September 12, 2003 9:18 am

Release
3.7.1

REACTION DESIGN

TRANSPORT Core Utility Manual

Pre-processor and subroutine library for the analysis of gas-phase transport properties

CHEMKIN Collection

RD02371-C02-001-001A
September 2003—Revision A

Licensing:

For licensing information, please contact Reaction Design at **(858) 550-1920** (USA) or **CHEMKIN@ReactionDesign.com**.

Technical Support:

Reaction Design provides an allotment of technical support to its Licensees free of charge. To request technical support, please include your license number along with input or output files, and any error messages pertaining to your question or problem. Requests may be directed in the following manner: E-Mail: **Support@ReactionDesign.com**, Fax: **(858) 550-1925**, Phone: **(858) 550-1920**.
Technical support may also be purchased. Please contact Reaction Design for the technical support hourly rates at Support@ReactionDesign.com or (858) 550-1920 (USA).

Copyright:

Trademark:

AURORA, CHEMKIN, The CHEMKIN Collection, CONP, CRESLAF, EQUIL, EQUILIB, OPPDIF, OVEND, PASR, PLUG, PREMIX, Reaction Design, SENKIN, SHOCK, SPIN, SURFACE CHEMKIN, SURFTHERM, TRANSPORT, TWAFER, TWOPNT are all trademarks of Reaction Design or Sandia National Laboratories.

Limitation of Warranty:

The software is provided “as is” by Reaction Design, without warranty of any kind including without limitation, any warranty against infringement of third party property rights, fitness or merchantability, or fitness for a particular purpose, even if Reaction Design has been informed of such purpose. Furthermore, Reaction Design does not warrant, guarantee, or make any representations regarding the use or the results of the use, of the software or documentation in terms of correctness, accuracy, reliability or otherwise. No agent of Reaction Design is authorized to alter or exceed the warranty obligations of Reaction Design as set forth herein. Any liability of Reaction Design, its officers, agents or employees with respect to the software or the performance thereof under any warranty, contract, negligence, strict liability, vicarious liability or other theory will be limited exclusively to product replacement or, if replacement is inadequate as a remedy or in Reaction Design’s opinion impractical, to a credit of amounts paid to Reaction Design for the license of the software.

Literature Citation for TRANSPORT:

The TRANSPORT Pre-processor and Subroutine Library are part of the CHEMKIN Collection.
R. J. Kee, F. M. Rupley, J. A. Miller, M. E. Coltrin, J. F. Grcar, E. Meeks, H. K. Moffat, A. E. Lutz, G. Dixon-Lewis, M. D. Smooke, J. Warnatz, G. H. Evans, R. S. Larson, R. E. Mitchell, L. R. Petzold, W. C. Reynolds, M. Caracotsios, W. E. Stewart, P. Glarborg, C. Wang, O. Adigun, W. G. Houf, C. P. Chou, and S. F. Miller, Chemkin Collection, Release 3.7.1, Reaction Design, Inc., San Diego, CA (2003).

Acknowledgements:

This document is based on the Sandia National Laboratories Report SAND86-8246B, authored by R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, J. A. Miller, H. K. Moffat.[1]

TRANSPORT

Pre-processor and subroutine library for the analysis of gas-phase transport properties

Abstract

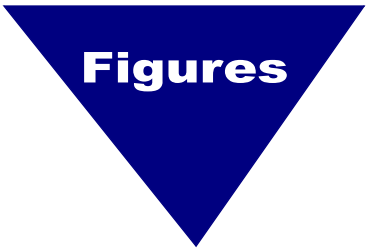
TRANSPORT is a software package that is used for the evaluation of gas-phase multicomponent viscosities, thermal conductivities, diffusion coefficients, and thermal diffusion coefficients for CHEMKIN Applications. The TRANSPORT package consists of two parts. The first is a Pre-processor that computes 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 the TRANSPORT Subroutine Library may be called from a CHEMKIN Application to return either pure species properties or multicomponent gas mixture properties. The TRANSPORT package interfaces with the GAS-PHASE KINETICS Subroutine Library.

Contents

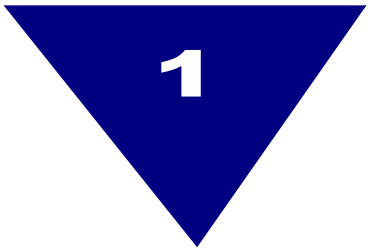
Table of Contents

Tables

List of Tables



List of Figures



1 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 evaluation of 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. However, there are some applications in which the approximate averaging rules are not adequate. The software package described here therefore addresses both the mixture-averaged approach and the full multicomponent approach to transport properties. The TRANSPORT package is part of the CHEMKIN Collection and is designed for use with the CHEMKIN Thermodynamic Database and the GAS-PHASE KINETICS package. The multicomponent methods are based on the work of Dixon-Lewis[2] and the methods for mixture-averaged approach are reported in Warnatz[3] and Kee, *et al*.[4]

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.[5] 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.[4, 6] 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.

To maximize computational efficiency, TRANSPORT is structured to do a large portion of the calculations in a Pre-processor that provides information to the CHEMKIN Application through a Linking File. Polynomial fits are thus computed *a priori* 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 subroutines in the TRANSPORT Subroutine Library, which can be used to return either mixture-averaged properties or multicomponent properties. With this fitting procedure, expensive operations, such as evaluation of collision integrals, are only done once and not every time a property value is needed.

1.1 Organization of this Manual

Chapter 2 of this manual first reviews the kinetic theory expressions for the pure species viscosities and the binary diffusion coefficients. *Section 2.4* describes how momentum, energy, and species mass fluxes are computed from the velocity, temperature and species gradients and either mixture-averaged or multicomponent transport properties. Based on these concepts, *Section 2.5* through *Section 2.7* describe the procedures to determine multicomponent transport properties from the pure species expressions. *Chapter 3* outlines how to use the TRANSPORT package and how it relates to the GAS-PHASE KINETICS package. *Chapter 4* and *Chapter 5* include details on each of the multicomponent subroutines that can be called by a CHEMKIN Application. Finally, *Chapter 6* lists the database that is currently provided with the TRANSPORT package.

1.2 Nomenclature

A global list of nomenclature used in the CHEMKIN Collection is available in the GAS-PHASE KINETICS User Manual.

2

2 The TRANSPORT Equations

2.1 Pure Species Viscosity and Binary Diffusion Coefficients

The single component viscosities are given by the standard kinetic theory expression,[7]

Equation 2-1

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

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

Equation 2-2

$$T_k^* = \frac{k_B T}{\varepsilon_k}$$

and the reduced dipole moment, given by

Equation 2-3

$$\delta_k^* = \frac{1}{2}\frac{\mu_k^2}{\varepsilon_k \sigma_k^3}$$

In the above expression ε_k is the Lennard-Jones potential well depth and μ_k is the dipole moment. The collision integral value is determined by a quadratic interpolation of the tables based on Stockmayer potentials given in Monchick and Mason.[8]

The binary diffusion coefficients[7] are given in terms of pressure and temperature as

Equation 2-4

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

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

Equation 2-5

$$m_{jk} = \frac{m_j m_k}{m_j + m_k}$$

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:

Equation 2-6

$$\frac{\varepsilon_{jk}}{k_B} = \sqrt{\left(\frac{\varepsilon_j}{k_B}\right)\left(\frac{\varepsilon_k}{k_B}\right)}$$

Equation 2-7

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

Equation 2-8

$$\mu_{jk}^2 = \mu_j\mu_k$$

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

Equation 2-9

$$\frac{\varepsilon_{np}}{k_B} = \xi^2\sqrt{\left(\frac{\varepsilon_n}{k_B}\right)\left(\frac{\varepsilon_p}{k_B}\right)}$$

Equation 2-10

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

Equation 2-11

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

where,

Equation 2-12

$$\xi = 1 + \frac{1}{4}\alpha_n^* \mu_p^* \sqrt{\frac{\varepsilon_p}{\varepsilon_n}}$$

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

Equation 2-13

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

Equation 2-14

$$\mu_p^* = \frac{\mu_p}{\sqrt{\varepsilon_p \sigma_p^3}}$$

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

Equation 2-15

$$T_{jk}^* = \frac{k_B T}{\varepsilon_{jk}}$$

and the reduced dipole moment,

Equation 2-16

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

Although one could add a second-order correction factor to the binary diffusion coefficients[9] we have chosen to neglect this since, in the multicomponent case, we specifically need only the first approximation to the diffusion coefficients. When higher accuracy is required for the diffusion coefficients, we therefore recommend using the full multicomponent option.

