Appendix A

Formulation

A.1 The governing equations for compressible reacting flows

The governing equations used for the simulation in this study have been derived before and used extensively (e.g., [Williams 1985] [Buckmaster & Ludford 1982]). However, for completeness, we will review those equations and the difficulties they pose to the numerical study.

First the compressible Navier-Stokes equations will be reviewed, and then several different sets of equations under different levels of assumptions will be derived.

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0, \tag{A.1a}
\]
\[
\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho uu - T) = \rho f, \tag{A.1b}
\]
\[
\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho eT - u \cdot T + q) = \rho f \cdot u \tag{A.1c}
\]
\[
\frac{\partial \rho Y_m}{\partial t} + \nabla \cdot (\rho Y_m (u + V_m)) = \dot{\rho}_m, \tag{A.1d}
\]

where \( q \) is heat flux given by

\[
q = -\lambda_T \nabla T + \rho \sum_m h_m Y_m V_m + R_u T \sum_m \sum_n \left( \frac{X_n D_{T,m}}{W_mD_{mn}} \right) (V_m - V_n) + q_R, \tag{A.2}
\]

where \( \lambda_T \) is the thermal conductivity, \( h_m \) is the specific enthalpy of species \( m \), \( V_m \) is the diffusion velocity for species \( m \), \( R_u \) is the universal gas constant, \( D_{T,m} \) is the thermal diffusion coefficient for species \( m \), \( W_m \) is molecular weight of species \( m \), \( D_{mn} \) is binary diffusion
The stress tensor $\mathbf{T}$ is given by

$$
\mathbf{T} = -p\mathbf{I} + \mu \left[ (\nabla \mathbf{u}) + (\nabla \mathbf{u})^T \right] + \lambda (\nabla \cdot \mathbf{u}) \mathbf{I}.
$$

(A.3)

$V_m$ may be approximated by (Curtiss & Bird, 1999; Bird et al., 2002),

$$
V_m = -\frac{D^T}{T} \nabla T - \sum_n D_{mn} d_n
$$

(A.4)

where $d_n$ is the diffusion driving force given by,

$$
d_n = \nabla X_n - (Y_m - X_m) \frac{\nabla p}{p}.
$$

(A.5)

The first term in Eqn. (A.4) is called Soret effect, which is an effect of thermal diffusion. The species diffusion velocity combines concentration diffusion and pressure diffusion effects, both of which are described in the diffusion driving force.

In these equations, and throughout the thesis, $\rho$ is reserved for density, $\mathbf{u}$ for the velocity vector, $e_T$ for the total energy, $Y_m$ for the mass fraction of species $m$, and $X_m$ for the mole fraction of species $m$.

For the transport properties, $\mu$ denotes the dynamic viscosity, $\lambda$ denotes the second coefficient of viscosity, which is equal to $\mu_B - 2/3\mu$, where $\mu_B$ is the bulk viscosity. $\lambda_T$ denotes the thermal conductivity, $D_{mn}$ denotes the binary diffusion coefficient between species $m$ and $n$, and $D_{T,m}$ denotes the thermal diffusion coefficient as already defined.

### A.1.1 Derivation of other forms of the energy equation

First, we derive several forms of the energy transport equation.

We start with the total energy conservation equation,

$$
\frac{\partial \rho e_T}{\partial t} + \nabla \cdot (\rho \mathbf{u} e_T - \mathbf{u} \cdot \mathbf{T} + \mathbf{q}) = \rho \mathbf{f} \cdot \mathbf{u}.
$$

(A.6)

We then use the momentum equation to obtain the transport equation for the internal energy,

$$
\rho \frac{\partial \mathbf{e}}{\partial t} = \mathbf{T} : \nabla \mathbf{u} - \nabla \cdot \mathbf{q}.
$$

(A.7)
We then use

\[ e = h - p/\rho \]  \hspace{1cm} (A.8)

to get the enthalpy transport equation,

\[ \rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \Sigma : \nabla \mathbf{u} - \nabla \cdot \mathbf{q}, \]  \hspace{1cm} (A.9)

where \( \Sigma = 2\mu \mathbf{S} + \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} \) is the viscous stress tensor. Eqns. (A.6), (A.7), and (A.9) are equivalent, and there are no additional assumptions in deriving one from another. The enthalpy of the mixture is given by,

\[ h = \sum_{m \in M} h_m Y_m, \]  \hspace{1cm} (A.10)

where,

\[ h_m(T) = h_{m,ref}^0 + \int_{T_{ref}}^T C_{p,m}(T')dT'. \]  \hspace{1cm} (A.11)

Eqn. (A.9) can be written in terms of the temperature evolution as follows:

\[ \rho C_{p,mix} \frac{DT}{Dt} = \frac{Dp}{Dt} + \Sigma : \nabla \mathbf{u} - \nabla \cdot \mathbf{q} + \sum_m \nabla \cdot (\rho h_m Y_m \mathbf{V}_m) - \sum_m \rho Y_m \mathbf{V}_m \cdot \nabla h_m - \sum_m h_m \rho \dot{\rho}_m. \]  \hspace{1cm} (A.12)

This form of the energy equation is used later to obtain the low Mach number limit of the energy equation.

A.2 The governing equations in the low Mach number limit

A.2.1 The momentum equation

Following Majda & Sethian (1985) and McMurtry et al. (1986), we derive the low Mach number limit of the Navier-Stokes equations with a special consideration to bulk viscosity. First we nondimensionalize the equations:

\[ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{uu} + p \mathbf{I}) - \mu [ (\nabla \mathbf{u}) + (\nabla \mathbf{u})^T ] - \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} = \rho \mathbf{f}. \]  \hspace{1cm} (A.13)

Here, \( \mu \) is the dynamic viscosity and \( \lambda \) is the second coefficient of viscosity, which is related to the bulk viscosity by \( \mu_B = \lambda + 2/3 \mu \).
We scale the primary variables: \( \hat{\rho} = \rho / \rho_0 \), \( \hat{u} = u / u_0 \), \( \hat{x} = x / x_0 \), \( \hat{t} = t / t_0 \) where \( t_0 = x_0 / u_0 \), \( \hat{p} = p / p_0 \), and \( \hat{f} = f / (u_0^2 / x_0) \). Furthermore, define \( c_0^2 = p_0 / \rho_0 \), \( \hat{\mu} = \mu / \mu_0 \), \( \hat{\lambda} = \lambda / \mu_0 \), \( M = u_0 / c_0 \) and \( Re = \rho_0 u_0 x_0 / \mu_0 \). The subscript 0 denotes the reference values of the corresponding variables which are constants. \( M \) denotes the Mach number (\( Ma \) is reserved for the Markstein number in this study.)

\[
\frac{\partial \hat{\rho} \hat{u}}{\partial \hat{t}} + \nabla \cdot \left( \hat{\rho} \hat{u} \hat{u} + \frac{1}{M^2} \hat{p} \hat{I} - \frac{1}{Re} \left( \hat{\mu} \right) \left( \nabla \hat{u} + (\nabla \hat{u})^T \right) - \hat{\lambda} (\nabla \cdot \hat{u}) \hat{I} \right) = \hat{\rho} \hat{f}.
\] (A.14)

When the external force is gravity, i.e., \( f = -g e_z \), the RHS becomes \( -\hat{\rho} / Fr \), where \( Fr \) is the Froude number.

