OpenFOAM implementation of multi-temperature thermochemical non-equilibrium models for high-enthalpy air

Marco Allanda Daniele Bergamaschi
Lorenzo Esposito Francesco Faggion
Francesco Grassi Giorgio Venezia
Francesco Virgulti

January 2026

1 Introduction

OpenFOAM is an open-source software widely used for computational fluid dynamics (CFD) simulations. Owing to its significant impact in both academia and industry, a deep understanding of its structure and functionalities is essential. Beyond simply using existing solvers and test cases, it is equally important to be able to adapt and extend the code to address different physical models and application scenarios.

The objective of this project is to simulate a two-temperature open-source CFD model for hypersonic reacting flows (e.g., air). This work is based on the reference paper [1], which serves as the theoretical and methodological foundation for the model implementation.

The report is structured as follows:

- analysis of the underlying physical model in the one-temperature formulation,
- development of the two-temperature model,
- implementation of the model within the OpenFOAM framework.

2 Navier–Stokes Equations

The motion of a Newtonian fluid is governed by the conservation of mass, momentum, and energy. [2]

2.1 Continuity Equation (Mass Conservation)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1}$$

2.2 Momentum Equation

$$\frac{\partial(\rho\mathbf{u})}{\partial t}+\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})=-\nabla p+\nabla\cdot\tau+\rho\mathbf{f} \tag{2}$$

where the viscous stress tensor for a Newtonian fluid is given by

$$\tau=\mu\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I} \tag{3}$$

2.3 Energy Equation

The conservation of total energy for a compressible Newtonian fluid can be written as

$$\frac{\partial(\rho E)}{\partial t}+\nabla\cdot[\mathbf{u}(\rho E+p)]=\nabla\cdot(k\nabla T)+\nabla\cdot(\tau\cdot\mathbf{u})+\rho\mathbf{f}\cdot\mathbf{u}, \tag{4}$$

where the total energy per unit mass is defined as

$$E=e+\frac{1}{2}|\mathbf{u}|^2, \tag{5}$$

with e denoting the specific internal energy.

For many engineering applications, it is convenient to express the energy equation in terms of temperature or enthalpy rather than total energy. Introducing the specific heat at constant pressure c_p, defined as

$$c_p=\left(\frac{\partial h}{\partial T}\right)_p, \tag{6}$$

where $h=e+p/\rho$ is the specific enthalpy, the energy equation can be reformulated, under the assumption of a calorically perfect fluid, as

$$\rho c_p\left(\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T\right)=\nabla\cdot(k\nabla T)+\Phi-\frac{Dp}{Dt}. \tag{7}$$

Here, T is the temperature, k is the thermal conductivity, $\Phi=\tau:\nabla\mathbf{u}$ represents the viscous dissipation term, and D/Dt denotes the material derivative.

The specific heat at constant pressure c_p represents the amount of energy required to increase the temperature of a unit mass of fluid by one degree while maintaining constant pressure. Its appearance in the temperature-based energy equation reflects the fact that, in a flowing fluid, thermal energy variations are naturally described in terms of enthalpy rather than internal energy alone. For incompressible flows with constant properties, the pressure-work term $-Dp/Dt$ vanishes, and the energy equation reduces to a pure convection–diffusion equation for the temperature field.

2.4 Equation of State

To close the system, an equation of state is required, for example for an ideal gas:

$$p=\rho RT \tag{8}$$

2.5 Multi-Species Flow Formulation

When moving from a single-component fluid to a multi-species mixture (for example, air described by eleven chemical species), the underlying physical principles remain unchanged. The conservation of mass, momentum, and energy is still governed by the same equations introduced in the previous sections. However, the presence of multiple chemical species introduces additional transport equations and increases the level of coupling between thermodynamic and flow variables.

Physical interpretation Even in a multi-species mixture, the flow is characterized by a single macroscopic velocity field $\mathbf{u}$ and a single pressure field p. The mixture behaves as a single continuum from a mechanical point of view. The complexity arises from the fact that the mixture is composed of several chemical species, indexed by $i = 1, 2, \ldots, 11$, each characterized by its mass fraction Y_i, molecular weight M_i, and diffusive motion relative to the mixture-averaged velocity.

Governing equations The conservation equations for total mass, momentum, and energy retain exactly the same form as in the single-species case and apply to the mixture as a whole. Momentum conservation is therefore not written for each individual species, but for the mixture treated as a single fluid.

