POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

High-Performance CFD Development with OpenFOAM

Course:
High Performance Scientific Computing in Aerospace Engineering

Group ID: 1

Group Members:
Ettore Cirillo
Mattia Gotti
Giulio Martella
Michele Milani
Stefano Pedretti
Daniele Piano
Federico Pinto

Advisors:
Prof. Federico Piscaglia

Academic year:
2025-2026

Contents

1. Introduction

The second part of the *High Performance Scientific Computing in Aerospace* course focuses on development inside a mature, production-ready CFD framework, rather than writing a solver from scratch. The goal is to understand OpenFOAM's software architecture and learn how to extend it in a controlled, maintainable, and reproducible way, with attention to both physics (modelling) and engineering aspects (build system, dynamic linking, code quality, performance).
OpenFOAM implements the CFD pipeline using the finite-volume method (FVM), segregated solution strategies, and a set of thermophysical libraries providing the equation of state and the properties required by the solver. In this project we do not aim to rewrite this infrastructure, but to extend it toward high-enthalpy, non-equilibrium modelling typical of hypersonic applications.
Our specific objectives are:

- understand how OpenFOAM organises and loads thermophysical libraries, shared objects and runtime registration of templated classes;
- develop a user-side library without modifying the system installation and dynamically link it to OpenFOAM test applications and, ultimately, to existing solvers;
- design a clean interface toward external thermodynamic backends (in our case `Mutation++`), so that OpenFOAM can query advanced thermodynamic models without coupling the entire solver to external implementation details.

A key constraint, also highlighted in discussions with the instructor, is to avoid a local "one-off" prototype. OpenFOAM evolves over time, runs on clusters, and is validated over years. Therefore, we deliberately avoid modifying the OpenFOAM core and instead develop the extension as a user-side dynamically linked library, ensuring portability and long-term maintainability.

2. Problem analysis

High-enthalpy flows are encountered in several aerospace applications of practical relevance, such as atmospheric re-entry of space vehicles, hypersonic cruise flight, and scramjet propulsion systems. In these regimes, strong shock waves convert a significant portion of the kinetic energy of the flow into internal energy, leading to extremely high temperatures in the post-shock region.
Under such conditions, the gas cannot be described as a simple thermodynamic medium. Molecular internal degrees of freedom become progressively excited, energy exchange processes occur over finite time scales, and chemical reactions are strongly activated. An accurate representation of these phenomena is essential for predicting quantities of engineering interest, like aerodynamic forces, heat fluxes to thermal protection systems, and shock layer structure.
Despite this, most industrial and academic CFD solvers have historically been developed for low-to-moderate enthalpy flows, where the assumption of local thermal equilibrium is acceptable. When extending an existing framework such as OpenFOAM toward hypersonic applications, it is therefore necessary to critically assess the validity of standard thermodynamic models and identify their limitations.

2.1. Limitations of single-temperature models

Classical compressible flow solvers assume that all molecular energy modes (i.e., translational, rotational, vibrational, and electronic) are characterized by a single temperature. This assumption of **local thermal equilibrium** is valid only when internal energy exchanges occur much faster than the characteristic flow time scales.
In high-enthalpy hypersonic flows, this condition is no longer satisfied. Vibrational and electronic energy modes may relax much more slowly than translational and rotational modes, resulting in a pronounced state of thermal non-equilibrium. Enforcing a single temperature under these conditions leads to a physically inconsistent redistribution of energy among the internal modes.
The consequences of this approximation include:

- an inaccurate evaluation of thermodynamic properties such as internal energy, specific heats, and speed of sound;
- an incorrect coupling between thermodynamics and finite-rate chemistry, since reaction rates are highly sensitive to the non-equilibrium state of the gas;
- potential numerical stiffness and loss of robustness in the solution process.

Therefore, the limitations of single-temperature models are not only related to physical fidelity, but also directly affect the stability and reliability of CFD simulations in the hypersonic regime.

2.2. Handling hypersonic flows: the two-temperature model

To overcome the shortcomings of equilibrium models while keeping the computational complexity under control, an extended thermodynamic description is required. A widely adopted compromise is the **two-temperature model**, which introduces two distinct temperatures to characterize the thermodynamic state of the gas.
In this framework:

- a **translational-rotational temperature** governs pressure, density, and momentum coupling;
- a **vibrational-electronic temperature** accounts for the delayed excitation of internal modes and their interaction with chemical reactions.

Energy exchange between the two subsystems is modeled through relaxation source terms, allowing the system to naturally evolve toward equilibrium when the physical conditions permit. This formulation represents the minimal extension of classical single-temperature models capable of capturing the dominant non-equilibrium effects encountered in hypersonic flows.
An important advantage of the two-temperature approach is its compatibility with segregated finite-volume solvers. This makes it particularly suitable for integration within the existing thermophysical architecture of OpenFOAM, especially when coupled with specialized external libraries for high-temperature thermodynamics and chemistry.

3. Development process

To address the limitations detailed in the previous section, a hybrid computational strategy is adopted by coupling the finite volume solver with the `Mutation++` library. The chapter details the software engineering approach and the numerical formulation of the physical models, concluding with the proposed methodology for embedding the thermochemical logic into the transport equations.

3.1. Object-Oriented Programming in OpenFOAM

To implement such two-temperature model and the associated chemical relaxation processes, it is necessary to understand the underlying software architecture of OpenFOAM. This section outlines the structural features of the library, the development methodology adopted to ensure code maintainability, and the specific class expansions required for hypersonic flows.

