User Guide for Detailed Modeling of Laminar Flames and High-Resolution LES in OpenFOAM®

Adhiraj Dasgupta

February 12, 2019

Contents

Abstract

In this work a set of libraries and solvers have been developed within OpenFOAM®'s code framework that can be used to model laminar flames as well as perform high-resolution LES of reacting flows. Transport properties are calculated using a detailed, mixture-averaged model, and both a fully-compressible, and an 'incompressible', low-Mach-number solver have been developed. This theory manual provides a brief summary of the governing equations and models used in the code; it also serves as an installation guide, and provides a step-by-step guide for setting up a simulation along with all the settings with their recommended values.

Additionally a new feature has been added to complement OpenFOAM®'s chemistry model that allows for the use of reduced chemical mechanisms that are available as Fortran subroutines.

Chapter 1

Numerical Model

The governing equations for reacting flows are essentially the statements of conservation of mass, momentum, species, and energy. A detailed summary has been provided elsewhere [1, 2], and only a brief overview will be provided here.

1.1 The fully-compressible formulation

The fully-compressible version of the conservation equations may be summarized as

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \ , \tag{1.1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_j u_i)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \ , \tag{1.2}$$

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial (\rho u_j Y_k)}{\partial x_j} = -\frac{\partial J_{k,j}}{\partial x_j} + \dot{\omega}_k \ , \tag{1.3}$$

$$\begin{aligned} \frac{\partial \rho h_s}{\partial t} + \frac{\partial (\rho u_j h_s)}{\partial x_j} + \frac{\partial \rho K}{\partial t} + \frac{\partial (\rho u_j K)}{\partial x_j} = & -\frac{\partial q_j}{\partial x_j} + \frac{\partial p}{\partial t} + \rho u_i f_i + q_{rad} \\ & - \sum_{l=1}^{N} \dot{\omega}_l \Delta h^{\circ}_{f,l} + \frac{\partial (\tau_{ij} u_i)}{\partial x_j} \ , \end{aligned} \tag{1.4}$$

$$p = \rho R_u T \sum_{l=1}^{N} \frac{Y_l}{W_l} \ . \tag{1.5}$$

Here ρ is the gas density, u_i is the gas velocity, p is pressure, τ_{ij} is the viscous stress tensor, Y_k is the mass fraction of species k, $J_{k,j}$ is the molecular diffusion flux of the kth species, $\dot{\omega}_k$ is the production rate of species k, h_s is the mixture sensible enthalpy, K is the kinetic energy per unit mass of the flow, q_i is the heat flux, f_i is the body force per unit volume (e.g., due to gravity), q_{rad} is the radiative heat source, $\Delta h^{\circ}_{f,k}$ is the standard state heat of formation of species k, R_u is the universal gas constant, and W_k is the molecular weight of species k. Also, the term $\frac{\partial (\tau_{ij} u_i)}{\partial x_j}$ contains the the viscous dissipation function $\Phi = \tau_{ij} \frac{\partial u_i}{\partial x_j}$.

1.2 The low-Mach-number formulation

In the low-Mach-number or the 'incompressible' formulation the pressure driving the flow is decoupled from the thermodynamic state of the system; the pressure is split into two parts as $p = p_0 + p_d$ [3, 4]. Here p_0 is the thermodynamic pressure (assumed constant), and the hydrodynamic pressure p_d drives the flow field. The governing equations are also modified; the flow kinetic energy K is considered negligible, and the viscous dissipation term is also dropped. The resulting equations are Eqs. (1.1), (1.3), and

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_j u_i)}{\partial x_j} = -\frac{\partial p_d}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \ , \tag{1.6}$$

$$\frac{\partial \rho h_s}{\partial t} + \frac{\partial (\rho u_j h_s)}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} + q_{rad} - \sum_{l=1}^{N} \dot{\omega}_l \Delta h^\circ_{f,l} \ , \tag{1.7}$$

$$p_0 = \rho R_u T \sum_{l=1}^{N} \frac{Y_l}{W_l}. \tag{1.8}$$

When running high-resolution, implicit LES all the variables in these equations may be interpreted as filtered quantities.

