EGLIB: A GENERAL-PURPOSE FORTRAN LIBRARY FOR MULTICOMPONENT TRANSPORT PROPERTY EVALUATION ⋆

Alexandre Ern[a] and Vincent Giovangigli[b]

[a] *CERMICS, Ecole Nationale des Ponts et Chaussées*
6 et 8 av. Blaise Pascal
77455 Marne la Vallée cedex 2, France.
`ern@cermics.enpc.fr`

[b] *Centre de Mathématiques Appliquées, CNRS, Ecole Polytechnique*
91128 Palaiseau cedex, France.
`giovangi@cmapx.polytechnique.fr`

Abstract. EGLIB is a general-purpose Fortran library designed to evaluate transport properties in gas mixtures. The theoretical framework for the present computational algorithms is thoroughly described in a monograph by the authors [2]. More detailed descriptions focusing on either mathematical, numerical or physical aspects can also be found in a series of papers [3–8,12]. In this note, we describe the practical use of the library. The library is available upon sending e-mail request to the authors.

1 Introduction

Transport property evaluation plays an important and often time-consuming role in the computational modeling of gaseous multicomponent reacting flows. The implementation of multicomponent transport algorithms in an accurate and computationally effective form is thus extremely useful for a wide range of engineering problems. Traditionally, two approaches were mostly considered for evaluating transport coefficients. In a first approach, a direct numerical inversion of the transport linear systems derived from kinetic theory was considered. In a second approach, an empirical average expression was used. While the former strategy often becomes computationally expensive, the latter often yields poorly accurate transport coefficients. A third approach has become recently available for the systematic evaluation of multicomponent transport coefficients [2]. In this new approach, an iterative method is used to obtain an approximate solution of the transport linear systems. Truncation then provides, at a moderate computational cost, approximate expressions of a desired level of accuracy for all the transport coefficients.

EGLIB is a Fortran library which implements the multicomponent transport algorithms derived by the authors [2–8]. These algorithms include convergent iterative methods, direct inversions of a positive definite form of the transport linear systems, as well as new empirical expressions. The library contains an initialization routine and three packages, called EGSLIB, EGMLIB, and EGFLIB. EGSLIB contains *single* input data routines which evaluate transport coefficients for a given state of the mixture, i.e., for given temperature, pressure, and species mass fractions. EGMLIB and EGFLIB contain *multiple* input data routines which evaluate transport coefficients for a family of states of the mixture. The family

⋆ EGLIB Version 3.4; July 12, 2004.

of states is typically indexed by the grid nodes discretizing the computational domain. The difference between EGMLIB and EGFLIB arises in the way transport coefficients are stored.

This manual provides a practical description of the library. More detailed discussions regarding mathematical, numerical or physical aspects of the transport algorithms are presented in a series of papers [3–8]. Sections 2 and 3 offer a brief introduction to multicomponent transport property evaluation. A general description of the library is given in Section 4. In Section 5 we describe how to initialize the library. A detailed description of the various routines offered by EGLIB is given in Sections 6 to 10. Section 6 deals with the initialization of molecular parameters, Section 7 with viscosity coefficients, Section 8 with thermal conduction and thermal diffusion, Section 9 with multicomponent diffusion and Section 10 with additional routines evaluating simultaneously various coefficients. For each routine, we indicate the calling sequence with the corresponding input and output variables. The output variables are underlined. The dimension of the arrays is given in parenthesis. In Section 11 the computational efficiency of the library is illsutrated by numerical results obtained on various computer architectures. Finally, in Section 12 we present an example of routines used in combustion applications [13,14].

2 Theoretical Framework

The equations governing multicomponent gaseous reacting flows are derived from the kinetic theory of gases by considering a generalized Boltzmann equation and the first order Enskog expansion. The collision term in the Boltzmann equation accounts for both nonreactive collisions in which the molecules exchange translational and internal degrees of freedom (rotational and vibrational) and reactive collisions. In the present theory, we restrict ourselves to the Maxwellian reaction regime in which the transport coefficients of the mixture can be evaluated as if there were no chemical reactions [2]. In addition, the present formalism applies either to mixtures in which vibrational energy is not transferred in inelastic collisions or to mixtures in strong vibrational desequilibrium in such a way that a chemical species in two different vibrational levels is actually treated as two different species and collisions with vibrational energy exchange are treated formally as chemical reactions.

2.1 Transport Coefficients

The equations governing multicomponent gaseous reacting flows express conservation of species mass, momentum, and energy and are written in terms of various transport fluxes. Specifically, these equations read

$$\begin{aligned}
\partial_t(\rho Y_i) + \nabla\cdot(\rho v Y_i) &= -\nabla\cdot(\rho Y_i V_i) + \omega_i, \qquad i \in \mathcal{S}, \\
\partial_t(\rho v) + \nabla\cdot(\rho v \otimes v) &= -\nabla\cdot\Pi + \sum_{i\in\mathcal{S}} \rho Y_i b_i, \\
\partial_t(\rho e) + \nabla\cdot(\rho v e) &= -\nabla\cdot(q + \Pi v) + \sum_{i\in\mathcal{S}} \rho Y_i (v + V_i)\cdot b_i,
\end{aligned}$$

where ∂_t and ∇ denote differentiation with respect to time and space, respectively, ρ denotes the density, Y_i the mass fraction of the i^{th} species, v the hydrodynamic velocity, V_i the diffusion velocity of the i^{th} species, ω_i its mass production rate, $\mathcal{S} = [1, \mathrm{NS}]$ the set of species indices, NS the number of species, Π the pressure tensor, b_i the external force per unit mass acting on the i^{th} species, e the total energy per unit mass of the mixture, and q the heat flux vector. The total energy per unit mass of the mixture reads $e = \frac{1}{2} v\cdot v + u$, where u is the internal energy per unit mass of the mixture.

Using the kinetic theory of dilute polyatomic gas mixtures [1], the pressure tensor Π may be expressed as

$$\Pi = pI - (\kappa - \tfrac{2}{3}\eta)(\nabla\cdot v)I - \eta\big(\nabla v + (\nabla v)^t\big), \tag{1}$$

the species diffusion velocities V_i, $i \in \mathcal{S}$, as

$$V_i = -\sum_{j\in\mathcal{S}} D_{ij} d_j - \theta_i \nabla \log T, \qquad i \in \mathcal{S}, \tag{2}$$

or, alternatively, as

$$V_i = -\sum_{j\in\mathcal{S}} D_{ij}(d_j + \chi_j \nabla \log T), \qquad i \in \mathcal{S}, \tag{3}$$

and the heat flux vector q as

$$q = \sum_{i\in\mathcal{S}} \rho h_i Y_i V_i - \lambda' \nabla T - p \sum_{i\in\mathcal{S}} \theta_i d_i, \tag{4}$$

or, alternatively, as

$$q = \sum_{i\in\mathcal{S}} \rho h_i Y_i V_i - \lambda \nabla T + p \sum_{i\in\mathcal{S}} \chi_i V_i, \tag{5}$$

where the transport coefficients are:

η,	the shear viscosity,
κ,	the volume viscosity,
$D = (D_{ij})_{i,j\in\mathcal{S}}$,	the diffusion matrix,
$\theta = (\theta_i)_{i\in\mathcal{S}}$,	the thermal diffusion vector,
$\chi = (\chi_i)_{i\in\mathcal{S}}$,	the thermal diffusion ratios,
λ',	the partial thermal conductivity,
λ,	the thermal conductivity.

In the above equations, p denotes the thermodynamic pressure, I the identity matrix, d_i the diffusion driving force of the i^{th} species, T the absolute temperature, and h_i the enthalpy per unit mass of the i^{th} species. The vectors d_i incorporate the effects of various state variable gradients and external forces and are given by

$$d_i = \nabla X_i + (X_i - Y_i)\frac{\nabla p}{p} + \frac{\rho}{p}\sum_{j\in\mathcal{S}} Y_i Y_j (b_j - b_i), \qquad i \in \mathcal{S}, \tag{6}$$

where X_i denotes the mole fraction of the i^{th} species. It follows from (1)–(6) that detailed modeling of a polyatomic gas mixture requires the evaluation of its transport coefficients. These coefficients, in turn, are functions of the state of the mixture, as given by the variables p, T, and $Y_1, \ldots, Y_{\text{NS}}$.

Remark 1: **Physical constraints.** The transport coefficients must satisfy some mathematical properties which are important from a thermodynamic viewpoint:

(i) Symmetry of the diffusion matrix: $D_{ij} = D_{ji}$, $i, j \in \mathcal{S}$.

(ii) Mass conservation constraints: $\sum_{i\in\mathcal{S}} Y_i D_{ij} = 0$, $j \in \mathcal{S}$, $\sum_{i\in\mathcal{S}} Y_i\theta_i = 0$, and $\sum_{i\in\mathcal{S}} \chi_i = 0$. These constraints imply that the species diffusion velocities satisfy the mass conservation relation $\sum_{i\in\mathcal{S}} Y_i V_i = 0$.

