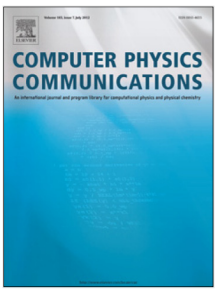
Journal Pre-proof



CoolFOAM: The CoolProp wrapper for OpenFOAM

Ettore Fadiga, Nicola Casari, Alessio Suman, Michele Pinelli

PII: S0010-4655(19)30377-7
DOI: https://doi.org/10.1016/j.cpc.2019.107047
Reference: COMPHY 107047

To appear in: *Computer Physics Communications*

Received date : 8 March 2019
Revised date : 27 September 2019
Accepted date : 11 November 2019

Please cite this article as: E. Fadiga, N. Casari, A. Suman et al., CoolFOAM: The CoolProp wrapper for OpenFOAM, *Computer Physics Communications* (2019), doi: https://doi.org/10.1016/j.cpc.2019.107047.

CoolFOAM: the CoolProp wrapper for OpenFOAM

Ettore Fadiga*, Nicola Casari, Alessio Suman, Michele Pinelli

Department of Engineering, University of Ferrara, Ferrara, Italy

Abstract

Nowadays, the evaluation of the transport and thermodynamic properties in computational fluid dynamics (CFD) analysis is based on common real gas equations (such as Redlich-Kwong and Peng-Robinson equations of state) and polynomial models. Such quantities are not always accurate: very large pressure ranges, working conditions close to the critical point or phase change could introduce an error during the computation of the thermophysical properties of interest. Moreover, the computational effort of numerical simulations is strongly affected by the evaluation of such properties. In light of these considerations, and to extend the modeling capabilities of CFD software suites, a fluid thermophysical property library can be included in the computation of the quantities required for the solution of the flow field. The focus of this work is the coupling of an open source CFD tool (OpenFOAM) with the open source CoolProp library. The proposed wrapper is intended to provide a connection between the most popular open-source thermophysical library and one of the most used open-source CFD software. CoolFOAM extends the thermophysical modelling possibilities of OpenFOAM. The formal implementation of this library follows the coding standards of the CFD suite.

Keywords: CoolProp, OpenFOAM, Helmholtz, 64.10.+h, 47.11.-j

1. Introduction

The evaluation of thermophysical properties of working fluids is a fundamental step in computational fluid dynamics (CFD) simulations. For example, it is quite common to operate with compressible fluids that exhibit real gas behavior: in this case, cubic equations of state (EoS) are frequently used to predict fluids properties in both single-phase and two-phase region. These EoS represent a good compromise between the simplicity of the formulation and the uncertainty in the determination of such quantities. Cubic EoS are frequently adopted to characterize phase equilibria. Moreover, for these kind of equations, different approaches to model mixtures exist.[1] Nevertheless, the critical region is an area characterized by significant deviations from experimental data, because the cubic models do not account for critical phenomena. Furthermore, these models lack accuracy in the calculation of thermodynamic properties, in particular at dense homogeneous states.[2] These kind of inaccuracy sources can be relevant during the numerical simulation of systems operating in different fields, such as refrigeration (e.g., vapor compression cycles), cogeneration (e.g., organic Rankine cycles) and energy production (e.g., supercritical CO_2 cycles) [3, 4, 5, 6, 7, 8].

Recently, more accurate models for the thermodynamic properties calculation have been developed: modern equations are often formulated in terms of the reduced Helmholtz energy, which is split up into one ideal gas part and a second residual part. The first one defines the nature of the hypothetical ideal gas at given values of temperature and density, while the second one characterizes the residual behavior of the real fluid. The resulting equation of state is explicit in temperature and density normalized by the fluid's critical values. [9]. Multi-parameter equations of state can characterize experimental measurements in Helmholtz-explicit relations. Using these equations of state, thermodynamic properties can be then calculated using thermodynamic relationships, through the differen-

*Corresponding author

Email address: ettore.fadiga@unife.it (Ettore Fadiga)

tiation of the equations. The range of applicability of these models includes the liquid phase, differently from the cubic equations, and the behavior in the critical region is represented with a reasonable error [10].

There are various libraries and databases that have implemented Helmholtz-explicit equations of state for different fluids [11, 12, 13, 14]. In addition to thermodynamic models, high accuracy formulations for transport properties are included in these utilities. The state of the art of transport properties modeling is less mature than the thermodynamic properties one: one of the causes is that an accurate formulation of thermodynamic properties is needed to develop a high-accuracy transport properties model. For this reason, a natural delay between the publication of an equation of state and the related transport properties model occurs [12]. The CoolProp library represents the most used library in the open-source scenario for the evaluation of thermo-physical properties [12]. The code is written in C++, and several high-level and low-level interfaces have been developed to couple CoolProp with different software (e.g. Labview, Microsoft Excel, MATLAB).

This article concerns the development of a wrapper of CoolProp for OpenFOAM. OpenFOAM [15] is a widely used open-source C++ toolbox for the numerical simulation of a wide series of CFD problems. This package contains several libraries and applications to pre-process, solve and post-process cases from different fields of application (e.g, incompressible flow, compressible flow, multiphase flow and conjugate heat transfer problems). The structure of OpenFOAM enables users to modify existing solvers and libraries, or to create new ones with a reasonable effort. As a consequence, a great number of personalized solvers and libraries have been developed by the scientific community. The exponential growth of this open-source software's capabilities in modeling real world problems is recognizable in several fields. The range of applications varies from incompressible flows with heat transfer [16] to density-based solvers for low and high Mach number [17]. Moreover, wall modelling in LES simulations [18] and mass-transfer in solid oxide fuel cells [19] are recently treated in literature works. In such a wide framework, the CoolFOAM wrapper intends to fill a gap in the OpenFOAM capabilities of modelling the thermophysical properties. Similar tools have already been developed in the field of commercial CFD software (e.g., the link between ANSYS fluent and REFPROP [11]). This work extends the applicability of one of the most widespread open-source CFD software, eliminating this gap.

