University of
Strathclyde
Engineering

hyFoam

User Guide

Vincent Casseau, Dr Thomas J. Scanlon

March 2017

James Weir
Fluids Laboratory

Centre for Future Air-Space
Transportation Technology

1 Introduction to *hyFoam*

hyFoam is a newly coded open-source single-temperature computational fluid dynamics (CFD) solver implemented within the OpenFOAM framework. It is based on the code developments made for *hy2Foam*. The code solves the Navier–Stokes–Fourier equations and it has the capability to model a chemically-reacting environment in the continuum regime. Various transport models have been implemented to determine species and mixture thermal properties at high-temperatures. There are also several new species diffusion models and boundary conditions available.

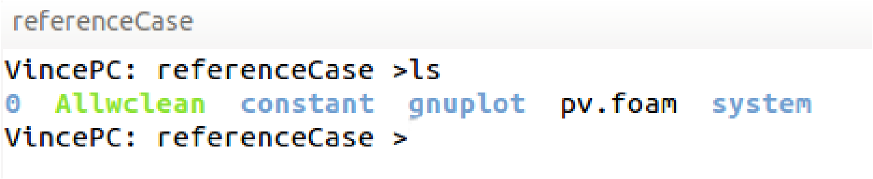
2 Installation of *hyFoam*

The installation of *hyFoam* is straightforward. Please make sure that the path to your OpenFOAM user directory is correct in the bash script called ***install.sh*** before execution. This script will install the new ***strath*** libraries located into the ***src*** folder, will install the *hyFoam* solver located into the ***applications*** folder (as well as a few other apps) and will copy some tutorials into your ***run*** directory.

3 *hyFoam* working environment

3.1 The main folder

The main working folder is no different from any other OpenFOAM solvers. The bash script ***Allwclean*** removes any temporary files within the main directory.



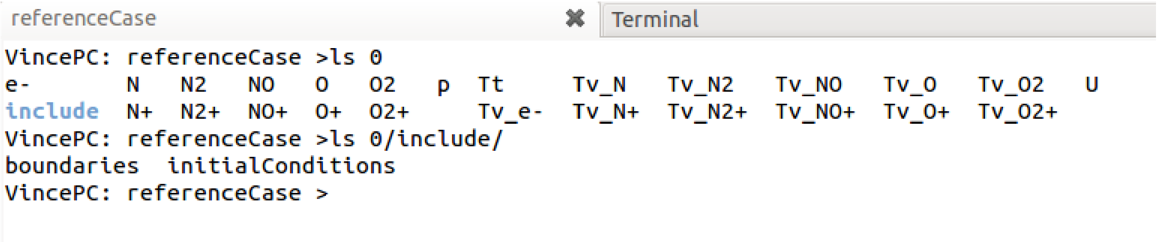
```
VincePC: referenceCase >ls
0  Allwclean  constant  gnuplot  pv.foam  system
VincePC: referenceCase >
```

Figure 1: Main folder contents

NB: The command line to run *hyFoam* is *hy2Foam*.

3.2 The ***0*** folder

The ***0*** folder contains the traditional ***p*** and ***U*** files. The other mandatory files are the temperature, printed as ***Tt***, and the mass-fraction for each species. The prefix *Y_* is omitted which means that the **N2** file is the mass-fraction of N_2.



```
VincePC: referenceCase >ls 0
e-       N   N2   NO   O   O2   p  Tt    Tv_N   Tv_N2   Tv_NO   Tv_O   Tv_O2   U
include  N+  N2+  NO+  O+  O2+     Tv_e-  Tv_N+  Tv_N2+  Tv_NO+  Tv_O+  Tv_O2+
VincePC: referenceCase >ls 0/include/
boundaries  initialConditions
VincePC: referenceCase >
```

Figure 2: The ***0*** folder contents

It is good practice to initialise the fields via an include statement to a dictionary and to set some of the boundary conditions by calling another dictionary. These dictionaries are regrouped into the ***0/include*** folder. For instance, the ***initialConditions*** dictionary can be defined as follow