In contrast to the single-species formulation, the thermodynamic and transport properties appearing in these equations (such as density, specific heat, and viscosity) generally depend on both temperature and mixture composition.

Species transport equations The main extension introduced by a multi-species formulation is the inclusion of additional transport equations, one for each chemical species. For species i, the transport equation describes the evolution of its mass fraction Y_i and accounts for convection with the bulk flow, molecular diffusion, and chemical reactions:

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y_i) = -\nabla \cdot \mathbf{J}_i + \dot{\omega}_i. \tag{9}$$

Here, $\mathbf{J}_i$ denotes the diffusive flux of species i relative to the mixture, while $\dot{\omega}_i$ represents the chemical source term associated with production or consumption due to chemical reactions.

For a system composed of eleven species, eleven such equations are introduced. However, the species mass fractions are constrained by

$$\sum_{i=1}^{11} Y_i = 1, \tag{10}$$

which implies that only ten of the species equations are independent. The remaining mass fraction can be obtained from this constraint.

Role of the energy equation In multi-species flows, the energy equation generally plays a central role even at low Mach number. This is due to the explicit dependence of thermodynamic properties on both temperature and composition, for example

$$c_p = c_p(T, Y_i).$$

In addition, energy is transported not only by convection and conduction, but also by species diffusion, and chemical reactions may release or absorb heat. As a result, the governing variables become strongly coupled, which can be summarized schematically as

$$Y_i \leftrightarrow T \leftrightarrow \rho \leftrightarrow p \leftrightarrow \mathbf{u}.$$

This coupling highlights the essential role of species transport and energy conservation in determining the overall behavior of multi-species flows.

3 Two-temperature reacting mixture

In a two-temperature reacting mixture [1], the continuity equation (1), the momentum equation (2), and the species transport equations (9) retain the same form as in the single-temperature formulation. The main modification concerns the definition of the internal energy e appearing in the total energy equation (4).

3.1 Internal energy decomposition

Instead of a single-temperature caloric relation $e = e(T, Y)$, the internal energy is decomposed into contributions associated with different molecular energy modes:

$$e(T, T_v, Y) = e_{tr}(T, Y) + e_{rot}(T, Y) + e_{vib}(T_v, Y) + e_{elec}(T_v, Y). \tag{11}$$

In this work we group the translational and rotational modes into a single translational–rotational energy e_t, and the vibrational and electronic modes into a vibrational–electronic energy e_v:

$$e_t(T, Y) = e_{tr}(T, Y) + e_{rot}(T, Y), \tag{12}$$

$$e_v(T_v, Y) = e_{vib}(T_v, Y) + e_{elec}(T_v, Y), \tag{13}$$

so that

$$e = e_t + e_v. \tag{14}$$

3.2 Canonical two-temperature energy equations

A common two-temperature closure consists of solving a conservative total energy equation together with an additional balance for the vibrational–electronic energy. The total energy equation reads

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot \big(\mathbf{u}(\rho E + p)\big) = \nabla \cdot (\tau \cdot \mathbf{u}) - \nabla \cdot \mathbf{q} + \rho \mathbf{f} \cdot \mathbf{u}, \tag{15}$$

with

$$E = e + \frac{1}{2}|\mathbf{u}|^2. \tag{16}$$

In addition, a separate balance is introduced for the vibrational–electronic energy,

$$\frac{\partial(\rho e_v)}{\partial t} + \nabla \cdot (\rho \mathbf{u} e_v) = -\nabla \cdot \mathbf{q}_v + Q_{tr\leftrightarrow v} + Q_{chem\rightarrow v}. \tag{17}$$

3.3 Simplification of the system for a single-cell test

To reproduce the zero-dimensional test case considered in [1], we simplify the governing equations as follows. The computational domain consists of a single cell, therefore spatial transport terms (fluxes) are neglected. Moreover, the velocity is initially set to zero, $\mathbf{u} = \mathbf{0}$. Under these assumptions,

- the total energy reduces to the internal energy,

  $$E = e, \tag{18}$$

  since $\mathbf{u} = \mathbf{0}$ implies $\frac{1}{2}|\mathbf{u}|^2 = 0$;

- the vibrational–electronic energy equation (17) reduces to a purely local evolution equation (ODE),

  $$\frac{de_v}{dt} = Q_{tr\leftrightarrow v} + Q_{chem\rightarrow v}, \tag{19}$$

  i.e. only relaxation and chemistry coupling terms are retained, with no convection or diffusion of e_v.