1.3 The transport model

In this work mass diffusion is modeled using the Hirschfelder and Curtiss approximation [5]. Mass conservation is ensured by using a correction velocity approach [6, 7]. Thus the diffusion flux for species k is given as

$$J_{k,i} = \rho \left(-D_k \frac{\partial X_k}{\partial x_i} + Y_k V_i^c \right),$$

where X_k is the mole fraction of species k, D_k is the mixture-averaged diffusivity for species k, and V_i^c is a spatially-varying correction velocity, which is the same for all species. The heat flux is given by

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \sum_{l=1}^{N} h_l J_{l,i},$$

where λ is the thermal conductivity of the gas mixture, and h_l is the enthalpy of species l. Lastly, the viscous stress tensor τ_{ij} is computed as

$$\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu S_{kk}\delta_{ij} \ ,$$

where μ is the dynamic viscosity of the gas mixture, and

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

is the strain rate tensor.

Evaluation of the transport properties follows the procedure outlined by Kee et al. [6]. The viscosity of the gas-phase species k is given by [5, 6]

$$\mu_k = \frac{5}{16}\frac{\sqrt{\pi m_k k_B T}}{\pi \sigma_k^2 \Omega^{(2,2)*}}, \tag{1.9}$$

where σ_k is the Lennard-Jones collision diameter, m_k is the molecular mass for species k, k_B is Boltzmann's constant, and $\Omega^{(2,2)*}$ is the reduced collision integral, which is a function of the reduced temperature T_k^* given by

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

where ε_k is the Lennard-Jones potential well depth for species k. The collision integrals are evaluated from the reduced temperature using curve-fits generated by Neufeld et al. [8].

The binary diffusion coefficient for gas-phase species j and k is given by [5, 6]

$$D_{jk} = \frac{3}{16}\frac{\sqrt{2\pi k_B^3 T^3/m_{jk}}}{p\pi \sigma_{jk}^2 \Omega^{(1,1)*}}, \tag{1.10}$$

where m_{jk} is the reduced molecular mass for species j and k, defined as

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

σ_{jk} is the reduced colllision diameter, and the reduced collision integral $\Omega^{(1,1)*}$ is a function of the reduced temperature T_{jk}^* [1]. These parameters are calculated based on whether the two molecules involved are both polar, both non-polar or one polar and the other non polar.

For the case of two non-polar molecules or two polar molecules interacting, the reduced quantitites are defined as

$$\frac{\varepsilon_{jk}}{k_B} = \sqrt{\left(\frac{\varepsilon_j}{k_B}\right)\left(\frac{\varepsilon_k}{k_B}\right)},$$
$$\sigma_{jk} = \frac{1}{2}\left(\sigma_j + \sigma_k\right).$$

For a non-polar molecule interacting with a polar one, the reduced polarizability α_n^* and the reduced dipole moment μ_p^* are defined as

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

[1] In the low-Mach-number formulation p_0 is used in place of p.

where α denotes the polarizability and μ denotes the dipole moment; the subscript p refers to the polar species and the subscript n to the non-polar species. The reduced temperature T^*_{jk} is defined as

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

The reduced quantities are calculated as

$$\frac{\varepsilon_{np}}{k_B} = \zeta^2 \sqrt{\left(\frac{\varepsilon_n}{k_B}\right)\left(\frac{\varepsilon_p}{k_B}\right)},$$
$$\sigma_{np} = \frac{1}{2}\left(\sigma_n + \sigma_p\right)\zeta^{-\frac{1}{6}},$$

where

$$\zeta = 1 + \frac{1}{4}\alpha^*_n \mu^*_p \sqrt{\frac{\varepsilon_p}{\varepsilon_n}}.$$

