VECTOR COMPUTERS AND COMPLEX CHEMISTRY COMBUSTION

V. GIOVANGIGLI †‡ and N. DARABIHA ‡

‡ Laboratoire EM2C du CNRS et de l'ECP, Ecole Centrale des Arts et Manufactures, 92295 Chatenay-Malabry cedex, France.
† Laboratoire d'Acoustique et Mécanique, UA868, Université Paris 6, T66 5Et., 4 place Jussieu, 75252 Paris cedex 05, France.

1. INTRODUCTION

Combustion chemistry usually involves many interacting species and many elementary reactions. Typically, tens of species and hundreds of reversible chemical reactions are used in hydrocarbon combustion kinetics. As a result, the governing conservation equations for total mass, species mass, momentum and energy, in a reactive flow, are large and stiff systems of partial differential equations. A consequence is that only zero and one dimensional models have formerly been used to look in detail at the coupling of a large number of species interactions [8] [13] [17] [19] [20] [22].

Among the numerical problems for computational solutions are those associated with multiple time and space scales and the costs due to complex chemistry. In addition, there are problems not directly related to chemistry such as complicated geometries or linear algebra which can be investigated separately. To overcome numerical difficulties arising from multiple time and space scales one has to use time-implicit algorithms and adaptive gridding techniques respectively [13] [17] [21]. In this paper we focus on computational costs inherent in complex chemistry due to multiple sums and products performed when evaluating thermodynamic properties, chemical production rates and multicomponent transport coefficients.

Use of a vector computer can significantly reduce these costs provided the corresponding algorithms vectorize. This problem is addressed in the present paper where high degrees of vectorization are obtained by structuring properly the corresponding multiple nested loops.

The formulas for evaluating chemical production rates, thermodynamic properties and diffusion coefficients are problem independent, and two packages, CHEMKIN and TRANSPORT, have been written at Sandia national laboratories by Kee et al. for evaluating these quantities [14] [15]. Since the vectorized algorithms are also problem independent, although the choice of an optimum algorithm may depend on various problem parameters like the number of species or the number of grid points, the resulting subroutines can easily be embeded in the CHEMKIN and TRANSPORT packages, which were originally written for scalar machines.

Finally, to test the resulting vectorized subroutines we have computed various flame structures on a Cray-1 computer, characterized by a vector/scalar speed ratio between 5.-10., and we have found vector/scalar performances around 6.-8. for the routines and 4. for the flame solver.

2. A DETAILED COMBUSTION MODEL

2.1 Governing equations

The governing equations of a gaseous laminar reacting flow are the equations for conservation of total mass, species mass, momemtum and energy, and the equation of state. The total mass conservation can be written :

$$\frac{\partial \rho}{\partial t} + \nabla . (\rho v) = 0, \tag{1}$$

where ρ is the density, v the mass averaged flow velocity, t the time, and the species mass conservation equation is :

$$\rho \frac{\partial Y_k}{\partial t} + \rho(v.\nabla)Y_k = -\nabla . (\rho Y_k V_k) + W_k \omega_k, \qquad k = 1, \ldots, K, \tag{2}$$

where Y_k is the mass fraction of the kth species, V_k the diffusion velocity of the kth species, W_k the molecular weight of the kth species, ω_k the molar production rate of the kth species and K the number of species. The momentum conservation equation is :

$$\rho\frac{\partial v}{\partial t}+\rho(v.\nabla)v=-\nabla p+\nabla.\Big\{\eta\Big(\nabla v+(\nabla v)^T-\tfrac{2}{3}(\nabla.v)\mathbf{I}\Big)\Big\}+\sum_{k=1}^{K}\rho Y_k f_k, \qquad (3)$$

where p is the pressure, η the viscosity and f_k the external force per unit mass of the k^{th} species and the energy conservation equation is :

$$\rho c_p\frac{\partial T}{\partial t}+\rho c_p(v.\nabla)T=\nabla.(\lambda\nabla T)-\Big(\sum_{k=1}^{K}\rho Y_k V_k c_{pk}\Big).\nabla T-\sum_{k=1}^{K}h_k W_k\omega_k+\frac{\partial p}{\partial t}+(v.\nabla)p$$
$$+\eta\Big(\nabla v+(\nabla v)^T-\tfrac{2}{3}(\nabla.v)\mathbf{I}\Big):\nabla v+\sum_{k=1}^{K}\rho Y_k V_k.f_k, \qquad (4)$$

where T is the temperature, c_p the constant pressure heat capacity of the mixture, c_{pk} the constant pressure heat capacity of the k^{th} species, λ the thermal conductivity and h_k the specific enthalpy of the k^{th} species. Finally the equation of state is

$$\rho=pW/RT, \qquad (5)$$

where W is the molecular weight of the mixture and R the universal gas constant.

Equations (1)-(5), which may include other effects like radiative heat losses, etc., are usually further simplified according to the problem under study, e.g., steady flow, boundary layer flow, low mach number flow, zero dimensional model, etc.. These equations have to be completed by formulas expressing the transport coefficients λ and η, the diffusion velocities V_k, the thermodynamic properties c_p, c_{pk} and h_k and the chemical production rates ω_k in terms of the state variables T, p, Y_k, $k=1,\ldots,K$, and their gradients. These relations are written in the next sections.

2.2 Chemistry and thermodynamics

We consider I elementary reversible reactions involving K chemical species, which can be represented in the general form :

$$\sum_{k=1}^{K}\nu'_{ki}\mathcal{X}_k \rightleftharpoons \sum_{k=1}^{K}\nu''_{ki}\mathcal{X}_k, \qquad i=1,\ldots,I, \qquad (6)$$

where the stoichiometric coefficients ν'_{ki} and ν''_{ki} are integers and $\mathcal{X}_k$ denotes the symbol of the k^{th} species. The production rate of the k^{th} species can be written as :

$$\omega_k=\sum_{i=1}^{I}\nu_{ki}q_i, \qquad \nu_{ki}=\nu''_{ki}-\nu'_{ki}, \qquad (7)$$

where q_i is the rate of progress variable of the i^{th} reaction. The rate of progress q_i is the difference between the forward and reverse rates :

$$q_i=\mathcal{K}_{fi}\prod_{k=1}^{K}[\mathcal{X}_k]^{\nu'_{ki}}-\mathcal{K}_{ri}\prod_{k=1}^{K}[\mathcal{X}_k]^{\nu''_{ki}}, \qquad (8)$$

where $[\mathcal{X}_k]$ is the molar concentration of the k^{th} species and $\mathcal{K}_{fi}$ and $\mathcal{K}_{ri}$ are the forward and reverse rate constants of the i^{th} reaction. Denoting X_k the mole fraction of the k^{th} species the quantities X_k, Y_k and $[\mathcal{X}_k]$ are related by :

$$X_k=Y_k\frac{W}{W_k}, \qquad [\mathcal{X}_k]=X_k\frac{p}{RT}, \qquad W=\Big(\sum_{k=1}^{K}\frac{Y_k}{W_k}\Big)^{-1} \qquad (9)$$