2.2 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 formula stated in this section. Here we assume the individual species conductivities to be composed of translational, rotational, and vibrational contributions as given by Warnatz.[3]

Equation 2-17

$$\lambda_k = \frac{\eta_k}{W_k}(f_{\text{trans.}} C_{\upsilon,\,\text{trans.}} + f_{\text{rot.}} C_{\upsilon,\,\text{rot.}} + f_{\text{vib.}} C_{\upsilon,\,\text{vib.}})$$

where

Equation 2-18

$$f_{\text{trans.}} = \frac{5}{2}\left(1 - \frac{2}{\pi}\frac{C_{\upsilon,\,\text{rot.}}}{C_{\upsilon,\,\text{trans.}}}\frac{A}{B}\right)$$

Equation 2-19

$$f_{\text{rot.}} = \frac{\rho \mathcal{D}_{kk}}{\eta_k}\left(1 + \frac{2}{\pi}\frac{A}{B}\right)$$

Equation 2-20

$$f_{\text{vib.}} = \frac{\rho \mathcal{D}_{kk}}{\eta_k}$$

and,

Equation 2-21

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

Equation 2-22

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

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

Equation 2-23

$$\frac{C_{\upsilon,\,\text{trans.}}}{R} = \frac{3}{2}$$

Equation 2-24

$$\frac{C_{\upsilon,\,\text{rot.}}}{R} = 1$$

Equation 2-25

$$C_{\upsilon,\,\text{vib.}} = C_\upsilon - \frac{5}{2}R$$

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

Equation 2-26

$$\frac{C_{\upsilon,\,\text{trans.}}}{R} = \frac{3}{2}$$

Equation 2-27

$$\frac{C_{\upsilon,\,\text{rot.}}}{R} = \frac{3}{2}$$

Equation 2-28

$$C_{\upsilon,\,\text{vib.}} = C_\upsilon - 3R$$

The translational part of C_υ is always the same,

Equation 2-29

$$C_{\upsilon,\,\text{trans.}} = \frac{3}{2}R$$

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

Equation 2-30

$$\lambda_k = \frac{\eta_k}{W_k}\left(f_{\text{trans.}}\frac{3}{2}R\right)$$

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

Equation 2-31

$$\mathcal{D}_{kk} = \frac{3}{8}\frac{\sqrt{\pi k_B^3 T^3/m_k}}{P\pi\sigma_k^2\Omega^{(1,1)*}}$$

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

Equation 2-32

$$\rho = \frac{PW_k}{RT}$$

with P being the pressure and W_k the species molecular weight.

The rotational relaxation collision number is a parameter that we assume is available at 298K (included in the database). It has a temperature dependence given in an expression by Parker[10] and Brau and Jonkman,[11]

Equation 2-33

$$Z_{\text{rot.}}(T) = Z_{\text{rot.}}(298)\frac{F(298)}{F(T)}$$

where,

Equation 2-34

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

2.3 The Pure-Species Fitting Procedure

To expedite the evaluation of transport properties in a CHEMKIN Application, such as the PREMIX Application, 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

Equation 2-35

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

and the thermal conductivity,

Equation 2-36

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

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

Equation 2-37

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

By default TRANSPORT uses third-order polynomial fits (i.e., $N = 4$) and we 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 cannot 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 TRANSPORT 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.

2.4 The Mass, Momentum, and Energy Fluxes

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

Equation 2-38

$$\tau = -\eta(\nabla\upsilon + (\nabla\upsilon)^{\mathrm{T}}) + \left(\frac{2}{3}\eta - \kappa\right)(\nabla\cdot\upsilon)\hat{\delta}$$

where υ is the velocity vector, $\nabla\upsilon$ is the dyadic product, $(\nabla\upsilon)^{\mathrm{T}}$ is the transpose of the dyadic product, and $\hat{\delta}$ is the unit tensor.[5] In the TRANSPORT 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

Equation 2-39

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

where,

Equation 2-40

$$\mathbf{d}_k = \nabla X_k + (X_k - Y_k)\frac{1}{P}\nabla P$$

The multicomponent species flux is given by

Equation 2-41

$$\mathbf{j}_k = \rho Y_k V_k$$

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

Equation 2-42

$$\mathbf{V}_k = \frac{1}{X_k \overline{W}} \sum_{j \neq k}^{K} W_j D_{k,j} \mathbf{d}_j - \frac{D_k^T}{\rho Y_k}\frac{1}{T}\nabla T$$

The species molar masses are denoted by W_k and the mean molar mass by $\overline{W}$. $D_{k,j}$ 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,

Equation 2-43

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

The mixture diffusion coefficient for species k is computed as[5]

Equation 2-44

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

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, the numerical implementation must ensure that the diffusion coefficients behave reasonably and that the program 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.

Equation 2-45

$$D_{km} = \frac{\sum_{j \neq k}^{K} X_j W_j}{\overline{W} \sum_{j \neq k}^{K} X_j / \mathcal{D}_{jk}}$$

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

Equation 2-46

$$X_k = \hat{X}_k + \varepsilon$$

where $\hat{X}_k$ is the actual mole fraction and ε 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 ε works well.

In some cases (for example, Warnatz[12] and Coltrin, *et al*.[13]) 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 supposes that a mixture diffusion coefficient can be defined in such a way that the diffusion velocity is written as *Equation 2-43* rather than *Equation 2-42*. This equivalent Fickian diffusion coefficient is then derived by equating *Equation 2-42* and *Equation 2-43* and solving for D_{km} as

Equation 2-47

$$D_{km} = -\frac{\sum_{j \neq k}^{K} W_j D_{k,j} \mathbf{d}_j}{\overline{W} \mathbf{d}_k}$$

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

Equation 2-48

$$D_{km} = -\frac{\sum_{j \neq k}^{K} W_j D_{kj} \mathbf{d}_j + \varepsilon}{\overline{W}(\mathbf{d}_k + \varepsilon)}$$

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 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 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,[13] 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, we cannot confidently recommend which formulation should be preferred for any given application.

2.5 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[14] and modified by Bird, *et al*.[5] The Wilke formula for mixture viscosity is given by

Equation 2-49

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

where

Equation 2-50

$$\Phi_{kj} = \frac{1}{\sqrt{8}}\left(1 + \frac{W_k}{W_j}\right)^{-\frac{1}{2}}\left(1 + \left(\frac{\eta_k}{\eta_j}\right)^{\frac{1}{2}}\left(\frac{W_j}{W_k}\right)^{\frac{1}{4}}\right)^2$$

for the mixture-averaged thermal conductivity we use a combination averaging formula[15]

Equation 2-51

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

2.6 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 method is included here for the sake of backward compatibility. However, this approximate method is considerably less accurate than the thermal diffusion coefficients that are computed from the multicomponent formulation. It is therefore recommended that users employ the multicomponent formulation when thermal diffusion is likely to be important.

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

Equation 2-52

$$\mathcal{W}_{ki} = -\frac{D_{km}\Theta_k}{X_k}\frac{1}{T}\frac{\partial T}{\partial x_i}$$

where x_i is a spatial coordinate. The mole fractions are given by X_k, and the D_{km} are mixture diffusion coefficients *Equation 2-43*. 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[16] is given by

Equation 2-53

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

where

Equation 2-54

$$\theta_{kj} = \frac{15}{2}\frac{(2A_{kj}^* + 5)(6C_{kj}^* - 5)}{A_{kj}^*(16A_{kj}^* - 12B_{kj}^* + 55)}\frac{W_k - W_j}{W_k + W_j}X_k X_j$$

Three ratios of collision integrals are defined by

Equation 2-55

$$A_{ij}^{*} = \frac{1}{2}\frac{\Omega_{ij}^{(2,2)}}{\Omega_{ij}^{(1,1)}}$$

Equation 2-56

$$B_{ij}^{*} = \frac{1}{3}\frac{5\Omega_{ij}^{(1,2)} - \Omega_{ij}^{(1,3)}}{\Omega_{ij}^{(1,1)}}$$

Equation 2-57

$$C_{ij}^{*} = \frac{1}{3}\frac{\Omega_{ij}^{(1,2)}}{\Omega_{ij}^{(1,1)}}$$

We have fit polynomials to tables of A_{ij}^{*}, B_{ij}^{*}, and C_{ij}^{*}.[8]

In the TRANSPORT Pre-processor (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 diffusing into all the other species. That is, we fit $\theta_{kj}/(X_j X_k)$ for all species pairs in which $W_k < 5$. 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 TRANSPORT Linking File.