Now we perform an asymptotic expansion for small Mach number. Let \( \epsilon \) be a small parameter:

\[
\begin{align*}
\hat{p}(\hat{x}, \hat{t}) &= p^0(\hat{x}, \hat{t}) + \epsilon \hat{p}^1(\hat{x}, \hat{t}) + O(\epsilon^2) \\
\hat{u}(\hat{x}, \hat{t}) &= u^0(\hat{x}, \hat{t}) + \epsilon \hat{u}^1(\hat{x}, \hat{t}) + O(\epsilon^2) \\
\hat{\rho}(\hat{x}, \hat{t}) &= \rho^0(\hat{x}, \hat{t}) + \epsilon \hat{\rho}^1(\hat{x}, \hat{t}) + O(\epsilon^2) \\
\hat{\mu}(\hat{x}, \hat{t}) &= \mu^0(\hat{x}, \hat{t}) + \epsilon \hat{\mu}^1(\hat{x}, \hat{t}) + O(\epsilon^2) \\
\hat{\lambda}(\hat{x}, \hat{t}) &= \lambda^0(\hat{x}, \hat{t}) + \epsilon \hat{\lambda}^1(\hat{x}, \hat{t}) + O(\epsilon^2)
\end{align*}
\] (A.15)

and substitute these into Eqn.(A.14). We obtain

\[
\frac{1}{M^2} \hat{\nabla} \left( \hat{p}^0 + \epsilon \hat{p}^1 + O(\epsilon^2) \right) = \\
- \frac{\partial \hat{\rho}^0 \hat{u}^0}{\partial \hat{t}} - \hat{\nabla} \cdot \left( \hat{\rho}^0 \hat{u}^0 \hat{u}^0 + \frac{1}{Re} \left( \hat{\mu}^0 \left[ (\hat{\nabla} \hat{u}^0) + (\hat{\nabla} \hat{u}^0)^T \right] - \hat{\lambda}^0 (\hat{\nabla} \cdot \hat{u}^0) \hat{I} \right) \right) + \hat{\rho}^0 \hat{f}^0 + O(\epsilon).
\] (A.16)

The leading order of the RHS are the \( O(1) \) terms. If \( \hat{p}^0 \) balances with this, \( M \sim 1 \) and we simply obtain the original compressible Navier-Stokes equation. If \( \hat{p}^1 \) balances with the RHS, \( M = O(\epsilon^{1/2}) \) and \( \nabla \hat{p}^0 = 0 \) must hold, which dictates that the leading order pressure term is uniform and a function only of time. For external flow, which is the subject of the current study, we assume that the ambient pressure is constant, thus \( \hat{p}^0 \) is constant and unity for the present study. However, for internal flows, such as reacting flows in an internal
combustion engine, this term can vary with time.

The \( O(\epsilon) \) equation leads to the low Mach number equations that are used in the present study,

\[
\nabla \hat{p}^1 = -\frac{\partial \hat{\rho}^0 \hat{u}^0}{\partial t} - \nabla \cdot \left( \hat{\rho}^0 \hat{u}^0 \hat{u}^0 \right) - \frac{1}{\text{Re}} \left( \hat{\rho}^0 \left[ \left( \nabla \hat{u}^0 + (\nabla \hat{u}^0)^T \right) - \hat{\lambda}^0 (\nabla \cdot \hat{u}^0) I \right] \right) + \hat{p}^0 \hat{T}^0. \tag{A.17}
\]

This equation is, in dimensional form,

\[
\frac{\partial \rho u}{\partial t} + \nabla \cdot \left( \rho uu + (p(x,t) - p_0) I - \mu \left[ \left( \nabla u \right) + \left( \nabla u \right)^T \right] - \lambda (\nabla \cdot u) I \right) = \rho f. \tag{A.18}
\]

Eqn. (A.18) can also be written as follows:

\[
\frac{\partial \rho u}{\partial t} + \nabla \cdot \left( \rho uu + p^* I - \mu \left[ \left( \nabla u \right) + \left( \nabla u \right)^T \right] \right) = \rho f, \tag{A.19}
\]

where

\[
p^* = (p(x,t) - p_0) - \left( \mu_B - \frac{2}{3} \mu \right) (\nabla \cdot u). \tag{A.20}
\]

The bulk viscosity is now combined with the perturbational part of the pressure field.

### A.2.2 The equation of state

The equation of state is

\[
p = \rho R_{\text{gas}} T. \tag{A.21}
\]

We normalize this equation as before and we obtain

\[
\hat{p} = \hat{\rho} \hat{T} \tag{A.22}
\]

where \( \hat{T} = T/T_0 \) and \( T_0 = p_0/\rho_0 R_{\text{gas}} \). Again, using the asymptotic expansion, we obtain the equation of state in the low Mach number limit.

\[
\hat{p}^0 = \hat{\rho}^0 \hat{T}^0. \tag{A.23}
\]

Note that \( \hat{T}^0 \) is defined as \( \hat{T}(\hat{x}, \hat{t}) = \hat{T}^0(\hat{x}, \hat{t}) + \epsilon \hat{T}^1(\hat{x}, \hat{t}) + O(\epsilon^2) \).

The LHS of the equation is constant, and thus this equation relates density and temperature, and they cannot vary independently.
Also note that the reference states satisfy the equation of state, \( p_0 = \rho_0 \bar{R}_\text{gas} T_0 \).

### A.2.3 The energy equation

Starting from the temperature transport equation,

\[
\rho C_{p,\text{mix}} \frac{DT}{Dt} = \frac{Dp}{Dt} + \sum m \nabla \cdot (\rho \dot{h}_m \dot{Y}_m \dot{V}_m) - \sum m \rho Y_m \dot{V}_m \cdot \nabla \dot{h}_m - \sum m \dot{h}_m \dot{\rho}_m. \tag{A.24}
\]

In a similar way to the momentum equations, we obtain the following leading-order equations.

\[
\hat{\rho} \hat{C}_{p,\text{mix}} \frac{\hat{D}T}{\hat{Dt}} = \frac{\hat{Dp}}{\hat{Dt}} + \frac{M^2}{Re} \hat{\sigma} : (\hat{\nabla} \hat{u}) - \frac{1}{RePr} \hat{\nabla} \cdot \hat{q} \\
+ \sum m \hat{\nabla} \cdot (\hat{\rho} \hat{h}_m \hat{Y}_m \hat{V}_m) - \sum m \hat{\rho} Y_m \hat{V}_m \cdot \nabla \hat{h}_m - \sum m \hat{h}_m \hat{\rho}_m. \tag{A.25}
\]

Here we defined \( Pr = \mu_0 C_{p,\text{mix0}} / \lambda T_0, \hat{\lambda}_T = \lambda T / \lambda T_0, \) and \( C_{p,\text{mix0}} = \bar{R}_\text{gas} \).

