mech2Foam

mech2Foam is a code that generates the required combustion, thermodynamic, and transported properties model coefficients for a computational fluid dynamic (CFD) simulation with the model/solver XiFoam part of the open-source toolbox OpenFOAM [1], [2]. The code is written in the open-source programming language Python version 3.6 [3], and the output files are suited and tested for OpenFOAM version 7. Although the code is written for the CFD solver XiFoam, other CFD software that uses the same input parameters can utilize *mech2Foam*. This document explains how *mech2Foam* generates combustion, thermodynamic, and transported properties model coefficients. The thermodynamic and transport properties are in the *Generating thermophysicalProperties file* section, whereas the combustion properties section documents the generation of the Gülder model/equation coefficient.

Generating thermophysicalProperties file

The *thermophysicalProperties* file in the XiFoam case folder contains the transport and thermodynamic properties (model coefficients) and defines how these properties are modeled. There is one mixture specific and three specie specific parameters needed for simulating inhomogeneous combustion. The global mixture parameter is the stoichiometric air-fuel mass ratio. Furthermore, the mole weight, NASA polynomial coefficients, and Sutherland coefficients are the three specie specific parameters. These parameters are calculated using the open-source package Cantera version 2.4 [4] and a reaction mechanism/chemical kinetics. As input parameters to *mech2Foam*, the fuel and oxidizer composition, the initial pressure (p), temperature (T), and a reaction mechanism are needed.

The thermodynamic parameters can be evaluated directly for the oxidant and the fuel since all species and concentrations are input values. For the combustion products, the species are not input parameters and must therefore be calculated. Using the inbuilt equilibrate function in Cantera, the burnt product species and concentrations can be determined. The fuel and air mixture is equilibrated at constant pressure and enthalpy to determine the burnt product species at initial input (T,p) conditions. Species with a mole fraction below 0.001 are removed to reduce the number of species in the product gas. The species concentration is then normalized before the parameters can be calculated. The function

calculate_burnt_product_gas_composition in the script *calculateThermoProperties.py* calculates the stoichiometric air-fuel mass ratio.

For the thermodynamic model Janaf in OpenFOAM, the NASA polynomials [5] are used to calculate the heat capacity (C_p), enthalpy (h), and entropy (s). The NASA polynomials are functions of temperature, with seven polynomial coefficients. There are two temperature ranges separated by the minimum, shared/common, and maximum reference temperature. The NASA polynomial equations are shown below.

$$\frac{C_p(T)}{R_u} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 \quad \text{Eq. 1}$$

$$\frac{h(T)}{R_u} = a_0 T + \frac{a_1 T^2}{2} + \frac{a_2 T^3}{3} + \frac{a_3 T^4}{4} + \frac{a_4 T^5}{5} + a_5 \quad \text{Eq. 2}$$

$$\frac{s(T)}{R_u} = a_0 \ln T + a_1 T + \frac{a_2 T^2}{2} + \frac{a_3 T^3}{3} + \frac{a_4 T^4}{4} + a_6 \quad \text{Eq. 3}$$

where: C_p – specific heat capacity at constant pressure, [J/(mol·K)]; R_u – Universal gas constant, [J/(mol·K)]; h – enthalpy, [J/mol]; s – entropy, [J/(mol·K)]; T – temperature, [K], a_0 – polynomial coefficient velocity vector [J/(mol·K)];], a_1 – polynomial coefficient, [J/(mol·K^2)]; a_2 – polynomial coefficient, [J/(mol·K^3)]; a_3 – polynomial coefficient, [J/(mol·K^4)]; a_4 – polynomial coefficient, [J/(mol·K^5)]; a_5 – polynomial coefficient, [J/(mol)]; a_6 – polynomial coefficient, [J/(mol·K)]

As mentioned previously, one of the required inputs to *mech2Foam* is a reaction mechanism/chemical kinetics model. The NASA polynomials (Eq. 1 to Eq. 3) are the most common thermodynamic model used in these reaction mechanisms. Each species in the reaction mechanism has its own set of NASA polynomial coefficients. In mech2Foam, mole weighted NASA polynomial coefficients are calculated from the species-specific NASA polynomial coefficient using Eq. 4. The function *calculate_NASApolynomial_coefficients_for_mixture*, calculates the mole weighted NASA polynomial coefficients used in XiFoam.

$$\bar{a_i} = \sum_{i=0}^{k} a_{i,k} X_k \quad \text{Eq. 4}$$

Where: $\bar{a_i}$ – mole weighted NASA polynomial coefficient of index i; $a_{i,k}$ – NASA polynomial coefficient of index i and specie k; X – mole fraction of specie k; i – polynomial coefficient index; k – species.

