functions. The function CSA produces the average side solid angle fraction for $L \leq 0$, the average end solid angle fraction for $L = 1$, $\langle r/L \rangle_{side}$ for $L = 2$ and $\langle r/L \rangle_{end}$ for $L \geq 3$. Function LPI gives $\langle \bar{p} \rangle$ and WLP gives the line profile constant k for the input of the line width ratio.

The subroutine ESCAPE calculates the emission factors E_{ji} for the transition $j \rightarrow i$. (Cavity effects are not involved.) The factors use the symbol PHI and reside in TEMPS. Required input is the excited level populations ANX for type KR. The optical depth or gain logarithm is denoted by G. An expression needed for the Jacobian matrix, the rate of change of the logarithm of the emission factor with respect to the logarithm of G, is placed as the symbol W in TEMPS. If the magnitude of the excess emission (Appendix B) is less than 10^{-3}, this expression equals the excess emission and is not needed; W then contains a multiplicative factor that describes the power broadening. The broadening is simply this power factor times the Einstein A-value (noncavity effects only). For the larger magnitude excess emission, the power factor equals the excess emission factor times the upper level population density and divided by the inversion density. (For absorbing transitions, both the excess emission and inversion density are negative giving a positive power factor.) Note that the excess emission and inversion density approach zero at the same rate so that the power factor as contained in W can be determined from an expression that is a limit ratio. Arrays GP and LU in TEMPS are used to store data on inverted levels in the following manner: GP(1,i) are the positive gain logarithms in order of descending values (up to i=4), GP(2,i) are the corresponding line width ratios, GP(3,i) the transition center profile values, LU(1,i) the lower level indices and LU(2,i) the upper level indices. The data corresponding to a cavity transition are used in the cavity rate equations. If a cavity transition is not one of the four maximum of gains greater than unity for a given atom or ion type, then data are not made available for the cavity and coupling of the cavity intensity to the laser medium is neglected. (The diagonal elements of the emission factor are used to hold the total gain of a level for transitions to all lower levels. An upper bound on any neglected cavity coupling thus may be established.)

```
FOR,SI  SUB9
      FUNCTION TVEF(C,R,P)
      DIMENSION A(7,3),B(3)
      DATA A/0.96213,-1.3732,4.3505,-4.7683,2.4813,-0.59608,0.052728,
     C     3.1943,-7.6577,12.775,-12.875,6.6044,-1.5975,0.14369,3.8401,
     C     3.7817,-15.395,17.138,-8.7419,2.0886,-0.18833/
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      REAL LPI
      X=AMIN1(3.912,ALOG(R))
      DO 10 I=1,3
      B(I)=0.0
      DO 10 J=7,1,-1
      B(I)=X*B(I)
10    B(I)=B(I)+A(J,I)
      D=CFST(3)
      ENTRY TV2(C,R,P)
      CE=C
      IF(C.GT.50.0) CE=50.0+30.0*(C-50.0)/(C-20.0)
      W=1.0/(1.0+12.5*(CE/R)**4)
      TVEF=(1.0-W)*WLP(P)*TVQ(CE)+W*CE*D*LPI(P)
      RETURN
      FUNCTION TVQ(DUM)
      X=0.1*DUM
      X=-B(1)+B(2)*X+B(3)*X*X
      IF(X.GT.10.0) X=10.0
      TVQ=EXP(X)
      RETURN
      END
```

```
-TFOR,SI  SUB10
      FUNCTION AXEF(C,R,P)
      DIMENSION A(6,3),B(2),CA(3)
      DATA A/-8.47794E-3,1.20125E-1,5.06073E-3,-6.86903E-3,8.67579E-4
     C    -3.25246E-5,-1.59631E-1,2.80494E-1,-1.21696E-1,2.19665E-2,
     C    -1.79975E-3,5.5E-5,-1.42061E-2,1.06835E-1,-4.34725E-2,
     C    6.91377E-3,-4.9476E-4,1.33524E-5/B/0.0337633,-0.0080375/
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      REAL LPI
      CE=C
      IF(C.GT.50.0) CE=50.0+30.0*(C-50.0)/(C-20.0)
      DO 10 I=1,3
      CA(I)=0.0
      DO 10 J=6,1,-1
      CA(I)=CE*CA(I)
 10   CA(I)=CA(I)+A(J,I)
      IF(CE.LE.4.0) GO TO 20
      G=B(1)+B(2)*CE
      IF(CE.GE.6.0) GO TO 19
      W=2.0**CE/16.0
      CA(2)=CA(2)+(1.0-W)*(CA(2)-G)/3.0
      GO TO 20
 19   CA(2)=G
 20   G=ALOG(R)
      IF(CE.LE.1.0) GO TO 40
      W=(CA(1)+CA(2)*G+CA(3)*G*G)/(CE*CE)
 30   G=W*EXP(CE)*SQRT(CE)/(R*R)
      GO TO 50
 40   W=0.110674+0.019389*G+0.055589*G*G
      GC TO 30
 50   W=1.0/(1.0+50.0*CE)
      AXEF=(1.0-W)*G*WLP(P)+W*CE*CFST(4)*LPI(P)
      RETURN
      END
```

```
-TFOR,SI  SUB11
      FUNCTION CSA(R,L)
      DIMENSION A(4,2)
      DATA A/0.4244,-0.172,0.09,-0.025,0.66,-1.09,2.3,-1.6/
      IF(L.GE.3) GO TO 30
      M=MAX0(1,L)
      B=0.0
      DO 10 I=4,1,-1
 10   B=(B+A(I,M))/R
      IF(L.GT.0) GO TO 20
      CSA=1.0-B
      RETURN
 20   CSA=B
      RETURN
 30   CSA=(0.9428+ALOG(R))/(8.0*R*R)+0.0132/R**4
      RETURN
      END
-FOR,IS SUB11B
      FUNCTION LPI(X)
      REAL LPI
      LPI=(0.707+1.25*X)/(1.0+2.5*X)
      RETURN
      END
-FOR,IN SUB11C
      FUNCTION WLP(X)
      WLP=(1.0+0.7304*X+0.5811*X*X)/(1.0+1.2946*X+1.0299*X*X)
      RETURN
      END
```

```
-TFOR,SI   SUB12
      SUBROUTINE ESCAPE(ANX,KR,NL,AN)
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      INCLUDE SPEC4,LIST
      DIMENSION ANX(NLM)
      REAL LPI
      F(X)=(1.0+0.93644*X+0.3299*X*X)/(1.0+2.06482*X+1.65979*X*X+
     C     0.58473*X*X*X)
      DEP(X)=(SIGN(0.25,X)-0.75)
      L=INDS(KR+1)
      RL=L
      RL=RL+0.5
      L1=INX(KR,5)-1
      L2=INX(KR,8)-1-L
      KD=(L*(L+1))/2
      DO 10 K=1,4
      GP(1,K)=0.0
      GP(2,K)=0.0
      GP(3,K)=0.0
      LU(1,K)=0
 10   LU(2,K)=0
      DO 100 K=KD,1,-1
      P=2*K
      P=RL*RL-P
      I=RL+1.0-0.5*(SQRT(P)+SQRT(P+2.0))
      J=K-IFN(I,L)
      IF(J.GT.NL.OR.J.EQ.I) GO TO 90
      P=DS(J+L1)/DS(I+L1)
      RS=1.49736E5*ABS(DS(J+L*I+L2))*CL/WD(K)
      G=RS*(ANX(J)-P*ANX(I))
      P=W(K)/WD(K)
      PF=F(P)
      IF(G.GE.0.0) GO TO 50
      PA=-G/(2.0*CAR)
      PF=PF*LPI(P)*(CFST(3)+CFST(4))
      IF((-G*PF).LT.1.0E-3) GO TO 20
      G=-G*PF
      PHI(K)=0.84*SQRT(P/PA)
      PA=0.9/(PA*SQRT(ALOG(AMAX1(PA,2.72))))
      IF(PA.GT.PHI(K)) PHI(K)=-PA
      RS=ABS(PHI(K))
      PA=5.0*RS
C     ARRAY W IS USED IN JACOBIAN MATRIX
      W(K)=(1.0+PA-PA*PA)*EXP(-PA)*RS*DEP(PHI(K))-G*EXP(-G)
      PA=RS*(1.0+PA)*EXP(-PA)+EXP(-G)
      W(K)=W(K)/PA
      PHI(K)=PA
      GO TO 30
 20   PHI(K)=1.0+G*PF
      W(K)=RS*PF*ANX(J)
      GO TO 100
 30   IF(ABS(PHI(K)-1.0).LT.1.0E-3) PHI(K)=1.0+SIGN(1.0E-3,PHI(K)-1.0)
      GO TO 100
 50   G=G*PF
      RS=RS*PF
      DO 60 J1=1,4
      IF(G.LE.GP(1,J1)) GO TO 60
      DO 54 J2=4,1,-1
```

```
      IF(J2.EQ.J1) GO TO 56
      GP(1,J2)=GP(1,J2-1)
      GP(2,J2)=GP(2,J2-1)
      GP(3,J2)=GP(3,J2-1)
      LU(1,J2)=LU(1,J2-1)
      LU(2,J2)=LU(2,J2-1)
 54   CONTINUE
 56   GP(1,J1)=G
      GP(2,J1)=P
      GP(3,J1)=0.5642*PF/WD(K)
      LU(1,J1)=I
      LU(2;J1)=J
      GO TO 62
 60   CONTINUE
 62   J2=1
      DO 70 J1=1,NCAV
      IF(ICAV(1,J1).NE.KR) GO TO 70
      IF(ICAV(2,J1).NE.I)  GO TO 70
      IF(ICAV(3,J1).NE.J)  GO TO 70
      J2=0
 70   CONTINUE
      PA=CFST(3)
      IF(J2.GT.0) PA=PA+CFST(4)
      PF=   PA*LPI(P)
      IF((G*PF).LT.1.0E-3) GO TO 20
      PHI(K)=TV2(G,CAR,P)
      IF(J2.LE.0) GO TO 80
      PHI(K)=PHI(K)+AXEF(G,CAR,P)
      RS=TV2(1.01*G,CAR,P)+AXEF(1.01*G,CAR,P)
      GO TO 85
 80   RS=TV2(1.01*G,CAR,P)
 85   PA=PHI(K)
      PHI(K)=1.0+PA
      W(K)=100.0*PA*ALOG(RS/PA)/PHI(K)
      J1=J+IFN(J,L)
      RS=AMIN1(G,88.0)
      PHI(J1)=PHI(J1)+EXP(RS)
      GO TO 30
 90   PHI(K)=0.0
100   CONTINUE
      RETURN
      END
```

The subroutine EQUIL is used to establish an equilibrium initial excited level distribution. Let

$$\sigma^{s,z} = \sum_n \beta_n^{s,z} \quad ,$$

then the total population of type s,z is $N_e \sigma^{s,z} N_1^{s,z+1}$.