(iii) Positive entropy production: the diffusion matrix D is positive semi-definite, i.e., for any vector $x = (x_i)_{i\in\mathcal{S}}$, we have $\langle Dx, x\rangle \geq 0$. Moreover, this matrix is positive definite on the hyperplane of zero sum gradients, i.e., we have $\langle Dx, x\rangle > 0$ for $x \neq 0$ and $\sum_{i\in\mathcal{S}} x_i = 0$. In addition, the shear viscosity η and the thermal conductivity λ are positive. Finally, the volume viscosity κ is positive only if there are polyatomic species in the mixture and vanishes otherwise.

The transport coefficients evaluated by EGLIB always satisfy properties (i)–(iii).

Remark 2: **Flux diffusion matrix.** EGLIB always evaluates the *flux diffusion matrix* $\widetilde{D} = (\widetilde{D}_{ij})_{i,j\in\mathcal{S}}$ formed by the flux diffusion coefficients

$$\widetilde{D}_{ij} = Y_i D_{ij}, \qquad i, j \in \mathcal{S}. \tag{7}$$

All the flux diffusion coefficients have a finite limit when some mass fractions become arbitrarily small [2]. This is not the case of the diffusion coefficient D_{ii} when Y_i becomes arbitrarily small. Another incentive for evaluating flux diffusion coefficients rather than diffusion coefficients is that the diffusion fluxes $Y_i V_i$ arising in the governing equations of gas mixtures can then be readily evaluated as

$$\begin{aligned} Y_i V_i &= -\sum_{j\in\mathcal{S}} \widetilde{D}_{ij} d_j - Y_i \theta_i \nabla \log T \\ &= -\sum_{j\in\mathcal{S}} \widetilde{D}_{ij}(d_j + \chi_j \nabla \log T), \qquad i \in \mathcal{S}. \end{aligned} \tag{8}$$

Remark 3: **Rescaled thermal diffusion ratios.** Since the thermal diffusion ratio of the i^{th} species is proportional to its mole fraction X_i, it is convenient to introduce the *rescaled thermal diffusion ratios* in such a way that

$$\chi_i = X_i \widetilde{\chi}_i, \qquad i \in \mathcal{S}. \tag{9}$$

EGLIB always evaluates the rescaled thermal diffusion ratios $\widetilde{\chi} = (\widetilde{\chi}_i)_{i\in\mathcal{S}}$ rather than the thermal diffusion ratios χ. The last term in the definition of the heat flux vector (5) can be conveniently rewritten in the form

$$\sum_{i\in\mathcal{S}} \chi_i V_i = \sum_{i\in\mathcal{S}} \frac{W}{W_i} \widetilde{\chi}_i \, Y_i V_i, \tag{10}$$

where $Y_i V_i$ is given by (8), W_i is the molar weight of the i^{th} species, and W the mean molar weight of the mixture.

Remark 4: **Flux diffusion vector.** An approximate flux diffusion matrix can sometimes be written in the form of a projected diagonal matrix

$$\widetilde{D} = P \mathrm{diag}(\widetilde{\Upsilon}) P, \tag{11}$$

where the projector matrix P has components given by $P_{ij} = \delta_{ij} - Y_i / \sum_{k\in\mathcal{S}} Y_k$ for $i, j \in \mathcal{S}$. In this situation, the vector $\widetilde{\Upsilon} = (\widetilde{\Upsilon}_i)_{i\in\mathcal{S}}$ is referred to as the flux diffusion vector. Specifically, the flux diffusion vector has components given by

$$\widetilde{\Upsilon}_i = \frac{W_i}{W} D_{i,m}, \qquad i \in \mathcal{S}. \tag{12}$$

Here, $D_{i,m}$ denotes the diffusion coefficient of the i^{th} species in the mixture

$$D_{i,m} = \frac{\sum_{j\neq i} Y_j}{\sum_{j\neq i} X_j / \mathcal{D}_{ij}}, \qquad i \in \mathcal{S}, \tag{13}$$

where $\mathcal{D}_{ij}$ is the binary diffusion coefficient of species pair (i, j). For completeness, EGLIB also offers routines evaluating the flux diffusion vector $\widetilde{\Upsilon}$. In this case, the user must apply the projector matrix P when evaluating the species diffusion velocities.

Remark 5: **Pressure independent transport coefficients.** All the transport coefficients are pressure independent except the flux diffusion matrix $\widetilde{D}$, the flux diffusion vector $\widetilde{\Upsilon}$, and the thermal diffusion vector θ which are inversely proportional to the pressure. For convenience, EGLIB also provides routines that evaluate the pressure independent transport coefficients $\rho\widetilde{D}$, $\rho\widetilde{\Upsilon}$, and $\rho\theta$. These latter quantities are particularly useful when evaluating terms such as $\rho Y_i V_i$ in the governing equations.

2.2 Transport Linear Systems

The transport coefficients in multicomponent mixtures are not given explicitly by the kinetic theory. Indeed, their evaluation requires solving linear systems, referred to as the *transport linear systems*. The transport linear systems used in the library are those obtained by the authors in [2]. They have been derived in their naturally constrained, singular, and *symmetric* form, using the formalism of Waldmann and Trübenbacher [1]. On the contrary, in an earlier work, Monchick, Yun, and Mason [9] have systematically eliminated the singularities arising in the linear systems. This *nonsymmetric* formulation of the linear systems presents three important drawbacks. First, the authors have obtained in [2] symmetric positive definite forms of the singular systems which can be inverted at a lower computational cost than that required for nonsymmetric systems. Second, the original constrained singular symmetric systems are preferable for iterative techniques. Third, the original systems have simpler analytic expressions. Finally, we point out that two misprints have been identified in the paper of Monchick, Yun, and Mason [2].

For each transport coefficient, various transport linear systems can be considered, corresponding to different choices for the polynomial expansions of the species distribution functions. While standard choices are already well-known, new transport linear systems associated with the use of variational spaces of lower dimension have been introduced recently [2,3]. These reduced systems yield new approximations for the transport coefficients which are generally within a few percent or even a fraction of a percent accuracy of the transport coefficients obtained with the standard systems. Since the reduced systems are of smaller size, they constitute in some cases a computationally interesting alternative to the standard systems. In EGLIB, the reduced systems considered in [3] are available in order to evaluate the corresponding transport coefficients.

The evaluation of the transport linear systems may account for a significant amount of the CPU time required to implement the iterative algorithms. Therefore, it is interesting to consider some simplifications in the modeling of ratios of collision integrals. This approach yields transport linear systems that can be evaluated faster and thus more cost-effective transport algorithms. In EGLIB, the option of considering particular ratios of collision integrals as temperature independent is available.

2.3 Vanishing Mass Fractions

Gas mixtures with vanishing mass fractions can be handled by the library. The limit of vanishing mass fractions has been investigated thoroughly in [2]. The main result is that the shear viscosity, the volume viscosity, the thermal conductivity, the partial thermal conductivity, the rescaled thermal diffusion ratios, the thermal diffusion coefficients, and the flux diffusion matrix tend continuously to a finite limit as some mass fractions become arbitrarily small. Based on this continuity result and to avoid artificial singularities caused by divisions by very small numbers, EGLIB uses the following procedure:

(i) Given the actual mole fractions $(X_i)_{i\in\mathcal{S}}$, evaluate the perturbed mole fractions

$$X_i^{\mathrm{tr}} = X_i + \varepsilon\left(\frac{\sum_{j\in\mathcal{S}} X_j}{\mathrm{NS}} - X_i\right), \qquad i \in \mathcal{S}, \tag{14}$$

with $\varepsilon = 10^{-16}$. Note that (14) ensures that $0 \le X_i^{\mathrm{tr}} \le 1$, $i \in \mathcal{S}$, and that $\sum_{i\in\mathcal{S}} X_i^{\mathrm{tr}} = \sum_{i\in\mathcal{S}} X_i$.

(ii) Evaluate the perturbed mean molar weight $W^{\mathrm{tr}} = \sum_{i\in\mathcal{S}} X_i^{\mathrm{tr}} W_i$ and the perturbed mass fractions

$$Y_i^{\mathrm{tr}} = \frac{W_i}{W^{\mathrm{tr}}} X_i^{\mathrm{tr}}, \qquad i \in \mathcal{S}. \tag{15}$$

(iii) Form all the transport linear systems and evaluate all the multicomponent transport coefficients using the perturbed mole fractions $(X_i^{\mathrm{tr}})_{i\in\mathcal{S}}$ and the perturbed mass fractions $(Y_i^{\mathrm{tr}})_{i\in\mathcal{S}}$.

Remark 6. As a consequence of the above procedure, the library evaluates the flux diffusion matrix

$$\widetilde{D}_{ij}^{\mathrm{tr}} = Y_i^{\mathrm{tr}} D_{ij}^{\mathrm{tr}}, \qquad i, j \in \mathcal{S}, \tag{16}$$

where $D^{\mathrm{tr}} = (D_{ij}^{\mathrm{tr}})_{i,j\in\mathcal{S}}$ is the symmetric diffusion matrix resulting from a transport linear system evaluated using the perturbed mole fractions $(X_i^{\mathrm{tr}})_{i\in\mathcal{S}}$.