3.1.1 Template architecture and polymorphism

OpenFOAM is designed as a distinct C++ library that prioritizes abstraction and flexibility. The core of its architecture relies heavily on *template* classes. This feature allows the software to define generic types and algorithms, enabling the implementation of mathematical equations.
Furthermore, the solver structure is built upon *polymorphism* and *run-time selection*. OpenFOAM does not hardcode physical models into the solver. Instead, it uses a generic base class interface where the specific model to be used is selected at run-time. This mechanism is fundamental for our work, as it allows us to inject new custom thermodynamic models into the standard solver without modifying its legacy source code.

3.1.2 Development methodology

A professional development approach was adopted to ensure the robustness and maintainability of the project [2]. Modifying the central installation directory of OpenFOAM is a bad practice, as it breaks dependency chains, complicates updates, and prevents multi-user usage.
Instead, we utilized a "detached" workflow, operating entirely in the *user-space*. This methodology involves:

- **Core engine:** The original OpenFOAM installation remains untouched.
- **Workspace:** The custom code is developed in a separate directory structure that links dynamically to the core libraries.

This approach offers significant advantages regarding code maintainability. Since our custom libraries are decoupled from the core source, migrating to a newer version of OpenFOAM typically requires only minor adjustments to the compilation instructions, rather than a complete rewrite of the code.

3.1.3 Code navigation and Class identification via Doxygen

Due to the extensive use of templates and the size of OpenFOAM, navigating the source code manually is inefficient and nearly impossible. To understand the class hierarchy and identify the correct points for expansion, we utilized the official Doxygen documentation.
By analyzing the inheritance trees, we identified the specific classes that required to be extended in order to support hypersonic flows:

- **Thermodynamic coefficients:** We analyzed the `janafThermo` class, which uses standard tables for equilibrium properties.
- **Fluid thermodynamics:** We examined the `hePsiThermo` family to understand how internal energy and enthalpy are managed, identifying where to integrate the two-temperature handling.

3.1.4 Library expansion and Dynamic linking

The final objective of this development phase is to implement a two-temperature model for hypersonic simulations. To achieve this, two main custom classes were developed and linked to the main solver:

1. **RRHO:** Standard OpenFOAM thermodynamics relies on JANAF polynomials, which are valid for standard combustion but insufficient for high-temperature, non-equilibrium flows. To address this, we introduced a new class, `RRHO`, structurally derived from the standard JANAF implementation.
2. **HighEnthalpyMulticomponentThermo:** Standard OpenFOAM solvers transport a single temperature field (T). To simulate thermal non-equilibrium, we want to extend the basic thermodynamic library. This class acts as a container and manager: it introduces the second temperature field (T_{ve} for vibrational-electronic energy) and manages the coupling between the fluid dynamics and the finite-rate chemistry provided by the `Mutation++` library.

These extensions are integrated using dynamic linking. The compilation process is managed by the `wmake` utility. Within the `Make` directory of our source folder, the `files` document specifies the compilation target location (the user's library directory), while the `options` document instructs the compiler to link against the standard OpenFOAM libraries. The result is a shared object library that is loaded dynamically by the solver, seamlessly extending the capabilities of the standard OpenFOAM core installation.

3.2. OpenFOAM limitations

The physical constraints of the hypersonic regime highlight the inherent architectural limits of OpenFOAM. Specifically, the native compressible flow classes (derived from `hePsiThermo`) are designed around a single temperature variable (T). In the hypersonic scenarios we are targeting, the characteristic flow times are comparable to the relaxation times of the internal energy modes. Consequently, the standard class structure is insufficient to handle the stiff chemical kinetics and the state-dependent properties.

Rather than re-implementing these complex constitutive relations within the CFD solver, the development adopts a hybrid architecture. The macroscopic transport and finite volume discretization remain within the OpenFOAM framework, while the closure of the micro-physical terms is delegated to an external block. This approach avoids hardcoding constitutive laws, allowing the solver to query external thermochemical databases.

3.3. Mutation++

The `Mutation++` library plays an important role in the numerical solution of the problem addressed in this work.

3.3.1 Library overview and Architecture

`Mutation++` (*MUlticomponent Thermodynamic And Transport properties for IONized gases in C++*) is an open-source library originally developed at the von Karman Institute for Fluid Dynamics. It is designed to be coupled with conventional CFD solvers to provide high-fidelity thermodynamic, transport, chemical, and energy transfer properties required for simulating subsonic to hypersonic flows [4].
Unlike other thermochemical libraries, it was engineered specifically to handle the full spectrum of thermochemical non-equilibrium timescales. Its architecture relies on a unique object-oriented design that allows for the seamless integration of equilibrium, finite-rate chemistry, multi-temperature, and state-to-state models. The framework is highly extensible, facilitating the implementation of new physical models, data sets, or algorithms. Furthermore, its API is designed to simplify coupling with external solvers.

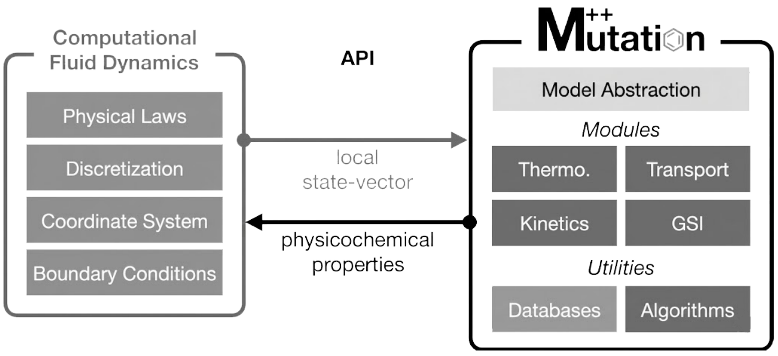


Figure 1: Data exchange workflow between the CFD solver and the `Mutation++` library [3].