Again in the low Mach number limit, the leading order equation becomes

\[
\hat{\rho}^0 \hat{C}_{p,\text{mix}}^0 \frac{\hat{D}T^0}{\hat{Dt}} = \frac{\partial \hat{p}^0}{\partial t} - \frac{1}{RePr} \hat{\nabla} \cdot \hat{\dot{q}} \\
+ \sum m \hat{\nabla} \cdot (\hat{\rho}^0 \hat{h}_m^0 \hat{Y}_m^0 \hat{V}_m^0) - \sum m \hat{\rho}^0 Y_m^0 \hat{V}_m^0 \cdot \nabla \hat{h}_m^0 - \sum m \hat{h}_m^0 \hat{\rho}_m^0. \tag{A.26}
\]

The first term on the RHS drops for external flow where ambient pressure is constant in time.

### A.2.4 Species transport equations

The species transport equation, Eqn.\[A.1d\] does not change its form at the low Mach number limit. However, in the low Mach number limit, the pressure diffusion term in Eqn.(\[A.5\]) is dropped from the leading-order balance for the diffusion velocity. Further, if the thermal diffusion (Soret effect) is negligible, we can use a Fickian diffusion model:

\[
Y_m \dot{V}_m = -D_m \nabla Y_m, \tag{A.27}
\]

where \( D_m \) is the mixture-averaged diffusion coefficient of species \( m \).

It should be pointed out that the Soret effect can be of comparable order to the Fickian diffusion term [Ern & Giovangigli 1998], if included, near the flame front where the
temperature gradient is large (Dimotakis 2005). Therefore, computed flame thickness may contain some error. However, its influence on flame speed is not significant as shown by Ern & Giovangigli (1999), and the difference was mostly observed in rich flames in their study.

It is known that the inclusion of the Soret effect needs to be accommodated with a change in a wall boundary condition. $V_m = 0$ at the boundary does not necessarily translate to $n \cdot \nabla Y_m = 0$ on the boundary when additional diffusion terms are included. In particular, when iso-thermal wall boundary condition is used, the boundary conditions to species transport equations must be derived from Eqns. (A.4) and (A.5) to be consistent with a diffusion model.

A.2.5 Transport properties

For viscosity and other transport properties, we know $\mu$ and $\lambda$, or alternatively $\hat{\mu}$ and $\hat{\lambda}$ as a function of temperature $\theta$. To solve the leading-order momentum equations, we need $\mu^0$ and $\lambda^0$ as a function of $\theta^0$. The Taylor series expansion gives us $\hat{\mu}(\hat{\theta}) = \mu^0(\theta^0) + O(\epsilon)$ and $\hat{\lambda}(\hat{\theta}) = \lambda^0(\theta^0) + O(\epsilon)$.

Even when the compositions ($Y_m$s) change, the same argument follows.

A.3 The governing equations used in this study

The low Mach number equations derived above can be written in the following form for equations in dimensional form:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0, \tag{A.28a}
\]

\[
\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho} \nabla p^* + \frac{1}{\rho} \nabla \cdot (\mu [\nabla u + \nabla u^t]) + f, \tag{A.28b}
\]

\[
\frac{\partial Y_m}{\partial t} + u \cdot \nabla Y_m = -\frac{1}{\rho} \nabla \cdot (\rho Y_m V_m + \omega_m), \tag{A.28c}
\]

\[
\rho C_{p,\text{mix}} \frac{DT}{Dt} = \nabla \cdot (\lambda_T \nabla T) - \rho \sum_m h_m \omega_m - \sum_m \rho Y_m V_m \cdot \nabla h_m, \tag{A.28d}
\]

\[
\rho = \rho_0 / \left( R_{\text{gas}} T \right). \tag{A.28e}
\]
Expand the viscous term of Eqn. (A.28b) to obtain

\[ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p^* + \mathbf{L}^e(\mathbf{u}) + \mathbf{L}^i(\mathbf{u}) + \mathbf{f}, \]  

(A.29)

where

\[ \mathbf{L}^i(\mathbf{u}) = \frac{\mu}{\rho} \nabla^2 \mathbf{u} \]  

(A.30a)

\[ \mathbf{L}^e(\mathbf{u}) = \frac{\mu}{\rho} \nabla(\nabla \cdot \mathbf{u}) + \frac{\nabla \mu}{\rho} \cdot \nabla \mathbf{u} + \nabla \mathbf{u}^T \]  

(A.30b)

This particular form of the momentum equation is integrated. One could combine the axial and radial momentum equations to obtain the fully coupled weak formulation of the momentum equations that allows the entire viscous term to be treated implicitly. However, for efficiency reasons, the above approach is taken in this study. The viscous term is further split into the explicit part and the implicit part to save computational time in forming the stiffness matrix used in the implicit integration of the viscous term. Details can be found in Chapter 3.

For the species transport equations, \( \mathbf{V}_m \) needs to be modeled, and the Fickian diffusion model is used in this study.

The diffusion velocity of species \( m \) must satisfy the following:

\[ \sum_m Y_m \mathbf{V}_m = 0. \]  

(A.31)

This constraint is the result of mass conservation. However, a drawback of using the Fickian diffusion model is that this constraint on \( \mathbf{V}_m \) may not be satisfied. A technique proposed by Coffee & Heimerl (1981) defines a corrective differential velocity, \( \mathbf{V}_c \) such that

\[ \mathbf{V}_c = -\sum_m Y_m \tilde{\mathbf{V}}_m, \]  

(A.32)

where \( \tilde{\mathbf{V}}_m \) now follows the Fickian model:

\[ \tilde{\mathbf{V}}_m = -\frac{D_m}{Y_m} \nabla Y_m, \]  

(A.33)
so that

\[ \mathbf{V}_m = \mathbf{\hat{V}}_m + \mathbf{V}_c, \]  \hspace{1cm} (A.34)

therefore

\[ \sum_m Y_m \mathbf{V}_m = 0. \]  \hspace{1cm} (A.35)

Effectively, this method subtracts the correct amount of excessive diffusion distributed equally among all species existing in the system. For example, Day & Bell (2000) used this algorithm in their study. However, use of this algorithm in an implicit method is not practical because Eqn.(A.32) couples the entire species through summation. If one of the species is inert and abundant,

\[ V_{m'} = \frac{1}{Y_{m'}} \left( \sum_{m \in M'} D_m \nabla Y_m \right), \]  \hspace{1cm} (A.36)

where \( m' \) is the abundant species (nitrogen in this study), and the set \( M' \) contains all the other species but \( m' \). This is a more practical method to satisfy the constraint without sacrificing efficiency.

Again, similar to the momentum equation, the diffusion term is expanded into an explicit part and an implicit part:

\[ \frac{\partial Y_m}{\partial t} + \mathbf{u} \cdot \nabla Y_m = D_m \nabla^2 Y_m + \frac{1}{\rho} \nabla \rho D_m \cdot \nabla Y_m + \dot{\omega}_m \]  \hspace{1cm} (A.37)

for \( m \in M' \). It should be noted that as a result of the constraint Eqn.(A.36) and the mass conservation, \( Y_{N_2} \) is uniquely determined by the constraint, Eqn.(2.50).