The need for an extension of the OpenFOAM capabilities in modeling the thermophysical properties has been satisfied with the possibility of obtaining directly the required values from the CoolProp library. A C++ shared library has been written, starting from the OpenFOAM templates. The library has been developed to facilitate its use in a similar fashion with two of the OpenFOAM solvers, *rhoPimpleFoam* and *rhoSimpleFoam*, which are a transient and steady state solver for compressible flow cases, respectively. Alternatively, other solvers within OpenFOAM have to be slightly modified to adapt them to the new thermo-physical properties package. The conformity of the library to the OpenFOAM coding standards allows easy adoption of the new feature in a wide series of cases. Once the new library and applications are compiled, the user only needs to add the correct textual inputs in a configuration file to use the CoolProp library.

2. Theory

2.1. OpenFOAM thermophysical models

The computation of thermophysical properties in an OpenFOAM simulation is based on a pressure-temperature system. These variables are the independent variables, from which other properties are determined. Frequently, the user has to evaluate thermophysical properties in applications where the ideal gas approximation is not usable with reasonable accuracy. In such cases, the most accurate model for the evaluation of the fluid density is the Peng-Robinson (PR) equation of state, one of the most common cubic models:

$$p = \frac{RT}{\nu - b} - \frac{a_c \alpha(T_R, \omega)}{\nu(\nu + b) + b(\nu - b)} \tag{1}$$

with the parameters:

$$a_c = 0.45724 \left(\frac{R^2 T_C^2}{P_C} \right)$$

$$b = 0.07780 \left(\frac{R T_C}{P_C} \right)$$

$$\alpha(T_R, \omega) = \left[1 + m(\omega) \left(1 - \sqrt{T_R} \right) \right]^2$$

$$m(\omega) = 0.37464 + 1.54226\omega - 0.26992\omega^2$$

where ν is the molar volume, T_R is the reduced temperature, ω is the acentric factor, R is the gas constant, and T_C and P_C are the critical values of temperature and pressure, respectively. PR is quite accurate in different conditions, but for saturated liquid densities, conditions close to critical point and caloric properties in the homogeneous region the error increases [20, 21]. Nevertheless, PR and other cubic equations of state are frequently adopted for the evaluation of vapor pressures and equilibrium-phase compositions of mixtures. In such cases, these equations of state need less computational resources than multiparameter equations of state [2], yielding relatively accurate results. In CFD analysis, the computational overhead has to be taken into account, evaluating different fluid property models on a case-by-case basis. Cubic EOS may represent an effective trade-off between accuracy and computational load, as pointed out by Abdelli et al. [22].

It is also possible to adopt a polynomial model, where the density trend is fitted with constant pressure and the temperature as the independent variable. Polynomial functions inherently lack accuracy in the presence of wide pressure ranges and phase change, as well as density trends difficult to shape. Polynomials are usable also for the specific heat and transport properties (viscosity, thermal conductivity and thermal diffusivity). Thermodynamic properties, which are derived from the specific heat, can be also evaluated from the JANAF tables of thermodynamics [23]. In addition to that, the Sutherland model for the transport properties calculation is available. Again, wide pressure ranges and complex trends of the variables can be significant sources of inaccuracy.

2.2. CoolProp thermophysical models

The determination of thermodynamic properties of all the fluids included in the CoolProp library is based on equations of state explicit in the Helmholtz energy. This formulation is a common denominator for all the high-accuracy equations of state currently available in the literature. In this formulation the sum of a residual part (α^r) and of an ideal gas part (α^0) gives the expression for the nondimensionalized Helmholtz-energy:

$$\alpha = \alpha^0 + \alpha^r \tag{2}$$

The most fascinating aspect of this model is the fact that all the other thermodynamic properties can be found through analytic derivatives of the two just mentioned terms. As an example, the equation for the pressure is:

$$Z = \frac{p}{\rho RT} = 1 + \delta \left(\frac{\partial \alpha^r}{\partial \delta} \right)_\tau \tag{3}$$

where Z is the compressibility factor, p is the pressure, R is the mass specific gas constant, ρ is the density, T is the temperature, δ is the reduced density given by $\delta = \rho/\rho_{red}$ and the reciprocal reduced temperature is $\tau = T_{red}/T$. Generally the reducing variables ρ_{red} and T_{red} correspond to the critical values. For some fluids, such as R134a, these reducing parameters have to be determined during the fitting process. The pressure example is followed by several other derivatives, which can be found in literature works of authors like Thorade and Sadat [24], Span [25] and Lemmon [26]. The independent variables of the Helmholtz-energy equations are temperature and density. If there is the availability of different state variables, it is necessary to use numerical solvers in order to calculate the independent variables. Various state variables inputs are analyzed in Span [27], and a solver for enthalpy/entropy is included in Coolprop.

