Chapter 7

Models and physical properties

OpenFOAM includes a large range of solvers each designed for a specific class of problem. The equations and algorithms differ from one solver to another so that the selection of a solver involves the user making some initial choices on the modelling for their particular case. The choice of solver typically involves scanning through their descriptions in section 3.5 to find the one suitable for the case. It ultimately determines many of the parameters and physical properties required to define the case but leaves the user with some modelling options that can be specified at runtime through the entries in dictionary files in the *constant* directory of a case. This chapter deals with many of the more common models and associated properties that must be specified at runtime.

7.1 Thermophysical models

Thermophysical models are concerned with energy, heat and physical properties. The *thermophysicalProperties* dictionary is read by any solver that uses the **thermophysical** model library. A thermophysical model is constructed in OpenFOAM as a pressure-temperature $p - T$ system from which other properties are computed. There is one compulsory dictionary entry called `thermoType` which specifies the package of thermophysical modelling that is used in the simulation. OpenFOAM includes a large set of pre-compiled combinations of modelling, built within the code using C++ templates. This coding approach assembles thermophysical modelling packages beginning with the equation of state and then adding more layers of thermophysical modelling that derive properties from the previous layer(s). The keyword entries in `thermoType` reflects the multiple layers of modelling and the underlying framework in which they combined. Below is an example entry for `thermoType`:

```
thermoType
{
    type            hePsiThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState perfectGas;
    specie          specie;
    energy          sensibleEnthalpy;
}
```

The keyword entries specify the choice of thermophysical models, *e.g.* constant `transport` (constant viscosity, thermal diffusion), Perfect Gas `equationOfState`, *etc.* In addition there is a keyword entry named `energy` that allows the user to specify the form of energy to be used in the solution and thermodynamics. The following sections explains the entries and options in the `thermoType` package.

7.1.1 Thermophysical and mixture models

The most general solvers that use thermophysical modelling construct a fluidThermo model that allows the user to specify the thermophysical model through the `type` entry (described below) at run-time. These solvers include rhoSimpleFoam, rhoPorousSimpleFoam and rhoPimpleFoam.

Each solver that uses thermophysical modelling constructs an object of a specific thermophysical model class. The model classes are listed below.

psiThermo Thermophysical model for fixed composition, based on compressibility $\psi = (RT)^{-1}$, where R is Gas Constant and T is temperature. The solvers that construct psiThermo include rhoCentralFoam, uncoupledKinematicParcelFoam and coldEngineFoam.

rhoThermo Thermophysical model for fixed composition, based on density ρ. The solvers that construct rhoThermo include the buoyantSimpleFoam, buoyantPimpleFoam, rhoPorousSimpleFoam, twoPhaseEulerFoam and thermoFoam.

psiReactionThermo Thermophysical model for reacting mixture, based on ψ. The solvers that construct psiReactionThermo include many of the *combustion* solvers, *e.g.*sprayFoam, engineFoam, fireFoam and reactingFoam, and some *lagrangian* solvers, *e.g.* coalChemistryFoam.

psiuReactionThermo Thermophysical model for combustion, based on compressibility of unburnt gas ψ_u. The solvers that construct psiuReactionThermo include *combustion* solvers that model combustion based on laminar flame speed and regress variable, *e.g.*XiFoam, XiEngineFoam, PDRFoam.

rhoReactionThermo Thermophysical model for reacting mixture, based on ρ. The solvers that construct rhoReactionThermo include chtMultiRegionFoam, some *combustion* solvers, *e.g.*chemFoam, rhoReactingFoam, rhoReactingBuoyantFoam, and some *lagrangian* solvers, *e.g.* reactingParcelFoam and simpleReactingParcelFoam.

multiphaseMixtureThermo Thermophysical models for multiple phases. The solvers that construct multiphaseMixtureThermo include compressible *multiphase* interface-capturing solvers, *e.g.*compressibleInterFoam, and compressibleMultiphaseInterFoam.

The `type` keyword specifies the underlying thermophysical model. Options are listed below.

- `hePsiThermo`: for solvers that construct psiThermo and psiReactionThermo.
- `heRhoThermo`: for solvers that construct rhoThermo, rhoReactionThermo and multiphaseMixtureThermo.
- `heheuPsiThermo`: for solvers that construct psiuReactionThermo.

The `mixture` specifies the mixture composition. The option typically used for thermophysical models without reactions is `pureMixture`, which represents a mixture with fixed composition. When `pureMixture` is specified, the thermophysical models coefficients are specified within a sub-dictionary called `mixture`.

