.

SENGA2 User Guide

Stewart Cant

CUED/A–THERMO/TR67
September 2012

1 Introduction

Direct Numerical Simulation (DNS) of turbulent combustion involves the solution of the augmented Navier–Stokes equations without modelling of the turbulence, and with an adequate treatment of the chemistry and the molecular transport of momentum, heat and mass. Ideally the chemical reaction mechanism would be represented in fully detailed form, as would the various mechanisms of molecular transport, and a separate balance equation would be solved for the mass fraction of every chemical species present in the reacting gas mixture. Such a full treatment remains computationally unaffordable for practical simulations, and some degree of compromise is necessary in order to make progress.

The `SENGA2` code was derived from the earlier `SENGA` code [1] which remains in use for DNS with simplified chemistry. The purpose of the `SENGA2` code is to facilitate combustion DNS with any desired level of chemistry, from single-step Arrhenius mechanisms through all classes of reduced reaction mechanisms up to fully detailed reaction mechanisms wherever this is feasible computationally. Molecular transport can be represented using any type of Fickian diffusion law or more complex treatments. The Navier–Stokes momentum equations are solved in fully compressible form together with the continuity equation and a conservation equation for the stagnation internal energy, as well as any required number of balance equations for species mass fraction. Each component of the reacting mixture is assumed to obey the equations of state for a semi–perfect gas. Boundary conditions are specified using an extended form of the Navier–Stokes Characteristic Boundary Condition (NSCBC) formulation [2, 3], and available boundary condition types include periodic as well as several varieties of walls, inflows and outflows.

The numerical framework is based on a high–order finite–difference approach for spatial discretisation together with a Runge–Kutta algorithm for time–stepping. Spatial differencing schemes are implemented in the code through a set of modular spatial differentiator routines, and examples of those tested include Fourier spectral schemes and high–order Padé finite–difference schemes [4]. High–order explicit schemes are preferred due to their speed of execution and ease of parallel implementation, and a 10th order explicit scheme is standard for interior points. The accuracy of the scheme is reduced to 8th, 6th and 4th order as a non–periodic boundary is approached. For time–stepping, the Runge–Kutta algorithm may be adjusted to have any number of sub–steps in order to achieve the required order of accuracy. Adaptive time–stepping is available with error control using an embedded Runge–Kutta scheme. The standard time–stepping algorithm is a five–step fourth–order explicit Runge–Kutta method with a third–order embedded scheme [5] together with a PID–type time step controller [6]. The code is fully parallel using domain decomposition over a Cartesian topology.

The chemical and thermodynamic data required for each simulation is handled in a compact tabular form in order to maximise computational efficiency during a simulation. A pre–processor code called `PPCHEM` is available to convert data from a simple text–based format into the required form for input to `SENGA2`.

2 Mathematical Formulation

2.1 Governing Equations

The governing equations to be solved using DNS are the partial differential equations for compressible reacting flow, consisting of the continuity equation

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_k}\rho u_k = 0, \tag{1}$$

the Navier–Stokes momentum equations

$$\frac{\partial}{\partial t}\rho u_i + \frac{\partial}{\partial x_k}\rho u_k u_i = -\frac{\partial}{\partial x_i}p + \frac{\partial}{\partial x_k}\tau_{ki}, \tag{2}$$

and the internal energy equation

$$\frac{\partial}{\partial t}\rho E + \frac{\partial}{\partial x_k}\rho u_k E = -\frac{\partial}{\partial x_k}p u_k - \frac{\partial}{\partial x_k}q_k + \frac{\partial}{\partial x_k}\tau_{km}u_m, \tag{3}$$

supplemented by an equation for the mass fraction of each of the N chemical species present in the reacting gas mixture

$$\frac{\partial}{\partial t}\rho Y_\alpha + \frac{\partial}{\partial x_k}\rho u_k Y_\alpha = w_\alpha - \frac{\partial}{\partial x_k}\rho V_{\alpha,k} Y_\alpha. \tag{4}$$

The thermal equation of state for the mixture is

$$p = \rho R^0 T \sum_{\alpha=1}^{N} \frac{Y_\alpha}{W_\alpha} \tag{5}$$

while the caloric equation of state provides the definition of the stagnation internal energy as

$$E = \sum_{\alpha=1}^{N} Y_\alpha h_\alpha - \frac{P}{\rho} + \frac{1}{2} u_k u_k \tag{6}$$

where the enthalpy of species α is defined as

$$h_\alpha = \int_{T_0}^{T} C_{p\alpha} dT + h_\alpha^0, \tag{7}$$

in which $C_{p\alpha}$ is the mass–based specific heat capacity of species α and h_α^0 is the species enthalpy at the reference temperature T_0. The viscous stress tensor is given by

$$\tau_{ki} = \mu \left(\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k} \right) - \frac{2}{3}\mu \frac{\partial u_m}{\partial x_m} \delta_{ki}, \tag{8}$$

and the heat flux vector by

$$q_k = -\lambda \frac{\partial T}{\partial x_k} + \sum_{\alpha=1}^{N} \rho V_{\alpha,k} Y_\alpha h_\alpha \tag{9}$$

The species mass fractions are subject to the compatibility condition

$$\sum_{\alpha=1}^{N} Y_\alpha = 1 \tag{10}$$

while the diffusion velocities must satisfy the compatibility condition

$$\sum_{\alpha=1}^{N} V_{\alpha,k} Y_\alpha = 0. \tag{11}$$

For a reaction mechanism involving M steps of the form

$$\sum_{\alpha=1}^{N} \nu'_{\alpha,m} \mathcal{M}_\alpha \to \sum_{\alpha=1}^{N} \nu''_{\alpha,m} \mathcal{M}_\alpha, \quad m = 1, \dots, M \tag{12}$$

the chemical production rate w_α for species α is expressed as

$$w_\alpha = W_\alpha \sum_{m=1}^{M} \left(\nu''_{\alpha,m} - \nu'_{\alpha,m} \right) k_m(T) \prod_{\beta=1}^{N} c_\beta^{\nu'_{\beta,m}}. \tag{13}$$

where the specific reaction rate coefficient $k_m(T)$ is given by the Arrhenius expression

$$k_m(T) = A_m T^{n_m} \exp\left(-\frac{E_m}{R^0 T} \right) \tag{14}$$

and the concentration c_β of species β is related to mass fraction by

$$c_\beta = \frac{\rho Y_\beta}{W_\beta}. \tag{15}$$

For the reaction rates the compatibility condition is

$$\sum_{\alpha=1}^{N} w_\alpha = 0. \tag{16}$$

2.2 Thermodynamic Quantities

For a semi–perfect gas, the molar specific heat capacity at constant pressure $\bar{C}_{p\alpha}$ is known to depend on temperature, and for present purposes the dependence is represented in an approximate manner using a polynomial of the form

$$\frac{\bar{C}_{p\alpha}}{R^0} = \sum_{j=1}^{J} \bar{a}_{\alpha,j}^{(l)} T^{j-1} \tag{17}$$

where the degree $J-1$ of the polynomial is typically taken to be 4 or 5. The polynomial coefficients $a_{\alpha,j}^{(l)}$ may take different values for each species α in different intervals l of temperature. For example, the popular CHEMKIN database [7] uses polynomials of degree 5 with two temperature intervals: for most species the intervals are $0 < T < 1000\text{K}$ ($l = 1$) and $1000\text{K} \le T < 3000\text{K}$ ($l = 2$). The molar enthalpy for each species is given by integrating over L successive temperature intervals $l = 1, \ldots, L$ to yield the recursive formula

$$\begin{aligned} \bar{h}_\alpha &= R^0 \int_{T_{L-1}}^{T} \sum_{j=1}^{J} \bar{a}_{\alpha,j}^{(L)} T^{j-1} dT + R^0 \sum_{l=1}^{L-1} \left[\int_{T_{l-1}}^{T_l} \sum_{j=1}^{J} \bar{a}_{\alpha,j}^{(l)} T^{j-1} dT \right] + \bar{h}_\alpha^0 \\ &= R^0 \sum_{j=1}^{J} \frac{\bar{a}_{\alpha,j}^{(L)}}{j} T^j - R^0 \sum_{j=1}^{J} \frac{\bar{a}_{\alpha,j}^{(L)}}{j} T_{L-1}^j + \bar{h}_\alpha^{(L-1)}. \end{aligned} \tag{18}$$

where the reference temperature T_0 has been taken at 0K, and $h_\alpha^{(L-1)}$ denotes the enthalpy at the lower end of the current (Lth) temperature interval. The last two terms may be combined to form a single coefficient $\bar{a}_{\alpha,J+1}^{(L)}$, and the final form for the molar enthalpy becomes

$$\bar{h}_\alpha = R^0 \left[\sum_{j=1}^{J} \frac{\bar{a}_{\alpha,j}^{(L)}}{j} T^j + \bar{a}_{\alpha,J+1}^{(L)} \right] \tag{19}$$

Similarly, the molar entropy for each species may be found using the definition

$$\bar{s}_\alpha = R^0 \int_{T_0}^{T} \frac{\bar{C}_{p\alpha}}{T} dT + \bar{s}_\alpha^0, \tag{20}$$

substituting for $\bar{C}_{p\alpha}$ and integrating over successive temperature intervals to give

$$\begin{aligned} \bar{s}_\alpha &= R^0 \left[\bar{a}_{\alpha,1}^{(L)} \ln T + \sum_{j=2}^{J} \frac{\bar{a}_{\alpha,j}^{(L)}}{j-1} T^{j-1} \right] \\ &- R^0 \left[\bar{a}_{\alpha,1}^{(L)} \ln T_{L-1} + \sum_{j=2}^{J} \frac{\bar{a}_{\alpha,j}^{(L)}}{j-1} T_{L-1}^{j-1} \right] + \bar{s}_\alpha^{(L-1)} \end{aligned} \tag{21}$$

where once again the formula may be applied recursively, and $s_\alpha^{(L-1)}$ is the entropy at the lower end of the current temperature interval. Combining the last three terms to form a single coefficient $\bar{a}^{(L)}_{\alpha,J+2}$, the final form for the molar entropy is

$$\bar{s}_\alpha = R^0 \left[\bar{a}^{(L)}_{\alpha,1} \ln T + \sum_{j=2}^{J} \frac{\bar{a}^{(L)}_{\alpha,j}}{j-1} T^{j-1} + \bar{a}^{(L)}_{\alpha,J+2} \right] \tag{22}$$

The mass-based specific heat capacity $C_{p\alpha}$ is given by

$$C_{p\alpha} = \frac{R^0}{W_\alpha} \sum_{j=1}^{J} \bar{a}^{(l)}_{\alpha,j} T^{j-1} = \sum_{j=1}^{J} a^{(l)}_{\alpha,j} T^{j-1} \tag{23}$$

where the mass-based polynomial coefficients are defined as

$$a^{(l)}_{\alpha,j} = \frac{R^0}{W_\alpha} \bar{a}^{(l)}_{\alpha,j}. \tag{24}$$

Then the mass-based specific enthalpy for each species is

$$h_\alpha = \sum_{j=1}^{J} \frac{a^{(L)}_{\alpha,j}}{j} T^j + a^{(L)}_{\alpha,J+1} \tag{25}$$

and the mass-based specific entropy for each species is

$$s_\alpha = a^{(L)}_{\alpha,1} \ln T + \sum_{j=2}^{J} \frac{a^{(L)}_{\alpha,j}}{j-1} T^{j-1} + a^{(L)}_{\alpha,J+2}. \tag{26}$$

2.2.1 Temperature

The temperature may be obtained using the caloric equation of state (6) in the form

$$E = \sum_{\alpha=1}^{N} Y_\alpha h_\alpha - R_m T + \frac{1}{2} u_k u_k \tag{27}$$

where the specific gas constant for the mixture is

$$R_m = \sum_{\alpha=1}^{N} Y_\alpha R_\alpha. \tag{28}$$

Substituting the polynomial form of the species enthalpy h_α from (25) gives

$$E = \sum_{\alpha=1}^{N} Y_\alpha \left[\sum_{j=1}^{J} \frac{a^{(l)}_{\alpha,j}}{j} T^j + a^{(l)}_{\alpha,J+1} \right] - R_m T + \frac{1}{2} u_k u_k \tag{29}$$

and gathering terms in successive powers of T produces the polynomial

$$\begin{aligned} f(T) \quad &= \quad \left[\left(\frac{1}{2} u_k u_k - E \right) + \sum_{\alpha=1}^{N} Y_\alpha a^{(l)}_{\alpha,J+1} \right] \\ &+ \quad \left[\sum_{\alpha=1}^{N} Y_\alpha \left(a^{(l)}_{\alpha,1} - R_\alpha \right) \right] T + \sum_{j=2}^{J} \left[\sum_{\alpha=1}^{N} Y_\alpha \frac{a^{(l)}_{\alpha,j}}{j} \right] T^j \end{aligned} \tag{30}$$

which may be used to form the non-linear algebraic equation $f(T) = 0$ whose solution determines the temperature.

2.3 Transport Coefficients

The molecular transport coefficients are represented using the relationship

$$\frac{\lambda}{C_p} = A_\lambda \left(\frac{T}{T_0}\right)^r \tag{31}$$

for the mixture thermal conductivity λ, where C_p is the mixture value of the specific heat capacity at constant pressure, and A_λ, r and T_0 are constants. Then the mixture dynamic viscosity μ is given by

$$\mu = \frac{\lambda}{C_p}\mathrm{Pr} \tag{32}$$

where Pr is the mixture Prandtl number which is taken as a constant.

The diffusive mass flux for species α may be represented using Fick's law as:

$$\rho V_{\alpha,k} Y_\alpha = -\rho D_\alpha \frac{\partial Y_\alpha}{\partial x_k}. \tag{33}$$

in which the diffusion coefficient D_α for each species is given by

$$D_\alpha = \frac{\lambda}{\rho C_p Le_\alpha} \tag{34}$$

where Le_α is the Lewis number. The standard approach assumes that Le_α is constant for each species.

Applying this assumption in Fick's Law does not guarantee that the continuity equation will be recovered when all N of the species mass fraction equations are summed. Substituting (33) into (4) and summing over all species yields

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_k}\rho u_k = \sum_{\alpha=1}^{N} \frac{\partial}{\partial x_k}\rho D_\alpha \frac{\partial Y_\alpha}{\partial x_k}. \tag{35}$$

where the compatibility conditions (10) and (16) have been applied to the mass fractions and reaction rates respectively. Clearly, by comparison with the continuity equation (1) the quantity on the right-hand side is an error term. This term can be removed by making a slight modification to Fick's law [8]:

$$\rho V_{\alpha,k} Y_\alpha = -\rho D_\alpha \frac{\partial Y_\alpha}{\partial x_k} + \rho V_k^{(c)} Y_\alpha \tag{36}$$

where the correction velocity $V_k^{(c)}$ is given by

$$\rho V_k^{(c)} = \sum_{\alpha=1}^{N} \rho D_\alpha \frac{\partial Y_\alpha}{\partial x_k}. \tag{37}$$

Applying the compatibility condition (11), the modified form (36) ensures that the continuity equation (1) is recovered as required.

2.4 Reaction Rate

Evaluation of the chemical production rate w_α for species mass fraction makes use of the general formulation expressed by (13)-(15). There is a summation over all steps in the reaction mechanism

$$w_\alpha = W_\alpha \sum_{m=1}^{M} \bar{w}_{\alpha,m} \tag{38}$$

where $\bar{w}_{\alpha,m}$ is the molar production rate of species α in the step. Each step in the reaction mechanism may involve one of several possible special cases.

2.4.1 Forward Reaction Rate

For a forward reaction step as expressed by

$$\sum_{\alpha=1}^{N} \nu'_{\alpha,m} \mathcal{M}_\alpha \rightarrow \sum_{\alpha=1}^{N} \nu''_{\alpha,m} \mathcal{M}_\alpha \tag{39}$$

the molar production rate $\bar{w}_{\alpha,m}$ is given by

$$\bar{w}_{\alpha,m} = \left(\nu''_{\alpha,m} - \nu'_{\alpha,m}\right) k_m(T) \prod_{\beta=1}^{N} c_\beta^{\nu'_{\beta,m}}. \tag{40}$$

with $k_m(T)$ given by (14) and c_β by (15). This is the simplest and most common type of reaction step.