2.7 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-matrices, and in this form the system is given by

Equation 2-58

$$\begin{pmatrix} L^{00,00} & L^{00,10} & 0 \\ L^{10,00} & L^{10,10} & L^{10,01} \\ 0 & L^{01,10} & L^{01,01} \end{pmatrix} \begin{pmatrix} a_{00}^{1} \\ a_{10}^{1} \\ a_{01}^{1} \end{pmatrix} = \begin{pmatrix} 0 \\ X \\ X \end{pmatrix}$$

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

Equation 2-59

$$D_{i,j} = X_i \frac{16T}{25P}\frac{\overline{W}}{W_j}(p_{ij} - p_{ii})$$

where

Equation 2-60

$$(p) = (L^{00,\,00^{-1}})$$

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

Equation 2-61

$$\lambda_{0,\,\mathrm{tr.}} = -4\sum_{k=1}^{K} X_k a_{k10}^{1}$$

Equation 2-62

$$\lambda_{0,\,\mathrm{int.}} = -4\sum_{k=1}^{K} X_k a_{k01}^{1}$$

Equation 2-63

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

and the thermal diffusion coefficients are given by

Equation 2-64

$$D_k^{\,T} = \frac{8 m_k X_k}{5R} a_{k00}^{1}$$

The components of the L matrix are given by Dixon-Lewis,[2]

Equation 2-65

$$L_{ij}^{00,\,00} = \frac{16T}{25P}\sum_{k=1}^{K}\frac{X_k}{m_i\mathcal{D}_{ik}}\{m_jX_j(1-\delta_{ik}) - m_iX_j(\delta_{ij}-\delta_{jk})\}$$

Equation 2-66

$$L_{ij}^{00,\,10} = \frac{8T}{5P}\sum_{k=1}^{K}X_jX_k(\delta_{ij}-\delta_{ik})\frac{m_k(1.2C_{jk}^{*}-1)}{(m_j+m_k)\mathcal{D}_{jk}}$$

Equation 2-67

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

Equation 2-68

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

Equation 2-69

$$L_{ij}^{10,\,10} = \frac{16T}{25P}\sum_{k=1}^{K}\frac{m_i}{m_j}\frac{X_iX_k}{(m_i+m_k)^2\mathcal{D}_{jk}}\times\left\{(\delta_{jk}-\delta_{ij})\left[\frac{15}{2}m_j^2+\frac{25}{4}m_k^2-3m_k^2B_{ik}^{*}\right]\right.$$
$$\left.-4m_im_kA_{ik}^{*}(\delta_{jk}+\delta_{ij})\left[1+\frac{5}{3\pi}\left(\frac{c_{i,\,\mathrm{rot.}}}{k_B\xi_{ik}}+\frac{c_{k,\,\mathrm{rot.}}}{k_B\xi_{ki}}\right)\right]\right\}$$

Equation 2-70

$$L_{ii}^{10,\,10} = -\frac{16m_iX_i^2}{R\eta_i}\left(1+\frac{10c_{i,\,\mathrm{rot.}}}{k_B\xi_{ii}}\right) - \frac{16T}{25P}\sum_{k\neq i}^{K}\frac{X_iX_k}{(m_i+m_k)^2\mathcal{D}_{ik}}\times$$
$$\left\{\frac{15}{2}m_i^2+\frac{25}{4}m_k^2-3m_k^2B_{ik}^{*}+4m_im_kA_{ik}^{*}\times\left[1+\frac{5}{3\pi}\left(\frac{c_{i,\,\mathrm{rot.}}}{k_B\xi_{ik}}+\frac{c_{k,\,\mathrm{rot.}}}{k_B\xi_{ki}}\right)\right]\right\}$$

Equation 2-71

$$L_{ij}^{10,\,01} = \frac{32T}{5\pi Pc_{j,\,\mathrm{int.}}}\sum_{k=1}^{K}\frac{m_jA_{jk}^{*}}{(m_j+m_k)\mathcal{D}_{jk}}(\delta_{ik}+\delta_{ij})X_jX_k\frac{c_{j,\,\mathrm{rot.}}}{k_B\xi_{jk}}$$

Equation 2-72

$$L_{ii}^{10,01} = \frac{16}{3\pi R}\frac{m_i X_i^2 k_B}{\eta_i c_{i,\text{int.}}}\frac{c_{i,\text{int.}}}{k_B \xi_{ii}} + \frac{32 T k_B}{5\pi P c_{i,\text{int.}}}\sum_{k\neq i}^{K}\frac{m_i A_{ik}^*}{(m_i + m_k)\mathcal{D}_{ik}} X_i X_k \frac{c_{i,\text{rot.}}}{k_B \xi_{ik}}$$

Equation 2-73

$$L_{ij}^{01,10} = L_{ji}^{10,01}$$

Equation 2-74

$$L_{ii}^{01,10} = -\frac{8k_B^k}{\pi c_{i,\text{int.}}^2}\frac{m_i X_i^2}{R\eta_i}\frac{c_{i,\text{rot.}}}{k_B \xi_{ii}} - \frac{4k_B T}{c_{i,\text{int.}} P}\left\{\sum_{k=1}^{K}\frac{X_i X_k}{D_{i\text{int.},k}} + \sum_{k\neq i}^{K}\frac{12 X_i X_k}{5\pi c_{i\text{int.}}}\frac{m_i}{m_k}\frac{A_{ik}^*}{\mathcal{D}_{ik}}\frac{c_{i\text{rot.}}}{\xi_{ii}}\right\}$$

Equation 2-75

$$L_{ij}^{01,01} = 0 (i \neq j)$$

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 *Equation 2-55* through *Equation 2-57*. 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 $c_{k,\text{rot.}}$ and $c_{k,\text{int.}}$. For a linear molecule

Equation 2-76

$$\frac{c_{k,\text{rot.}}}{k_B} = 1$$

and for a nonlinear molecule

Equation 2-77

$$\frac{c_{k,\text{rot.}}}{k_B} = \frac{3}{2}$$

The internal component of heat capacity is computed by subtracting the translational part from the full heat capacity as evaluated from the CHEMKIN Thermodynamic Database.

Equation 2-78

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

Following Dixon-Lewis,[2] 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 database, and its temperature dependence was given in *Equation 2-33* and *Equation 2-34*.

For non-polar gases the binary diffusion coefficients for internal energy $\mathcal{D}_{i\mathrm{int.},\,k}$ 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,

Equation 2-79

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

Equation 2-80

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

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. This is apparent 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

Equation 2-81

$$X_k = \frac{\overline{W} Y_k}{W_k} + \varepsilon$$

A value of $\varepsilon = 10^{-12}$ 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.

2.8 Species Conservation

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

Equation 2-82

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

is not automatically satisfied. Therefore, applying these mixture diffusion relationships in the solution of a system of species conservation equations may 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. These corrections are implemented within a CHEMKIN Application.

One attractive method is to define a “conservation diffusion velocity” as Coffee and Heimerl[6] recommend. In this approach we assume that the diffusion velocity vector is given as

Equation 2-83

$$\mathbf{V}_k = \hat{\mathbf{V}}_k + \mathbf{V}_c$$

where $\hat{\mathbf{V}}_k$ is the ordinary diffusion velocity *Equation 2-43* and $\mathbf{V}_c$ is a constant correction factor (independent of species, but spatially varying) introduced to satisfy *Equation 2-82*. The correction velocity is defined by

Equation 2-84

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

This approach is the one used in OPPDIF, for example.

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 *Equation 2-82*. The PREMIX Application includes both this trace-species approach and the correction-velocity approach as user options.

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.

3

3 The Mechanics of Using the Package

3.1 General Flow of Information

The TRANSPORT package must be used in conjunction with the GAS-PHASE KINETICS package. The general flow of information is depicted in *Figure 3-1*.