3 Numerical Considerations

The transport linear systems can be solved by using either a direct numerical method or an iterative method. Direct numerical inversion is performed on the symmetric positive definite form of the transport linear systems derived by the authors [2] by constructing the LDL^t decomposition of the system matrix. On the other hand, all the transport coefficients can be expressed as convergent series, for which all the partial sums satisfy the important properties (i)–(iii) [2]. Rigorously derived, analytic, approximate expressions are then obtained for all the transport coefficients by truncation. Truncation of the series results from a compromise between computational cost and accuracy. Details of the algorithms and related performance estimates are discussed in [3].

In order to illustrate the advantage of iterative methods versus direct inversions, we compare the computational costs associated with each approach using an operation count. As per convention, we define an operation to be one multiplication plus one addition. First, the computational cost associated with a direct numerical inversion is $C_{\mathrm{DI}} = \omega^3/6 + \mathcal{O}(\mathrm{NS}^2)$ operations. Here, $\omega = \mathrm{NS}$, 2NS or 3NS is the size of the transport linear system. (For the monatomic species, dummy rows and columns are added so that ω is indeed proportional to NS [2,3].) On the other hand, the cost of performing m steps of an iterative method is $C_{m,\mathrm{IT}} = m\ \omega^2 + \mathcal{O}(\mathrm{NS})$ operations. It is, therefore, expected that an iterative method will be more cost-effective than a direct numerical inversion provided that the first few iterations already yield approximate expressions with a high enough level of accuracy. EGLIB typically performs one to three iterations yielding relative errors generally ranging from 1E-3 to 5E-2. Note that more accurate expressions are not appropriate since only a few terms have been retained in the polynomial expansions of the species distribution functions. For numerical experiments illustrating the convergence rate of iterative algorithms, we refer to [2,3].

Note that it is impossible to obtain approximate expressions rigorously derived from the kinetic theory at a cost of $\mathcal{O}(\mathrm{NS})$ operations, since each gas interacts with all the other gases present in the mixture, which yields $O(\mathrm{NS}^2)$ interactions to be taken into account. We still note that it is possible to obtain *empirical* expressions at a cost of $\mathcal{O}(\mathrm{NS})$ operations, such as the average formulas of order α given by

$$\begin{cases} \mathcal{A}_\alpha(\mu) = \big(\sum_{k\in\mathcal{S}} X_k(\mu_k)^\alpha\big)^{1/\alpha}, & \alpha \neq 0, \\ \mathcal{A}_0(\mu) = \exp\big(\sum_{k\in\mathcal{S}} X_k \log(\mu_k)\big), & \end{cases} \tag{17}$$

where the quantities μ_k, $k \in \mathcal{S}$, denote the transport coefficient obtained for each pure species [2]. EGLIB offers the possibility to evaluate the shear viscosity, the thermal conductivity, and the volume viscosity using the above empirical expressions [2].

4 General Description of the Library

4.1 Single and Multiple Input Data Routines

In numerical models, it is often necessary to evaluate transport coefficients over a wide range of grid nodes discretizing the computational domain. As a consequence, two approaches can be considered in order to evaluate transport coefficients

- ⋆ single input data routines which only consider one state of the mixture;
- ⋆ multiple input data routines which compute simultaneously a family of states of the mixture.

We restate that the state of the mixture is given by the pressure, the temperature, and the species mass fractions. Single input data routines typically return transport coefficients at a given grid node. On the other hand, multiple input data routines typically evaluate transport coefficients over the whole computational domain or over a given subdomain. As described in the next section, EGLIB offers both types of routines.

4.2 Library Structure

The library contains the initialization routine EGINI and three packages, called EGSLIB, EGMLIB, and EGFLIB. EGSLIB contains *single* input data routines which evaluate transport coefficients for a given state of the mixture. EGMLIB and EGFLIB contain *multiple* input data routines which evaluate transport coefficients for a family of states of the mixture.

The difference between EGMLIB and EGFLIB arises in the way transport coefficients of vector and matrix type are indexed. These transport coefficients can be viewed as multidimensional arrays with one index referring to the grid node and one or two indices referring to the species. In EGMLIB, the grid index is ordered after the species indices. On the contrary, in EGFLIB, the grid index is ordered before the species indices.

In each package, the name of the routines takes on the generic form EG•STRINGm (• stands for S, M, or F). Here, 'STRING' is a character string referring to the quantity being evaluated. More specifically, 'BIN' refers to $\mathcal{D}_{ij}$, 'D' refers to $\widetilde{D}$, 'DR' to $\rho\widetilde{D}$, 'E' to η, 'K' to κ, 'L' to λ, 'LCT' to λ

and $\widetilde{\chi}$, 'LTD' to λ', θ, and $\widetilde{D}$, 'LTDR' to λ', $\rho\theta$, and $\rho\widetilde{D}$, 'TD' to θ, and $\widetilde{D}$, 'TDR' to $\rho\theta$ and $\rho\widetilde{D}$, 'V' to $\widetilde{\Upsilon}$, 'VR' to $\rho\widetilde{\Upsilon}$, 'YV' to the vector with components $Y_i V_i$ for $i \in \mathcal{S}$, and 'RYV' to the vector with components $\rho Y_i V_i$ for $i \in \mathcal{S}$. In addition, the integer m refers to the method of evaluation. Increasing the integer m generally means using a more expensive algorithm and obtaining a more accurate expression for the transport coefficient(s).

The following groups of routines are available

- ⋆ routines EG•BIN evaluate the species binary diffusion coefficients $\mathcal{D}_{ij}$ for $i, j \in \mathcal{S}$ and $i \neq j$.
- ⋆ routines EG•D(R)m ($m = 1$ to 2) evaluate the flux diffusion matrix $\widetilde{D}$ or, respectively, the rescaled flux diffusion matrix $\rho\widetilde{D}$.
- ⋆ routines EG•Em ($m = 1$ to 4) evaluate the shear viscosity η.
- ⋆ routines EG•Km ($m = 1$ to 6) evaluate the volume viscosity κ.
- ⋆ routines EG•Lm ($m = 1$ to 5) evaluate the thermal conductivity λ.
- ⋆ routines EG•LCTm ($m = 1$ to 4) evaluate the thermal conductivity λ and the rescaled thermal diffusion ratios $\widetilde{\chi}$.
- ⋆ routines EG•LTD(R)m ($m = 1$ to 6) evaluate the partial thermal conductivity λ', the thermal diffusion vector θ, and the flux diffusion matrix $\widetilde{D}$ or, respectively, the partial thermal conductivity λ', the rescaled thermal diffusion vector $\rho\theta$, and the rescaled flux diffusion matrix $\rho\widetilde{D}$.
- ⋆ routines EG•TD(R)m ($m = 1$) evaluate the thermal diffusion vector θ and the flux diffusion matrix $\widetilde{D}$ or, respectively, the rescaled thermal diffusion vector $\rho\theta$ and the rescaled flux diffusion matrix $\rho\widetilde{D}$.
- ⋆ routines EG•V(R)m ($m = 1$) evaluate the flux diffusion vector $\widetilde{\Upsilon}$ or, respectively, the rescaled flux diffusion vector $\rho\widetilde{\Upsilon}$.
- ⋆ routines EG•(R)YV evaluate the vector with components $Y_i V_i$ for $i \in \mathcal{S}$ or, respectively, the vector with components $\rho Y_i V_i$ for $i \in \mathcal{S}$.

A detailed description of the routines is given in Section 6.

4.3 The Mechanism for Using the Library

The library must be initialized first by calling routine EGINI. This initialization must be done only once and at the beginning of the user's program. However, if mixtures with different chemical species are to be considered, the library has to be reinitialized before computing transport coefficients for each mixture. Two important flags are passed as input variables in routine EGINI. In addition, this routine reads various thermomolecular parameters from a binary file called Linkeg and subsequently stores them in the library work arrays. More details are given in Section 5.

After the library has been initialized, routines pertaining to *any* package EG•LIB (•=S, M, or F) can be called. Before calling one or a group of routines in a *given* package, routine EG•PAR must be called. These routines initialize the temperature and the species mass fractions that will be used to evaluate the transport coefficients. In the course of the user's program, whenever the temperature or the species mass fractions are changed, EG•PAR must be called again before evaluating new transport coefficients. No call is needed if only the pressure has changed. Once EG•PAR has been called, all the remaining routines in the corresponding package EG•LIB can be called. More details regarding routines EG•PAR are given in Section 6.1.

4.4 Optimal Implementation of the Library

Optimal implementation of the library depends on the machine architecture. The key observation [10] is that transport property evaluation only depends on the local state of the mixture. Vectorization is hence straightforward for the iterative algorithms as well as for the direct inversion procedures (recall that the latter construct the LDL^t decomposition of the system matrix which does not need any pivoting.)

Optimal vectorization of transport property evaluation thus requires multiple input data routines that compute simultaneously transport properties over a wide number of grid nodes [10].