The thermal conductivity is assumed to have contributions from translational, rotational, and vibrational components. The thermal conductivity of species k is given by [6, 9]

$$\lambda_k = \frac{\mu_k}{W_k}\left(f_{trans,k} C_{vk,trans} + f_{rot,k} C_{vk,rot} + f_{vib,k} C_{vk,vib}\right), \tag{1.11}$$

where $C_{vk,trans}$, $C_{vk,rot}$, and $C_{vk,vib}$ are respectively the translational, rotational, and vibrational contributions to the constant-volume specific heat C_{vk} of species k,

$$f_{trans,k} = \frac{5}{2}\left(1 - \frac{2}{\pi}\frac{C_{vk,rot}}{C_{vk,trans}}\frac{A_k}{B_k}\right),$$
$$f_{rot,k} = \frac{\rho_k D_{kk}}{\mu_k}\left(1 + \frac{2}{\pi}\frac{A_k}{B_k}\right),$$
$$f_{vib,k} = \frac{\rho_k D_{kk}}{\mu_k}.$$

Here the self-diffusion coefficient D_{kk} is obtained by substituting $j = k$ in Eq. 1.10, and the density of gas-phase species k is obtained as

$$\rho_k = \frac{p W_k}{R_u T}.$$

Further,

$$A_k = \frac{5}{2} - \frac{\rho_k D_{kk}}{\mu_k},$$
$$B_k = Z_{rot,k} + \frac{2}{\pi}\left(\frac{5}{3}\frac{C_{vk,rot}}{R_u} + \frac{\rho_k D_{kk}}{\mu_k}\right).$$

Here $Z_{rot,k}$ is the rotational relaxation number for species k and is given by

$$Z_{rot,k}(T) = Z_{rot,k}(298)\frac{F_k(298)}{F_k(T)},$$

where

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

The translational and rotational components of the gas specific heat are dependent on the kind of molecule under consideration. For a linear molecule,

$$\begin{aligned}\frac{C_{vk,trans}}{R_u} &= \frac{3}{2},\\ \frac{C_{vk,rot}}{R_u} &= 1,\\ C_{vk,vib} &= C_{vk} - \frac{5}{2}R_u.\end{aligned}$$

For a non-linear molecule

$$\begin{aligned}\frac{C_{vk,trans}}{R_u} &= \frac{3}{2},\\ \frac{C_{vk,rot}}{R_u} &= \frac{3}{2},\\ C_{vk,vib} &= C_{vk} - 3R_u.\end{aligned}$$

For a single atom (for example H, O, etc.) there are no internal contributions to the specific heat. Hence

$$\begin{aligned}\frac{C_{vk,trans}}{R_u} &= \frac{3}{2},\\ \frac{C_{vk,rot}}{R_u} &= 0,\\ C_{vk,vib} &= 0.\end{aligned}$$

Transport properties for the mixture are computed using a mixture-averaged approach. The mixture-averaged diffusivities are obtained as [6, 10]

$$D_k = \frac{1 - Y_k}{\sum_{\substack{l=1\\ l\neq k}}^{N} \frac{X_l}{D_{kl}}}, \tag{1.12}$$

where D_{kl} is the binary diffusivity for the species pair k and l. The mixture viscosity is obtained as [6, 11]

$$\mu = \sum_{l=1}^{N} \frac{X_l \mu_l}{\sum_{j=1}^{N} X_j \Phi_{lj}}, \tag{1.13}$$

where μ_l is the dynamic viscosity of species l, and

$$\Phi_{mn} = \frac{1}{\sqrt{8}} \left(1 + \frac{W_m}{W_n}\right)^{-\frac{1}{2}} \left[1 + \left(\frac{\mu_m}{\mu_n}\right)^{\frac{1}{2}} \left(\frac{W_n}{W_m}\right)^{\frac{1}{4}}\right]^2 . \tag{1.14}$$

The mixture-averaged thermal conductivity is obtained as [6, 12]

$$\lambda = \frac{1}{2} \left(\sum_{l=1}^{N} X_l \lambda_l + \frac{1}{\sum_{l=1}^{N} X_l / \lambda_l} \right), \tag{1.15}$$

where λ_l is the thermal conductivity of species l.