When the Fickian diffusion model is employed, the last term in Eqn.(A.28d) disappears. The radiant heat flux is ignored because it has a noticeable effect only at the lean flammability limits (Egolfopoulos, 1994). The Dufour heat flux is also negligible (Ern & Giovangigli, 1998; Williams, 1985). By putting those assumptions into the low Mach number limit of the temperature transport equations, we obtain,

\[ \rho C_{p,\text{mix}} \frac{DT}{Dt} = \lambda_T \nabla^2 T + \nabla \lambda_T \cdot \nabla T - \rho \sum_m h_m \dot{\omega}_m. \]  \hspace{1cm} (A.38)

Again, the diffusion term has been split into explicit and implicit parts.
A.4 The differential form of the equations in cylindrical coordinates

Vector notation is used throughout this thesis, however, an explicit derivative formulation is necessary to implement the evaluation of derivatives. The differential form of various tensor expressions in cylindrical coordinates can be found in Lagerstrom (1964) and Emmons (1958).

Let \((r, \theta, z)\) be the radial, azimuthal, and axial coordinates of the cylindrical polar coordinate respectively, and \((u_r, u_\theta, u_z)\) be the corresponding velocity vector. To compute the stress tensor in cylindrical coordinates, we need the following:

\[
\nabla \mathbf{u} = \begin{pmatrix}
\frac{\partial u_r}{\partial r} & \frac{1}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} & \frac{\partial u_r}{\partial z} \\
\frac{\partial u_\theta}{\partial r} & \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} & \frac{\partial u_\theta}{\partial z} \\
\frac{\partial u_z}{\partial r} & \frac{1}{r} \frac{\partial u_z}{\partial \theta} & \frac{\partial u_z}{\partial z}
\end{pmatrix}
\]

(A.39)

and let \(\mathbf{T}\) be a 3-dimensional tensor of rank 2, then

\[
\nabla \cdot \mathbf{T} = \begin{pmatrix}
\frac{\partial \tau_{rr}}{\partial r} + \frac{\tau_{rr}}{r} + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial \theta} - \frac{\tau_{\theta \theta}}{r} + \frac{\partial \tau_{rz}}{\partial z} \\
\frac{\partial \tau_{r\theta}}{\partial r} + \frac{\tau_{r\theta}}{r} + \frac{1}{r} \frac{\partial \tau_{\theta \theta}}{\partial \theta} + \frac{\tau_{\theta \theta}}{r} + \frac{\partial \tau_{\theta z}}{\partial z} \\
\frac{\partial \tau_{rz}}{\partial r} + \frac{\tau_{rz}}{r} + \frac{1}{r} \frac{\partial \tau_{r z}}{\partial \theta} + \frac{\tau_{r z}}{r} + \frac{\partial \tau_{zz}}{\partial z}
\end{pmatrix}.
\]

(A.40)

Therefore, the viscous stress tensor, \(\mathbf{T} = \mu[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T]\) has the following form,

\[
\mathbf{T} = \begin{pmatrix}
2\mu \frac{\partial u_r}{\partial r} & \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} + \frac{\partial u_r}{\partial z} \right) & \mu \left( \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \\
\mu \left( \frac{1}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_\theta}{r} + \frac{\partial u_r}{\partial z} \right) & 2\mu \left( \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \right) & \mu \left( \frac{\partial u_\theta}{\partial z} + \frac{1}{r} \frac{\partial u_z}{\partial \theta} \right) \\
\mu \left( \frac{\partial u_z}{\partial z} + \frac{1}{r} \frac{\partial u_z}{\partial \theta} \right) & \mu \left( \frac{\partial u_z}{\partial z} + \frac{1}{r} \frac{\partial u_z}{\partial \theta} \right) & 2\mu \frac{\partial u_z}{\partial z}
\end{pmatrix}.
\]

(A.41)
So that, the viscous term in the momentum equation is,

\[
\nabla \cdot T = \begin{pmatrix}
\frac{1}{r} \frac{\partial}{\partial r} \left( 2 \mu r \frac{\partial u_r}{\partial r} \right) + \frac{1}{\rho} \frac{\partial}{\partial \theta} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_r}{r} + \frac{\partial u_\theta}{\partial r} \right) \right) - \frac{1}{r} \left( 2 \mu \left( \frac{1}{r} \frac{\partial u_\theta}{\partial r} + \frac{u_r}{r} \right) \right) \\
\frac{1}{r} \frac{\partial}{\partial \theta} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_r}{r} + \frac{\partial u_\theta}{\partial r} \right) \right) + \frac{1}{\rho} \frac{\partial}{\partial z} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) \\
\frac{1}{r} \frac{\partial}{\partial z} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) + \frac{1}{\rho} \frac{\partial}{\partial \theta} \left( \mu \left( \frac{1}{r} \frac{\partial u_\theta}{\partial z} + \frac{\partial u_z}{\partial \theta} \right) \right)
\end{pmatrix}
\]

(A.42)

which is in the case of axisymmetry,

\[
\nabla \cdot T = \begin{pmatrix}
\frac{1}{r} \frac{\partial}{\partial r} \left( 2 \mu r \frac{\partial u_r}{\partial r} \right) - 2 \mu \frac{u_r}{r^2} + \frac{\partial}{\partial z} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) \\
0 \\
\frac{1}{r} \frac{\partial}{\partial r} \left( \mu \left( \frac{1}{r} \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) + \frac{\partial}{\partial z} \left( 2 \mu \frac{\partial u_z}{\partial z} \right)
\end{pmatrix}
\]

(A.43)

Similarly, the nonlinear terms in the momentum equations are

\[
\mathbf{u} \cdot \nabla \mathbf{u} = \begin{pmatrix}
\frac{u_r}{r} \frac{\partial u_r}{\partial r} + \frac{u_\theta}{r} \frac{\partial u_r}{\partial \theta} - \frac{u_r^2}{r} + u_z \frac{\partial u_r}{\partial z} \\
u_r \frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r u_\theta}{r} + u_z \frac{\partial u_\theta}{\partial z} \\
u_r \frac{\partial u_z}{\partial r} + \frac{u_\theta}{r} \frac{\partial u_z}{\partial \theta} + u_z \frac{\partial u_z}{\partial z}
\end{pmatrix}
\]

(A.44)

which is in the axisymmetric case,

\[
\mathbf{u} \cdot \nabla \mathbf{u} = \begin{pmatrix}
\frac{u_r}{r} \frac{\partial u_r}{\partial r} + u_z \frac{\partial u_r}{\partial z} \\
0 \\
u_r \frac{\partial u_z}{\partial r} + u_z \frac{\partial u_z}{\partial z}
\end{pmatrix}
\]

(A.45)

The vector equation (A.28b) is comprised of two equations in differential form,

\[
\frac{\partial u_z}{\partial t} + \left( u_z^2 \frac{\partial u_z}{\partial z} + u_r \frac{\partial u_z}{\partial r} \right) = - \frac{1}{\rho} \frac{\partial p^*}{\partial z} \\
+ \frac{1}{\rho} \frac{1}{r} \frac{\partial}{\partial r} \left( \mu r \left( \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) + \frac{\partial}{\partial z} \left( 2 \mu \frac{\partial u_z}{\partial z} \right) + f_z
\]