On the other hand, parallel optimization of transport property evaluation depends on the problem granularity. For coarse-granularity parallel computations in which each processor is responsible for a different portion of the computational domain, multiple input data routines are preferable. For fine-grained distributed parallel architectures with a large number of processors, single input data routines are generally more useful.

Deciding whether to use EGMLIB or EGFLIB obviously depends on the structure of the user's code. In computers with a cache of moderate size, EGFLIB will generally yield less memory swaps than EGMLIB and might thus be more cost-effective. On the other hand, on vector supercomputers, the multiple input data packages EGMLIB and EGFLIB will generally perform much better than the single input data package EGSLIB. For more details about the relative performances, we refer to Section 7.

4.5 Physical Units

With the exception of some molecular parameters read from binary file Linkeg (see next section), all the input and output data is handled by the library in the following units: grams [g], centimeters [cm], seconds [s], and degrees Kelvin [K]. Units are thus ergs [erg $\equiv$ g.cm^2/s^2] for energy and dynes [dyne $\equiv$ g.cm/s^2] for forces.

5 Initializing the Library

The library is initialized by calling routine EGINI. This routine initializes the double precision and the integer work arrays used by the library and checks their length. Two important flags are passed as input variables to the initialization routine. In addition, this routine reads all the necessary physical parameters from a binary data file called Linkeg. These parameters are subsequently stored in the library work arrays. Only one call to routine EGINI is needed at the beginning of the user's code.

SUBROUTINE EGINI (NP, LOUT, IFLAG, ITLS, WEG, LWEG, IWEG, LIWEG)

NP	number of nodes (1 if only EGSLIB is used)
LOUT	output file number
IFLAG	flag for evaluating parameters and space allocation (see Table 1)
ITLS	flag for space allocation (see Table 1)
WEG	double precision work array for EGLIB
LWEG	length of WEG declared in user's code
IWEG	integer work array for EGLIB
LIWEG	length of IWEG declared in user's code

Among the input variables, special attention should be given to the two transport flags ITLS and IFLAG. The value of these flags depends on the routines that will be subsequently used. For a given routine, the size of the largest transport linear system considered is given by ITLS $*$ NS. On the other hand, IFLAG indicates which thermomolecular parameters are needed in order to evaluate the transport linear systems. The value of flags ITLS and IFLAG thus affects both computer time and memory. The value of these flags is set only once when calling initialization routine EGINI. This routine should be called with the highest possible values for IFLAG and ITLS as read from Table 1.

In order to determine the length required for the double precision work array WEG, the following formulas can be used. For ITLS = 1, set

$$\mathrm{LWEG} = 23 + 14 * \mathrm{NS} + 32 * \mathrm{NS}^2 + 13 * \mathrm{NP} + 14 * \mathrm{NP} * \mathrm{NS} + \mathrm{NP} * \mathrm{NS}^2. \tag{18}$$

For ITLS = 2, set

$$\begin{aligned} \mathrm{LWEG} = {} & 23 + 14 * \mathrm{NS} + 32 * \mathrm{NS}^2 + 13 * \mathrm{NP} + 21 * \mathrm{NP} * \mathrm{NS} \\ & + \mathrm{NP} * \left\{ 2 * \mathrm{NS}^2 + \frac{\mathrm{NS} * (\mathrm{NS} + 1)}{2} \right\}. \end{aligned} \tag{19}$$

Table 1. How to choose ITLS and IFLAG.

Routine	ITLS	IFLAG
EG•BIN	0	2
EG•D(R)1	1	2
EG•D(R)2	1	2
EG•E1	0	1
EG•E2	1	2
EG•E3	1	3
EG•E4	1	3
EG•K1	0	4
EG•K2	1	4
EG•K3	1	5
EG•K4	2	4
EG•K5	2	5
EG•K6	2	5
EG•L1	0	1
EG•L2	1	6
EG•L3	1	7
EG•L4	2	6
EG•L5	2	7
EG•LCT1	1	7
EG•LCT2	1	7
EG•LCT3	2	7
EG•LCT4	2	7
EG•LTD(R)1	2	7
EG•LTD(R)2	2	7
EG•LTD(R)3	3	7
EG•LTD(R)4	3	7
EG•LTD(R)5	3	7
EG•LTD(R)6	3	7
EG•TD(R)1	3	7
EG•V(R)1	0	2
EG•(R)YV	1	2

Note. The symbol '•' stands for S, M or F.

For ITLS = 3, set

$$\mathrm{LWEG} = 23 + 14*\mathrm{NS} + 32*\mathrm{NS}^2 + 13*\mathrm{NP} + 30*\mathrm{NP}*\mathrm{NS} + 5*\mathrm{NP}*\mathrm{NS}^2. \tag{20}$$

The above formulas have been obtained for IFLAG = 7 which should cover most applications of the library. The general formula for LWEG is

$$\begin{aligned}\mathrm{LWEG} = {} & 23 + 14*\mathrm{NS} + 29*\mathrm{NS}^2 + 3*\mathrm{MAX}(\mathrm{NS}^2,\mathrm{NP}) + 10*\mathrm{NP} + 5*\mathrm{NS}*\mathrm{NP}\\ & + \mathbb{I}_{\mathrm{IFLAG}\geq 2}*\mathrm{NP}*\frac{\mathrm{NS}*(\mathrm{NS}+1)}{2} + \mathrm{NP}*\frac{\mathrm{ITLS}*\mathrm{NS}*(\mathrm{ITLS}*\mathrm{NS}+1)}{2}\\ & + \mathrm{NS}*\mathrm{NP}*\Big\{2*\mathbb{I}_{\mathrm{IFLAG}\geq 4} + 5*\mathrm{ITLS} + \frac{\mathrm{ITLS}*(\mathrm{ITLS}+1)}{2}\Big\},\end{aligned} \tag{21}$$

where

$$\mathbb{I}_{\mathrm{IFLAG}\geq 2} = \begin{cases} 1 & \text{if IFLAG} \geq 2,\\ 0 & \text{otherwise,}\end{cases} \tag{22}$$

and

$$\mathbb{I}_{\mathrm{IFLAG}\geq 4} = \begin{cases} 1 & \text{if IFLAG} \geq 4,\\ 0 & \text{otherwise.}\end{cases} \tag{23}$$

On the other hand, the length required for the integer work array IWEG is

$$\mathrm{LIWEG} = \mathrm{NS}. \tag{24}$$

The following parameters are read from the binary file Linkeg

NS	number of species
NO	number of terms in polynomial fits (see below)
WT(NS)	species molar weights [g/mol]
EPS(NS)	Lennard-Jones potential well depth for the species [K]
SIG(NS)	Lennard-Jones potential diameter for the species [angström]
DIP(NS)	dipole moments of the species [debye]
POL(NS)	species polarizability [angström^3]
ZROT(NS)	species rotational relaxation number at 298K [no units]
LIN(NS)	integer flag describing the molecule
	0 monatomic
	1 linear
	2 otherwise
COFE(NO,NS)	coefficients for polynomial fits of the species shear viscosity
COFL(NO,NS)	coefficients for polynomial fits of the species thermal conductivity
COFD(NO,NS,NS)	coefficients for polynomial fits of the species binary diffusion coefficients

In addition to the library, an independent driver program generating the data file Linkeg from the CHEMKIN transport library [11] is provided. Parameters EPS, SIG, DIP, POL, ZROT, and LIN describe the collisional characteristics of the molecules. On the other hand, the fits for the pure species shear viscosities, thermal conductivities and binary diffusion coefficients take on the form of polynomial approximations with respect to the logarithm of the temperature

$$\begin{cases} \log(\eta_i) = \sum_{m=1}^{\mathrm{NO}} \mathrm{COFE}(m,i)\big(\log(T)\big)^{m-1}, & i \in \mathcal{S},\\ \log(\lambda_i) = \sum_{m=1}^{\mathrm{NO}} \mathrm{COFL}(m,i)\big(\log(T)\big)^{m-1}, & i \in \mathcal{S},\\ \log(\mathcal{D}_{ij}) = \sum_{m=1}^{\mathrm{NO}} \mathrm{COFD}(m,i,j)\big(\log(T)\big)^{m-1}, & i,j \in \mathcal{S}.\end{cases} \tag{25}$$

In the above equations, T is expressed in [K], η_i in [g/cm.s], λ_i in [erg/cm.s.K] and $\mathcal{D}_{ij}$ in [cm^2/s].

6 Initializing the Molecular Parameters

Routines EG•PAR initialize the molecular parameters needed to evaluate the transport linear systems. For more details about the physical underlying assumptions, we refer to [2,11].

EG•PAR routines must be called before using routines from the corresponding package EG•LIB. If in the course of the user's program, the temperature or the species mass fractions are changed, EG•PAR must be called again before evaluating new transport coefficients. However, if only the pressure has been changed, the routine need not be called again.