For the transport properties the state of the art is more confused. There is a higher quantity of methods available in the literature for the determination of these properties. It is possible to find high-accuracy methodologies for some fluids, but for others the possibilities are definitely scarce. The viscosity is usually divided in two terms: the first considers the viscosity in relation to temperature in the condition of dilute-gas, while the second includes the temperature and density-dependent residual viscosity:

$$\eta = \eta^{(0)}(\tau) + \eta^{(r)}(\tau, \delta) \tag{4}$$

The critical enhancement of viscosity represents a divergence of transport properties of fluids close to the critical point [28]. Usually this phenomenon is not taken into account, but for a small number of fluids it is considerable. This is possible if enough information about viscosity in the critical region is provided. In CoolProp the only fluids having this phenomenon modeled are water [29] and carbon dioxide [30]. On the contrary, for the thermal conductivity the critical enhancement term is not negligible in areas distant from the critical point. For this reason, three terms are considered for the thermal conductivity:

$$\lambda = \lambda^{(0)}(\tau) + \lambda^{(r)}(\tau, \delta) + +\lambda^{(c)}(\tau, \delta) \tag{5}$$

Some of the fluids included in CoolProp do not have high-accuracy correlations for the transport properties. For these less-studied fluids it is necessary to find a method which can substitute these correlations: one of the most used choices is the extended corresponding states (ECS) methodology. It allows for the obtainment of the properties for the fluid of interest starting from a well-known reference fluid: this reference should have a similar p-v-T surface and an accurate estimation of transport properties. The extended corresponding states method proposed in CoolProp follows the work proposed by Huber et al. [31], which has already been implemented in REFPROP[11].

3. The CoolProp-OpenFOAM wrapper

3.1. OpenFOAM thermophysical classes

The core of the OpenFOAM code is a large library containing its basic capabilities. In addition to differential equations, tensor operations and other basic functionality, dynamic meshing and several physical models have been implemented. The library is the starting point for the development of applications, which can be distinguished into two categories: solvers and utilities. Solvers carry on the calculations in order to deal with a wide range of problems: starting from the simple potential flow solver (*potentialFoam*), it is possible to use dynamic mesh (*icoDyMFoam*) solvers and compressible transient solvers, such as *rhoPimpleFoam*. More complex applications are conjugate heat transfer solvers, like *chtMultiRegionFoam*, and supersonic turbulent (*sonicTurbFoam*) solvers. The operations different from actually solving the case (meshing, pre-processing, post-processing, etc.) are performed by the utilities.

OpenFOAM has two main thermophysical model classes for fixed composition fluids: the first is based on density, *rhoThermo*, and the other on compressibility, *psiThermo*. Every solver that needs thermophysical properties constructs an object of one of these classes.

The rhoThermo class is constituted by two source files, *rhoThermo.C* and *rhoThermos.C*, and one header file, *rhoThermo.H*. The header file is included in the related OpenFOAM solvers in order to have access to the functions needed to determine the thermodynamic properties. Three main objects are created in this class: the density *rho*, the compressibility *psi* and the dynamic viscosity *mu*. This model calculates the basic thermodynamic properties in relation to density variation of the fluid. It is applied particularly to heat transfer cases, where changes of temperature imply variations of density. The psiThermo class has basically the same structure as the one described above (*psiThermo.C*, *psiThermos.C* and *psiThermo.H*). The main difference is related to the density definition: this model is based on compressibility, which gives the density if multiplied by pressure. Moreover, there is a *fluidThermo* model which makes it possible to choose the thermophysical models at run-time. The objects of this class are currently constructed only by the *rhoSimpleFoam* and *rhoPimpleFoam* solvers.

3.2. The CoolProp library

More than one hundred equations of state and transport properties correlations for pure and pseudo-pure fluids are stored in the C++ CoolProp library. Moreover, this library implements mixtures, incompressible fluids and brines properties, high accuracy psychrometric routines and cubic equations of state. The code is based on an abstract base class (AbstractState) which delineates a protocol that must be implemented by the property backends. It is possible to have access to the fluid properties through two different interfaces of CoolProp:

- The high-level interface allows for the obtainment of the required fluid property with a simple call to the *PropsSI* function. As an example, the line of code for specific heat at constant pressure ("C") of R134a at temperature ("T") and pressure ("P") of respectively 273.15 K and 2 MPa is:

```
Cp = PropsSI('C','T',273.15,'P',2e06,'R134a')
```

- The low-level interface provides access to deeper levels of the CoolProp code. This solution is definitely faster: actually, the low-level interface is always internally called by the high-level one. The low-level interface operates with enumerated values and floating point numbers, avoiding the strings usage. This obviously increases the efficiency and speeds-up the execution. In order to maximize the efficiency of the low-level interface, it is necessary to instantiate an instance of the backend for each fluid, calling then methods within the instance. Calling the constructor for the backend instance is not

computationally insignificant, so it is better to only do it one time.

3.3. The CoolFOAM library

The main contribution of this work is a new C++ shared library for the determination of thermophysical properties in OpenFOAM: **libCoolPropThermophysical.so**. It contains two new thermophysical model classes for fixed composition fluids, *psiThermoCool* and *rhoThermoCool*. These new classes are very similar to the OpenFOAM's original ones, described in section 2.1. The main difference is the inclusion of three new classes, the first containing the new equation of state, the second for the thermodynamic properties and the last one for the transport properties.

The *rhoCoolProp* class allows the evaluation of the fluid density directly from the CoolProp library: firstly, a shared pointer *rhoFluid* to a new-allocated instance of one of the AbstractState backends is instantiated. The density value is returned by an appropriate member function, which updates initially the pressure and the temperature of the instance, and then calculates the requested property.

Thermodynamic and transport properties are evaluated respectively as objects of the *hCoolProp* and *CoolPropTransport* classes. The basic principle of these classes is the same as the *rhoCoolProp* class, described above.

For what concerns the numerical simulations of two-phase flows, OpenFOAM handles the thermophysical properties of the two phases separately. For this reason, the calculation of such properties inside the saturation dome has not been implemented in the CoolFOAM library.

4. Compiling and using the library

The compilation of the CoolFOAM library does not require additional operations than the ones required for OpenFOAM standard libraries. The only action required is building the CoolProp database as a shared library, in order to be able to link it with OpenFOAM. In order to correctly perform the compilation, it is necessary to include the CoolProp shared library and all the directories related to it. Detailed instructions for the installation of the product of this work are included in a text file attached to the library itself, which is available upon request. The present work has been developed for OpenFOAM 6, but it is simply adaptable to different versions of the software.