3.4 Species

In the present work, air is modeled as a reacting mixture composed of five chemical species (air_5). Consequently, five species mass fractions Y_i, $i = 1, \ldots, 5$, are considered, subject to the constraint

$$\sum_{i=1}^{5} Y_i = 1. \tag{20}$$

4 Test case and solver implementation details

4.1 Test case description

The numerical test case considered in this work is based on the OpenFOAM tutorial `test/chemistry/air11`. Although the folder name refers to an eleven-species air model, only five chemical species are effectively considered in the present configuration. The number of active species is defined through the files located in the `0/` directory of the case. It was initially suspected that renaming the folder could break the case setup; however, this is not the case, and the folder name does not affect the execution of the solver.

4.2 Solver

The solver employed is `ShockThermo`, which is derived from `ShockFluid`. At each time iteration, the solver executes several predictor and corrector routines. For the purposes of the present study, the only method of interest is `thermophysicalPredictor()`, which is responsible for updating the thermodynamic state of the flow.

4.3 General idea of the test

The test case consists of a single computational cell with the initial velocity set to $\mathbf{U} = \mathbf{0}$. As a consequence, transport effects and the momentum equation are not relevant for this study. The objective is not to simulate a flow, but rather to repeatedly call `solver.thermophysicalPredictor()` in order to update the thermodynamic variables, in particular the translational temperature T and the vibrational temperature T_v, according to the two-temperature model.

4.4 Key part of `thermophysicalPredictor`

The most important operation performed inside `thermophysicalPredictor()` is the call to

```
thermo_.correct();
```

which appears between lines 129 and 138 of the solver source code.

In an ideal implementation, the solver should not require any modification. The intended design is that, if a custom thermodynamics model is instantiated (e.g. `highEnthalpy MulticomponentThermo`), a call to `thermo_.correct()` should automatically dispatch the appropriate overridden method through polymorphism.

However, this mechanism does not work as expected in the present case. The class `highEnthalpyMulticomponentThermo` is derived from `PsiThermo`, and the `correct()` method that is effectively invoked corresponds to the base class `PsiThermo`, rather than the two-temperature implementation required for this work.

4.5 Workaround implemented in `ShockThermo`

To resolve this issue, the solver `ShockThermo` was modified by introducing an additional pointer:

```
highEnthalpyMulticomponentThermo* heThermoPtr_;
```

In the constructor of `ShockThermo`, this pointer is initialized if the selected thermodynamics model corresponds to `highEnthalpyMulticomponentThermo`; otherwise, it is set to `nullptr`. Inside `thermophysicalPredictor()`, the thermodynamic update is then handled as follows:

```
if (heThermoPtr_ == nullptr)
{
    thermo_.correct();
}
else
{
    heThermoPtr_->correct_he();
}
```

This approach guarantees that the correct thermodynamic update method is invoked when the two-temperature high-enthalpy model is active, while preserving the standard behavior for all other thermodynamics models.

4.6 OpenFOAM state variables

At each iteration, OpenFOAM provides the following state variables:

- $\mathbf{U}$: velocity field, where $\mathbf{U}[i]$ denotes the velocity of cell i;
- p: pressure field, where $p[i]$ denotes the pressure of cell i;
- Y_i: species mass fractions, where $Y_i[j]$ denotes the mass fraction of species i in cell j;
- T: translational (transport) temperature, where $T[i]$ denotes the temperature of cell i.

The number of species is defined through the case setup and, in the present study, is equal to five.

4.7 Additional variables for the two-temperature model

In order to reproduce the reference results presented in [1], additional thermodynamic variables are required:

- T_v: vibrational temperature;
- e_t: translational–rotational energy;
- e_v: vibrational–electronic energy.

These variables are created and initialized in the constructor of `highEnthalpy MulticomponentThermo`.

4.8 Mutation++ interface

The class `highEnthalpyMulticomponentThermo` relies on the Mutation++ library to model thermochemical nonequilibrium effects. Mutation++ requires precise information about the chemical species involved in the mixture. This configuration is handled in the constructor of `highEnthalpyMulticomponentThermo` (lines 119–124 of the source code).