2.4.2 Backward Reaction Rate: Gibbs Function

In cases where a reaction step m is specified using the equilibrium notation

$$\sum_{\alpha=1}^{N} \nu'_{\alpha,m} \mathcal{M}_\alpha \rightleftharpoons \sum_{\alpha=1}^{N} \nu''_{\alpha,m} \mathcal{M}_\alpha \tag{41}$$

the molar production rate for a single species is given by

$$\bar{w}_{\alpha,m} = \left(\nu''_{\alpha,m} - \nu'_{\alpha,m}\right) \left[k_{f,m}(T) \prod_{\beta=1}^{N} c_\beta^{\nu'_{\beta,m}} - k_{b,m}(T) \prod_{\beta=1}^{N} c_\beta^{\nu''_{\beta,m}} \right]. \tag{42}$$

Often, only the data for the forward rate coefficient $k_{f,m}(T)$ are supplied, and it becomes necessary to evaluate the backward rate coefficient $k_{b,m}(T)$ using the equilibrium constant for concentrations

$$K_{c,m} = \prod_{\alpha=1}^{N} c_\alpha^{(\nu''_{\alpha,m} - \nu'_{\alpha,m})} = \frac{k_{b,m}}{k_{f,m}} \tag{43}$$

The relation between $K_{c,m}$ and the equilibrium constant for partial pressures $K^0_{p,m}$ is given by

$$K_{c,m} = K^0_{p,m} \left(\frac{p_0}{R^0 T}\right)^{\Delta\nu_m} \tag{44}$$

where p_0 is a reference pressure used to make $K^0_{p,m}$ dimensionless, and $\Delta\nu_m = \sum_{\alpha=1}^{N}(\nu''_{\alpha,m} - \nu'_{\alpha,m})$ is the difference in the total number of moles between reactants and products. In turn, $K^0_{p,m}$ is related to the change in the molar Gibbs function according to

$$R^0 T \ln K^0_{p,m} = \Delta\hat{G}_m = \sum_{\alpha=1}^{N} \bar{g}_\alpha (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \tag{45}$$

where $\bar{g}_\alpha = \bar{h}_\alpha - T\bar{s}_\alpha$ is the molar Gibbs function for species α. Using (19) and (22), $\bar{g}_\alpha$ can be expressed conveniently as

$$\frac{\bar{g}_\alpha}{R^0 T} = \frac{\bar{a}^{(L)}_{\alpha,J+1}}{T} - \bar{a}^{(L)}_{\alpha,1} \ln T + (\bar{a}^{(L)}_{\alpha,1} - \bar{a}^{(L)}_{\alpha,J+2}) - \sum_{j=2}^{J} \frac{\bar{a}^{(L)}_{\alpha,j}}{j(j-1)} T^{j-1} \tag{46}$$

Writing $k_{f,m}$ and $k_{b,m}$ in Arrhenius form according to (14), and using (43) together with (44) and (46) yields a set of expressions for the parameters of the backwards reaction rate coefficient

$$\begin{aligned}
\ln A_{b,m} &= \ln A_{f,m} \\
&+ \sum_{\alpha=1}^{N} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left[\bar{a}^{(L)}_{\alpha,1} - \bar{a}^{(L)}_{\alpha,J+2} + \ln\left(\frac{P_0}{R^0}\right)\right] \\
&- \ln A^{\Sigma}_{b,m}(T) \qquad (47) \\
n_{b,m} &= n_{f,m} - \sum_{\alpha=1}^{N} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left[\bar{a}^{L}_{\alpha,1} + 1\right] \qquad (48) \\
\frac{E_{b,m}}{R^0} &= \frac{E_{f,m}}{R^0} + \sum_{\alpha=1}^{N} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left[\bar{a}^{L}_{\alpha,J+1}\right] \qquad (49)
\end{aligned}$$

Note that there is a non-Arrhenius contribution to (47) denoted by $\ln A^{\Sigma}_{b,m}(T)$. This term arises from the temperature dependence of $C_{p\alpha}$, and is given by

$$\ln A^{\Sigma}_{b,m}(T) = \sum_{\alpha=1}^{N} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \sum_{j=2}^{J} \frac{\bar{a}^{(L)}_{\alpha,j} T^j}{j(j+1)}. \tag{50}$$

Due to this term the backward rate coefficient $k_{b,m}$ has a temperature dependence over and above that indicated by the Arrhenius form. This means that the contribution $\ln A^{\Sigma}_{b,m}(T)$ cannot be computed in advance and hence $k_{b,m}$ must be evaluated "on the fly" during a simulation. This may be done by combining (43)-(46) to yield

$$\ln k_{b,m} = \ln k_{f,m} + \sum_{\alpha=1}^{N} (\nu''_{\alpha,m} - \nu'_{\alpha,m}) \left(\frac{\bar{g}_\alpha}{R^0 T} + \ln\frac{p_0}{R^0} - \ln T\right) \tag{51}$$

2.4.3 Third Bodies

In cases where a reaction step m is of the form

$$\sum_{\alpha=1}^{N} \nu'_{\alpha,m} \mathcal{M}_\alpha + M \to \sum_{\alpha=1}^{N} \nu''_{\alpha,m} \mathcal{M}_\alpha + M \tag{52}$$

the molar production rate for a species α is given by

$$\bar{w}_{\alpha,m} = \left(\nu''_{\alpha,m} - \nu'_{\alpha,m}\right) k_m(T) c_M \prod_{\beta=1}^{N} c_\beta^{\nu'_{\beta,m}}. \tag{53}$$

where the concentration c_M of the "third body" M is given by

$$c_M = \sum_{\alpha=1}^{N} \eta_{\alpha,M} c_\alpha \tag{54}$$

in which the coefficients $\eta_{\alpha,M}$ are the third-body efficiencies for M.

2.4.4 Lindemann Forms

Some three–body reaction steps may have a specific reaction rate coefficient k which depends on pressure as well as temperature. The simplest representation is the Lindemann form [9]:

$$k_{L,m} = k_{\infty,m} \left(\frac{P_r}{1 + P_r} \right) F_m \tag{55}$$

where $k_{\infty,m}$ is the rate coefficient in the limit of high pressure and F_m is a constant normally taken equal to unity although other values may be specified instead. The quantity P_r is a "reduced pressure" defined as

$$P_r = \frac{k_{0,m}}{k_{\infty,m}} c_M \tag{56}$$

where $k_{0,m}$ is the rate coefficient in the limit of low pressure and c_M is the third–body concentration as defined in (54). Both $k_{\infty,m}$ and $k_{0,m}$ have the standard Arrhenius form (14). A Lindemann reaction step requires the specification of seven values, i.e. the three parameters A, n and E for each of $k_{\infty,m}$ and $k_{0,m}$, together with the value of F_m.

2.4.5 Troe Forms

For some pressure–dependent three–body reaction steps the Lindemann form is found to be insufficient and the Troe form is preferred [10]. The Lindemann representation (55) is retained with the reduced pressure P_r defined as in (56), but now F_m is specified as a function

$$\log F_m = \left[1 + \left(\frac{\log P_r + c}{n - d(\log P_r + c)} \right)^2 \right]^{-1} \log F_{\text{cent}} \tag{57}$$

The quantities c and n are defined as:

$$c = c_1 - c_2 \log F_{\text{cent}}; \qquad n = n_1 - n_2 \log F_{\text{cent}} \tag{58}$$

where the standard values are $c_1 = -0.4$, $c_2 = 0.67$, $n_1 = -0.75$, $n_2 = 1.27$ and $d = 0.14$. The function F_{cent} is given by

$$F_{\text{cent}} = (1 - \alpha) \exp \left(-\frac{T}{T^{***}} \right) + \alpha \exp \left(\frac{T}{T^{*}} \right) + \exp \left(-\frac{T^{**}}{T} \right). \tag{59}$$

A total of fifteen values must be specified for a Troe step, consisting of the three parameters A, n and E for each of $k_{\infty,m}$ and $k_{0,m}$, the quantities α, T^*, T^{**} and T^{***}, and the constants c_1, c_2, n_1, n_2 and d.

2.4.6 SRI Forms

A third form for pressure–dependent three–body reactions is due to SRI International [11]. Again the Lindemann representation defined by (55) and (56) is retained, but now the function F_m is defined as

$$F_m = d \left[a \exp \left(-\frac{b}{T} \right) + \exp \left(-\frac{T}{c} \right) \right]^X T^e \tag{60}$$

where d and e are normally set to unity, and X is given by

$$X = \frac{1}{1 + \log^2 P_r}. \tag{61}$$

Each SRI step requires eleven values, which are the three parameters A, n and E for each of $k_{\infty,m}$ and $k_{0,m}$, together with the five constants a, b, c, d and e.

2.5 Boundary Conditions

The boundary conditions are specified according to the NSCBC (Navier–Stokes Characteristic Boundary Condition) formalism [2, 3]. At the boundary, the governing equations may be restated in conservative form as [3]

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla_{(n)} \cdot \mathbf{C}^{(n)} + \nabla_{(t)} \cdot \mathbf{C}^{(t)} = \mathbf{D} + \mathbf{s} \tag{62}$$

where the vector of conservative variables is given by $\mathbf{U} = \{\rho, \rho u, \rho v, \rho w, \rho E, \rho Y_\alpha\}^T$, and where $\mathbf{C}$ is the vector of convective fluxes, $\mathbf{D}$ is the vector of diffusive fluxes, $\mathbf{s}$ is the vector of source terms and the subscripts (n) and (t) denote the normal and tangential directions to the boundary. The NSCBC formalism operates within the Local One-Dimensional Inviscid (LODI) approximation, in which the tangential convective terms and all molecular transport terms are neglected for the purposes of the boundary-condition analysis. The conservative form of the LODI equation set is:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla_{(n)} \cdot \mathbf{C}^{(n)} = \mathbf{s} \tag{63}$$

in which $\mathbf{C}^{(n)} = \{\rho u_n, \rho u_n u_1 + p\delta_{n1}, \rho u_n u_2 + p\delta_{n2}, \rho u_n u_3 + p\delta_{n3}, \rho u_n E + p u_n, \rho u_n Y_\alpha\}^T$ is the boundary-normal convective flux vector and $\mathbf{s} = \{0, 0, 0, 0, 0, w_\alpha\}^T$ is the vector of source terms.

The characteristic analysis proceeds using a vector of primitive variables which may be defined in terms of pressure as $\mathsf{U} = \{\rho, u, v, w, p, Y_\alpha\}^T$. Derivatives of the convective and primitive variables are related using the Jacobian matrix $\mathbf{P}$ given by

$$\mathbf{P} = \frac{\partial \mathbf{U}}{\partial \mathsf{U}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \rho & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \rho & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \rho & 0 & 0 & 0 & \dots & 0 \\ E - C_v T & \rho u & \rho v & \rho w & \frac{1}{\gamma-1} & \rho\hat{h}_1 & \rho\hat{h}_2 & \dots & \rho\hat{h}_N \\ Y_1 & 0 & 0 & 0 & 0 & \rho & 0 & \dots & 0 \\ Y_2 & 0 & 0 & 0 & 0 & 0 & \rho & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_N & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \rho \end{bmatrix} \tag{64}$$

where C_v is the specific heat capacity at constant volume for the mixture

$$C_v = \sum_{\alpha=1}^{N} Y_\alpha C_{v\alpha}, \tag{65}$$

γ is the ratio of specific heats for the mixture

$$\gamma = \frac{C_p}{C_v} = \frac{\sum_{\alpha=1}^{N} Y_\alpha C_{p\alpha}}{\sum_{\alpha=1}^{N} Y_\alpha C_{v\alpha}} \tag{66}$$

and $\hat{h}_\alpha$ is the augmented species enthalpy given by

$$\hat{h}_\alpha = h_\alpha - C_p T \frac{R_\alpha}{R_m}. \tag{67}$$

The boundary-normal convective fluxes $\mathbf{C}^{(n)}$ may be related to the spatial gradients of the primitive variables according to $\mathbf{C}^{(n)} = \mathbf{Q}^{(n)}.\left(\nabla_{(n)}\mathsf{S}\right)$ where the matrix $\mathbf{Q}^{(n)} = \partial \mathbf{C}^{(n)}/\partial \mathsf{U} =$

$$\begin{bmatrix}
u_n & \rho\delta_{1n} & \rho\delta_{2n} & \rho\delta_{3n} & 0 & 0 & 0 & \dots & 0 \\
u_1u_n & \rho(u_n + u_1\delta_{1n}) & \rho u_1\delta_{2n} & \rho u_1\delta_{3n} & \delta_{1n} & 0 & 0 & \dots & 0 \\
u_2u_n & \rho u_2\delta_{1n} & \rho(u_n + u_2\delta_{2n}) & \rho u_2\delta_{3n} & \delta_{2n} & 0 & 0 & \dots & 0 \\
u_3u_n & \rho u_3\delta_{1n} & \rho u_3\delta_{2n} & \rho(u_n + u_3\delta_{3n}) & \delta_{3n} & 0 & 0 & \dots & 0 \\
u_n(E - C_vT) & Q_{5,2} & Q_{5,3} & Q_{5,4} & \frac{u_n\gamma}{\gamma-1} & \rho u_n\hat{h}_1 & \rho u_n\hat{h}_2 & \dots & \rho u_n\hat{h}_N \\
u_nY_1 & \rho Y_1\delta_{1n} & \rho Y_1\delta_{2n} & \rho Y_1\delta_{3n} & 0 & \rho u_n & 0 & \dots & 0 \\
u_nY_2 & \rho Y_2\delta_{1n} & \rho Y_2\delta_{2n} & \rho Y_2\delta_{3n} & 0 & 0 & \rho u_n & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
u_nY_N & \rho Y_N\delta_{1n} & \rho Y_N\delta_{2n} & \rho Y_N\delta_{3n} & 0 & 0 & 0 & \dots & \rho u_n
\end{bmatrix} \tag{68}$$

in which the terms $Q_{5,i+1} = (\rho E + p)\delta_{ni} + \rho u_n u_i$ for $i = 1, \dots, 3$.

Using these definitions the LODI equations (63) may be expressed in terms of primitive variables as

$$\frac{\partial \mathsf{U}}{\partial t} + \mathsf{A}^{(n)} \cdot \left(\nabla_{(n)}\mathsf{U}\right) = \mathsf{s} \tag{69}$$

where the matrix $\mathsf{A}^{(n)}$ is given by

$$\mathsf{A}^{(n)} = \mathbf{P}^{-1}\mathbf{Q}^{(n)} = \begin{bmatrix}
u_n & \rho\delta_{1n} & \rho\delta_{2n} & \rho\delta_{3n} & 0 & 0 & 0 & \dots & 0 \\
0 & u_n & 0 & 0 & \delta_{1n}/\rho & 0 & 0 & \dots & 0 \\
0 & 0 & u_n & 0 & \delta_{2n}/\rho & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & u_n & \delta_{3n}/\rho & 0 & 0 & \dots & 0 \\
0 & \gamma p\delta_{1n} & \gamma p\delta_{2n} & \gamma p\delta_{3n} & u_n & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & u_n & 0 & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & u_n & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & u_n
\end{bmatrix} \tag{70}$$

and the corresponding vector of source terms for the primitive variables is $\mathsf{s} = \{0, 0, 0, 0, \mathsf{s}_p, w_\alpha\}^T$ where $\mathsf{s}_p = -(\gamma - 1)\sum_{\alpha=1}^{N} h_\alpha w_\alpha$.

A final transformation into characteristic form is achieved through the decomposition $\mathsf{A}^{(n)} = \mathsf{S}^{(n)}\Lambda^{(n)}\left(\mathsf{S}^{(n)}\right)^{-1}$ where $\Lambda^{(n)}$ is the diagonal matrix of the eigenvalues of $\mathsf{A}^{(n)}$. The eigenvalues are listed in order as $\lambda_r^{(n)} = (u_n - a), u_n, u_n, u_n, (u_n + a), u_n, \dots, u_n$ with $r = 1, \dots, N + 5$, in which a is the speed of sound for the mixture given by $a^2 = \gamma R_m T$. The columns of the matrix $\mathsf{S}^{(n)}$ are the right eigenvectors of $\mathsf{A}^{(n)}$:

$$\mathsf{S}^{(n)} = \begin{bmatrix}
1/a^2 & 1 & 0 & 0 & 1/a^2 & 0 & 0 & \dots & 0 \\
-\delta_{1n}/\rho a & 0 & 1 - \delta_{1n} & 0 & \delta_{1n}/\rho a & 0 & 0 & \dots & 0 \\
-\delta_{2n}/\rho a & 0 & \delta_{1n} & \delta_{3n} & \delta_{2n}/\rho a & 0 & 0 & \dots & 0 \\
-\delta_{3n}/\rho a & 0 & 0 & 1 - \delta_{3n} & \delta_{3n}/\rho a & 0 & 0 & \dots & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1
\end{bmatrix} \tag{71}$$

The characteristic form is obtained by premultiplying (69) by $\left(\mathsf{S}^{(n)}\right)^{-1}$ to yield

$$\frac{\partial \mathcal{U}^{(n)}}{\partial t} + \mathcal{L}^{(n)} = \mathcal{S}^{(n)} \tag{72}$$

where the vectors $\partial\mathcal{U}^{(n)}/\partial t$, $\mathcal{L}^{(n)}$ and $\mathcal{S}^{(n)}$ are given by

$$\frac{\partial\mathcal{U}^{(n)}}{\partial t}=\left[\begin{array}{c}\frac{1}{2}\left(\frac{\partial p}{\partial t}-\rho a\frac{\partial u_n}{\partial t}\right)\\ \frac{\partial\rho}{\partial t}-\frac{1}{a^2}\frac{\partial p}{\partial t}\\ \frac{\partial}{\partial t}\left(u_1+(u_2-u_1)\delta_{1n}\right)\\ \frac{\partial}{\partial t}\left(u_3+(u_2-u_3)\delta_{3n}\right)\\ \frac{1}{2}\left(\frac{\partial p}{\partial t}+\rho a\frac{\partial u_n}{\partial t}\right)\\ \frac{\partial Y_1}{\partial t}\\ \frac{\partial Y_2}{\partial t}\\ \vdots\\ \frac{\partial Y_N}{\partial t}\end{array}\right]\;;\;\mathcal{L}^{(n)}=\left[\begin{array}{c}\frac{(u_n-a)}{2}\left(\frac{\partial p}{\partial x_n}-\rho a\frac{\partial u_n}{\partial x_n}\right)\\ u_n\left(\frac{\partial\rho}{\partial x_n}-\frac{1}{a^2}\frac{\partial p}{\partial x_n}\right)\\ u_n\frac{\partial}{\partial x_n}\left(u_1+(u_2-u_1)\delta_{1n}\right)\\ u_n\frac{\partial}{\partial x_n}\left(u_3+(u_2-u_3)\delta_{3n}\right)\\ \frac{(u_n+a)}{2}\left(\frac{\partial p}{\partial x_n}+\rho a\frac{\partial u_n}{\partial x_n}\right)\\ u_n\frac{\partial Y_1}{\partial x_n}\\ u_n\frac{\partial Y_2}{\partial x_n}\\ \vdots\\ u_n\frac{\partial Y_N}{\partial x_n}\end{array}\right]\;;\;\mathcal{S}^{(n)}=\left[\begin{array}{c}\frac{1}{2}\mathsf{s}_p\\ -\frac{1}{a^2}\mathsf{s}_p\\ 0\\ 0\\ \frac{1}{2}\mathsf{s}_p\\ w_1\\ w_2\\ \vdots\\ w_N\end{array}\right] \qquad (73)$$

In physical terms the vector $\partial\mathcal{U}^{(n)}/\partial t$ describes the amplitude variations of waves travelling across the boundary at the speeds given by the eigenvalues λ_r^n. The vector $\mathcal{L}^{(n)}=\Lambda^{(n)}\left(\mathsf{S}^{(n)}\right)^{-1}\cdot\left(\nabla_{(n)}\mathsf{U}\right)$ describes the contribution to the amplitude variations due to spatial gradients of the primitive variables, while the vector $\mathcal{S}^{(n)}$ describes the contribution due to chemical reaction.