Listing 1 thermophysicalProperties

```
thermoType
{
type                    hePsiThermo;
mixture                 pureMixture;
transport               CoolProp;
thermo                  hCoolProp;
equationOfState         rhoCoolProp;
specie                  specie;
energy                  sensibleEnthalpy;
}

mixture
{
    equationOfState
    {
        Fluid           "R134a";
        EOS             "HEOS";
    }

    specie
    {
        nMoles          1;
        molWeight       102.03;
    }

    thermodynamics
    {
        Fluid           "R134a";
        EOS             "HEOS";
        Hf              0;
        Sf              0;
    }

    equationOfState
    {
        Fluid           "R134a";
        EOS             "HEOS";
    }
}
```

In order to use the wrapper, the new thermophysical models have to be added to OpenFOAM solvers. It is necessary to compile the new applications including the new library, with opportune and simple modifications to some files. More precisely, it is necessary to replace the references to the original classes (e.g., *rhoThermo*, *psiThermo*) with the references to the new ones (*rhoThermoCool* and *psiThermoCool*). Only the solvers with

the generalized *fluidThermo* class (*rhoPimpleFoam* and *rhoSimpleFoam*) can work as they are. In this case, it is only necessary to include the correct thermophysical library in the *controlDict*, a setup file required in every OpenFOAM case.

The CoolFOAM library definitely enhances the user-friendliness of OpenFOAM during the set up of the thermophysical models. The only action needed is the specification of the fluid name and of the equation of state, checking the CoolProp database to select between the different possibilities, according to the library terminology. A small portion of the thermophysicalProperties dictionary, which is the setup file for thermophysical properties in an OpenFOAM case, is reported in listing 1.

The original OpenFOAM models, such as the polynomials or the Peng-Robinson equation, require several coefficients that have to be fitted from curves or sourced in literature, and then inserted in the dictionary. Furthermore, the PR equation implemented in OpenFOAM is not able to calculate density for liquids.

The computational time of the CoolFOAM library has been deeply analyzed for the R134a fluid. The tests have been conducted in a pressure range from 0.1 bar to 2000 bar and in a temperature range from 215 K to 415 K. The results of this investigation are reported in figure 1.The duration of a single call to the CoolFOAM library for the computation of the density has been normalized to the maximum value of 35 µs. Such call has to be done for each cell of the computational domain and for each time step of the numerical simulation. The highest computational effort appears to be in the region close to the critical point and in the nearby of the critical isothermal line. On the contrary, the fastest calculation are performed in the very right of the pressure-enthalpy diagram. The results here presented are relative to the low-level interface of CoolProp. Tests with the high-level interface have shown a computational effort 10-30 times higher. Therefore, the first approach has been implemented in CoolFOAM.

A single call to the CoolProp library for the evaluation of density requires a time approximately twenty times bigger than the PR equation. The polynomial models calculate the density values five times faster if compared to PR. When running simulations, the physical duration of a computational time step using the CoolFOAM library is increased from 500% to 1500% with respect to polynomial models. The variability of the duration is related

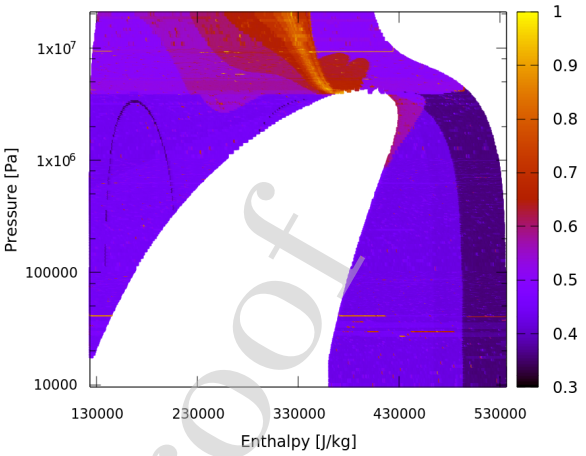


Figure 1: Computational time: density of R134a.

to the solver employed. The best way to run a case with the new thermophysical models is to initialize the solution with a first simulation using OpenFOAM native models, switching then to the CoolProp library for having the best accuracy. Next steps in this track will regard the implementation of CoolProp lookup tables, in order to reduce the computational effort related to the library.

5. Validation

This section contains three validation cases based on experimental studies available in literature. The first case concerns the investigation of natural convection of air in a heat cavity, originally performed by Betts and Bokhari [32]. The other test cases have been carried out on two different nozzles at the Test Rig for Organic VApors (TROVA) of Politecnico di Milano [33].

5.1. Buoyant cavity

The first comparison is based on the experimental investigation of the turbulent buoyant flow of air in a rectangular box-shaped cavity. The turbulent nature of the flow depends on the geometric proportions of the cavity, as well as the Rayleigh number. The cavity internal geometry is schematized in figure 2: the thickness (W) of the box is 0.076 m, the height (H) and the width (D) are respectively 0.52 and 2.18 m. A temperature difference of 19.2 K is maintained between the two y-z surfaces, while the remaining walls are considered as adiabatic. The computational domain has been discretized with a

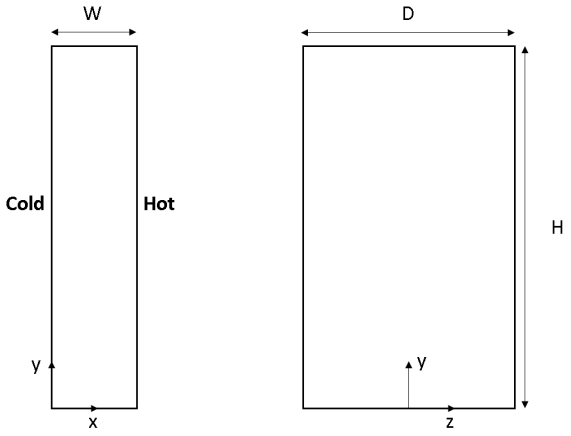


Figure 2: Buoyant cavity: geometry scheme.