3.3.2 Rationale for adoption and Data management

In the context of this work, the adoption of `Mutation++` is strictly necessary as the assumption of thermal equilibrium is invalid: the flow requires a description that accounts for the relaxation delay between the translational-rotational energy modes and the vibrational-electronic modes.
`Mutation++` serves as a robust backend interface that handles the complex constitutive relations required for this two-temperature (T_{tr}, T_{ve}) formulation. It manages the calculation of state-dependent properties such as specific heats, collision integrals, and reaction rate coefficients which behave non-linearly in non-equilibrium conditions.
A critical operational advantage is the library's separation of numerical algorithms from physical data. As utilized in the initialization of the `air_5` mixture, thermochemical properties and reaction mechanisms are not hardcoded into the solver but are loaded dynamically from external databases (XML files). This decoupling ensures that kinetic schemes and thermodynamic polynomials can be updated or substituted without requiring modifications to the source code, providing essential flexibility for testing different physical models.

3.3.3 Modeled physics

While the CFD solver manages the macroscopic conservation equations, `Mutation++` is tasked with providing the necessary closure relations for the gas mixture. In our context, the library is utilized to retrieve the fundamental thermochemical properties:

1. **Thermodynamics:** The library computes the species internal energies and specific heats consistent with the selected state model (`ChemNonEqTTv`). These properties are essential to solve the energy balances for the translational-rotational and vibrational-electronic modes.
2. **Finite-rate chemistry:** The library evaluates the net chemical production rates $(\dot{\omega}_i)$ for each species. Crucially, these rates are computed taking into account the non-equilibrium conditions, where the forward and backward reaction rates depend on the distinct temperatures (T_{tr}, T_{ve}) characterizing the mixture.

3.4. Implementation

Before integrating the two-temperature model into the complex 3D architecture of OpenFOAM, it is essential to validate the physical and numerical logic in a controlled environment. For this purpose, a zero-dimensional solver was developed to replicate the test case, as described in the reference literature [1].

3.4.1 Theoretical reference

The validation case consists of a reactor initialized in a state of strong thermal non-equilibrium. The gas is at rest with a translational temperature $T_{tr} = 30000$ K and a vibrational temperature $T_{ve} = 1000$ K.
Under these conditions, the system evolves through two coupled physical mechanisms governed by the energy conservation equations:

- **Vibrational relaxation:** Driven by the term (Q_{V-T}), energy flows from the highly energetic translational mode to the vibrational mode, attempting to equilibrate the two temperatures.
- **Dissociation chemistry:** The high translational energy triggers the dissociation reaction ($N_2 + M \rightleftharpoons 2N + M$). This process acts as a heat sink for the system.

Crucially, these processes are coupled via the Chemistry Vibration source term (Q_{C-V}). As dissociation occurs, it removes a specific amount of vibrational energy from the system (preferential dissociation), preventing T_{ve} from reaching the translational temperature immediately. The correct implementation must capture this interaction to ensure physical consistency.

3.4.2 Modernization of kinetic data

While the physical problem setup follows the work of the paper, our implementation introduces a significant improvement regarding the data. The reference paper utilizes constants and reaction rates based on older models. Conversely, our solver is coupled with the `Mutation++` library, which employs updated polynomial databases and more recent kinetic rates.
Consequently, while the qualitative behavior of the system (the relaxation trends and the equilibrium seeking) remains consistent with the literature, the quantitative results such as the exact equilibrium temperature and the reaction speed are expected to differ slightly. This deviation is intentional and represents an increase in physical fidelity compared to the older reference data.

3.4.3 Baseline implementation strategy

The numerical solver was implemented in C++ using an explicit strategy. The core logic operates within a temporal loop that advances the system state from the initial non-equilibrium condition to thermochemical equilibrium.
At each time step, the algorithm performs a sequential coupling:

1. **Chemical query:** The solver queries `Mutation++` to obtain the instantaneous net production rates ($\dot{\omega}_i$) based on the current density and temperatures.
2. **Energy source calculation:** The source terms Q_{V-T} and Q_{C-V} are calculated explicitly. This step is critical as it translates the chemical rates provided by the library into energy sinks/sources for the two-temperature equations.
3. **State update:** The ordinary differential equations (ODEs) for species densities and energy are integrated to update the state for the next time step.

This strategy ensures the strict conservation of total energy, as the enthalpy lost by the translational mode corresponds exactly to the sum of the energy gained by the vibrational mode and the energy consumed by bond breaking.

Numerical efficiency via Adaptive Time-Stepping: The heat bath problem is numerically stiff. At $t = 0$, the reaction rates are extremely high, causing temperature drops of thousands of Kelvin within nanoseconds. Conversely, as the system approaches equilibrium ($t > 10^{-4}$ s), the gradients vanish. Using a fixed time-step (Δt) would be computationally inefficient or numerically unstable.
To address this, an adaptive time-stepping was implemented. The solver monitors the rate of change of the translational temperature ($|dT_{tr}/dt|$) at every iteration:

- **Refinement:** If the temperature change exceeds a defined tolerance (e.g., 1%), the time-step is halved. This ensures that the violent initial transient is resolved with high temporal resolution.
- **Relaxation:** If the temperature change is minimal, the time-step is incrementally increased (up to a safe maximum). This allows the solver to reach the equilibrium state rapidly without wasting computational resources on the steady-state portion of the simulation.

This adaptive strategy proved essential to obtain a solution that accurately captures both the shock-like initial transient and the long-term relaxation behavior in an efficient way.

Low-level code optimizations In addition to the algorithmic efficiency gained through adaptive time-stepping, several low-level optimizations were applied to the C++ source code to minimize the CPU cycle consumption per iteration:

- **Pre-calculation of invariants:** All physical constants and parameters independent of the temporal evolution (e.g., specific gas constants, relaxation coefficients, thermal velocity pre-factors) were computed during the initialization phase, eliminating redundant operations within the main temporal loop.
- **Arithmetic optimization:** Recognizing that division operations are computationally more expensive than multiplications, the inverse of constant denominators was pre-computed. This allows the solver to perform faster floating-point multiplications instead of repeated divisions.
- **Mathematical function reduction:** The use of computationally intensive standard library functions (such as `std::pow` and `std::abs`) was minimized, replacing them with direct arithmetic operations or simplified logic where applicable.
- **Efficient I/O logic:** The data logging mechanism was refactored to avoid the overhead of type-casting. The write interval is now controlled through direct floating-point comparisons, removing the need for costly `float-to-int` conversions at every time step.

3.5. Integration

We see that the coupling between the `Mutation++` library and the two-temperature model implementation correctly reproduces the physics of thermochemical relaxation. The next logical step in the development process involves transitioning from this isolated reactor analysis to a full CFD simulation. This section outlines the integration strategy required to embed the logic into the OpenFOAM and to configure an hypersonic test case, such as shock Tube.

3.5.1 Embedding logic into the thermodynamic library

The simulation domain is discretized into thousands of control volumes (cells). Unlike the 0D case, where a single state vector evolves over time, the solver must manage spatial fields where thermodynamic properties vary from cell to cell.
The integration strategy relies on the custom thermodynamic class described in the development section `HighEnthalpyMulticomponentThermo`. The integration process involves the following conceptual steps:

1. **Field abstraction:** the logic validated in the 0D test must be encapsulated within a loop over all the computational cells. Instead of operating on scalar variables, the algorithm interacts with *Fields* (e.g., density, temperature becomes fields).
2. **Local state query:** for each cell, the thermodynamic library extracts the local thermodynamic state and queries the `Mutation++` engine.
3. **Source term storage:** the production rates ($\dot{\omega}_i$) and the energy source terms (Q_{V-T}, Q_{C-V}) calculated by the Paper model are stored in specific field variables. These fields are then exposed to the fluid solver, allowing the chemistry to influence the flow evolution.

3.5.2 Coupling with transport equations

While the thermodynamic library calculates the source terms, the evolution of the flow is governed by the conservation laws. To move from a static reactor to a dynamic flow simulation, the solver must resolve the transport equations that couple convection, diffusion, and the source terms derived from our logic.
The integration requires solving a specific set of governing equations:

1. **Species transport:** the mass balance for each chemical species must include the convective flux and the net production rates provided by `Mutation++`.
2. **Vibrational energy transport:** a specific transport equation for the vibrational-electronic energy (ρe_{ve}) is introduced. This equation balances the convective transport of vibrational energy with the relaxation and chemical coupling source terms ($Q_{V-T} + Q_{C-V}$) calculated by our custom logic.
3. **Total energy conservation:** the total energy equation remains conservative. It ensures that the energy consumed by dissociation or transferred to the vibrational mode is correctly subtracted from the translational-rotational energy, maintaining the *global energy balance* of the flow.

3.5.3 *Shock Tube* configuration strategy

To verify the implementation of these transport equations, the standard test case is the Shock Tube simulation. This problem represents a 1D Riemann problem and is ideal for observing non-equilibrium phenomena in a dynamic environment.
The configuration strategy involves setting up a domain with a sharp initial discontinuity:

- **driver section:** a region initialized with high-pressure and high-temperature nitrogen;
- **driven section:** a region with low-pressure and low-temperature gas.

Upon simulation start, this discontinuity generates a propagating shock wave. Unlike the 0D heat bath, where relaxation occurs uniformly, in the Shock Tube the thermochemical non-equilibrium is triggered exclusively across the moving shock front. The successful integration of the system would be demonstrated by observing the translational temperature peak at the shock front, followed by a relaxation zone where the vibrational temperature rises and chemical dissociation initiates, spatially resolving the physics.

4. Results

The presented results demonstrate the accuracy and reliability of the developed computational tool. The core implementation has been validated utilizing zero-dimensional benchmarks to verify the correctness of the thermochemical coupling against reference data. Furthermore, a performance analysis has been conducted to assess scalability, evaluating both weak and strong scaling through a comparison of sequential and parallel execution.

4.1. Code validation

Zero-dimensional heat bath configurations are used to assess the thermochemical model excluding flow transport and mesh errors. These tests verify that the temporal evolution of the system matches the data provided in the reference literature. Furthermore, the validation has been extended to include a multiple-species chemical model.