(A.46)
and

\[
\frac{\partial u_r}{\partial t} + \left( u_z \frac{\partial u_r}{\partial z} + u_r \frac{\partial u_r}{\partial r} \right) = -\frac{1}{\rho} \frac{\partial p^*}{\partial r} + \frac{1}{\rho} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left( 2\mu r \frac{\partial u_r}{\partial r} \right) - 2\mu \frac{u_r}{r^2} + \frac{\partial}{\partial z} \left( \mu \left( \frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r} \right) \right) \right\} + f_r
\]

(A.47)

for the cylindrical coordinates when the flow is axisymmetric (\(\partial/\partial \theta = 0\)) and there is no swirl (\(u_\theta = 0\)), both of which are assumed in this study.
Appendix B

Initial conditions to unsteady flame simulations

B.1 Introduction

Although the initial conditions to a time-dependent problem are important, not much attention has been paid to this aspect. Unfortunately, many multidimensional numerical studies on reacting flows that employed unsteady equations—Day & Bell (2000); Tomboulides et al. (1997); Prouzakis et al. (1998)—did not clearly describe the initial condition in their paper. In the present study, a one-dimensional solution using Cantera is used to create an initial condition for the Phase II flame simulation as follows:

\[
\begin{align*}
    u(z,r) &= f(r; 0.00425, 0.0025)u^{\text{Cantera}}(z) & (B.1a) \\
    v(z,r) &= f(r; 0.00425, 0.0025)rv^{\text{Cantera}}(z) & (B.1b) \\
    T(z,r) &= T^{\text{Cantera}}(z) & (B.1c) \\
    Y_m(z,r) &= f(r; 0.005, 0.0005)Y_m & (B.1d)
\end{align*}
\]

where \( f(r; r_0, \delta r) \) is a blending function that takes a value between 0 and 1, and is defined by

\[
f(r; r_0, \delta r) = \left\{ \tanh \left[ \frac{2.64665}{\delta r/2}(r_0 - r) \right] + 1 \right\} / 2 \quad (B.2)
\]

where the value of 2.64665 is obtained by \( \tanh^{-1}(0.99) \). In other words, \( r_0 \) specifies the half-value location, and \( \delta r \) specifies the width of the 99% window.

In obtaining the one-dimensional solution using Cantera, error tolerances and grid refine-
ment parameters are tightened to obtain the numerical solution on a successively increased number of points, starting from around 100. The final solution is solved on more than 1,000 points.

Then, the initial conditions to the Phase III simulations are obtained by combining two different numerical solutions. One is the Phase II solution. The other is the cold flow numerical solution of the nozzle interior flow with the nozzle filled by the mixture at the inflow condition.

\section*{B.2 Conversion tables}

Throughout this study, air is assumed to be composed of 21\% Oxygen and 79\% Nitrogen by volume, which is $Y_{\text{oxygen}} = 0.232918$ and $Y_{\text{nitrogen}} = 0.767082$. Table \ref{tab:mole-fractions} tabulates mole fractions and mass fractions of Methane, Oxygen, and Nitrogen at various equivalence ratios. Note the equivalence ratios tabulated here are accurate up to the sixth digit, \textit{i.e.}, 1.30 should read 1.300000.
### Table B.1: Methane / Air flame equivalence ratio to mass fraction

<table>
<thead>
<tr>
<th>$\Phi$</th>
<th>$X_{CH_4}$</th>
<th>$X_{O_2}$</th>
<th>$X_{N_2}$</th>
<th>$Y_{CH_4}$</th>
<th>$Y_{O_2}$</th>
<th>$Y_{N_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.049881</td>
<td>0.199525</td>
<td>0.750594</td>
<td>0.028366</td>
<td>0.226311</td>
<td>0.745324</td>
</tr>
<tr>
<td>0.55</td>
<td>0.054597</td>
<td>0.198535</td>
<td>0.746868</td>
<td>0.031114</td>
<td>0.225671</td>
<td>0.743215</td>
</tr>
<tr>
<td>0.60</td>
<td>0.059266</td>
<td>0.197554</td>
<td>0.743180</td>
<td>0.033847</td>
<td>0.225034</td>
<td>0.741119</td>
</tr>
<tr>
<td>0.65</td>
<td>0.063890</td>
<td>0.196583</td>
<td>0.739527</td>
<td>0.036564</td>
<td>0.224401</td>
<td>0.739035</td>
</tr>
<tr>
<td>0.70</td>
<td>0.068468</td>
<td>0.195622</td>
<td>0.735911</td>
<td>0.039266</td>
<td>0.223772</td>
<td>0.736962</td>
</tr>
<tr>
<td>0.75</td>
<td>0.073001</td>
<td>0.194670</td>
<td>0.732329</td>
<td>0.041954</td>
<td>0.223146</td>
<td>0.734901</td>
</tr>
<tr>
<td>0.80</td>
<td>0.077491</td>
<td>0.193727</td>
<td>0.728782</td>
<td>0.044626</td>
<td>0.222523</td>
<td>0.732851</td>
</tr>
<tr>
<td>0.85</td>
<td>0.081937</td>
<td>0.192793</td>
<td>0.725270</td>
<td>0.047283</td>
<td>0.221905</td>
<td>0.730813</td>
</tr>
<tr>
<td>0.90</td>
<td>0.086341</td>
<td>0.191868</td>
<td>0.721791</td>
<td>0.049925</td>
<td>0.221289</td>
<td>0.728786</td>
</tr>
<tr>
<td>0.95</td>
<td>0.090702</td>
<td>0.190952</td>
<td>0.718345</td>
<td>0.052553</td>
<td>0.220677</td>
<td>0.726770</td>
</tr>
<tr>
<td>1.00</td>
<td>0.095023</td>
<td>0.190045</td>
<td>0.714932</td>
<td>0.055167</td>
<td>0.220068</td>
<td>0.724765</td>
</tr>
<tr>
<td>1.05</td>
<td>0.099302</td>
<td>0.189147</td>
<td>0.711551</td>
<td>0.057766</td>
<td>0.219463</td>
<td>0.722772</td>
</tr>
<tr>
<td>1.10</td>
<td>0.103541</td>
<td>0.188256</td>
<td>0.708203</td>
<td>0.060350</td>
<td>0.218861</td>
<td>0.720789</td>
</tr>
<tr>
<td>1.15</td>
<td>0.107740</td>
<td>0.187375</td>
<td>0.704885</td>
<td>0.062921</td>
<td>0.218262</td>
<td>0.718817</td>
</tr>
<tr>
<td>1.20</td>
<td>0.111901</td>
<td>0.186501</td>
<td>0.701599</td>
<td>0.065477</td>
<td>0.217667</td>
<td>0.716856</td>
</tr>
<tr>
<td>1.25</td>
<td>0.116022</td>
<td>0.185635</td>
<td>0.698343</td>
<td>0.068020</td>
<td>0.217074</td>
<td>0.714905</td>
</tr>
<tr>
<td>1.30</td>
<td>0.120106</td>
<td>0.184778</td>
<td>0.695117</td>
<td>0.070549</td>
<td>0.216485</td>
<td>0.712966</td>
</tr>
<tr>
<td>1.35</td>
<td>0.124152</td>
<td>0.183928</td>
<td>0.691920</td>
<td>0.073064</td>
<td>0.215900</td>
<td>0.711036</td>
</tr>
<tr>
<td>1.40</td>
<td>0.128160</td>
<td>0.183086</td>
<td>0.688753</td>
<td>0.075566</td>
<td>0.215317</td>
<td>0.709117</td>
</tr>
<tr>
<td>1.45</td>
<td>0.132133</td>
<td>0.182252</td>
<td>0.685615</td>
<td>0.078054</td>
<td>0.214737</td>
<td>0.707209</td>
</tr>
<tr>
<td>1.50</td>
<td>0.136069</td>
<td>0.181425</td>
<td>0.682505</td>
<td>0.080529</td>
<td>0.214161</td>
<td>0.705310</td>
</tr>
</tbody>
</table>
Appendix C