Different species may have different reference temperatures, which can cause deviations in thermodynamic properties in some temperature regions. For different reference temperatures in a mixture, the species with the highest mole fraction will determine the mixture's reference temperatures. Suppose a mixture contains species with different reference temperatures. In that case, it will be documented in the support documentation written alongside the *thermophysicalProperties* file. If the minimum reference temperature exceeds 200 K, it is adjusted to 200 K to avoid warnings during OpenFOAM simulations.

The Sutherland equation is the chosen transport model in this case. The dynamic gas viscosity is calculated from the Sutherland equation below (Eq. 5) [6].

$$\mu = \frac{A_s\sqrt{T}}{1 + \frac{T_s}{T}} \qquad \text{Eq. 5}$$

where: μ – dynamic gas viscosity, [Pa·s]; A_S – Sutherland coefficient; T_S – Sutherland Coefficient; T – temperature [K]

In the Sutherland equation, two constants need to be curve fitted (A_s and T_s). The mixture-specific Sutherland coefficients are fitted to the dynamic viscosities gathered from a Cantera Solution object generated using the reaction mechanism, initial conditions, and the three species compositions (fuel, oxidizer, and burntProducts). The temperature range for the viscosity is set to the minimum and maximum NASA polynomial reference temperatures. The Sutherland coefficients are calculated in the *calculate_sutherland_coefficients_for_gas_mixture* function.

Combustion properties– Gülder coefficient

One of the laminar burning velocity (LBV) models in OpenFOAM is the Gülder correlations model [7] shown in Eq. 6.

$$S_L(\phi, T, p) = \omega\phi^{\eta} e^{\xi(\phi - 1.075)} \cdot \left(\frac{T}{T_{ref}}\right)^{\alpha} \cdot \left(\frac{p}{p_{ref}}\right)^{\beta} \cdot \left(1 - X_f \cdot f\right) \qquad \text{Eq. 6}$$

where: S_L – laminar burning velocity, [m/s]; ϕ – fuel-air equivalence ratio, [-]; T – temperature, [K]; p – pressure, [Pa]; ω – Gülder coefficient; η – Gülder coefficient; ξ – Gülder coefficient; T_{ref} – Reference

temperature; α – Gülder coefficient; p_{ref} – reference pressure; β – Gülder coefficient, X_f – mole fraction of inert [-]; f – Gülder coefficient

Five coefficients in the Gülder equation needs to be estimated, which is ω, η, ξ, α, and β. The coefficient f is set to 2.3, independent of the gas composition. X_f is the mole fraction of inert that is not part of the fuel and oxidizer mixture. The Gülder coefficients are fitted in three steps, Eq. 7 to Eq. 9. In the Cantera package, the routine *FreeFlame* was used to calculate the LBVs used in the fitting procedure. The *FreeFlame* routine solves the governing equation for a 1-D premixed, steady, laminar/planar, adiabatic flame. The reference pressure and temperature values are 101.3 kPa and 300 Kelvin, respectively.

$$S_L(\phi) = \omega \phi^{\eta} e^{\xi(\phi - 1.075)} \quad \text{Eq. 7}$$

$$S_L(T) = S_L(\phi = 1, T_{ref}, p_{ref}) \cdot \left(\frac{T}{T_{ref}}\right)^{\alpha} \quad \text{Eq. 8}$$

$$S_L(p) = S_L(\phi = 1, T_{ref}, p_{ref}) \cdot \left(\frac{p}{p_{ref}}\right)^{\beta} \quad \text{Eq. 9}$$

where: S_L – laminar burning velocity, [m/s]; ϕ – fuel-air equivalence ratio, [-]; T – temperature, [K]; p – pressure, [Pa]; ω – Gülder coefficient; η – Gülder coefficient; ξ – Gülder coefficient; T_{ref} – Reference temperature; α – Gülder coefficient; p_{ref} – reference pressure; β – Gülder coefficient.

References

[1] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Comput. Phys.*, vol. 12, no. 6, p. 620, 1998, doi: 10.1063/1.168744.

[2] CFD Direct, "OpenFOAM v7 User Guide," *OpenFOAM v7 User Guide*, Jul. 30, 2021. https://cfd.direct/openfoam/user-guide-v7/

[3] *Python*. [Online]. Available: https://www.python.org/

[4] D. G. Goodwin, H. K. Moffat, and R. L. Speth, *Cantera: An Object-Oriented Software Toolkit For Chemical Kinetics, Thermodynamics, And Transport Processes*. Zenodo, 2017. doi: 10.5281/zenodo.170284.