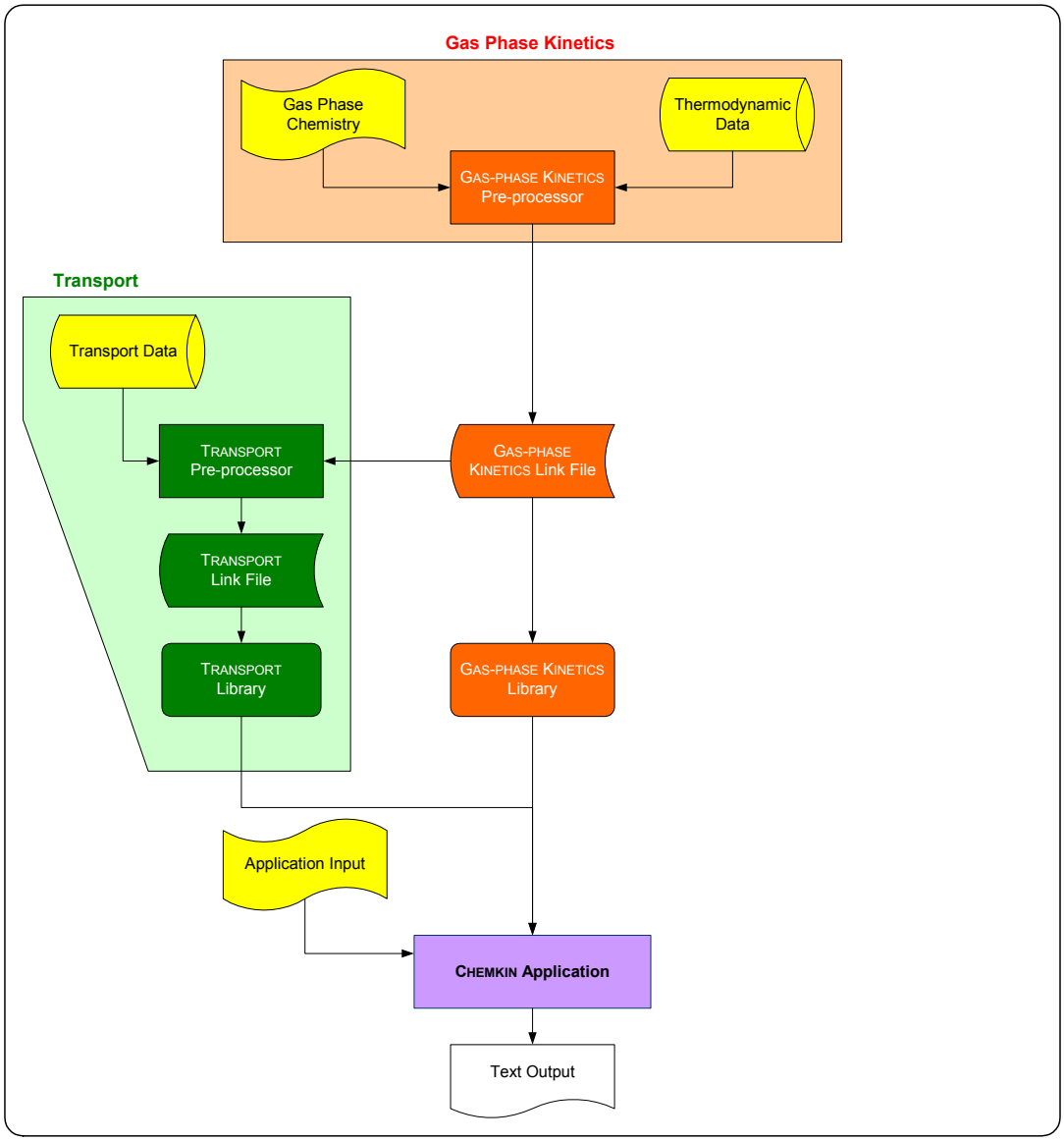
The first step is to execute the GAS-PHASE KINETICS Pre-processor. The GAS-PHASE KINETICS package is documented separately in the GAS-PHASE KINETICS Core Utility User Manual. The GAS-PHASE KINETICS Pre-processor first reads user-supplied information about the species and chemical reactions in a problem. It then extracts further information about the species' thermodynamic properties from the Thermodynamic Database and from a separate user input file, if provided. This information is stored on the GAS-PHASE KINETICS Linking File, a file that is needed by the TRANSPORT Pre-processor, and later by the GAS-PHASE KINETICS Subroutine Library.

The next program to be executed is the TRANSPORT Pre-processor. It needs input from the Transport database, and from the GAS-PHASE KINETICS Linking File. The Transport database 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, rotational relaxation collision number $Z_{\text{rot.}}$ and an indicator regarding the nature and geometrical configuration of the molecule. A supplemental input file can also contain this information, as described in *Section 3.3* below. The information coming from the GAS-PHASE KINETICS Linking File contains the species names. The names in both the Thermodynamic and the Transport databases must correspond exactly. Like the GAS-PHASE KINETICS Pre-processor, the TRANSPORT Pre-processor produces a TRANSPORT Linking File that is later needed in the TRANSPORT Subroutine Library.

3.2 Accessing the TRANSPORT Subroutine Library from an Application

Both the GAS-PHASE KINETICS and the TRANSPORT Subroutine Libraries must be initialized before 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 Applications.

Figure 3-1 Schematic representing the relationship of TRANSPORT, GAS-PHASE KINETICS, and the Application.



3.3 TRANSPORT Input File

As an alternative or supplement to the Transport database file, a separate input file may be specified for the TRANSPORT Pre-processor, either from the CHEMKIN Application User Interface or when invoked from the command line. The format of the input file is identical to that of the Transport database file described in *Chapter 6*.

4

4 Quick Reference Guide to the TRANSPORT Subroutine Library

Users creating their own Chemkin Applications may need to access the TRANSPORT Subroutine Library directly to calculate transport properties. This chapter is arranged by topical area to provide a quick reference to each of the TRANSPORT subroutines. In addition to the subroutine call list itself, the purpose of the subroutine is briefly described.

4.1 Mnemonics

There are seventeen 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 precede 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 only one other non-property subroutine, called `MCPRAM`; it is used to return the arrays of molecular parameters that came from the database 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 Kelvin, and the species mole fractions. 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 GAS-PHASE KINETICS Pre-processor input.

Here we provide a short description of each subroutine according to its function. In *Chapter 5* we list the subroutines in alphabetical order and provide a longer description of each subroutine including call-list details.

4.2 Initialization and Parameters

Table 4-1 Initialization and Parameter Subroutines

Subroutine	*Description*
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 GAS-PHASE KINETICS 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 database.
SUBROUTINE MCPNT (LSAVE, LOUT, NPOINT, V, P, LI, LR, IERR)	Reads from a binary file information about a TRANSPORT linkfile, pointers for the TRANSPORT Library, and returns lengths of work arrays.
SUBROUTINE MCSAVE (LOUT, LSAVE, IMCWRK, RMCWRK)	Writes to a binary file information about a TRANSPORT linkfile, pointers for the TRANSPORT library, and TRANSPORT work arrays.
SUBROUTINE MCREWR (LINKMC, LOUT, IMCWRK, RMCWRK, IFLAG)	This subroutine writes a new the TRANSPORT linkfile from the data stored in the integer and real work arrays, IMCWRK(*) and RMCWRK(*).
SUBROUTINE MCLEN (LINKMC, LOUT, LI, LR, IFLAG)	Returns the lengths required for work arrays.

4.3 Viscosity

Table 4-2 Viscosity Subroutines

Subroutine	*Description*
SUBROUTINE MCSVIS (T, RMCWRK, VIS)	This subroutine computes the array of pure species viscosities given the temperature.

Table 4-2 Viscosity Subroutines

Subroutine	*Description*
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.
SUBROUTINE MCCVEX (K, KDIM, RCKWRK, COFVIS)	Gets or puts values of the fitting coefficients for the polynomial fits to species viscosity.

4.4 Conductivity

Table 4-3 Conductivity Subroutines

Subroutine	*Description*
SUBROUTINE MCSCON (T, RMCWRK, CON)	This subroutine computes the array 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, IMCWRK, RMCWRK, ICKWRK, CKWRK, DT, COND)	This subroutine computes the thermal diffusion coefficients and mixture thermal conductivities given the pressure, temperature, and mole fractions.
SUBROUTINE MCCCEX (K, KDIM, RCKWRK, COFCON)	Gets or puts values of the fitting coefficients for the polynomial fits to species conductivity.

4.5 Diffusion Coefficients

Table 4-4 Diffusion Coefficients Subroutines

Subroutine	*Description*
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 the mixture-averaged diffusion coefficients given the pressure, temperature, and species mole 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.
SUBROUTINE MCCDEX (K, KDIM, RCKWRK, COFDIF)	Gets or puts values of the fitting coefficients for the polynomial fits to species binary diffusion coefficients.

4.6 Thermal Diffusion

Table 4-5 Thermal Diffusion Subroutines

Subroutine	*Description*
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, IMCWRK, RMCWRK, ICKWRK, CKWRK, DT, COND)	This subroutine computes the thermal diffusion coefficients, and mixture thermal conductivities given the pressure, temperature, and mole fractions.