Flame simulation data

C.1 Phase II: 2D numerical experiments in cylindrical domain
Figure C.1: $\Phi=0.70$, Case B070-2
Figure C.2: $\Phi=0.70$, Case B070-7
Figure C.3: $\Phi=0.70$, Case B070-12
Figure C.4: \( \Phi = 0.70 \), Case B070-17
Figure C.5: $\Phi=0.70$, Case B070-22
Figure C.6: $\Phi=0.70$, Case B070-27
Distance from wall, $z$ (m)
Axial velocity, $u$ (m/s)

-0.008 -0.006 -0.004 -0.002 0
0.0
0.5
1.0
1.5
2.0
2.5

(a) Axial velocity profile

CH mass fraction, $Y_{CH} \times 10^7$

-0.008 -0.006 -0.004 -0.002 0.000
0.0
0.2
0.4
0.6
0.8
1.0
1.2

(b) CH mass fraction

Pressure eigenvalue, $\Lambda \times 10^{-3}$ (kg/m$^3$-s$^2$)

-0.008 -0.006 -0.004 -0.002 0
-180
-160
-140
-120
-100
-80
-60
-40
-20
0
20

(c) Pressure eigenvalue

Distance from wall, $z$ (m)
Distance from axis, $r$ (m)

-0.008 -0.006 -0.004 -0.002 0.000
0.000
0.002
0.004
0.006
0.008
0.010

(d) CH contour

Figure C.7: $\Phi=0.70$, Case B070-30
Distance from wall, $z$ (m)
Axial velocity, $u$ (m/s)
-0.008 -0.006 -0.004 -0.002 0
0.0
0.5
1.0
1.5
2.0
2.5

CH mass fraction, $Y_{CH} \times 10^7$
-0.008 -0.006 -0.004 -0.002 0.000
0.0
1.0
2.0
3.0
4.0
5.0

Pressure eigenvalue, $\Lambda \times 10^{-3}$ (kg/m$^3$ s$^2$)
-0.008 -0.006 -0.004 -0.002 0
-180
-160
-140
-120
-100
-80
-60
-40
-20
0
20

Distance from axis, $r$ (m)
-0.008 -0.006 -0.004 -0.002 0.000
0.000
0.005
0.010

Distance from wall, $z$ (m)
Distance from axis, $r$ (m)
-0.008 -0.006 -0.004 -0.002 0.000
0.000
0.005
0.010

Figure C.8: $\Phi=0.80$, Case B080-17
Figure C.9: $\Phi=0.80$, Case B080-46
Figure C.10: $\Phi=0.90$, Case B090-30
Figure C.11: Φ=0.90, Case B090-50
Figure C.12: $\Phi=0.90$, Case B090-106
Figure C.13: $\Phi=0.90$, Case B090-112
Figure C.14: $\Phi=1.00$, Case B100-23
Figure C.15: $\Phi=1.20$, Case B120-10
C.2  Phase III: 2D simulation of laboratory flames
Figure C.16: \( \Phi=0.70 \), Case C070-4
Distance from wall, $z$ (m)

Axial velocity, $u$ (m/s)

-0.008 -0.006 -0.004 -0.002 0

0.0
0.5
1.0
1.5
2.0
2.5

(a) Axial velocity profile

CH mass fraction, $Y_{CH} \times 10^7$

-0.008 -0.006 -0.004 -0.002 0.000

0.0
0.2
0.4
0.6
0.8
1.0
1.2

(b) CH mass fraction

Pressure eigenvalue, $\Lambda \times 10^{-3}$

(kg/m$^3$-s$^2$)

Distance from wall, $z$ (m)

-0.008 -0.006 -0.004 -0.002 0

-120
-100
-80
-60
-40
-20
0
20

(c) Pressure eigenvalue

Distance from axis, $r$ (m)

-0.008 -0.006 -0.004 -0.002 0.000

0.000
0.005
0.010

(d) CH contour

Figure C.17: $\Phi=0.70$, Case C070-11
Figure C.18: Φ=0.70, Case C070-8
Figure C.19: \( \Phi=0.70 \), Case C070-9
Distance from wall, \( z \) (m)
Axial velocity, \( u \) (m/s)
-0.008 -0.006 -0.004 -0.002 0

(a) Axial velocity profile

CH mass fraction, \( Y_{CH} \) × 10^7
-0.008 -0.006 -0.004 -0.002 0.000
0.0
2.0
4.0
6.0
8.0
10.0

(b) CH mass fraction

Pressure eigenvalue, \( \Lambda \times 10^{-3} \) (kg/m\(^3\)-s\(^2\))
-0.008 -0.006 -0.004 -0.002 0
-40
-20
0
20
40
60
80
100
120
140
160

(c) Pressure eigenvalue

Distance from wall, \( z \) (m)
Distance from axis, \( r \) (m)
-0.008 -0.006 -0.004 -0.002 0.000
0.000
0.005
0.010

(d) CH contour

Figure C.20: \( \Phi=0.90 \), Case C090-5
Distance from wall, $z$ (m)
Axial velocity, $u$ (m/s)
-0.008 -0.006 -0.004 -0.002 0

Distance from wall, $z$ (m)
CH mass fraction, $Y_{CH} \times 10^7$
-0.008 -0.006 -0.004 -0.002 0.000

Distance from wall, $z$ (m)
Pressure eigenvalue, $\Lambda \times 10^{-3}$ (kg/m$^3$-s$^2$)
-0.008 -0.006 -0.004 -0.002 0

Distance from wall, $z$ (m)
Distance from axis, $r$ (m)
-0.008 -0.006 -0.004 -0.002 0.000
0.000
0.005
0.010

Figure C.21: $\Phi=0.90$, Case C090-2
Figure C.22: $\Phi=0.90$, Case C090-4
Figure C.23: Φ=0.90, Case C090-3
Figure C.24: $\Phi=1.20$, Case C120-1
Appendix D

Post-simulation analysis tool

D.1 PSV velocity simulation

The following MATLAB® script takes one-dimensional flow data and computes a particle motion. By filtering the particle motion as done in the laboratory, a simulation of PSV-measured velocity data can be obtained. For more details, Bergthorson (2005) should be consulted. The script was originally written by Jeff Bergthorson, with improvements made by the author and Laurent Benezech. It is included here for an archival purpose.