It is critical that the species ordering is consistent between OpenFOAM and Mutation++. For example, if the species indexed as $Y[0]$ in OpenFOAM corresponds to atomic oxygen, then atomic oxygen must also be the first species in the Mutation++ mixture. This consistency is enforced in lines 128–143 of the constructor. The variables e_t and e_v are also initialized at this stage.

4.9 Sending the thermodynamic state to Mutation++

Mutation++ requires the complete thermodynamic state of the mixture, including:

$$\mathbf{U}, \quad p, \quad Y_i, \quad T, \quad T_v, \quad e_t, \quad e_v.$$

These quantities are copied into temporary arrays for safety and passed to Mutation++ inside the method `correct_he()` (lines 199–355 of the source code). The core call to Mutation++ is

```
mutationMixPtr_->step(
    dtSolver,
    rho[celli],
    Y_mut,
    Et_local,
    Ev_local,
    Ttr,
    Tv
);
```

This function advances the translational and vibrational temperatures according to the two-temperature relaxation model described in [1].

4.10 Governing equations solved in the present configuration

In the present single-cell configuration, the solver advances a conservative total energy equation of the form

$$\frac{\partial(\rho E)}{\partial t} = Q_{\text{chem}}, \tag{21}$$

where

$$E = e = e_t + e_v, \tag{22}$$

and Q_{chem} denotes the chemical energy source term. The vibrational energy is not transported spatially, but is updated locally through a relaxation model,

$$\frac{de_v}{dt} = Q_{tr\leftrightarrow v} + Q_{chem\rightarrow v}, \tag{23}$$

which is evaluated during the thermodynamic correction step using Mutation++.

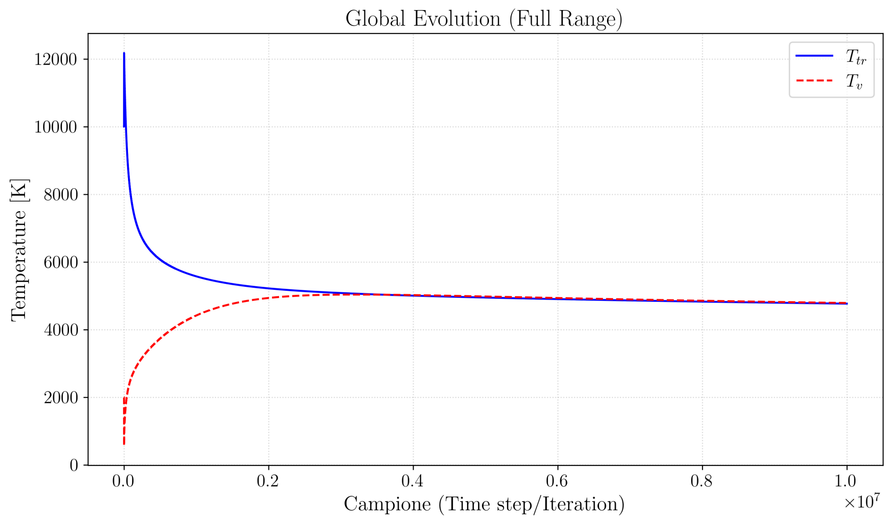

Figure 1: T_t and T_v over the time

5 Results

5.1 T_t vs T_v

Figure 1 shows the temporal evolution of the translational–rotational temperature T_t and the vibrational temperature T_v for the single-cell test case. The initial conditions are set to $T_t = 12000$ K and $T_v = 2000$ K, corresponding to a strongly non-equilibrium thermodynamic state.

As expected from the physics of thermochemical nonequilibrium, energy is transferred from the translational–rotational mode to the vibrational–electronic mode through vibration–translation (VT) relaxation. As a result, T_t decreases rapidly, while T_v increases, until the two temperatures approach each other. After the initial transient, both temperatures converge toward a common equilibrium value, indicating the progressive restoration of thermal equilibrium between the energy modes. Figure 2 provides a zoomed view of the early-time evolution, highlighting the region where the two temperatures intersect. This intersection corresponds to the point at which the vibrational–electronic energy has absorbed a sufficient amount of energy from the translational–rotational mode to balance the mode populations. Beyond this point, the relaxation process continues smoothly until both temperatures asymptotically converge to the same value.