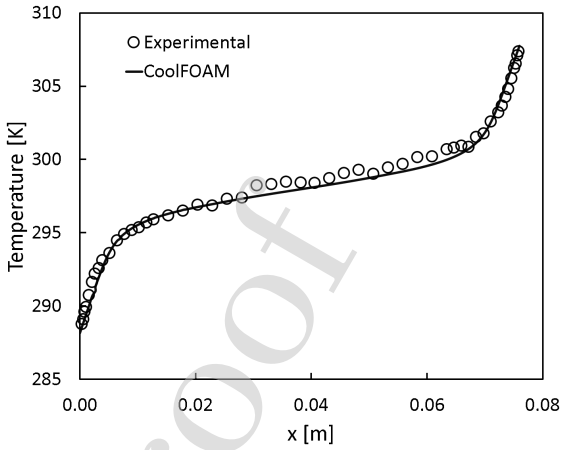


Figure 3: Buoyant cavity: temperature trends at $y/H = 0.5$.

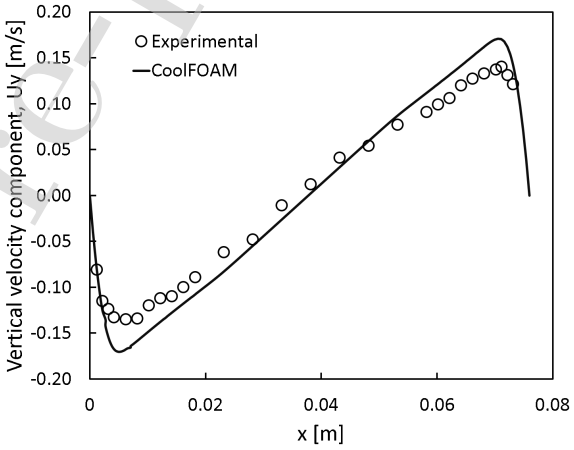


Figure 4: Buoyant cavity: vertical velocity trends at $y/H = 0.5$.

structured mesh, taking advantage of the rectangular shape of the system. After a grid convergence analysis, the simulation has been carried out with approximately 80,000 cells.

The ideal gas approximation is more than tenable for the operative temperature and pressure ranges investigated in this first analysis. Moreover, the specific heat, viscosity and thermal conductivity maximum variations are less than 5% in these conditions. For these reasons, the results obtained with the new library are very similar to the ones calculated with the original OpenFOAM thermophysical models. In fact, the absolute variations of temperature and vertical velocity, U_y, are in the order of 0.01 K and $1 \times 10^{-4}\,\mathrm{m\,s^{-1}}$ respectively. As a consequence, it has been chosen to include in the following graphs only the solutions obtained with the CoolFOAM library.
In particular, the figures 3 and 4 display the trends of temperature and vertical velocity in the x direction at $y/H = 0.5$, compared with experimental values.

The purpose of this case is to demonstrate the correct implementation of the CoolProp functionalities in OpenFOAM, using a verification case widely used in the community.

5.2. Convergent-divergent nozzle

The first experimental case of the TROVA test rig regards the expansion of octamethyltrisiloxane (MDM) in a convergent-divergent nozzle. The work by Spinelli et al. investigates two nozzles, characterized by downstream Mach numbers close to 1.5 and 2. The first nozzle's results are considered in this section: particularly, the case with upstream compressibility Z of approximately 0.81 is the object of the numerical analysis. The considered gas is then strongly non-ideal: this allows the full demonstration of the CoolFOAM interface potentialities. The experimental data available include static pressure measurements and Mach number evaluations. The pressure has been detected through pressure transducers connected to up to nine pressure taps located on the nozzle symmetry axis. The Mach number has been directly evaluated through the estimation of the Mach waves slope, identified from

schlieren images.

The computational domain has been discretized with a structured two-dimensional mesh, represented in figure 5. The independence of the solution to the dimension of the mesh elements has been reached with approximately 60,000 cells. The good quality of the mesh, characterized by a non-orthogonality of less than 30° and by a skewness minor of 0.3, has allowed the adoption of accurate numerical schemes. More precisely, the simulation, performed with the *rhoPimpleFoam* solver, has been executed with a second order discretization accuracy for the advective and Laplacian terms. Since

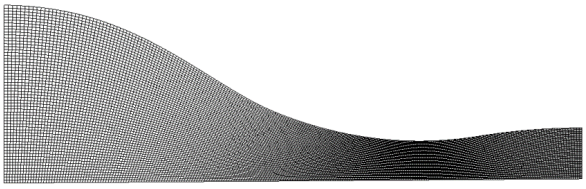


Figure 5: Convergent-divergent nozzle: computational domain.

the fluid reaches supersonic velocities, there is no need of boundary conditions at the outlet section. At the inlet, a total pressure of 4.59 bar and a total temperature of 512.6 K have been imposed. The solution has been firstly initialized with ideal gas conditions, switching then to the Helmholtz equation of state. After this first step, the hypothesis of inviscid flow has been considered valid for this kind of fluid, characterized by a very low viscosity. The simulation with zero viscosity and laminar approximation has produced very good results when compared to the experimental data, as shown in figure 6

Looking for a higher level of accuracy, the numerical analysis has been improved with polynomial correlations for the viscosity and the thermal conductivity. The reason of adopting polynomials instead of the CoolProp interface stands in the absence of correlations for these properties of the MDM fluid in the CoolProp library. In addition to that, a turbulence k-epsilon model has been implemented, in order to obtain a better representation of the real case's physics. The pressure trend along the symmetry axis for the turbulent-viscous simulation fits better the experimental values, as represented in figure 7. It has to be noticed that the inviscid simulation's results are still fairly close to