% PSV: PSV velocity correction.
% This program computes what PSV velocity data should be given simulation
% velocity data.
% 
% USAGE: [PM, PSV, PFORCE, converged] = psv(data)
% returns the particle motion (PM) data and particle-streak velocimetry
% (PSV) data as well as three forces acting on a particle at each point.
% PM contains three columns. The first column is the position of the
% particle. The second is the velocity of the particle at that point and
% the third column is the acceleration of the particle at that point. PSV
% contains two columns. The first column is the position where the
% computed PSV velocity is defiend and the second column is the
% corresponding velocity data. PFORCE contains three columns: Stokes drag,
% Thermophoretic force and gravity. Each row corresponds to the first
% column of PM where particle locations are specified. converged is a
% single integer value that returns 1 if the integration was successful and
% the particle reached to the end of the domain and 0 otherwise. The
% function takes one to five input arguments. The first input is the
% simulation data, which is mandatory and must be arranged in the following
% order: [x u V T r M mu] where the first column (x) is the axial position,
% u is the axial velocity, V is the spreading rate, dv/dr, T is the
% temperature, r is the density, M is the average molecular weight and mu
% is the mixture gas viscosity. It is assumed the data is stored such that
% it is an increasing order in x. The second to sixth inputs are optional.
% The second input argument can be 'Al2O3', 'Zeeo' or user-specified
% particle properties. When particle properties are specified, it must be
% 1x3 cell with the first cell being the particle diameter (m), the second
% being the particle density (Kg/m^3) and the last one is the thermal
% conductivity of the particle. This can be a single number (constant
% thermal conductivity), or an array with the first column being
% temperature and the second column being the corresponding thermal
% conductivity at the given temperature. (See example below) When
% particle data is omitted, 'Zeeo' is used by default. The third to sixth
% parameters are, the chopping frequency of the PSV, max iteration of
% temporal integration of particle motion, the number of subiterations
% within each PSV chopping cycle and the integration order (1 or 2). When
% omitted, 1600, 100000, 100 and 2 are used by default respectively.
%
% example 1:
% psv(simdata, {1.0e-5, 1000, 1.0}): simulates a particle whose diameter is
% 10 microns, density is 1000 (kg/m^3), and the thermal conductivity is 1
% (W/m-K) at all temperature.
%
% example 2:
% psv(simdata, {3.0e-6, 2400, [300 100.0; 2000 1000.0]}): simulates a
% particle whose diameter is 3 microns, density is 2400 (kg/m^3), and the
% thermal conductivity is 100 (W/m-K) at 300K and 1000 (W/m-K) at 2000 K.
At other temperature, the conductivity is linearly interpolated or extrapolated.

Example 3:
psv(simdata, 'Al2O3', 2400, 250000, 200) simulates a motion of Al2O3 particle with PSV chopping frequency of 2400. It tries to iterate up to a quarter of a million iterations at a time step of 1/2400 x 1/200.

For more details, Jeff Bergthorson's thesis should be consulted.

AUTHORS: Jeff M. Bergthorson, Laurent Benezech, and Kazuo Sone.

function [PM, PSV, PFORCE, converged] = psv(flowData, varargin)

narg = size(varargin,2);

if (nargin < 1)
    error('psv:NumInputs','Not enough input arguments.');
end

%%% Parameter definitions. %%%
if narg < 1
    % When particle is not specified, use 'Zeeo' as default.
    param = 'Zeeo';
else
    param = varargin{1};
end

% Particle properties, {diameter, density, conductivity}.
if iscell( param )
    % This is the case when property is specified by user.
% dpart: Diameter of the particle.
dpart = param{1};
% rhopart: Particle density in kg/m^3
rhopart = param{2};
pcond = param{3};
else
% Use preset values.
if strcmp( param, 'Al2O3' )
    dpart = 1.0e-6; % (m)
rhopart = 3830.0; % (Kg/m^3)
% Thermal conductivity of particles Incropera and deWitt
pcond = [200 55;
    400 26.4;
    600 15.8;
    800 10.4;
    1000 7.85;
    1200 6.55;
    1500 5.66;
    2000 6;
    2500 6.4];
elseif strcmp( param, 'Zeeo' )
    dpart = 3.0e-6; % (m)
rhopart = 2400.0; % (kg/m^3)
pcond = 2.3; % (W/mK)
else
    % Error!
    error('psv:ParticleInfo','Particle specification incorrect.');
end
end

% frequency: Chopping frequency of PSV.
if narg > 1
    frequency = varargin{2};
else
    frequency = 1600;
end

% max_iterations: Maximum number of integration steps.
if narg > 2
    max_iterations = varargin{3};
else
    max_iterations=100000;
end

% nsubcycle: number of iterations to get 1 PSV cycle.
% Default value is 100.
if narg > 3
    nsubcycle = uint16( varargin{4} );
else
    nsubcycle = 100;
end

% integorder: Order of time integration (1 or 2).
% Default value is 2.
% Use 1st order only when speed is important.
if narg > 4
    integorder = uint16( varargin{5} );
    if integorder < 1 || integorder > 2
        error('psv:IntegOrder', ...
            'Incorrect integration order. Must be 1 or 2');
    end
else
    integorder = 2;
% deltat: Integration of particle equations.
deltat = 1/frequency * 1/double(nsubcycle);
% halfdeltat: 1/2 deltat (used in 2nd order method)
halfdeltat = 0.5 * deltat;

%%% Universal constants. %%%
% Using the simulated flow, temperature and property fields,
% we need to model a particles behavior as it traverses the flowfield.
%Gravitational acceleration
a_gravity=9.81; %m/s^2

% mpart: Mass of the particle
mpart=rhopart*pi*4/3*(dpart/2)^3; %in kg;

% End of parameters.
%

xsim=flowData(:,1);
lsim=max(xsim); % End of the simulation domain.

usim=flowData(:,2);
Vsim=flowData(:,3);
T=flowData(:,4);
rho=flowData(:,5);
Mbar=flowData(:,6);
visc=flowData(:,7);

gradT=zeros(size(flowData,1),1);
for i=2:1:size(flowData,1)-1;
    gradT(i)=(T(i+1)-T(i-1))/(xsim(i+1)-xsim(i-1));
end

gradT(size(flowData,1))=gradT(size(flowData,1)-1);

%%
%%Initialize the particle with a corrected particle velocity and
%%acceleration
%i = 1;
tpart(1)=0;

xpart(1)=0;

%% CKWp: approximately 1+1.142*Kn for small Kn.
%% : see Eqn. (A.10) in Jeff Bergthorson's thesis.
CKWp(1)=1.05;

%% vpart: Eqns. (A.23) & (A.24) in Jeff Bergthorson's thesis
%% : vpart/vfluid = 1/(1+C_KW*tau_S*sigma_f)
%% : sigma_f = du/dx.
% dudx = -2*Vsim(1);
vpart(1)=usim(1)/(1+CKWp(1)*(rhopart*dpart^2/(18*visc(1)))*dudx);
%% ufp: fluid velocity at particle location.
ufp(1)=usim(1);

[apart(1), F_thermophoretic(1), F_stokes(1)] = ...
  Particle_acceleration(mpart, dpart, pcond, vpart(1), ...
    [ufp(1) T(1) gradT(1) rho(1) Mbar(1) visc(1)]);