To speed up calculations, the temperature-dependent parts of the transport properties for the individual species are fitted using the method of least squares implemented in the GNU Scientific Library [13]. The logarithms of the properties are fitted to polynomials in the logarithm of temperature, as was done by Kee et al. [6]. It is to be noted that the viscosity and the thermal conductivity of species k are a function of temperature only, and thus these curve-fits are straightforward and unambiguous. However, the binary diffusivities are functions of temperature and pressure; the fits are therefore performed at 1 bar pressure. The values obtained from the curve fits are divided by the pressure in bar in order to obtain the actual diffusivities. Details of the fitting process can be found elsewhere [2].

1.4 Using reduced chemical mechanisms

In some cases the chemical mechanism to be used is available only as a Fortran subroutine that evaluates the reaction rates for the species. The commonly used quasi-steady-state (QSS) assumption provides such an example. In this case since fewer species are transported, some savings in computational effort can be expected.

Chapter 2

Installation and Running

The present code framework is based on OpenFOAM® version 2.3.x. It is assumed that the user has a working installation of this version of OpenFOAM® on their computer.

2.1 Installation

The transport model library depends on the mole fraction library for some calculations, and so the latter must be built first. The easiest way to do that is to copy the moleFractions directory to the user's directory and issuing the command `wmake libso` in the terminal.

Once the mole fraction library is successfully built, the transport library may be built. The user may need to modify the provided options file in order to specify the location of the user's source code and the user's libraries. Then the command `wmake libso` should work.

If all goes well, as they should, at this stage the libraries have been built successfully; it remains now to build the applications that use the libraries. The transport fitting program must be built and executed before any solvers can be run. This utility is called `fitTransport`, and it has the GNU Scientific Library (https://www.gnu.org/software/gsl) as a dependency. If GSL is installed in the standard locations, as would be the case if it is built using the default options, the provided options file should work. Issuing the command `wmake` at the command prompt should install the fitting program.

The compressible solver is called `laminarReactingFoam` and the low-Mach-number solver is called `laminarReactingLMFoam` which is influenced by the reactingLMFoam code [4]. For each solver a sample options file is provided, and the user may need to change it to include the locations of the user's code and libraries. Then the solvers may be built by issuing the command `wmake` in the corresponding directory.

2.2 Setting up and running

Setting up a case for use with the solver is similar to the way the same case would be set up for `reactingFoam`, with a few key differences:

1. A file called *transportProperties* must be placed in the constant directory of the case.

2. For the low-Mach-number solver the variable pReff (corresponding to p_0) must be provided in the file *chemistryProperties* in the constant directory.

3. Additional numerical schemes must be specified in the file *fvSchemes* in the system directory of the case.

4. For running the low-Mach-number solver, solver options must be specified for the dynamic pressure p_d in the file *fvSolutions* in the system directory of the case.

5. Optionally, the extra term in the pressure equation can be turned on or off through a flag set in the file *thermoPhysicalProperties* in the constant directory of the case.

2.2.1 The transportProperties file

The file *transportProperties* has the following entries:

transportModel

This option tells the code which transport model to use. The available options are

1. *mixtureAverage*,

2. *LewisNumber*.

For each of these options, a corresponding subdictionary must be specified in the file.

Subdictionary *mixtureAverageProperties*

This subdictionary contains the options specific to running with the mixture-averaged transport model. The entries are:

thermophoresis	This option indicates whether thermophoresis will be included for light species. By default this is set to off.
CutOff	This is the molecular weight below which thermophoresis will be computed. The default value is 5.

gradX In the original formulation for Fickian diffusion velocities, the flux is taken to be proportional to the gradient of the mole fraction, and the diffusion velocity of species k is given as

$$\begin{aligned} V_{k,j} &= -\frac{D_k}{X_k}\frac{\partial X_k}{\partial x_j} \\ &= -\frac{D_k}{Y_k}\frac{\partial Y_k}{\partial x_j} - \frac{D_k}{\bar{M}}\frac{\partial \bar{M}}{\partial x_j}, \end{aligned}$$

where $\bar{M}$ is the mixture molecular weight.
If this option is set to on, the above equation is used to calculate the diffusion velocity. However, if it is set to be off, the flux is taken to be proportional to the gradient of the mass fraction,

$$V_{k,j} = -\frac{D_{km}}{Y_k}\frac{\partial Y_k}{\partial x_j}.$$

By default this is set to be on.