The observed behavior is fully consistent with the two-temperature relaxation model described in [1] and confirms the correct coupling between the total energy equation and the local vibrational energy relaxation implemented through the Mutation++ interface. In particular, the absence of spatial transport terms ensures that the evolution of T_t and T_v is governed solely by chemical energy exchange and mode relaxation, as

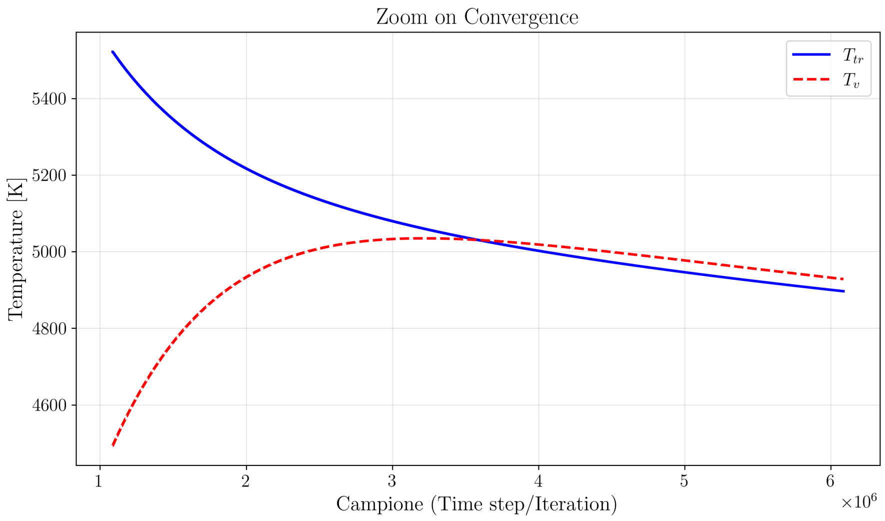

Figure 2: Area of intersection between two temperatures

intended for the present zero-dimensional configuration. .

5.2 Multi-cell speedup analysis

We also extended our analysis to a multi-cell configuration to evaluate the computational performance of the implemented two-temperature model. The test case consists of a three-dimensional domain discretized into a grid of $10 \times 10 \times 10$ cells, so that a total of 1000 cells are simulated simultaneously by using MPI. Each cell is initialized with the same non-equilibrium thermodynamic state as in the single-cell test case, with $T_t = 12000$ K and $T_v = 2000$ K, so the result is *expected to be the same as before for each cell*. The delta time used for the simulation is 1×10^{-8} s, and the end simulation time is 1×10^{-5} s, so that we keep the temporal resolution consistent with the single-cell test case. Figure 3 illustrates the time taken to complete the simulation as a function of the number of MPI processes used. The results demonstrate a significant speedup as the number of processes increases, until we reach a plateau around 4 processes, thus with 8 processes the time taken is almost the same as with 4 processes. This behavior is expected due to the overhead associated with MPI communication and the limited workload per process as the number of processes increases. The simulation has been run on a native Linux machine with heterogeneous core architecture, in particular with 4 performance cores and 8 efficiency cores, thus the last run has been compromised by this hardware limitation. We also implemented a OpenMP parallelization for the `correct_he()` method, but the results were not satisfactory. The only significant speedup has been achieved with just one core, while with more than one core the time taken increased with respect to the only MPI parallelization, making OpenMP not beneficial for this specific impelmentation. Nevertheless, the results confirm that the