The forward rate constant $\mathcal{K}_{fi}$ for the i^{th} reaction is given by Arrhenius' law :

$$\mathcal{K}_{fi} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{RT}\right), \tag{10}$$

where A_i is the pre-exponential factor, β_i the temperature exponent and E_i the activation energy. The reverse rate constant is related to the forward constant through the equilibrium constant $\mathcal{K}_{ei}$:

$$\mathcal{K}_{ri} = \frac{\mathcal{K}_{fi}}{\mathcal{K}_{ei}}, \qquad \mathcal{K}_{ei} = \left(\frac{p_{atm}}{RT}\right)^{\Delta\nu_i} \exp(\Delta Z_i), \tag{11}$$

$$\Delta\nu_i = \sum_{k=1}^{K} \nu_{ki}, \qquad \Delta Z_i = \sum_{k=1}^{K} \nu_{ki} Z_k, \qquad Z_k = \frac{S_k^0}{R} - \frac{H_k}{RT}, \tag{12}$$

where p_{atm} denotes the atmospheric pressure and H_k and S_k^0 the molar enthalpy and entropy of the k^{th} species at the standard state of one atmosphere, respectively. In some reactions an arbitrary third body, usually denoted M, is required for the reaction to proceed. In this situation the rate of progress variable becomes

$$q_i = ctb_i\left(\mathcal{K}_{fi}\prod_{k=1}^{K}[\mathcal{X}_k]^{\nu'_{ki}} - \mathcal{K}_{ri}\prod_{k=1}^{K}[\mathcal{X}_k]^{\nu''_{ki}}\right), \qquad ctb_i = \left(\sum_{k=1}^{K} \alpha_{ki}\,[\mathcal{X}_k]\right) \tag{13}$$

where the α_{ki} are one if all the species contribute equally as third bodies, in which case the extra factor of Eq. (13), —the third body concentration ctb_i, is the total concentration $[M]$:

$$[M] = p/RT. \tag{14}$$

However, if the species act more or less efficiently as third bodies the α_{ki} may differ from one. Finally, some hydrocarbon reactions have pressure dependent rate coefficients. These rate coefficients can be estimated either from the Lindemann formula

$$\mathcal{K}_{fi} = A_i T^{\beta_i} \exp(-\frac{E_i}{RT}) \Big/ \left(1 + \frac{A'_i T^{\beta'_i} \exp(-\frac{E'_i}{RT})}{[M]}\right), \tag{15}$$

where A'_i, β'_i and E'_i are constants, or from more involved formulas like the quantum mechanical Kassel sum [24].

Thermodynamic properties are usually estimated from polynomial fits of the JANAF data. As a concrete exemple we will consider the Nasa fits of [9] also used in CHEMKIN, but any other type of fits would lead to similar vectorized algorithms. In these packages molar constant pressure heat capacities are taken in the form of fourth degree polynomial fits for two temperature ranges :

$$\frac{C_{pk}}{R} = \begin{cases} a_{1k} + a_{2k}T + a_{3k}T^2 + a_{4k}T^3 + a_{5k}T^4, & \text{if } T_{\text{low}} \le T \le T_{\text{mid}}, \\ a_{8k} + a_{9k}T + a_{10k}T^2 + a_{11k}T^3 + a_{12k}T^4, & \text{if } T_{\text{mid}} \le T \le T_{\text{high}}, \end{cases} \tag{16}$$

with similar expressions for the corresponding molar enthalpy H_k and molar standard state entropy at one atmosphere S_k^0 . Moreover the midpoint temperature is allowed to be species dependent in the CHEMKIN package [14] :

$$T_{\text{mid}} = T_{\text{mid}k}. \tag{17}$$

Finally, the thermodynamic properties in mass units and the mixture constant pressure heat capacity are easily obtained from :

$$c_{pk} = \frac{C_{pk}}{W_k}, \qquad h_k = \frac{H_k}{W_k}, \qquad c_p = \sum_{k=1}^{K} Y_k c_{pk}. \tag{18}$$

2.3 Transport properties

Comparisons between different mathematical approximations of the multicomponent transport properties have shown that the following expressions provide a good trade off between precision and computational costs [3] [15] [19] [25]. The diffusion velocity is divided into three parts :

$$V_k = \mathcal{V}_k + \mathcal{W}_k + V_{cor}, \tag{19}$$

where $\mathcal{V}_k$ is the diffusion velocity due to species gradients and is given in the Hirschfelder-Curtiss approximation [10] by

$$\mathcal{V}_k = -D_k \frac{1}{X_k} \nabla X_k, \qquad D_k = (1 - Y_k) \Big/ \sum_{l \neq k} X_l / \mathcal{D}_{kl}, \tag{20}$$

where the $\mathcal{D}_{kl}$, $k = 1, \dots, K$, $l = 1, \dots, K$, are the binary diffusion coefficients. The velocity $\mathcal{W}_k$ is due to temperature gradients and is included only for low molecular weight species like H and H_2. The trace light component limit is used for $\mathcal{W}_k$ and leads to

$$\mathcal{W}_k = \frac{D_k \Theta_k}{X_k} \frac{1}{T} \nabla T, \qquad \Theta_k = \sum_{l \neq k} \theta_{kl}, \tag{21}$$

where the θ_{kl} can be expressed in terms of collision integrals [2]. The velocity V_{cor} is a correction velocity included to insure that the mass is conserved [3] [15] [17] :

$$V_{cor} = -\sum_{k=1}^{K} Y_k (\mathcal{V}_k + \mathcal{W}_k). \tag{22}$$

The mixture thermal conductivity λ can be estimated either by the simple semi-empirical formula

$$\lambda = \frac{1}{2} \{ \sum_{k=1}^{K} X_k \lambda_k + (\sum_{k=1}^{K} X_k \lambda_k^{-1})^{-1} \}, \tag{23}$$

where λ_k is the thermal conductivity of the k^{th} species, or by the more expensive Wilke formula :

$$\lambda = \sum_{k=1}^{K} \frac{X_k \lambda_k}{\sum_{l=1}^{K} X_l \phi_{kl}}, \qquad \phi_{kl} = \frac{1}{\sqrt{8}} \Big(1 + \frac{W_k}{W_l} \Big)^{-\frac{1}{2}} \Big(1 + (\frac{\eta_k}{\eta_l})^{\frac{1}{2}} (\frac{W_l}{W_k})^{\frac{1}{4}} \Big)^2, \tag{24}$$

where η_k is the viscosity of the k^{th} species, and the mixture viscosity η can be estimated with similar expressions [3] [15] [17].

The transport coefficients η_k, λ_k, $\mathcal{D}_{kl}$ and θ_{kl} have to be expressed in terms of the state variables p, T, $Y_k, k = 1, \dots, K$, and molecular parameters. The corresponding formulas, which can be found in [2] [11] [15], appear to be complex. However, while their dependence on the molecular parameters is indeed complex, their dependence on the temperature is relatively simple. As a consequence, it is possible to compute polynomial fits of the temperature dependent part of single component viscosities η_k and conductivities λ_k, binary diffusion coefficients $\mathcal{D}_{kl}$ and reduced thermal diffusion coefficients $\theta_{kl}/X_k X_l$ [3] [15] [17] [25]. This procedure leads to significant gains in computational efficiency with little or no compromise in accuracy.

As for the Thermodynamic properties, we will consider the polynomial fits of the TRANSPORT package [15] as a concrete exemple, but using any other type of fits would lead to similar vectorized algorithms. These fits are in the form

$$\log \eta_k = b_{1k} + b_{2k} \log T + b_{3k} (\log T)^2 + b_{4k} (\log T)^3, \qquad k = 1, \dots, K, \tag{25}$$

$$\log \lambda_k = c_{1k} + c_{2k} \log T + c_{3k} (\log T)^2 + c_{4k} (\log T)^3, \qquad k = 1, \dots, K, \tag{26}$$

$$\log \mathcal{D}_{kl} = d_{1kl} + d_{2kl} \log T + d_{3kl} (\log T)^2 + d_{4kl} (\log T)^3, \qquad \begin{cases} k = 1, \dots, K, \\ l = 1, \dots, K, \end{cases} \tag{27}$$

$$(\theta_{kl}/X_k X_l) = e_{1kl} + e_{2kl} T + e_{3kl} T^2 + e_{4kl} T^3, \qquad \begin{cases} k \in \mathcal{L}, \\ l = 1, \dots, K, \end{cases} \tag{28}$$

where $\mathcal{L} = \{light(l);\, l = 1, \dots, LIGHT\}$ denotes the indices of the light species, the fitting being done at the beginning of the calculation.

3. EVALUATION OF AERO-THERMO-CHEMISTRY EXPRESSIONS

3.1 Computational considerations

Examination of Eqs. (1)-(5) shows that, regardless of the particular problem under study, it is necessary to evaluate the thermodynamic properties such as c_p, c_{pk}, $k = 1, \dots, K$, h_k, $k = 1, \dots, K$, the chemical production rates ω_k, $k = 1, \ldots, K$, and the transport coefficients such as D_k, $k = 1, \ldots, K$, $\mathcal{D}_{kl}$, $k = 1, \ldots, K$, $l = 1, \ldots, K$, etc., in terms of the state variables T, p and Y_k, $k = 1, \ldots, K$. But evaluating these aero-thermo-chemistry quantities is expensive since they involve multiple sums and products as shown in the previous section. Typically, up to 70-95% of the CPU time may be spent for these evaluations in a one dimensional laminar flame solver [21].

Using a vector processor, e.g., Cray-1, CDC-205, Fujitsu VP200, NEC SX2, Hitachi S-820, FPS 264, etc., or a multiple vector processor, e.g., Cray X-MP, Cray-2, etc., can significantly reduce these computational costs, provided the corresponding algorithms vectorize. This problem is investigated in the following. We refer to [12] [16] [18] for more details on vectorization techniques. Altough different groups can have investigated this type of problem, there are few papers on the subject, e.g., Boris and Winsor [1]. The paper of Boris and Winsor, however, investigates the vectorization of reactive flow solvers based on timestep splitting, rather than the more specific problem of evaluating thermodynamic properties, chemical production rates and transport coefficients.

To vectorize subroutines for evaluating these aero-thermo-chemistry quantities, two fundamentally different situations have to be considered. Indeed, keeping in mind that in a flow field all these quantities only depend on the local state variables T, p and Y_k, $k = 1, \ldots, K$, and since most combustion problems are at least one dimensional, one has in general to evaluate these quantities for a large number of input data, each of these corresponding to a grid point, regardless of the grid structure. Nevertheless, the case of a single input data is still interesting for instance if an homogeneous problem is under study, say with a huge reaction mechanism, or if a grid point needs a special treatment. In particular, on a multiprocessor computer, both cases are of interest depending on how many grid points are treated per processor. These two different situations require different treatments and, to distinguish between them, the single imput data case is denoted SID and the multiple input data case MID. Moreover, if we note that the subscripts k or l are always used for the species and the subscript i for the chemical reactions, we introduce the subscript j, $j = 1, \ldots, J$, to index the different input conditions, J being the number of them. In the following, whenever the MID case is considered, we will add the subscript j to every quantity like c_{pj}, c_{pkj}, h_{kj}, ω_{kj}, D_{kj}, $\mathcal{D}_{klj}$, etc, which depend on the j^{th} input data T_j, p_j and Y_{kj}, $k = 1, \ldots, K$. Further note that the CHEMKIN and TRANSPORT libraries only consider the SID case. Anticipating the results of the next sections, we point out that the best performances for the MID subroutines are obtained when inner loops are j loops, that is to say loops in j, and the resulting routines are significantly faster than the corresponding SID routines used J times, provided J is not too small.

3.2 Vectorization of chemistry and thermodynamics

3.2.1 Evaluation of thermodynamic properties In the SID case, the simplest way to compute single component thermodynamic properties, e.g., C_{pk}/R, $k = 1, \ldots, K$, is to do a loop in k and to use explicitly the polynomial fits, such that † :

```
do $k = 1$, $K$
    if $T \le T_{\text{mid}k}$ then
        $C_{pk}/R = a_{1k} + a_{2k}T + a_{3k}T^2 + a_{4k}T^3 + a_{5k}T^4$
    else                                                              (29)
        $C_{pk}/R = a_{8k} + a_{9k}T + a_{10k}T^2 + a_{11k}T^3 + a_{12k}T^4$
    endif
enddo
```

Although introducing an extra loop for the evaluation of the polynomial fits would allow the user-friendly possibility of easily changing the degree of the fits, the faster unrolled algorithm (29) seems more appropiate since changing of thermodynamic data base is infrequent. The unrolling of the fitting polynomial save loads and more operation can be overlaped. It also minimize the number of loop setups, even in scalar mode. We further note that it is the dependence of the intermediate range temperature $T_{\text{mid}k}$ on the species k (17) that introduces a conditional loop. With a species independent intermediate temperature T_{mid} [9], the if statements could be move outside the loop. We remark also that using only one fit over a wide temperature range would clearly be more convenient as pointed out by Boris and Winsor [1]. Conditional loops are vectorized on a Cray-1 with mask vectors, but (29) has to be rewritten using non ANSI FORTRAN statements.

† Loops are sketched in a symbolic way.

In the MID case, a conditional loop is still needed since T_j has to be compared to $T_{\mathrm{mid}k}$. Since J is usually larger than K it is better to put the j loops as the inner loops :

```
do $k = 1, K$
    do $j = 1, J$
        if $T_j \leq T_{\mathrm{mid}k}$ then
            $C_{pkj}/R = a_{1k} + a_{2k}T_j + a_{3k}T_j^2 + a_{4k}T_j^3 + a_{5k}T_j^4$
        else                                                                    (30)
            $C_{pkj}/R = a_{8k} + a_{9k}T_j + a_{10k}T_j^2 + a_{11k}T_j^3 + a_{12k}T_j^4$
        endif
    enddo
enddo
```

Of course the evaluation of c_{pk}, h_k, $Z_k = (S_k^0/R) - (H_k/RT)$, etc., is similar.

In the SID case the mixture thermodynamic properties, e.g., c_p, are then obtained with one inner product (18). Keeping in mind that the number of species K is generally lower than 100, this type of scalar reduction loop slows down the computer speed in vector mode, even if an optimized module is used. However, in the MID case, no scalar reduction loops are introduced so that no optimized modules are needed when the j loops are the inner loops and the k loops are the outer loops. Note also that in this situation the outer k loop could be partially unrolled into the inner j loop.

3.2.2 Evaluation of chemical production rates Before computing ω_k, $k = 1, \ldots, K$, one has to evaluate the rate of progress q_i, $i = 1, \ldots, I$, of each reaction. This can be done with an i loop. However, difficulties arise due to the products Π_{fi} and Π_{ri} in (8), to the sums ΔZ_i in (12), and to third body concentrations ctb_i in (13) :

$$\Pi_{fi} = \prod_{k=1}^{K} [\mathcal{X}_k]^{\nu'_{ki}}, \quad \Pi_{ri} = \prod_{k=1}^{K} [\mathcal{X}_k]^{\nu''_{ki}}, \quad \Delta Z_i = \sum_{k=1}^{K} \nu_{ki} Z_k, \quad ctb_i = \sum_{k=1}^{K} \alpha_{ki}\, [\mathcal{X}_k]. \tag{31}$$

Indeed the number of reactants or products in a chemical reaction never exeeds three and rarely two since most three body reactions are treated differently (13), so that the matrix ν_{ki}, $k = 1, \ldots, K$, $i = 1, \ldots, I$, is quite sparse. As a consequence, forming the products Π_{fi} and Π_{ri} or the sums ΔZ_i with (31) would be a great waste of time, regardless of the algorithm. The formulas (31) have to be replaced by

$$\Pi_{fi} = \prod_{k \in \mathcal{R}(i)} [\mathcal{X}_k]^{\nu'_{ki}}, \qquad \Pi_{ri} = \prod_{k \in \mathcal{P}(i)} [\mathcal{X}_k]^{\nu''_{ki}}, \qquad \Delta Z_i = -\sum_{k \in R(i)} \nu'_{ki} Z_k + \sum_{k \in P(i)} \nu''_{ki} Z_k, \tag{32}$$

where $\mathcal{R}(i) = \{kreac(i,l);\ l = 1, \ldots, KREAC(i)\}$ denote the indices of the reactants the i^{th} reaction for which $\nu'_{ki} \neq 0$ and and $\mathcal{P}(i) = \{kprod(i,l);\ l = 1, \ldots, KPROD(i)\}$ the indices of the products the i^{th} reaction for which $\nu''_{ki} \neq 0$ respectively. As already pointed out the number of reactants $KREAC(i)$ or products $KPROD(i)$ of the reactions rarely exeeds two and never three so that the k loops over $\mathcal{R}(i)$ and $\mathcal{P}(i)$ of (32) have to be unrolled in an i loop. However the reactant indices $kreac(i,l)$, $l = 1, \ldots, KREAC(i)$, and the product indices $kprod(i,l)$, $l = 1, \ldots, KPROD(i)$, of the i^{th} reaction are arbitrary since kinetic mechanisms have no special structures, and their numbers $KREAC(i)$ and $KPROD(i)$ are varying with the reaction index i. Now on a CRAY-1 indirect addressing vector loops can only be obtained by using gathering-scattering macro-instructions, temporary arrays and loop splitting [18]. Furthermore variable-length loop unrolling can be achieved by introducing dummy species whose stoichiometric coefficients are zero in such a way that there are always three reactants and products in each reaction. Following these ideas, we introduce an extended set of stoichiometric coefficients n_1, n_2, n_3 and m_1, m_2, m_3 such that :

$$n_{li} = \begin{cases} \nu'_{kreac(i,l)i}, & \text{if } 1 \leq l \leq KREAC(i), \\ 0, & \text{if } KREAC(i) < l \leq 3, \end{cases} \qquad m_{li} = \begin{cases} \nu''_{kprod(i,l)i}, & \text{if } 1 \leq l \leq KPROD(i), \\ 0, & \text{if } KPROD(i) < l \leq 3, \end{cases} \tag{33}$$

and we set arbitrary values for $kreac(i,l)$ and $kprod(i,l)$ for $KREAC(i) < l \leq 3$ and $KPROD(i) < l \leq 3$ respectively. Then, if the molar concentrations of the reactants and products are gathered into the temporary arrays R_1, R_2, R_3, P_1, P_2 and P_3 :

$$R_{li} = \left[\mathcal{X}_{kreac(i,l)}\right], \qquad P_{li} = \left[\mathcal{X}_{kprod(i,l)}\right], \qquad l = 1, 2, 3, \tag{34}$$

and if the quantities $Z_k = (S_k^0/R) - (H_k/RT)$ of the reactants and products are gathered into the temporary arrays ZR_1, ZR_2, ZR_3, ZP_1, ZP_2 and ZP_3 :

$$ZR_{li} = Z_{kreac(i,l)}, \qquad ZP_{li} = Z_{kprod(i,l)}, \qquad l = 1,2,3, \tag{35}$$

then Π_{fi}, Π_{ri} and ΔZ_i can subsequently be estimated from

$$\Pi_{fi} = R_{1i}^{n_{1i}} R_{2i}^{n_{2i}} R_{3i}^{n_{3i}}, \qquad \Pi_{ri} = P_{1i}^{m_{1i}} P_{2i}^{m_{2i}} P_{3i}^{m_{3i}}, \tag{36}$$

$$\Delta Z_i = -n_{1i} ZR_{1i} - n_{2i} ZR_{2i} - n_{3i} ZR_{3i} + m_{1i} ZP_{1i} + m_{2i} ZP_{2i} + m_{3i} ZP_{3i}, \tag{37}$$

Note that (34)-(35) has to be rewritten with gathering non ANSI FORTRAN statements on a Cray-1. The extra cost due to the dummy variables in (36)-(37) is compensated by the vector speedup. Of course, the quantities $[\mathcal{X}_k]$, $k = 1, \ldots, K$, have to be evaluated with a simple k loop, and the quantities $Z_k = (S_k^0/R) - (H_k/RT)$, $k = 1, \ldots, K$, have to be evaluated like (29), before (34)-(35).

On the other hand, if $ithb$ is a flag concerning third bodies such that $ithb = 0$ if the i^{th} reaction does not contain arbitrary third bodies, $ithb = 1$ if it does with unity efficiencies and $ithb = 2$ if some efficiency differs from unity and if we denote $kthb(i,l)$, $l = 1, \ldots, KTHB(i)$, the species whose efficiency α_{ki} in the i^{th} reaction is not unity, then third body concentrations ctb_i can be obtained from

```
do $i = 1, I$
    $ctb_i = 1$
    if $ithb(i) \geq 1$ then
        $ctb_i = [M]$                                                    (38)
    endif
    if $ithb(i) = 2$ then
        do $l = 1, KTHB(i)$
            $ctb_i = ctb_i + (\alpha_{kthb(i,l)i} - 1)\,[\mathcal{X}_{kthb(i,l)}]$
        enddo
    endif
enddo
```

Indeed, only a few third body efficiencies differ from one, so that vectorizing the innermost loop of (38) is not interesting. For the same reasons, only a small number of the chemical reactions are such that $ithb(i) = 2$, so that the true rate of the corresponding test is small and the use of a mask vector is not interesting. Only the two first statements are worth vectorizing by splitting the i loop.

Now the rate of progress q_i, $i = 1, \ldots, I$, can easily be obtained with an i loop :

```
do $i = 1, I$
    $\mathcal{K}_{fi} = A_i T^{\beta_i} \exp\left(-\frac{E_i}{RT}\right)$
    $\mathcal{K}_{ri} = \mathcal{K}_{fi} \left(\frac{p_{atm}}{RT}\right)^{-\Delta\nu_i} \exp(+n_{1i}ZR_{1i} + n_{2i}ZR_{2i} + n_{3i}ZR_{3i}$
                $-m_{1i}ZP_{1i} - m_{2i}ZP_{2i} - m_{3i}ZP_{3i})$                        (39)
    $q_i = ctb_i \left(\mathcal{K}_{fi} R_{1i}^{n_{1i}} R_{2i}^{n_{2i}} R_{3i}^{n_{3i}} - \mathcal{K}_{ri} P_{1i}^{m_{1i}} P_{2i}^{m_{2i}} P_{3i}^{m_{3i}}\right)$
enddo
```

Note that the Lindeman form (15) or the irreversibility condition, i.e., $\mathcal{K}_{ri} = 0$, could easily be included. Finally, the chemical production rates ω_k, initialized to zero, are obtained with one scalar i loop :

```
do $i = 1, I$
    $\omega_{kreac(i,1)} = \omega_{kreac(i,1)} - n_{1i} q_i$
    $\omega_{kreac(i,2)} = \omega_{kreac(i,2)} - n_{2i} q_i$                    (40)
    ............................
    $\omega_{kprod(i,3)} = \omega_{kprod(i,3)} + m_{3i} q_i$
enddo
```

since none of the statements in (40) can vectorize [16] even by using scattering macro instructions, due to the fact that a species can be involved in different reactions. Note that only a few K and I temporary arrays are needed in the preceding algorithms.

In the MID case, everything is simpler. All the j loops have to be put as inner loops. In this situation, the number of operations is minimized, the vector lengths are maximized, and all the loops vectorize. The corresponding algorithm can be sketched as follows. First, the molar concentrations $[\mathcal{X}_{kj}]$, $k = 1, \ldots, K$, $j = 1, \ldots, J$, and the quantities Z_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, J$, are evaluated and stored in $K{*}J$ temporary arrays, and the production rates ω_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, J$, are initialized to zero. Then begins an outer loop over the chemical reactions, followed by the initialization to one, one and zero, of temporarry arrays of length J used to store the Π_{fij}, Π_{rij} and ΔZ_{ij}, respectively. The molar concentration products Π_{fij} of the reactants of the $i^{\rm th}$ reaction are then evaluated together with the reactants' contribution to the sums ΔZ_{ij} for every $j = 1, \ldots, J$, :

```
do $l = 1,\ KREAC(i)$
    do $j = 1,\ J$
        $\Pi_{fij} = \Pi_{fij}\,[\mathcal{X}_{kreac(i,l)j}]^{\nu'_{kreac(i,l)i}}$
        $\Delta Z_{ij} = \Delta Z_{ij} - \nu'_{kreac(i,l)i} Z_{kreac(i,l)j}$          (41)
    enddo
enddo
```

The calculation of the Π_{rij} and of the contribution of the products of the $i^{\rm th}$ reaction to the sums ΔZ_{ij} is then done as (41) for every $j = 1, \ldots, J$. Note that the number of operations for calculating Π_{fij}, Π_{rij} and ΔZ_{ij} is minimized, at variance with the SID case where dummy species are introduced. The third body concentrations for the $i^{\rm th}$ reaction ctb_{ij} are evaluated for every $j = 1, \ldots, J$, using only vector loops, which contrasts with the SID case :

```
if $ithb(i) = 0$ then
    do $j = 1,\ J$
        $ctb_{ij} = 1$
    enddo
elseif $ithb(i) \geq 1$ then
    do $j = 1,\ J$
        $ctb_{ij} = [M_j]$          (42)
    enddo
endif
if $ithb(i) = 2$ then
    do $l = 1,\ KTHB(i)$
        do $j = 1,\ J$
            $ctb_{ij} = ctb_{ij} + (\alpha_{kthb(i,l)i} - 1)\,[\mathcal{X}_{kthb(i,l)j}]$
        enddo
    enddo
endif
```

The rate of progress of the $i^{\rm th}$ reaction is then easily obtained for every $j = 1, \ldots, J$. Writing the corresponding algorithm is straigthforward. Finally the contribution of the $i^{\rm th}$ reaction to the chemical production rates of its reactants is evaluated from :

```
do $l = 1,\ KREAC(i)$
    do $j = 1,\ J$
        $\omega_{kreac(i,1)j} = \omega_{kreac(i,1)j} - \nu'_{kreac(i,l)i} q_{ij}$          (43)
    enddo
enddo
```

followed by the analogous to (43) for the products of the $i^{\rm th}$ reaction, and the loop over the chemical reactions ends. Note that the irreducible scalar kernel of (40) disappears in (43) and its analogous for the products, and that only two $K * J$ temporary arrays are needed, one for the $[\mathcal{X}_{kj}]$ and one for the Z_{kj}, together with some J temporary arrays.

3.3 Vectorization of transport properties

3.3.1 Evaluation of λ_k and η_k These evaluations are similar to those of c_{pk}, with an inner k loop in the SID case and an outer k loop with an inner j loop in the MID case, explicitely using the fits (25)-(26). Writing the corresponding algorithms is straightforward.

3.3.2 Evaluation of λ and η The vectorization of semi-empirical formulas like (23) is similar to that of (18), so that we will not discuss it again. The vectorization of the Wilke formula (24) leads to the following algorithm, where sum_k, $k = 1, \ldots, K$, denotes a temporary array initialized to zero :

```
do $l = 1, K$
    do $k = 1, K$
        $sum_k = sum_k + X_l \frac{1}{\sqrt{8}} \left(1 + \frac{W_k}{W_l}\right)^{-\frac{1}{2}} \left(1 + (\frac{\eta_k}{\eta_l})^{\frac{1}{2}} (\frac{W_l}{W_k})^{\frac{1}{4}}\right)^2$
    enddo                                                                 (44)
enddo
do $k = 1, K$
    $\lambda = \lambda + X_k \lambda_k / sum_k$
enddo
```

An optimized modules can also be used for the last loop which is a scalar reduction loop. In the MID case, as usual, the j loops are put as inner loops and writing the corresponding algorithm is straightforward.

3.3.3 Evaluation of $\mathcal{D}_{kl}$ and D_k In the SID case the binary diffusion coefficients can be evaluated in a (k, l) loop :

```
do $(k, l) = (1, 1), (K, K)$
    $\mathcal{D}_{kl} = \exp(d_{1kl} + d_{2kl} \log T + d_{3kl} (\log T)^2 + d_{4kl} (\log T)^3)$      (45)
enddo
```

Although recent compilers, e.g., Fujitsu VP200, directly vectorize (45) as a (k, l) loop of length $K{*}K$, it is necessary to rewrite it as a one-dimensional loop on a Cray-1. On the other hand, exploiting the natural symmetry of the binary diffusion coefficients leads to the the following algorithm :

```
do $k = 1, K$
    do $l = k+1, K$
        $\mathcal{D}_{kl} = \exp(d_{1kl} + d_{2kl} \log T + d_{3kl} (\log T)^2 + d_{4kl} (\log T)^3)$     (46)
        $\mathcal{D}_{lk} = \mathcal{D}_{kl}$
    enddo
enddo
```

which has to be used together with a compiler directive forcing the vectorization inhibited by a formal recursivity. Now, in scalar mode, exploiting the symmetry of the $\mathcal{D}_{kl}$ is always faster than (45) which approximately doubles the number of operations. This is true in vector mode if the number of interacting species K is large enough. On a Cray-1 this minimum number of species is between 20-30.

In the SID case, the mixture diffusion coefficients D_k, $k = 1, \ldots, K$, can then be obtained from the following algorithm, where sum_k, $k = 1, \ldots, K$, denotes a temporary array initialized to zero :

```
do $l = 1, K$
    do $k = 1, K$
        if $k \neq l$ then                                                  (47)
            $sum_k = sum_k + X_l / \mathcal{D}_{kl}$
        endif
    enddo
enddo
do $k = 1, K$
    $D_k = (1 - Y_k) / sum_k$
enddo
```

and where the conditional loop has to be rewritten with a non ANSI FORTRAN statement. Note also that D_k is not defined for a pure species mixture so that some care must be taken in this latter case. A procedure introduced by Kee, Warnatz and Miller [15] consists in computing

$$D_k = \frac{1}{W} \sum_{l \neq k} (X_l + \epsilon) W_l \Big/ \sum_{l \neq k} (X_l + \epsilon) / \mathcal{D}_{kl}, \tag{48}$$

where ϵ is a small positive constant. The corresponding extra cost is small.

In the MID case the evaluation of $\mathcal{D}_{klj}$ must be done with j loops as inner loops in (46), and that of D_{kj} with j loops as inner loops of (47). The extra cost introduced by (48) is small. Writing the corresponding algorithms is straightforward. Note however that if only the D_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, J$, are needed, it is not necessary to introduce a $K{*}K{*}J$ temporary array to store the $\mathcal{D}_{klj}$, which can be a problem for small memory computers and large chemical networks, since one can evaluate the $\mathcal{D}_{klj}$ only where they are needed. Note also that the choice of the fits (25)-(28) is not without consequences. For instance, the $(1/2)K*(K-1)*J$ exponentials needed for evaluating the $\mathcal{D}_{klj}$ are fairly expensive, especially in scalar mode. Finally, writing similar algorithms for the evaluation of the thermal diffusion ratio Θ_k is an easy task.

4. NUMERICAL EXPERIMENTS

In this section, performances of the SID and MID routines and of a laminar flame solver are presented. All tests are performed with Hydrogen-Air and Propane-Air mixtures. The kinetic scheme for Hydrogen-Air mixtures involves $K = 9$ species and $I = 19$ reactions and is taken from Miller and Kee [23]. For Propane-Air mixtures we have used a $K = 33$ species and $I = 126$ reactions reaction scheme due to Warnatz [24]. All tests have been done on a Cray-1 computer with the CFT 1.11 compiler.

Furthermore, the SID and MID routines have been written to use the work arrays of the CHEMKIN and TRANSPORT packages' subroutines. These work arrays store information about the species, the kinetic mechanism, the transport properties and they contain some work space. Larger work arrays were necessary however to store the extended set of stoichiometric coefficients and species names (33), and for the larger work space needed, especially for the MID routines. Up to these modifications, SID and MID routines may be embeded in the libraries of these packages. In particular, the excellent CHEMKIN FORTRAN code which reads symbolic descriptions of reaction mechanisms, the 'Interpreter', and the TRANSPORT FORTRAN code which computes the polynomial fits (25)-(28) can be used with the SID and MID routines. We refer to [14], [15] for more details.

