SAND86-8246
Unlimited Release
Printed December 1986

A FORTRAN COMPUTER CODE PACKAGE FOR THE EVALUATION OF GAS-PHASE, MULTICOMPONENT TRANSPORT PROPERTIES

Robert J. Kee
Computational Mechanics Division
Sandia National Laboratories
Livermore, CA 94550

Graham Dixon-Lewis
Department of Fuel and Combustion Science
University of Leeds
Leeds, LS2 9JT
England

Jürgen Warnatz
Institut für Angewandte Physikalische Chemie
Universität Heidelberg
6900 Heidelberg
Germany

Michael E. Coltrin
Laser and Atomic Physics Division
Sandia National Laboratories
Albuquerque, NM 87185

James A. Miller
Combustion Chemistry Division
Sandia National Laboratories
Livermore, CA 94550

Abstract

This report documents a Fortran computer code package that is used for the evaluation of gas-phase multicomponent viscosities, thermal conductivities, diffusion coefficients, and thermal diffusion coefficients. The package is in two parts. The first is a preprocessor that computes polynomial fits to the temperature dependent parts of the pure species viscosities and binary diffusion coefficients. The coefficients of these fits are passed to a library of subroutines via a linking file. Then, any subroutine from this library may be called to return either pure species properties or multicomponent gas mixture properties. This package uses the chemical kinetics package CHEMKIN, and transport property subroutines are designed to be used in conjunction with the CHEMKIN subroutine library. This package supersedes a previously-written transport property code package in which we used certain mixture averaging rules to compute mixture properties rather than the full multicomponent formulation.

A FORTRAN COMPUTER CODE PACKAGE FOR THE EVALUATION OF GAS-PHASE MULTICOMPONENT TRANSPORT PROPERTIES

I. INTRODUCTION

Characterizing the molecular transport of species, momentum, and energy in a multicomponent gaseous mixture requires the evaluation of diffusion coefficients, viscosities, thermal conductivities, and thermal diffusion coefficients. Although evaluating pure species properties follows standard kinetic theory expressions, one can choose from a range of possibilities for evaluating mixture properties. Moreover, computing the mixture properties can be expensive, and depending on the use of the results, it is often advantageous to make simplifying assumptions to reduce the computational cost.

For most applications, gas mixture properties can be determined from pure species properties via certain approximate mixture averaging rules. Recently, however, we have encountered applications in which the approximate averaging rules are not adaquate. As a result we have undertaken a software project to provide full multicomponent transport properties. This code package is fully compatible with our thermodynamic properties and chemical kinetics package CHEMKIN (Kee, Miller, and Jefferson, 1980) and it supersedes our previous transport package (Kee et al., 1983). The new package provides both the mixture-averaged forms as well as the multicomponent formulations. The multicomponent methods are based on the work of Dixon-Lewis (1968) and the methods for mixture-averaged approach are reported in Warnatz (1982) and Kee et al. (1983).

The multicomponent formulation has several important advantages over the relatively simpler mixture formulas. The first advantage is accuracy. The mixture formulas are only correct asymptotically in some special cases, such as in a binary mixture, or in diffusion of trace amounts of species into a nearly pure species, or systems in which all species except one move with nearly the same diffusion velocity (Bird et al., 1960). A second deficiency of the mixture formulas is that overall mass conservation is not necessarily preserved when solving the species continuity equations. To compensate for this shortcoming one has to apply some *ad hoc* correction procedure (cf., Coffee and Heimerl, 1981; Kee et al., 1983). The multicomponent formulation guarantees mass conservation without any correction factors, which is a clear advantage. The only real deficiency of the multicomponent formulation is its computational expense. Evaluating the ordinary multicomponent diffusion coefficients involves inverting a $K \times K$ matrix, and evaluating the thermal conductivity and thermal diffusion coefficients requires solving a $3K \times 3K$ system of algebraic equations, where K is the number of species.

The structure of the present multicomponent transport package is analogous to that of our previous transport package. That is, polynomial fits are first computed for the temperature-dependent parts of the kinetic theory expressions for pure species viscosities and binary diffusion coefficients. (The pure species thermal conductivities are also fit, but are only used in the mixture-averaged formulation.) The coefficients from the fit are passed to a library of subroutines that can be used to return either mixture-averaged properties or multicomponent properties. This fitting procedure is used so that expensive operations, such as evaluation of collision integrals, need be done only once and not every time a property is needed.

The first task in this document is to review the kinetic theory expressions for the pure species viscosities and the binary diffusion coefficients. Then, we describe how momentum, energy, and species mass fluxes are computed from the velocity, temperature and species gradients and either mixture-averaged or multicomponent transport properties. Having these relationships in mind, the report next describes the procedures to determine multicomponent transport properties from the pure species expressions. The third part of the report describes how to use the software package and how it relates to CHEMKIN and our previous transport package. The following chapter describes each of the multicomponent subroutines that can be called by the package's user. The last chapter lists the data base that we are currently using.

II. THE TRANSPORT EQUATIONS

Pure Species Viscosity and Binary Diffusion Coefficients

The single component viscosities are given by the standard kinetic theory expression (cf., Hirschfelder et al., 1954)

$$\eta_k = \frac{5}{16} \frac{\sqrt{\pi m_k k_B T}}{\pi \sigma_k^2 \Omega^{(2,2)*}}, \tag{1}$$

where σ_k is the Lennard-Jones collision diameter, m_k is the molecular mass, k_B is the Boltzmann constant, and T is the temperature. The collision integral $\Omega^{(2,2)*}$ depends on the reduced temperature given by

$$T_k^* = \frac{k_B T}{\epsilon_k},$$

and the reduced dipole moment given by

$$\delta_k^* = \frac{1}{2} \frac{\mu_k^2}{\epsilon_k \sigma_k^3}. \tag{2}$$

In the above expressions ϵ_k is the Lennard-Jones potential well depth and μ_k is the dipole moment. The collision integral value is determined by a quadradic interpolation of the tables based on Stockmayer potentials given in Monchick and Mason (1961).