The vector $\mathcal{L}^{(n)}$ is of central importance to the NSCBC formalism. The elements $\mathcal{L}_1^{(n)}$ and $\mathcal{L}_5^{(n)}$ describe the left–running and right–running acoustic waves respectively, while the remaining elements describe the convective transport of entropy ($\mathcal{L}_2^{(n)}$), transverse velocity components ($\mathcal{L}_3^{(n)}$ and $\mathcal{L}_4^{(n)}$), and species mass fraction ($\mathcal{L}_{5+\alpha}^{(n)}$). Manipulation of these elements allows for many different boundary conditions to be set in a physically consistent manner, as described below. Once the manipulation is complete, it is necessary to reconstruct the boundary-normal convective flux vector $\mathbf{C}^{(n)}$ in the conservative form of the equations. This is commonly done in two stages by pre–multiplying the characteristic form (72) by the matrix $\mathsf{S}^{(n)}$ to yield

$$\frac{\partial\mathsf{U}}{\partial t}+\mathsf{d}^{(n)}=\mathsf{s} \qquad (74)$$

where $\mathsf{d}^{(n)}=\mathsf{S}^{(n)}\mathcal{L}^{(n)}=\mathsf{A}_{(n)}\cdot\left(\nabla_{(n)}\mathsf{U}\right)$, and then by pre–multiplying (74) by $\mathbf{P}$ to return to the conservative form (63). Alternatively the same procedure may be carried out in a single step by pre–multiplying (72) by the matrix product $\mathbf{P}\mathsf{S}^{(n)}=$

$$\left[\begin{array}{ccccccccc}1/a^2 & 1 & 0 & 0 & 1/a^2 & 0 & 0 & \dots & 0\\ (u-a\delta_{1n})/a^2 & u & \rho(1-\delta_{1n}) & 0 & (u+a\delta_{1n})/a^2 & 0 & 0 & \dots & 0\\ (v-a\delta_{2n})/a^2 & v & \rho\delta_{1n} & \rho\delta_{3n} & (v+a\delta_{2n})/a^2 & 0 & 0 & \dots & 0\\ (w-a\delta_{3n})/a^2 & w & 0 & \rho(1-\delta_{3n}) & (w+a\delta_{3n})/a^2 & 0 & 0 & \dots & 0\\ E_{51} & E_{52} & E_{53} & E_{54} & E_{55} & \rho\hat{h}_1 & \rho\hat{h}_2 & \dots & \rho\hat{h}_N\\ Y_1/a^2 & Y_1 & 0 & 0 & Y_1/a^2 & \rho & 0 & \dots & 0\\ Y_2/a^2 & Y_2 & 0 & 0 & Y_2/a^2 & 1 & \rho & \dots & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\ Y_N/a^2 & Y_N & 0 & 0 & Y_N/a^2 & 0 & 0 & \dots & \rho\end{array}\right] \qquad (75)$$

where

$$\begin{aligned}E_{51} &= \frac{E-C_vT}{a^2}-\frac{u}{a}\delta_{1n}-\frac{v}{a}\delta_{2n}-\frac{w}{a}\delta_{3n}+\frac{1}{\gamma-1}\\ E_{52} &= E-C_vT\\ E_{53} &= \rho\left(u+(v-u)\delta_{1n}\right)\end{aligned}$$

$$\begin{aligned} E_{54} &= \rho\,(w + (v - w)\delta_{3n}) \\ E_{55} &= \frac{E - C_v T}{a^2} + \frac{u}{a}\delta_{1n} + \frac{v}{a}\delta_{2n} + \frac{w}{a}\delta_{3n} + \frac{1}{\gamma - 1}. \end{aligned} \tag{76}$$

Finally the new value of $\mathbf{C}^{(n)}$ obtained using the LODI analysis must be used to replace the original term in the full conservative form of the equations (62).

It should be noted that it may be necessary also to impose boundary conditions on the diffusive flux terms $\mathbf{D}$ in order to meet the theoretical requirements for the Navier–Stokes equations [2]. This is done in an *ad-hoc* manner as indicated below.

2.5.1 Outflow Boundary Conditions

Subsonic Non–reflecting Outflow
An acoustically non–reflecting outflow boundary condition may be imposed by allowing outgoing acoustic waves to leave the domain while setting the amplitude variation of any incoming acoustic waves to zero. On a left–hand boundary the right–going wave amplitude variation is $\mathcal{L}_5^{(n)}$, while on a right–hand boundary the left-going wave amplitude variation is $\mathcal{L}_1^{(n)}$. The non–reflecting outflow boundary condition is then given by

$$\begin{aligned} \mathcal{L}_5^{(n)} &= \mathcal{S}_5 \quad \text{(left)}; \\ \mathcal{L}_1^{(n)} &= \mathcal{S}_1 \quad \text{(right)}. \end{aligned} \tag{77}$$

Often it is desirable to impose a partially–reflecting boundary condition in order to allow the pressure at the boundary to track some required value p_∞. This is done using the prescription

$$\begin{aligned} \mathcal{L}_5^{(n)} &= \mathcal{S}_5 + \frac{\sigma}{2L} a(1 - M^2)(p - p_\infty) \quad \text{(left)}; \\ \mathcal{L}_1^{(n)} &= \mathcal{S}_1 + \frac{\sigma}{2L} a(1 - M^2)(p - p_\infty) \quad \text{(right)} \end{aligned} \tag{78}$$

where σ is a constant having the value 0.287, L is the length scale of the domain in the boundary–normal direction and M is the Mach number.

The diffusive flux vector $\mathbf{D}$ is modified by imposing the conditions

$$\frac{\partial \tau_{in}}{\partial x_n} = 0; \quad \frac{\partial q_n}{\partial x_n} = 0; \quad \frac{\partial}{\partial x_n}\rho V_{\alpha,n} Y_\alpha = 0 \tag{79}$$

(where no summation is implied) on the normal and tangential components of the viscous stress tensor, the heat flux vector and the diffusive flux vector.

2.5.2 Inflow Boundary Conditions

Subsonic Non–reflecting Laminar Inflow
An acoustically non–reflecting inflow condition may be set using the wave amplitude variations according to

$$\begin{aligned} \mathcal{L}_5^{(n)} &= \mathcal{S}_5 \quad \text{(left)}; \\ \mathcal{L}_1^{(n)} &= \mathcal{S}_1 \quad \text{(right)} \end{aligned} \tag{80}$$

together with

$$\mathcal{L}_2^{(n)} = \mathcal{S}_2; \quad \mathcal{L}_3^{(n)} = 0; \quad \mathcal{L}_4^{(n)} = 0; \quad \mathcal{L}_{5+\alpha}^{(n)} = \mathcal{S}_{5+\alpha} \tag{81}$$

as well as the viscous condition

$$\frac{\partial \tau_{nn}}{\partial x_n} = 0. \tag{82}$$

(with no summation). Note that this boundary condition does not allow for explicit control over the values of the primitive variables which instead are calculated as part of the solution. In particular this boundary condition does not allow the velocity components u_i to be be imposed at the boundary, and so this boundary condition cannot be used to specify a turbulent velocity field at the inlet.

Subsonic Reflecting Inflow With Specified Temperature
This condition is set by imposing the values of temperature T, all three velocity components u_i and all species mass fractions Y_α at all points on the boundary. If required, any or all of these variables may be specified as functions of time, and hence this condition does allow for a turbulent inflow condition. The viscous condition

$$\frac{\partial \tau_{nn}}{\partial x_n} = 0 \tag{83}$$

(no summation) is also imposed. The density ρ is not specified as part of the boundary condition and must be calculated as part of the solution. The LODI form of the continuity equation is

$$\frac{\partial \rho}{\partial t} + \left[\mathcal{L}_2^{(n)} + \frac{1}{a^2}\left(\mathcal{L}_5^{(n)} + \mathcal{L}_1^{(n)}\right)\right] = 0 \tag{84}$$

and hence the values of $\mathcal{L}_1^{(n)}$, $\mathcal{L}_2^{(n)}$ and $\mathcal{L}_5^{(n)}$ are required. These are obtained from the primitive variables as

$$\begin{aligned}
\mathcal{L}_5^{(n)} &= \mathcal{L}_1^{(n)} - \rho a \frac{\partial u_n}{\partial t} \quad \text{(left);} \\
\mathcal{L}_1^{(n)} &= \mathcal{L}_5^{(n)} + \rho a \frac{\partial u_n}{\partial t} \quad \text{(right);} \\
\mathcal{L}_2^{(n)} &= \frac{\rho}{T}\frac{\partial T}{\partial t} + \frac{\gamma - 1}{a^2}\left(\mathcal{L}_1^{(n)} + \mathcal{L}_5^{(n)}\right) + \rho W \sum_{\alpha=1}^{N} \frac{1}{W_\alpha}\frac{\partial Y_\alpha}{\partial t} - \frac{\rho}{p}\mathsf{s}_p
\end{aligned} \tag{85}$$

It is clear from these relations that the time derivatives of the normal velocity component u_n, temperature T and species mass fraction Y_α must also be specified at the boundary.

Subsonic Reflecting Inflow With Specified Density
This condition is set by imposing the values of density ρ, all three velocity components u_i and all species mass fractions Y_α at all spatial points on the boundary. All of these variables may be constant or they may be specified as functions of time. The viscous condition

$$\frac{\partial \tau_{nn}}{\partial x_n} = 0 \tag{86}$$

(no summation) is also imposed. The internal energy ρE is not specified as part of the boundary condition and must be calculated as part of the solution. The LODI form of the energy equation is

$$\begin{aligned}
\frac{\partial}{\partial t}\rho E + (E - C_v T)\left[\mathcal{L}_2^{(n)} + \frac{1}{a^2}\left(\mathcal{L}_5^{(n)} + \mathcal{L}_1^{(n)}\right)\right] + \frac{u}{a}\left(\mathcal{L}_5^{(n)} - \mathcal{L}_1^{(n)}\right) + \frac{1}{\gamma - 1}\left(\mathcal{L}_5^{(n)} + \mathcal{L}_1^{(n)}\right) & \\
+\rho u_{t1}\mathcal{L}_3^{(n)} + \rho u_{t2}\mathcal{L}_4^{(n)} + \sum_{\alpha=1}^{N} \rho \hat{h}_\alpha \mathcal{L}_{5+\alpha}^{(n)} &= 0
\end{aligned} \tag{87}$$

where u_{ti} is used to denote the two transverse velocity components at the boundary. Hence the values of all the wave amplitude variations $\mathcal{L}^{(n)}$ are required. These are obtained from the primitive variables as

$$\mathcal{L}_5^{(n)} = \mathcal{L}_1^{(n)} - \rho a \frac{\partial u_n}{\partial t} \quad \text{(left);}$$

$$
\begin{aligned}
\mathcal{L}_1^{(n)} &= \mathcal{L}_5^{(n)} + \rho a \frac{\partial u_n}{\partial t} \quad \text{(right);} \\
\mathcal{L}_2^{(n)} &= \mathcal{S}_2 - \frac{\partial \rho}{\partial t} - \frac{1}{a^2}\left(\mathcal{L}_5^{(n)} + \mathcal{L}_1^{(n)}\right) \\
\mathcal{L}_3^{(n)} &= -\frac{\partial u_{t1}}{\partial t} \\
\mathcal{L}_4^{(n)} &= -\frac{\partial u_{t2}}{\partial t} \\
\mathcal{L}_{5+\alpha}^{(n)} &= \mathcal{S}_{5+\alpha} - \frac{\partial Y_\alpha}{\partial t}
\end{aligned}
\tag{88}
$$

It is clear from these relations that the time derivatives of all the velocity components u_i together with ρ and Y_α must also be specified at the boundary.

Specifying a Turbulent Velocity Field at Inlet

A turbulent velocity field at the inlet of the computational domain is specified by passing a scanning plane through a pre–computed field of frozen turbulence. Values of all three velocity components and their time derivatives are interpolated onto the scanning plane which is then mapped onto the domain inlet plane. The speed u_{scan} at which the scanning plane moves through the box of turbulence is taken to be equal to the required constant inlet mean velocity u_{mean} plus a constant velocity increment u_{inc}, i.e. $u_{\text{scan}} = u_{\text{mean}} + u_{\text{inc}}$. The purpose of the velocity increment u_{inc} is to allow the inlet turbulence to evolve in time at the boundary, even in cases where Taylor's hypothesis is not strictly valid [12]. The location of the scanning plane is computed as $x_{\text{scan}}(t + \delta t) = x_{\text{scan}}(t) + u_{\text{scan}}\delta t$ for a time increment δt, with a suitable (arbitrary) pre–set initial location $x_{\text{scan}}(0)$.

The frozen turbulent velocity field is provided in physical space but is stored as a set of Fourier coefficients $\hat{u}_i(\bar{k}_1, x_2, x_3)$, where the Fourier transform is carried out in the boundary–normal direction only (taken as x_1 here for the purpose of illustration). The physical–space turbulent velocity components on the scanning plane are calculated using Fourier interpolation according to the discrete Fourier transform

$$
u_i(x_{\text{scan}}, x_2, x_3) = \frac{1}{L_x} \sum_{\hat{k}_x = -N_x/2}^{N_x/2} \hat{u}_i(\hat{k}_x, x_2, x_3) \exp\left(-2\pi i \bar{k}_x x_{\text{scan}}/L_x\right) \tag{89}
$$

These become the inlet–plane velocity components, after the addition of the inlet mean velocity u_{mean}. The velocity time–derivatives required by the NSCBC inlet conditions (85) and (88) are obtained by using the relation

$$
\frac{\partial u_i}{\partial t} = u_{\text{scan}} \frac{\partial u_i}{\partial x_1} \tag{90}
$$

where the spatial derivative is interpolated using the discrete Fourier transform

$$
\frac{\partial u_i}{\partial x_1}(x_{\text{scan}}, x_2, x_3) = \frac{1}{L_x^2} \sum_{\bar{k}_x = -N_x/2}^{N_x/2} -2\pi i \bar{k}_x \hat{u}_i(\bar{k}_x, x_2, x_3) \exp\left(-2\pi i \bar{k}_x x_{\text{scan}}/L_x\right). \tag{91}
$$

2.5.3 Wall Boundary Conditions

A wall boundary is treated using an impermeability condition and a no-slip condition, with all velocity components u_i set to zero, perfect reflection of acoustic waves and no convective transport of entropy, transverse velocity or mass fraction:

$$
\begin{aligned}
\mathcal{L}_1^{(n)} &= \mathcal{L}_5^{(n)}; \\
\mathcal{L}_2^{(n)} &= \mathcal{S}_2; \\
\mathcal{L}_3^{(n)} &= \mathcal{L}_4^{(n)} = 0; \\
\mathcal{L}_{5+\alpha}^{(n)} &= \mathcal{S}_{5+\alpha}
\end{aligned}
\tag{92}
$$

Isothermal Wall
For an isothermal wall there is an additional condition on the temperature:

$$T = T_{\mathrm{wall}} \tag{93}$$

where T_{wall} is a prescribed constant value of the temperature at the wall.

Adiabatic Wall
For an adabatic wall the temperature is not prescribed and instead the wall-normal component of the heat flux vector is set to zero:

$$q_n = 0. \tag{94}$$

2.6 Initial Conditions

Initial conditions are required for all of the conserved variables. Constant and spatially–uniform values are specified by default at start–up, but this is not usually sufficient and more realistic initial velocity and acalar fields must be provided.