%%The gravitational force is a constant and can be calculated prior to the
%%loop
F_gravity=-mpart*a_gravity;

continue_iteration = 1;
while continue_iteration
  % Increment time by deltat, and update x, u, and a through 1st order
%% explicit Euler method of integration.
    i = i + 1;
    tpart(i)=tpart(i-1)+deltat;
    xpart(i)=xpart(i-1)+deltat*vpart(i-1);
    vpart(i)=vpart(i-1)+deltat*apart(i-1);

%% Interpolate the simulated fields at the particle location.
    flow_at_particle = interp1(xsim, [usim T gradT rho Mbar visc], ...
                           xpart(i), [], 'extrap');

    if integorder == 2
        apart(i) = Particle_acceleration(mpart, dpart, pcond, vpart(i), ...
                          flow_at_particle);

        % Apply second order correction (2nd order Runge-Kutta method)
        xpart(i) = xpart(i-1) + halfdeltat*(vpart(i-1)+vpart(i));
        vpart(i) = vpart(i-1) + halfdeltat*(apart(i-1)+apart(i));

        % Interpolate the simulated fields at the particle location.
        flow_at_particle = interp1(xsim, [usim T gradT rho Mbar visc], ...
                           xpart(i), [], 'extrap');
    end

    [apart(i), F_thermophoretic(i), F_stokes(i)] = ...
        Particle_acceleration(mpart, dpart, pcond, vpart(i), ...
                          flow_at_particle);

    % Check if the particle has reached to the wall close enough.
    % and if so, exit the loop.
    % Note xpart has been computed way above here, but I have to get apart
    % otherwise the resulting array does not have the same number of rows.
    if xpart(i) >= lsim || i > max_iterations
continue_iteration = 0;
end
end

% Issue an warning message if not converged.
if i > max_iterations
    converged = 0;
    warning('Iteration may not have converged. ');
    warning('Increase maximum number of iterations.');
else
    converged = 1;
end

%now want to estimate the PSV velocity from this flowfield.
nb0 = nsubcycle+1;
b1 = size(tpart,2);
ne0 = 1;
e1 = size(tpart,2)-nsubcycle;
PSV(:,1) = (xpart(nb0:nb1)+xpart(ne0:ne1)) * 0.5;
PSV(:,2) = (xpart(nb0:nb1)-xpart(ne0:ne1)) * frequency;

% Assign to the returning matrix.
PM = [xpart; vpart; apart]';
PFORCE = [F_stokes; F_thermophoretic; F_gravity*ones(1,i)'];

%-------------------------
% Private functions
%-------------------------

%-----------------------------------------------
% Particle_acceleration: This function computes a particle acceleration.
function [apart, F_thermophoretic, F_stokes] = ...
Particle_acceleration(mpart, dpart, pcond, vpart, flow_p)

ufp = flow_p(1);
Tp = flow_p(2);
gradTp = flow_p(3);
rhop = flow_p(4);
Mbarp = flow_p(5);
viscp = flow_p(6);

% R_univ: Universal gas constant.
R_univ=8314; %J/kmolK

% kftransp: The translational part of the thermal conductivity
% : See Eqn. (A.26) in Jeff's thesis.
kftransp = 15*R_univ*viscp/(4*Mbarp);

% kpp : Thermal conductivity of the particle. When it is not a function of
% temperature, use it. If it is a function of temperature, pcond should be
% given by a table - simply interpolate.
if length(pcond) == 1
    kpp = pcond;
else
    kpp = interp1(pcond(:,1), pcond(:,2), Tp, [], 'extrap');
end

% Now calculate the parameters needed for the forces:
% Knp: Knudsen number.
Knp = Knudsen_number(Tp, Mbarp, viscp, rhop, dpart, R_univ);

% CKWp: Eqn. (A.10)
CKWp = 1+Knp*[1.142+0.558*exp(-0.999/Knp)];
%Now calculate the thermophoretic and stokes drag forces:
F_thermophoretic = Force_Thermophoretic_Talbot(viscp, dpart, rhop, ...
                 kftransp, kpp, Knp, gradTp, Tp);
F_stokes = Force_Stokes(viscp, dpart, vpart, ufp, CKWp);

%Calculate the acceleration: Eqn.(A.7)
apart = (F_stokes+F_thermophoretic)/mpart;
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% Force_Stokes: This function computes stokes drag.
function [F_stokes] = Force_Stokes(viscp, dpart, vpart, ufp, CKWp)

%Stokes drag: Eqn.(A.9)
F_stokes = -3*pi*viscp*dpart*(vpart-ufp)/CKWp;
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% Force_Thermophoretic_Talbot: This function computes thermophoretic force
% by Talbot’s model.
function [F_tp] = Force_Thermophoretic_Talbot(viscp, dpart, rhop, ...
                                            kftransp, kpp, Knp, gradTp, Tp)

%Talbot: Eqn.(A.28)
num = -(6*pi*viscp^2/rhop*dpart*1.17*(kftransp/kpp+2.18*Knp)*gradTp/Tp);
den = (1+3*1.14*Knp)*(1+2*kftransp/kpp+2*2.18*Knp);
F_tp = num/den;
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
% Knudsen_number: This function computes Knudsen number
function [Knp] = Knudsen_number(Tp, Mbarp, viscp, rhop, dpart, R_univ)
% cbarp: Eqn.(A.12)
cbarp = (8*R_univ*Tp/(Mbarp*pi))^0.5;

% Lambdap: Eqn.(A.11)
Lambdap = 2*viscp/(rhop*cbarp);

% Knp: Knudsen number.
Knp = 2*Lambdap/dpart;

%-----------------------------------------------
Appendix E

Opposed-jet hydrogen flames and comparison of mechanisms

This appendix is designed to demonstrate the importance of the hydrogen subset of hydrocarbon mechanisms. Understanding hydrogen flame behavior should supplement hydrocarbon combustion models because each hydrocarbon reaction model contains a hydrogen reaction submechanism (see Table 1.1). Figs. E.1–E.4 compare performance of three different hydrogen mechanisms: H1, H2, and H4 in Table 1.1 in the above opposed-jet configuration at the same condition. H2 and H4 tend to show an agreement, in particular, their velocity profile is nearly identical, while the GRI-Mech 3.0 submechanism shows discernible departure from the other two. However, they all disagree in the prediction of hydrogen radical profile. Some of the reactions that appear in the hydrogen reaction submodel have a significant effect on the flame speed of hydrocarbons, as pointed out by several authors (Williams, 2000; Turányi et al., 2002), and refinement in the hydrocarbon kinetics model should come in tandem with understanding of hydrogen reactions, including extinction characteristics.
Figure E.1: Comparison of hydroxyl radical mass fraction along the axis between three
different hydrogen mechanisms: GRI-Mech 3.0 [H1] (solid line), SD05 [H2] (dashed line),
and YDR91 [H4] (dot-dashed line)
Figure E.2: Comparison of hydrogen radical mass fraction along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)
Figure E.3: Comparison of temperature profile along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)
Figure E.4: Comparison of axial velocity profile along the axis between three different hydrogen mechanisms. (Legend as in Fig. E.1.)
Bibliography