The binary diffusion coefficients (cf., Hirschfelder et al., 1954) are given in terms of pressure and temperature as

$$\mathcal{D}_{jk} = \frac{3}{16} \frac{\sqrt{2\pi k_B^3 T^3 / m_{jk}}}{P \pi \sigma_{jk}^2 \Omega^{(1,1)*}}, \tag{3}$$

where m_{jk} is the reduced molecular mass for the (j, k) species pair

$$m_{jk} = \frac{m_j m_k}{m_j + m_k}, \tag{4}$$

and σ_{jk} is the reduced collision diameter. The collision integral $\Omega^{(1,1)*}$ (based on Stockmayer potentials) depends on the reduced temperature, T_{jk}^*, which in turn may depend on the species dipole moments μ_k, and polarizabilities α_k. In computing the reduced quantities, we consider two cases, depending on whether the collision partners are polar or nonpolar. For the case that the partners are either both polar or both nonpolar the following expressions apply:

$$\frac{\epsilon_{jk}}{k_B} = \sqrt{\left(\frac{\epsilon_j}{k_B}\right)\left(\frac{\epsilon_k}{k_B}\right)} \tag{5}$$

$$\sigma_{jk} = \frac{1}{2}(\sigma_j + \sigma_k) \tag{6}$$

$$\mu_{jk}^2 = \mu_j \mu_k. \tag{7}$$

For the case of a polar molecule interacting with a nonpolar molecule:

$$\frac{\epsilon_{np}}{k_B} = \xi^2 \sqrt{\left(\frac{\epsilon_n}{k_B}\right)\left(\frac{\epsilon_p}{k_B}\right)} \tag{8}$$

$$\sigma_{np} = \frac{1}{2}(\sigma_n + \sigma_p)\xi^{-\frac{1}{6}} \tag{9}$$

$$\mu_{np}^2 = 0 \tag{10}$$

where,

$$\xi = 1 + \frac{1}{4}\alpha_n^* \mu_p^* \sqrt{\frac{\epsilon_p}{\epsilon_n}}. \tag{11}$$

In the above equations α_n^* is the reduced polarizability for the nonpolar molecule and μ_p^* is the reduced dipole moment for the polar molecule. The reduced values are given by

$$\alpha_n^* = \frac{\alpha_n}{\sigma_n^3} \tag{12}$$

$$\mu_p^* = \frac{\mu_p}{\sqrt{\epsilon_p \sigma_p^3}}. \tag{13}$$

The table look-up evaluation of the collision integral $\Omega^{(1,1)*}$ depends on the reduced temperature

$$T_{jk}^* = \frac{k_B T}{\epsilon_{jk}}, \tag{14}$$

and the reduced dipole moment,

$$\delta_{jk}^* = \frac{1}{2}\mu_{jk}^{*2}. \tag{15}$$

In our previous transport package we added a second-order correction factor to the binary diffusion coefficients (Marrero and Mason, 1972). However, in the multicomponent case, we specifically need only the first approximation to the diffusion coefficients, and therefore the second-order correction is not made. As a result, the binary diffusion coefficients computed by the two codes are different. For the mixture-averaged diffusion coefficients, the present code is presumably less accurate than the previous one because it lacks the second-order correction. However, we view this as quite acceptable since the new subroutines are now available to compute multicomponent properties when high accuracy is important.

Pure Species Thermal Conductivities

The pure species thermal conductivities are computed only for the purpose of later evaluating mixture-averaged thermal conductivities; the mixture conductivity in the multicomponent case does not depend on the pure species formulas stated in this section. Here we assume the individual species conductivities to be composed of translational, rotational, and vibrational contributions as given by Warnatz (1982),

$$\lambda_k = \frac{\eta_k}{M_k}(f_{\mathtt{trans.}} C_{v,\mathtt{trans.}} + f_{\mathtt{rot.}} C_{v,\mathtt{rot.}} + f_{\mathtt{vib.}} C_{v,\mathtt{vib.}}) \tag{16}$$

where

$$f_{\mathtt{trans.}} = \frac{5}{2}(1 - \frac{2}{\pi}\frac{C_{v,\mathtt{rot.}}}{C_{v,\mathtt{trans.}}}\frac{A}{B}) \tag{17}$$

$$f_{\mathtt{rot.}} = \frac{\rho \mathcal{D}_{kk}}{\eta_k}(1 + \frac{2}{\pi}\frac{A}{B}) \tag{18}$$

$$f_{\mathtt{vib.}} = \frac{\rho \mathcal{D}_{kk}}{\eta_k} \tag{19}$$

and,

$$A = \frac{5}{2} - \frac{\rho \mathcal{D}_{kk}}{\eta_k} \tag{20}$$

$$B = Z_{\mathtt{rot.}} + \frac{2}{\pi}\left(\frac{5}{3}\frac{C_{v,\mathtt{rot.}}}{R} + \frac{\rho \mathcal{D}_{kk}}{\eta_k}\right). \tag{21}$$

The molar heat capacity C_v relationships are different depending on whether or not the molecule is linear or not. In the case of a linear molecule,

$$\frac{C_{v,\mathtt{trans.}}}{R} = \frac{3}{2} \tag{22}$$

$$\frac{C_{v,\mathtt{rot.}}}{R} = 1 \tag{23}$$

$$C_{v,\mathtt{vib.}} = C_v - \frac{5}{2}R. \tag{24}$$

In the above, C_v is the specific heat at constant volume of the molecule and R is the universal gas constant. For the case of a nonlinear molecule,

$$\frac{C_{v,\mathtt{trans.}}}{R} = \frac{3}{2} \tag{25}$$

$$\frac{C_{v,\mathtt{rot.}}}{R} = \frac{3}{2} \tag{26}$$

$$C_{v,\mathtt{vib.}} = C_v - 3R. \tag{27}$$

The translational part of C_v is always the same,

$$C_{v,\mathtt{trans.}} = \frac{3}{2}R. \tag{28}$$

In the case of single atoms (H atoms, for example) there are no internal contributions to C_v, and hence,

$$\lambda_k = \frac{\eta_k}{W_k}(f_{\mathtt{trans.}}\frac{3}{2}R), \tag{29}$$

where $f_{\mathtt{trans.}} = 5/2$. The "self-diffusion" coefficient comes from the following expression,

$$\mathcal{D}_{kk} = \frac{3}{16}\frac{\sqrt{2\pi k_B^3 T^3/m_k}}{P\pi\sigma_k^2\Omega^{(1,1)*}}. \tag{30}$$

The density comes from the equation of state for a perfect gas,

$$\rho = \frac{PM_k}{RT}, \tag{31}$$

with P being the pressure and M_k the species molar mass.

The rotational relaxation collision number is a parameter that we assume is available at 298K (included in the data base). It has a temperature dependence given in an expression by Parker (1959) and Brau and Jonkman (1970),

$$Z_{\mathtt{rot.}}(T) = Z_{\mathtt{rot.}}(298)\frac{F(298)}{F(T)}, \tag{32}$$

where,

$$F(T) = 1 + \frac{\pi^{\frac{3}{2}}}{2}\left(\frac{\epsilon/k_B}{T}\right)^{\frac{1}{2}} + \left(\frac{\pi^2}{4} + 2\right)\left(\frac{\epsilon/k_B}{T}\right) + \pi^{\frac{3}{2}}\left(\frac{\epsilon/k_B}{T}\right)^{\frac{3}{2}}. \tag{33}$$

The Pure-Species Fitting Procedure

To expedite the evaluation of transport properties in a computer code, such as a flame code, we fit the temperature dependent parts of the pure species property expressions. Then, rather than evaluating the complex expressions for the properties, only comparatively simple fits need to be evaluated.

We use a polynomial fit of the logarithm of the property versus the logarithm of the temperature. For the viscosity

$$\ln \eta_k = \sum_{n=1}^{N} a_{n,k} (\ln T)^{n-1}, \tag{34}$$

and for thermal conductivity,

$$\ln \lambda_k = \sum_{n=1}^{N} b_{n,k} (\ln T)^{n-1}. \tag{35}$$

The fits are done for each pair of binary diffusion coefficients in the system.

$$\ln \mathcal{D}_{jk} = \sum_{n=1}^{N} d_{n,jk} (\ln T)^{n-1}. \tag{36}$$

We have used third order polynomial fits (i.e., $N = 4$) in the computer codes and find that the fitting errors are well within one percent. The fitting procedure must be carried out for the particular system of gases that is present in a given problem. Therefore, the fitting can not be done "once and for all," but must be done once at the beginning of each new problem.

The viscosity and conductivity are independent of pressure, but the diffusion coefficients depend inversely on pressure. The diffusion coefficient fits are computed at unit pressure; the later evaluation of a diffusion coefficient is obtained by simply dividing the diffusion coefficient as evaluated from the fit by the actual pressure.

Even though the single component conductivities are fit and passed to the subroutine library they are not used in the computation of multicomponent thermal conductivities; they are used only for the evaluation of the mixture-averaged conductivities.

The Mass, Momentum, and Energy Fluxes

The momentum flux is related to the gas mixture viscosity and the velocities by

$$ = -\eta\left(\nabla v + (\nabla v)^{\dagger}\right) + \left(\frac{2}{3}\eta - \kappa\right)(\nabla \cdot v), \tag{37}$$

where v is the velocity vector, (∇v) is the dyadic product, $(\nabla v)^{\dagger}$ is the transpose of the dyadic product, and is the unit tensor (Bird et al., 1960). In this software package we provide average values for the mixture viscosity η, but we do not provide information on the bulk viscosity κ.

The energy flux is given in terms of the thermal conductivity λ_0 by

$$\mathbf{q} = \sum_{k=1}^{K} \mathbf{j}_k h_k - \lambda_0 \nabla T - \sum_{k=1}^{K} \frac{RT}{M_k X_k} D_k^T \mathbf{d}_k, \tag{38}$$

where,

$$\mathbf{d}_k = \nabla X_k + (X_k - Y_k)\frac{1}{p}\nabla p. \tag{39}$$

The multicomponent species flux is given by

$$\mathbf{j}_k = \rho Y_k \mathbf{V}_k, \tag{40}$$

where Y_k are the mass fractions and the diffusion velocities are given by

$$\mathbf{V}_k = \frac{1}{X_k \overline{M}} \sum_{j \neq k}^{K} M_j D_{kj} \mathbf{d}_j - \frac{D_k^T}{\rho Y_k}\frac{1}{T}\nabla T. \tag{41}$$

The species molar masses are denoted by M_k and the mean molar mass by $\overline{M}$. D_{kj} are the ordinary multicomponent diffusion coefficients, and D_k^T are the thermal diffusion coefficients.

By definition in the mixture-average formulations, the diffusion velocity is related to the species gradients by a Fickian formula as,

$$\mathbf{V}_k = -\frac{1}{X_k} D_{km} \mathbf{d}_k - \frac{D_k^T}{\rho Y_k}\frac{1}{T}\nabla T. \tag{42}$$

The mixture diffusion coefficient for species k is computed as (Bird et al., 1960)

$$D_{km} = \frac{1 - Y_k}{\sum_{j \neq k}^{K} X_j / \mathcal{D}_{jk}}. \tag{43}$$

A potential problem with this expression is that it is not mathematically well-defined in the limit of the mixture becoming a pure species. Even though diffusion itself has no real meaning in the case of a pure species, a computer-program implementation should ensure that the diffusion coefficients behave reasonably and that the code does not "blow up" when the pure species condition is reached. We circumvent these problems by evaluating the diffusion coefficients in the following equivalent way.

$$D_{km} = \frac{\sum_{j \neq k}^{K} X_j M_j}{\overline{M} \sum_{j \neq k}^{K} X_j / \mathcal{D}_{jk}} \tag{44}$$

In this form the roundoff is accumulated in roughly the same way in both the numerator and denominator, and thus the quotient is well-behaved as the pure species limit is approached. However, if the mixture is exactly a pure species, the formula is still undefined.

To overcome this difficulty we always retain a small quantity of each species. In other words, for the purposes of computing mixture diffusion coefficients, we simply do not allow a pure species situation to occur; we always maintain a residual amount of each species. Specifically, we assume in the above formulas that

$$X_k = \hat{X}_k + \delta, \tag{45}$$

where $\hat{X}_k$ is the actual mole fraction and δ is a small number that is numerically insignificant compared to any mole fraction of interest, yet which is large enough that there is no trouble representing it on any computer. A value of 10^{-12} for δ works well.

In some cases (for example, Warnatz, 1978 and Coltrin et al., 1986) it can be useful to treat multicomponent diffusion in terms of an equivalent Fickian diffusion process. This is sometimes a programming convenience in that the computer data structure for the multicomponent process can be made to look like a Fickian process. To do so suppose that a mixture diffusion coefficient can be defined in such a way that the diffusion velocity is written as Eq. (42) rather than Eq. (41). This equivalent Fickian diffusion coefficient is then derived by equating Eq. (41) and (42) and solving for D_{km} as

$$D_{km} = -\frac{\sum_{j \neq k}^{K} M_j D_{kj} \mathbf{d}_j}{\overline{M} \mathbf{d}_k}. \tag{46}$$

Unfortunately, this equation is undefined as the mixture approaches a pure species condition. To help deal with this difficulty a small number ($\epsilon = 10^{-12}$) may be added to both the numerator and denominator to obtain

$$D_{km} = -\frac{\sum_{j \neq k}^{K} M_j D_{kj} \mathbf{d}_j + \epsilon}{\overline{M} \left(\mathbf{d}_k + \epsilon\right)}. \tag{47}$$

Furthermore, for the purposes of evaluating the "multicomponent" D_{km}, it may be advantageous to compute the $\mathbf{d}_k$ in the denominator using the the fact that $\nabla X_k = -\sum_{j\neq k}^{K} \nabla X_j$. In this way the summations in the numerator and the denominator accumulate any rounding errors in roughly the same way, and thus the quotient is more likely to be well behaved as the pure species limit is approached. Since there is no diffusion due to species gradients in a pure species situation, the exact value of the diffusion coefficient is not as important as the need for it simply to be well defined, and thus not cause computational difficulties.

In practice we have found mixed results using the equivalent Fickian diffusion to represent multicomponent processes. In some marching or parabolic problems, such as boundary layer flow in channels (Coltrin et al., 1986), we find that the equivalent Fickian formulation is preferable. However, in some steady state boundary value problems, we have found that the equivalent Fickian formulation fails to converge, whereas the regular multicomponent formulation works quite well. Thus, as of this date, we cannot confidently recommend which formulation should be preferred for any given application.

The Mixture-Averaged Properties

Our objective in this section is to determine mixture properties from the pure species properties. In the case of viscosity, we use the semi-empirical formula due to Wilke (1950) and modified by Bird et al. (1960). The Wilke formula for mixture viscosity is given by

$$\eta = \sum_{k=1}^{K} \frac{X_k \eta_k}{\sum_{j=1}^{K} X_j \Phi_{kj}}, \tag{48}$$

where,

$$\Phi_{kj} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_k}{M_j}\right)^{-\frac{1}{2}} \left(1 + \left(\frac{\eta_k}{\eta_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_k}\right)^{\frac{1}{4}}\right)^2 . \tag{49}$$

For the mixture-averaged thermal conductivity we use a combination averaging formula (Mathur et al., 1967)

$$\lambda = \frac{1}{2}\left(\sum_{k=1}^{K} X_k \lambda_k + \frac{1}{\sum_{k=1}^{K} X_k/\lambda_k}\right) \tag{50}$$

Both of these formulas were used in our previous mixture transport code (Kee et al., 1983).

Thermal Diffusion Ratios

The thermal diffusion coefficients are evaluated in the following section on multicomponent properties. This section describes a relatively inexpensive way to estimate the thermal diffusion of light species into a mixture. This is the method that is used in our previous transport package, and it is included here for the sake of upward compatibility. This approximate method is considerably less accurate than the thermal diffusion coefficients that are computed from the multicomponent formulation.

A thermal diffusion ratio Θ_k can be defined such that the thermal diffusion velocity $\mathcal{W}_{k_i}$ is given by

$$\mathcal{W}_{k_i} = \frac{D_k \Theta_k}{X_k} \frac{1}{T} \frac{\partial T}{\partial x_i} \tag{51}$$

where x_i is a spatial coordinate. The mole fractions are given by X_k, and the D_{km} are mixture diffusion coefficients Eq.(42). In this form we only consider thermal diffusion in the trace, light component limit (specifically, species k having molecular mass less than 5). The thermal diffusion ratio (Chapman and Cowling, 1970) is given by

$$\Theta_k = \sum_{j \neq k}^{K} \theta_{kj} \tag{52}$$

where

$$\theta_{kj} = \frac{15}{2} \frac{(2A^*_{kj} + 5)(6C^*_{kj} - 5)}{A^*_{kj}(16A^*_{kj} - 12B^*_{kj} + 55)} \frac{M_j - M_k}{M_j + M_k} X_j X_k \tag{53}$$

Three ratios of collision integrals are defined by

$$A^*_{ij} = \frac{1}{2} \frac{\Omega_{ij}^{(2,2)}}{\Omega_{ij}^{(1,1)}} \tag{54}$$

$$B^*_{ij} = \frac{1}{3} \frac{5\Omega_{ij}^{(1,2)} - \Omega_{ij}^{(1,3)}}{\Omega_{ij}^{(1,1)}} \tag{55}$$

$$C^*_{ij} = \frac{1}{3} \frac{\Omega_{ij}^{(1,2)}}{\Omega_{ij}^{(1,1)}} \tag{56}$$

We have fit polynomials to tables of A^*_{ij}, B^*_{ij}, and C^*_{ij} (Monchick and Mason, 1961).

In the preprocessor fitting code (where the pure species properties are fit) we also fit the temperature dependent parts of the pairs of the thermal diffusion ratios for each light species into all the other species. That is, we fit $\theta_{kj}/(X_j X_k)$ for all species pairs in which $W_k \leq 5$. Since the θ_{kj} depend weakly on temperature, we fit to polynomials in temperature, rather than the logarithm of temperature. The coefficients of these fits are written onto the linking file.

The Multicomponent Properties

The multicomponent diffusion coefficients, thermal conductivities, and thermal diffusion coefficients are computed from the solution of a system of equations defined by what we call the L matrix. It is convenient to refer to the L matrix in terms of its nine block sub-matricies, and in this form the system is given by

$$\begin{pmatrix} L^{00,00} & L^{00,10} & 0 \\ L^{10,00} & L^{10,10} & L^{10,01} \\ 0 & L^{01,10} & L^{01,01} \end{pmatrix} \begin{pmatrix} a^1_{00} \\ a^1_{10} \\ a^1_{01} \end{pmatrix} = \begin{pmatrix} 0 \\ X \\ X \end{pmatrix} \tag{57}$$

where right hand side vector is composed of the mole fraction vectors X_k. The multicomponent diffusion coefficients are given in terms of the inverse of the $L^{00,00}$ block as

$$D_{i,j} = X_i \frac{16T}{25p} \frac{\overline{m}}{m_j} \left(P_{ij} - P_{ii}\right), \tag{58}$$

where

$$(P) = \left(L^{00,00}\right)^{-1}. \tag{59}$$

The thermal conductivities are given in terms of the solution to the system of equations by

$$\lambda_{0,\mathrm{tr.}} = -\sum_{k=1}^{K} X_k a^1_{k10} \tag{60}$$

$$\lambda_{0,\mathrm{int.}} = -\sum_{k=1}^{K} X_k a^1_{k01} \tag{61}$$

$$\lambda_0 = \lambda_{0,\mathrm{tr.}} + \lambda_{0,\mathrm{int.}} \tag{62}$$

and the thermal diffusion coefficients are given by

$$D^T_k = \frac{8 m_k X_k}{5R} a^1_{k00} \tag{63}$$

The components of the L matrix are given by Dixon-Lewis (1968).

$$L^{00,00}_{ij} = \frac{16T}{25p} \sum_{k=1}^{K} \frac{X_k}{m_i \mathcal{D}_{ik}} \left\{ m_j X_j \left(1 - \delta_{ik}\right) - m_i X_i \left(\delta_{ij} - \delta_{jk}\right) \right\}$$

$$L^{00,10}_{ij} = \frac{8T}{5p} \sum_{k=1}^{K} X_j X_k \left(\delta_{ij} - \delta_{ik}\right) \frac{m_k \left(1.2 C^*_{jk} - 1\right)}{\left(m_j + m_k\right) \mathcal{D}_{jk}}$$

$$L_{ij}^{10,00} = L_{ji}^{00,10}$$

$$L_{ij}^{01,00} = L_{ji}^{00,01} = 0$$

$$\begin{aligned}
L_{ij}^{10,10} &= \frac{16T}{25p}\sum_{k=1}^{K}\frac{m_i}{m_j}\frac{X_iX_k}{(m_i+m_k)^2\,\mathcal{D}_{ik}} \\
&\quad\times\left\{\left(\delta_{jk}-\delta_{ij}\right)\left[\frac{15}{2}m_j^2+\frac{25}{4}m_k^2-3m_k^2B_{ik}^*\right]\right. \\
&\quad\left.-\,4m_jm_kA_{ik}^*\left(\delta_{jk}+\delta_{ij}\right)\left[1+\frac{5}{3\pi}\left(\frac{c_{i,\mathtt{rot.}}}{k_B\xi_{ik}}+\frac{c_{k,\mathtt{rot.}}}{k_B\xi_{ki}}\right)\right]\right\} \qquad (64)\\
L_{ii}^{10,10} &= -\frac{16m_iX_i^2}{R\eta_i}\left(1+\frac{10c_{i,\mathtt{rot.}}}{k_B\xi_{ii}}\right)-\frac{16T}{25p}\sum_{k\neq i}^{K}\frac{X_iX_k}{(m_i+m_k)^2\,\mathcal{D}_{ik}} \\
&\quad\times\left\{\frac{15}{2}m_i^2+\frac{25}{4}m_k^2-3m_k^2B_{ik}^*+4m_im_kA_{ik}^*\right. \\
&\quad\left.\times\left[1+\frac{5}{3\pi}\left(\frac{c_{i,\mathtt{rot.}}}{k_B\xi_{ik}}+\frac{c_{k,\mathtt{rot.}}}{k_B\xi_{ki}}\right)\right]\right\} \\
L_{ij}^{10,01} &= \frac{32T}{5\pi pc_{j,\mathtt{int.}}}\sum_{k=1}^{K}\frac{m_jA_{jk}^*}{(m_j+m_k)\,\mathcal{D}_{jk}}\left(\delta_{ik}+\delta_{ij}\right)X_jX_k\frac{c_{j,\mathtt{rot.}}}{k_B\xi_{jk}} \\
L_{ii}^{10,01} &= \frac{16}{3\pi}\frac{m_iX_i^2k_B}{R\eta_ic_{i,\mathtt{int.}}}\frac{c_{i,\mathtt{rot.}}}{k_B\xi_{ii}} \\
&\quad+\frac{32Tk_B}{5\pi pc_{i,\mathtt{int.}}}\sum_{k\neq i}^{K}\frac{m_iA_{ik}^*}{(m_i+m_k)\,\mathcal{D}_{ik}}X_iX_k\frac{c_{i,\mathtt{rot.}}}{k_B\xi_{ik}} \\
L_{ij}^{01,10} &= L_{ji}^{10,01} \\
L_{ii}^{01,01} &= -\frac{8k_B^2}{\pi c_{i,\mathtt{int.}}^2}\frac{m_iX_i^2}{R\eta_i}\frac{c_{i,\mathtt{rot.}}}{k_B\xi_{ii}} \\
&\quad-\frac{4k_BT}{c_{i,\mathtt{int.}}p}\left\{\sum_{k=1}^{K}\frac{X_iX_k}{\mathcal{D}_{i\ \mathtt{int.},k}}+\sum_{k\neq i}^{K}\frac{12X_iX_k}{5\pi c_{i,int.}}\frac{m_i}{m_k}\frac{A_{ik}^*}{\mathcal{D}_{ik}}\frac{c_{i,\mathtt{rot.}}}{\xi_{ii}}\right\} \\
L_{ij}^{01,01} &= 0\,(i\neq j)
\end{aligned}$$

In these equations T is the temperature, p is the pressure, X_k is the mole fraction of species k, $\mathcal{D}_{ik}$ are the binary diffusion coefficients, and m_i is the molecular mass of species i. Three ratios of collision integrals A_{jk}^*, B_{jk}^*, and C_{jk}^* are defined by Eqs. (54-56). The universal gas constant is represented by R and the pure species viscosities are given as η_k. The rotational and internal parts of the species molecular heat capacities are represented by $c_{k,\mathtt{rot.}}$ and $c_{k,\mathtt{int.}}$. For a linear molecule

$$\frac{c_{k,\mathtt{rot}}}{k_B} = 1, \tag{65}$$

and for a nonlinear molecule

$$\frac{c_{k,\mathtt{rot}}}{k_B} = \frac{3}{2}. \tag{66}$$

The internal component of heat capacity is computed by subtracting the translational part from the full heat capacity as evaluated from the CHEMKIN thermodynamic data base,

$$\frac{c_{k,\mathtt{int}}}{k_B} = \frac{c_p}{k_B} - \frac{3}{2}. \tag{67}$$

Following Dixon-Lewis (1968), we assume that the relaxation collision numbers ξ_{ij} depend only on the species i, i.e., all $\xi_{ij} = \xi_{ii}$. The rotational relaxation collision number at 298K is one of the parameters in the transport data base, and its temperature dependence was given in Eqs. (32) and (33).

For non-polar gases the binary diffusion coefficients for internal energy $\mathcal{D}_{i\ \mathtt{int.},k}$ are approximated by the ordinary binary diffusion coefficients. However, in the case of collisions between polar molecules, where the exchange is energetically resonant, a large correction of the following form is necessary,

$$\mathcal{D}_{p\ \mathtt{int.},p} = \frac{\mathcal{D}_{pp}}{(1 + \delta'_{pp})}, \tag{68}$$

where,

$$\delta'_{pp} = \frac{2985}{\sqrt{T^3}} \tag{69}$$

when the temperature is in Kelvins.

There are some special cases that require modification of the L matrix. First, for mixtures containing monatomic gases, the rows that refer to the monatomic components in the lower block row and the corresponding columns in the last block column must be omitted. That this is required is clear by noting that the internal part of the heat capacity appears in the denominator of terms in these rows and columns (e.g., $L_{ij}^{10,01}$). An additional problem arises as a pure species situation is approached, because all X_k except one approach zero, and this causes the L matrix to become singular. Therefore, for the purposes of forming L we do not allow a pure species situation to occur. We always retain a residual amount of each species by computing the mole fractions from

$$X_k = \frac{\overline{M} Y_k}{M_k} + \delta. \tag{70}$$

A value of $\delta = 10^{-12}$ works well; it is small enough to be numerically insignificant compared to any mole fraction of interest, yet it is large enough to be represented on nearly any computer.

Species Conservation

Some care needs to be taken in using the mixture-averaged diffusion coefficients as described here. The mixture formulas are approximations, and they are not constrained to require that the net species diffusion flux is zero, i.e., the condition,

$$\sum_{k=1}^{K} \mathbf{V}_k Y_k = 0 \tag{71}$$

need not be satisfied. Therefore, one must expect that applying these mixture diffusion relationships in the solution of a system of species conservation equations should lead to some nonconservation, i.e., the resultant mass fractions will not sum to one. Therefore, one of a number of corrective actions must be invoked to ensure mass conservation.

Unfortunately, resolution of the conservation problem requires knowledge of species flux, and hence details of the specific problem and discretization method. Therefore, it is not reasonable in the general setting of the present code package to attempt to enforce conservation. Nevertheless, the user of the package must be aware of the difficulty, and consider its resolution when setting up the difference approximations to his particular system of conservation equations.

One attractive method is to define a "conservation diffusion velocity" as Coffee and Heimerl (1981) recommend. In this approach we assume that the diffusion velocity vector is given as

$$\mathbf{V}_k = \hat{\mathbf{V}}_k + \mathbf{V}_c, \tag{72}$$

where $\hat{\mathbf{V}}_k$ is the ordinary diffusion velocity Eq.(42) and $\mathbf{V}_c$ is a constant correction factor (independent of species, but spatially varying) introduced to satisfy Eq. (71). The correction velocity is defined by

$$\mathbf{V}_c = -\sum_{k=1}^{K} Y_k \hat{\mathbf{V}}_k. \tag{73}$$

This approach is the one followed by Miller et al. (1982, 1983, 1985) in their flame models.

An alternative approach is attractive in problems having one species that is always present in excess. Here, rather than solving a conservation equation for the one excess species, its mass fraction is computed simply by subtracting the sum of the remaining mass fractions from unity. A similar approach involves determining locally at each computational cell which species is in excess. The diffusion velocity for that species is computed to require satisfaction of Eq. (71).

Even though the multicomponent formulation is theoretically forced to conserve mass, the numerical implementations can cause some slight nonconservation. Depending on the numerical method, even slight inconsistencies can lead to difficulties. Methods that do a good job of controlling numerical errors, such as the differential/algebraic equation solver DASSL (Petzold, 1982), are especially sensitive to inconsistencies, and can suffer computational inefficiencies or convergence failures. Therefore, even when the multicomponent formulation is used, it is often advisable to provide corrective measures such as those described above for the mixture-averaged approach. However, the magnitude of any such corrections will be significantly smaller.

III. THE MECHANICS OF USING THE PACKAGE

Using the transport package requires the manipulation of several Fortran programs, libraries and data files. Also, it must be used in conjunction with the chemical kinetics package CHEMKIN. The general flow of information is depicted in Fig. 1.

The first step is to execute the CHEMKIN Interpreter. CHEMKIN is documented separately (Kee et al., 1980), so we only outline its use here. The CHEMKIN Interpreter first reads (Unit 5) user-supplied information about the species and chemical reactions in a problem. It then extracts further information about the species' thermodynamic properties from a data base (Unit 21). This information is stored on the CHEMKIN Linking File (Unit 25), a file that is needed by the transport property fitting code TRANFIT, and later by the CHEMKIN subroutine library.

The next code to be executed is the transport property fitting code, TRANFIT. It needs input from a transport property data base (Unit 31), and from the CHEMKIN Linking File. The transport data base contains molecular parameters for a number of species; these parameters are: the Lennard-Jones well depth ϵ/k_B in Kelvins, the Lennard-Jones collision diameter σ in Angstroms, the dipole moment μ in Debyes, the polarizability α in cubic angstroms, the rotational relaxation collision number, Z_{rot}, and an indicator regarding the nature and geometrical configuration of the molecule. The information coming from the CHEMKIN Linking File contains the species names, and their molar masses and heat capacities. For a given species, the species names in both the CHEMKIN and the TRANFIT data bases must correspond exactly. Like the CHEMKIN Interpreter, the TRANFIT code produces a Linking File (Unit 35) that is later needed in the transport property subroutine library.

Both the CHEMKIN and the transport subroutine libraries must be initialized before

Figure 1. Schematic representing the relationship of the transport package, CHEMKIN, and the user application code.

use and there is a similar initialization subroutine in each. The transport subroutine library is initialized by a call to SUBROUTINE MCINIT. Its purpose is to read the transport Linking File and set up the internal working and storage space that must be made available to all other subroutines in the library. Once initialized, any subroutine in the library may be called from the user's Fortran code.

For compatibility purposes the Linking File for the present code is exactly the same as is used for the previous mixture transport package. The fitting codes are essentially the same, except for the second order corrections to the binary diffusion coefficients, which are not made in the present code. In the case of the subroutines that evaluate multicomponent thermal conductivities and thermal diffusion coefficients, the present code does not use computed fits to pure species thermal conductivities and thermal diffusion ratios.

IV. SUBROUTINE DESCRIPTIONS

This section provides the detailed descriptions of all subroutines in the library. There are eleven user-callable subroutines in the package. All subroutine names begin with MC. The following letter is either an S an A or an M, indicating whether pure species (S), mixture-averaged (A), or multicomponent (M) properties are returned. The remaining letters indicate which property is returned: CON for conductivity, VIS for viscosity, DIF for diffusion coefficients, CDT for both conductivity and thermal diffusion coefficients, and TDR for the thermal diffusion ratios.

A call to the initialization subroutine MCINIT must preceed any other call. This subroutine is normally called only once at the beginning of a problem; it reads the linking file and sets up the internal storage and working space – arrays IMCWRK and RMCWRK. These arrays are required input to all other subroutines in the library. Besides MCINIT there is one other non-property subroutine, called MCPRAM; it is used to return the arrays of molecular parameters that came from the data base for the species in the problem. All other subroutines are used to compute either viscosities, thermal conductivities, or diffusion coefficients. They may be called to return pure species properties, mixture-averaged properties, or multicomponent properties.

In the input to all subroutines, the state of the gas is specified by the pressure in dynes per square centimeter, temperature in Kelvins, and the species mole fractions. (Note: The previous package, Kee et al., 1983, used mass fractions as input.) The properties are returned in standard CGS units. The order of vector information, such as the vector of mole fractions or pure species viscosities, is the same as the order declared in the Chemkin Interpreter input.

We first provide a short description of each subroutine according to its function. Then, a longer description of each subroutine, listed in alphabetical order, follows.

Initialization and Parameters

SUBROUTINE MCINIT (LINKMC, LOUT, LENIMC, LENRMC, IMCWRK, RMCWRK)

This subroutine serves to read the linking file from the fitting code and to create the internal storage and work arrays, IMCWRK(*) and RMCWRK(*). MCINIT must be called before any other transport subroutine is called. It must be called after the Chemkin package is initialized.

SUBROUTINE MCPRAM (IMCWRK, RMCWRK, EPS, SIG, DIP, POL, ZROT, NLIN)

This subroutine is called to return the arrays of molecular parameters as read from the transport data base.

Viscosity

SUBROUTINE MCSVIS (T, RMCWRK, VIS)

This subroutine computes the array of pure species viscosities given the temperature.

SUBROUTINE MCAVIS (T, X, RMCWRK, VISMIX)

This subroutine computes the mixture viscosity given the temperature and the species mole fractions. It uses modifications of the Wilke semi-empirical formulas.

Conductivity

SUBROUTINE MCSCON (T, RMCWRK, CON)

This subroutine computes the array of pure species conductivities given the temperature.

SUBROUTINE MCACON (T, X, RMCWRK, CONMIX)

This subroutine computes the mixture thermal conductivity given the temperature and the species mole fractions.

SUBROUTINE MCMCDT (P, T, X, KDIM, IMCWRK, RMCWRK, ICKWRK, CKWRK, DT, COND)

This subroutine computes the thermal diffusion coefficients and mixture thermal conductivities given the pressure, temperature, and mole fractions.

Diffusion Coefficients

SUBROUTINE MCSDIF (P, T, KDIM, RMCWRK, DJK)

This subroutine computes the binary diffusion coefficients given the pressure and temperature.

SUBROUTINE MCADIF (P, T, X, RMCWRK, D)

This subroutine computes mixture-averaged diffusion coefficients given the pressure, temperature, and species mass fractions.

SUBROUTINE MCMDIF (P, T, X, KDIM, IMCWRK, RMCWRK, D)

This subroutine computes the ordinary multicomponent diffusion coefficients given the pressure, temperature, and mole fractions.

Thermal Diffusion

SUBROUTINE MCATDR (T, X, IMCWRK, RMCWRK, TDR)

This subroutine computes the thermal diffusion ratios for the light species into the mixture.

SUBROUTINE MCMCDT (P, T, X, KDIM, IMCWRK, RMCWRK, ICKWRK, CKWRK, DT, COND)

This subroutine computes the thermal diffusion coefficients, and mixture thermal conductivities given the pressure, temperature, and mole fractions.

Detailed Subroutine Descriptions

The following pages list detailed descriptions for the user interface to each of the package's eleven user-callable subroutines. They are listed in alphebetical order.

V. TRANSPORT DATA BASE

In this section we list the data base that we currently use. New species are easily added and as new or better data becomes available, we expect that users will change their versions of the data base to suit their own needs. This data base should not be viewed as the last word in transport properties. Instead, it is a good starting point from which a user will provide the best available data for his particular application. However, when adding a new species to the data base, be sure that the species name is exactly the same as it is in the CHEMKIN thermodynamic data base.

The numbers in the data base have been determined by computing "best fits" to experimental measurements of some transport property (e.g. viscosity). The procedure has been used and tested successfully by Warnatz in his flame models, and he has generated most of the parameters in this data base. We note also that the Lennard-Jones parameters may be estimated following the methods outlined in Svehla (1962).

The first 15 columns in each line of the data base are reserved for the species name, and the first character of the name must begin in column 1. (Presently CHEMKIN is programmed to allow no more than 10-character names.) Columns 16 through 80 are unformatted, and they contain the molecular parameters for each species. They are, in order:

1. An index indicating whether the molecule has a monatomic, linear or nonlinear geometrical configuration. If the index is 0, the molecule is a single atom. If the index is 1 the molecule is linear, and if it is 2, the molecule is nonlinear.
2. The Lennard-Jones potential well depth ϵ/k_B in Kelvins.
3. The Lennard-Jones collision diameter σ in Angstroms.
4. The dipole moment μ in Debye. Note: a Debye is $10^{-18}\mathrm{cm}^{3/2}\mathrm{erg}^{1/2}$.
5. The polarizability α in cubic Angstroms.
6. The rotational relaxation collision number Z_{rot} at 298K.
7. After the last number, a comment field can be enclosed in parenthesis.

REFERENCES

Bird, R. B., Stewart, W. E. and Lightfoot, E. N., (1960), *Transport Phenomena*, John Wiley and Sons, New York, p. 258.

Brau, C. C., and Jonkman, R. M., (1970), "Classical Theory of Rotational Relaxation in Diatomic Gases," *J. Chem. Phys.*, **52**, p. 477.

Chapman, S. and Cowling, T. G., (1970), The Mathematical Theory of Non-Uniform Gases, Third Edition, Cambridge University Press, Cambridge.

Coffee, T. P. and Heimerl, J. M., (1981), "Transport Algorithms for Premixed Laminar, Steady-State Flames," *Comb. Flame*, **43**, p. 273.

Coltrin, M. E., Kee, R. J., Miller, J. A., (1986), "A Mathematical Model of Silicon Chemical Vapor Deposition: Further Refinements and the Effects of Thermal Diffusion," *J. Electrochem. Soc.* **133** no. 6, 1206-1213.

Curtiss, C. F. and Hirschfelder, J. O., (1949), "Transport Properties of Multicomponent Gas Mixtures," *J. Chem. Phys.*, **17**, p. 550.

Dixon-Lewis, G. (1968) "Flame Structure and Flame Reaction Kinetics, II. Transport Phenomena in Multicomponent Systems," *Proc. Roy. Soc. A* **307** 111-135.

Dixon-Lewis, G. (1984) "Computer Modeling of Combustion Reactions in Flowing Systems with Transport," *Combustion Chemistry* W. C. Gardiner, Ed., Springer-Verlag, New York.

Hirschfelder, J. O., Curtiss, C. F., and Bird, R. B., (1954), "Molecular Theory of Gases and Liquids," John Wiley and Sons, Inc., New York.

Kee, R. J., Miller, J. A. and Jefferson, T. H., (1980), "CHEMKIN: A General-Purpose, Problem-Independent, Transportable, Fortran Chemical Kinetics Code Package," Sandia National Laboratories Report, SAND80-8003.

Kee, R. J., Warnatz, J, and Miller, J. A., (1983), "A FORTRAN Computer Code Package for the Evaluation of Gas-Phase Viscosities, Conductivities, and Diffusion Coefficients," Sandia National Laboratories Report, SAND83-8209.

Marrero T. R. and Mason, E. A., (1972), "Gaseous Diffusion Coefficients," *J. Phys. Chem. Ref. Dat.*, **1**, p. 3.

Mathur, S., Tondon, P. K., and Saxena, S. C., (1967), "Thermal Conductivity of Binary, Ternary and Quaternary Mixtures of Rare Gases," *Mol. Phys.*, **12**, p. 569.

Miller, J. A., Mitchell, R. E., Smooke, M. D., and Kee, R. J., (1982), "Toward A Comprehensive Chemical Kinetic Mechanism for the Oxidation of Acetylene: Comparison of Model Predictions with Results from Flame and Shock Tube Experiments," *Nineteenth Symposium (International) on Combustion*, The Combustion Institute, Pittsburgh, PA, p. 181.

Miller, J. A., Smooke, M. D., Green, R. M., and Kee, R. J., (1983), "Kinetic Modeling of the Oxidation of Ammonia in Flames," *Comb. Sci. Tech.*, **34**, p. 149-176.

Miller, J. A., Branch, M. C., McLean, W. J., Chandler, D. W., Smooke, M. D., and Kee, R. J., (1985), "On the Conversion of HCN to NO and N_2 in H_2-O_2-HCN-Ar Flames at Low Pressure," *Twentieth Symposium (International) on Combustion*, The Combustion Institute, Pittsburgh, PA, p. 673.

Monchick, L. and Mason, E. A., (1961), "Transport Properties of Polar Gases", *J. Chem. Phys.*, **35**, p. 1676.

Parker, J. G., (1959), "Rotational and Vibrational Relaxation in Diatomic Gases," *Phys. of Fluids*, **2**, p. 449.

Petzold, L. R., (1982), "A Description of DASSL," Sandia National Laboratories Report SAND82-8637.

Svehla, R. A., (1962), "Estimated Viscosities and Thermal Conductivities of Gases at High Temperatures," NASA Technical Report R-132.

Warnatz, J., (1978), "Calculation of the Structure of Laminar Flat Flames I: Flame Velocity of Freely Propagating Ozone Decomposition Flames," *Ber. Bunsenges. Phys. Chem.* **82** 193.

Warnatz, J., (1982), "Influence of Transport Models and Boundary Conditions on Flame Structure," Numerical Methods in Flame Propagation, Eds. N. Peters and J. Warnatz, Friedr. Vieweg and Sohn, Wiesbaden.

Wilke, C. R., (1950), "A Viscosity Equation for Gas Mixtures," *J. Chem. Phys.*, **18**, 517.