2.6.1 Turbulence Initial Conditions

The flow field of interest is usually turbulent, and the initial turbulent velocity field must be specified with care to ensure that it is already a good approximation to a solution of the Navier–Stokes equations. This minimises the magnitude of any initial transients and ensures that a true turbulent solution is achieved as quickly as possible. A Fourier spectral method for the generation of high–quality turbulent initial velocity fields was proposed by Orszag [13] and refined by Rogallo [14], and has become the *de facto* standard procedure in DNS.

The method generates the physical–space velocity components $u_i(x, y, z; t = 0)$ corresponding to a field of incompressible homogeneous isotropic turbulence with a prescribed energy spectrum function $E(\bar{k})$, where $\bar{k}$ is the wavenumber vector magnitude in Fourier space. The spectrum function $E(\bar{k})$ is used to specify the initial values of the RMS velocity fluctuation magnitude u', the integral length scale L, the Taylor length scale λ and the Kolmogorov length scale η. Along with the viscosity ν, the spectrum function also specifies the initial turbulence energy dissipation rate ε. It should be noted that the restriction to incompressible flow does not preclude the use of the initial velocity field to start a compressible or reacting flow simulation, provided that the scalar field is also well specified.

The default spectrum function is the Batchelor–Townsend spectrum [15] which is appropriate to low–Reynolds number turbulence of the kind achievable in current DNS, although other spectrum functions are also available. The form of the Batchelor–Townsend spectrum is given by

$$E(k) = c_0 \frac{\bar{k}^4}{\bar{k}_0^5} \exp\left[-2\left(\frac{\bar{k}}{\bar{k}_0}\right)^2\right] \tag{95}$$

where the two parameters are c_0 which controls the total kinetic energy, and $\bar{k}_0$ which is the wavenumber of the maximum energy point in the spectrum. Using the Batchelor–Townsend spectrum the principal turbulence quantities may be expressed in closed form as:

Turbulent kinetic energy

$$K = \frac{3}{32}\sqrt{\frac{\pi}{2}} c_0$$

Turbulence energy dissipation rate

$$\varepsilon = \frac{15}{16}\sqrt{\frac{\pi}{2}} \pi^2 \nu c_0 \bar{k}_0^2$$

Longitudinal integral length scale

$$L_p = \frac{1}{\sqrt{2\pi}\bar{k}_0}$$

Taylor length scale

$$\lambda^2 = \frac{1}{2\pi^2 \bar{k}_0^2}$$

Kolmogorov length scale

$$\eta = \left[\frac{\nu^2}{\frac{15}{16}\sqrt{\frac{\pi}{2}}\pi^2 c_0 \bar{k}_0^2} \right]^{1/4}$$

The relevant theoretical background is described in detail by Batchelor [16]. For a given spectrum function, the Orszag–Rogallo method uses a random number generator to produce a single realisation of a three–dimensional turbulent velocity field in Fourier space. A three–dimensional parallel inverse Fourier transform then provides the required velocity field in physical space. Different statistical realisations of the same velocity field may be obtained by changing the integer seed provided as input to the random number generator. Some further details of the parallel inverse Fourier transform are provided in section 3.4 and a more detailed description of the turbulence initialisation procedure is provided in a separate report [17].

2.6.2 Thermochemical Initial Conditions

As well as the velocity field it is necessary also to specify the scalar field. This requires two thermodynamic variables (e.g., P and T), together with a set of composition variables (e.g., $Y_\alpha, \alpha = 1, N-1$) to be specified at every point in the domain. This is done most conveniently by means of a subroutine named `FLAMIN` which is called during start–up, and which provides the flexibility to specify any required problem configuration by coding it in `FORTRAN`.

A common example of an initial thermochemical field is a laminar premixed flame solution. A straightforward approach to the initialisation of a one–dimensional laminar premixed flame is to use an error–function profile for an arbitrary reaction progress variable c:

$$c(x; t = 0) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x - x_0}{\delta}\right)\right] \tag{96}$$

where x is the coordinate normal to the flame, x_0 is the location of the centre of the profile and δ is the thickness. The mass fractions of the major species may be computed directly from the progress variable using:

$$Y_\alpha(x; t = 0) = Y_{\alpha,R} + c(x)\left(Y_{\alpha,P} - Y_{\alpha,R}\right) \tag{97}$$

where the subscripts R and P denote the limiting values in the reactants and products respectively. The initial temperature may be specified using the same approach, and the initial density follows under the assumption of constant thermochemical pressure. The initial velocity may be computed by assuming constant mass flux through the flame. If required, initial profiles of minor species may be specified using a Gaussian function. The procedure as outlined is often sufficient to allow a laminar flame solution to develop fairly rapidly, even when employing a detailed chemical reaction mechanism. The resulting one–dimensional solution may be used subsequently to initialise a three–dimensional turbulent flame simulation.

3 Numerical Formulation

The governing equations are solved numerically using high–order methods to ensure the high accuracy required for DNS. The set of differential equations (1–4) expressed in conservative form is first expanded out to expose the derivatives of all quantities of interest. The continuity equation becomes:

$$\frac{\partial}{\partial t}\rho = -\frac{\partial}{\partial x}\rho u - \frac{\partial}{\partial y}\rho v - \frac{\partial}{\partial z}\rho w, \tag{98}$$

In the remaining conservation equations, a skew-symmetric form is used for the convective terms in order to minimise the spatial coherence of the discretisation error [18]. The remaining terms are expanded out in derivatives of single quantities. The Navier–Stokes equation for the x–component of momentum is written as:

$$\begin{aligned}
\frac{\partial}{\partial t}\rho u = \; & - \; \frac{1}{2}\left(\frac{\partial}{\partial x}\rho uu + \frac{\partial}{\partial y}\rho vu + \frac{\partial}{\partial z}\rho wu\right) \\
& - \; \frac{1}{2}\left(\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} + \rho w\frac{\partial u}{\partial z}\right) - \frac{1}{2}u\left(\frac{\partial}{\partial x}\rho u + \frac{\partial}{\partial y}\rho v + \frac{\partial}{\partial z}\rho w\right) \\
& - \; \frac{\partial p}{\partial x} \\
& + \; \mu\left[\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x^2} + \frac{1}{3}\left(\frac{\partial^2 v}{\partial x\partial y} + \frac{\partial^2 w}{\partial x\partial z}\right)\right] \\
& + \; \left[\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial \mu}{\partial x} + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial y} + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\frac{\partial \mu}{\partial z}
\end{aligned} \tag{99}$$

The y–momentum equation is:

$$\begin{aligned}
\frac{\partial}{\partial t}\rho v = \; & - \; \frac{1}{2}\left(\frac{\partial}{\partial x}\rho uv + \frac{\partial}{\partial y}\rho vv + \frac{\partial}{\partial z}\rho wv\right) \\
& - \; \frac{1}{2}\left(\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} + \rho w\frac{\partial v}{\partial z}\right) - \frac{1}{2}v\left(\frac{\partial}{\partial x}\rho u + \frac{\partial}{\partial y}\rho v + \frac{\partial}{\partial z}\rho w\right) \\
& - \; \frac{\partial p}{\partial y} \\
& + \; \mu\left[\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial z^2} + \frac{1}{3}\left(\frac{\partial^2 u}{\partial y\partial x} + \frac{\partial^2 w}{\partial y\partial z}\right)\right] \\
& + \; \left[\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial \mu}{\partial y} + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial x} + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\frac{\partial \mu}{\partial z}
\end{aligned} \tag{100}$$

while the z–momentum equation is:

$$\begin{aligned}
\frac{\partial}{\partial t}\rho w = \; & - \; \frac{1}{2}\left(\frac{\partial}{\partial x}\rho uw + \frac{\partial}{\partial y}\rho vw + \frac{\partial}{\partial z}\rho ww\right) \\
& - \; \frac{1}{2}\left(\rho u\frac{\partial w}{\partial x} + \rho v\frac{\partial w}{\partial y} + \rho w\frac{\partial w}{\partial z}\right) - \frac{1}{2}w\left(\frac{\partial}{\partial x}\rho u + \frac{\partial}{\partial y}\rho v + \frac{\partial}{\partial z}\rho w\right) \\
& - \; \frac{\partial p}{\partial z} \\
& + \; \mu\left[\frac{4}{3}\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{1}{3}\left(\frac{\partial^2 u}{\partial z\partial x} + \frac{\partial^2 v}{\partial z\partial y}\right)\right] \\
& + \; \left[\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\frac{\partial \mu}{\partial z} + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\frac{\partial \mu}{\partial x} + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\frac{\partial \mu}{\partial y}
\end{aligned} \tag{101}$$

The energy equation is treated in the same manner:

$$
\begin{aligned}
\frac{\partial}{\partial t}\rho E = & - \frac{1}{2}\left(\frac{\partial}{\partial x}\rho u E + \frac{\partial}{\partial y}\rho v E + \frac{\partial}{\partial z}\rho w E\right) \\
& - \frac{1}{2}\left(\rho u \frac{\partial E}{\partial x} + \rho v \frac{\partial E}{\partial y} + \rho w \frac{\partial E}{\partial z}\right) - \frac{1}{2} E \left(\frac{\partial}{\partial x}\rho u + \frac{\partial}{\partial y}\rho v + \frac{\partial}{\partial z}\rho w\right) \\
& - p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) - u\frac{\partial p}{\partial x} - v\frac{\partial p}{\partial y} - w\frac{\partial p}{\partial z} \\
& + u\mu\left[\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial x^2} + \frac{1}{3}\left(\frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 w}{\partial x \partial z}\right)\right] \\
& + u\left[\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial \mu}{\partial x} + u\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial y} + u\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\frac{\partial \mu}{\partial z} \\
& + v\mu\left[\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial z^2} + \frac{1}{3}\left(\frac{\partial^2 u}{\partial y \partial x} + \frac{\partial^2 w}{\partial y \partial z}\right)\right] \\
& + v\left[\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial \mu}{\partial y} + v\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\frac{\partial \mu}{\partial x} + v\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\frac{\partial \mu}{\partial z} \\
& + w\mu\left[\frac{4}{3}\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{1}{3}\left(\frac{\partial^2 u}{\partial z \partial x} + \frac{\partial^2 v}{\partial z \partial y}\right)\right] \\
& + w\left[\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\frac{\partial \mu}{\partial z} + w\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\frac{\partial \mu}{\partial x} + w\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\frac{\partial \mu}{\partial y} \\
& + \mu\left[\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial u}{\partial x} + \mu\left[\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)\right]\frac{\partial v}{\partial y} + \mu\left[\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\frac{\partial w}{\partial z} \\
& + \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2 \\
& + \lambda\frac{\partial^2 T}{\partial x^2} + \lambda\frac{\partial^2 T}{\partial y^2} + \lambda\frac{\partial^2 T}{\partial z^2} + \frac{\partial T}{\partial x}\frac{\partial \lambda}{\partial x} + \frac{\partial T}{\partial y}\frac{\partial \lambda}{\partial y} + \frac{\partial T}{\partial z}\frac{\partial \lambda}{\partial z} \\
& + \sum_{\alpha=1}^{N}\rho D_\alpha h_\alpha\left(\frac{\partial^2 Y_\alpha}{\partial x^2} + \frac{\partial^2 Y_\alpha}{\partial y^2} + \frac{\partial^2 Y_\alpha}{\partial z^2}\right) + \sum_{\alpha=1}^{N}\rho D_\alpha\left(\frac{\partial Y_\alpha}{\partial x}\frac{\partial h_\alpha}{\partial x} + \frac{\partial Y_\alpha}{\partial y}\frac{\partial h_\alpha}{\partial y} + \frac{\partial Y_\alpha}{\partial z}\frac{\partial h_\alpha}{\partial z}\right) \\
& + \sum_{\alpha=1}^{N} h_\alpha\left(\frac{\partial Y_\alpha}{\partial x}\frac{\partial}{\partial x}\rho D_\alpha + \frac{\partial Y_\alpha}{\partial y}\frac{\partial}{\partial y}\rho D_\alpha + \frac{\partial Y_\alpha}{\partial z}\frac{\partial}{\partial z}\rho D_\alpha\right) \qquad (102)
\end{aligned}
$$

Finally, the conservation equation for the mass fraction of species α becomes:

$$
\begin{aligned}
\frac{\partial}{\partial t}\rho Y_\alpha = & - \frac{1}{2}\left(\frac{\partial}{\partial x}\rho u Y_\alpha + \frac{\partial}{\partial y}\rho v Y_\alpha + \frac{\partial}{\partial z}\rho w Y_\alpha\right) \\
& - \frac{1}{2}\left(\rho u \frac{\partial Y_\alpha}{\partial x} + \rho v \frac{\partial Y_\alpha}{\partial y} + \rho w \frac{\partial Y_\alpha}{\partial z}\right) - \frac{1}{2} Y_\alpha \left(\frac{\partial}{\partial x}\rho u + \frac{\partial}{\partial y}\rho v + \frac{\partial}{\partial z}\rho w\right) \\
& + w_\alpha \\
& + \rho D_\alpha\left(\frac{\partial^2 Y_\alpha}{\partial x^2} + \frac{\partial^2 Y_\alpha}{\partial y^2} + \frac{\partial^2 Y_\alpha}{\partial z^2}\right) + \frac{\partial Y_\alpha}{\partial x}\frac{\partial}{\partial x}\rho D_\alpha + \frac{\partial Y_\alpha}{\partial y}\frac{\partial}{\partial y}\rho D_\alpha + \frac{\partial Y_\alpha}{\partial z}\frac{\partial}{\partial z}\rho D_\alpha \qquad (103)
\end{aligned}
$$

where the chemical reaction rate w_α is given by 13.

3.1 Spatial Discretisation

The computational domain is taken to be a cuboid of size (L_x, L_y, L_z), and is discretised in space using a structured Cartesian mesh (see Figure 1). The mesh contains (N_x, N_y, N_z) points in each direction with uniform spacings given by

$$\delta x = \frac{L_x}{N_x - 1}; \qquad \delta y = \frac{L_y}{N_y - 1}; \qquad \delta z = \frac{L_z}{N_z - 1}. \tag{104}$$

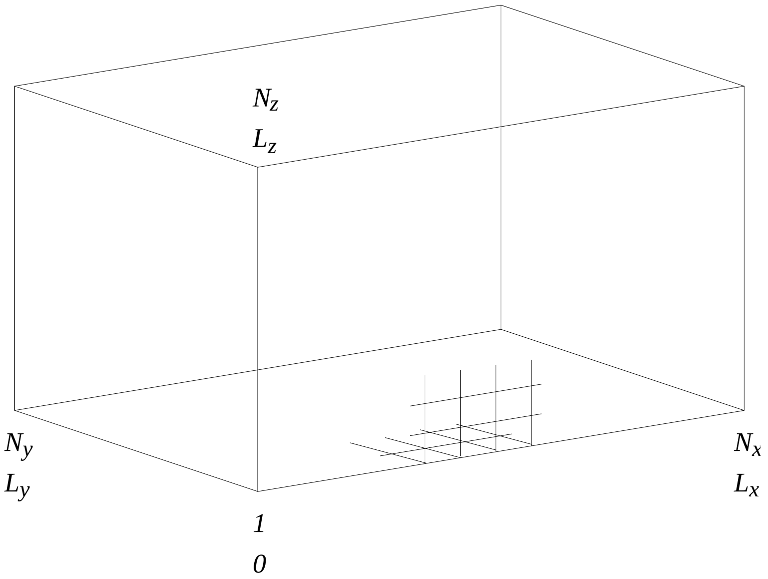

Figure 1: The computational domain.

Each of the spatial derivatives in the conservation equations is evaluated at each mesh point in each direction using a high–order explicit centred finite difference operator. For the first derivatives, the difference formula for any quantity f at a general interior mesh point i is:

$$f'_i = \sum_{m=1}^{M/2} a_m^{(M)} \frac{f_{i+m} - f_{i-m}}{2mh} \tag{105}$$

where M is the order of the scheme whose coefficients are denoted by $a_m^{(M)}$, and $h \in \{\delta x, \delta y, \delta z\}$ is the mesh spacing in the required direction.

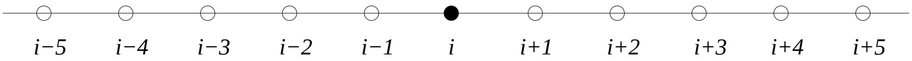

Figure 2: Interior–point stencil for the 10th order finite–difference scheme.

A tenth–order centred scheme (i.e. $M = 10$), requiring a stencil of eleven points in total (see Figure 2), is used for all interior points that are five or more points away from a non–periodic boundary. The order of the centred scheme is reduced as the boundary is approached (see Figure 3). At the fourth point from the boundary, the scheme is eighth order. At the third point it is sixth order and at the second point it is fourth order. Coefficients for all of these centred first derivative schemes are given in Table 1.

At the first interior point a fourth–order ($M = 4$) skewed scheme (denoted by S) is used. For a boundary located at $i = 1$ the difference formula for the point $i = 2$ is:

$$f'_i = a_1^{M,S} \frac{f_i - f_{i+1}}{h} + \sum_{m=2}^{M} a_m^{(M,S)} \frac{f_{i+m} - f_{i+1}}{(m-1)h} \tag{106}$$

m	1	2	3	4	5
$a_m^{(4)}$	4/3	-1/3			
$a_m^{(6)}$	3/2	-3/5	1/10		
$a_m^{(8)}$	8/5	-4/5	8/35	-1/35	
$a_m^{(10)}$	5/3	-20/21	5/14	-5/63	1/126

Table 1: Difference coefficients for the centred first–derivative schemes.

At the boundary point, a fourth-order ($M = 4$) one–sided scheme (denoted by O) is used according to the difference formula:

$$f_i' = \sum_{m=1}^{M} a_m^{(M,O)} \frac{f_{i+m} - f_i}{mh}. \tag{107}$$

Both of these schemes are also used for a boundary located at $i \in \{N_x, N_y, N_z\}$ after a straightforward reflection of coordinates.

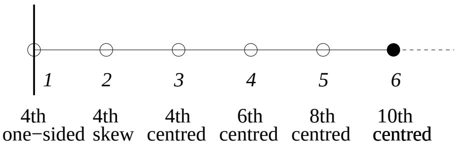

Figure 3: Boundary stencil for 4th to 10th order finite–difference schemes.

m	1	2	3	4
$a_m^{(4,S)}$	-1/4	3/2	-1/2	1/12
$a_m^{(4,O)}$	4	-3	4/3	-1/4

Table 2: Difference coefficients for the skewed and one–sided first–derivative boundary schemes.

For the second derivatives, the difference formula at a general interior mesh point is

$$f_i'' = \sum_{m=1}^{M/2} b_m^{(M)} \frac{f_{i+m} - 2f_i + f_{i-m}}{[mh]^2} \tag{108}$$

where the coefficients of the second–derivative scheme are denoted by $b_m^{(M)}$. A tenth–order centred scheme is used for interior points five or more points distant from non–periodic boundaries, and reduced–order centred schemes are used in the same manner as for the first derivatives as the boundary is approached. It is interesting to note that for the second derivative centred schemes the difference coefficients are identical to those for the first derivative centred schemes, i.e. $b_m^{(M)} = a_m^{(M)}$. At the first interior point away from the boundary the skewed fourth–order second–derivative scheme is given by

$$f_i'' = b_1^{(M,S)} \frac{f_i - f_{i+1}}{h^2} + \sum_{m=2}^{M+1} b_m^{(M,S)} \frac{f_{i+m} - f_{i+1}}{[(m-1)h]^2} \tag{109}$$

where $M = 4$ and $i = 2$. At the boundary point the one–sided second–derivative scheme is

$$f_i'' = \sum_{m=1}^{M+1} b_m^{(M,O)} \frac{f_{i+m} - f_i}{[mh]^2} \tag{110}$$

where $M = 4$ and $i = 1$. Again, both of these schemes are also used for $i \in \{N_x, N_y, N_z\}$ after reflection of coordinates.

m	1	2	3	4	5
$b_m^{(4,S)}$	5/6	-1/3	7/6	-1/2	1/12
$b_m^{(4,O)}$	-77/6	107/6	-13	61/12	-5/6

Table 3: Difference coefficients for the skewed and one–sided second–derivative boundary schemes.

The second cross–derivatives are evaluated using a compatible set of schemes. For a general interior point (i, j) the difference formula is

$$f''_{i,j} = \sum_{m=1}^{M/2} c_m^{(M)} \frac{f_{i+m,j+m} - f_{i+m,j-m} - f_{i-m,j+m} + f_{i-m,j-m}}{4m^2 h_1 h_2} \tag{111}$$

where the coefficients are denoted by $c_m^{(M)}$ and the mesh spacings are $h_1, h_2 \in \{\delta x, \delta y, \delta z\}$. A tenth–order scheme is used for interior points at least five points distant from the boundary in either of the two relevant directions, As a non–periodic boundary is approached in either direction, the order of the scheme is reduced as for the first and second derivatives. The difference coefficients for the centred second cross–derivative schemes are identical to those for the centred first derivative schemes, i.e. $c_m^{(M)} = a_m^{(M)}$.

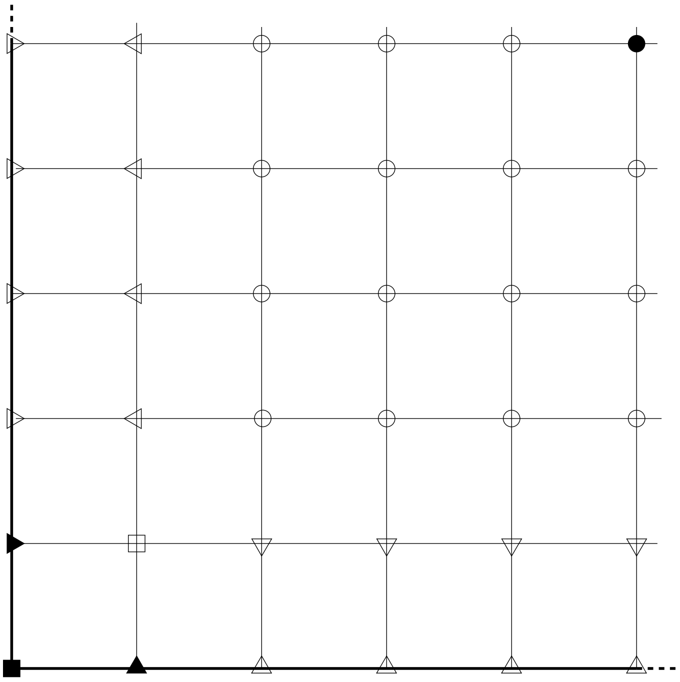

Figure 4: Bottom–left corner of the mesh for second cross–derivative evaluation. Points requiring special treatment are indicated.

The mesh in the vicinity of a bottom–right non-periodic boundary corner is shown in Figure 4. The point at the top right indicated by a filled circle is the closest point to the corner at which the full 10th–order centred cross–derivative scheme can be applied. Close to non-periodic boundaries there are five special cases as indicated in the Figure:

1. For a corner point (filled square), a doubly one–sided fourth order scheme is used which evaluates the first derivative of the first derivative by combining two instances of (107), one for each direction.
2. For a point on the boundary adjacent to a corner (filled triangles), a mixed one–sided/skewed fourth–order scheme is used combining instances of (106) and (107);
3. For an interior point located one point away from a corner in both directions (open square), a doubly–skewed fourth–order scheme is used which combines two instances of (106);
4. For an interior point located one point away from the boundary in one direction but two or more points away from the boundary in the other direction (open inverted and left–facing triangles), a mixed skewed/standard fourth order scheme is used combining instances of (105) and (106);
5. For a boundary point located two or more points away from the boundary in the other direction (open upright and right–facing triangles), a mixed one–sided/standard fourth order scheme is used which combines instances of (105) and (107).

In all of the special cases, the appropriate combined differencing scheme is applied to boundaries on all sides after suitable coordinate reflections.

3.2 Parallel Domain Decomposition

Parallel computation is facilitated using a simple domain decomposition approach. The global computational domain is decomposed into cuboidal sub–domains which are each assigned to a CPU core. The total number of cores required is $P = P_x P_y P_z$ where P_x, P_y and P_z are the numbers of cores assigned to the x, y and z directions respectively. The cores are ranked from $p = 0$ to $p = P - 1$, and are also given Cartesian indices (p_x, p_y, p_z) ranging in $x - y - z$ order from $(1, 1, 1)$ to $(P_x, P_x.P_z)$. Hence

$$p = (p_x - 1) + (p_y - 1)P_x + (p_z - 1)P_x P_y \tag{112}$$

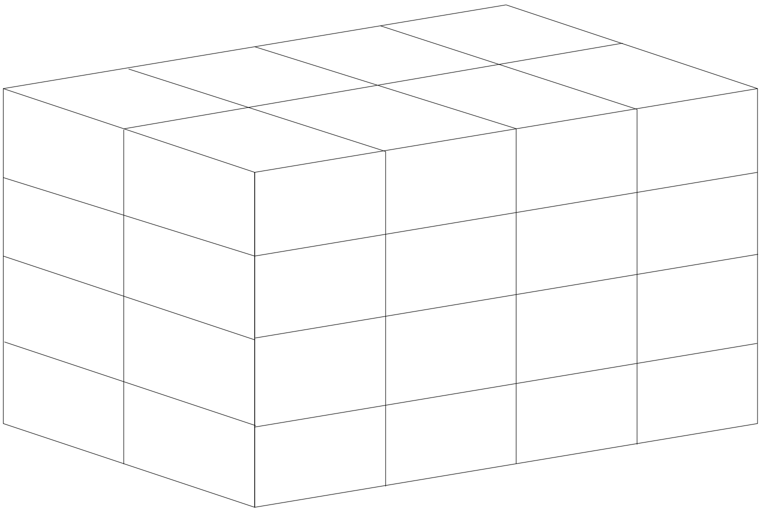

Figure 5: Example of parallel domain decomposition.

The domain decomposition is carried out by dividing the number of mesh points on each side of the global domain by the number of cores assigned to that direction. Hence the number of points allocated to the core with rank p is given by $(n_x^{(p)}, n_y^{(p)}, n_z^{(p)})$ where $n_x^{(p)} = N_x/P_x$ and similarly for the y and z directions. Load–balancing is achieved by ensuring that the same number of mesh points is allocated to each sub–domain. Where this is not possible, for example due to N_x not being divisible by P_x, an integer division is carried out in each direction and the remainder is allocated to the first–indexed core in that direction. Figure 5 shows an example of a domain decomposition with $P_x = 4$, $P_y = 2$, $P_z = 4$ and equal sub–domains. Communication between adjacent sub–domains is done using a layer of halo cells on the surface of each sub–domain. Since the interior tenth–order differencing stencil requires five points on each side of the central point (see Figure 2), the halo layer is five points thick. For the scalar quantities, a halo layer is required only on each face of each sub–domain. For the velocity components, the need to evaluate second cross–derivatives means that the halo layer must include the edges and corners also.

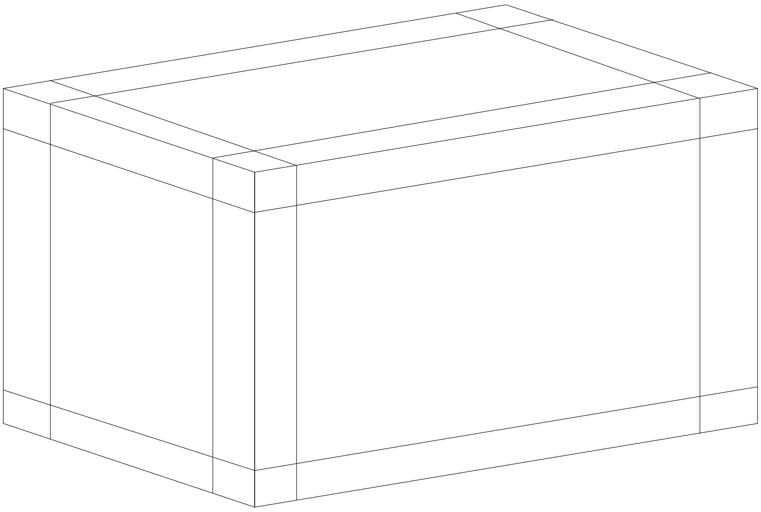

Figure 6: Full halo layer on the surface of a parallel sub–domain.

For a sub–domain of size $n_x n_y n_z$ and a halo layer thickness h, the number of points in the augmented sub–domain illustrated in Figure 6 is given by

$$(n_x + 2h)(n_y + 2h)(n_z + 2h) = n_x n_y n_z + 2h\,(n_x n_y + n_x n_z + n_y n_z) + 4h^2\,(n_x + n_y + n_z) + 8h^3 \quad (113)$$

The terms on the right–hand side correspond to the number of points in the original sub–domain, the number of points in the halos attached to each face of the original sub–domain, the number of points in the halo strips along each edge, and finally the number of points in the eight corners of the halo. The fractional increase in the number of points compared to the original sub–domein is given by the expression

$$\frac{(n_x + 2h)(n_y + 2h)(n_z + 2h)}{n_x n_y n_z} - 1 = \frac{2h}{n_x} + \frac{2h}{n_y} + \frac{2h}{n_z} + \frac{4h^2}{n_x n_y} + \frac{4h^2}{n_x n_z} + \frac{4h^2}{n_y n_z} + \frac{8h^3}{n_x n_y n_z} \quad (114)$$

Since in general $h << n_{x,y,z}$, it is clear that the computational overhead associated with the halo is dominated by the first three terms on the right–hand side. The lowest overhead is obtained when $n_x = n_y = n_z = n$, i.e. when the surface–to–volume ratio of the sub–domain is at a minimum. Parallel communication is implemented using MPI.

3.3 Time Stepping

Time advancement of the solution is carried out using a low–storage explicit Runge–Kutta method with adaptive time step control [5]. The governing equations (1)–(4) are written as

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}(\mathbf{U}, t) \quad (115)$$

where $\mathbf{U}$ is the vector of conserved variables given by $\mathbf{U} = \{\rho, \rho u, \rho v, \rho w, \rho E, \rho Y_\alpha\}^T$ and $\mathbf{R}$ is the right–hand side vector containing all other terms in the equations. Clearly $\mathbf{R}$ depends on $\mathbf{U}$ and may also depend explicitly on time, for example through source terms or boundary conditions. The general s–stage explicit Runge–Kutta scheme may be stated as:

$$\begin{aligned}
\mathbf{U}^{(i)} &= \mathbf{U}^{(n)} + \delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{R}^{(j)} \\
\mathbf{U}^{(n+1)} &= \mathbf{U}^{(n)} + \delta t \sum_{j=1}^{s} b_j \mathbf{R}^{(j)} \\
t^{(i)} &= t^{(n)} + c_i \delta t \\
\hat{\mathbf{U}}^{(n+1)} &= \mathbf{U}^{(n)} + \delta t \sum_{j=1}^{s} \hat{b}_j \mathbf{R}^{(j)} \qquad (116)
\end{aligned}$$

where $\mathbf{R}^{(i)} = \mathbf{R}(\mathbf{U}^{(i)}, t^{(i)})$ and the last line denotes the lower–order embedded scheme used to provide error estimates for the adaptive time–step controller. Memory requirements are minimised by using a two–register scheme which requires the storage of only $\mathbf{U}^{(i)}$ and $\mathbf{R}^{(i)}$ at each stage. In practice it is necessary also to store $\hat{\mathbf{U}}^{(i)}$ and to provide some temporary storage for intermediate data required in the evaluation of $\mathbf{R}^{(i)}$. The order of the scheme is determined by constraints on the coefficients a_{ij}, b_i and c_i while the order of the embedded scheme is determined using the coefficients $\hat{b}_i$ in place of b_i. Several schemes were assessed, and a five–stage fourth–order two–register method with a third–order embedded scheme was chosen. This scheme is denoted RK4(3)5[2R+]C in the classification of Kennedy at al. [5] and its Butcher array is given in Table 4.

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & b_1 & a_{32} & & & \\
c_4 & b_1 & b_2 & a_{43} & & \\
c_5 & b_1 & b_2 & b_3 & a_{54} & \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5
\end{array}$$

Table 4: Butcher array for the RK4(3)5[2R+]C explicit Runge–Kutta method.

The numerical values of the coefficients are given in Table 5.

a_{21}	$\frac{970286171893}{4311952581923}$	b_1	$\frac{1153189308089}{22510343858157}$	$\hat{b}_1$	$\frac{1016888040809}{7410784769900}$
a_{32}	$\frac{6584761158862}{12103376702013}$	b_2	$\frac{1772645290293}{4653164025191}$	$\hat{b}_2$	$\frac{11231460423587}{58533540763752}$
a_{43}	$\frac{2251764453980}{15575788980749}$	b_3	$\frac{-1672844663538}{4480602732383}$	$\hat{b}_3$	$\frac{-1563879915014}{6823010717585}$
a_{54}	$\frac{26877169314380}{34165994151039}$	b_4	$\frac{2114624349019}{3568978502595}$	$\hat{b}_4$	$\frac{606302364029}{971179775848}$
		b_5	$\frac{5198255086312}{14908931495163}$	$\hat{b}_5$	$\frac{1097981568119}{3980877426909}$

Table 5: Coefficients for the RK4(3)5[2R+]C explicit Runge–Kutta method.

The two–register Runge–Kutta method is implemented as a sequence of sub–steps. At the beginning of a new time step, the time is set to $t^{(1)} = t^{(n)}$. The first register contains the initial stage value $\mathbf{S}^{(1)} = \mathbf{U}^{(n)}$ while the second register contains the initial solution value $\mathbf{U}^{(1)} = \mathbf{U}^{(n)}$. The error accumulator $\mathbf{E}^{(1)}$ is set initially to zero. In each sub–step j, the right–hand side function $\mathbf{R}^{(j)}$ is evaluated using the current time and the current solution value $\mathbf{U}^{(j)}$. The new value of $\mathbf{R}^{(j)}$ is stored in Register 2, overwriting $\mathbf{U}^{(j)}$. Then new values of both $\mathbf{S}$ and $\mathbf{U}$ are formed by linear combinations of $\mathbf{S}^{(j)}$ and $\mathbf{R}^{(j)}$ and are stored in place. At the final step, only $\mathbf{S}$ is updated and becomes the new solution at time $t^{(n+1)}$. The error accumulator is updated at each sub–step as new values of $\mathbf{R}$ become available.

Time	Register 1	Register 2	Error accumulator
$t^{(1)}$	$\mathbf{S}^{(1)}$	$\mathbf{U}^{(1)}$	$\mathbf{E}^{(1)}$
	$\mathbf{S}^{(1)}$	$\mathbf{R}^{(1)}\left(\mathbf{U}^{(1)}, t^{(1)}\right)$	$\mathbf{E}^{(1)}$
	$\mathbf{S}^{(2)} = \mathbf{S}^{(1)} + b_1 \delta t \mathbf{R}^{(1)}$	$\mathbf{U}^{(2)} = \mathbf{S}^{(1)} + a_{21} \delta t \mathbf{R}^{(1)}$	$\mathbf{E}^{(2)} = \mathbf{E}^{(1)} + \left(b_1 - \hat{b}_1\right) \delta t \mathbf{R}^{(1)}$
$t^{(2)} = t^{(1)} + c_2 \delta t$	$\mathbf{S}^{(2)}$	$\mathbf{R}^{(2)}\left(\mathbf{U}^{(2)}, t^{(2)}\right)$	$\mathbf{E}^{(2)}$
	$\mathbf{S}^{(3)} = \mathbf{S}^{(2)} + b_2 \delta t \mathbf{R}^{(2)}$	$\mathbf{U}^{(3)} = \mathbf{S}^{(2)} + a_{32} \delta t \mathbf{R}^{(2)}$	$\mathbf{E}^{(3)} = \mathbf{E}^{(2)} + \left(b_2 - \hat{b}_2\right) \delta t \mathbf{R}^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t^{(5)} = t^{(1)} + c_5 \delta t$	$\mathbf{S}^{(5)}$	$\mathbf{R}^{(5)}\left(\mathbf{U}^{(5)}, t^{(5)}\right)$	$\mathbf{E}^{(5)}$
	$\mathbf{S}^{(6)} = \mathbf{S}^{(5)} + b_5 \delta t \mathbf{R}^{(5)}$	$\mathbf{U}^{(5)}$	$\mathbf{E}^{(6)} = \mathbf{E}^{(5)} + \left(b_5 - \hat{b}_5\right) \delta t \mathbf{R}^{(5)}$
$t^{(n+1)} = t^{(5)}$	$\mathbf{U}^{(n+1)} = \mathbf{S}^{(6)}$	$\mathbf{U}^{(n+1)}$	$\mathbf{E}^{(n+1)} = \mathbf{E}^{(6)}$

Table 6: Sequence of sub–steps for Runge–Kutta implementation.

The time step is set adaptively using a PID–based controller [5]. The new time step $\delta t^{(n+1)}$ is given by

$$\delta t^{(n+1)} = \kappa \delta t^{(n)} \left(\frac{\varepsilon}{||E^{(n+1)}||_\infty} \right)^\alpha \left(\frac{||E^{(n)}||_\infty}{\varepsilon} \right)^\beta \left(\frac{\varepsilon}{||E^{(n-1)}||_\infty} \right)^\gamma \tag{117}$$

where E is an error estimate obtained by finding the maximum of the normalised elements of $\mathbf{E}$, ε is a pre–set error tolerance, κ is a safety factor and α, β and γ are the parameters of the controller. The controller requires some tuning for optimal performance, but good results have been obtained with $\varepsilon = 1.0 \times 10^{-3}$, $\kappa = 0.9$, $\alpha = 0.49/p$, $\beta = 0.34/p$ and $\gamma = 0.1/p$ where $p = 3$ is the order of the embedded method.

3.4 Initial Turbulent Field

A three–dimensional parallel inverse Fourier transform is implemented as part of the turbulence initialisation procedure. The three–dimensional transform algorithm is based on a superposition of three separate one–dimensional transforms. During each one–dimensional transform, all of the data along each single line of the global computational mesh (i.e. a single “pencil” of data) is gathered onto the lowest–ranked processor on that line. A one–dimensonal inverse Fourier transform is carried out on that processor, and the pencil of transformed data is scattered back to its original locations. Multiple pencils may be handled at the same time, in order to reduce parallel communication overheads. The one–dimensional transform is based on the prime–factor fast Fourier transform algorithm [19] as implemented in a locally–developed FFT library [20]. In principle the FFT algorithm will handle any length of transform data, although optimal performance is obtained for lengths which are powers of two or three. The final one–dimensional transform procedure includes a step which imposes the symmetry conditions necessary to ensure that the physical–space velocity field is purely real. Further details are given in a separate report [17].

4 Running the Code

Setting up a DNS run using SENGA2 requires several steps. First the size of the computational domain and the size parameters for the thermochemical problem must be set by editing the COMMON blocks. Then the control parameters for the run must be set using the control data file, and the relevant chemical data must be provided using the chemical data file.

4.1 Common Data

The common data for SENGA2 is contained in the COMMON block file com_senga2.h. This file must be edited in order to set the global and local sizes of the computational domain, the number of processors and the size of the parallel transfer array.

The section PHYDIM is the first section of the file com_senga2.h and an example is shown below:

```
C     PHYDIM-----------------------------------------------------------------

C     PHYSICAL DIMENSIONS OF ARRAYS
C     -----------------------------
C     NOTE: ALL ARRAY SIZES MUST BE CONSISTENT
C     NXSIZE MUST BE >= NXGLBL/NXPROC
C     NYSIZE MUST BE >= NYGLBL/NYPROC
C     NZSIZE MUST BE >= NZGLBL/NZPROC
C     WITH AN EXTRA ALLOWANCE FOR ANY REMAINDER
C
C     GLOBAL GRID SIZE
      INTEGER NXGLBL,NYGLBL,NZGLBL
      PARAMETER(NXGLBL=64, NYGLBL=64, NZGLBL=64)
      INTEGER NGZMAX
C     SET NGZMAX=MAX(NXGLBL,NYGLBL,NZGLBL)
      PARAMETER(NGZMAX=NXGLBL)

C     NUMBER OF PROCESSORS
      INTEGER NXPROC,NYPROC,NZPROC
      PARAMETER(NXPROC=2, NYPROC=2, NZPROC=2)
      INTEGER NPRMAX
C     SET NPRMAX=MAX(NXPROC,NYPROC,NZPROC)
      PARAMETER(NPRMAX=NYPROC)

C     LOCAL GRID SIZE
      INTEGER NXSIZE,NYSIZE,NZSIZE
      PARAMETER(NXSIZE=32, NYSIZE=32, NZSIZE=32)
      INTEGER NSZMAX
C     SET NSZMAX=MAX(NXSIZE,NYSIZE,NZSIZE)
      PARAMETER(NSZMAX=NZSIZE)

C     SIZE OF HALO
      INTEGER NHALOX,NHALOY,NHALOZ
      PARAMETER(NHALOX=5,NHALOY=5,NHALOZ=5)

C     SIZE OF PARALLEL TRANSFER ARRAY
C     NPARAY MUST BE >= MAX(NHALOX,NHALOY,NHALOZ)
C                           *MAX((NXSIZE+2*NHALOX)*(NYSIZE+2*NHALOY),
C                                (NXSIZE+2*NHALOX)*(NZSIZE+2*NHALOZ),
C                                  (NYSIZE+2*NHALOY)*(NZSIZE+2*NHALOZ))
C     AND ALSO LARGE ENOUGH FOR PARALLEL BROADCAST OF CHEMICAL DATA
C     AND ALSO LARGE ENOUGH FOR PARALLEL TRANSFER OF INITIAL TURBULENCE DATA
      INTEGER NPARAY
      PARAMETER(NPARAY=8820)
      .
      .
      .
C     PHYDIM-----------------------------------------------------------------
```

Here, the parameters NXGLBL, NYGLBL and NZGLBL must be set to define the global size of the computational domain in terms of the number of grid points required in the x, y and z directions respectively. These must be the exact global domain sizes required. The parameter NGZMAX must be set equal to the largest of the three global domain sizes.

It is assumed that the problem will be decomposed over a topologically cubical array of processors. The parameters NXPROC, NYPROC and NZPROC must be set to define the number of processors to be used in each direction, and NPRMAX must be set equal to the largest of these.

The parameters NXSIZE, NYSIZE and NZSIZE must be set to define the maximum size of the computational domain on each processor. The global domain is decomposed by SENGA2 as evenly as possible between the processors in each direction: thus NXSIZE must be set to a value greater than or equal to NXGLBL/NXPROC. If NXGLBL is not exactly divisible by NXPROC then the remainder is evenly distributed between the highest-ranked processors in the x-direction. In that case NXSIZE must be set greater than or equal to 1 + NXGLBL div NXPROC. Clearly the same procedure must be followed also for the other two directions in setting NYSIZE and NZSIZE. Then NSZMAX must be set equal to the largest of the three local domain sizes.

The parameters NHALOX, NHALOY and NHALOZ must be set to define the width of the halo layer of the computational grid that is passed between adjacent processors in each direction, and the value set must match the corresponding interior spatial differencing scheme in use. These parameters must be set even if the problem uses only a single processor. The default values for the standard 10th order interior differencing scheme are NHALOX=NHALOY=NHALOZ=5.

The parameter NPARAY controls the the size of the parallel transfer array. The value of NPARAY must be set greater than or equal to the maximum halo size required in each direction, i.e. the maximum of NHALOX, NHALOY, NHALOZ, multiplied by the maximum of
(NXSIZE+2*NHALOX)*(NYSIZE+2*NHALOY) for the z-direction,
(NXSIZE+2*NHALOX)*(NZSIZE+2*NHALOZ) for the y-direction, and
(NYSIZE+2*NHALOY)*(NZSIZE+2*NHALOZ) for the x-direction.
The value of NPARAY must be greater than or equal to the maximum required to broadcast the control data and the chemical data during initialisation of the code. This is checked automatically during start-up.

If an initial turbulent field is required, it may be useful to edit the the section IFTURB of the file com_senga2.h which is shown below:

```
C     IFTURB-----------------------------------------------------------------

C     DATA FOR INITIAL TURBULENCE FIELD

C     NUMBER OF PENCILS
      INTEGER NPENMX
      PARAMETER(NPENMX=16)

      DOUBLE PRECISION FFTROW(2*NGZMAX,3,NPENMX),
     +                 FTPART(2*NSZMAX,3,NPENMX),
     +                 FFTINX(2*NGZMAX)
```

```
      COMMON/IFTURB/FFTROW,FTPART,FFTINX

C     IFTURB-------------------------------------------------------------
```

The parameter NPENMX controls the number of FFT pencils (i.e. lines of one-dimensional transform data) which are transferred between processors in a single batch. The minimum value is NPENMX=1, and any larger value provides a gain in the speed of inter-processor communication during the generation of the initial turbulent field at the cost of the memory required to store the transform data.

The section CHEMIC of the file com_senga2.h contains the thermochemical data and is shown below:

```
C     CHEMIC-------------------------------------------------------------

C     PARAMETERS
C     ==========
C     MAX NO OF SPECIES, NO OF STEPS
      INTEGER NSPCMX,NSTPMX
      PARAMETER(NSPCMX=1, NSTPMX=1)

C     THERMODYNAMIC DATA
C     MAX NO OF TEMPERATURE INTERVALS, THERMO POLYNOMIAL COEFFICIENTS
      INTEGER NTINMX,NCOFMX
      PARAMETER(NTINMX=2, NCOFMX=7)
C     MAX NO OF TEMPERATURE COEFFICIENTS, DITTO MINUS ONE
      INTEGER NCTMAX,NCTMM1
      PARAMETER(NCTMAX=5, NCTMM1=NCTMAX-1)

C     TEMPERATURE INTERVAL INDEXING
C     NTBASE = NUMBER BASE FOR INDEXING:
C     MUST BE A POWER OF TWO >= MAX NO OF TEMPERATURE INTERVALS PER SPECIES
C     NSPIMX = MAX NO OF SPECIES STORED PER SINGLE (32-BIT) SIGNED INTEGER:
C     MUST BE SET EQUAL TO 31 DIV LOG2(NTBASE)
C     NINTMX = NO OF INTEGERS REQUIRED PER SPATIAL POINT:
C     MUST BE SET EQUAL TO (1 + NSPCMX DIV NSPIMX)
      INTEGER NTBASE,NSPIMX,NINTMX
      PARAMETER(NTBASE=4, NSPIMX=15, NINTMX=1)

C     CHEMICAL RATE DATA
C     MAX NO OF THIRD BODIES
      INTEGER NBDYMX
      PARAMETER(NBDYMX=10)
C     MAX SIZE OF STEP SPECIES-LIST, STEP REACTANT-LIST
      INTEGER NSSMAX,NRSMAX
      PARAMETER(NSSMAX=10, NRSMAX=10)
C     MAX NO OF LINDEMANN STEPS
      INTEGER NLLMAX
      PARAMETER(NLLMAX=10)

C     TRANSPORT COEFFICIENTS
      DOUBLE PRECISION ALAMDC,RLAMDA,TLAMDA
      PARAMETER(ALAMDC=2.58D-5, RLAMDA=7.0D-1, TLAMDA=2.98D2)
      DOUBLE PRECISION PRANTL
      PARAMETER(PRANTL=7.0D-1)

C     UNIVERSAL GAS CONSTANT
      DOUBLE PRECISION RGUNIV
      PARAMETER(RGUNIV=8.3142D3)
      .
      .
      .
```

The parameter NSPCMX sets the maximum number of chemical species while NSTPMX sets the maximum number of steps in the reaction mechanism. The minimum value for each is 1, and it is expected that both of these parameters will be set equal to the corresponding values in the chemical data file (see below). This is checked automatically during startup. A legitimate exception occurs for non-reacting flow, when NSTPMX=1 and the number of reaction steps is actually equal to zero.

The parameters NTINMX, NCOFMX and NCTMAX control the maximum sizes required to store the thermodynamic data for each species expressed in polynomial form (eq. 17). The parameter NTINMX sets the maximum number of temperature intervals required for any species while NCOFMX sets the maximum number of coefficients required in each interval. The parameter NCTMAX sets the maximum number of coefficients required in the polynomial for temperature (eq. 30).

The index of the current temperature interval for each species is stored in SENGA2 in order to avoid the need for repeated searching and a bit-wise compression algorithm is used to conserve memory. The parameter NTBASE defines the number base for the compression algorithm and must be set to an integer value that is a power of two and is greater than or equal to the maximum number of temperature intervals per species NTINMX. The parameter NSPIMX defines the maximum number of species temperature indices that can be stored in a single 32-bit signed integer using the number base NTBASE and its value must be set equal to NSPIMX = 31 DIV LOG2(NTBASE). The parameter NINTMX defines the number of integers required per spatial point. Clearly the product NSPIMX*NINTMX must be greater than or equal to the maximum number of species NSPCMX.

Size parameters for the chemical rate data must also be set. The parameter NBDYMX must be set greater than or equal to the maximum number of distinct third bodies. The parameter NSSMAX must be set greater than or equal to the maximum number of species involved in any single reaction step. Similarly, the parameter NRSMAX must be set greater than or equal to the maximum number of reactant species involved in any single reaction step. The parameter NLLMAX must be set greater than or equal to the maximum number of reaction steps requiring Lindemann rate data. Note that these parameters are used only to set array sizes, and actual values for these quantities are set in the chemical data file.

The values of the parameters used to evaluate the transport coefficients according to (eq. 31) are set. Clearly ALAMDA and RLAMDA represent the multiplicative coefficient and temperature exponent respectively while TLAMDA is the reference temperature. The mixture Prandtl number in (eq. 32) is set using the parameter PRANTL.

Finally the value of the universal gas constant is set (in J/kmol K) using the parameter RGUNIV.

When the editing of com_senga2.h is complete, all FORTRAN source files for SENGA2 must be recompiled in order to incorporate the changes into the code.

4.2 Control Data

The control data file cont.dat must be edited to set up the control parameters for the run. The file format is as shown:

```
***********************************************
**                                           **
**      SENGA2: Run control data file        **
**                                           **
***********************************************
```

```
Global domain size (x,y,z) in metres
 1.0D-2   1.0D-2   1.0D-2

Global domain size (nx,ny,nz)
 64    64     64

No. of processors (x,y,z)
 2      2      2

Time step; start step, no of steps, step switch (0=fixed, 1=adaptive)
 1.0D-12    1   1000   1

Intervals between dumps, reports, statistics
 500    1     500

Cold start switch (0=cold start, 1=restart)
 0

Initial turbulence generator
switch (0=off, 1=new, 2=inlet); random seed; spectrum parameters
 1      -1      8.51064D0   5.0D0  0.0D0  0.0D0

Flame generator switch (0=off, 1=on)
 0

Default initial conditions
pressure, temperature, velocity components u,v,w
 1.0D5   3.0D2   3.9D-1  0.0D0  0.0D0
mass fractions
 1
 1    1.0D0

Global boundary condition types
one per line: x-left; x-right; y-left; y-right; z-left; z-right
(1=periodic; 1a=inlet; 2b=outlet; 3c=wall; a,b,c denotes BC subtype)
(four integer and four real parameters allowed for each)
13    1    0    0   0   3.9D-1    0.0D0    0.0D0     0.0D0
21    0    0    0   0   1.0D5     2.87D0   1.0D-3    0.0D0
1     0    0    0   0   0.0D0     0.0D0    0.0D0     0.0D0
1     0    0    0   0   0.0D0     0.0D0    0.0D0     0.0D0
1     0    0    0   0   0.0D0     0.0D0    0.0D0     0.0D0
1     0    0    0   0   0.0D0     0.0D0    0.0D0     0.0D0

End of file
```

The first five lines are treated as a header and are read but ignored by the `SENGA2` code. The various data items are then listed in groups, and the format of the control file is fixed to the extent that the number of lines in each group and the number of data items per line must be preserved.

The global size of the domain in the x, y and z directions must be specified in metres, together with the global size of the domain in terms of the number of grid points in each direction and the number of processors in each direction. The values for global grid size and number of processors must be set to match those already specified in `com_senga2.h`, and this is checked by `SENGA2` during initialisation. Note that the processors are ranked from 0 such that the rank index increases by counting along the x, y and z directions in that order.

The initial time step must be set (in seconds), together with the index of the first time step and the required number of time steps for the run. A switch must be set to control whether a fixed time step (switch=`0`) or adaptive time-stepping (switch=`1`) is required. The number of time steps between dumps must be set. A restart file is written for each processor at the start of a run, and a further restart file is dumped for each processor after the specified number of time steps. Throughout the remainder of the run. In order to help guard against loss of data, two restart files are kept for each processor and the older file of the two is overwritten at each dump. The restart file names conform to the pattern `dmpiXXXY.dat` where tt XXX is the rank index of the corresponding processor (starting from `000`) and `Y` is the restart file index which is either `0` or `1`. The latest dump may be contained in either restart file and alternates between the two. The number of time steps between updates to the report file and the statistics file must also be set. A single report file and a single statistics file are each created at the start of the run by the lowest-ranked processor. Each is updated after the specified number of time steps.

Initial conditions for the run must be set. The cold start switch must be set to `0` for a cold start, in which the run is initialised from scratch using specified initial data. Alternatively, the cold start switch must be set to 1 for a restart, using data from a previous dump. A restart file must exist for each processor, and the restart file names must conform to the pattern `dmpiXXXY.dat` where tt XXX is the rank index of the corresponding processor and `Y` is the restart file index. It is expected that `Y` will be set to `0` for a restart.

The initial turbulence generator switch must be set to `0` if no initial turbulent field is to be generated, or set to `1` if a fresh initial turbulent field is to be generated. If the switch is set to `1` then the random seed must also be set in order to initialise the random number generator. If the (integer) value of the random seed is set to be non-negative, the value set is added to the rank of each processor in order to produce a different random seed for each processor. This ensures that there is no repetition of the same random sequence on each processor. Conversely, if the value of the random seed is set to be a negative integer, the value is used globally in order to ensure that the same global initial turbulent field is generated for a given global grid size irrespective of the number of processors in use. Up to four parameters may be set in order to control the initial turbulence spectrum function defined in subroutine `ESPECT` within `SENGA2`. The default spectrum function is the Batchelor–Townsend spectrum [15] which is described in section 2.6.1 and which requires the use of only the first two parameters: these control the total amount of turbulence kinetic energy and the wavenumber at which the spectrum function has its peak. If the initial turbulence generator switch is set to `2`, then a previously–generated field of frozen inlet turbulence is copied into the domain. This facility exists to ensure continuity of the turbulent field across the inlet boundary in cases where a turbulent inflow is specified.

The initial flame generator switch must be set to `1` if an initial scalar field is to be specified, or set to `0` if not. This switch controls whether or not the subroutine `FLAMIN` is called within `SENGA2`. This subroutine is designed to allow for the specification of an initial scalar field of arbitrary complexity.

Default initial conditions must be set for each variable. Pressure, temperature and the three velocity components must be set using SI units. The total number of species must be set and must correspond to the number of species specified in `com_senga2.h`. This is checked by `SENGA2` during startup. For each species an index number and a value for its initial mass fraction must be set. Each index number must correspond to the species number specified in the chemical data file `chem.dat` as described below. The total of all mass fractions must be equal to unity, and this is checked by `SENGA2` during startup.

Boundary condition types must be specified for the global domain. This is done using an indexing system, and up to four real and four integer parameters may be set for each boundary condition subtype.

Periodic boundary conditions are specified using an index value of `1`. Both of the corresponding periodic faces of the domain must have this index value and this is checked during startup.

Inlet boundaries are specified using an index value of `1a`, where `a` denotes the inlet boundary condition subtype. Three inlet boundary condition subtypes are implemented:

`11` Subsonic non-relecting laminar inflow
This condition requires no additional parameters to be set.

`12` Subsonic reflecting turbulent inflow with specified temperature
By default the inlet temperature is set to the same value as specified for the default initial temperature.

`13` Subsonic reflecting turbulent inflow with specified density
By default the inlet density is set to the same value as the initial density as computed from the values specified for the default initial pressure, temperature and species mass fractions.

For inlet boundary condition subtypes `12` and `13`, the value of the first integer parameter determines the nature of the inlet velocity field. A value of `1` specifies a laminar inflow with constant velocity, whereupon the first real parameter is taken to specify the inflow velocity component normal to the boundary. A value of `2` specifies a laminar inflow with velocity varying sinusoidally in time, whereupon the first two real parameters specify the amplitude and period of the oscillation. A value of `3` specifies a turbulent inflow. In this case, the second integer parameter must be set to `0` for a cold start of the turbulent inlet velocity field, or to `1` for a restart. The turbulent inflow velocity field is specified using Fourier interpolation onto a scanning plane passing through a stored cubic box of precomputed frozen turbulence, as described above. For a cold start, the first three real parameters must be set. In order, these specify the inlet mean velocity, the difference between the scanning plane velocity and the inlet mean velocity, and the initial scanning plane location as a distance (in metres) from the downstream end of the stored box. An input file is required for each processor. The cold start input filename has the form `tcxlXXX.dat` where `XXX` is the rank index of the corresponding processor. The file has the same format as a `SENGA2` dump file, i.e a `FORTRAN` unformatted file containing all of the variables required for a full restart of the code. For an inlet cold start only the velocity field data is extracted from the file. The intention is that the inlet velocity field would be generated from a previous run of `SENGA2` with appropriate initial conditions and possibly using periodic boundary conditions. For a restart, the input filename has the form `tixlXXX.dat` where `XXX` is the rank index of the corresponding processor. This file is generated by `SENGA2` during an inlet cold start and subsequently at the same time as each scheduled dump. It is a `FORTRAN` unformatted file containing the Fourier coefficients of the three velocity components together with the current values of the scanning plane location, the scanning velocity and the inlet mean velocity.

Outlet boundaries are specified using an index of `2b`, where `b` denotes the outlet boundary condition subtype.

`21` Subsonic non–reflecting outflow
This condition requires three real parameters to be set. In order, these specify the pressure at infinity, the relaxation parameter and the nominal boundary Mach number.

Wall boundaries are specified using an index of `3c`, where `c` denotes the wall boundary condition subtype. The list of boundary condition types as implemented in the code is given below:

`31` Adiabatic no-slip wall
This condition requires no additional parameters to be set.

`32` Isothermal no-slip wall
This condition requires a single real parameter to be set in order to specify the wall temperature.

This completes the description of the control data file.

4.3 Chemical Data

The chemical data file `chem.dat` must be set up to contain the thermodynamic and chemical data required for the run. The file may be edited directly or it may be generated using the chemical preprocessor `PPCHEM` (see below). The file format is as shown:

```
****************************
*                          *
*  Output file from ppchem  *
*                          *
****************************
Species list:
   8
   1   CH4
   2   O2
   3   CO2
   4   H2O
   5   H2
   6   H
   7   CO
   8   N2
Species data:
 1.0000E+05
   1  1.6000E+01
 9.7000E-01
   2
 3.0000E+02  1.0000E+03    7
 7.7874150E-01
 1.7476680E-02
-2.7834090E-05
 3.0497080E-08
-1.2239307E-11
-9.8252290E+03
 1.3722195E+01
 1.0000E+03  5.0000E+03    7
 1.6834780E+00
 1.0237236E-02
-3.8751280E-06
 6.7855850E-10
-4.5034230E-14
-1.0080787E+04
 9.6233950E+00
   .
   .
   .
   8  2.8000E+01
 1.0000E+00
   2
 3.0000E+02  1.0000E+03    7
 3.2986770E+00
 1.4082404E-03
-3.9632220E-06
 5.6415150E-09
```

```
-2.4448540E-12
-1.0208999E+03
 3.9503720E+00
 1.0000E+03  5.0000E+03    7
 2.9266400E+00
 1.4879768E-03
-5.6847600E-07
 1.0097038E-10
-6.7533510E-15
-9.2279770E+02
 5.9805280E+00
Step rate data:
   6
   1  2.2000E-05  3.0000E+00  3.6600E+07
   2  2.0400E+04  1.5000E+00  9.0230E+07
   3  2.3000E+12 -8.0000E-01  0.0000E+00
   4  2.0000E+05  0.0000E+00  7.0300E+07
   5  5.8300E+05  1.5000E+00  1.3474E+08
   6  1.5000E+00  0.0000E+00  1.4505E+08
Step species-list:
   1    5
   1    1       1
   1    2       4
   1    3       5
   1    4       6
   1    5       7
   .
   .
   .
   6    4
   6    1       2
   6    2       4
   6    3       5
   6    4       6
Step reactant-list:
   1    4
   1    1       1
   1    2       4
   1    3       6
   1    4       6
   .
   .
   .
   6    4
   6    1       4
   6    2       4
   6    3       6
   6    4       6
Step product-list:
   1    5
   1    1       5
   1    2       5
   1    3       5
   1    4       5
   1    5       7
   .
   .
   .
   6    4
   6    1       2
   6    2       5
   6    3       5
   6    4       5
Step reactant coefficient-list:
   1    0
   2    0
   3    0
   4    0
   5    0
   6    0
Step product coefficient-list:
   1    0
   2    0
   3    0
   4    0
   5    0
   6    0
Species delta-list:
   1    1 -1.0
   1    2 -1.0
   1    3  4.0
   1    4 -2.0
   1    5  1.0
   .
   .
   .
   6    1  1.0
   6    2 -2.0
   6    3  3.0
   6    4 -2.0
Third-body list:
   1
   1   M
Third-body step-list:
   1       0
   2       0
   3       1
   4       0
   5       0
   6       0
Third-body efficiencies:
   1    1  1.0000E+00
   1    2  1.0000E+00
   1    3  1.5000E+00
   1    4  6.5000E+00
   1    5  1.0000E+00
   1    6  1.0000E+00
   1    7  1.0000E+00
   1    8  4.0000E-01
 Gibbs step-list:
   0
 Lindemann step-list:
   0
 Lindemann step rate data:
 Troe step-list:
   0
 Troe step rate data:
 SRI step-list:
   0
 SRI step rate data:
 End of file
```

The example shown is an abbreviated version of the chemical data file for the Peters-Williams 4-step methane oxidation mechanism [21]. In most cases it is expected that the chemical data file will be generated using the chemical data preprocessor PPCHEM described below, although some manual editing may be required also.

The first five lines are treated as a header and are read but ignored by the SENGA2 code. The various data items are then listed in groups, where the format of each group is fixed and is determined by the nature of the data involved. Note that not all groups are required for all reaction mechanisms, and in this case a one-line header appears as a placeholder for the group.

The *species list* associates an integer identifier with each species involved in the reaction mechanism. After the one-line header, the first line contains an integer N specifying the number of species in the list and hence the length of the list. Each subsequent line contains an integer identifier α and a string representing the species chemical symbol $\mathcal{M}_\alpha$. Each integer identifier must be unique, and each species string must contain no spaces. The default maximum length of a species string is 10 characters.

The *species data* group contains the thermodynamic data required for each species. Following the one-line header the first line specifies the reference pressure in pascals (Pa). There follows a series of data blocks with one block per species. For each block the first line contains the species integer identifier and the molar mass of the species in kg/kmol, and the second line specifies the Lewis number for the species. The next line contains an integer which specifies the number of temperature ranges over which the thermodynamic data is defined. The data for each temperature range is specified in a sub-block whose first line contains two real numbers specifying the lower and upper limits of temperature in degrees Kelvin (K) and an integer specifying the number of temperature coefficients for that range. The remainder of the sub-block contains the coefficients arranged one per line. There are as many sub-blocks as there are temperature ranges for each species, and as many data blocks as there are species in the species list.

The *step rate data* group contains the Arrhenius rate parameters for each forward step in the reaction mechanism. The group consists of a one-line header followed by a line containing an integer M specifying the number of steps in the reaction mechanism. This is followed in turn by one line for each step containing a unique integer identifier m for the step followed by three real numbers specifying the Arrhenius rate parameters A_m, n_m and E_m, in accordance with eq. 14. For Lindemann, Troe and SRI forms these values correspond to the rate coefficient $k_{\infty.m}$ in the high–pressure limit.

The *step species list* provides a compact list of the species involved in each step of the reaction mechanism, i.e. a species for which either $\nu'_{\alpha,m}$ or $\nu''_{\alpha,m}$ (or possibly both) takes a non-zero value. Note that third bodies are not included in this list. The group consists of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each block begins with a line containing two integer values, the first being the value of m that uniquely identifies the step and the second specifing the number of species involved in that step. Subsequent lines in each block contain three integer values of which the first is a repeat of the step identifier m, the second is an index number applicable only within the current step, and the third identifies a species using the unique integer identifier α as previously defined in the species list. There are as many lines in each block as there are species taking part in the step, and there are as many blocks as there are steps in the mechanism.

The *step reactant list* provides a compact list of the species involved in each step of the mechanism as molecular reactant species, i.e. a species for which $\nu'_{\alpha,m}$ takes an integer value greater than zero. Note that third bodies are not included in the list. The group has the same structure as the step species list, consisting of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each block begins with a line containing two integer values, the first being the value of m that uniquely identifies the step and the second specifing the number of entries in the list for that step. Subsequent lines in each block contain three integer values of which the first is a repeat of the step identifier m, the second is an index number applicable only within the current step, and the third uses the unique species identifier α to identify a reactant species within the current step. If $\nu'_{\alpha,m}$ has an integer value greater than unity then the entry for species α is repeated to appear a total of $\nu'_{\alpha,m}$ times, each with a unique index number but with the same species identifer. There are as many lines in each block as required for the number of species for each step including repeats, and there are as many blocks as there are steps in the mechanism.

The *step product list* provides a compact list of the species involved in each step of the mechanism as molecular product species, i.e. a species for which $\nu''_{\alpha,m}$ takes an integer value greater than zero. Note that third bodies are not included. The group has the same structure as the step species list, consisting of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each block begins with a line containing two integer values, the first being the value of m that uniquely identifies the step and the second specifing the number of entries in the list for that step. Subsequent lines in each block contain three integer values of which the first is a repeat of the step identifier m, the second is an index number applicable only within the current step, and the third uses the unique species identifier α to identify a product species within the current step. If $\nu''_{\alpha,m}$ has an integer value greater than unity then the entry for species α is repeated to appear a total of $\nu''_{\alpha,m}$ times each with a unique index number but with the same species identifer. There are as many lines in each block as required for the number of species in each step including repeats, and there are as many blocks as there are steps in the mechanism.

The *step reactant coefficient list* provides a list of the reactant species for which the value of $\nu'_{\alpha,m}$ is non-zero but is not a positive integer. Note that third bodies are not included. The group has the same structure as the step species list, consisting of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each block begins with a line containing two integer values, the first being the value of m that uniquely identifies the step and the second specifing the number of entries in the list for that step. Subsequent lines in each block contain three integer values and one real value. The first integer is a repeat of the step identifier m, the second is an index number applicable only within the current step, and the third uses the unique species identifier α to identify a reactant species within the current step. The real value is the value of $\nu'_{\alpha,m}$ for that species within the step. There are as many lines in each block as required to specify all of the non-zero non-positive-integer values of $\nu'_{\alpha,m}$ for that step, and there are as many blocks as there are steps in the mechanism.

The *step product coefficient list* provides a list of the product species for which the value of $\nu''_{\alpha,m}$ is non-zero but is not a positive integer. Note that third bodies are not included. The group has the same structure as the step species list, consisting of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each block begins with a line containing two integer values, the first being the value of m that uniquely identifies the step and the second specifing the number of entries in the list for that step. Subsequent lines in each block contain three integer values and one real value. The first integer is a repeat of the step identifier m, the second is an index number applicable only within the current step, and the third uses the unique species identifier α to identify a product species within the current step. The real value is the value of $\nu''_{\alpha,m}$ for that species within the step. There are as many lines in each block as required to specify all of the non-zero non-positive-integer values of $\nu''_{\alpha,m}$ for that step, and there are as many blocks as there are steps in the mechanism.