Subdictionary *LewisNumberProperties*

This subdictionary contains the options specific to running with the constant Lewis number model. The options are to be specified in the form of a subdictionary of *LewisNumberProperties*, named *Le*.

Le This contains the names of the species with the corresponding Lewis numbers to be used for each species. By default the Lewis numbers are taken to be unity, so a simulation with an empty *Le* dictionary will enforce unity Lewis number for all species.

CHEMKINFile

This is the path to the kinetics file in Chemkin format [14].

CHEMKINThermoFile

This is the path to the thermo data in Chemkin format.

CHEMKINTransportFile

This is the path to the transport database in Chemkin format.

Tmin* and *Tmax

The transport properties are fitted between these two temperatures. By default they are taken to be 300 K and 3000 K.

viscousDissipation

For the compressible formulation, this option indicates whether or not the viscous dissipation term $\tau_{ij}\frac{\partial u_i}{\partial x_j}$ should be included in the energy equation. By default this is set to off.

2.2.2 Modifications to the file *fvSchemes*

The numerical schemes and solver options are very similar to the ones used by `reactingFoam`. In addition, some new entries are added to the *divSchemes* subdictionary. The new keywords are

1. `div(phi,Yi)`,
2. `div((mu*dev2(T(grad(U)))))`,
3. `div(JHs)`,
4. `div(Hconduction)` or `div(Econduction)`, and
5. `div(phi,he)`.

Additionally, if the low-Mach-number solver is used, the variable `pd` must be added to the *fluxRequired* subdictionary.

2.2.3 The keyword *pCorr*

The implementation of the pressure equation in OpenFOAM® contains an extra flux term, likely for added stability. This term can cause problems in some cases. In this code the user has the option of disabling or enabling this extra term through the keyword *pCorr* in the file *thermoPhysicalProperties*.

By default the term is dropped.

2.2.4 Fitting the transport properties and running

The utility `fitTransport` generates fits to the transport properties. The coefficients for the fits are written to the following files in the constant directory of the case: *conductivityProperties*, *diffusivityProperties*, *thermophoreticProperties*, and *viscosityProperties*. For each species, these files contain the fit coeffients, and the fitting error.

In addition, the molecular parameters read from the transport database are summarized in the file *transportData* in the constant directory. Also, the transport properties are written out as a function of temperature in a new directory called transport.

Once the fits are generated, the code can be run simply by typing `laminarReactingFoam`, or `laminarReactingLMFoam`.

2.2.5 Using reduced chemical mechanisms

As an added option, the present work implemented a way to incorporate reduced chemical mechanisms within the OpenFOAM® framework. The relevant code is placed under the provided *chemistryModel* directory that should be copied to the user's own directory. For now this implementation only works for mechanisms that are available as Fortran routines called *ckwyp*, but may be extended easily to include other forms. Compiling this piece of code produces the shared library `libCustomChemistryModel.so`.

It should be noted that the library does not by itself contain the mechanism information; the Fortran mechanism must be compiled separately into a shared library. Since the description of the chemical mechanism is incomplete, directly linking the library `libCustomChemistryModel.so` to the solvers will not work. Thus the two libraries—`libCustomChemistryMode.so`, and the Fortran shared library must be added to the *controlDict* of the case.