4.1.1 Single-species Model

The implementation followed an incremental approach. Initially, the solver was developed for a single-species heatbath and validated against the benchmark case shown in Figure 3a of the reference paper. The computed results were then directly compared with the literature data.
The single-specie simulated system is a heatbath of N_2 starting from a strongly non-equilibrium thermal state:

- Translational-rotational temperature: $T_{tr,0} = 10000$ K;
- Vibrational-electronic temperature: $T_{ve,0} = 1000$ K;
- Pressure: $P_0 = 1$ atm;
- Simulated time interval: $t \in [10^{-9}, 2 \times 10^{-5}]$ s.

The result is shown in Figure 2.
The figure demonstrates a strong agreement between the computed results and the reference data, confirming the accuracy of the solver. While the introduction of a second species adds significant complexity to the physical model, this single-species test served as an optimal baseline for the initial implementation.

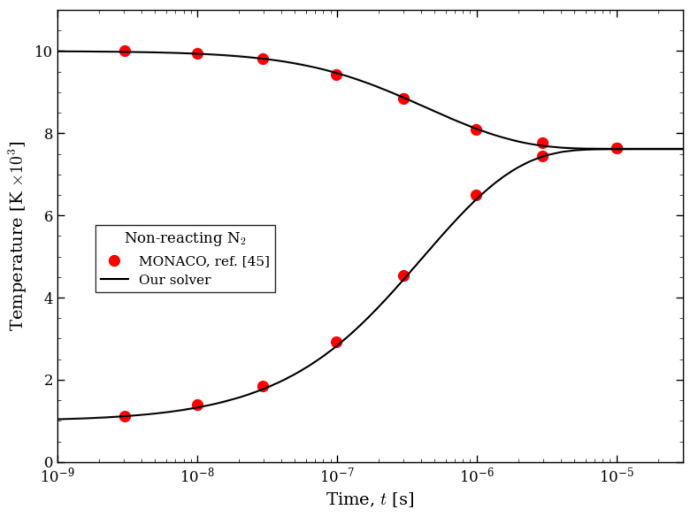


Figure 2: Non-reacting single specie N_2 heatbath validated against reference data

4.1.2 Multiple-species Model

After validating the single-specie case, we started working on a multiple-species model. As before, the implemented two-temperature thermochemical kernel was validated using a zero-dimensional *heatbath* test case. This configuration is deliberately chosen because it isolates the thermochemistry and internal-energy relaxation mechanisms, while completely excluding spatial discretization errors (mesh, gradients, fluxes) and flow-transport effects.
At this stage, the goal is **not** a strict pointwise numerical replication of a reference curve, but to verify that the code:

- *Reproduces* the expected qualitative non-equilibrium behaviour (energy transfer and relaxation);
- Remains *stable* over stiff transients and long relaxation times;
- Yields characteristic temperature levels and time scales *consistent* with the reference literature.

Test case set-up and initial conditions The first two-species simulated system is a reacting heatbath of $N_2 + N$ starting from a strongly non-equilibrium thermal state:

- Translational-rotational temperature: $T_{tr,0} = 30000$ K;
- Vibrational-electronic temperature: $T_{ve,0} = 1000$ K;
- Pressure: $P_0 = 1$ atm;
- Simulated time interval: $t \in [10^{-9}, 10^{-3}]$ s.

The time integration uses an adaptive step size starting from $dt = 10^{-13}$ s, combined with stability controls on chemical source terms and on the vibrational energy update. The solver outputs the time histories of temperatures and species densities for validation.

Species set consistency The reference paper compares different modelling options for a reacting $N_2 + N$ heat bath (Fig. 7 in [1]). In our implementation, `Mutation++` is instantiated with the mixture `air_5` and a two-temperature state model; however, the 0D test is effectively restricted to nitrogen chemistry:

- Only N_2 and N are initialized with non-zero densities;
- During the run, the computed production rates confirm that O, O_2, and NO remain identically zero (i.e., $\dot{\omega}_O = \dot{\omega}_{O_2} = \dot{\omega}_{NO} = 0$), while N_2 and N exchange mass through dissociation/recombination.

Therefore, the present test is a consistent reacting $N_2 + N$ heat bath, analogous to the reacting case discussed in the paper.

Observed relaxation behaviour Starting from the initial non-equilibrium condition, the solution exhibits the expected dynamics of high-temperature nitrogen with finite-rate chemistry:

- T_{tr} decreases rapidly as energy is transferred to internal modes and as endothermic dissociation becomes active;
- T_{ve} rises from its initial low value, reaches a peak, and subsequently relaxes toward equilibrium as the system approaches a chemically and thermally relaxed state.

From the obtained curves, the following characteristic values can be identified:

- Peak vibrational-electronic temperature: $T_{ve,\max} \approx 1.4 \times 10^4$ K;

- Peak time: $t_{\max} \approx 3 \times 10^{-7}$ s;
- Quasi-equilibrium onset: $t \approx 10^{-4}$ s;
- Final temperature levels: $T_{tr} \approx T_{ve} \approx (6.5\text{-}7) \times 10^3$ K.

The temporal evolution of both temperatures is shown in Fig. 3.