The calling sequences only contain input parameters, including the temperature, the species mass and mole fractions, and the species heat capacities at constant pressure.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSPAR (T, X, Y, CP, WEG, IWEG)

T	temperature [K]
X(NS)	species mole fractions
Y(NS)	species mass fractions
CP(NS)	species heat capacities at constant pressure per unit mass [erg/g.K]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMPAR (NP, T, X, Y, CP, WEG, IWEG)

NP	number of nodes
T(NP)	temperature [K]
X(NS,NP)	species mole fractions
Y(NS,NP)	species mass fractions
CP(NS,NP)	species heat capacities at constant pressure per unit mass [erg/g.K]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFPAR (NP, T, X, Y, CP, WEG, IWEG)

NP	number of nodes
T(NP)	temperature [K]
X(NP,NS)	species mole fractions
Y(NP,NS)	species mass fractions
CP(NP,NS)	species heat capacities at constant pressure per unit mass [erg/g.K]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB

7 Evaluating Viscosities

In this section we describe the routines available to evaluate the shear viscosity and the volume viscosity.

7.1 Routines Evaluating η

In order to evaluate the shear viscosity, the following routines are available.

1 EG•E1 implements the empirical expression (14). Note that the Wilke expression is not implemented in the library since even a direct inversion of the transport linear system is more cost effective. Transport flags should be set at least to ITLS=0 and IFLAG=1.

2 EG•E2 performs one step of the conjugate gradient method. The transport linear system is of size NS and is evaluated with temperature *independent* ratios of collision integrals. Transport flags should be set at least to ITLS=1 and IFLAG=2.

3 EG•E3 performs one step of the conjugate gradient method and uses temperature *dependent* ratios of collision integrals. The transport linear system is of size NS. Transport flags should be set at least to ITLS=1 and IFLAG=3.

4 EG•E4 performs a direct inversion of the same transport linear system as the one considered by EG•E3. Transport flags should be set at least to ITLS=1 and IFLAG=3.

⋆⋆⋆ **EGSLIB** ⋆⋆⋆

SUBROUTINE EGSE1 (ALPHA, T, X, WEG, <u>ETA</u>)

ALPHA	parameter for empirical average formula (14)
T	temperature [K]
X(NS)	species mole fractions
WEG	double precision work array for EGLIB
ETA	shear viscosity η [g/cm.s]

SUBROUTINE EGSE{2,3,4} (T, Y, WEG, <u>ETA</u>)

T	temperature [K]
Y(NS)	species mass fractions
WEG	double precision work array for EGLIB
ETA	shear viscosity η [g/cm.s]

⋆⋆⋆ **EGMLIB** ⋆⋆⋆

SUBROUTINE EGME1 (NP, ALPHA, T, X, WEG, <u>ETA</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NS,NP)	species mole fractions
WEG	double precision work array for EGLIB
ETA(NP)	shear viscosity η [g/cm.s]

SUBROUTINE EGME{2,3,4} (NP, T, Y, WEG, <u>ETA</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
ETA(NP)	shear viscosity η [g/cm.s]

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFE1 (NP, ALPHA, T, X, WEG, <u>ETA</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NP,NS)	species mole fractions
WEG	double precision work array for EGLIB
ETA(NP)	shear viscosity η [g/cm.s]

SUBROUTINE EGFE{2,3,4} (NP, T, Y, WEG, <u>ETA</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
ETA(NP)	shear viscosity η [g/cm.s]

7.2 Routines Evaluating κ

In order to evaluate the volume viscosity, the following routines are available.

<u>1</u> EG•K1 implements the empirical expression (14). Transport flags should be set at least to ITLS=0 and IFLAG=4.

<u>2</u> EG•K2 considers the transport system matrix associated with the internal energy approach discussed in [6]. This matrix is of size NS and is evaluated here with temperature *independent* ratios of collision integrals. Furthermore, the system matrix is diagonal so that the solution of the linear system is readily obtained. Transport flags should be set at least to ITLS=1 and IFLAG=4.

<u>3</u> EG•K3 considers the same transport linear system as EG•K2, but uses temperature *dependent* ratios of collision integrals. The system matrix is still diagonal so that the solution of the linear system is readily obtained. Transport flags should be set at least to ITLS=1 and IFLAG=5.

<u>4</u> EG•K4 considers the transport system matrix associated with the translational and internal energy approach discussed in [6]. The system matrix is of size 2*NS and is evaluated with temperature *independent* ratios of collision integrals. An accurate approximation for κ is obtained after one step of a standard iterative method applied to an appropriate Schur complement [6]. Transport flags should be set at least to ITLS=2 and IFLAG=4.

<u>5</u> EG•K5 considers the same transport linear system and the same iterative algorithm as EG•K4, but uses temperature *dependent* ratios of collision integrals. Transport flags should be set at least to ITLS=2 and IFLAG=5.

<u>6</u> EG•K6 considers the same transport linear system as EG•K5, but performs a direct inversion in order to evaluate κ. Transport flags should be set at least to ITLS=2 and IFLAG=5.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSK1 (ALPHA, T, X, WEG, <u>KAPPA</u>)

ALPHA	parameter for empirical average formula (14)
T	temperature [K]
X(NS)	species mole fractions
WEG	double precision work array for EGLIB
KAPPA	volume viscosity κ [g/cm.s]

SUBROUTINE EGSK{2,3,4,5,6} (T, Y, WEG, <u>KAPPA</u>)

T	temperature [K]
Y(NS)	species mass fractions

WEG	double precision work array for EGLIB
KAPPA	volume viscosity κ [g/cm.s]

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMK1 (NP, ALPHA, T, X, WEG, <u>KAPPA</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NS,NP)	species mole fractions
WEG	double precision work array for EGLIB
KAPPA(NP)	volume viscosity κ [g/cm.s]

SUBROUTINE EGMK{2,3,4,5,6} (NP, T, Y, WEG, <u>KAPPA</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
KAPPA(NP)	volume viscosity κ [g/cm.s]

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFK1 (NP, ALPHA, T, X, WEG, <u>KAPPA</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NP,NS)	species mole fractions
WEG	double precision work array for EGLIB
KAPPA(NP)	volume viscosity κ [g/cm.s]

SUBROUTINE EGFK{2,3,4,5,6} (NP, T, Y, WEG, <u>KAPPA</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
KAPPA(NP)	volume viscosity κ [g/cm.s]

8 Evaluating Thermal Conduction and Thermal Diffusion

In this section we describe the routines available to evaluate the thermal conductivity alone or the thermal conductivity and the rescaled thermal diffusion ratios.

8.1 Routines Evaluating λ Only

In order to evaluate the thermal conductivity only, the following routines are available.

<u>1</u> EG•L1 implements the empirical expression (14). Transport flags should be set at least to ITLS=0 and IFLAG=1.

<u>2</u> EG•L2 considers the transport system matrix associated with the internal energy approach discussed in [4]. The system matrix is of size NS and is evaluated here with temperature *independent* ratios of collision integrals. An accurate approximation for λ is obtained after one step of the conjugate gradient method. Transport flags should be set at least to ITLS=1 and IFLAG=6.

3 EG•L3 considers the same transport linear system as EG•L2, but uses temperature *dependent* ratios of collision integrals. Transport flags should be set at least to ITLS=1 and IFLAG=7.

4 EG•L4 considers the transport system matrix associated with the translational and internal energy approach discussed in [4]. The system matrix is of size 2*NS and is evaluated here with temperature *independent* ratios of collision integrals. An accurate approximation for λ is obtained after one step of the conjugate gradient method. Transport flags should be set at least to ITLS=2 and IFLAG=6.

5 EG•L5 considers the same transport linear system as EG•L4, but uses temperature *dependent* ratios of collision integrals. Transport flags should be set at least to ITLS=2 and IFLAG=7.

⋆⋆⋆ **EGSLIB** ⋆⋆⋆

SUBROUTINE EGSL1 (ALPHA, T, X, WEG, <u>TC</u>)

ALPHA	parameter for empirical average formula (14)
T	temperature [K]
X(NS)	species mole fractions
WEG	double precision work array for EGLIB
TC	thermal conductivity λ [erg/cm.s.K]

SUBROUTINE EGSL{2,3,4,5} (T, Y, WEG, IWEG, <u>TC</u>)

T	temperature [K]
Y(NS)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC	thermal conductivity λ [erg/cm.s.K]

⋆⋆⋆ **EGMLIB** ⋆⋆⋆

SUBROUTINE EGML1 (NP, ALPHA, T, X, WEG, <u>TC</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NS,NP)	species mole fractions
WEG	double precision work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]

SUBROUTINE EGML{2,3,4,5} (NP, T, Y, WEG, IWEG, <u>TC</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]

⋆⋆⋆ **EGFLIB** ⋆⋆⋆

SUBROUTINE EGFL1 (NP, ALPHA, T, X, WEG, <u>TC</u>)

NP	number of nodes
ALPHA	parameter for empirical average formula (14)
T(NP)	temperature [K]
X(NP,NS)	species mole fractions
WEG	double precision work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]

SUBROUTINE EGFL{2,3,4,5} (NP, T, Y, WEG, IWEG, TC)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]

8.2 Routines Evaluating λ and $\widetilde{\chi}$

In order to evaluate the thermal conductivity and the rescaled thermal diffusion ratios, the following routines are available.