4.1 Methodology

To measure the performance of a given program, we have used two types of tests. The first one is related to Amdhal's law which states that if P is the vector/scalar performance of a code,—the ratio of the execution times T_s and T_v with the vector mode inactive and active respectively, if S is the ratio of the average rates S_v and S_s at which the computer process the code in vector mode and scalar, i.e., sequential, mode respectively, —sometimes called the vector speedup :

$$P = T_s/T_v, \qquad S = S_v/S_s, \tag{49}$$

and if α is the the fraction of the code executed in vector mode, i.e., in the pipelined functional units, — also called the vectorization ratio, then we have :

$$P = 1\Big/\left(\frac{\alpha}{S} + 1 - \alpha\right). \tag{50}$$

This formula indicates that the higher the S and α values, the faster the program is processed. The vector/scalar speed ratio S depends of course on the computer but it also depends on the nature of the program. The program attributes affecting the value of S are for instance the vector lengths and the data reference method, i.e., the type of indexing. On a Cray-1 the workload dependent vector speedup S can range from 5. to 10.. Assuming an average value of $S = 7.5$, (50) gives that $\alpha = 0.00$ for $P = 1$, $\alpha = 0.58$ for $P = 2$, $\alpha = 0.77$ for $P = 3$, $\alpha = 0.87$ for $P = 4$, $\alpha = 0.92$ for $P = 5$, $\alpha = 0.96$ for $P = 6$, $\alpha = 0.99$ for $P = 7$, so that Amdhal's law provided an approximated method to measure the vectorization ratio of a code by measuring P.

On the other hand, it is also usefull to compare the execution times T_s and T_v of a program when the vector mode is inactive and active, respectively with the corresponding execution times T_s^{old} and T_v^{old} of different versions of the code :

$$\mathcal{P}_s = T_s^{\text{old}}/T_s, \qquad \mathcal{P}_v = T_v^{\text{old}}/T_v, \tag{51}$$

to measure the improvements of the new versions.

4.2 NUMERICAL RESULTS

4.2.1 SID subroutines The performances of the SID routines returning the species enthalpies h_k, the chemical production rates ω_k, the semi-empirical mixture conductivity λ, the Wilke mixture conductivity λ and the mixture diffusion coefficients D_k exploiting or not the symmetry of the $\mathcal{D}_{kl}$, are presented in Table 1 for a Hydrogen-Air mixture with $K = 9$ and $I = 19$, in Table 2 for a Propane-Air mixture with $K = 33$ and $I = 126$, and in the columns referenced by h_k, ω_k, λ_e, λ_w, D_k^{sym} and D_k, respectively. These subroutines represent quite well all the different types of algorithms considered in Section 3. In Tables 1 and 2, the first lines are the vector/scalar performances P of the SID routines, while the other two are the performance ratios $\mathcal{P}_s$ and $\mathcal{P}_v$ of the SID routines over the corresponding modules of the CHEMKIN and TRANSPORT libraries. The TRANSPORT subroutines [15] were modified in such a way that the mole fractions X_k are in the calling list instead of the mass fractions Y_k, which lead to faster routines. Mixture diffusion coefficients are estimated with the modified formula (48).

	h_k	ω_k	λ_e	λ_w	D_k^{sym}	D_k
P	2.10	1.95	1.62	4.06	2.04	3.89
$\mathcal{P}_s$	2.10	1.26	1.39	1.05	2.43	1.72
$\mathcal{P}_v$	6.05	2.51	1.67	1.43	3.64	4.90

TABLE 1
Performances of the SID routines for a Hydrogen-Air mixture.

	h_k	ω_k	λ_e	λ_w	D_k^{sym}	D_k
P	4.04	2.55	3.40	8.16	4.21	6.19
$\mathcal{P}_s$	2.32	1.39	1.53	1.03	2.87	1.81
$\mathcal{P}_v$	13.59	3.57	2.96	1.31	8.36	7.72

TABLE 2
Performances of the SID routines for a Propane-Air mixture.

The first important point is that the performances P of the SID routines depend on the reaction mechanism. Indeed, the larger are K and I, the longer are the vector lengths involved in the SID routines and therefore the larger is the speed ratio S, with a resulting larger P according to (50). The presence of scalar reduction loops in the algorithms for λ_e and the irreducible scalar kernel (40) in the evaluation of ω_k explain the corresponding performances. Note also the differences between D_k^{sym} and D_k due to the small vector lengths in (46).

In scalar mode, the performances over the CHEMKIN and TRANSPORT routines are sligthly larger than one. Indeed, a vectorized algorithm often run a little bit faster than a corresponding scalar version because fewer loop setups, vector loads, etc., are usually needed in the vectorized version. The higher values $\mathcal{P}_s = 2.43$–2.87 for D_k^{sym} are due to the smaller number of arithmetic operations.

Finally, in vector mode, the improvements over the CHEMKIN and TRANSPORT routines range from 1.31 to 13.5. The value $\mathcal{P}_v = 1.31$, for the evaluation of λ_w, one of the most expensive subroutines, shows that the corresponding TRANSPORT module is quite well vectorized. The unusual value $\mathcal{P}_v = 13.5$, for the evaluation of h_k $k = 1, \ldots, K$, is due in part to the bad vector/scalar performance $P = 0.69$ of the corresponding CHEMKIN module, due to scalar reduction loops of small length used in the evaluation of the thermodynamic fits. More important, however, is the improvement for evaluating D_k, $k = 1, \ldots, K$, which usually is a costly calculation. Note also that exploiting the symmetry of the $\mathcal{D}_{kl}$ is slower than (45) for Hydrogen-Air mixtures.

4.2.2 MID subroutines The performances of the MID routines returning the species enthalpies h_{kj}, the chemical production rates ω_{kj}, the semi-empirical mixture conductivity λ_j, the Wilke mixture conductivity λ_j and the mixture diffusion coefficients D_{kj}, are presented in Table 3 for a Hydrogen-Air flame with $K = 9$, $I = 19$ and $J = 77$, and in Table 4 for a Propane-Air flame with $K = 33$, $I = 126$ and $J = 63$, and in the columns referenced by h_k, ω_k, λ_e, λ_w and D_k^{sym}, respectively. Again, these subroutines represent quite well all the different types of algorithms considered in Section 3. In Tables 3 and 4, the first lines are the vector/scalar performances P of the MID routines, while the other two are the performance ratios $\mathcal{P}_s$ and $\mathcal{P}_v$ of the MID routines over the corresponding SID routines used repetitively. In the column D_k^{sym} the performance ratios are measured in comparison with the SID routine using the algorithm (45) ,which does not exploit the symmetry of the binary diffusion coefficients, and mixture diffusion coefficients are estimated with the modified formula (48).

At variance with the SID routines, the performances P of the MID routines are independent of the reaction mechanism, and essentially depend on the number of input data J. Fairly uniform values are obtained for P, ranging from 6.91 up to 7.88, except with h_k for which P=3.72–3.79, due to the complex formulation (16) (17) and to the fact that there is only one path to/from memory on a Cray-1. These very good results of the MID subroutines are due to their vectorization by 'replication' for which all statements fully vectorize with vectors of length J and which minimizes the number of arithmetic operations. The only disavantage could be the amount of memory needed, but the largest temporary array, the one used to store the $\mathcal{D}_{klj}$ of length $K*K*J$, can be omitted if there are memory problems. A consequence is that in vector mode the MID routines are significantly faster than the corresponding SID modules used repetitively, especially for small reaction schemes like Hydrogen-Air.

	h_k	ω_k	λ_e	λ_w	D_k^{sym}
P	3.72	7.11	7.73	7.11	7.03
$\mathcal{P}_s$	2.45	1.75	1.13	3.36	1.63
$\mathcal{P}_v$	3.95	6.59	5.14	5.82	2.88

TABLE 3
Performances of the MID routines for a Hydrogen-Air flame.

	h_k	ω_k	λ_e	λ_w	D_k^{sym}
P	3.79	7.36	7.88	6.91	7.33
$\mathcal{P}_s$	2.11	1.53	0.97	3.58	1.59
$\mathcal{P}_v$	1.82	4.69	2.08	3.01	1.91

TABLE 4
Performances of the MID routines for a Propane-Air flame.