For mixtures with variable composition, required by thermophysical models with reactions, the `reactingMixture` option is used. Species and reactions are listed in a chemistry file, specified by the `foamChemistryFile` keyword. The `reactingMixture` model then requires the thermophysical models coefficients to be specified for each specie within sub-dictionaries named after each specie, *e.g.* `O2`, `N2`.

For combustion based on laminar flame speed and regress variables, constituents are a set of mixtures, such as `fuel`, `oxidant` and `burntProducts`. The available mixture models for this combustion modelling are `homogeneousMixture`, `inhomogeneousMixture` and `veryInhomogeneousMixture`.

Other models for variable composition are `egrMixture`, `multiComponentMixture` and `singleStepReactingMixture`.

7.1.2 Transport model

The transport modelling concerns evaluating dynamic viscosity μ, thermal conductivity κ and thermal diffusivity α (for internal energy and enthalpy equations). The current transport models are as follows:

const assumes a constant μ and Prandtl number $Pr = c_p\mu/\kappa$ which is simply specified by a two keywords, `mu` and `Pr`, respectively.

sutherland calculates μ as a function of temperature T from a Sutherland coefficient A_s and Sutherland temperature T_s, specified by keywords `As` and `Ts`; μ is calculated according to:

$$\mu = \frac{A_s\sqrt{T}}{1 + T_s/T}. \tag{7.1}$$

polynomial calculates μ and κ as a function of temperature T from a polynomial of any order N, *e.g.*:

$$\mu = \sum_{i=0}^{N-1} a_i T^i. \tag{7.2}$$

logPolynomial calculates $\ln(\mu)$ and $\ln(\kappa)$ as a function of $\ln(T)$ from a polynomial of any order N; from which μ, κ are calculated by taking the exponential, *e.g.*:

$$\ln(\mu) = \sum_{i=0}^{N-1} a_i [\ln(T)]^i. \tag{7.3}$$

7.1.3 Thermodynamic models

The thermodynamic models are concerned with evaluating the specific heat c_p from which other properties are derived. The current thermo models are as follows:

hConst assumes a constant c_p and a heat of fusion H_f which is simply specified by a two values c_p H_f, given by keywords `Cp` and `Hf`.

eConst assumes a constant c_v and a heat of fusion H_f which is simply specified by a two values c_v H_f, given by keywords `Cv` and `Hf`.

janaf calculates c_p as a function of temperature T from a set of coefficients taken from JANAF tables of thermodynamics. The ordered list of coefficients is given in Table 7.1. The function is valid between a lower and upper limit in temperature T_l and T_h respectively. Two sets of coefficients are specified, the first set for temperatures above a common temperature T_c (and below T_h), the second for temperatures below T_c (and above T_l). The function relating c_p to temperature is:

$$c_p = R((((a_4 T + a_3)T + a_2)T + a_1)T + a_0). \tag{7.4}$$

In addition, there are constants of integration, a_5 and a_6, both at high and low temperature, used to evaluating h and s respectively.

hPolynomial calculates c_p as a function of temperature by a polynomial of any order N:

$$c_p = \sum_{i=0}^{N-1} a_i T^i. \tag{7.5}$$

ePolynomial calculates c_v as a function of temperature by a polynomial of any order N:

$$c_v = \sum_{i=0}^{N-1} a_i T^i. \tag{7.6}$$

hPower calculates c_p as a power of temperature according to:

$$c_p = c_0 \left(\frac{T}{T_{\text{ref}}} \right)^{n_0}. \tag{7.7}$$

ePower calculates c_v as a power of temperature according to:

$$c_v = c_0 \left(\frac{T}{T_{\text{ref}}} \right)^{n_0}. \tag{7.8}$$

hTabulated calculates c_p by interpolating tabulated data of (T, c_p) value pairs, *e.g.*:

```
Cp ( (200 1005) (400 1020) );
```

7.1.4 Composition of each constituent

There is currently only one option for the `specie` model which specifies the composition of each constituent. That model is itself named `specie`, which is specified by the following entries.

- `nMoles`: number of moles of component. This entry is only used for combustion modelling based on regress variable with a homogeneous mixture of reactants; otherwise it is set to 1.
- `molWeight` in grams per mole of specie.

Description	Entry	Keyword
Lower temperature limit	T_l (K)	`Tlow`
Upper temperature limit	T_h (K)	`Thigh`
Common temperature	T_c (K)	`Tcommon`
High temperature coefficients	$a_0 \ldots a_4$	`highCpCoeffs (a0 a1 a2 a3 a4...`
High temperature enthalpy offset	a_5	`a5...`
High temperature entropy offset	a_6	`a6)`
Low temperature coefficients	$a_0 \ldots a_4$	`lowCpCoeffs (a0 a1 a2 a3 a4...`
Low temperature enthalpy offset	a_5	`a5...`
Low temperature entropy offset	a_6	`a6)`

Table 7.1: JANAF thermodynamics coefficients.