1 EG•LCT1 considers the transport system matrix associated with the internal energy approach discussed in [4]. The system matrix is of size NS and is evaluated with temperature *dependent* ratios of collision integrals. An accurate approximation for λ and $\widetilde{\chi}$ is obtained after three steps of the conjugate gradient method. Transport flags should be set at least to ITLS=1 and IFLAG=7.

2 EG•LCT2 considers the same transport linear system as EG•LCT1, but performs a direct inversion. Transport flags should be set at least to ITLS=1 and IFLAG=7.

3 EG•LCT3 considers the transport system matrix associated with the translational and internal energy approach discussed in [4]. The system matrix is of size 2*NS and is evaluated with temperature *dependent* ratios of collision integrals. An accurate approximation for λ and $\widetilde{\chi}$ is obtained after three steps of the conjugate gradient method. Transport flags should be set at least to ITLS=2 and IFLAG=7.

4 EG•LCT4 considers the same transport linear system as EG•LCT3, but performs a direct inversion. Transport flags should be set at least to ITLS=2 and IFLAG=7.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSLCT{1,2,3,4} (T, Y, WEG, IWEG, TC, TDR)

T	temperature [K]
Y(NS)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC	thermal conductivity λ [erg/cm.s.K]
TDR(NS)	rescaled thermal diffusion ratios $\widetilde{\chi}$ [no units]

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMLCT{1,2,3,4} (NP, T, Y, WEG, IWEG, TC, TDR)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]
TDR(NS,NP)	rescaled thermal diffusion ratios $\widetilde{\chi}$ [no units]

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFLCT{1,2,3,4} (NP, T, Y, WEG, IWEG, TC, TDR)

NP	number of nodes
T(NP)	temperature [K]

Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TC(NP)	thermal conductivity λ [erg/cm.s.K]
TDR(NP,NS)	rescaled thermal diffusion ratios $\widetilde{\chi}$ [no units]

9 Evaluating Multicomponent Diffusion

In this section we describe the routines related to multicomponent diffusion. They include routines evaluating:
- the flux diffusion vector,
- directly the diffusion fluxes,
- the flux diffusion matrix,
- the species binary diffusion coefficients.

When the user wants to account for off diagonal effects in the flux diffusion matrix, we strongly recommend the routines evaluating directly the diffusion fluxes (section 9.2) rather than those evaluating the flux diffusion matrix (section 9.3). Indeed, with the latter routines, the user needs to implement the product of the flux diffusion matrix by the species diffusion driving forces. The routines evaluating the flux diffusion matrix are useful only if one is interested in the actual values of the coefficients of the flux diffusion matrix.

9.1 Routines Evaluating $\widetilde{\Upsilon}$ or $\rho\widetilde{\Upsilon}$

Routines EG•V(R)1 consider the first order transport system matrix, i.e., translational and internal energy of the molecules are neglected. The system matrix is of size NS. Transport flags should be set at least to ITLS=0 and IFLAG=2.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSV1 (PRES, T, Y, WW, WEG, <u>DV</u>)

PRES	pressure [dyne/cm^2]
T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
DV(NS)	flux diffusion vector $\widetilde{\Upsilon}$ [cm^2/s]

SUBROUTINE EGSVR1 (T, Y, WEG, <u>DV</u>)

T	temperature [K]
Y(NS)	species mass fractions
WEG	double precision work array for EGLIB
DV(NS,NS)	rescaled flux diffusion vector $\rho\widetilde{\Upsilon}$ [g/cm.s]

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMV1 (NP, PRES, IPRES, T, Y, WW, WEG, <u>DV</u>)

NP	number of nodes
PRES($\star$)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB

DV(NS,NP)	flux diffusion vector $\widetilde{\Upsilon}$ [cm^2/s]

SUBROUTINE EGMVR1 (NP, T, Y, WEG, <u>DV</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
DV(NS,NP)	rescaled flux diffusion vector $\rho\widetilde{\Upsilon}$ [g/cm.s]

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFV1 (NP, PRES, IPRES, T, Y, WW, WEG, <u>DV</u>)

NP	number of nodes
PRES(⋆)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, ≠0: node dependent)
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
DV(NP,NS)	flux diffusion vector $\widetilde{\Upsilon}$ [cm^2/s]

SUBROUTINE EGFVR1 (NP, T, Y, WEG, <u>DV</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
DV(NP,NS)	rescaled flux diffusion vector $\rho\widetilde{\Upsilon}$ [g/cm.s]

9.2 Routines Evaluating Directly the Diffusion Fluxes or the Rescaled Diffusion Fluxes

Routines EG•(R)YV take as input the vector formed by the species diffusion driving forces. They compute internally the transport linear system corresponding to the first order diffusion coefficients, i.e., translational and internal energy of the molecules are neglected, and form the Choleski decomposition of the system matrix. Finally, these routines evaluate the diffusion fluxes or the rescaled diffusion fluxes. The output vector is *overwritten* on the input vector. Transport flags should be set at least to ITLS=1 and IFLAG=2.

An important input parameter for routines EG•(R)YV is IDDEC. This flag should be set to zero if the Choleski decomposition of the system matrix has already been computed. In multidimensional problems, routines EG•(R)YV should be called with IDDEC set to one (or any nonzero value) when diffusion driving forces in the first spatial coordinate are considered and subsequently set to zero for the remaining spatial coordinates.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSYV (PRES, Y, WW, WEG, <u>YV</u>, IDDEC)

PRES	pressure [dyne/cm^2]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
YV(NS)	input: species diffusion driving forces $(d_i)_{i\in\mathcal{S}}$ [cm^{-1}]
	output: diffusion fluxes $(Y_iV_i)_{i\in\mathcal{S}}$ with $Y_iV_i = -\sum_{j\in\mathcal{S}} \widetilde{D}_{ij}d_j$ [cm/s]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, ≠0: no)

SUBROUTINE EGSRYV (T, Y, WEG, YV, IDDEC)

T	temperature [K]
Y(NS)	species mass fractions
WEG	double precision work array for EGLIB
YV(NS)	input: species diffusion driving forces $(d_i)_{i \in \mathcal{S}}$ [cm^{-1}] output: rescaled diffusion fluxes $(\rho Y_i V_i)_{i \in \mathcal{S}}$ with $\rho Y_i V_i = -\sum_{j \in \mathcal{S}} \rho \widetilde{D}_{ij} d_j$ [$\mathrm{g/cm^2.s}$]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, $\neq$0: no)

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMYV (NP, PRES, IPRES, Y, WW, WEG, YV, IDDEC)

NP	number of nodes
PRES(⋆)	pressure [$\mathrm{dyne/cm^2}$]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
YV(NS,NP)	input: species diffusion driving forces $(d_i)_{i \in \mathcal{S}}$ [cm^{-1}] output: diffusion fluxes $(Y_i V_i)_{i \in \mathcal{S}}$ with $Y_i V_i = -\sum_{j \in \mathcal{S}} \widetilde{D}_{ij} d_j$ [cm/s]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, $\neq$0: no)

SUBROUTINE EGMRYV (NP, T, Y, WEG, YV, IDDEC)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WEG	double precision work array for EGLIB
YV(NS,NP)	input: species diffusion driving forces $(d_i)_{i \in \mathcal{S}}$ [cm^{-1}] output: rescaled diffusion fluxes $(\rho Y_i V_i)_{i \in \mathcal{S}}$ with $\rho Y_i V_i = -\sum_{j \in \mathcal{S}} \rho \widetilde{D}_{ij} d_j$ [$\mathrm{g/cm^2.s}$]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, $\neq$0: no)

⋆⋆⋆ EGFLIB ⋆⋆⋆

SUBROUTINE EGFYV (NP, PRES, IPRES, Y, WW, WEG, YV, IDDEC)

NP	number of nodes
PRES(⋆)	pressure [$\mathrm{dyne/cm^2}$]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
YV(NP,NS)	input: species diffusion driving forces $(d_i)_{i \in \mathcal{S}}$ [cm^{-1}] output: diffusion fluxes $(Y_i V_i)_{i \in \mathcal{S}}$ with $Y_i V_i = -\sum_{j \in \mathcal{S}} \widetilde{D}_{ij} d_j$ [cm/s]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, $\neq$0: no)

SUBROUTINE EGFRYV (NP, T, Y, WEG, YV, IDDEC)

NP	number of nodes
T(NP)	temperature [K]

Y(NP,NS)	species mass fractions
WEG	double precision work array for EGLIB
YV(NP,NS)	input: species diffusion driving forces $(d_i)_{i\in\mathcal{S}}$ [cm^{-1}] output: rescaled diffusion fluxes $(\rho Y_i V_i)_{i\in\mathcal{S}}$ with $\rho Y_i V_i = -\sum_{j\in\mathcal{S}} \rho \widetilde{D}_{ij} d_j$ [g/cm^2.s]
IDDEC	flag indicating whether the Choleski decomposition of the system matrix is already available (0: yes, $\neq$0: no)

9.3 Routines Evaluating $\widetilde{D}$ or $\rho\widetilde{D}$

In order to evaluate the flux diffusion matrix, the following routines are available.