The *step species delta-list* contains the value of the difference of stoichiometric coefficients $\left(\nu''_{\alpha,m} - \nu'_{\alpha,m}\right)$ for each species in each step in the mechanism. Note that third bodies are not included. The group consists of a one-line header followed by a number of data blocks, one for each step in the mechanism. Each line in each block contains two integer values and one real value. The first integer is a repeat of the step identifier m while the second is an index number applicable only within the current step. The real value is the value of $\left(\nu''_{\alpha,m} - \nu'_{\alpha,m}\right)$ for that species within the step. There are as many lines in each block as required for the number of species involved in the step, and there are as many blocks as there are steps in the mechanism.

The *third body list* associates an integer identifier with each third body involved in the reaction mechanism. After the one-line header, the first line contains an integer specifying the number of distinct third bodies required by the mechanism and hence the length of the list. Each subsequent line contains an integer identifier and a string representing the generic chemical symbol for the corresponding third body. Each third-body integer identifier must be unique, and each third-body string must contain no spaces. The default maximum length of a third-body string is 10 characters.

The *third-body step-list* indicates which steps in the reaction mechanism involve a third body. After the one-line header, each line contains two integer values. The first integer corresponds to the step number m. If a step does not involve a third body then the second integer value is set equal to zero. If a step does involve a third body then the second integer value is the third body identifier for that step, as already defined in the third body list.

The *third-body efficiencies* are listed for each third body identified in the third body list. For every third body there is a data block consisting of a number of lines equal to the number of species N. Each line contains two integer values and a real value. The first integer is the unique identifier for the third body as previously specified in the third body list. The second integer is the species number α as previously specified in the species list, and the real value is the value of the third body efficiency $\eta_{\alpha,M}$ as required by eq. 54.

The *Gibbs step–list* indicates which steps in the mechanism need to have the backward reaction rate evaluated using the Gibbs function according to eq. 51. The first line contains a single integer value specifying the number of Gibbs steps. If this number is non-zero, there follows a number of lines equal to the number of steps in the reaction mechanism. Each line contains two integer values, of which the first is the step index number m as previously specified in the step species list. If the step requires evaluation of the reaction rate using the Gibbs function then the second integer value is equal to m, otherwise it is equal to zero.

The *Lindemann step–list* indicates which steps in the mechanism need to have the reaction rate evaluated using a Lindemann form according to eq. 55. The first line contains a single integer value specifying the number of Lindemann steps. If this number is non-zero, there follows a number of lines equal to the number of steps in the reaction mechanism. Each line contains two integer values, of which the first is the step index number m as previously specified in the step species list. If the step requires evaluation of the reaction rate using a Lindemann form then the second integer value specifies a unique integer identifier for the Lindemann step, otherwise it is equal to zero.

The *Lindemann step rate data* is listed for each Lindemann step. Each entry consists of an integer identifier for the Lindemann step together with the real–number values of the four Lindemann parameters $A_{0,m}$, $n_{0,m}$, $E_{0,m}$ and F_m, as described in section 2.4.4. The length of the list is equal to the number of Lindemann steps as already specified in the Lindemann step list.

The *Troe step–list* indicates which steps in the mechanism are treated using a Troe form (see eq. 57). The first line contains a single integer value specifying the number of Troe steps. If this number is non-zero, there follows a number of lines equal to the number of steps in the reaction mechanism. Each line contains two integer values, of which the first is the step index number m. If the step is a Troe step then the second integer value specifies a unique integer identifier for the Troe step, otherwise it is equal to zero.

The *Troe step rate data* is listed for each Troe step. Each entry consists of a pair of lines, each starting with an integer identifier for the Troe step. The first line also contains the real–number values of the six parameters $A_{0,m}$, $n_{0,m}$, $E_{0,m}$, α, T^* and T^{**} while the second line also contains the six parameters T^{***}, c_1, c_2, n_1, n_2 and d as listed in section 2.4.5. The length of the list is equal to the number of Troe steps.

The *SRI step–list* indicates which steps in the mechanism are treated using the SRI form (see eq. 60). The first line contains a single integer value specifying the number of SRI steps. If this number is non-zero, there follows a number of lines equal to the number of steps in the reaction mechanism. Each line contains two integer values, of which the first is the step index number m. For each SRI step the second integer value specifies a unique integer identifier for the SRI step, otherwise it is equal to zero.

The *SRI step rate data* is listed for each SRI step. Each entry consists of a pair of lines, each starting with an integer identifier for the SRI step. The first line also contains the real–number values of the four parameters $A_{0,m}$, $n_{0,m}$, $E_{0,m}$ and a while the second line also contains the four parameters b, c, d and e as listed in section 2.4.6. The length of the list is equal to the number of SRI steps.

5 Chemistry Pre-Processor PPCHEM

The chemical data file is most easily constructed using a chemistry pre–processor program named PPCHEM. This program requires two input files: a reaction mechanism file and a thermochemical data file. At start-up, PPCHEM will prompt for the names of the reaction mechanism file and the thermochemical data file to be read, and for the name of the chemical data output file to be written. If required, a simple control file containing these filenames can be constructed and preconnected to the standard input. By default, reaction mechanism filenames have the extension .mec, thermochemical data filenames have the extension .thr and chemical data output filenames have the extension .chm. The chemical data output file from PPCHEM is in the correct format to be read by SENGA2 using the fixed filename chem.dat.
The reaction mechanism file has the following format:

```
#
# Peters-Williams 4-step methane mechanism
#
# From Peters, N. and Williams, F.A.: Combust.Flame 68, 185-207, 1987.
#
Species list
1    CH4    methane
2    O2     oxygen
3    CO2    carbon dioxide
4    H2O    water vapour
5    H2     hydrogen
6    H      hydrogen atom
7    CO     carbon monoxide
8    N2     nitrogen
END
Third body list
1     M
CO2   1.5
H2O   6.5
N2    0.4
END
Mechanism step list
1    CH4    +2H    +H2O     =>CO     +4H2       2.200E+04  3.0       36.60
2    CO     +H2O            =>CO2    +H2        2.040E+07  1.5       90.23
3    2H     +M              =>H2     +M         2.300E+18 -0.8        0.00
4    O2     +3H2            =>2H2O   +2H        2.000E+14  0.0       70.30
5    CO2    +H2             =>CO     +H2O       5.830E+08  1.5      134.74
6    2H2O   +2H             =>O2     +3H2       1.500E+09  0.0      145.05
END
Conversion factor list (length, kmols, time, temp, energy)
1.0D-2   1.0D-3   1.0D0   1.0D0   1.0D3
END
End of file
```

There is a five-line header which is ignored by PPCHEM and which is intended solely for labelling and comments. This is followed by the species list, the third-body list and the mechanism step list. Each list begins with a one-line header and is terminated by a line containing the keyword "END" only.

Each line of the species list corresponds to a single species, and contains the species number, the species symbol and the species name. Each species number must be unique and consecutive, starting at 1. The species symbol has a maximum length of 16 characters, must not contain any spaces, and must not end with "D" or "E". The species name has a maximum length of 50 characters, and can contain any alphanumeric character including spaces.

The third body list consists of a series of data blocks. Each data block begins with a line containing the third body number and the third body symbol. The number must be consecutive and unique, and the symbol must be unique and follow the same rules as for species symbols. The remainder of each data block consists of a list, with each line containing a species symbol as defined in the species list together with a real number representing the third body efficiency for that species for that third body. There are as many data blocks as there are distinct third bodies, and if the reaction mechanism requires no third body there will be no data blocks.

Each line of the mechanism step list corresponds to one step in the chemical reaction mechanism. The line begins with a step number which must be unique and consecutive, starting at 1. There follows the list of reactant species symbols for the step, separated by "+" signs and terminated by the symbol "==" or "=>". This is followed by the list of product species symbols for the step separated by "+" signs. Spaces within the reactant and product species lists are optional and are ignored. All species symbols must correspond to species symbols or third body symbols as previously declared. The line ends with three real numbers separated by spaces, and giving the values of A, n and E for the step.

Reaction steps having a Lindemann form are indicated by the symbol "L" at the end of the line. This line contains the values of A, n and E for the high–pressure rate coefficient k_∞. For a Lindemann step the following line contains four real numbers separated by spaces and giving the value of F followed by the values of A, n and E for the low–pressure rate coefficient k_0.

Reaction steps of Troe form are indicated by the symbol "T" at the end of the line. This line contains the values of A, n and E for the high–pressure rate coefficient k_∞. For a Troe step the following line contains seven real numbers separated by spaces, giving the values of α, T^*, T^{**} and T^{***} followed by the values of A, n and E for the low–pressure rate coefficient k_0. This is followed by a further line containing five real numbers separated by spaces and giving the values of the constants c_1, c_2, n_1, n_2 and d.

Reaction steps of SRI form are indicated by the symbol "S" at the end of the line. This line contains the values of A, n and E for the high–pressure rate coefficient k_∞. For each SRI step the following line contains three real numbers separated by spaces and giving the values of A, n and E for the low–pressure rate coefficient k_0. This is followed by a further line containing five real numbers separated by spaces and giving the values of the constants a, b, c, d and e.

The format for Lindemann, Troe and SRI steps is illustrated by the following fragment of a (fictitious) mechanism file:

```
21   H2O2   +M1             ==OH     +OH    +M1 1.200E+17  0.0    45500.0 L
21                                          0.5 2.950E+14  0.0    48400.0
22   H2O2   +OH    +M1      ==H2O    +HO2   +M1 5.800E+14  0.0     9560.0 T
22   0.5          300.0       400.0       500.0 2.950E+14  0.0    48400.0
22                                             -0.4 -0.67 0.75 -1.27 0.14
23   H2O2   +OH    +M1      ==H2O    +HO2   +M1 5.800E+14  0.0     9560.0 S
23                                              2.950E+14  0.0    48400.0
23                                              0.9  0.63  0.85  1.0  1.0
```

Following the end of mechanism step list, there is a conversion factor list. This list specifies the factors required to convert the values of A, n and E as given in the mechanism step list into SI units, i.e. using metres, kmols, seconds, degrees Kelvin and Joules. This allows for the fact that reaction mechanism parameters are often specified using non–SI units, In principle, any units may be used within the mechanism step list, provided only that the same units are used for every step in the mechanism. The conversion factor list consists of a header line (which is ignored) followed by a line containing five real numbers separated by spacess and giving the values of the conversion factors for length, amount of substance, time, temperature and energy. The file is terminated by a line containing the phrase "`End of file`".

5.1 Thermochemical Data

The thermochemical data file has the following format:

```
****************************
*                          *
*  Output file from ppthrm  *
*                          *
****************************
 1.0000E+05
   CH4
     1.6000E+01
     9.7000E-01
   2
   1  300.00 1000.00  7
    7.7874150E-01
    1.7476680E-02
   -2.7834090E-05
    3.0497080E-08
   -1.2239307E-11
   -9.8252290E+03
    1.3722195E+01
   2 1000.00 5000.00  7
    1.6834780E+00
    1.0237236E-02
   -3.8751280E-06
    6.7855850E-10
   -4.5034230E-14
   -1.0080787E+04
    9.6233950E+00
    END
   CO
     2.8000E+01
     1.1000E+00
   2
   1  300.00 1000.00  7
    3.2624510E+00
    1.5119409E-03
   -3.8817550E-06
    5.5819440E-09
   -2.4749510E-12
   -1.4310539E+04
    4.8488970E+00
   2 1000.00 5000.00  7
    3.0250780E+00
    1.4426885E-03
   -5.6308270E-07
    1.0185813E-10
   -6.9109510E-15
   -1.4268350E+04
    6.1082170E+00
    END
END
```

There is a five-line header which is ignored. This is followed by a value specifying the reference pressure for the thermodynamic data. There follows a series of data blocks. There must be exactly one data block for each species as declared in the mechanism file, but there may be more species and hence more data blocks than are required by the mechanism. Each data block begins with a line containing only a species symbol, constructed according to the rules for species symbols. The two subsequent lines each contain a single real number corresponding respectively to the molar mass and the Lewis number for that species. The next line contains a single integer corresponding to the number of temperature intervals for which thermodynamic data is provided for the species in question. There follows a number of data sub-blocks, one for each temperature interval. Each sub-block has a first line containing an integer, two real numbers and a second integer. The first integer gives the number of the sub-block, the two real numbers provide the lower and upper limits of the temperature range for this interval, and the second integer gives the number of coefficients to follow. The sub-block is completed by that number of lines, each containing a real number corresponding to a thermodynamic data item. The data block is terminated by a line containing the keyword "END", and the file is terminated by an extra line containing the keyword "END".

5.2 Thermochemical Pre-Processor PPTHRM

The thermochemical data file may be constructed either manually or by computer using data from any preferred source. A particularly useful data source is the CHEMKIN database. A further preprocessing program is PPTHRM which reads a CHEMKIN format database file together with a species data file in order to generate a thermochemical data file for PPCHEM. At startup, PPTHRM will prompt for the names of the species data file and the CHEMKIN format thermodynamic database file, for the value of the reference pressure used in the construction of the thermodynamic data, and for the name of the thermochemical data file to be output. Alternatively, this information can be provided in a control file preconnected to standard input. The default filename extension for the species data file is .spc, for the thermodynamic database file it is .src, and for the thermochemical data file it is .thr as previously indicated.

The CHEMKIN format of the database file is given in relevant publications. The format of the species data file is:

```
#
# Peters-Williams 4-step methane mechanism
# Molar masses and Lewis numbers
# From Peters, N. and Williams, F.A.: Combust.Flame 68, 185-207, 1987.
#
Species list
1    CH4       16.0        0.97
2    O2        32.0        1.11
3    CO2       44.0       1.39
4    H2O       18.0        0.83
5    H2         2.0        0.30
6    H          1.0        0.18
7    CO        28.0        1.10
8    N2        28.0        1.00
END
End of file
```

As before, the first five lines are ignored. The rules for the file format and for the format of the species list are the same as for the species list in the mechanism file, except that each line contains two real numbers in place of the species name. These numbers correspond to the molar mass and the Lewis number respectively.

References

[1] K.W. Jenkins, R.S. Cant: Direct Numerical Simulation of turbulent flame kernels, in “Recent Advances in DNS and LES”, eds. D. Knight and L. Sakell, pp191–202, Kluwer Academic, New York, 1999.

[2] T. Poinsot, S. Lele: Boundary conditions for direct simulations of compressible viscous flows, J. Comp. Phys, **101**, 104–129, 1992.

[3] J.C. Sutherland, C.A. Kennedy: Improved boundary conditions for viscous, reacting compressible flows, J. Comp. Phys. **191**, 502–524, 2003.

[4] S.K. Lele: Compact finite difference schemes with spectral–like resolution, J. Comp. Phys. **103**, 16–42, 1992.

[5] C.A. Kennedy, M.H. Carpenter, R.M. Lewis: Low–storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, Appl. Numer. Math. **35**, 177–219, 2000.

[6] C.A. Kennedy, M.H. Carpenter: Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. **44**, 139–181, 2003.

[7] R.J. Kee, F.M. Rupley, E. Meeks, J.A. Miller: CHEMKIN–III: A Fortran chemical kinetics package for the evaluation of gas–phase chemical and plasma kinetics, report SAND96-8216, Sandia National Laboratories, 1996.

[8] A. Ern, V. Giovangigli: *Multicomponent Transport Algorithms*, Lecture Notes in Physics **m24**, Springer–Verlag, 1994.

[9] F.A. Lindemann, Discussion on “The Radiation Theory of Chemical Action”, Trans. Faraday Soc. **17**, 598–606, 1922.

[10] R.G. Gilbert, K. Luther, J. Troe: Theory of thermal unimolecular reactions in the fall–off range. II. Weak collision rate constants, Ber. Bunsenges. Phys. Chem. **87**, 169–177, 1983.

[11] P.H. Stewart, C.W. Larson, D. Golden: Pressure and temperature dependence of reactions proceeding via a bound complex. 2. Application to $2CH_3 \rightarrow C_2H_5 + H$, Combust. Flame **75**, 25–32, 1989.

[12] M. Wille, Large Eddy Simulation of jets in cross–flows, PhD thesis, Imperial College, 1997.

[13] S.A. Orszag: Numerical methods for the simulation of turbulence, Phys. Fluids Suppl. II, 250–257, 1972.

[14] R.S. Rogallo: Numerical experiments in homogeneous turbulence, NASA Tech. Memo. 81315, 1981.

[15] G.K. Batchelor, A.A. Townsend: Decay of turbulence in the final period, Proc. Roy. Soc. Lond. **A194**, 527–543, 1948.

[16] G.K. Batchelor: *The Theory of Homogeneous Turbulence*, Cambridge University Press, 1953.

[17] R.S. Cant. Initial conditions for Direct Numerical Simulation of turbulence, Technical Report CUED/A–THERMO/TR66, Cambridge University Engineering Department, 2012.

[18] G.A. Blaisdell, N.N. Mansour, W.C. Reynolds: Compressiblity effects on the growth and structure of homogeneous turbulent shear flow, J. Fluid Mech. **256**, 443–485, 1993.

[19] C. Temperton: A generalised prime factor FFT algorithm for any $N = 2^p 3^q 5^r$, SIAM J. Sci. Stat. Comp. **13**, 676–686, 1992.

[20] R.S. Cant: Discrete Fourier transforms for turbulence, Technical Report CUED/A–THERMO/TR65, Cambridge University Engineering Department, 2012.

[21] N. Peters, F.A. Williams: The asymptotic structure of stoichiometric methane–air flames, Combust. Flame, **68**, 183–207, 1987.