7.1.5 Equation of state

The following equations of state are available in the thermophysical modelling library.

rhoConst Constant density:

$$\rho = \text{constant}. \tag{7.9}$$

perfectGas Perfect gas:

$$\rho = \frac{1}{RT} p. \tag{7.10}$$

icoTabulated Tabulated data for an incompressible fluid using (T, ρ) value pairs, *e.g.*
`rho ( (200 1010) (400 980) );`

incompressiblePerfectGas Perfect gas for an incompressible fluid:

$$\rho = \frac{1}{RT} p_{\text{ref}}, \tag{7.11}$$

where p_{ref} is a reference pressure.

perfectFluid Perfect fluid:

$$\rho = \frac{1}{RT} p + \rho_0, \tag{7.12}$$

where ρ_0 is the density at $T = 0$.

linear Linear equation of state:

$$\rho = \psi p + \rho_0, \tag{7.13}$$

where ψ is compressibility (not necessarily $(RT)^{-1}$).

adiabaticPerfectFluid Adiabatic perfect fluid:

$$\rho = \rho_0 \left(\frac{p + B}{p_0 + B} \right)^{1/\gamma}, \tag{7.14}$$

where ρ_0, p_0 are reference density and pressure respectively, and B is a model constant.

Boussinesq Boussinesq approximation

$$\rho = \rho_0 \left[1 - \beta \left(T - T_0\right)\right] \tag{7.15}$$

where β is the coeffient of volumetric expansion and ρ_0 is the reference density at reference temperature T_0.

PengRobinsonGas Peng Robinson equation of state:

$$\rho = \frac{1}{zRT} p, \tag{7.16}$$

where the complex function $z = z(p, T)$ can be referenced in the source code in *PengRobinsonGasI.H*, in the *$FOAM_SRC/thermophysicalModels/specie/equationOfState/* directory.

icoPolynomial Incompressible, polynomial equation of state:

$$\rho = \sum_{i=0}^{N-1} a_i T^i, \tag{7.17}$$

where a_i are polynomial coefficients of any order N.

rPolynomial Reciprocal polynomial equation of state for liquids and solids:

$$\frac{1}{\rho} = C_0 + C_1 T + C_2 T^2 - C_3 p - C_4 p T \tag{7.18}$$

where C_i are coefficients.

7.1.6 Selection of energy variable

The user must specify the form of energy to be used in the solution, either internal energy e and enthalpy h, and in forms that include the heat of formation Δh_f or not. This choice is specified through the `energy` keyword.

We refer to *absolute* energy where heat of formation is included, and *sensible* energy where it is not. For example absolute enthalpy h is related to sensible enthalpy h_s by

$$h = h_s + \sum_i c_i \Delta h_f^i \tag{7.19}$$

where c_i and h_f^i are the molar fraction and heat of formation, respectively, of specie i. In most cases, we use the sensible form of energy, for which it is easier to account for energy change due to reactions. Keyword entries for `energy` therefore include *e.g.* `sensibleEnthalpy`, `sensibleInternalEnergy` and `absoluteEnthalpy`.

7.1.7 Thermophysical property data

The basic thermophysical properties are specified for each species from input data. Data entries must contain the name of the specie as the keyword, *e.g.* `O2`, `H2O`, `mixture`, followed by sub-dictionaries of coefficients, including:

`specie` containing *i.e.* number of moles, `nMoles`, of the specie, and molecular weight, `molWeight` in units of g/mol;

`thermodynamics` containing coefficients for the chosen thermodynamic model (see below);

`transport` containing coefficients for the chosen tranpsort model (see below).

The following is an example entry for a specie named `fuel` modelled using sutherland transport and janaf thermodynamics:

```
fuel
{
    specie
    {
        nMoles       1;
        molWeight    16.0428;
    }
    thermodynamics
    {
        Tlow         200;
        Thigh        6000;
        Tcommon      1000;
        highCpCoeffs (1.63543 0.0100844 -3.36924e-06 5.34973e-10
                      -3.15528e-14 -10005.6 9.9937);
        lowCpCoeffs  (5.14988 -0.013671 4.91801e-05 -4.84744e-08
                      1.66694e-11 -10246.6 -4.64132);
    }
    transport
    {
        As           1.67212e-06;
        Ts           170.672;
    }
}
```

The following is an example entry for a specie named `air` modelled using const transport and hConst thermodynamics:

```
air
{
    specie
    {
        nMoles          1;
        molWeight       28.96;
    }
    thermodynamics
    {
        Cp              1004.5;
        Hf              2.544e+06;
    }
```