If

$$P^{s,z_s} = 1 \quad ,$$

$$P^{s,z} = \prod_{y=z+1}^{z_s} \beta_1^{s,y} \quad ,$$

then

$$N_1^{s,z+1} = (N_e)^{z_s-z} P^{s,z} N_1^s \quad .$$

Specie conservation is given by $N_s = N_1^s Q_s$ where

$$Q_s = 1 + \sum_z (N_e)^{z_s-z+1} \sigma^{s,z} P^{s,z} \quad .$$

Charge conservation is

$$\sum_s z_s N_s = N_e \left(1 + \sum_s N_s \, \partial \ell n \, Q_s / \partial N_e \right) .$$

Function Q_s is a polynomial in N_e. An initial estimate of N_e is obtained by computing the next stage ionization fraction of each type ordered with increasing β_1. A fraction ≥ 0.86 is considered equivalent to unity; when the fraction is less, the calculations are halted. The N_e estimate is used to calculate $\partial \ell n \, Q_s / \partial \ell n \, N_e$ and hence a new N_e, the new and old estimate are averaged and used to calculate a new N_e, etc., until convergence is obtained. Once N_e is known, populations are readily calculated. The symbol ANG is used for ground state populations in the subroutine rather than AN. STOR, AN and ISZ are scratch variables used for ordering; other vector quantities are self explanatory or conventional.

The subroutine QSTAT is used to establish a quasistationary initial distribution. The form of this distribution is $\overline{N}_n^{s,z} \neq 0$ if and only if $z = z_i(s)$ and $z = z_i + 1$, $n = 1$. That is, only one ionization stage of each specie is considered in detail. If

$$N_1^{s,z_i+1} \equiv \alpha_s N_s \quad ,$$

then

$$(1 - \alpha_s)N_s = \sum_n N_n^{s,z_i}$$

and

$$N_e = \sum_s (z_i - 1 + \alpha_s)N_s \quad .$$

Inputs through namelist NAME14 are z_i as IZ and P to determine α_s. If P is negative, $0 < |P| \leq 1$, then α_s is set equal to $|P|$ for all s. If P is positive,

$$\alpha_s = \left(1 + P\, N_e\, \beta_1^{s,z_i}\right)^{-1} \quad .$$

P=1 corresponds to equilibrium and P<1 for a recombining plasma. Levels for $n \geq 2$ are assumed quasistationary. For positive P, N_e must be obtained by an iterative procedure. If $z_i > z_s$ or $z_i \leq 0$, then specie s is considered fully ionized into the z_s+1 ground level. Populations are determined by iteration using subroutine SRJM and starting with essentially null quasistationary levels (value of 10^{-8}). Iteration is halted when the change of ground level populations of z_i are sufficiently small.

EQUIL and QSTAT place ground state populations (AN or ANG and ANP) into MSTOR common for use by the program at integration start.

```
-TFOR,SI  SUB13
      SUBROUTINE EQUIL
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      INCLUDE SPEC7,LIST
      PARAMETER N13S0=1+2*(NSP+NCM)
      PARAMETER NIT=10,N13S1=NSP+14,N13S2=N13S1+NSP,N13S3=N13S2+NSP,
     C    N13S4=N13S3+NSP,N13S5=N13S4+NSP
      DIMENSION BETA(NSP),SIGMA(NSP),P(NSP),AN(NSP),ISZ(NSP),STOR(NSP)
     C    ANG(NSP),ANP(NCM)
      EQUIVALENCE (ANR(1),STRMTX(1)),(ANPH(1),STRMTX(1)),(ANG(1),
     C   STRMTX(1)),(ANP(1),STRMTX(N13S0))
      EQUIVALENCE (L,TMP(1)),(LN,TMP(2)),(LG,TMP(3)),(MC,TMP(4)),
     C    (A,TMP(5)),(B,TMP(6)),(C,TMP(7)),(D,TMP(8)),(E,TMP(9)),
     C    (RI,TMP(10)),(GNS,TMP(11)),(Q,TMP(12)),(S,TMP(13)),
     C    (BETA(1),TMP(14)),(SIGMA(1),TMP(N13S1)),(P(1),TMP(N13S2)),
     C    (AN(1),TMP(N13S3)),(ISZ(1),TMP(N13S4)),(STOR(1),
     C    TMP(N13S5)),(CGME,TMP(1)),(DGME,TMP(2)),(GLTE,TMP(3))
  4   FORMAT(1H0,2X30HSUM ON Z IN SUBR EQUIL - ERROR)
  5   FORMAT(1H0,2X16HSLOPE FUNCTION =1PE10.3,2X10HREL ERROR=1PE10.3,
     C        2XI2,1X13HSTEP IN EQUIL)
  6   FORMAT(1H0,2X9HFINAL NE=1PE10.3,2X6HERROR=1PE10.3)
      DATA ERROR/1.0E-3/
      B=2.0708E-7/(TE*SQRT(TE))
      C=15.79/TE
      DO 10 K=1,NSP
      LN=INX(K,2)-1
      LG=INX(K,5)-1
      L=INX(K,1)
      GNS=DS(L+2)
      L=INDS(L)
      RI=L
      L=INDS(K+1)
      SIGMA(K)=0.0
      DO 10 J=1,L
      A=RI/DS(J+LN)
      A=B*DS(J+LG)*EXP(C*A*A)/GNS
      IF(J.NE.1) GO TO 8
      BETA(K)=A
  8   SIGMA(K)=SIGMA(K)+A
 10   CONTINUE
      IF(NSP.EQ.1) GO TO 17
      LN=1
      MC=1
      DO 16 K=2,NSP
      L=INX(K,1)
      L=INDS(L)
      IF(L.EQ.1) GO TO 12
      GO TO 15
 12   RI=1.0
      DO 14 J=MC,LN,-1
      P(J)=RI
 14   RI=RI*BETA(J)
      LN=K
 15   MC=MC+1
      IF(MC.EQ.NSP) GO TO 12
 16   CONTINUE
      GO TO 18
 17   P(1)=1.0
```

```
18    ISZ(1)=1
      AN(1)=ANC(1)
      STOR(1)=BETA(1)
      IF(NSP.EQ.1) GO TO 26
      MC=1
      DO 24 K=2,NSP
      L=INX(K,1)
      L=INDS(L)
      IF(L.EQ.1) MC=MC+1
      RI=ANC(MC)
      AN(K)=RI
      STOR(K)=BETA(K)
      ISZ(K)=K
      DO 20 J=K,2,-1
      IF(BETA(K).GT.STOR(J-1)) GO TO 24
      ISZ(J)=ISZ(J-1)
      ISZ(J-1)=K
      AN(J)=AN(J-1)
      AN(J-1)=RI
      STOR(J)=STOR(J-1)
      STOR(J-1)=BETA(K)
20    CONTINUE
24    CONTINUE
26    GNS=0.0
      DO 40 J=1,NSP
      RI=AN(J)*STOR(J)
      B=GNS*STOR(J)+1.0
      D=4.0*RI/(B*B)+1.0
      IF(B.LT.1.09.AND.D.LT.1.30) GO TO 36
      A=  B*(SQRT(D)-1.0)/(2.0*RI)
      ANE=A*AN(J)+GNS
      GO TO 42
36    GNS=GNS+AN(J)
40    CONTINUE
42    NSTP=1
44    C=0.0
      D=1.0
      LN=1
      DO 60 J=1,NC
      GNS=ANC(J)
      IF(J.EQ.NC) GO TO 48
      LG=IC(J+1)-1
      GO TO 50
48    LG=NSP
50    L=INX(LG,1)
      L=INDS(L)
      RI=L
      C=C+RI*GNS
      MC=0
      RI=0.0
      Q=1.0
      DO 54 K=LN,LG,1
      MC=MC+1
      B=SIGMA(K)*P(K)*ANE**(L-MC)
      Q=Q+ANE*B
      A=L-MC+1
54    RI=RI+A*B
      D=D+RI*GNS/Q
      IF(LG.EQ.NSP) GO TO 62
```

```
      LN=LG+1
      IF(L.NE.MC) WRITE(6,4)
 60   CONTINUE
 62   GNS=C/D
      E=(GNS-ANE)/GNS
      IF(NSTP.EQ.1) GO TO 64
      IF(ABS(E).LT.1.0E-8) GO TO 64
      S=D+2.0*(D-S)/E
      IF(NSTP.GE.NIT.OR.S.LE.0.0) WRITE(6,5) S,E,NSTP
 64   S=D
      ANE=0.5*(GNS+ANE)
      IF(ABS(E).LT.ERROR.OR.NSTP.GE.NIT) GO TO 70
      NSTP=NSTP+1
      GO TO 44
 70   LN=1
      IF(NSTP.LT.NIT) GO TO 71
      IF(ABS(E).LT.0.01) GO TO 71
      NSTP=NSTP+1
      IF(NSTP.LT.3*NIT) GO TO 44
      WRITE(6,6) ANE,E
 71   CONTINUE
      DO 80 J=1,NC
      GNS=ANC(J)
      IF(J.EQ.NC) GO TO 72
      LG=IC(J+1)-1
      GO TO 74
 72   LG=NSP
 74   L=INX(LG,1)
      L=INDS(L)
      MC=0
      Q=1.0
      DO 76 K=LN,LG,1
      MC=MC+1
 76   Q=Q+SIGMA(K)*P(K)*ANE**(L-MC+1)
      MC=0
      DO 78 K=LN,LG,1
      MC=MC+1
 78   BETA(K)=GNS*P(K)*ANE**(L-MC)/Q
      IF(LG.EQ.NSP) GO TO 82
      LN=LG+1
 80   CONTINUE
 82   MC=0
      DO 90 K=1,NSP
      L=INDS(K+1)
      GNS=BETA(K)
      DO 90 J=1,L
      MC=MC+1
 90   ANB(MC)=GNS
      CALL CONSRV(GT,ANP,G1,G2,G3,G4,ANG,0)
      E=(ANE-GT)/GT
      ANE=GT
      WRITE(6,6) ANE,E
      CGME=0.0
      DGME=0.0
      CALL ELEN(GT,0.0,0.0,0.0,ANG,ANP)
      GLTE=GT
      RETURN
      END
```

```
TFOR,SI  SUB14
      SUBROUTINE QSTAT
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      INCLUDE SPEC7,LIST
      DIMENSION NBRS(NSP),ALPH(NCM),BETG(NCM),IS(NCM),KS(NCM),IZS(NCM),
     C    ANP(NCM),AN(NSP),R3(NLM),LC(4),D(6),R1(NLM,NPL),R2(NLM,NPL),
     C    R4(2,4),R5(4),R6(3,4),IZ(NCM)
      LOGICAL LGP,LERR
      PARAMETER N1=2*NSP+NCM+1,N2=N1+5*NCM+6,N3=N2+2*(NLM+NSQ),
     C    N14S1=NSP+1,N14S2=N14S1+NSP,N14S3=N1+NCM,N14S4=N14S3+NCM,
     C    N14S5=N14S4+NCM,N14S6=N14S5+NCM,N14S7=N14S6+NCM,N14S8=N2+NLM+
     C    NSQ,N14S9=N3+NLM,NIT=10
      EQUIVALENCE (CGME,TMP(1)),(DGME,TMP(2)),(GLTE,TMP(3)),(AN(1),
     C    STRMTX(1)),(NBRS(1),STRMTX(N14S1)),(ALPH(1),STRMTX(N14S2)),
     C    (BETG(1),STRMTX(N1)),(ANP(1),STRMTX(N14S3)),(IZS(1),
     C    STRMTX(N14S4)),(IS(1),STRMTX(N14S5)),(KS(1),STRMTX(N14S6)),
     C    (D(1),STRMTX(N14S7)),(R1(1,1),STRMTX(N2)),(R2(1,1),
     C    STRMTX(N14S8)),(R3(1),STRMTX(N3)),(R4(1,1),STRMTX(N14S9)),
     C    (ANR(1),STRMTX(1))
      NAMELIST/NAME14/ERROR,P,IZ
 1    FORMAT(1H0,3X18HQSTAT ERROR, TYPE I2)
 2    FORMAT(1H0,3X14HQSTAT NE DIFF=1PE10.3)
 3    FORMAT(1H0,3X14HQSTAT NG DIFF=1PE10.3)
 4    FORMAT(1H0,3X15HQSTAT GIVES NE=1PE10.3)
      READ(5,NAME14)
      LIO=0
      LGP=.FALSE.
      DO 10 K=1,ND
      BT(K)=0.0
10    ANB(K)=1.0E-8
      DO 11 K=1,NSP
      AN(K)=0.0
      NBRS(K)=NBAR(K)
11    NBAR(K)=2
      DO 12 K=1,4
12    ANPH(K)=0.0
      C=2.0708E-7/(TE*SQRT(TE))
      IF(P.LT.0.0) GO TO 50
      DN=0.0
      A=15.79/TE
      DO 20 J=1,NC
      ANP(J)=0.0
      IS(J)=0
      KS(J)=0
      K1=IC(J)
      IF(J.LT.NC) K2=IC(J+1)-1
      IF(J.EQ.NC) K2=NSP
      MK=K2-K1+1
      IF(IZ(J).GT.MK) GO TO 18
      IF(IZ(J).LE.0) GO TO 19
      DO 15 K=K1,K2
      MK=K-K1+1
      IF(MK.NE.IZ(J)) GO TO 15
      Z=MK
      MK=INX(K,1)+2
      GD=DS(MK)
      MK=INX(K,5)
      GD=DS(MK)/GD
```

```
      MK=INX(K,2)
      Z=Z/DS(MK)
      BETG(J)=C*GD*EXP(A*Z*Z)
      B=IZ(J)-1
      GO TO 20
15    CONTINUE
18    IZ(J)=-IZ(J)
19    BETG(J)=0.0
      B=MK-1
20    DN=DN+B*ANC(J)
      IT=0
      ANE=DN
24    Y=0.0
      DO 30 J=1,NC
30    Y=Y+ANC(J)/(1.0+P*ANE*BETG(J))
      A=DN+Y
      IT=IT+1
      B=(ANE-A)/A
      ANE=(ANE+A)/2.0
      IF(ABS(B).LT.ERROR.OR.IT.GT.NIT) GO TO 32
      GO TO 24
32    IF(IT.GT.NIT) GO TO 36
33    DO 34 J=1,NC
34    ALPH(J)=1.0/(1.0+P*ANE*BETG(J))
      GO TO 60
36    WRITE(6,2) B
      IF(ABS(B).LE.0.01) GO TO 33
      IF(IT.LT.2*NIT) GO TO 24
      IF(ABS(B).GT.0.05) GO TO 41
      GO TO 33
40    WRITE(6,1) LIO
      STOP
41    LIO=1
      GO TO 40
42    LIO=2
      GO TO 40
44    LIO=4
      GO TO 40
45    LIO=5
      GO TO 40
50    A=0.0
      IF(ABS(P+0.5).GE.0.5) GO TO 42
      DO 56 J=1,NC
      ANP(J)=0.0
      IS(J)=0
      KS(J)=0
      ALPH(J)=-P
      IF(J.LT.NC) K2=IC(J+1)-1
      IF(J.EQ.NC) K2=NSP
      IT=K2+1-IC(J)
      IF(IZ(J).GT.IT) IZ(J)=-IZ(J)
      IF(IZ(J).GT.0) GO TO 52
      B=IT
      GO TO 56
52    B=IZ(J)-1
      B=B-P
56    A=A+B*ANC(J)
      ANE=A
60    DO 61 J=1,NC
```

```
61    ALPH(J)=ALPH(J)*ANC(J)
      MK=0
      QM=0.0
      DO 64 J=1,NC
      K1=IC(J)
      IF(J.LT.NC) K2=IC(J+1)-1
      IF(J.EQ.NC) K2=NSP
      IZS(J)=K2-K1+1
      B=IZS(J)
      QM=QM+B*ANC(J)
      DO 63 K=K1,K2
      IT=K-K1+1
      IF(IT.NE.IZ(J)) GO TO 63
      KS(J)=K
      IT=MK+1
      IS(J)=IT
      AN(K)=ANC(J)-ALPH(J)
      BT(IT)=BETG(J)
      ANB(IT)=AN(K)/(ANE*BETG(J))
      IF(K.EQ.K2) GO TO 62
      AN(K+1)=ALPH(J)
      A=15.79/TE
      Z=IZ(J)+1
      IT=INX(K+1,2)
      Z=Z/DS(IT)
      IT=INX(K+1,5)
      GD=DS(IT)
      IT=INX(K+1,1)+2
      GD=GD/DS(IT)
      B=C*GD*EXP(A*Z*Z)
      IT=IS(J)+INDS(K+1)
      BT(IT)=B
      ANB(IT)=ALPH(J)/(ANE*B)
      GO TO 63
62    ANP(J)=ALPH(J)
63    MK=MK+INDS(K+1)
64    CONTINUE
      IT=0
70    DN=0.0
      A=0.0
      B=0.0
      C=0.0
      DO 80 J=1,NC
      IF(IZ(J).LE.0) GO TO 78
      ANHS=0.0
      K=KS(J)
      MK=INDS(K+1)+IS(J)
      BTPL=BT(MK)
      IF(IZS(J).NE.IZ(J)) GO TO 71
      ANHS=ALPH(J)
      BTPL=1.0/ANE
71    MK=IS(J)-1
      ZS=IZS(J)
      Z=IZ(J)
      GD=ANB(MK+1)
      ANEP=ANE
      CALL SRJM(R1,R2,R3,AI,AR,ARR,QRI,QRR,K,MK,AN,ANEP,QM,ANPH,ANHS,
     C    BTPL,0.0,0.0,0.0,ZS,Z,D(1),LC,R4,LGP,J,0,D(2),D(3),D(4),D(5),
     C    D(6),LERR,K1,R5,R6)
```

```
      IF(LERR) GO TO 44
      ZS=ZS+1.0-Z
      A=A+ZS*(ANE*AI-ALPH(J)*AR)
      B=B+ZS*ALPH(J)*ARR
      C=C+ANE*(ALPH(J)*GRR-ANE*GRI)
      MK=IS(J)
      ANHS=0.0
      K1=NSTAR(K)-2
      IF(K1.LT.1) GO TO 73
      DO 72 K2=1,K1
72    ANHS=ANHS+ANE*BT(MK+K2)*ANB(MK+K2)
73    ANHS=ANC(J)-ALPH(J)-ANHS
      IF(ANHS.LT.0.0) ANHS=0.0
      ANHS=ANHS/(ANE*BT(MK))
      ZS=(GD+ANHS)/2.0
      IF(ZS.EQ.0.0) GO TO 75
      Z =ABS(GD-ANHS)/ZS
      IF(Z.GT.DN) DN=Z
75    ANB(MK)=ZS
      AN(K)=ANE*ZS*BT(MK)
78    IF(J.LT.NC) GO TO 80
      IT=IT+1
      IF(DN.LT.ERROR.OR.IT.GT.NIT) GO TO 82
80    CONTINUE
      GO TO 70
82    IF(IT.GT.NIT) GO TO 84
      GO TO 86
84    WRITE(6,3) DN
      IF(DN.GT.0.05) GO TO 45
86    WRITE(6,4) ANE
      DO 90 K=1,NSP
90    NBAR(K)=NBRS(K)
      CALL ELEN(DN,0.0,C,A,AN,ANP)
      CGME=A
      DGME=A+B
      GLTE=DN
      RETURN
      END
```