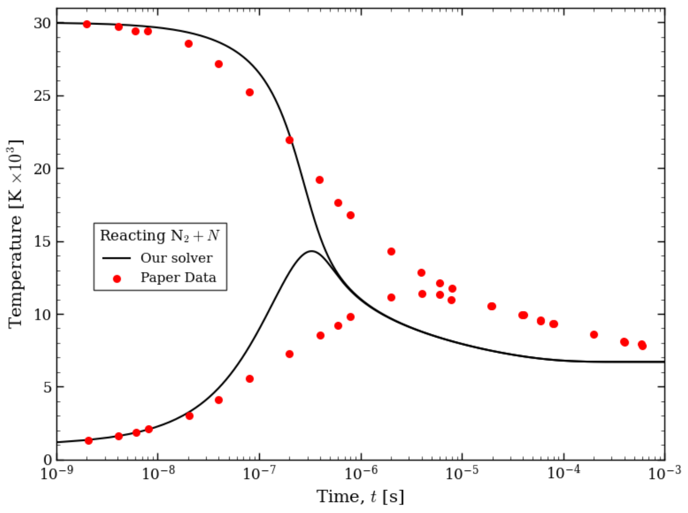


Figure 3: Temporal evolution of translational-rotational and vibrational-electronic temperatures for the 0D reacting nitrogen heat bath ($N_2 + N$).

4.1.3 Comparison with the reference literature

$N_2 + N$ heatbath Figure 7a of [1] reports the reacting $N_2 + N$ heat bath temperature evolution for different modelling choices (e.g. a Park two-temperature set-up) and compares against DSMC results. The current simulation successfully captures the qualitative behavior of the reference case:

- Sharp drop of T_{tr} from the initial 30,000 K level;
- A pronounced T_{ve} overshoot before relaxation;
- Relaxation toward a common equilibrium temperature at late times.

A strict numerical match is **not expected** neither here nor in the paper for two concrete, verifiable reasons:

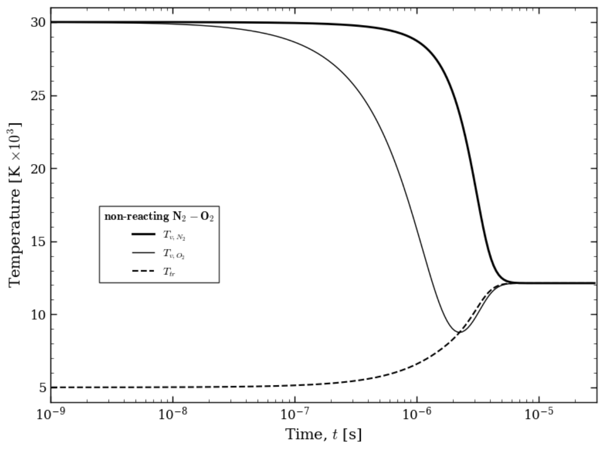
1. **Different initial density/pressure.** The paper sets an initial number density of $n_0 = 5 \times 10^{22}\,\text{m}^{-3}$ for both species (Fig. 7a in [1]), corresponding to a pressure significantly higher than 1 atm at $T_{tr,0} = 30000$ K. Since relaxation and reaction time scales depend on collision frequency, different density levels directly shift both peak values and time scales.
2. **Different thermo-chemical data and modelling choices.** The reference curves in Fig. 7a are produced with specific model combinations (e.g. Park TTv model and selected rate constants), whereas our implementation relies on `Mutation++` for high-temperature thermodynamic properties and production rates. Differences in kinetics/thermodynamics translate into quantitative shifts even when the qualitative behaviour is the same.

To make the comparison explicit, Tab 1 reports a few key indicators from this work and approximate values estimated from Fig. 7a of [1] (visual reading, therefore intended as *order-of-magnitude* validation rather than exact benchmarking).

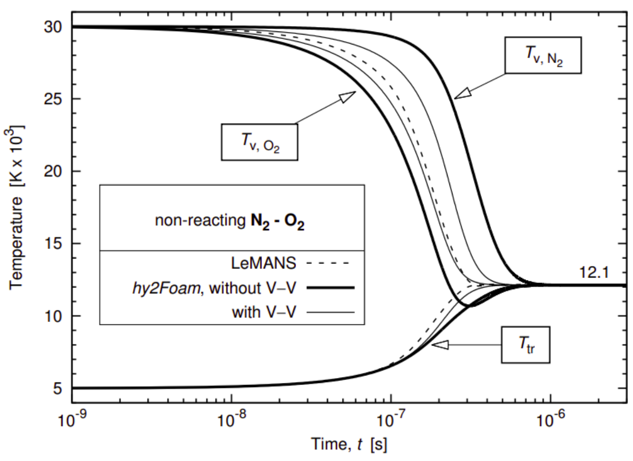
Quantity	**This work**	**Paper (Fig. 7a)**
$T_{tr,0}$ [K]	30000	30000
$T_{ve,0}$ [K]	1000	1000
$T_{ve,\max}$ [K]	$\approx 1.4 \times 10^4$	$\approx (1.8\text{-}2.0) \times 10^4$
$t_{\max}$ [s]	$\approx 3 \times 10^{-7}$	$\approx (2\text{-}5) \times 10^{-7}$
T_{eq} [K]	$\approx (6.5\text{-}7) \times 10^3$	$\approx (7\text{-}8) \times 10^3$

Table 1: Key quantities for the reacting nitrogen heat bath. Paper values are visually estimated from Fig. 7a of [1].

$N_2 - O_2$ heatbath The solver was also tested on a non-reacting $N_2 - O_2$ configuration. By modifying the initial mixture composition while maintaining the `Mutation++` coupling, the simulation results were validated against the corresponding reference curves.
Given that the reference results are provided only as continuous curves, the computed solution is compared against the literature data by displaying the two figures separately.