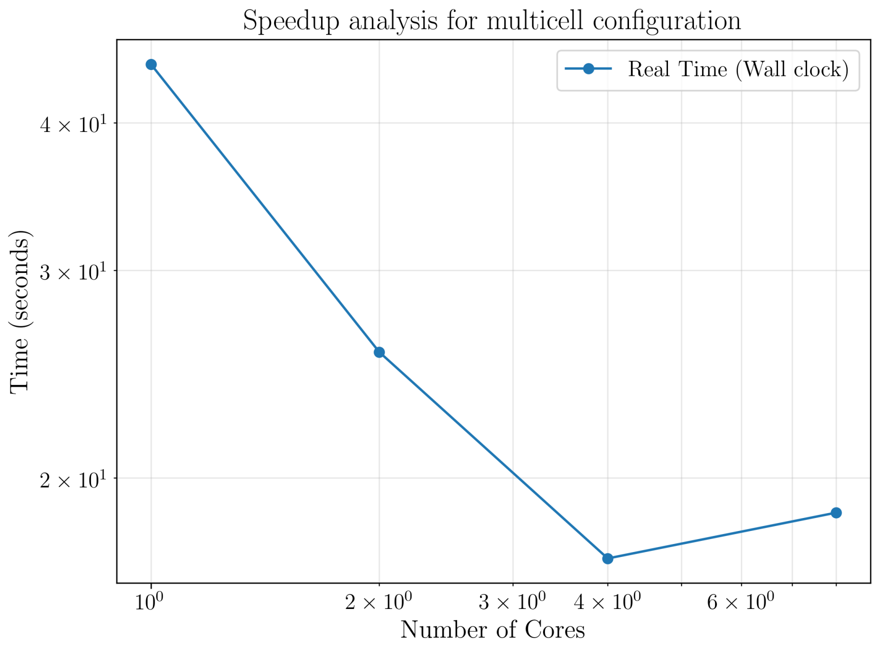

Figure 3: Speedup analysis for multi-cell configuration

implemented two-temperature model scales well with the number of MPI processes, making it suitable for larger-scale simulations involving multiple cells.

5.3 Comparison with reference results

We compared our results with the reference data provided in [1] to validate the accuracy of our implementation. The temperatuere profiles obtained from our OpenFOAM implementation show excellent agreement with the reference results, confirming that the two-temperature relaxation dynamics are accurately captured. Both the initial rapid decrease of T_t and the corresponding increase of T_v match closely with the reference curves, demonstrating that the energy exchange between the translational–rotational and vibrational–electronic modes is correctly modeled. We also compared the evolution of species mass fractions over time as presented in Figure 4. The evolution of

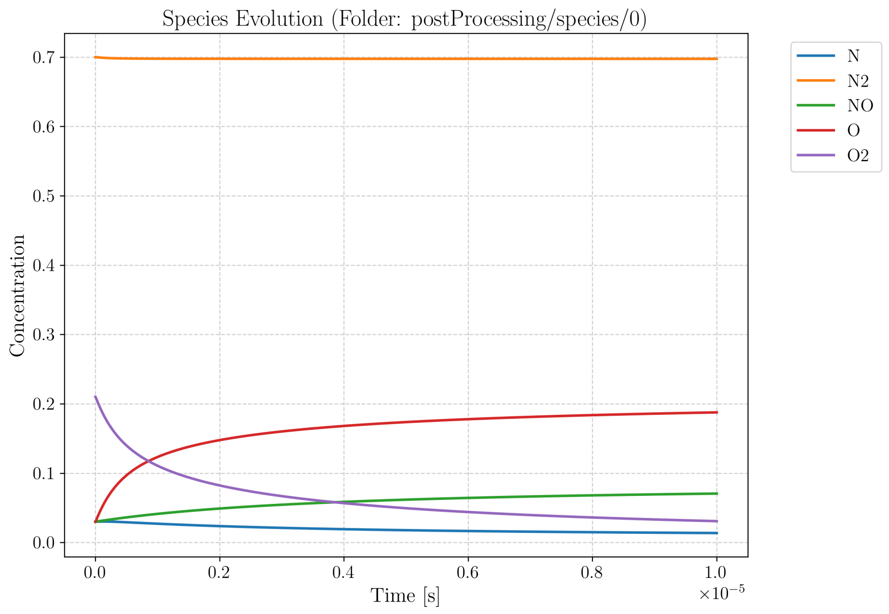

Figure 4: Species mass fractions over time

key species such as O, O_2, and N_2 also aligns well with the reference data. The consumption of O_2 and the production of atomic O are accurately reproduced, indicating that the chemical kinetics are properly integrated with the two-temperature thermodynamics. Overall, the comparison with reference results validates the fidelity of our OpenFOAM implementation of the two-temperature thermochemical nonequilibrium model.

6 Conclusions

This report presents the implementation of a two-temperature thermochemical nonequilibrium model for high-enthalpy air within the OpenFOAM framework. The model is based on the theoretical formulation described in [1] and leverages the Mutation++ library for accurate thermodynamic and kinetic calculations. We successfully implemented teh singel cell test case, demonstrating the correct relaxation, and then we extended the analysis to a multi-isolated-cell configuration to evaluate computational speedup provided by the OpenFOAM framework through MPI parallelization.

References

[1] Vincent Casseau, Rodrigo C. Palharini, Thomas J. Scanlon, and Richard E. Brown. A two-temperature open-source cfd model for hypersonic reacting flows, part one: Zero-dimensional analysis. *Physics of Fluids*, 28(6):–, 2016.

[2] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson, 2nd edition, 2007.