Subroutine CONSRV calculates the electron density and densities of maximum ionization given the electron temperature and excited level normal populations. Densities of ground levels are also output. Symbol EPD is used for the electron density, EP is the equilibrium bound states factor ε, DI = $2a_o/\ell_D$ is proportional to the lowering of the ionization potential, DEL = $A/N_e - B$ (see conservation discussion in main text) is a measure of the correction term for finite ε in N_e, QM = $(1 + D)N_e$ is needed by subroutine SRJM and IBT is set ≤ 0 if array BT of Saha factors is to be calculated and >0 if not (e.g., during a second pass at a particular electron temperature).

```
-TFOR,SI  SUB15
      SUBROUTINE CONSRV(EPD,ANP,EP,DI,DEL,QM,AN,IBT)
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      PARAMETER N15S1=NCM+9,N15S2=N15S1+NCM,N15S3=N15S2+NCM,N15S4=
     C    N15S3+NCM
      DIMENSION ANP(NCM),SA(NCM),SB(NCM),SC(NCM),SD(NCM),ZC(NCM),AN(NSP)
      EQUIVALENCE (MS,TMP(1)),(MR,TMP(2)),(GP,TMP(3)),(ZC,TMP(N15S4)),
     C    (A,TMP(5)),(B,TMP(6)),(C.TMP(7)),(D,TMP(8)),(SA(1),TMP(9)),
     C    (SB(1),TMP(N15S1)),(SC(1),TMP(N15S2)),(SD(1),TMP(N15S3))
      DO 2 J=1,NCM
      SA(J)=0.0
      SB(J)=0.0
      SC(J)=0.0
  2   SD(J)=0.0
      MS=0
      MR=0
      IF(IBT.GT.0) GO TO 3
      A=15.79/TE
      C=2.0708E-7/(TE*SQRT(TE))
  3   DO 10 K=1,NSP
      L1=INX(K,2)-1
      L2=INX(K,5)-1
      L=INX(K,1)
      GP=DS(L+2)
      L=INDS(L)
      IF(L.EQ.1) MS=MS+1
      ZC(1)=L
      L=INDS(K+1)
      DO 10 J=1,L
      MR=MR+1
      IF(J.GE.NSTAR(K)) GO TO 10
      IF(IBT.GT.0) GO TO 4
      B=ZC(1)/DS(J+L1)
      B=C*DS(J+L2)*EXP(A*B*B)/GP
      BT(MR)=B
      GO TO 6
  4   B=BT(MR)
  6   B=B*ANB(MR)
      D=ZC(1)-1.0
      SA(MS)=SA(MS)+B
      SC(MS)=SC(MS)+B*D
      IF(J.NE.1) GO TO 10
      AN(K)=B
      D=D*SQRT(D)
      SB(MS)=SB(MS)+B*D
      SD(MS)=SD(MS)+B*D*(ZC(1)-2.0)
 10   CONTINUE
      C=0.0
      D=1.0
      DO 20 J=1,NC
      IF(J.EQ.NC) GO TO 12
      K=IC(J+1)-1
      GO TO 14
 12   K=NSP
 14   L=INX(K,1)
      L=INDS(L)
      ZC(J)=L
      C=C+ZC(J)*ANC(J)
```

```
      D=D+ZC(J)*SA(J)-SC(J)
20    CONTINUE
      EPD=C/D
      EP=TE+TE*TE/T
      EP=EPD/EP**3
      EP=SQRT(EP)
      EP=1.3E-2*SQRT(EP)
      A=0.0
      B=0.0
      DO 30 J=1,NC
      GP=ZC(J)*SQRT(ZC(J))
      ANP(J)=1.0+EP*GP
      A=A+GP*ANC(J)/ANP(J)
      B=B+SD(J)-ZC(J)*SB(J)+GP*(SA(J)+EP*SB(J))/ANP(J)
30    CONTINUE
      EPD=(C-EP*A)/(D-EP*B)
      DEL=A/EPD-B
      QM=D*EPD
      DI=4.85E-4*SQRT(EPD/TE+EPD/T)
      DO 40 J=1,NC
40    ANP(J)=(ANC(J)-EPD*(SA(J)+EP*SB(J)))/ANP(J)
      DO 50 K=1,NSP
50    AN(K)=EPD*AN(K)
      RETURN
      END
```

Subroutine SRJM is used to compute rates and the Jacobian for a particular atom or ion. The argument list is as follows.

RL	rate matrix for the reduced set of equations. The vector G_n used to couple to the next ionization stage is the set of elements in RL with second index NPL.
JCB	Jacobian matrix for the reduced equations. The element set with second index NPL is the vector $\left((z_s + 1 - z)N_e\right)^{-1} (\partial R_e/\partial \overline{N}_n)$, used for cross terms in the overall Jacobian.
F	the vector $\partial R_n/\partial N_e$, plus $(1 + D)\ \partial R_n/\partial N_1^+$ if $z = z_s$, also used for cross terms
AI	ionization coefficient
AR	recombination coefficient (collisional and radiative)
ARR	radiative recombination coefficient
QCRI	coefficient of energy absorption divided by kT_e
QCRR	recombination heating coefficient divided by kT_e
KS	atom or ion type index (input)
MS	location of ground level in ANB, BT vectors
AN	ground level density vector (input)
ANEP	predicted value of N_e (input)
QM	$(1 + D)N_e$ (input, may be obtained from CONSRV)
NPH	photon density vector (input)
ANHS	N_1^s (input)
BETAPL	β_1^{z+1} (input)
GLTE	$(d/dt)\ln T_e$ (input)
CGME	predicted R_e (input)
DGME	$N_e\ \partial R_e/\partial N_e$ (input)
ZS	maximum specie charge number (input)
ZC	parent ion charge number (input)
ANPL	$\overline{N}_1^{z+1}$ (output)
LC	if LC(j) = k > 0, then index k is used in the GP and LU data arrays in TEMPS common for cavity equation j

CVP	cavity rate matrix: CVP(1,j) is the induced transition rate for cavity j and CVP(2,j) is the total spontaneous emission rate. The equation is $\frac{d}{dt} N_{ph}(j) = CVP(1,j) + n_e \, CVP(2,j)$.
LGP	logical constant set to true for consideration of the Penning effect
NCORE	specie index (input)
NEXP	index for excited level of Penning effect (set negative for donor)
KPEN	Penning rate constant (set negative for two electron excitation)
RPEN	net Penning rate (output)
ANMET	metastable density at first input (to receptor atom)
ANMEQ	equilibrium metastable density at first input
EMET	metastable excitation energy divided by kT_e at first input
LERR	logical constant output, set true in case of a matrix singularity that prevents an inversion (quasistationary levels)
IGMX	location of maximum positive noncavity gain in TEMPS data arrays GP and LU. Set null if there are no inversions or if all inversions are coupled to the cavity.
FMD	vector containing the mode spacing parameters $p_o \Delta\nu$
CVOUT	cavity output array: CVOUT(1,j) is the gain logarithm and CVOUT(2,j) is the transition width ratio for cavity j

The function $E_1(x)$ is used in the calculation of radiative recombination; t is approximated by the define procedure FX. At the start of the subroutine, he collisional rate coefficient matrix is placed in JCB. Values of normal denity for the atom or ion are placed in ANR. Corresponding Saha factors are laced in BETA. Array AM and SRMR are used to hold derivative forms of the rate atrix with respect to populations.

The internal subroutine CAVITY calculates the cavity rate coefficient $C = s\omega/t_c$ and its logarithmic derivative with respect to the gain logaritm, RCP. he internal subroutine INVERS calculates the inverse of an upper quadrant of atrix AMS (corresponding to the quasistationary level indices) and multiplies the inverse into the array BVS composed of other elements of AMS and possibly vector F. These operations serve in the formation of reduced rate and Jacobian matrices.

The vector $\partial N_e/\partial \overline{N}_n$ is proportional to the vector of Saha factors; it is used in calculating cross terms involving different atom or ion types. The 'reduced' form of the Saha factors is output through STORE common in symbol ANR. Also at output, GP(3,i) contains the relevant transition escape factor for $i \geq$ IGMX > 0, and for cavity transitions contains the upper level to ground escape factor.