To use the new model, the following steps should be taken:

1. The Fortran mechanism must be compiled to produce a shared library.
2. Both libraries should be added to the file *controlDict* in the system directory.
3. The file *chemistryProperties* should be modified:
   - The keyword *chemistrySolver* should be set to *QSS*.
   - A subdictionary called *QSSCoeffs* should be added. The content of this should be similar to the *odeCoeffs*, since the stiff ODE system is integrated the same way; only the reaction rates are computed in a new way.
4. For a mechanism using the QSS assumption there are not reactions specified in the Chemkin mechanism file; however a dummy reaction must be specified to keep OpenFOAM®'s Chemkin parser happy. For instance, if modeling ethylene combustion, the reaction block can look like this:

   REACTIONS
   $C2H4 + 3O2 => C2H4 + 3O2$ 0.0 0.0 0.0
   END

2.2.6 Tips and tricks

Most of the other settings in use are analogous to the corresponding settings used by `reactingFoam`. However, a few settings were found to work well, and as such may be used as rough guidelines. They are mentioned here:

- Since a lot of terms in the equations are directly calculated from gradients, very sharp gradients can be problematic. The *cellMDLimited* scheme has been found to be stable.

- Whenever possible, running with the low-mach-number solver is to be preferred, for numerical stability even when using relatively coarse time steps. The savings in computational time are likely to be at least an order of magnitude.
- The term *pCorr* may cause problems, and it may be better to keep it turned off unless there is evidence to indicate otherwise.

Bibliography

[1] A. Dasgupta, E. Gonzalez-Juez, and D. C. Haworth, "Flame simulations with an open-source code," *Computer Physics Communications*, vol. 237, pp. 219 – 229, 2019.

[2] A. Dasgupta, *Numerical Simulation of AxiSymmetric Laminar Diffusion Flames with Soot*, Ph.D. thesis, Pennsylvania State University, 2015.

[3] A. G. Tomboulides, J. C. Y. Lee, and S. A. Orszag, "Numerical Simulation of Low Mach Number Reactive Flows," *Journal of Scientific Computing*, vol. 12, no. 2, pp. 139–167, Jun 1997.

[4] K.-J. Nogenmyr, C. Chan, and C. Duwig, "Finite rate chemistry effects and combustor liner heat transfer studies in a frame-work of LES of turbulent flames-investigation of pollutant formation using OpenFOAM," 5th OpenFOAM Workshop, Chalmers, Gothenburg, Sweden, 2010.

[5] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird, *Molecular Theory of Gases and Liquids*, John Wiley & Sons, New York, 1994.

[6] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller, "A Fortran Computer Code Package for the Evaluation of Gas Phase Multicomponent Transport Properties," Sand86-8246 unlimited release, Sandia National Laboratory, 1986.

[7] T.P. Coffee and J.M. Heimerl, "Transport algorithms for premixed, laminar steady-state flames," *Combustion and Flame*, vol. 43, pp. 273 – 289, 1981.

[8] P. D. Neufeld, A. R. Janzen, and A. R. Aziz, "Empirical Equations to Calculate 16 of the Transport Collision Integrals $\Omega^{(l,s)*}$ for the Lennard-Jones (12–6) Potential," *The Journal of Chemical Physics*, vol. 57, pp. 1100–1102, 1972.

[9] N. Peters and J. Warnatz, Eds., *Numerical Methods in Laminar Flame Propagation*, Vieweg & Teubner Verlag, 1982.

[10] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, "Transport phenomena," 1960.

[11] C. R. Wilke, "A Viscosity Equation for Gas Mixtures," *The Journal of Chemical Physics*, vol. 18, no. 4, pp. 517–519, 1950.

[12] S. Mathur, P. K. Tondon, and S. C. Saxena, "Thermal conductivity of binary, ternary and quaternary mixtures of rare gases," *Molecular Physics*, vol. 12, pp. 569–579, 1967.

[13] Brian Gough, *GNU Scientific Library Reference Manual - Third Edition*, Network Theory Ltd., 3rd edition, 2009.

[14] R. J. Kee, F. M. Rupley, and J. A. Miller, "CHEMKIN-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics," Tech. Rep. SAND89-8009B, Sandia National Laboratories, 1989.