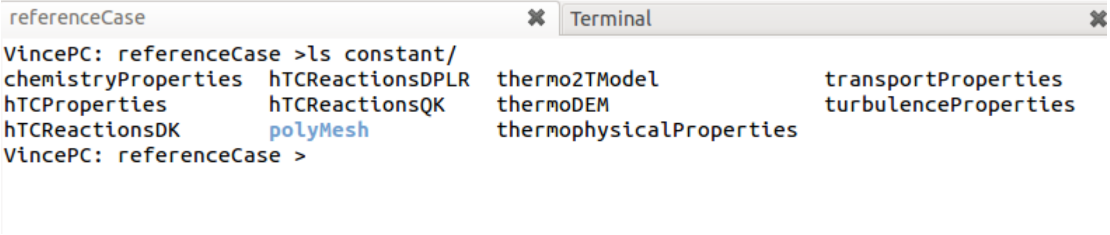
```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Tt        1000;

inletPressure  1.01325e5;

inletVelocity  (800 0 0);
initialVelocityField  (0 0 0);

Y_N2      0.767;
Y_O2      0.233;
Y_NO      0;
Y_CO2     0;
Y_H2O     0;
Y_CO      0;
Y_N       0;
Y_O       0;
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

while the ***boundaries*** dictionary is for the case of an adiabatic heat bath

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
fixedWalls
{
    type            empty;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

3.3 The *constant* folder

The ***constant*** folder regroups all the dictionaries relative to physicochemical phenomena occuring at supersonic/hypersonic speed and that have been implemented into *hy2Foam*. The mesh remains located into the ***polyMesh*** sub-folder.



```
VincePC: referenceCase >ls constant/
chemistryProperties  hTCReactionsDPLR  thermo2TModel            transportProperties
hTCProperties        hTCReactionsQK    thermoDEM                turbulenceProperties
hTCReactionsDK       polyMesh          thermophysicalProperties
VincePC: referenceCase >
```

Figure 3: The ***constant*** folder contents

Recently, a new folder named ***chemDicts*** (not shown in Figure 3) has been created in ***constant*** to contain all chemistry dictionaries. It is advised to make a copy inside the ***constant*** directory before starting working on it.

A short description of the different dictionaries is given below. It is recommended to follow the order listed here while editing the entries of the dictionaries.

- 1) ***thermophysicalProperties***: the standard OpenFOAM dictionary that prescribes which thermophysical models are going to be used and the location of some sub-dictionaries.

- 2) ***turbulenceProperties***: the standard OpenFOAM dictionary that deals with turbulence modelling. In *hy2Foam*, the options are *laminar* and *RAS2Model*.

- 3a) ***hTCProperties***: the dictionary that allows the user to activate or deactivate the high-temperature chemistry (hTC) features and to output some desired species-related fields such as number densities or vibrational degrees of freedom.

- 3b) ***chemistryProperties***: the dictionary that defines the different chemistry inputs

- 3c) ***hTCReactionsDPLR/QK/DK/ES/...***: the list of the species composing the mixture is defined in those dictionaries. It is strongly recommended not to change the order of the species in the list (molecules - ions - atoms - free electrons). Please uncomment the species and chemical reactions you may need.

- 4) ***thermoDEM***: the dictionary that provides the species thermodynamic data for the different species loaded in dictionary 3c).

- 5) ***transportProperties***: the dictionary that defines the models used for species diffusion and the mixing rule. Some parameters such as the mean free path or the local gradient-length Knudsen number can be printed from this dictionary.

In the subsequent subsections, the main features of the dictionaries listed above are reviewed.

3.3.1 The *thermophysicalProperties* dictionary

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

thermoType
{
    type            heRho2Thermo;
    mixture         reacting2Mixture;
    transport       BlottnerEucken;
    thermo          decoupledEnergyModes;
    energy          sensible2InternalEnergy;
    equationOfState perfect2Gas;
    specie          advancedSpecie;
}

hyLight yes;

downgradeToSingleTv no;
downgradeToSingleTemperature yes;

chemistryReader foam2ChemistryReader;

foamChemistryFile "$FOAM_CASE/constant/hTCReactionsJ92";

foamChemistryThermoFile "$FOAM_CASE/constant/thermoDEM";

temperatureBounds
{
    Tlow    200;
    Thigh   22000;
}

// ************************************************************************* //
```

The *thermoType* defined in the ***thermophysicalProperties*** dictionary uses the new ***strath*** thermophysical models. The user has the choice between the five transport models listed in Table 1.

Model name	shear viscosity	thermal diffusivity
constant	constant	constant Eucken
BlottnerEucken	Blottner	Eucken
powerLawEucken	power law	Eucken
SutherlandEucken	Sutherland	Eucken
CEA	CEA2 (NASA)	CEA2 (NASA)

Table 1: Available transport models in *hyFoam*

hyFoam is a single-temperature solver. The switch *downgradeToSingleTemperature* should always be turned to 'yes', as shown above. *hyLight* is an option that permits to run a light version of *hyFoam*, thus saving time. The temperature bounds are here to make the code more robust during the transient phase. Please make sure that the temperature field is not reaching these bounds once the desired solution is obtained.

The *foamChemistryFile* entry tells which set of chemical rate constants is going to be used by linking to the appropriate sub-dictionary. There are various sets of chemical rate constants available and their respective meanings are listed in Table 2.

Set	Meaning
DPLR	Park's rates (used in the DPLR solver), will soon be replaced
QK	Quantum Kinetics rates
DK	Dunn and Kang rates
ES	Evans and Shexnayder (1980)
J92	Jachimoski's rates (1992)
J92_fwd	Jachimoski's rates (1992), forward only
Mars94	Park's rates (1994)
Earth93	Park's rates (1993), available soon (to replace DPLR)

Table 2: Available chemical rate constant sets in *hyFoam*

3.3.2 The *turbulenceProperties* dictionary

As stated earlier, the turbulence models (*simulationType*) available in *hyFoam* are *laminar* and *RAS2Model*.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType  laminar;

// ************************************************************************* //
```

If *RAS2Model* is chosen, then the definition of a new dictionary, ***RASProperties***, is required

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      RASProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

RAS2Model       kOmegaSST;

turbulence      off;

printCoeffs     on;

// ************************************************************************* //
```

The different choices for the *RAS2Model* entry are *laminar*, *kOmegaSST*, *kEpsilon* and *SpalartAllmaras*.

3.3.3 The ***hTCProperties*** dictionary

The high-temperature chemistry library is the high-level ***strath*** library. The *hTC2Model* entry should be consistent with the turbulent model chosen in ***turbulenceProperties***. Please turn off the *active* switch if the chemistry module does not need to be loaded. The molar-fractions are printed with the prefix *X_*, the number densities with the prefix *nD_*, the partial densities with the prefix *pD_*, and finally the partial pressures with the prefix *pP_*.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      hTCProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

hTC2Model  laminar2<rho2ChemistryHTC>;

active    true;

laminar2Coeffs
{}

chemistryOutputs
{
    molarFraction       off;
    numberDensity       off;
    partialDensity      off;
    partialPressure     off;
}

// ************************************************************************* //
```

3.3.4 The ***chemistryProperties*** dictionary

hyFoam uses finite rate chemistry. This translates into the use of the new class *Euler2Implicit*. Please turn on/off the *chemistry* switch to enable/disable chemical reactions. The *initialChemicalTimeStep* option should be available soon.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      chemistryProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

chemistryType
{
    chemistrySolver   Euler2Implicit;
    chemistryThermo   rho2;
}

chemistry           off;

initialChemicalTimeStep 1.0e-9;

// ************************************************************************* //
```

3.3.5 The *hTCReactionsDPLR/QK/DK/ES* dictionaries

QK reactions should be defined as irreversible while Park (DPLR) reactions can be considered either as reversible or irreversible. Printed below is an example of the implementation of one chemical reaction in the ***hTCReactionsDPLR*** dictionary

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      hTCReactionsDPLR;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

species
(
    //N2
    O2
    //NO
    //N2+
    //O2+
    //NO+
    N
    O
    //N+
    //O+
    //e-
);

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
/* REACTIONS NO 1 TO 15 AND 18 TO 32 -> DISSOCIATION
   REACTIONS NO 16 AND 17            -> EXCHANGE
   REACTION NO 33                    -> ELECTRON IMPACT DISSOCIATION
   REACTIONS NO 34 AND 35            -> ELECTRON IMPACT IONIZATION
   REACTIONS NO 36 TO 38             -> ASSOCIATIVE IONIZATION
   REACTIONS NO 39 TO 49             -> CHARGE EXCHANGE                    */
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

reactions
{
    // Reaction no 1
    oxygenAtomicNitrogenReaction
    {
        type     reversibleArrheniusReaction;
        reaction "O2 + N = 2O + N";
        A        1.0e19;
        beta     -1.5;
        Ta       59500;
        ni       (1e20 1e21 1e22 1e23 1e24 1e25); // local mixture number density in 1/m^3
        A0       (1.8103 0.91354 0.64183 0.55388 0.52455 0.50989);
        A1       (1.9607 2.316 2.4253 2.46 2.4715 2.4773);
        A2       (3.5716 2.2885 1.9026 1.7763 1.7342 1.7132);
        A3       (-7.3623 -6.7969 -6.6277 -6.572 -6.5534 -6.5441);
        A4       (0.083861 0.046338 0.035151 0.031445 0.030209 0.029591);
    }
}

// ************************************************************************* //
```

Please uncomment only the necessary species, here O_2, N, and O. Please uncomment the appropriate chemical reactions, for instance Reaction no 1. The *type* of the reaction specifies whether the reaction is irreversible or reversible. A, *beta*, and Ta are the standard Arrhenius coefficients. The equilibrium constant is a function of the local number density of the mixture. The lists of coefficients $A0$ to $A4$ are linearly interpolated if the local mixture number density is within the range $[1 \times 10^{20}, 1 \times 10^{25}]$ m^{-3}. If not, the interpolation is bounded.

3.3.6 The *thermoDEM* dictionary

This dictionary provides the species thermodynamic data for the different species loaded in ***hTCReactionsDPLR***. The dictionary structure is given here for the N_2 molecule.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermoDEM;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

N2
{
    specie
    {
        nMoles          1;
        molWeight       28.0134;
        particleType    2;
        charge          0;
        diameter        4.17e-10;
        dissocEnergy    3.36e7;
        omega           0.74;
        noVibTemp       1;
        noElecLevels    15;
    }
    thermodynamics
    {
        Tlow                 200;
        Thigh                40000;
        decoupledCvCoeffs    ( 1.5 1 1 0 0 0 0 );
        vibrationalList      ( 1  3371 );
        electronicList       (
                                1  0
                                3  7.223157e4
                                6  8.577863e4
                                6  8.605027e4
                                3  9.535119e4
                                1  9.805636e4
                                2  9.968268e4
                                2  1.048976e5
                                5  1.116490e5
                                1  1.225836e5
                                6  1.248857e5
                                6  1.282476e5
```

```
                            10 1.338061e5
                            6  1.404296e5
                            6  1.504959e5
                         );
    }
    transport
    {
        constant
        {
            mu          0;
        }

        SutherlandEucken
        {
            As          1.41e-6;
            Ts          111.0;
        }

        BlottnerEucken
        {
            A           2.68e-2;
            B           3.18e-1;
            C          -1.13e1;
        }

        CEA
        {
            temp  (200 1000 5000 15000);
            visco (
                    (0.62526577 -31.779652 -1640.7983  1.7454992)
                    (0.87395209  561.52222 -173948.09 -0.39335958)
                    (0.88503551  909.02171 -731290.61 -0.53503838)
                  );
            kappa (
                    (0.85439436 105.73224 -12347.848  0.47793128)
                    (0.88407146 133.57293 -11429.64   0.24417019)
                    (2.4176185  8047.7749  3105580.2 -14.517761)
                  );
        }
    }
}

// ************************************************************************* //
```

Inside the *specie* sub-dictionary, new entries have been added. The *particleType* gives the type of the particle: electron (0), atom (1), molecule (2), atom (3). The *charge* entry indicates the charge of particle: atom and molecule (0), ion (1), electron (-1). The *diameter* of the particle and its temperature coefficient of viscosity, *omega*, are also specified. The dissociation energy of the particle, named *dissocEnergy*, is given in particular for use in *hy2Foam*. The last two entries indicate the number of vibrational modes (*noVibTemp*) and electronic levels (*noElecLevels*) of the particle. Further detail about vibrational and electronic energy modes are then given into the *thermodynamics* sub-directory thanks to the *vibrationalList* and *electronicList* lists. These two lists contain a sequence of scalars grouped by pair: the degeneracy and the associated characteristic temperature for a given mode or level.
The following *decoupledCvCoeffs* list has seven elements. Their meaning is detailed in Table 3.

```
        decoupledCvCoeffs    ( 1.5 1 1 0 0 0 0 );
```

Element number	Meaning
1	translational coefficient
2	rotational coefficient
3	vibrational coefficient
4	electronic coefficient
5	electron coefficient
6	chemical enthalpy at 298.15 K
7	chemical entropy at 298.15 K (unused)

Table 3: The *decoupledCvCoeffs* list

Hence, the scalar number 4 should be set to 0 to deactivate the electronic energy mode in a simulation. The same can be repeated for the scalar number 3 to have vibrationless nitrogen.

The *transport* sub-dictionary specifies some constants depending on the transport model that has been chosen in the ***thermophysicalProperties*** dictionary. Using the *constant* transport model in combination with the following set-up in ***thermoDEM***

```
constant
{
    mu            0;
}
```

allows the user to run inviscid calculations. As and Ts are the standard coefficients to determine the viscosity from the Sutherland's law while coefficients A, B, and C coefficients appear in Blottner's formula. CEA stands for the CEA2 (Chemical Equilibrium with Applications) library from NASA Glenn Research Center.

3.3.7 The *transportProperties* dictionary

This dictionary is composed of two sections. The first one concerns the quantities used to gauge the degree of rarefaction of a gas. The characteristic length of the case scenario should be entered here. There are two mean free path models (*mfpModel*) available: *variableHardSphere* and *hardSphere*. The rarefied parameters can be printed from this dictionary. *KnGLL* is the local gradient-length Knudsen number defined as the maximum of the three local gradient-length Knudsen numbers based on the density, velocity, and temperature fields. Turning the *writeKnGLL_components* switch on and the three *KnGLL* numbers will be printed.

The second section of the ***transportProperties*** dictionary completes the information already given in ***thermophysicalProperties*** and ***thermoDEM*** about the different transport models.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

rarefiedParameters
{
    computeFieldAndBoundaries       false;
    computeMfpBoundaries            true;

    mfpModel                        variableHardSphere;
    writeMfpSpecies                 off;
    writeMfpMixture                off;

    characteristicLength           1e-5;
    writeKn_overall                off;

    writeKnGLL                     off;
    writeKnGLL_components          off;
}


transportModels
{
    viscosityAndThermalConductivityModels   "see thermophysicalProperties";
    mixingRule                     molar;
    writeViscositySpecies          off;
    writeViscosityMixture          off;
    writeThermalConducSpecies      off;
    writeThermalConducMixture      off;

    writeHeatFluxes                off;
    writeWallHeatFlux              on;

    multiSpeciesTransport          LewisNumber;
    binaryDiffusivityModel         noBinaryDiffusivityModel;
}

// ************************************************************************* //
```

To this date, there are three mixing rule implemented in *hyFoam*: a simple and generally unsuitable molar average using the keyword *molar*, and those of *Wilke* and *ArmalySutton*. The viscosity and thermal conductivity can be printed, either for each species or for the mixture. The *transportModels* sub-dictionary also prescribes the multi-species transport model used in *hyFoam*. The different possibilities are listed in Table 4.

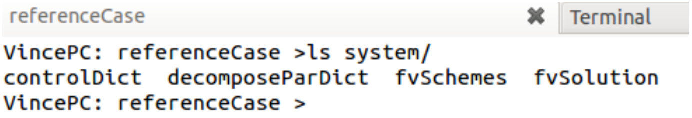
multiSpeciesTransport entry	Meaning
noSpeciesDiffusion	disables species diffusion
LewisNumber	uses the Lewis number
modifiedLewisNumber	see Gollan's PhD thesis
Fick	generalisation of Fick's binary formula for a mixture
binaryDiffusivityModel entry	Meaning
noBinaryDiffusivityModel	no binary diffusivity model
constant	constant binary diffusion coefficients (used with Fick only)
Hirschfelder	Hirschfelder model (used with Fick only)
Stephani	see K. A. Stephani's PhD thesis (used with Fick only)
Gupta	available soon, see Gupta 1989 (used with Fick only)

Table 4: Available multi-species transport models in *hy2Foam*

3.4 The *system* folder

A few modifications have been made to the ***system*** folder. They are located in ***fvSchemes*** and ***fvSolution***. Local time stepping can be switched on by simply selecting the *localEuler rDeltaT* temporal scheme in place of the traditional *Euler* scheme. Three divergence scheme entries have been added.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
```



```
VincePC: referenceCase >ls system/
controlDict  decomposeParDict  fvSchemes  fvSolution
VincePC: referenceCase >
```

Figure 4: The *system* folder

```
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

fluxScheme            Kurganov;

ddtSchemes
{
    default           Euler; // Euler or localEuler rDeltaT
}

// ************************************************************************* //
```

4 New boundary conditions

4.1 Velocity field

4.1.1 *rampInlet*

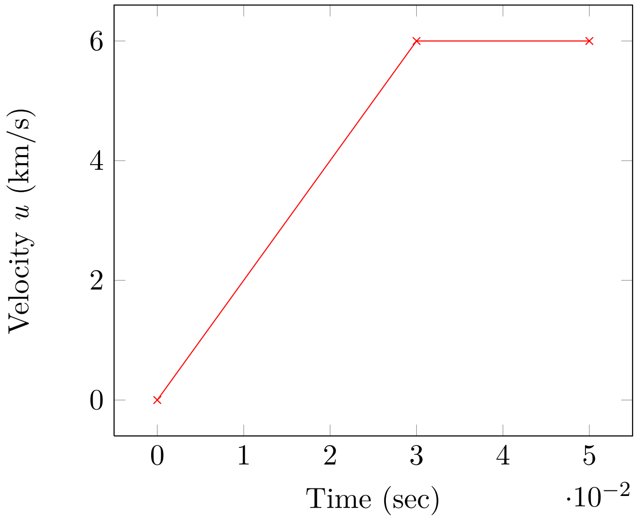
The time-dependent *rampInlet* boundary condition is aiming at reducing the very steep initial velocity gradients and thus falicitating the convergence of hypervelocity simulations. Its implementation is given hereafter

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

inlet
{
        type            rampInlet;
        refValue        uniform (1 0 0);
        offset          (0 0 0);
        amplitude       6000.0;
        tRamp           3e-2;
}

// ************************************************************************* //
```

Using the *rampInlet* boundary *type*, the inlet velocity vector is defined as follow

$$\begin{aligned} \boldsymbol{U}(t) &= (u(t),\, v(t),\, w(t))^T \\ &= \Big(amplitude \times \boldsymbol{refValue} \; - \boldsymbol{offset} \odot \boldsymbol{refValue}\Big) \\ &\qquad \times min(t/tRamp,\, 1) + \boldsymbol{offset} \odot \boldsymbol{refValue} \end{aligned} \tag{1}$$

where $\odot$ is the Hadamard product.



4.1.2 *nonEqMaxwellSlipU*

The original *maxwellSlipU* velocity boundary condition implemented in *rhoCentralFoam* is modified and takes the following form in *hy2Foam*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

wall
{
        type                nonEqMaxwellSlipU;
        Uwall               uniform (0 0 0);
        refValue            uniform (0 0 0);
        accommodationCoeff  1.0;
        valueFraction       uniform 1.0;
        thermalCreep        off;
        curvature           off;
        value               $Uwall;
}


// ************************************************************************* //
```

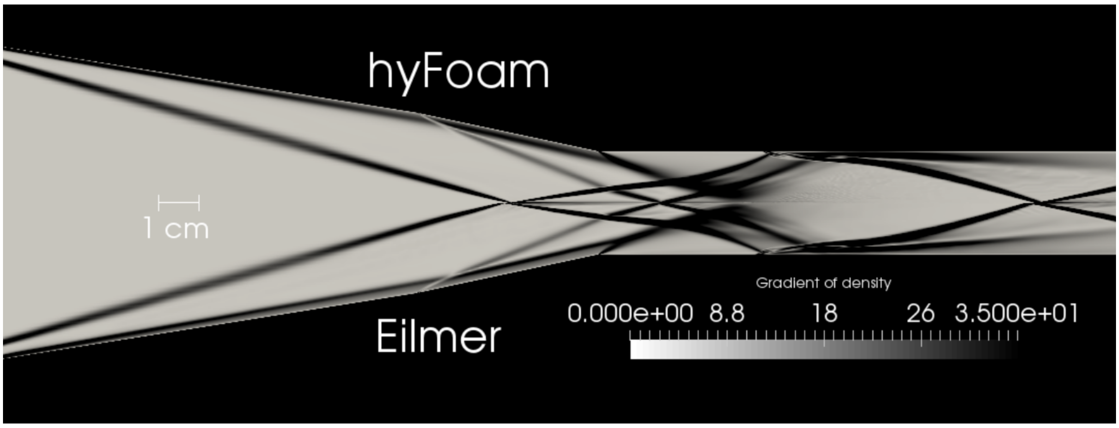
4.2 Temperature field

4.2.1 *nonEqSmoluchowskiJumpT*

The original *smoluchowskiJumpT* temperature boundary condition implemented in *rhoCentralFoam* is modified and takes the following form in *hyFoam*

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

wall
{
        type                nonEqSmoluchowskiJumpT;
        Twall               uniform 800.0;
        accommodationCoeff  1.0;
        value               $Twall;
}

// ************************************************************************* //
```

These formulations are using the overall free path boundary field.



Credit: J.-J. Hoste et al. 'Numerical Modeling and Simulation of Supersonic Flows in Propulsion Systems by Open-Source Solvers', AIAA Hypersonics, Xiamen, China, 6-9 March 2017)