Now in scalar mode, the MID routines are slightly faster except for scalar reduction loops in the Propane case. The relatively large values $\mathcal{P}_s$=3.36–3.58 obtained for λ_w are due to the smaller number of operations in the MID case, since for instance the quantities like $(1/\sqrt{8})(1+(W_k/W_l))^{-1/2}$ and $(W_l/W_k)^{1/4}$ involved in ϕ_{kl} (31) have to be evaluated only once for $j=1,\ldots,J$.

4.2.3 A laminar flame solver To test the MID routines in a practical situation we have computed several laminar premixed stagnation point flow flames, including extinction limits. For the purpose of the present paper it is enough to know that in this flow configuration Eqs. (1)-(5) reduce to a two-point boundary value problem. More details on the modeling can be found in [6]. The laminar flame solver is based on Newton's method, adaptive gridding and continuation techniques. We refer to [5] [6] [7] [21] for more details on the solution method.

For a wide variety of Hydrogen-Air, Methane-Air and Propane-Air flame structure calculations, we have found that typically 70–80% of the CPU time is spent in the MID routines in scalar mode, so that this part of the CPU time is approximately divided by a factor of 7. in vector mode. The remaining 20–30% of the CPU time in scalar mode is spent for linear algebra, adaptive gridding and the form of the governing equations once thermodynamic properties, chemical production rates and transport coefficients are known. These parts of the solver are only partially vectorized by the compiler. Nevertheless, average vector/scalar performance P around 4. have been obtained for the laminar flame solver, allowing considerable savings in CPU time in our different applications [4] [6] [7].

5. CONCLUSION

The evaluation of thermodynamic properties, chemical production rates, and multicomponent transport coefficients, has been investigated. Vectorizing algorithms have been obtained for single and multiple input data subroutines. These subroutines have allowed important savings in CPU time.

ACKNOWLEDGMENT

We would like to thank Pr. S. Candel, Dr. R. J. Kee, Pr. M. D. Smooke and Pr. J. Warnatz for interesting discussions concerning this material. We would also like to thank the Scientific Council of the 'Centre de Calcul Vectoriel pour la Recherche' for providing the Cray-1 resources and the 'Centre Inter Régional de Calcul Electronique' for providing computational resources. The research was supported in part by the 'Direction des Recherche et Etudes Techniques'.

REFERENCES

1 Boris J. P. and Winsor N. K., *Vectorized Computation of Reactive Flow*, in Parallel Computations, G. Rodrigue Ed., Academic press, (1982), p. 173–215.

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

3 Coffee T. P. and Heimerl J. M., *Transport Algorithms for Premixed, Laminar Steady-state Flames*, Comb. and Flame, **43**, (1981), p. 273–289.

4 Darabiha N., Giovangigli V., Candel S. and Smooke M., *Extinction of Strained Premixed Propane-Air Flames with Complex Chemistry*, submitted to Comb. Sci. and Tech..

5 Giovangigli V. and Smooke M., *Extinction Limits for Premixed Laminar Flames in a Stagnation Point Flow*, J. Comp. Phys. **68**, (1987), p. 327–345.

6 Giovangigli V. and Smooke M., *Extinction Limits of Strained Premixed Laminar Flames with Complex Chemistry*, Comb. Science and Tech., in press , (1987).

7 Giovangigli V. and Smooke M., *Adaptive Continuation Algorithms with Application to Combustion Problems*, submitted to Applied Numerical Methods.

8 Glowinsky R., Larrouturou B. and Temam R., Eds, *Numerical Simulation of Combustion Phenomena*, Springer Verlag, Heidelberg, (1985).

9 Gordon S. and Mcbride B. J., *Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks and Chapman-Jauguet Detonations*, NASA SP-273, (1971).

10 Hirschfelder J. O. and Curtiss C. F., *Flame Propagation in Explosive Gas Mixtures*, Third Symposium (International) on Combustion, Reinhold, New York, (1949), p. 121–127.

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

12 Hockney R. W. and Jesshope C. R., *Parallel Computers*, Adam Hilger Ltd, (1981).

13 Kee R. J., and Dwyer H. A., *Review of Stiffness and Implicit Finite-Difference Methods in Combustion Modeling*, Proceedings of the 7th ICODER, (1981), p. 485–500.

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

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

16 Neves K. N., *Vectorization of Scientific Software*, in High Speed Computation, J. S. Kowalik Ed., NATO ASI Series Vol. F7, Springer-Verlag, (1984), p. 277–291.

17 Oran E. S. and Boris J. P., *Detailed Modeling of Combustion Systems*, Prog. Energy Combust. Sci., **7**, (1981), p. 1–72.

18 Pacific-Sierra Research Corp., *Fortran Programming on the Cray-1*, Pacific-Sierra Research Corp., Los Angeles, (1983).

19 Peters N. and Warnatz J., Eds., *Numerical Methods in Laminar Flame Propagation*, Vieweg Verlag, (1982).

20 Rogg B., *Numerical Analysis of Confined Laminar Jet Diffusio Flames.*, this volume, (1987).

21 Smooke M. D., *Solution of Burner-Stabilized Premixed Laminar Flames by Boundary Value Methods*, J. Comp. Phys., **48**, (1982), p. 72–105.

22 Smooke M. D., *Solution of Two-Dimensional Axisymmetric Laminar Diffusion Flames by Adaptive Boundary Value Methods*, this volume, (1987).

23 Smooke M. D., Miller J. A. and Kee R. J., *Determination of Adiabatic Flame Speeds by Boundary Value Methods*, Comb. Sci. and Tech., **34**, (1983), p. 79–89.

24 Warnatz J., *The Mechanism of High Temperature Combustion of Propane and Butane*, Comb. Sci. and Tech., **34**, (1983), p. 144–200.

25 Warnatz J., *Calculation of the Structure of Laminar Flat Flames II : Flame Velocity and structure of Freely Propagating Hydrogen-Oxygen and Hydrogen-Air Flames*, Ber. Bunsenges. Phys. Chem., **32**, (1978), p. 643–649.