(a) Our simulation



(b) Paper's simulation

Figure 4: $N_2 - O_2$ non-reacting heatbath simulations comparison

For the *Hy2Foam* simulation without V-V exchange, the temperature profiles match the qualitative behavior of the reference curves. Furthermore, the quantitative accuracy is confirmed by the final equilibrium temperature, which settles at the expected value of 12100 K.
Overall, the matching trends confirm the solver's correctness. Any remaining discrepancies are consistent with the known sensitivity to initial conditions and thermochemical properties.

4.1.4 Numerical stability and robustness

The simulation remains **numerically stable** for the entire integration interval. No non-physical values (negative densities or temperatures) are observed, and the adaptive time stepping successfully handles the stiff initial transient without spurious oscillations. This provides a reliable basis for the subsequent extension to multi-cell OpenFOAM simulations, where the same thermochemical logic will be applied locally at the cell level.

4.2. Performance analysis

In this section, the computational performance of the developed 0D kernel is analysed. The objective is to quantify the impact of code-level optimisations and to assess the behaviour of the implementation when OpenMP parallelism is introduced. The analysis is structured in three steps: a baseline sequential implementation, an optimised sequential version, and an OpenMP-parallel implementation of the optimised code.

4.2.1 Test platform and methodology

All measurements were performed on an Intel Core i7-1355U processor, featuring 4 physical cores and 8 hardware threads. Execution times are reported as total wall-clock times. To reduce run-to-run variability, each configuration was executed multiple times and the median value was retained.
For the OpenMP tests, thread placement was controlled using the environment variables `OMP_PLACES=cores` and `OMP_PROC_BIND=close`, in order to limit thread migration and improve reproducibility.

4.2.2 Sequential comparison: baseline vs. optimised version

The first comparison concerns two purely sequential implementations of the kernel:
- baseline version, representing a straightforward and readable implementation;
- optimised version, obtained by reorganising the code without altering the underlying numerical or physical model.

Implementation	**Time [s]**	**Speedup**
Baseline	0.283	1.00
Optimised	0.217	1.30

Table 2: Sequential performance comparison between baseline and optimised implementations.

Results are summarized in Table 2.

The observed improvement mainly originates from reducing avoidable overhead inside the time-marching loop, such as repeated lookups, unnecessary temporary allocations, and non-essential recomputations. While these changes do not modify the algorithm itself, their cumulative effect becomes significant when the same kernel is executed many times.

Parallelisation context The optimised kernel was extended to exploit shared-memory parallelism using OpenMP by executing multiple independent instances of the same heat-bath problem concurrently. Each instance represents an identical 0D thermochemical system evolving in time, and is introduced to replicate the computational workload rather than to model a coupled physical simulation. Because each instance evolves independently, no data exchange or synchronisation is required between them, making the kernel naturally suitable for thread-level parallel execution.

OpenMP implementation strategy From an implementation perspective, parallelism is introduced by distributing the replicated kernel executions across OpenMP threads. A single parallel region combined with a work-sharing construct is used to assign independent kernel instances to different threads (`#pragma omp parallel` and `#pragma omp for`). This approach allows multiple realizations of the same thermochemical problem to be executed concurrently, providing a simple and controlled way to assess the parallel behaviour of the kernel.

Thread-safety issue in mutation While the numerical structure of the kernel is trivially parallel, the coupling with the external thermochemical backend introduces a non-trivial software constraint. During development, it was observed that the library cannot be safely accessed by multiple threads through a single shared `Mutation::Mixture` instance. When a common instance was used across threads, the execution exhibited non-deterministic behaviour and, in some cases, runtime failures, indicating the presence of internal mutable state not designed for concurrent access.

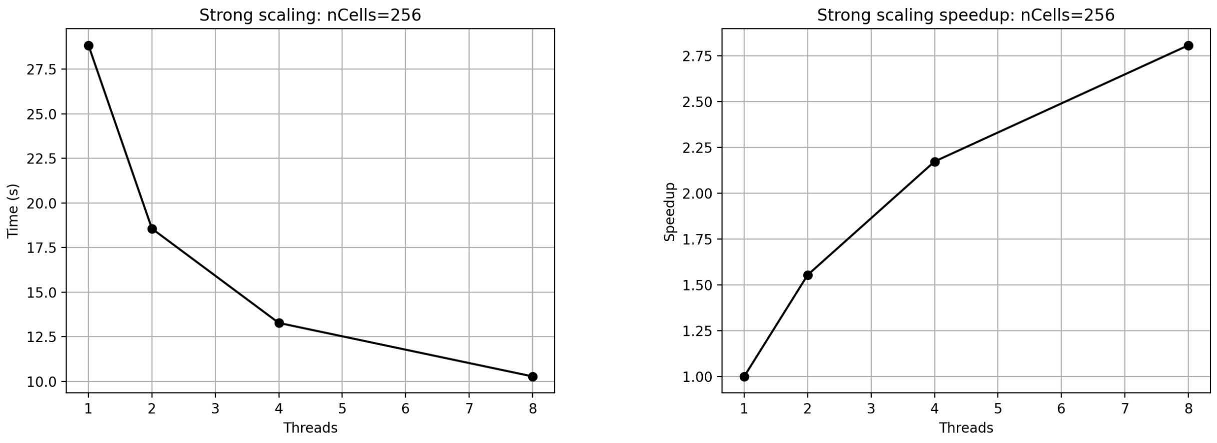
Adopted mitigation strategy To guarantee numerical correctness and reproducibility, the parallel implementation adopts a thread-local ownership model. A dedicated `Mutation::Mixture` instance is assigned to each OpenMP thread, and all instances are created serially before entering the parallel region. Inside the parallel region, each thread operates exclusively on its own instance, ensuring that the thermochemical state handled by `Mutation++` remains strictly thread-local throughout the execution.