[5] B. J. McBride, S. Gordon, and M. A. Reno, "Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species," NASA, Technical NASA TM-4513, Oct. 1993. [Online]. Available: https://ntrs.nasa.gov/api/citations/19940013151/downloads/19940013151.pdf

[6] W. Sutherland, "LII. *The viscosity of gases and molecular force*," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 36, no. 223, pp. 507–531, Dec. 1893, doi: 10.1080/14786449308620508.

[7] Ö. L. Gülder, "Correlations of Laminar Combustion Data for Alternative S.I. Engine Fuels," presented at the West Coast International Meeting and Exposition, Aug. 1984. doi: 10.4271/841000.

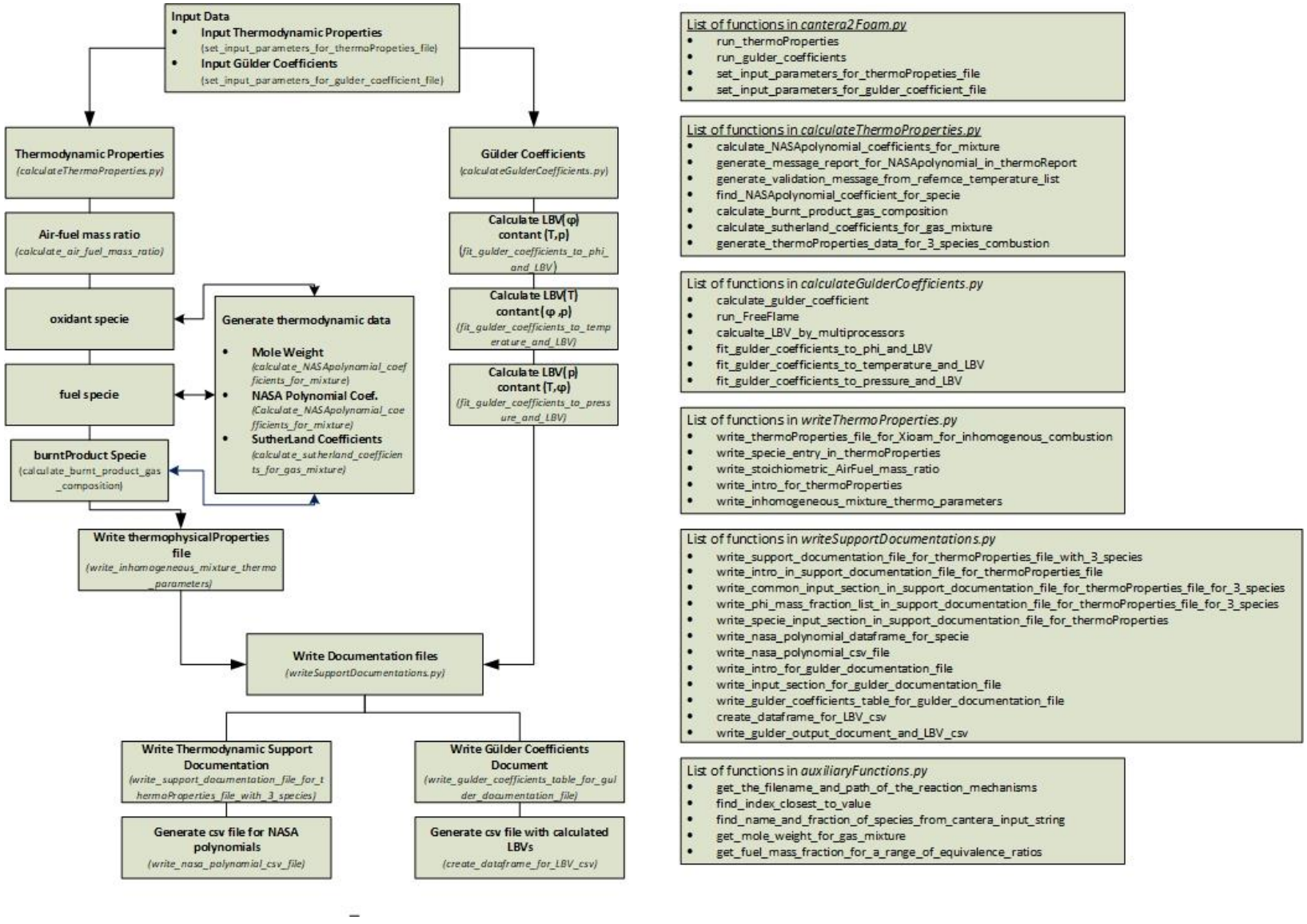

Figure 1. A diagram that illustrates the structure of the mech2Foam code and all the scripts and underlying functions.