1 EG•D(R)1 use two steps of a projected standard iterative method. The transport linear system corresponds to the first order diffusion coefficients, i.e., translational and internal energy of the molecules are neglected. The system matrix is of size NS. Transport flags should be set at least to ITLS=1 and IFLAG=2.

2 EG•D(R)2 perform a direct numerical inversion. The transport linear system is the same as the one considered by routines EG•D(R)1. Transport flags should be set at least to ITLS=1 and IFLAG=2.

★★★ **EGSLIB** ★★★

SUBROUTINE EGSD{1,2} (PRES, T, Y, WW, WEG, D)

PRES	pressure [dyne/cm^2]
T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
D(NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGSDR{1,2} (T, Y, WW, WEG, D)

T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
D(NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

★★★ **EGMLIB** ★★★

SUBROUTINE EGMD{1,2} (NP, PRES, IPRES, T, Y, WW, WEG, D)

NP	number of nodes
PRES($\star$)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
D(NS,NS,NP)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGMDR{1,2} (NP, T, Y, WW, WEG, D)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]

WEG	double precision work array for EGLIB
D(NS,NS,NP)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

$\star\star\star$ **EGFLIB** $\star\star\star$

SUBROUTINE EGFD{1,2} (NP, PRES, IPRES, T, Y, WW, WEG, D)

NP	number of nodes
PRES($\star$)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, ≠0: node dependent)
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
D(NP,NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGFDR{1,2} (NP, T, Y, WW, WEG, D)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
D(NP,NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

9.4 Routines Evaluating the Species Binary Diffusion Coefficients

In order to evaluate the species binary diffusion coefficients, the following routines are available. For these routines, transport flags should be set at least to ITLS=0 and IFLAG=2.

$\star\star\star$ **EGSLIB** $\star\star\star$

SUBROUTINE EGSBIN (T, WEG, BIN)

T	temperature [K]
WEG	double precision work array for EGLIB
BIN(NS,NS)	array of species binary diffusion coefficients [cm^2/s]
	only the values BIN(i,j) for $i \neq j$ are returned

$\star\star\star$ **EGMLIB** $\star\star\star$

SUBROUTINE EGMBIN (NP, T, WEG, BIN)

NP	number of nodes
T(NP)	temperature [K]
WEG	double precision work array for EGLIB
BIN(NS,NS,NP)	array of species binary diffusion coefficients [cm^2/s]
	only the values BIN(i,j,n) for $i \neq j$ are returned

$\star\star\star$ **EGFLIB** $\star\star\star$

SUBROUTINE EGFBIN (NP, T, WEG, BIN)

NP	number of nodes
T(NP)	temperature [K]
WEG	double precision work array for EGLIB
BIN(NP,NS,NS)	array of species binary diffusion coefficients [cm^2/s]
	only the values BIN(n,i,j) for $i \neq j$ are returned

10 Other routines

10.1 Routines Evaluating λ', θ, and $\widetilde{D}$ or λ', $\rho\theta$, and $\rho\widetilde{D}$

In order to evaluate the partial thermal conductivity, the thermal diffusion vector, and the flux diffusion matrix, the following routines are available.

1 EG•LTD(R)1 consider the transport system matrix obtained with the internal energy approach discussed in [4]. The system matrix is of size 2*NS and is evaluated with temperature *dependent* ratios of collision integrals. An accurate approximation for λ' and θ is obtained after three steps of the conjugate gradient method. For $\widetilde{D}$, two steps of a projected standard iterative method are performed. Transport flags should be set at least to ITLS=2 and IFLAG=7.

2 EG•LTD(R)2 consider the same transport linear systems as EG•LTD(R)1. The transport coefficients are evaluated by performing a direct inversion of the corresponding linear systems. Transport flags should be set at least to ITLS=2 and IFLAG=7.

3 EG•LTD(R)3 consider two transport system matrices. For λ' and θ, the matrix obtained with the translational and internal energy approach discussed in [4] is used. This matrix is of size 3*NS and is evaluated with temperature *dependent* ratios of collision integrals. For $\widetilde{D}$, the first order transport system matrix of size NS is considered, i.e., translational and internal energy of the molecules are neglected. An accurate approximation for λ' and θ is obtained after three steps of the conjugate gradient method. For $\widetilde{D}$, two steps of a projected standard iterative method are performed. Transport flags should be set at least to ITLS=3 and IFLAG=7.

4 EG•LTD(R)4 consider the same transport linear systems as EG•LTD(R)3. The transport coefficients are evaluated by performing a direct inversion of the corresponding linear systems. Transport flags should be set at least to ITLS=3 and IFLAG=7.

5 EG•LTD(R)5 consider the transport system matrix associated with the translational and internal energy approach discussed in [4]. The system matrix is of size 3*NS and is evaluated with temperature *dependent* ratios of collision integrals. An accurate approximation for λ' and θ is obtained after three steps of the conjugate gradient method. For $\widetilde{D}$, two steps of a projected standard iterative method are performed. Transport flags should be set at least to ITLS=3 and IFLAG=7.

6 EG•LTD(R)6 consider the same transport linear systems as EG•LTD(R)5. The transport coefficients are evaluated by performing a direct inversion of the corresponding linear systems. Transport flags should be set at least to ITLS=3 and IFLAG=7.

★★★ **EGSLIB** ★★★

SUBROUTINE EGSLTD{1,2,3,4,5,6} (PRES, T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

PRES	pressure [dyne/cm^2]
T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
PTC	partial thermal conductivity λ' [erg/cm.s.K]
TD(NS)	thermal diffusion vector θ [cm^2/s]
D(NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGSLTDR{1,2,3,4,5,6} (T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB

PTC	partial thermal conductivity λ' [erg/cm.s.K]
TD(NS)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

⋆⋆⋆ **EGMLIB** ⋆⋆⋆

SUBROUTINE EGMLTD{1,2,3,4,5,6} (NP, PRES, IPRES, T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

NP	number of nodes
PRES(⋆)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
PTC(NP)	partial thermal conductivity λ' [erg/cm.s.K]
TD(NS,NP)	thermal diffusion vector θ [cm^2/s]
D(NS,NS,NP)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGMLTDR{1,2,3,4,5,6} (NP, T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
PTC(NP)	partial thermal conductivity λ' [erg/cm.s.K]
TD(NS,NP)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NS,NS,NP)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

⋆⋆⋆ **EGFLIB** ⋆⋆⋆

SUBROUTINE EGFLTD{1,2,3,4,5,6} (NP, PRES, IPRES, T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

NP	number of nodes
PRES(⋆)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
PTC(NP)	partial thermal conductivity λ' [erg/cm.s.K]
TD(NP,NS)	thermal diffusion vector θ [cm^2/s]
D(NP,NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGFLTDR{1,2,3,4,5,6} (NP, T, Y, WW, WEG, IWEG, <u>PTC</u>, <u>TD</u>, <u>D</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
PTC(NP)	partial thermal conductivity λ' [erg/cm.s.K]

TD(NP,NS)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NP,NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

10.2 Routines Evaluating $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{D}}$ or $\boldsymbol{\rho\theta}$ and $\boldsymbol{\rho}\widetilde{\boldsymbol{D}}$ Only

Routines EG•TD(R)1 consider two transport system matrices. For θ, the matrix obtained with the translational and internal energy approach discussed in [4] is used. The system matrix is of size 3*NS and is evaluated with temperature *dependent* ratios of collision integrals. For $\widetilde{D}$, the first order transport system matrix of size NS is considered, i.e., translational and internal energy of the molecules are neglected. An accurate approximation for θ is obtained after three steps of the conjugate gradient method. For $\widetilde{D}$, two steps of a projected standard iterative method are performed. Transport flags should be set at least to ITLS=3 and IFLAG=7.

⋆⋆⋆ EGSLIB ⋆⋆⋆

SUBROUTINE EGSTD1 (PRES, T, Y, WW, WEG, IWEG, <u>TD</u>, <u>D</u>)

PRES	pressure [dyne/cm^2]
T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NS)	thermal diffusion vector θ [cm^2/s]
D(NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGSTDR1 (T, Y, WW, WEG, IWEG, <u>TD</u>, <u>D</u>)

T	temperature [K]
Y(NS)	species mass fractions
WW	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NS)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

⋆⋆⋆ EGMLIB ⋆⋆⋆

SUBROUTINE EGMTD1 (NP, PRES, IPRES, T, Y, WW, WEG, IWEG, <u>TD</u>, <u>D</u>)

NP	number of nodes
PRES($\star$)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NS,NP)	thermal diffusion vector θ [cm^2/s]
D(NS,NS,NP)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGMTDR1 (NP, T, Y, WW, WEG, IWEG, <u>TD</u>, <u>D</u>)

NP	number of nodes
T(NP)	temperature [K]
Y(NS,NP)	species mass fractions
WW(NP)	mean molar weight [g/mol]

WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NS,NP)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NS,NS,NP)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

$\star\star\star$ **EGFLIB** $\star\star\star$

SUBROUTINE EGFTD1 (NP, PRES, IPRES, T, Y, WW, WEG, IWEG, TD, D)

NP	number of nodes
PRES($\star$)	pressure [dyne/cm^2]
IPRES	flag for pressure (0: node independent, $\neq$0: node dependent)
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NP,NS)	thermal diffusion vector θ [cm^2/s]
D(NP,NS,NS)	flux diffusion matrix $\widetilde{D}$ [cm^2/s]

SUBROUTINE EGFTDR1 (NP, T, Y, WW, WEG, IWEG, TD, D)

NP	number of nodes
T(NP)	temperature [K]
Y(NP,NS)	species mass fractions
WW(NP)	mean molar weight [g/mol]
WEG	double precision work array for EGLIB
IWEG	integer work array for EGLIB
TD(NP,NS)	rescaled thermal diffusion vector $\rho\theta$ [g/cm.s]
D(NP,NS,NS)	rescaled flux diffusion matrix $\rho\widetilde{D}$ [g/cm.s]

11 Performance Estimates

In this section we illustrate the numerical performances of EGLIB. Results have been obtained on a C98 Cray computer—in either scalar or vector mode—, a C3 Convex computer in vector mode, an IBM RS6000 scalar workstation, and a HP750 scalar workstation. For brevity, only a few representative routines are considered, but similar results are obtained for the other ones as well. For more details, we refer to [12].