4.2.3 Strong scaling

Strong scaling was evaluated by keeping the total number of cells fixed ($nCells = 256$) while increasing the number of OpenMP threads. The measured wall times are reported in Table 3.

Threads	**Time** $[s]$	**Speedup**	**Efficiency**
1	28.84	1.00	1.00
2	18.57	1.55	0.78
4	13.28	2.17	0.54
8	10.28	2.81	0.35

Table 3: Strong scaling results with fixed number of cells ($nCells = 256$).



(a) Wall time versus number of threads.

(b) Speedup versus number of threads.

Figure 5: Strong scaling behaviour of the OpenMP-parallel 0D kernel.

Results show a clear reduction in execution time up to 4 threads, followed by a progressive saturation. On this laptop-class CPU, the speedup is significantly sub-linear despite the embarrassingly parallel structure of the problem. This behaviour is attributed to the relatively heavy thermochemical computations (including `Mutation++` calls), combined with shared-resource effects and hyper-threading limitations.

4.2.4 Weak scaling

Weak scaling was assessed by keeping the workload per thread constant (8 cells per thread), such that $nCells = 8 \cdot N_{\text{threads}}$. The measured execution times are reported in Table 4.

Threads	**Cells/thread**	**Total cells**	**Time** $[s]$
1	8	8	0.90
2	8	16	1.20
4	8	32	1.74
8	8	64	2.52

Table 4: Weak scaling results with constant workload per thread (8 cells/thread).

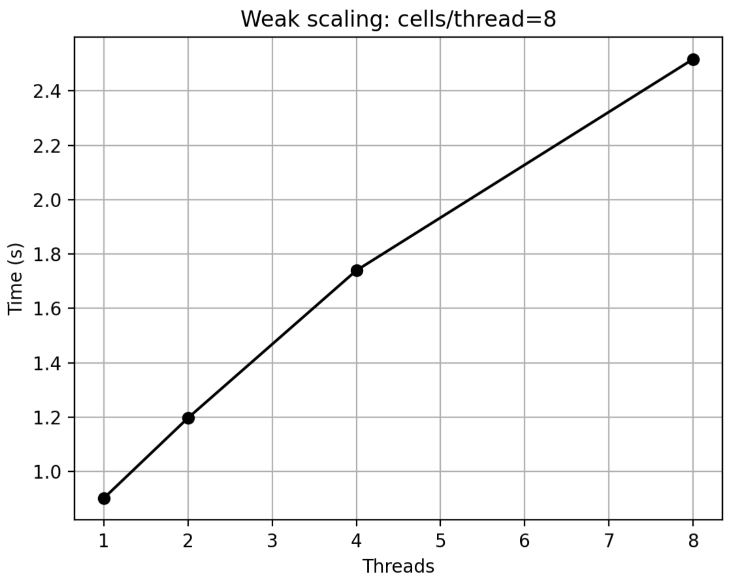


Figure 6: Weak scaling results with constant number of cells per thread.

In ideal weak scaling, the wall time should remain approximately constant as the number of threads increases. In this case, the observed growth in execution time indicates that the kernel is not purely compute-bound on the tested architecture. Thread contention, shared resources, and frequency scaling effects become increasingly relevant as the level of parallelism grows. Despite these limitations, the results provide a realistic estimate of the achievable parallel efficiency on a single node and motivate more detailed performance studies on larger platforms.

5. Conclusions

This work addressed the implementation of high-enthalpy thermochemical effects within the OpenFOAM framework. The project focused on integrating stiff relaxation dynamics and finite-rate chemistry into the finite volume architecture.

Summary of achievements A hybrid computational strategy was implemented to decouple macroscopic transport from microscopic thermochemistry. By interfacing the solver with the `Mutation++` library, the management of thermodynamic properties and chemical source terms was delegated to an external backend. Numerically, an adaptive time-stepping algorithm, supported by low-level code optimizations, provided the necessary stability to resolve stiff initial transients. The validation on zero-dimensional heat bath cases verified the correctness of the coupling. Computed relaxation times and equilibrium compositions matched the reference literature, confirming the consistency of the thermochemical kernel.

Future developments Following the validation of the reaction kinetics, the development roadmap consists of:

- **1D Shock Tube implementation:** Coupling the validated zero-dimensional source terms with convective flux reconstruction to capture shock structures and relaxation layers in a spatially resolved domain.
- **Multi-dimensional extension:** Application of the solver to standard 2D benchmarks (e.g., cylinder in cross-flow) to evaluate grid dependence and boundary condition behavior under high-enthalpy loads.

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. *Aerospace*, 3(4), 2016.

[2] Federico piscaglia. *Lessons*. 2025.

[3] J.B. Scoggins, V. Leroy, G. Bellas-Chatzigeorgis, B. Dias, and T.E. Magin. Mutation++: Multicomponent thermodynamic and transport properties for ionized gases library in C++. *SoftwareX*, 12:100575, 2020.

[4] J.B. Scoggins and T.E. Magin. Mutation++ library repository. `https://github.com/mutationpp/Mutationpp`, 2020. Accessed: 2023-10-27.