The Penning source code uses the following conventions. For two electron excitation, the rate constant KPEN is set negative in calls involving the receptor specie. Donor status is indicated by setting level index NEXP negative. In receptor calls, NEXP equals the total level increment and is $\leq n^*$ for one electron excitation/ionization, $>n^*$ for the two electron case in the receptor atom call. For the two electron case or single ionization, the receptor (single charge) ion type must also be called with the Penning effect invoked (LGP true) and with NEXP $\leq n^*$. Inputs to the receptor ion call are obtained from the atom call outputs. Consider the increment δF_n in an appropriate element of vector F due to added Penning rates. ANMET contains $(-\beta_1 \delta F_1)$ for the atom at output. If u denotes the receptor ion terminal level ($1 \leq u \leq n^*$), the receptor ion input ANMET = $\beta_u \delta F_u$. For ionization plus excitation, these are equal and the ANMET value is simple passed in the calls; for single or double ionization, RPEN/N_e^2 must be added to ANMET between calls. Also at receptor atom output (NEXP $\geq n^*$ at input), the equilibrium density is multiplied by β_1 and the ionization potential is subtracted from the metastable energy. Both these new forms are passed as inputs to the ion call. The final call to SRJM involving the Penning effect is that of the donor. Inputs are: ANMET - the equilibrium donor ground level density based on the upper level density, ANMEQ - the actual ground level density, and RPEN - $(\beta_1 \delta F_1)$ for the donor which equals the negative of the same expression for the receptor (ANMET receptor atom output). During any call invoking the Penning effect, the excitation/ionization energy is subtracted from the EMET input. After the last receptor specie call, the increment in the electron rate heating is thus the product of the output EMET and the rate RPEN.

```
-TFOR,SI  SUB16
      SUBROUTINE SRJM(RL,JCB,F,AI,AR,ARR,QCRI,QCRR,KS,MS,AN,ANEP,QM,NPH,
     C    ANHS,BETAPL,GLTE,CGME,DGME,ZS,ZC,ANPL,LC,CVP,LGP,NCORE,NEXP,
     C    KPEN,RPEN,ANMET,ANMEQ,EMET,LERR,IGMX,FMD,CVOUT)
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      INCLUDE SPEC4,LIST
      INCLUDE SPEC6,LIST
      REAL JCB,KPEN,NPH(4),LPI
      LOGICAL LGP,LGI,LPRNT,LGL,LERR
      PARAMETER   NWS=4*NLM,NPP=NPL+1,KBV=(NPP*NPP)/4,
     C NSBT=2*NLM+1,NRF1=NPL+KBV+NLM
      DIMENSION RL(NLM,NPL),JCB(NLM,NPL),F(NLM),ANX(NLM),BETA(NLM),
     C    ANR(NLM),LC(4),BVS(KBV),AM(NLM,NLM),DW(NWS),RMT(NLM,NPL),
     C    CVP(2,4),FMD(4),CVOUT(3,4)
      EQUIVALENCE(SV(1),ANR(1)),(SV(NPL),BETA(1)),(SV(NSBT),BVS(1)),
     C    (TMP(21), DW(1)),(SV(NRF1),AM(1,1)),(TMP(1),RMT(1,1)),
     C    (SV(NSBT),ANX(1))
      FX(X)=1.12*EXP(-X)/(X+0.935*X**0.3)
      DATA S6/7.95775E-2/LPRNT/.TRUE./
111   FORMAT(1H0,3X31HIMPROPER PENNING RECEPTOR LEVEL)
112   FORMAT(4X22HPENNING EFFECT IGNORED)
113   FORMAT(1H0,4X20HRATE MATRIX SINGULAR)
114   FORMAT(1H0,4X24HJACOBIAN MATRIX SINGULAR)
      LERR=.FALSE.
      IRST=MS
      LN=INX(KS,2)-1
      LG=INX(KS,5)-1
      LW=INX(KS,7)-1
      LF=INX(KS,8)-1
      L=INX(KS,1)+2
      GNS=DS(L)
      L=INDS(KS+1)
      LF=LF-L
      NB=NBAR(KS)
      NL=NSTAR(KS)-1
      CALL CRATE(TE,KS,NL)
      DO 10 K=1,NL
      JCB(K,NPL)=RMT(K,NPL)
      DO 10 J=1,NL
      RL(J,K)=0.0
 10   JCB(J,K)=RMT(J,K)
      DO 12 K=1,L
      IF(K.EQ.NB) MR=MS
      MS=MS+1
 12   ANR(K)=ANB(MS)
      ANPL=ANHS
      IF(ZC.LT.ZS) ANPL=ANB(MS+1)
      S1=2.0708E-7/(GNS*TE*SQRT(TE))
      S2=15.79*ZC*ZC/TE
      S3=157.46*ZC**4*GNS
      S4=DS(NL+LN+1)**2
      S4=S2/S4
      S5=1.0
      IF(ZC.LT.ZS) S5=ANEP*BETAPL
      AI=0.0
      AR=0.0
      ARR=0.0
      QCRI=0.0
```

```
      QCRR=0.0
      DO 20 K=1,NL
      RSUM=0.0
      DO 22 J=1,NL
   22 RSUM=RSUM+JCB(K,J)*(ANR(K)-ANR(J))
C     ARRAY F IS USED IN THE JACOBIAN MATRIX
      F(K)=(S5*ANPL-ANR(K))*JCB(K,NPL)-RSUM
      IF(K.LT.NB)  F(K)=F(K)-ANR(K)*DGME/ANEP
      Q=DS(K+LN)**2
      U=S2/Q
      A=S3*DS(K+LW)*FX(U-S4)/(DS(K+LG)*Q*DS(K+LN))
      Q=S1*DS(K+LG)*EXP(U)
      ARR=ARR+Q*A
      ANX(K)=ANEP*Q*ANR(K)
C     ARRAY BETA IS USED IN THE JACOBIAN MATRIX
      BETA(K)=Q
      A=A+ANEP*JCB(K,NPL)
      AR=AR+Q*A
      QCRR=QCRR+Q*U*A
      A=S5*A
      RL(K,NPL)=A
      SRMR=ANPL
      IF(.NOT.(ZC.LT.ZS)) SRMR=SRMR+QM-ANC(NCORE)
      F(K)=F(K)+A*SRMR/ANEP
      A=Q*ANR(K)*JCB(K,NPL)
      AI=AI+A
      QCRI=QCRI+U*(A+Q*RSUM)
   20 CONTINUE
      CALL LINWID(TE,T,ANEP,AN,KS,NL)
      CALL ESCAPE(ANX,KS,NL,AN)
      DO 25 K=1,4
      LC(K)=0
      IF(ICAV(1,K).NE.KS) GO TO 25
      CVP(1,K)=0.0
      CVP(2,K)=0.0
      DO 24 J=1,4
      IF(ICAV(2,K).NE.LU(1,J)) GO TO 24
      IF(ICAV(3,K).NE.LU(2,J)) GO TO 24
      LC(K)=J
   24 CONTINUE
      IF(LC(K).EQ.0) LC(K)=-1
   25 CONTINUE
      S3=0.5101*S3/GNS
      DO 48 J=1,NL
      S1=0.0
      S4=0.0
      RSUM=0.0
      U=0.0
      SRMR=0.0
      R(J+IRST)=0.0
      R1=1.0/DS(J+LN)**2
      DO 46 K=1,NL
      IF(J.EQ.K) GO TO 46
      RL(J,K)=ANEP*JCB(J,K)
      S1=S1+RL(J,K)
      Q=R1-1.0/DS(K+LN)**2
      A=S3*Q*Q
      R2=DS(K+LG)/DS(J+LG)
      R3=BETA(K)/BETA(J)
```

```
      R4=ANR(K)/ANR(J)
      IF(J.GT.K) GO TO 37
      A=A*ABS(DS(K+L*J+LF))
      IS=K+IFN(J,L)
      Q=R3*A*PHI(IS)
      RL(J,K)=RL(J,K)+Q
      LGI=(ABS(PHI(IS)-1.0).LT.1.0E-3)
      IF(LGI) AM(J,K)=Q*(PHI(IS)-1.0)
      IF(.NOT.LGI) AM(J,K)=Q*W(IS)
      RSUM=RSUM+AM(J,K)*ANR(K)
      IF(.NOT.LGI) GO TO 27
      AM(J,K)=Q*W(IS)
      SRMR=SRMR-R2*AM(J,K)/R3
27    Q=1.0/(R3*R4)-1.0/R2
      IF(LGI) GO TO 28
      IF(ABS(Q).LT.1.0E-7) Q=SIGN(1.0E-7,Q)
      SRMR=SRMR+AM(J,K)/(R3*Q)
      AM(J,K)=-AM(J,K)/(R2*Q)
28    IF(NCAV.LE.0) GO TO 36
      DO 35 J1=1,NCAV
      I=LC(J1)
      IF(I.LE.0) GO TO 35
      IF(LU(1,I).NE.J) GO TO 35
      IF(LU(2,I).NE.K) GO TO 35
      CALL CAVITY(RC,RCP,GP(1,I),GP(2,I),GP(3,I),FMD(J1))
      CVOUT(1,J1)=GP(1,I)
      CVOUT(2,J1)=GP(2,I)
      CVP(1,J1)=RC*NPH(J1)
      IF((-R2*Q).LE.1.0E-6) GO TO 29
      RCP=NPH(J1)*(RCP-RC)
      CVP(2,J1)=RCP
      RCP=RCP*R3/ANX(K)
      AM(J,K)=AM(J,K)-RCP/(R2*Q)
      SRMR=SRMR+RCP/(R3*Q)
      RSUM=RSUM+CVP(2,J1)/(ANEP*BETA(J))
      RC=R3*CVP(1,J1)/ANX(K)
      S1=S1-RC/(R3*Q)
      RL(J,K)=RL(J,K)-RC/(R2*Q)
      GO TO 30
29    CVP(2,J1)=-1.0
30    IGMX=K+IFN(1,L)
      GP(3,I)=PHI(IGMX)
35    CONTINUE
36    A=S6*A
      IF(LGI) A=R2*A
      GO TO 45
37    A=A*ABS(DS(J+L*K+LF))
      IS=J+IFN(K,L)
      Q=A*PHI(IS)
      S4=S4+Q
      LGI=(ABS(PHI(IS)-1.0).LT.1.0E-3)
      IF(LGI) AM(J,K)=Q*(PHI(IS)-1.0)