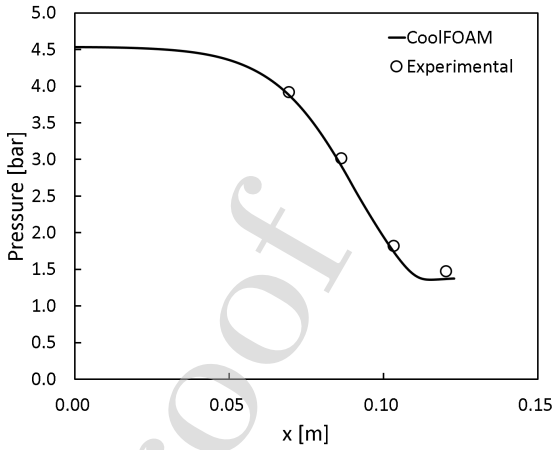


Figure 6: Convergent-divergent nozzle: pressure values on the symmetry axis, inviscid flow.

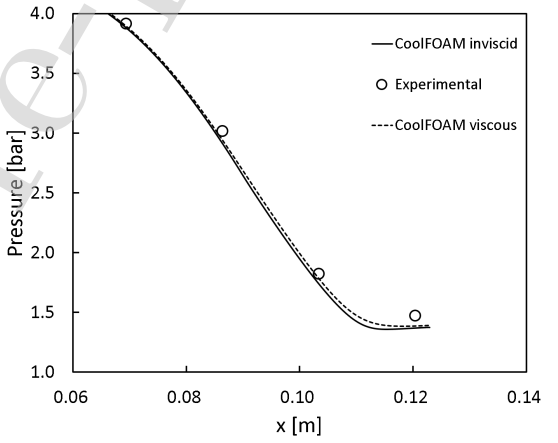


Figure 7: Convergent-divergent nozzle: pressure values on the symmetry axis, viscous and inviscid flow.

the experimental values: the possibility to adopt this hypothesis remains a valid possibility for the siloxane fluid analyzed, if there is the will of reducing the general level of complexity of the simulation.

The results of the viscous-turbulent simulation have been compared also with the experimental evaluation of the Mach number along the symmetry axis. The graph represented in figure 8 represents this comparison in the region close to the outlet section, where experimental data are available.

A very high accuracy can be noticed in the final part of the nozzle, where the Mach number values are well replicated by the numerical anal-

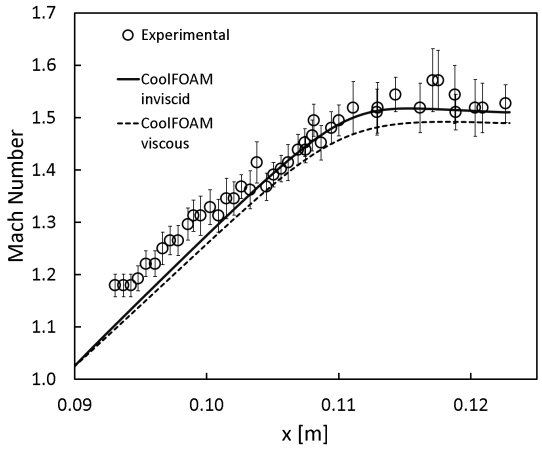


Figure 8: Convergent-divergent nozzle: Mach number values on the symmetry axis, viscous and inviscid flow.

ysis. For lower values of the x coordinate, both the viscous and inviscid models underestimate the dynamic contribution of the flow, if compared to experimental evidences. It is interesting to notice how the inviscid model better fits these references: the dissipating action of viscous effect has a reducing effect on the Mach number.

5.3. Compressible turbulent flow over a Backward-Facing Step

The last validation case reported in this paper is based on a nozzle similar to the one described in section 5.2. The main difference is the presence of a backward facing step in correspondence of the throat section of the duct. The consequence is the generation of an oblique shock at the reattachment point after the flow separation originated by the wall discontinuity. This phenomenon and the deriving shock-waves system is highlighted in the schlieren image of the flow: the complex behavior of the flow has been extensively described in [34]. A detail of the mesh close to the discontinuity has been represented in figure 9. The structured grid has been refined in proximity of the shock-waves system according to the experimental reference. In addition, the cells faces have been aligned with the shocks, trying to capture the first pressure discontinuity as accurately as possible. This approach has allowed the maintenance of a very good quality of the mesh with a reasonable number of elements (less than 100,000).

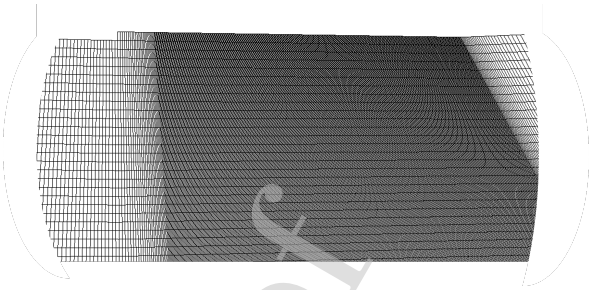


Figure 9: Backward Facing Step: detail of the computational domain.

The boundary conditions required for this CFD analysis are the inlet total pressure and total temperature, respectively of 4.58 bar and 520.1 K. These conditions correspond to a compressibility factor of 0.82 at the inlet section. The outlet conditions are not required, because of the univocal behavior of the supersonic flow.