5

5 Alphabetical Listing of the TRANSPORT Subroutine Library

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

5.1 MCACON

```
MCACON    MCACON    MCACON    MCACON    MCACON    MCACON    MCACON    MCACON
             ******************************************
                   *********************************
                         *********************

  SUBROUTINE MCACON (T, X, RMCWRK, CONMIX)
  Returns the mixture thermal conductivity given temperature and
  species mole fractions.

  INPUT
  T         - Real scalar, temperature.
                   cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  CONMIX    - Real scalar, mixture thermal conductivity.
                   cgs units, erg/cm*K*s
```


5.2 MCADIF

```
MCADIF   MCADIF   MCADIF   MCADIF   MCADIF   MCADIF   MCADIF   MCADIF
            ******************************************
                  ********************************
                        ********************

  SUBROUTINE MCADIF (P, T, X, RMCWRK, D)
  Returns mixture-averaged diffusion coefficients given pressure,
  temperature, and species mole fractions.

  INPUT
  P         - Real scalar, pressure.
                 cgs units, dynes/cm**2
  T         - Real scalar, temperature.
                 cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  D(*)      - Real array, mixture diffusion coefficients;
              dimension at least KK, the total species count.
                 cgs units, cm**2/s
```

5.3 MCATDR

```
MCATDR   MCATDR   MCATDR   MCATDR   MCATDR   MCATDR   MCATDR   MCATDR
            ******************************************
                  ********************************
                        ********************

  SUBROUTINE MCATDR (T, X, IMCWRK, RMCWRK, TDR)
  This subroutine computes the thermal diffusion ratios for the light
  species into the mixture.

  INPUT
  T         - Real scalar, temperature.
                 cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.
  IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
  RMCWRK(*) - Real    workspace array; dimension at least LENRMC.

  OUTPUT
  TDR(*)    - Real array, thermal diffusion ratios for the species;
              dimension at least KK, the total species count.
              TDR(K) = 0 for any species with molecular weight less
              than 5.
```

5.4 MCAVIS

```
MCAVIS   MCAVIS   MCAVIS   MCAVIS   MCAVIS   MCAVIS   MCAVIS   MCAVIS
            ******************************************
                 *********************************
                      ********************

  SUBROUTINE MCAVIS (T, X, RMCWRK, VISMIX)
  Returns mixture viscosity, given temperature and species mole
  fractions.  It uses modification of the Wilke semi-empirical
  formulas.

  INPUT
  T         - Real scalar, temperature.
                  cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  VISMIX    - Real scalar, mixture viscosity.
                  cgs units, gm/cm*s
```

5.5 MCCCEX

```
MCCCEX   MCCCEX   MCCCEX   MCCCEX   MCCCEX   MCCCEX   MCCCEX   MCCCEX
            ******************************************
                 *********************************
                      ********************

  SUBROUTINE MCCCEX (K, KDIM, RMCWRK, COFCON)
  Gets or puts values of the fitting coefficients for the
  polynomial fits to species conductivity.

  INPUT
  K         - Integer scalar, species index.
              K > 0 gets coefficients from RMCWRK
              K < 0 puts coefficients into RMCWRK
  KDIM      - Dimension for COFCON - the total number of species
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  If K < 1:
  COFCON    - Real vector of polynomial coefficients for
              the species' conductivity; dimension at least NO,
              usually 4.

  OUTPUT
  If K > 1:
  COFCON    - Real vector of polynomial coefficients for
              the species' conductivity; dimension at least NO,
              usually 4.
```


5.6 MCCDEX

```
MCCDEX   MCCDEX   MCCDEX   MCCDEX   MCCDEX   MCCDEX   MCCDEX   MCCDEX
            ******************************************
                  *********************************
                        *********************

  SUBROUTINE MCCDEX (K, KDIM, RMCWRK, COFDIF)
  Gets or puts values of the fitting coefficients for the
  polynomial fits to species binary diffusion coefficients.

  INPUT
  K         - Integer scalar, species index.
              K > 0 gets coefficients from RMCWRK
              K < 0 puts coefficients into RMCWRK
  KDIM      - Dimension for COFDIF - the total number of species
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  If K < 1:
  COFDIF    - Real matrix of polynomial coefficients for
              the species' binary diffusion coefficient with all
              other species; The first dimension should be KK;
              the second dimension should be NO, usually 4;

  OUTPUT
  If K > 1:
  COFDIF    - Real matrix of polynomial coefficients for
              the species' binary diffusion coefficient with all
              other species; first dimension should be NO, usually 4;
              The second dimension should be NKK
```

5.7 MCCVEX

```
MCCVEX   MCCVEX   MCCVEX   MCCVEX   MCCVEX   MCCVEX   MCCVEX   MCCVEX
            ******************************************
                  *********************************
                        *********************

  SUBROUTINE MCCVEX (K, KDIM, RMCWRK, COFVIS)
  Gets or puts values of the fitting coefficients for the
  polynomial fits to species viscosity.

  INPUT
  K         - Integer scalar, species index.
              K > 0 gets coefficients from RMCWRK
              K < 0 puts coefficients into RMCWRK
  KDIM      - Dimension for COFVIS - the total number of species
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  If K < 1:
  COFVIS    - Real vector of polynomial coefficients for
              the species' viscosity; dimension at least NO, usually 4

  OUTPUT
  If K > 1:
  COFVIS    - Real vector of polynomial coefficients; dimension
              at least NO, usually = 4
```

5.8 MCINIT

```
MCINIT   MCINIT   MCINIT   MCINIT   MCINIT   MCINIT   MCINIT   MCINIT
            ******************************************
                  *********************************
                        ********************


  SUBROUTINE MCINIT (LINKMC, LOUT, LENIMC, LENRMC, IMCWRK, RMCWRK,
                     IFLAG)
  This subroutine reads the transport linkfile from the fitting code
  and creates 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.

  INPUT
  LINKMC    - Integer scalar, transport linkfile input unit number.
  LOUT      - Integer scalar, formatted output file unit number.
  LENIMC    - Integer scalar, minimum dimension of the integer
              storage and workspace array IMCWRK(*);
              LENIMC must be at least:
              LENIMC = 4*KK + NLITE,
              where KK is the total species count, and
                    NLITE is the number of species with molecular
                          weight less than 5.
  LENRMC    - Integer scalar, minimum dimension of the real storage
              and workspace array RMCWRK(*);
              LENRMC must be at least:
              LENRMC = KK*(19 + 2*NO + NO*NLITE) + (NO+15)*KK**2,
              where KK is the total species count,
                    NO is the order of the polynomial fits (NO=4),
                    NLITE is the number of species with molecular
                          weight less than 5.
  OUTPUT
  IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
  RMCWRK(*) - Real    workspace array; dimension at least LENRMC.
```

5.9 MCLEN

```
MCLEN    MCLEN    MCLEN    MCLEN    MCLEN    MCLEN    MCLEN    MCLEN
            ******************************************
                  *********************************
                        ********************

  SUBROUTINE MCLEN (LINKMC, LOUT, LI, LR, IFLAG)
  Returns the lengths required for work arrays.

  INPUT
  LINKMC   - Integer scalar, input file unit for the linkfile.
  LOUT     - Integer scalar, formatted output file unit.

  OUTPUT
  LI       - Integer scalar, minimum length required for the
             integer work array.
  LR       - Integer scalar, minimum length required for the
             real work array.
  IFLAG    - Integer scalar, indicates successful reading of
             linkfile; IFLAG>0 indicates error type.
```


5.10 MCMCDT

```
MCMCDT   MCMCDT   MCMCDT   MCMCDT   MCMCDT   MCMCDT   MCMCDT   MCMCDT
            ******************************************
                  *********************************
                        ********************

  SUBROUTINE MCMCDT (P, T, X, IMCWRK, RMCWRK, ICKWRK, CKWRK,
                     DT, COND)
  Returns thermal diffusion coefficients, and mixture thermal
  conductivities, given pressure, temperature, and mole fraction.

  INPUT
  P         - Real scalar, pressure.
              cgs units, dynes/cm**2
  T         - Real scalar, temperature.
              cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.

  IMCWRK(*) - Integer TRANSPORT workspace array;
              dimension at least LENIMC.
  RMCWRK(*) - Real    TRANSPORT workspace array;
              dimension at least LENRMC.
  ICKWRK(*) - Integer CHEMKIN workspace array;
              dimension at least LENICK.
  RCKWRK(*) - Real    CHEMKIN workspace array;
              dimension at least LENRCK.

  OUTPUT
  DT(*)     - Real array, thermal multicomponent diffusion
              coefficients;
              dimension at least KK, the total species count.
                 cgs units, gm/(cm*sec)
                  CGS UNITS - GM/(CM*SEC)
  COND      - Real scalar, mixture thermal conductivity.
              cgs units, erg/(cm*K*s)
```