      IF(.NOT.LGI) AM(J,K)=Q*W(IS)
      U=U+AM(J,K)
      IF(.NOT.LGI) GO TO 38
      Q=Q*W(IS)
      AM(J,K)=R3*Q/R2
      SRMR=SRMR-Q
38    Q=R3*R4/R2-1.0
```

```
      ST=R2/R3-R4
      IF(LGI) GO TO 39
      IF(ABS(Q).LT.1.0E-7) Q=SIGN(1.0E-7,Q)
      IF(ABS(ST).LT.1.0E-7) ST=SIGN(1.0E-7,ST)
      SRMR=SRMR+AM(J,K)/Q
      AM(J,K)=AM(J,K)/ST
 39   IF(NCAV.LE.0) GO TO 44
      DO 42 J1=1,NCAV
      I=LC(J1)
      IF(I.EQ.0) GO TO 42
      IF(I.LT.0) GO TO 41
      IF(LU(1,I).NE.K) GO TO 42
      IF(LU(2,I).NE.J) GO TO 42
      IF(CVP(2,J1).LT.0.0) GO TO 40
      RCP=CVP(2,J1)
      RCP=RCP/ANX(J)
      AM(J,K)=AM(J,K)+RCP/ST
      SRMR=SRMR+RCP/Q
      RSUM=RSUM-CVP(2,J1)/(ANEP*BETA(J))
      RC=CVP(1,J1)/ANX(J)
      S1=S1-RC/Q
      RL(J,K)=RL(J,K)+RC/ST
 40   CVP(1,J1)=CVP(1,J1)-NPH(J1)*FCAV
      CVP(2,J1)=A*ANX(J)
      GO TO 42
 41   IF(ICAV(2,J1).NE.K) GO TO 42
      IF(ICAV(3,J1).NE.J) GO TO 42
      GO TO 40
 42   CONTINUE
 44   A=S6*A
 45   IF(LGI) A=A*W(IS)
      IF(.NOT.LGI) A=A*(1.0-PHI(IS))/Q
      R(J+IRST)=R(J+IRST)+A
 46   CONTINUE
      S1=S1+ANEP*JCB(J,NPL)
      RL(J,J)=-S1-S4
      AM(J,J)=SRMR
      F(J)=F(J)+(RSUM-U*ANR(J))/ANEP
      R(J+IRST)=R(J+IRST)+S6*S1
      IF(J.GE.NB) GO TO 48
      Q=1.5+S2*R1
      RL(J,J)=RL(J,J)+Q*GLTE-CGME
      Q=ANEP*(ZS+1.0-ZC)*ANR(J)
      DO 47 K=1,NL
 47   AM(J,K)=AM(J,K)-Q*BETA(K)*JCB(K,NPL)
 48   CONTINUE
C     FORM JACOBIAN MATRIX
      S1=ANEP*ANEP*(ZS+1.0-ZC)/QM
      DO 49 J=1,NL
      JCB(J,NPL)=BETA(J)*JCB(J,NPL)
      DO 49 K=1,NL
 49   JCB(K,J)=RL(K,J)+AM(K,J)-S1*F(K)*BETA(J)
C     START PENNING SOURCE CODE
C     PENNING EFFECT TEST
      IF(.NOT.LGP) GO TO 90
      LGI=.TRUE.
      IF(NEXP.GE.NSTAR(KS)) GO TO 70
      IF(KPEN.LT.0.0.OR.NEXP.EQ.1) GO TO 80
      IF(NEXP.GE.NB) GO TO 60
```

```
50    IS=IABS(NEXP)
      R1=BETA(1)/BETA(IS)
      R2=ANMET*KPEN
      R3=ANMEQ*KPEN
      IF(NEXP.GT.0) GO TO 51
      R2=R2/R1
      R3=R3/R1
      NEXP=-NEXP
      R4=-RPEN/BETA(1)
      GO TO 52
51    R4=R2*ANR(1)-R3*ANR(NEXP)
      RPEN=ANEP*BETA(1)*R4
      R4=R4/ANEP
52    RL(NEXP,1)=RL(NEXP,1)+R1*R2
      JCB(NEXP,1)=JCB(NEXP,1)+R1*R2
      RL(1,NEXP)=RL(1,NEXP)+R3
      JCB(1,NEXP)=JCB(1,NEXP)+R3
53    Q=1.0/DS(1+LN)**2-1.0/DS(NEXP+LN)**2
      F(NEXP)=F(NEXP)+R1*R4
      R(NEXP+IRST)=R(NEXP+IRST)+S6*R1*R3
      RL(NEXP,NEXP)=RL(NEXP,NEXP)-R1*R3
      JCB(NEXP,NEXP)=JCB(NEXP,NEXP)-R1*R3
54    LGL=(KPEN.GE.0.0.OR.NEXP.GT.NSTAR(KS)).AND.(NEXP.NE.1)
      DO 56 J=1,NL
      S4=S1*R4*BETA(J)
      IF(LGI) JCB(NEXP,J)=JCB(NEXP,J)-R1*S4
56    IF(LGL) JCB(1,J)=JCB(1,J)+S4
      IF(.NOT.LGL) GO TO 82
      RL(1,1)=RL(1,1)-R2
      JCB(1,1)=JCB(1,1)-R2
      F(1)=F(1)-R4
      R(1+IRST)=R(1+IRST)+S6*R2
      EMET=EMET-S2*Q
      GO TO 90
60    IF(LPRNT) WRITE(6,111)
      LPRNT=.FALSE.
      IF(KPEN.GE.0.0) GO TO 50
      LGP=.FALSE.
      WRITE(6,112)
      GO TO 90
70    Q=1.0/DS(1+LN)**2
      R2=ANMET*ABS(KPEN)
      R4=ABS(KPEN)/ANEP
      S3=2.0*ANPL
      IF(.NOT.(ZC.LT.ZS)) S3=S3-ANC(NCORE)
      S3=S5*S3
      IF(KPEN.LT.0.0) GO TO 74
72    S4= ANMET*ANR(1)-ANMEQ*S3
      RPEN=ANMET*ANR(1)-ANMEQ*S5*ANPL
      GO TO 78
74    IF(ZC.LT.ZS.OR.NEXP.EQ.NSTAR(KS)) GO TO 76
      KPEN=-KPEN
      NEXP=NSTAR(KS)
      GO TO 72
76    IS=NEXP-NSTAR(KS)
      IF(IS.LE.0) GO TO 86
      IF(IS.EQ.NSTAR(KS+1)) GO TO 79
      IS=MS+IS
      S3=2.0*ANB(IS)
```

```
      RPEN=ANMET*ANR(1)-S5*ANMEQ*ANB(IS)
   77 S4=ANMET*ANR(1)-S5*ANMEQ*S3
   78 R4=S4*R4
      ANMET=R4*BETA(1)
      IF(NEXP.EQ.NSTAR(KS)) R4=R4-ANMEQ*QM*ABS(KPEN)/ANEP
      IF(NEXP.EQ.(NSTAR(KS)+NSTAR(KS+1)))   R4=R4-S5*ANMEQ*QM*
     C    ABS(KPEN)/(2.0*ANEP)
      RPEN=ANEP*BETA(1)*RPEN*ABS(KPEN)
      LGI=.FALSE.
      ANMEQ=ANMEQ*BETA(1)
      GO TO 54
   79 IF(ZC.LT.ZS-1) GO TO 60
      S3=3.0*ANHS-ANC(NCORE)
      RPEN=ANMET*ANR(1)-S5*ANMEQ*ANHS
      GO TO 77
   80 R1=1.0/BETA(NEXP)
   81 R2=ANEP*BETA(1)
      R3=R2*ANMEQ*ABS(KPEN)
      R4=ANMET
      IF(NEXP.GE.NSTAR(KS)) GO TO 82
      GO TO 53
   82 EMET=EMET-S2*Q
      ANMET=R2
      ANMEQ=R1*ANMEQ
      GO TO 90
   86 R1=ANEP/S5
      GO TO 81
   90 CONTINUE
C     END PENNING SOURCE CODE
      J=1
      IF(NCAV.LE.0) GO TO 96
   92 CONTINUE
      DO 94 K=1,NCAV
      I=LC(K)
      IF(I.NE.J) GO TO 94
      J=J+1
      GO TO 92
   94 CONTINUE
   96 IGMX=0
   97 IF(J.GE.5) GO TO 100
      IF(GP(1,J).GT.0.0) GO TO 98
      GO TO 100
   98 IF(IGMX.EQ.0) IGMX=J
      IS=LU(1,J)
      IS=LU(2,J)+IFN(IS,L)
      GP(3,J)=PHI(IS)
      J=J+1
      GO TO 97
  100 IF(NB.GE.NSTAR(KS)) GO TO 160
      IS=NB-1
      NV=NSTAR(KS)-NB
      CALL  INVERS(RL,BVS,NV,.TRUE.,F,$130)
      DO 120 K=1,NV
      ANB(K+MR)=-RL(K+IS,NPL)*ANPL
      DO 120 J=1,IS
      ANB(K+MR)=ANB(K+MR)-RL(K+IS,J)*ANB(J+IRST)
      F(J)=F(J)-RL(J,K+IS)*F(K+IS)
      RL(J,NPL)=RL(J,NPL)-RL(J,K+IS)*RL(K+IS,NPL)
      DO 120 I=1,IS
```

```
  120   RL(J,I)=RL(J,I)-RL(J,K+IS)*RL(K+IS,I)
        GO TO 135
  130   WRITE(6,113)
  133   LERR=.TRUE.
        GO TO 160
  135   CALL  INVERS(JCB,BVS,NV,.FALSE.,ANR,$150)
        DO 140 J=1,IS
        ANR(J)=BETA(J)
        DO 140 K=1,NV
        ANR(J)=ANR(J)-BETA(K+IS)*JCB(K+IS,J)
        JCB(J,NPL)=JCB(J,NPL)-JCB(K+IS,NPL)*JCB(K+IS,J)
        DO 140 I=1,IS
  140   JCB(I,J)=JCB(I,J)-JCB(I,K+IS)*JCB(K+IS,J)
        GO TO 160
  150   WRITE(6,114)
        GO TO 133
  160   RETURN
C     IF NBAR IS LESS THAN NSTAR THE FOLLOWING SUBROUTINE IS USED IN
C     CALCULATING  EFFECTS OF QUASI-STATIONARY LEVELS
        SUBROUTINE  INVERS(AMS,BVS,NV,LOGC,F,$)
        LOGICAL LOGC
        REAL AMS(NLM,1),BVS(NV,1),F(NLM)
        DO 10 K=1,NV
        DO 10 J=1,NV
   10   AM(J,K)=AMS(J+IS,K+IS)
        DO 20 J=1,NV
        DO 14 K=1,IS
   14   BVS(J,K)=AMS(J+IS,K)
        IF(LOGC) BVS(J,NB)=AMS(J+IS,NPL)
   20   IF(LOGC) BVS(J,NB+1)=F(J+IS)
        I=IS
        IF(LOGC) I=I+2
        CALL SOR(AM,NLM,NV,BVS,NV,I,$30,DW)
        GO TO 40
   30   RETURN 6
   40   DO 50 J=1,NV
        DO 44 K=1,IS
   44   AMS(J+IS,K)=BVS(J,K)
        IF(LOGC) AMS(J+IS,NPL)=BVS(J,NB)
   50   IF(LOGC) F(J+IS)=BVS(J,NB+1)
        RETURN
C     THE FOLLOWING SUBROUTINE CALCULATES FREQUENCY AVERAGED CAVITY
C     RATE FUNCTIONS NEGLECTING HOLE BURNING
        SUBROUTINE CAVITY(RC,RCP,G,X,P,CF)
        CF=CGM*P
        CK=(1.0-LPI(X))/(1.0+CF)
        S=-G/ALOG(CMR)
        ST=AMAX1(1.0-S,0.0)
        RC=FCAV*S*(1.0-CK*ST**0.3
        ST=AMIN1(S,0.99)
        IF(S.LE.0.99) GO TO  20
        IF(S.GE.1.01) GO TO  10
        CK=CK*EXP(1.0E3*(ST-S))
        GO TO  20
   10   CK=0.0
   20   RCP=FCAV*S*(1.0-CK*(1.0-1.3*ST)/(1.0-ST)**0.7)
        RETURN
        END
```

Subroutine MATRIX is used to collect the results from calls to SRJM and form the overall rates and Jacobian. The argument list is as follows:

ANR	vector of dependent variables (lower level densities, photon densities, $d\ell nT_e/dt$), input
FF	vector of lower level rates
DMJ	Jacobian matrix
BETA	vector of lower level Saha factors
ANEP	predicted N_e input
ANP	vector of maximum ionization densities
GLTE,CGME,DGME	(see SRJM list)
DE	electron rate heating/kT_e
LCD	input integer vector; total number of lower level equations of species with indices $<i$ is LCD(i)
AN,EP,DI,DEL	(see CONSRV list)
LERR,CVP	(see SRJM list)
ANPH	photon density vector
FMD	vector of mode spacing parameters
CVT	same as CVOUT in SRJM (cavity outputs)
GOUT, LOUT	maximum positive noncavity gain outputs: GOUT(1,k) is gain logarithm, GOUT(2,k) is escape factor, LOUT(i,k) are level indices (i = 1 lower, i = 2 upper) for ion or atom types k

In the subroutine DEM is the overall rate matrix. Also RPEN is the Penning rate constant rather than KPEN and PRATE is the net Penning rate rather than RPEN (see SRJM list). DPEN is the increment to the electron rate heating due to the Penning effect. At output, ANEP is the corrected electron density.

```
-TFOR,SI  SUB17
       SUBROUTINE MATRIX(ANR,FF,DMJ,BETA,ANEP,ANP,GLTE,CGME,DGME.DE,LCD,
      C     AN,EP,DI,DEL,LERR,CVP,ANPH,FMD,CVT,GOUT,LOUT)
       INCLUDE SPEC1,LIST
       INCLUDE SPEC2,LIST
       INCLUDE SPEC4,LIST
       INCLUDE SPEC5,LIST
       INCLUDE SPEC6,LIST
       INCLUDE SPEC7,LIST
       PARAMETER NLL=NCM+1,NST1=3*KM+NPL,NST2=NST1+KM*KM,NST3=NST2+2*NHSQ
       PARAMETER N17S=KM+1,N17S2=N17S+KM,N17S3=N17S2+KM
       DIMENSION ANP(NCM),LCD(NLL),BETP(NLM),DEM(KM,KM),DMJ(KM,KM),
      C     FF(KM),BETA(KM),GG(KM),H(KM),BTP(KM),RL(NLM,NPL),
      C     AJB(NLM,NPL),F(NLM),LC(4),CVP(2,4),FMD(4),CVT(3,4),
      C     GOUT(2,NSP),LOUT(2,NSP)
       LOGICAL LERR,LGP
       EQUIVALENCE (BETP(1),SV(1)),(GG(1),STRMTX(1)),(H(1),STRMTX(N17S)),
      C (BTP(1),STRMTX(N17S2)), (F(1),STRMTX(N17S3)), (DEM(1,1),
      C STRMTX(NST1)),(RL(1,1),STRMTX(NST2)),(AJB(1,1),STRMTX(NST3))
       MS=0
       IS=0
       DO 5 J=1,NC
       IF(J.EQ.NC) GO TO 1
       K2=IC(J+1)-1
       GO TO 2
   1   K2=NSP
   2   K1=IC(J)
       DO 5 K=K1,K2
       L=INDS(K+1)
       MI=NBAR(K)-1
       DO 3 I=1,MI
   3   ANP(I+MS)=ANR(I+IS)
       IS=IS+MI
       MS=MS+L
   5   CONTINUE
       DO 7 K=1,IS
       DO 7 J=1,IS
       DEM(J,K)=0.0
   7   DMJ(J,K)=0.0
       CALL CONSRV(DPEN,ANP,EP,DI,DEL,GM,AN,0)
       B1=0.0
       B2=0.0
       B3=0.0
       DPEN=0.0
       R1=0.0
       R2=0.0
       LR1=0
       LR2=0
       LD1=0
       LD2=0
       MS=0
       IS=0
       DO 100 J=1,NC
       IF(J.EQ.NC) GO TO 31
       K2=IC(J+1)-1
       GO TO 51
  31   K2=NSP
  51   K1=IC(J)
       ZS=K2-K1+1
```

```
      ANHS=ANP(J)
      NEXP=0
      L1=LCD(J)+1
      L2=LCD(J+1)
      DO 100 K=K1,K2
      ZC=K-K1+1
      IF(K.LT.K2) GO TO 61
      BETAPL=1.0/ANEP
      GO TO 71
 61   L=INDS(K+1)
      BETAPL=BT(MS+L+1)
 71   LGP=.FALSE.
      L=NBAR(K)-1
      DO 81 I=1,L
      BETA(I+IS)=BT(I+MS)
 81   GG(I+IS)=ANEP*ANEP*(ZS-ZC+1.0)/QM
      IF(.NOT.LPN) GO TO 40
     PENNING SOURCE CODE - SET A
      IPEN=0
      IF(J.EQ.JRC.AND.K.EQ.K1)    GO TO 10
      IF(J.EQ.JRC.AND.K.EQ.K1+1)  GO TO 20
      IF(J.EQ.JDC.AND.K.EQ.K1)    GO TO 30
      GO TO 40
 10   IPEN=1
      RATIN=RPEN
      LGP=.TRUE.
      NEXP=JRL
      IF(RPEN.LT.0.0) NEXP=NEXP+NSTAR(K)
      L=IC(JDC)
      I=INX(L,2)-1
      EMET=15.79/TE
      EMET=EMET/DS(I+1)**2-EMET/DS(I+JDL)**2
      L=L-1
      MI=0
      DO 12 I=1,L
 12   MI=MI+INDS(I+1)
      ANMET=ANEP*BT(MI+JDL)*ANB(MI+JDL)
      ANMEQ=ANEP*BT(MI+JDL)*ANB(MI+1)
      A1=ANEP*BT(MS+1)
      ANGR=A1*ANB(MS+1)
      IF(NEXP.LT.NSTAR(K)) ANUR=A1*ANB(MS+JRL)
      IF(NEXP.EQ.NSTAR(K).AND.K.EQ.K2) ANUR=A1*ANHS
      GO TO 40
 20   IPEN=2
      RATIN=RPEN
      IF(NEXP.LT.NSTAR(K1)) GO TO 4
      LGP=.TRUE.
      NEXP=1
      IF(RPEN.LT.0.0) GO TO 22
      ANUR=A1*ANEP*BT(MS+1)*ANPL
      GO TO 40
 22   NEXP=JRL
      IF(NEXP.EQ.NSTAR(K)) GO TO 24
      ANUR=A1*ANB(MS+JRL)
      GO TO 40
 24   ANUR=A1*ANHS
      GO TO 40
 30   IPEN=3
      RATIN=ABS(RPEN)
```

```
      LGP=.TRUE.
      PRATE=SAVE
      NEXP=-JDL
      EMET=0.0
      IF(RPEN.LT.0.0) ANUR=ANMET*ANUR
      ANMET=ANUR
      ANMEQ=ANGR
C    END SOURCE SET A
 40   CONTINUE
      CALL SRJM(RL,AJB,F,AI,AR,ARR,QCRI,QCRR,K,MS,AN,ANEP,QM,ANPH,
     C     ANHS,BETAPL,GLTE,CGME,DGME,ZS,ZC,ANPL,LC,CVP,LGP,J,NEXP,RATIN,
     C     PRATE,ANMET,ANMEQ,EMET,LERR,IGMX,FMD,CVT)
      IF(LERR) GO TO 110
      GOUT(1,K)=0.0
      GOUT(2,K)=0.0
      LOUT(1,K)=0
      LOUT(2,K)=0
      IF(IGMX.LE.0) GO TO 41
      GOUT(1,K)=GP(1,IGMX)
      GOUT(2,K)=GP(3,IGMX)
      LOUT(1,K)=LU(1,IGMX)
      LOUT(2,K)=LU(2,IGMX)
 41   A2=K2+1-K
      A3=ANEP*BETAPL*ANPL
      B1=B1+A2*(ANEP*AI-A3*AR)
      B2=B2+A2*A3*(ARR-AR)
      IF(K.EQ.K2) B2=B2+AR*ANC(J)
      B3=B3+A3*QCRR-ANEP*QCRI
      L=NBAR(K)-1
      DO 42 MI=1,L
      BTP(MI+IS)=BETP(MI)
      H(MI+IS)=-F(MI)
      IF(K.EQ.K2) H(MI+IS)=H(MI+IS)+QM*RL(MI,NPL)/A
 42   GG(MI+IS)=GG(MI+IS)*BETP(MI)
      DO 50 MI=1,L
      IF(K.LT.K2) GO TO 46
      FF(MI+IS)=RL(MI,NPL)*ANP(J)
      DO 44 NI=L1,L2
 44   DMJ(MI+IS,NI)=-RL(MI,NPL)*ANEP*BTP(NI)
      GO TO 48
 46   FF(MI+IS)=0.0
      DEM(MI+IS,L+1+IS)=RL(MI,NPL)
      DMJ(MI+IS,L+1+IS)=RL(MI,NPL)
 48   DO 50 NI=1,L
      DEM(NI+IS,MI+IS)=RL(NI,MI)
      DMJ(NI+IS,MI+IS)=DMJ(NI+IS,MI+IS)+AJB(NI,MI)
 50   CONTINUE
      NI=LCD(NC+1)
      A3=A2*ANEP
      A2=A2*AR
      DO 60 I=1,NI
      IF(K.LT.K2) GO TO 54
      DO 52 MI=L1,L2
 52   DMJ(I,MI)=DMJ(I,MI)-ANR(I)*A2*ANEP*BTP(MI)
      GO TO 56
 54   DMJ(I,L+1+IS)=DMJ(I,L+1+IS)+A2*ANEP*BETAPL*ANR(I)
 56   IF(I.GT.IS.AND.I.LE.IS+L) GO TO 60
      DO 58 MI=1,L
      DMJ(I,MI+IS)=DMJ(I,MI+IS)-ANR(I)*A3*AJB(MI,NPL)
```

```
   58 CONTINUE
   60 CONTINUE
      IF(.NOT.LPN) GO TO 99
C     PENNING SOURCE CODE - SET B
      IF(IPEN.GT.0) GO TO (70,80,90),IPEN
      GO TO 98
   70 IF(LGP) GO TO 72
      LPN=.FALSE.
      GO TO 99
   72 A2=0.0
      A3=0.0
      A4=0.0
      SAVE=PRATE/ANEP**2
      LR1=IS+1
      LR2=IS+JRL
      IF(RPEN.LT.0.0) LR2=LR2+L
      IF(RPEN.GE.0.0.AND.JRL.GE.NBAR(K1).OR.RPEN.LT.0.0.AND.
     C   JRL.GE.NBAR(K1+1)) LR2=0
      IF(NEXP.GE.NSTAR(K1)) GO TO 73
      R1=SAVE
      ANMET=SAVE
      GO TO 76
   73 R1=ANMET
      IF(RPEN.LT.0.0.AND.NEXP.LT.NSTAR(K1)+NSTAR(K1+1)) GO TO 74
      IF(RPEN.GE.0.0.AND.K1.EQ.K2) A2=RPEN
      IF(RPEN.LE.0.0) A2=ANEP*BETAPL*ABS(RPEN)/2.0
      A2=A2*ANMEQ*QM/ANEP
      A3=SAVE
   74 SAVE=ANMET-A2
      ANMET=ANMET+A3
   76 R2=ANMET
      IF(JRL.EQ.NSTAR(K1+1).AND.RPEN.LT.0.0.OR.JRL.EQ.NSTAR(K1).AND.
     C   K1.EQ.K2) GO TO 78
      GO TO 79
   78 R2=0.0
      A4=ANMEQ*ANEP*ABS(RPEN)
      IF(RPEN.LT.0.0) A4=A4*ANEP*BETAPL
   79 IF(RPEN.LT.0.0.OR.K1.LT.K2) GO TO 98
   80 DPEN=EMET*PRATE
      GO TO 98
   90 LD1=IS+1
      LD2=IS+JDL
      IF(JDL.GE.NBAR(K1)) LD2=0
      GO TO 98
   96 DO 97 I=L1,L2
      IF(J.EQ.JRC) DMJ(LR1,I)=DMJ(LR1,I)+A4*BTP(I)/BETA(LR1)
      IF(J.EQ.JDC) DMJ(LD1,I)=DMJ(LD1,I)-A4*BTP(I)/BETA(LD1)
      IF(J.EQ.JDC.AND.LD2.GT.0)DMJ(LD2,I)=DMJ(LD2,I)+A4*BTP(I)/BETA(LD2)
   97 CONTINUE
      GO TO 99
   98 IF(K.EQ.K2.AND.NEXP.NE.0) GO TO 96
C     END SOURCE SET B
   99 IS=IS+L
  100 CONTINUE
      IS=0
      NI=LCD(NC+1)
      DO 104 K=1,NSP
      L=NBAR(K)-1
      DO 102 MI=1,L
```

```
      J=MI+IS
      A3=0.0
      IF(J.EQ.LR1) A3=R1/BETA(LR1)
      IF(J.EQ.LR2) A3=-R2/BETA(LR2)
      IF(J.EQ.LD1) A3=-R1/BETA(LD1)
      IF(J.EQ.LD2) A3=R1/BETA(LD2)
      DO 102 I=1,NI
      IF(I.GT.IS.AND.I.LE.IS+L) GO TO 102
      DMJ(J,I)=DMJ(J,I)+(H(J)+A3)*GG(I)
102   CONTINUE
104   IS=IS+L
      CALL CONSRV(ANEP,ANP,EP,DI,DEL,QM,AN,1)
      CGME=B1
      DGME=B1+B2
      DE=B3*ANEP+DPEN
      DO 106 J=1,NI
      DO 106 K=1,NI
106   FF(J)=FF(J)+DEM(J,K)*ANR(K)
110   RETURN
      END
```

Function TMSA is used to calculate the spontaneous emission factor for the cavity. (DM is the mode spacing parameter.)

Subroutine EICP calculates electron-ion collision parameters according to the results of Ref. 26. (see also Refs. 1 and 5.) The ratio of collision integrals B^*_{ei} is not explicitly used in the program and is set equal to a nominal value of unity. Argument Z is the ion charge number.

Subroutine ELEN calculates $d\ell nT_e/dt$ at a given time TAU (see argument lists of SRJM, MATRIX). The electron energy equation is discussed in Refs. 1 and 5. Input forcing functions are given by FORIN in the following manner: output BM is the magnetic field intensity (nominally null), IDIS is an integer that defines the type of forcing function, FT is T_e and FP is $d\ell nT_e/dt$ if IDIS ≤ 0, FT is the total electric field tangent to the magnetic field and FP is the perpendicular electric field if IDIS = 1 and FT, FP are the component current densities if IDIS > 1. The dummy subroutine calls ADUMYn are to be replaced by subroutines giving electron-atom collision parameters (elastic cross section Q and collision integral ratios B* and C*). Nominal (final) variables in TEMPS common are: H, H1, H3 - Hall parameters (see references), SIGMA - electrical conductivity, ALPHT - thermal diffusion ratio, FEC - electron-heavy particle elastic energy exchange collision frequency, Q - joule dissipation (units of 10^8 Jm^{-3} sec^{-1}) and QR - continuum radiation (MKS units if multiplied by 5.4×10^5).

```
-TFOR,SI  SUB18
      FUNCTION TMSA(ETA,PS,DM,CL,CAR,WL)
      TMSA=ETA
      RETURN
      END
-TFOR,SI  SUB19
      SUBROUTINE EICP(TE,ANE,Z,QEI,BEI,CEI)
      BEI=1.0
      DL=864.88*SQRT(TE/ANE)
      TS=0.151*DL*TE/Z
      IF(TS.LE.1.0) GO TO 10
      DL=DL/TS
      QEI=DL*DL*ALOG(3.0*TS)
      TS=TS**0.2
      CEI=0.351*TS/(TS-0.325)
      RETURN
10    QEI=1.0986*DL*DL/TS
      CEI=0.52
      RETURN
      END
```

```
-TFOR,SI . SUB2C
      SUBROUTINE ELEN(GLTE,TAU,DE,CGME,AN,ANP)
      INCLUDE SPEC1,LIST
      INCLUDE SPEC2,LIST
      DIMENSION CSTR(8),A(2,2),F(2),TN(2),AN(NSP),ANP(NCM)
      EQUIVALENCE (A(1,1),CSTR(1)),(TN(1),CSTR(5)),(F(1),CSTR(7))
      EQUIVALENCE (TMP(21),CSTR(1)),(TMP(29),S),(TMP(30),Q),(TMP(31),B)
     C    (TMP(32),C),(TMP(33),ALPHT),(TMP(34),DEL),(TMP(35),H1),
     C    (TMP(36),H3),(TMP(37),H),(TMP(38),SIGMA),(TMP(39),FEC),
     C    (TMP(40),QR),(TMP(41),Z),(TMP(42),AM)
      CALL FORIN(TAU,IDIS,BM,FT,FP)
      IF(IDIS.LE.0) GO TO 20
      H=15.79/TE
      QR=0.0
      FEC=0.0
      DO 1 J=1,8
    1 CSTR(J)=0.0
      DO 10 J=1,NC
      IF(J.EQ.NC) GO TO 2
      K2=IC(J+1)-1
      GO TO 3
    2 K2=NSP
    3 K1=IC(J)
      H1=K2-K1
      DO 8 K=K1,K2
      Z=K-K1
      I=INX(K,1)+1
      AM=DS(I)
      C=DS(I+2)
      IF(K.LT.K2) S=AN(K+1)
      IF(K.EQ.K2) S=ANP(J)
      I=INX(K,2)+NSTAR(K)-1
      Q=(Z+1.0)/(DS(I)-0.5)
      QR=QR+S*C*(Z+1.0)**4*EXP(H*Q*Q
      S=AN(K)
      IF(K.GT.K1) GO TO 6
      I=1
C     DUMMY SUBR CALLS ARE TO BE CONVERTED BY COLLECTOR
      IF(J.EQ.1) CALL ADUMY1(TE,Q,B,C)
      IF(J.EQ.2) CALL ADUMY2(TE,Q,B,C)
      IF(J.EQ.3) CALL ADUMY3(TE,Q,B,C)
      IF(J.EQ.4) CALL ADUMY4(TE,Q,B,C)
      IF(J.EQ.5) CALL ADUMY5(TE,Q,B,C)
      IF(J.EQ.6) CALL ADUMY6(TE,Q,B,C)
      GO TO 7
    6 I=2
      CALL EICP(TE,ANE,Z,Q,B,C)
    7 TN(I)=TN(I)+S
      S=Q*S
      F(I)=F(I)+S
      FEC=FEC+S/AM
      A(1,I)=A(1,I)+B*S
      A(2,I)=A(2,I)+C*S
      IF(ABS(H1-Z).GT.0.5) GO TO 8
      Z=H1+1.0
      S=ANP(J)
      GO TO 6
    8 CONTINUE
   10 CONTINUE
```

```
      DO 12 K=1,2
      S=F(K)
      DO 12 J=1,2
12    A(J,K)=A(J,K)/S
      S=F(2)/F(1)
      Q=1.0+S
      B=1.0+0.7*S-0.48*A(1,1)
      C=1.2*(A(2,1)+S*A(2,2))-Q
      ALPHT=C/B
      DEL=ALPHT*C/Q
      H1=SQRT(TE)
      QR=ANE*QR/H1
      FEC=FEC*H1
      H1=H1*(F(1)+F(2))
      SIGMA=3.402E3*ANE/H1
      H1=2.1233E3*BM/H1
      H3=0.4*H1*Q/B
      Q=DEL/(1.0+H3*H3)
      H=H1+Q*H3
      Q=1.0-Q
      H=H/Q
      DEL=1.0-DEL
      IF(IDIS.EQ.1) Q=SIGMA*(FT*FT/(Q*(1.0+H*H))+FP*FP/DEL)
      IF(IDIS.GT.1) Q=(Q*FT*FT+DEL*FP*FP)/SIGMA
      GLTE=DE/(1.5*ANE)+(4.829E-3*Q-2.6085E-5*QR)/(ANE*TE)-9.088E-4*
     C    FEC*(TE-T)/TE-CGME
      GO TO 30
20    TE=FT
      GLTE=FP
30    RETURN
      END
```

NASA-JPL-Coml., L.A., Calif.

JPL PUBLICATION 77-11