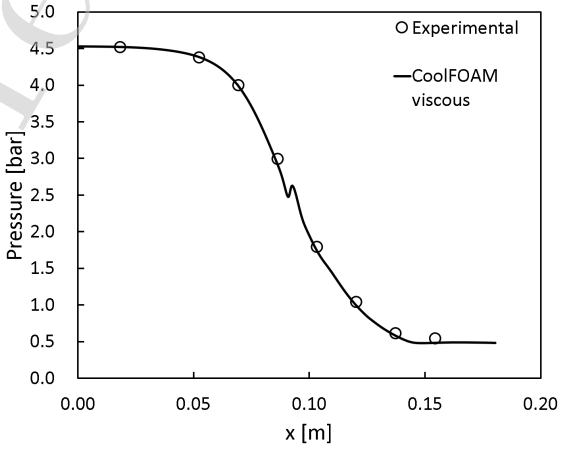


Figure 10: Backward Facing Step: pressure values on the symmetry axis, viscous flow.

The pressure trend along the symmetry axis gives a very accurate representation of the experimental values, as represented in figure 10. The shock is captured with a reasonable accuracy from the viscous numerical simulation. In this case, no experimental data for the Mach number have been released, but the results of the numerical simulation are shown in figure 11. It is interesting to notice how the design Mach number at the outlet (approximately 2.0) has been confirmed from the CFD analysis.

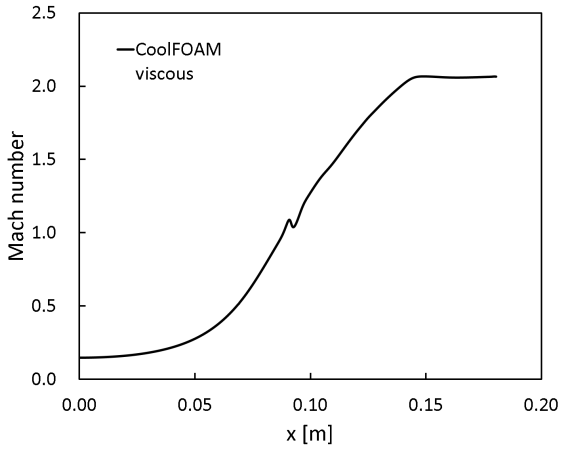


Figure 11: Backward Facing Step: Mach number values on the symmetry axis, viscous flow.

6. Conclusions

A new library for the determination of thermophysical properties in OpenFOAM has been developed. The work presented in this article is based on the direct evaluation of the fluid properties through the open-source C++ library CoolProp. Moreover, one of the OpenFOAM solvers, *buoyantSimpleFoam* has been slightly modified in order to include the CoolFOAM library. The final result, *buoyantSimpleFoamCP*, is an example of how an OpenFOAM solver can be adapted to use the main product of this paper.

The consistency of the developed models has been compared with experimental data for three validation cases. For the first test case, regarding a buoyant cavity, a typical OpenFOAM verification case has been solved in order to check the correct implementation of the two models. In the second and third cases, compressible flows with strongly non ideal behavior have been investigated. The results obtained from the numerical analysis present a very good level of agreement with the experimental references.

7. Acknowledgments

The research was partially supported by the Italian Ministry of Economic Development within the framework of the Program Agreement MSE-CNR "Micro co/tri generazione di Bioenergia Efficiente e Stabile (Mi-Best)".

References

[1] M.-J. Huron, J. Vidal, New mixing rules in simple equations of state for representing vapour-liquid equilibria of strongly non-ideal mixtures, Fluid Phase Equilibria 3 (4) (1979) 255 – 271. doi:https://doi.org/10.1016/0378-3812(79)80001-1.
URL http://www.sciencedirect.com/science/article/pii/0378381279800011

[2] J. Gernert, A. Jäger, R. Span, Calculation of phase equilibria for multi-component mixtures using highly accurate helmholtz energy equations of state, Fluid Phase Equilibria 375 (2014) 209 – 218. doi:https://doi.org/10.1016/j.fluid.2014.05.012.
URL http://www.sciencedirect.com/science/article/pii/S0378381214002817

[3] N. Casari, A. Suman, D. Ziviani, M. Van Den Broek, M. De Paepe, M. Pinelli, Computational models for the analysis of positive displacement machines: Real gas and dynamic mesh, Energy Procedia 129 (2017) 411–418. doi:10.1016/j.egypro.2017.09.124.

[4] N. Casari, A. Suman, M. Morini, M. Pinelli, Real gas expansion with dynamic mesh in common positive displacement machines, Energy Procedia 129 (2017) 248–255. doi:10.1016/j.egypro.2017.09.150.

[5] Y. Bartosiewicz, Z. Aidoun, Y. Mercadier, Numerical assessment of ejector operation for refrigeration applications based on cfd, Applied Thermal Engineering 26 (5-6) (2006) 604–612. doi:10.1016/j.applthermaleng.2005.07.003.

[6] J. Honra, M. Berana, L. Danao, M. Manuel, Cfd analysis of supersonic ejector in ejector refrigeration system for air conditioning application, Lecture Notes in Engineering and Computer Science 2230 (2017) 1019–1024.

[7] A. Ameli, A. Uusitalo, T. Turunen-Saaresti, J. Backman, Numerical sensitivity analysis for supercritical co2 radial turbine performance and flow field, Energy Procedia 129 (2017) 1117–1124. doi:10.1016/j.egypro.2017.09.233.

[8] N. Baltadjiev, C. Lettieri, Z. Spakovszky, An investigation of real gas effects in supercritical co2 centrifugal compressors, Journal of Turbomachinery 137 (9). doi:10.1115/1.4029616.

[9] R. Span, W. Wagner, E. Lemmon, R. Jacobsen, Multiparameter equations of state - recent trends and future challenges, Fluid Phase Equilibria 183-184 (2001) 1–20. doi:10.1016/S0378-3812(01)00416-2.