5.11 MCMDIF

```
MCMDIF   MCMDIF   MCMDIF   MCMDIF   MCMDIF   MCMDIF   MCMDIF   MCMDIF
            ******************************************
                  *********************************
                        ********************

  SUBROUTINE MCMDIF (P, T, X, KDIM, IMCWRK, RMCWRK, D)
  Returns the ordinary multicomponent diffusion coefficients,
  given pressure, temperature, and mole fractions.

  INPUT
  P         - Real scalar, pressure.
              cgs units, dynes/cm**2
  T         - Real scalar, temperature.
              cgs units, K
  X(*)      - Real array, mole fractions of the mixture;
              dimension at least KK, the total species count.
  KDIM      - Integer scalar, actual first dimension of D(KDIM,KK);
              KDIM must be at least KK, the total species count.
  IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
  RMCWRK(*) - Real    workspace array; dimension at least LENRMC.

  OUTPUT
  D(*,*)    - Real matrix, ordinary multicomponent diffusion
              coefficients;
              dimension at least KK, the total species count, for
              both the first and second dimensions.
                 cgs units, cm**2/s
```

5.12 MCPNT

```
MCPNT    MCPNT    MCPNT    MCPNT    MCPNT    MCPNT    MCPNT    MCPNT
           ********************************************
                 *********************************
                       *********************

  SUBROUTINE MCPNT (LSAVE, LOUT, NPOINT, V, P, LI, LR, IERR)
  Reads from a binary file information about a Transport linkfile,
  pointers for the Transport Library, and returns lengths of work
  arrays.

  INPUT
  LSAVE  - Integer scalar, input unit for binary data file.
  LOUT   - Integer scalar, formatted output file unit.

  OUTPUT
  NPOINT - Integer scalar, total number of pointers.
  V      - Real scalar, version number of the Transport linkfile.
  P      - Character string, machine precision of the linkfile.
  LI     - Integer scalar, minimum dimension required for integer
           workspace array.
  LR     - Integer scalar, minimumm dimension required for real
           workspace array.
  IERR   - Logical, error flag.
```

5.13 MCPRAM

```
MCPRAM   MCPRAM   MCPRAM   MCPRAM   MCPRAM   MCPRAM   MCPRAM   MCPRAM
           ********************************************
                 *********************************
                       *********************

  SUBROUTINE MCPRAM (IMCWRK, RMCWRK, EPS, SIG, DIP, POL, ZROT, NLIN)
  Returns the arrays of molecular parameters as read from the
  transport database.

  INPUT
  IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
  RMCWRK(*) - Real    workspace array; dimension at least LENRMC.

  OUTPUT
  EPS(*)    - Real array, Lennard-Jones Potential well depths for
              the species;
              dimension at least KK, the total species count.
                 cgs units, K
  SIG(*)    - Real array, Lennary-Jones collision diameters for
              the species;
              dimension at least KK, the total species count.
                 cgs units, Angstrom
  DIP(*)    - Real array, dipole moments for the species;
              dimension at least KK, the total species count.
                 cgs units, Debye
  POL(*)    - Real array, polarizabilities for the species;
              dimension at least KK, the total species count.
                 cgs units, Angstrom**3
  ZROT(*)   - Real array, rotational collision numbers evaluated at
              298K for the species;
              dimension at least KK, the total species count.
  NLIN(*)   - Integer array, flags for species linearity;
              dimension at least KK, the total species count.
              NLIN=0, single atom,
              NLIN=1, linear molecule,
              NLIN=2, linear molecule.
```


5.14 MCREWR

```
MCREWR   MCREWR   MCREWR   MCREWR   MCREWR   MCREWR   MCREWR   MCREWR
            ******************************************
                  *********************************
                        *********************

   SUBROUTINE MCREWR (LINKMC, LOUT, IMCWRK, RMCWRK, IFLAG)
   This subroutine writes a new the transport linkfile from
   the data stored in the integer and real work arrays,
   IMCWRK(*) and RMCWRK(*).

   INPUT
   LINKMC    - Integer scalar, transport linkfile output unit number.
   LOUT      - Integer scalar, formatted output file unit number.

   OUTPUT
   IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
   RMCWRK(*) - Real    workspace array; dimension at least LENRMC.
```

5.15 MCSAVE

```
MCSAVE   MCSAVE   MCSAVE   MCSAVE   MCSAVE   MCSAVE   MCSAVE   MCSAVE
            ******************************************
                  *********************************
                        *********************

  SUBROUTINE MCSAVE (LOUT, LSAVE, IMCWRK, RMCWRK)
  Writes to a binary file information about a Transport linkfile,
  pointers for the Transport library, and Transport work arrays.

  INPUT
  LOUT      - Integer scalar, formatted output file unit number.
  LSAVE     - Integer scalar, unformatted output file unit number.
  IMCWRK(*) - Integer workspace array; dimension at least LENIMC.
  RMCWRK(*) - Real    workspace array; dimension at least LENRMC.
```

5.16 MCSCON

```
MCSCON   MCSCON   MCSCON   MCSCON   MCSCON   MCSCON   MCSCON   MCSCON
            ******************************************
                  *********************************
                        *********************

  SUBROUTINE MCSCON (T, RMCWRK, CON)
  Returns the array of pur species conductivities given temperature.

  INPUT
  T         - Real scalar, temperature.
                cgs units, K
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  CON(*)    - Real array, species thermal conductivities;
              dimension at least KK, the total species count.
                cgs units, erg/cm*K*s
```

5.17 MCSDIF

```
MCSDIF   MCSDIF   MCSDIF   MCSDIF   MCSDIF   MCSDIF   MCSDIF   MCSDIF
            ******************************************
                  *********************************
                        ********************

  SUBROUTINE MCSDIF (P, T, KDIM, RMCWRK, DJK)
  Returns the binary diffusion coefficients given pressure and
  temperature.

  INPUT
  P         - Real scalar, pressure.
                 cgs units, dynes/cm**2
  T         - Real scalar, temperature.
                 cgs units, K
  KDIM      - Integer scalar, actual first dimension of DJK(KDIM,KK).
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  DJK(*,*)  - Real matrix, binary diffusion coefficients;
              dimension at least KK, the total species count, for
              both the first and second dimensions.
                 cgs units, cm**2/s
              CJK(J,K) is the diffusion coefficient of species J
              in species K.
```

5.18 MCSVIS

```
MCSVIS   MCSVIS   MCSVIS   MCSVIS   MCSVIS   MCSVIS   MCSVIS   MCSVIS
            ******************************************
                  *********************************
                        ********************

  SUBROUTINE MCSVIS (T, RMCWRK, VIS)
  Returns the array of pure species viscosities, given temperature.

  INPUT
  T         - Real scalar, temperature.
                 cgs units, K
  RMCWRK(*) - Real workspace array; dimension at least LENRMC.

  OUTPUT
  VIS(*)    - Real array, species viscosities;
              dimension at least KK, the total species count.
                 cgs units, gm/cm*s
```


6 Transport Database

In this section we list the database file that is currently included with the TRANSPORT software. While this database file is more of a historical record, we expect that users will want to add their own collection of data to suit their own needs. The database file included with CHEMKIN 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.

User data of the same format as the database file can now be provided in a supplemental input data file.

We recommend using this approach, rather than editing the TRANSPORT database file, since the database file may be updated with subsequent versions of the CHEMKIN Collection. When adding a new species to either the database file or to a supplemental TRANSPORT data input file, be sure that the species name is exactly the same as it is in the CHEMKIN Thermodynamic Database.

Some of the numbers in the database have been determined by computing “best fits” to experimental measurements of some transport property (e.g. viscosity). In other cases the Lennard-Jones parameters have been estimated following the methods outlined in Svehla.[20]

The first 16 columns in each line of the database are reserved for the species name. Presently CHEMKIN is programmed to allow no more than 16-character names. Columns 17 through 80 are free-format, 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.

A Debye is $10^{-18}\text{cm}^{3/2}\text{erg}^{1/2}$.

5. The polarizability α in cubic Angstroms.
6. The rotational relaxation collision number $Z_{\text{rot.}}$ at 298K.
7. A "comment" line is one that has either a period (.), slash (/), or exclamation mark (!) as the first non-blank character. In addition, on any line, any input that follows an exclamation mark is taken as a comment.

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
Al2Me6	2	471.	6.71	0.0	0.0	1.0
AlMe3	2	471.	5.30	0.0	0.0	1.0
AR	0	136.500	3.330	0.000	0.000	0.000
AR*	0	136.500	3.330	0.000	0.000	0.000
AS	0	1045.5	4.580	0.000	0.000	0.000
AS2	1	1045.5	5.510	0.000	0.000	1.000
ASH	1	199.3	4.215	0.000	0.000	1.000
ASH2	2	229.6	4.180	0.000	0.000	1.000
ASH3	2	259.8	4.145	0.000	0.000	1.000
AsH3	2	259.8	4.145	0.000	0.000	1.000
BCL3	2	337.7	5.127	0.000	0.000	1.000
C	0	71.400	3.298	0.000	0.000	0.000
C-SI3H6	2	331.2	5.562	0.000	0.000	1.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
C2	1	97.530	3.621	0.000	1.760	4.000
C2F4	2	202.6	5.164	0.000	0.000	1.000
C2F6	2	194.5	5.512	0.000	0.000	1.000
C2H	1	209.000	4.100	0.000	0.000	2.500
C2H2	1	209.000	4.100	0.000	0.000	2.500
C2H2OH	2	224.700	4.162	0.000	0.000	1.000
C2H3	2	209.000	4.100	0.000	0.000	1.000
C2H4	2	280.800	3.971	0.000	0.000	1.500
C2H5	2	252.300	4.302	0.000	0.000	1.500
C2H5OH	2	362.6	4.53	0.000	0.000	1.000
C2H6	2	252.300	4.302	0.000	0.000	1.500
C2N	1	232.400	3.828	0.000	0.000	1.000
C2N2	1	349.000	4.361	0.000	0.000	1.000
C2O	1	232.400	3.828	0.000	0.000	1.000
C3H2	2	209.000	4.100	0.000	0.000	1.000
C3H3	1	252.000	4.760	0.000	0.000	1.000
C3H4	1	252.000	4.760	0.000	0.000	1.000
C3H4P	1	252.000	4.760	0.000	0.000	1.000
C3H6	2	266.800	4.982	0.000	0.000	1.000
C3H7	2	266.800	4.982	0.000	0.000	1.000
C3H8	2	266.800	4.982	0.000	0.000	1.000
C4H	1	357.000	5.180	0.000	0.000	1.000
C4H2	1	357.000	5.180	0.000	0.000	1.000
C4H2OH	2	224.700	4.162	0.000	0.000	1.000
C4H3	1	357.000	5.180	0.000	0.000	1.000
C4H4	1	357.000	5.180	0.000	0.000	1.000
C4H6	2	357.000	5.180	0.000	0.000	1.000
C4H8	2	357.000	5.176	0.000	0.000	1.000
C4H9	2	357.000	5.176	0.000	0.000	1.000
C4H9	2	357.000	5.176	0.000	0.000	1.000
C5H2	1	357.000	5.180	0.000	0.000	1.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
C5H3	1	357.000	5.180	0.000	0.000	1.000
C5H5OH	2	450.000	5.500	0.000	0.000	1.000
C6H2	1	357.000	5.180	0.000	0.000	1.000
C6H5	2	412.300	5.349	0.000	0.000	1.000
C6H5(L)	2	412.300	5.349	0.000	0.000	1.000
C6H5O	2	450.000	5.500	0.000	0.000	1.000
C6H6	2	412.300	5.349	0.000	0.000	1.000
C6H7	2	412.300	5.349	0.000	0.000	1.000
CF	1	94.2	3.635	0.000	0.000	1.000
CF2	2	108.0	3.977	0.000	0.000	1.000
CF3	2	121.0	4.320	0.000	0.000	1.000
CF4	2	134.0	4.662	0.000	0.000	1.000
CH	1	80.000	2.750	0.000	0.000	0.000
CH2	1	144.000	3.800	0.000	0.000	0.000
CH2(S)	1	144.000	3.800	0.000	0.000	0.000
CH2(SING)	1	144.000	3.800	0.000	0.000	0.000
CH2CHCCH	2	357.000	5.180	0.000	0.000	1.000
CH2CHCCH2	2	357.000	5.180	0.000	0.000	1.000
CH2CHCH2	2	260.000	4.850	0.000	0.000	1.000
CH2CHCHCH	2	357.000	5.180	0.000	0.000	1.000
CH2CHCHCH2	2	357.000	5.180	0.000	0.000	1.000
CH2CO	2	436.000	3.970	0.000	0.000	2.000
CH2F2	2	318.0	4.080	0.000	0.000	1.000
CH2HCO	2	436.000	3.970	0.000	0.000	2.000
CH2O	2	498.000	3.590	0.000	0.000	2.000
CH2OH	2	417.000	3.690	1.700	0.000	2.000
CH3	1	144.000	3.800	0.000	0.000	0.000
CH3CC	2	252.000	4.760	0.000	0.000	1.000
CH3CCCH2	2	357.000	5.180	0.000	0.000	1.000
CH3CCCH3	2	357.000	5.180	0.000	0.000	1.000
CH3CCH2	2	260.000	4.850	0.000	0.000	1.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
CH3CH2CCH	2	357.000	5.180	0.000	0.000	1.000
CH3CHCH	2	260.000	4.850	0.000	0.000	1.000
CH3CHO	2	436.000	3.970	0.000	0.000	2.000
CH3CO	2	436.000	3.970	0.000	0.000	2.000
CH3O	2	417.000	3.690	1.700	0.000	2.000
CH3OH	2	481.800	3.626	0.000	0.000	1.000
CH4	2	141.400	3.746	0.000	2.600	13.000
CH4O	2	417.000	3.690	1.700	0.000	2.000
CHF3	2	240.0	4.330	0.000	0.000	1.000
CL	0	130.8	3.613	0.000	0.000	1.000
CL-	0	130.8	3.613	0.000	0.000	1.000
CL2BNH2	2	337.7	5.127	0.000	0.000	1.000
CN	1	75.000	3.856	0.000	0.000	1.000
CN2	1	232.400	3.828	0.000	0.000	1.000
CNC	1	232.400	3.828	0.000	0.000	1.000
CNN	1	232.400	3.828	0.000	0.000	1.000
CO	1	98.100	3.650	0.000	1.950	1.800
CO2	1	244.000	3.763	0.000	2.650	2.100
DMG	2	675.8	5.22	0.000	0.000	1.000
E	0	850.	425.	0.000	0.000	1.000
F	0	80.000	2.750	0.000	0.000	0.000
F2	1	125.700	3.301	0.000	1.600	3.800
GA	0	2961.8	4.62	0.000	0.000	0.000
GACH3	2	972.7	4.92	0.000	0.000	1.000
GAH	1	335.5	4.24	0.000	0.000	1.000
GAME	2	972.7	4.92	0.000	0.000	1.000
GAME2	2	675.8	5.22	0.000	0.000	1.000
GAME3	2	378.2	5.52	0.000	0.000	1.000
GaMe3	2	378.2	5.52	0.000	0.000	1.000
H	0	145.000	2.050	0.000	0.000	0.000
H2	1	38.000	2.920	0.000	0.790	280.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
H2ASCH3	2	408.0	4.73	0.000	0.000	1.000
H2C4O	2	357.000	5.180	0.000	0.000	1.000
H2CCCCH	2	357.000	5.180	0.000	0.000	1.000
H2CCCCH2	2	357.000	5.180	0.000	0.000	1.000
H2CCCH	2	252.000	4.760	0.000	0.000	1.000
H2CN	1	569.000	3.630	0.000	0.000	1.000
H2NO	2	116.700	3.492	0.000	0.000	1.000
H2O	2	572.400	2.605	1.844	0.000	4.000
H2O2	2	107.400	3.458	0.000	0.000	3.800
H2S	2	301.000	3.600	0.000	0.000	1.000
H2SISIH2	2	312.6	4.601	0.000	0.000	1.000
H3SISIH	2	312.6	4.601	0.000	0.000	1.000
HC2N2	1	349.000	4.361	0.000	0.000	1.000
HCCHCCH	2	357.000	5.180	0.000	0.000	1.000
HCCO	2	150.000	2.500	0.000	0.000	1.000
HCCOH	2	436.000	3.970	0.000	0.000	2.000
HCL	1	344.7	3.339	0.000	0.000	1.000
HCN	1	569.000	3.630	0.000	0.000	1.000
HCNO	2	232.400	3.828	0.000	0.000	1.000
HCO	2	498.000	3.590	0.000	0.000	0.000
HCO+	1	498.000	3.590	0.000	0.000	0.000
HE	0	10.200	2.576	0.000	0.000	0.000
HF	1	330.000	3.148	1.920	2.460	1.000
HF0	1	352.000	2.490	1.730	0.000	5.000
HF1	1	352.000	2.490	1.730	0.000	5.000
HF2	1	352.000	2.490	1.730	0.000	5.000
HF3	1	352.000	2.490	1.730	0.000	5.000
HF4	1	352.000	2.490	1.730	0.000	5.000
HF5	1	352.000	2.490	1.730	0.000	5.000
HF6	1	352.000	2.490	1.730	0.000	5.000
HF7	1	352.000	2.490	1.730	0.000	5.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
HF8	1	352.000	2.490	1.730	0.000	5.000
HNCO	2	232.400	3.828	0.000	0.000	1.000
HNNO	2	232.400	3.828	0.000	0.000	1.000
HNO	2	116.700	3.492	0.000	0.000	1.000
HNOH	2	116.700	3.492	0.000	0.000	1.000
HO2	2	107.400	3.458	0.000	0.000	1.000
HOCN	2	232.400	3.828	0.000	0.000	1.000
HSO2	2	252.000	4.290	0.000	0.000	1.000
I*C3H7	2	266.800	4.982	0.000	0.000	1.000
I*C4H9	2	357.000	5.176	0.000	0.000	1.000
K	0	850.	4.25	0.000	0.000	1.000
K+	0	850.	4.25	0.000	0.000	1.000
KCL	1	1989.	4.186	0.000	0.000	1.000
KH	1	93.3	3.542	0.000	0.000	1.000
KO	1	383.0	3.812	0.000	0.000	1.000
KO2	2	1213.	4.69	0.000	0.000	1.000
KOH	2	1213.	4.52	0.000	0.000	1.000
N	0	71.400	3.298	0.000	0.000	0.000
N*C3H7	2	266.800	4.982	0.000	0.000	1.000
N2	1	97.530	3.621	0.000	1.760	4.000
N2H2	2	71.400	3.798	0.000	0.000	1.000
N2H3	2	200.000	3.900	0.000	0.000	1.000
N2H4	2	205.000	4.230	0.000	4.260	1.500
N2O	1	232.400	3.828	0.000	0.000	1.000
NCN	1	232.400	3.828	0.000	0.000	1.000
NCNO	2	232.400	3.828	0.000	0.000	1.000
NCO	1	232.400	3.828	0.000	0.000	1.000
NH	1	80.000	2.650	0.000	0.000	4.000
NH2	2	80.000	2.650	0.000	2.260	4.000
NH3	2	481.000	2.920	1.470	0.000	10.000
NNH	2	71.400	3.798	0.000	0.000	1.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
NO	1	97.530	3.621	0.000	1.760	4.000
NO2	2	200.000	3.500	0.000	0.000	1.000
O	0	80.000	2.750	0.000	0.000	0.000
O(Si(OC2H5)3)2	2	522.7	5.25	0.000	0.000	1.000
O2	1	107.400	3.458	0.000	1.600	3.800
O3	2	180.000	4.100	0.000	0.000	2.000
OH	1	80.000	2.750	0.000	0.000	0.000
OSI(OC2H5)2	2	522.7	7.03	0.000	0.000	1.000
PH3	2	251.5	3.981	0.000	0.000	1.000
S	0	847.000	3.839	0.000	0.000	0.000
S*C4H9	2	357.000	5.176	0.000	0.000	1.000
S2	1	847.000	3.900	0.000	0.000	1.000
SH	1	847.000	3.900	0.000	0.000	1.000
SI	0	3036.	2.910	0.000	0.000	0.000
Si(OC2H5)4	2	522.7	7.03	0.000	0.000	1.000
SI(OC2H5)4	2	522.7	7.03	0.000	0.000	1.000
Si(OH)(OC2H5)3	2	522.7	7.03	0.000	0.000	1.000
SI(OH)(OC2H5)3	2	522.7	7.03	0.000	0.000	1.000
SI(OH)2(OC2H5)2	2	522.7	6.35	0.000	0.000	1.000
Si(OH)2(OC2H5)2	2	522.7	6.35	0.000	0.000	1.000
SI(OH)3(OC2H5)	2	522.7	5.75	0.000	0.000	1.000
SI(OH)4	2	522.7	5.25	0.000	0.000	1.000
SI2	1	3036.	3.280	0.000	0.000	1.000
SI2H2	2	323.8	4.383	0.000	0.000	1.000
SI2H3	2	318.2	4.494	0.000	0.000	1.000
SI2H4	2	312.6	4.601	0.000	0.000	1.000
SI2H5	2	306.9	4.717	0.000	0.000	1.000
SI2H6	2	301.3	4.828	0.000	0.000	1.000
SI3	2	3036.	3.550	0.000	0.000	1.000
SI3H8	2	331.2	5.562	0.000	0.000	1.000
SIF	1	585.0	3.318	0.000	0.000	1.000

Table 6-1 Species in TRANSPORT Database

Species Name	*Geometry*	ε/k_B	σ	μ	α	$Z_{\text{rot.}}$
SIF3	2	309.6	4.359	0.000	0.000	1.000
SIF3NH2	2	231.0	4.975	0.000	0.000	1.000
SIF4	2	171.9	4.880	0.000	0.000	1.000
SIH	1	95.8	3.662	0.000	0.000	1.000
SIH2	2	133.1	3.803	0.000	0.000	1.000
SIH2(3)	2	133.1	3.803	0.000	0.000	1.000
SIH3	2	170.3	3.943	0.000	0.000	1.000
SIH3SIH2SIH	2	331.2	5.562	0.000	0.000	1.000
SIH4	2	207.6	4.084	0.000	0.000	1.000
SIHF3	2	180.8	4.681	0.000	0.000	1.000
SO	1	301.000	3.993	0.000	0.000	1.000
SO2	2	252.000	4.290	0.000	0.000	1.000
SO3	2	378.400	4.175	0.000	0.000	1.000
TMG	2	378.2	5.52	0.000	0.000	1.000

7 References

1 R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller, *A Fortran Computer Code Package for the Evaluation of Gas-Phase Multicomponent Transport Properties*, Sandia National Laboratories Report SAND86-8246, (1986).

2 G. Dixon-Lewis, *Proceedings of the Royal Society A*. **304**:111 (1968).

3 J. Warnatz, in *Numerical Methods in Flame Propagation*, edited by N. Peters and J. Warnatz Friedr. Vieweg and Sohn, Wiesbaden, 1982.

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

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

6 T. P. Coffee and J. M. Heimerl, *Combustion and Flame* **43**:273 (1981).

7 J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird, *Molecular Theory of Gases and Liquids*, John Wiley and Sons, New York, 1954.

8 L. Monchick and E. A. Mason, *Journal of Chemical Physics* **35**:1676 (1961).

9 T. R. Marrero and E. A. Mason, *J. of Phys. and Chem. Ref. Data* **1**:3 (1972).

10 J. G. Parker, *Physics of Fluids* **2**:449 (1959).

11 C. C. Brau and R. M. Jonkman, *Journal of Chemical Physics* **52**:447 (1970).

12 J. Warnatz, *Ber. Bunsenges. Phys. Chem.* **82**:193 (1978).

13 M. E. Coltrin, R. J. Kee, and J. A. Miller, *Journal of the Electrochemical Society* **133**:1206 (1986).

14 C. R. Wilke, *Journal of Chemical Physics* **18**:517 (1950).

15 S. Mathur, P. K. Tondon, and S. C. Saxena, *Molecular Physics* **12**:569 (1967).

16 S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases Cambridge University Press*, Cambridge, 1970.

17 J. A. Miller, R. E. Mitchell, M. D. Smooke, and R. J. Kee, in *Proceedings of the Nineteenth Symposium (International) on Combustion*, The Combustion Institute, Pittsburgh, Pennsylvania, 1982, p. 181.

18 J. A. Miller, M. D. Smooke, R. M. Green, and R. J. Kee, *Combustion Science and Technology* **34**:149 (1983).

19 J. A. Miller, M. C. Branch, W. J. McLean, D. W. Chandler, M. D. Smooke, and R. J. Kee, in *Proceedings of the Twentieth Symposium (International) on Combustion*, The Combustion Institute, Pittsburgh, Pennsylvania, 1985, p. 673.

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

Index

Index

P

Q

R

S

T

U

V

W