We first investigate vectorization of the library on the Cray C98. We consider an hydrogen mixture of 9 chemical species with concentrations and temperature varying over 2,500 grid nodes. In Table 2 we report the vector speedup $\mathcal{S}_{\rm V}$ defined as the ratio of the execution time in scalar mode divided by the execution time in vector mode. EGSLIB vectorizes very poorly since vector speedups between 1 and 2 are obtained. On the other hand, vector speedups for EGMLIB and EGFLIB are generally of the order of 20. The above results clearly show the effectiveness of multiple input data libraries on vector supercomputers.

In Table 3 we report the ratio of the execution time obtained with EGSLIB divided by that obtained with EGMLIB on several computers. Similar results are obtained with EGFLIB, but are omitted for brevity. On the Cray C98 with vector optimization, EGMLIB runs consistently more than 20 times faster. The Convex C3 also performs vector optimization and the ratio of the execution times varies between 2.6 and 9.0. The results are somewhat different on the IBM RS6000 and the HP750 workstations which are scalar machines. On the IBM workstation, EGMLIB is more cost-effective than EGSLIB except for routine EG$\bullet$D1. On the other hand, EGSLIB performs better than EGMLIB on the HP workstation. These differences, due to the larger memory requirements of the multiple input data libraries, are linked to machine-dependent, cache swapping issues.

Table 2.
Vector speedups obtained on the Cray C98.

Coefficient	η		κ		λ,$\widetilde{\chi}$		λ',θ,$\widetilde{D}$		$\widetilde{D}$	
Routine EG•	E3	E4	K5	K6	LCT3	LCT4	LTD5	LTD6	D1	D2
$T_{\mathrm{S}}/T_{\mathrm{V}}$ (S)	1.2	1.2	1.1	1.2	1.2	1.2	1.3	1.7	1.7	1.5
$T_{\mathrm{S}}/T_{\mathrm{V}}$ (M)	20.7	21.2	20.1	23.3	22.1	21.8	23.5	21.4	23.0	21.8
$T_{\mathrm{S}}/T_{\mathrm{V}}$ (F)	21.1	21.4	20.3	23.3	21.8	22.0	22.9	21.8	18.3	19.7

Table 3.
Performance of EGMLIB with respect to EGSLIB on various computers.

Coefficient	η		κ		λ,$\widetilde{\chi}$		λ',θ,$\widetilde{D}$		$\widetilde{D}$	
Routine EG•	E3	E4	K5	K6	LCT3	LCT4	LTD5	LTD6	D1	D2
C98 vect	27.7	32.5	28.1	30.4	25.2	29.0	23.3	21.4	18.5	29.7
Convex C3	8.3	8.3	9.0	7.6	6.5	7.2	5.2	4.8	2.6	5.4
IBM RS6000	1.5	1.5	1.9	1.7	1.5	1.7	0.8	1.3	0.3	0.6
HP750	1.0	0.9	1.0	0.8	0.7	0.7	0.6	0.8	0.7	1.0

Table 4.
Performance of EGLIB with respect to existing transport software
on various computers.

Coefficient	η		λ',θ,$\widetilde{D}$		$\widetilde{D}$		$\widetilde{\Upsilon}$
Routine EG•	E3	E4	LTD3	LTD4	D1	D2	V1
C98 vect	23.3	19.7	72.0	28.0	83.2	45.9	27.0
Convex C3	5.9	4.3	20.0	12.4	14.8	9.5	7.2
IBM RS6000	5.9	4.6	6.2	4.1	6.1	2.5	1.6
HP750	2.3	1.5	10.6	2.0	7.3	1.9	1.8

Finally, we compare the present transport library to the one discussed in [11]. In this latter package, the shear viscosity η is approximated using the Wilke empirical expression, while a direct inversion of a nonsymmetric form of the transport linear systems is used to evaluate λ',θ,$\widetilde{D}$ or $\widetilde{D}$ alone. For the flux diffusion matrix, only the first order approximation is considered in [11]. It is also possible to evaluate the flux diffusion vector $\widetilde{\Upsilon}$. In Table 4, we report the ratios of the resulting execution times divided by those obtained with the present library on several computers. On scalar workstations such as the IBM or the HP workstations, speedups range from 1.5 to 10.6. On vector computers such as the Convex C3 or the Cray C98, where multiple input data routines take full advantage of vectorization capabilities, speedups range from 4.3 to 83.2.

12 Example of Routines for Combustion Applications

Multicomponent transport effects on various flame structures have been investigated in [13,14]. These studies were concerned with freely propagating planar laminar flames, counterflow premixed flames and multidimensional Bunsen flames. The fuel was either hydrogen or methane. Equivalence ratios were varied from lean to rich conditions. In these studies, the following routines were used:
- EG•E3 for the shear viscosity;
- EG•LCT1 for the thermal conductivity and the rescaled thermal diffusion ratios;
- EG•RYV for the species diffusion fluxes.

The volume viscosity was not evaluated since the flows were in the low Mach number regime.

Acknowledgments. The authors are grateful to Prof. M.D. Smooke and Dr. M.A. Tanoff (Yale University) for stimulating discussions. They are also thankful to the Institut du Développement et des Ressources Informatiques et Scientifiques, France, and to the Service d'Etudes Techniques de Routes et Autoroutes, France, for providing part of the computational resources.

References

[1] L. Waldmann und E. Trübenbacher, Formale kinetische Theorie von Gasgemischen aus anregbaren Molekülen, *Zeitschr. Naturforschg.* **17a**, 363–376 (1962).

[2] A. Ern and V. Giovangigli, *Multicomponent Transport Algorithms*, Lecture Notes in Physics, New Series Monographs, **m 24**, (Springer-Verlag, Heidelberg, 1994).

[3] A. Ern and V. Giovangigli, Fast and accurate multicomponent transport property evaluation, *J. Comput. Phys.* **120**, 105–116 (1995).

[4] A. Ern and V. Giovangigli, Thermal conduction and thermal diffusion in dilute polyatomic gas mixtures, *Physica A* **214**, 526–546 (1995).

[5] A. Ern and V. Giovangigli, On the evaluation of thermal diffusion coefficients in chemical vapor deposition processes, *J. Chem. Vapor Depos.* **3**, 3–31 (1995).

[6] A. Ern and V. Giovangigli, Volume viscosity of dilute polyatomic gas mixtures, *European J. Mech., B/Fluids* **14**, 653–669 (1995).

[7] A. Ern and V. Giovangigli, Projected iterative algorithms with application to multicomponent transport, *Linear Algebra Appl.* **250**, 289–315 (1997).

[8] A. Ern and V. Giovangigli, The structure of transport linear systems in dilute isotropic gas mixtures, *Phys. Rev. E* **53**, 485–492 (1996).

[9] L. Monchick, K. S. Yun, and E. A. Mason, Formal kinetic theory of transport phenomena in polyatomic gas mixtures, *J. Chem. Phys.* **39**, 654–669 (1963).

[10] V. Giovangigli and N. Darabiha, Vector computers and complex chemistry combustion, in *Proceedings Conference Mathematical Modeling in Combustion and Related Topics*, C. Brauner and C. Schmidt-Lainé eds., NATO Adv. Sci. Inst. Ser. E, **140**, 491–503 (Martinus Nijhoff, Dordrecht, 1988).

[11] 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).

[12] A. Ern and V. Giovangigli, Optimized transport algorithms for flame codes, *Combust. Sci. Tech.* **118**, 387–395 (1996).

[13] A. Ern et V. Giovangigli, Thermal diffusion effects in hydrogen/air and methane/air flames, *Combust. Theory Model.*, **2**, 349–372 (1998).

[14] A. Ern et V. Giovangigli, Impact of detailed multicomponent transport on planar and counterflow hydrogen/air and methane/air flames, *Combust. Sci. and Tech.*, **149**, 157–181 (1999).