[10] A. Goodwin, J. Sengers, C. Peters, R. S. of Chemistry (Great Britain), I. U. of Pure, A. C. Physical, B. C. Division, I. A. of Chemical Thermodynamics, Applied Thermodynamics of Fluids, Royal Society of Chemistry, 2010.

[11] E. W. Lemmon, I. Bell, M. L. Huber, M. O. McLinden, NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0, National Institute of Standards and Technology (2018).
URL https://www.nist.gov/srd/refprop

[12] I. Bell, J. Wronski, S. Quoilin, V. Lemort, Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop, Industrial and Engineering Chemistry Research 53 (6) (2014) 2498–2508. doi:10.1021/ie4033999.

[13] W. Wagner, Fluidcal, http://www.thermo.rub.de/de/prof-w-wagner/software/fluidcal.html.
[14] S. Klein, Engineering Equation Solver; F-Chart Software (2010).
[15] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics 12 (6) (1998) 620. doi:10.1063/1.168744.
[16] V. D'Alessandro, L. Binci, S. Montelpare, R. Ricci, On the development of openfoam solvers based on explicit and implicit high-order runge–kutta schemes for incompressible flows with heat transfer, Computer Physics Communications 222 (2018) 14–30. doi:10.1016/j.cpc.2017.09.009.
[17] C. Shen, F. Sun, X. Xia, Implementation of density-based solver for all speeds in the framework of openfoam, Computer Physics Communications 185 (10) (2014) 2730–2741. doi:10.1016/j.cpc.2014.06.009.
[18] T. Mukha, S. Rezaeiravesh, M. Liefvendahl, A library for wall-modelled large-eddy simulation based on openfoam technology, Computer Physics Communicationsdoi:10.1016/j.cpc.2019.01.016.
[19] V. Novaresio, M. García-Camprubí, S. Izquierdo, P. Asinari, N. Fueyo, An open-source library for the numerical modeling of mass-transfer in solid oxide fuel cells, Computer Physics Communications 183 (1) (2012) 125–146. doi:10.1016/j.cpc.2011.08.003.
[20] O. Kunz, W. Wagner, The gerg-2008 wide-range equation of state for natural gases and other mixtures: An expansion of gerg-2004, Journal of Chemical & Engineering Data 57 (11) (2012) 3032–3091. doi:10.1021/je300655b.
[21] J. Valderrama, The state of the cubic equations of state, Industrial and Engineering Chemistry Research 42 (8) (2003) 1603–1618. doi:10.1021/ie020447b.
[22] J. D. Lazhar Abdelli, J. Vierendeels, Cfd analysis of an expansion process using different real gas models., Master of Science in Electromechanical Engineering.
URL https://lib.ugent.be/catalog/rug01:002224377
[23] T. Allison, Janaf thermochemical tables, nist standard reference database 13 (1996). doi:10.18434/T42S31.
URL http://kinetics.nist.gov/janaf/
[24] M. Thorade, A. Saadat, Partial derivatives of thermodynamic state properties for dynamic simulation, Environmental Earth Sciences 70 (8) (2013) 3497–3503. doi:10.1007/s12665-013-2394-z.
[25] R. Span, Multiparameter Equations of State, Springer, 2000. doi:10.1007/978-3-662-04092-8.
[26] E. Lemmon, R. Jacobsen, S. Penoncello, D. Friend, Thermodynamic properties of air and mixtures of nitrogen, argon, and oxygen from 60 to 2000 k at pressures to 2000 mpa, Journal of Physical and Chemical Reference Data 29 (3) (2000) 331–362.
[27] R. Span, E. Lemmon, R. Jacobsen, W. Wagner, A. Yokozeki, A reference equation of state for the thermodynamic properties of nitrogen for temperatures from 63.151 to 1000 k and pressures to 2200 mpa, Journal of Physical and Chemical Reference Data 29 (6) (2000) 1361–1401.
[28] J. Sengers, Transport properties of fluids near critical points, International Journal of Thermophysics 6 (3) (1985) 203–232, cited By 155. doi:10.1007/BF00522145.
[29] M. L. Huber, R. A. Perkins, A. Laesecke, D. G. Friend, J. V. Sengers, M. J. Assael, I. N. Metaxa, E. Vogel, R. Mareš, K. Miyagawa, New international formulation for the viscosity of h2o, Journal of Physical and Chemical Reference Data 38 (2) (2009) 101–125. doi:10.1063/1.3088050.
[30] S. Bock, E. Bich, E. Vogel, A. S. Dickinson, V. Vesovic, Calculation of the transport properties of carbon dioxide. III. volume viscosity, depolarized rayleigh scattering, and nuclear spin relaxation, The Journal of Chemical Physics 121 (9) (2004) 4117–4122. doi:10.1063/1.1778384.
[31] M. L. Huber, A. Laesecke, R. A. Perkins, Model for the viscosity and thermal conductivity of refrigerants, including a new correlation for the viscosity of r134a, Industrial & Engineering Chemistry Research 42 (13) (2003) 3163–3178. doi:10.1021/ie0300880.
[32] P. Betts, I. Bokhari, Experiments on turbulent natural convection in an enclosed tall cavity, International Journal of Heat and Fluid Flow 21 (6) (2000) 675–683. doi:10.1016/S0142-727X(00)00033-3.
[33] A. Spinelli, G. Cammi, S. Gallarini, M. Zocca, F. Cozzi, P. Gaetani, V. Dossena, A. Guardone, Experimental evidence of non-ideal compressible effects in expanding flow of a high molecular complexity vapor, Experiments in Fluids 59 (8). doi:10.1007/s00348-018-2578-0.
[34] G. Gori, M. Zocca, G. Cammi, A. Spinelli, A. Guardone, Experimental assessment of the open-source su2 cfd suite for orc applications, Energy Procedia 129 (2017) 256–263. doi:10.1016/j.egypro.2017.09.151.

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: