

DETAILED TRANSPORT AND PERFORMANCE OPTIMIZATION FOR MASSIVELY PARALLEL SIMULATIONS OF TURBULENT COMBUSTION WITH OPENFOAM

THORSTEN ZIRWES^{1,2}, FEICHI ZHANG², JORDAN A. DENEV¹, PETER HABISREUTHER²
HENNING BOCKHORN², DIMOSTHENIS TRIMIS²

¹Karlsruhe Institute of Technology, Steinbuch Centre for Computing, [thorsten.zirwes, jordan.denev]@kit.edu

²Karlsruhe Institute of Technology, Engler-Bunte-Institute, Combustion Technology,
[thorsten.zirwes, feichi.zhang, peter.habisreuther, henning.bockhorn, dimosthenis.trimis]@kit.edu

Keywords: Detailed Transport, Performance Optimization, Parallel Scaling, Validation, Turbulent Combustion, HPC

This work describes the implementation of two key features for enabling high performance computing (HPC) of highly resolved turbulent combustion simulations: detailed molecular transport for chemical species and efficient computation of chemical reaction rates. The transport model is based on an implementation of the thermo-chemical library Cantera [1] and is necessary to resolve the inner structure of flames. The chemical reaction rates are computed from automatically generated chemistry-model classes [2], which contain highly optimized code for a specific reaction mechanism. In combination with Sundials' [3] ODE solver, this leads to drastic reductions in computing time. The new features are validated and applied to a turbulent flame with inhomogeneous mixing conditions on a grid with 150 million cells. The simulation is performed on Germany's fastest supercomputer "Hazel Hen" [4] on 28,800 CPU cores, showing very good scalability. The good agreement with experimental data shows that the proposed implementations combined with the capabilities of OpenFOAM are able to accurately and efficiently simulate even challenging flame setups.

1 Detailed Molecular Transport Coefficients

Although many transport models are available in OpenFOAM for standard solvers like `reactingFoam`, they all have in common that each chemical species in the fluid has the same diffusion coefficient. This assumption is justified if turbulent transport is more important than molecular transport. In highly resolved simulations however, where no turbulence models are used as shown in Sect. 3, detailed diffusion coefficients for each species are necessary in order to correctly capture the flame structure. Therefore, a coupling interface [5] between OpenFOAM's thermo classes and Cantera [1] was created, following an idea by Gschaider and Rhem [6], where Cantera was used as an external library to compute the transport coefficients and chemical reaction rates. Since then, the relevant parts of the Cantera code have been extracted and directly compiled into a thermo-physical model for OpenFOAM 5.x and 1712+. This eliminates indirections leading to better performance and making the new solver independent of Cantera as a third-party library.

In the detailed thermo-physical model, the transport of each chemical species is described by six gas kinetic properties: Lennard-Jones collision diameter and energy well depth, polarizability, dipole moment, rotational relaxation collision number and molecule geometry. These information are usually provided as part of the chemical reaction mechanism in CHEMKIN format which can easily be converted into Cantera's xml format. The xml file then serves as input for the thermo-physical model. OpenFOAM provides a similar utility `chemkinToFoam` which converts reaction mechanisms from CHEMKIN format to OpenFOAM format, but ignores the aforementioned transport properties. In the detailed transport model, the gas kinetic properties are used to compute binary diffusion coefficients $\mathcal{D}_{k,i}$, viscosity μ_k and heat conductivity λ_k for each species k from the Chapman-Enskog solution of the Boltzmann equation. Mixing laws are applied to compute mixture-averaged properties, like Wilke's mixing law for viscosity, or the Hirschfelder-Curtiss approximation for the mass diffusion coefficient $\frac{1}{D_k} = \sum_{i \neq k} \frac{X_i}{\mathcal{D}_{k,i}} + \frac{X_k}{1-Y_k} \sum_{i \neq k} \frac{Y_i}{\mathcal{D}_{k,i}}$ [7], where X_k is the mole fraction and Y_k the mass fraction of the k -th species. Because every species has its own diffusion coefficient D_k and molecular mass diffusion flux \vec{j}_k , the governing equations for the species masses and energy have to be adapted:

$$\frac{\partial (\rho Y_k)}{\partial t} + \nabla \cdot (\rho(\vec{u} + \vec{u}_c) Y_k) = \dot{\omega}_k - \nabla \cdot \vec{j}_k, \quad \vec{j}_k = -\rho D_k \nabla Y_k, \quad k = 1 \dots N-1 \quad (1)$$

$$\frac{\partial (\rho h_{s,t})}{\partial t} + \nabla \cdot (\rho \vec{u} h_{s,t}) = \nabla \cdot (\alpha \nabla h_s) + \frac{\partial p}{\partial t} - \sum_k h_k^\circ \dot{\omega}_k - \nabla \cdot \sum_k h_{s,k} \left((\vec{j}_k + \rho Y_k \vec{u}_c) + \alpha \nabla Y_k \right) \quad (2)$$

Here, ρ is the density, t time, \vec{u} the fluid velocity, $\dot{\omega}$ the chemical reaction rate, N the number of species, $h_{s,t} = h_s + \frac{1}{2} \vec{u} \cdot \vec{u}$ the total sensible enthalpy of the mixture, T the temperature, p the pressure, α the ratio of thermal conductivity to isobaric heat capacity, h_k° the enthalpy of formation and $h_{s,k}$ the sensible enthalpy of species k . Both equations are the same as

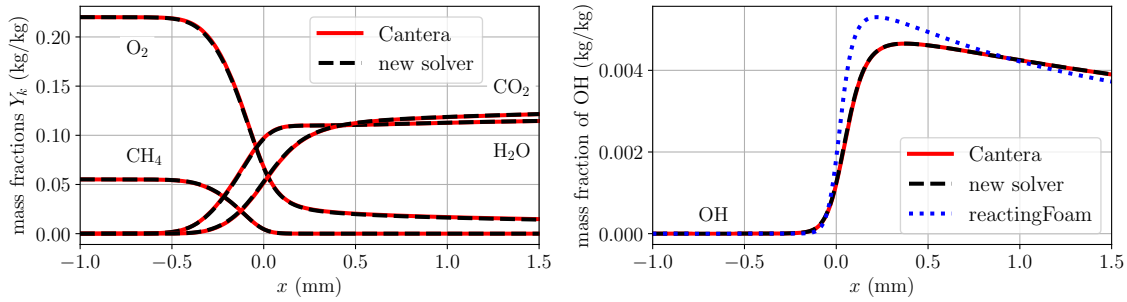


Figure 1: Spatial profiles of species mass fractions in a one-dimensional flame. Computed with Cantera [1], the new solver including detailed transport and the standard reactingFoam solver with the Sutherland transport model in OpenFOAM 5.x.

in the standard reactingFoam family of solvers, with the exception of the correction velocity $\vec{u}_c = -\frac{1}{\rho} \sum_k \vec{j}_k$ in the convective term of Eq. (1), which is necessary in order to ensure overall mass conservation, and the last term on the r.h.s. of Eq. (2), which is a consequence of nonequidiffusion or non-unity Lewis number.

Figure 1 on the left shows the main species profiles for a canonical flame setup, the one-dimensional premixed flame. The domain consists of 10,000 cells spanning 10 cm. The left boundary is an inlet with an unburnt methane-air mixture at an equivalence ratio of $\phi = 1$, $T = 300$ K and $u = 0.37$ m/s, and the right side is an outlet. The chemistry is described by the GRI 3.0 [8] reaction mechanism. The domain is initialized with cold, unburnt gas in the left half, and hot burnt gas in the right half. After some time, a stationary flame develops. The same setup is simulated in Cantera. The comparison on the left of Fig. 1 validates the implementation of the new solver by comparing spatial species profiles in the flame with Cantera. On the right of Fig. 1, results for the profile of the intermediate radical species OH from the new solver and the standard reactingFoam solver are compared. Again, Cantera's reference solution coincides with the new solver. The simulation with reactingFoam uses the exact same numerical and physical settings as the new solver except for the transport model. As in many of the standard OpenFOAM tutorial cases, the Sutherland transport model is used for the GRI 3.0 mechanism in reactingFoam, which shows large deviation for the intermediate species profiles. This demonstrates that detailed transport coefficients for each species are necessary to correctly predict the flame properties.

2 Performance Optimization for Chemical Reaction Rates

Combustion processes are governed by a large number of intermediate species and chemical reactions. The computation of chemical reaction rates is therefore often the performance bottleneck. Because of this, a new approach has been introduced in order to speed up the chemistry computations. When preparing a new case with standard solvers like reactingFoam, OpenFOAM's chemkinToFoam utility can be used to convert a reaction mechanism in CHEMKIN format to a new set of files in OpenFOAM's format, which serve as input files for the general chemistry model. In the new approach [2], instead of using chemkinToFoam, a self-developed converter tool can be used which takes a reaction mechanism in CHEMKIN or Cantera format as input and creates a directory containing C++ source code for a new chemistry model class specifically for that mechanism. For example, applying the converter tool to the GRI 3.0 mechanism would generate code for a new chemistry model class named optimizedChemistryModel_GRI. Compiling this class results in a chemistry model which can directly be used in the simulation through OpenFOAM's runtime selection mechanism and is compatible to the general chemistry implementation, for example it can be combined with OpenFOAM's TDAC model. The code in the generated class contains all information for computing the chemical reaction rates from detailed Arrhenius and other reaction type formulations without simplification, in the same way that OpenFOAM and Cantera do. This approach has several advantages: it requires little effort by the user because all steps of the conversion are performed automatically. During the conversion, species and reactions are reordered by their type in order to allow auto-vectorization of critical loops by the compiler. Redundant operations are eliminated and more compiler optimizations are enabled because otherwise unknown parameters for the species and reactions are known at compile time. The data is laid out in a cache-friendly way and is explicitly aligned. The resulting code has been shown to perform significantly faster and to reduce cache misses by up to a factor of 30. For more information, see [2].

Figure 2 shows results for another canonical flame setup, the zero-dimensional auto-ignition of a hydrogen-air mixture using the reaction mechanism by Li et al. [9]. The computational domain consists of only one cell, similar to chemFoam. Therefore, transport processes are irrelevant in this setup. The domain is initially filled with hot hydrogen and air, which then auto-ignite. On the left of Fig. 2, the mass fraction profile of H_2O_2 is depicted over time, which is an intermediate species during ignition. The simulation has been performed with the first order Euler time discretization with the standard reactingFoam solver and the new solver. For the Euler scheme, the chemical reaction rates are computed from the linear approximation $\dot{\omega}_k \approx M_k (C_k^{n+1} - C_k^n) / \Delta t$ in all solvers, where M_k is the molar mass of the species, Δt the time step and C_k^n the species concentration at the current time step n . C_k^{n+1} is an estimate for the concentration at the next time step obtained from an operator splitting approach, where the concentrations in each cell are integrated over the time step by an ODE integrator, allowing to use adaptive sub-time stepping. The results coincide with the reference

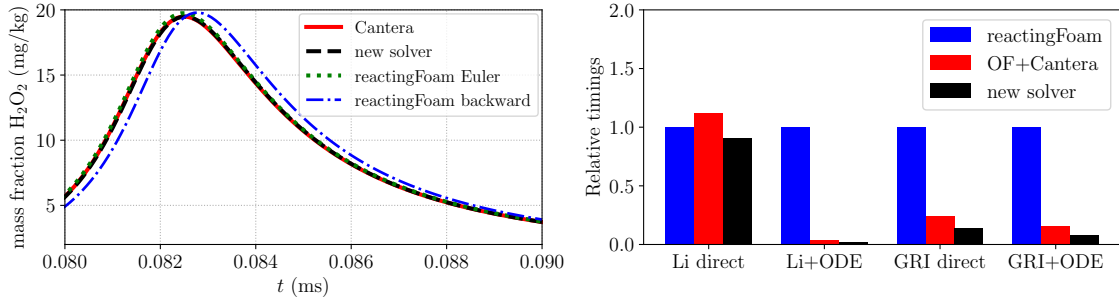


Figure 2: Left: Mass fraction of H_2O_2 over time during ignition of hydrogen. Right: Time spent computing chemical reaction rates for the auto-ignition case normalized to the time of `reactingFoam`. Measurements are done for two reaction mechanism (GRI 3.0 [8] and Li [9]) using either the operator splitting approach (“ODE”) or direct computation.

solution obtained from Cantera. In the case of the three-point backward time discretization, the new solver also gives accurate results while `reactingFoam` deviates from the reference solution. The reason is that the chemistry model in `reactingFoam` always computes the chemical reaction rates from the linear approximation above, while the new solver switches to the more consistent formulation $\dot{\omega}_k \approx M_k \left(\frac{3}{2} C_k^{n+1} - 2 C_k^n + \frac{1}{2} C_k^{n-1} \right) / \Delta t$ for the backward scheme. In Fig. 2 on the right, the time spent on computing chemical reaction rates, which corresponds to the call of `reaction->correct()` in `YEqn.H`, is plotted normalized to the respective time needed by `reactingFoam` for the auto-ignition simulations. All codes have been compiled with the Intel compiler `icpc 18` with `-fast`. In general, the reaction rates can either be computed directly (corresponds to combustion model setting `integrateRates false;`) denoted by “direct”, or from the operator splitting approach which requires an ODE integrator. In the timings denoted with “ODE”, `seulex` is used as the ODE integrator in `reactingFoam`. In the new solver containing the automatically generated optimized code (“new solver”) and in the solver which is coupled to Cantera as an external library for computing the reaction rates (“OF+Cantera”), Sundials’ [3] `CVODE` integrator is available and has been used in the measurements. The times needed to compute the reaction rates from the mechanism by Li directly (“Li direct”) is about 10 % slower with `reactingFoam` and 20 % slower with Cantera compared to the optimized code in the new solver. This is due to Li being a relatively small mechanism with only 9 species and 21 reactions. As the mechanisms become larger, the performance benefit of the optimized code becomes larger as well. With the GRI 3.0 mechanism, which contains 53 species and 325 reactions, direct computation of reaction rates with `reactingFoam` takes seven times longer and Cantera’s implementation twice as long compared to the optimized code. There is also a large difference in runtime which stems from the choice of ODE integrator. In the measurements, the initial time step and absolute and relative tolerances are the same for `seulex` and `CVODE`, and have been chosen to keep errors below 1 % compared to the reference solution by Cantera. With these strict tolerances, `CVODE` combined with the optimized chemistry code is by an order of magnitude faster than the standard OpenFOAM implementation while still giving the same simulation results. Similar performance results have been obtained for the Rosenbrock integrator instead of `seulex`.

3 Massively Parallel Simulation of a Turbulent Flame

This section describes an application of the new solver where both implementations are required—detailed transport and efficient chemistry computations. Subject of the simulation is the Sandia/Sydney burner [10]. This burner is experimentally well investigated and is operated with methane-air, but due to very inhomogeneous mixing conditions at the burner nozzle most combustion models are not able to correctly predict the flame properties. Therefore, this simulation utilizes the detailed transport model and fully resolves the flame, allowing a model free simulation. The computational grid consists of 150 mil. cells, uses `backward` time discretization, `cubic` for spatial discretizations and a complex reaction mechanism with 19 species [2]. Due to the optimized chemistry code and the use of the `CVODE` integrator, the total simulation time can be reduced by 50 % compared to the Cantera implementation [2]. This makes it possible to run the simulation on Germany’s fastest supercomputer [4] on 28,800 CPU cores. The total simulation required about 15 mil. core hours and produced 15 TB of data, which includes transient fields of flow variables and chemical scalars.

Figure 3 depicts the temperature profile of the flame where the inner fuel-air jet is ignited by a hot pilot gas and leads to a high-temperature combustion zone along the shear layer. Fig. 4 on the left shows a snapshot from the simulation. Depicted is an iso-surface of vorticity near the burner nozzle. It illustrates the turbulent flow structures which are formed by the inner unburnt jet and become destroyed due to increasing viscosity when the flame begins to burn at larger radii. Parallel

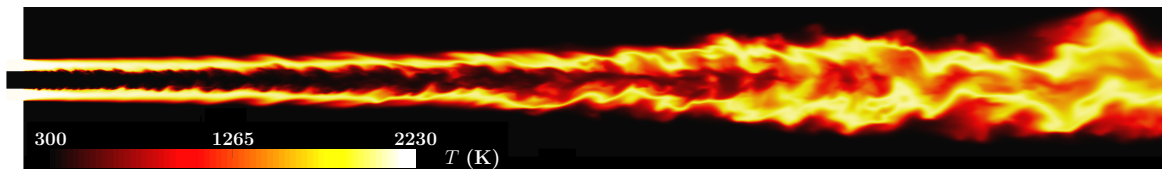


Figure 3: 2D cut of the temperature profile from the 3D simulation of the Sandia/Sydney flame [10].

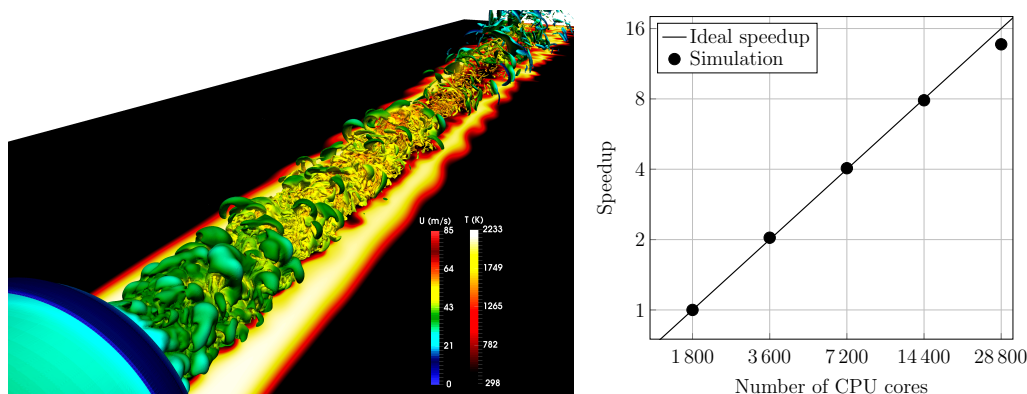


Figure 4: Left: Iso-surface of vorticity near the inlet nozzle colored by fluid velocity U and a 2D cutting plane of the temperature, illustrating the turbulent flow structures in the inner jet region. Right: Parallel speedup for a strong scaling case with 175 million cells and optimized chemistry performed with OpenFOAM 5.x on the Hazel Hen supercomputer at HLRS [4].

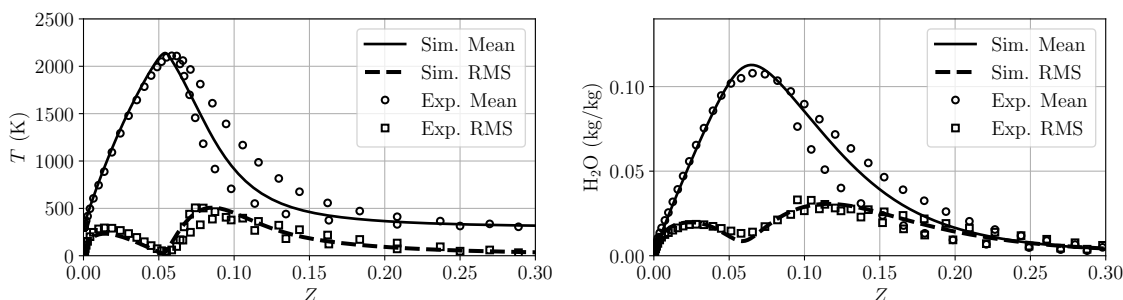


Figure 5: Comparison of radial temperature (left) and water mass fraction (right) profiles of time averaged and RMS values at fixed axial positions plotted against radial mixture fraction Z profiles with experimental measurements [10].

scaling on the right of Fig. 4 is almost linear up to 14,400 cores and still has an efficiency of 86 % at 28,800 CPU cores. The good scaling results are achieved because a large part of the simulation time is spent on computing chemical reaction rates, which does not require MPI communication. In Fig. 5, the simulation results are compared with experimental measurements. Time averaged and RMS values show very good quantitative agreement, e.g. the temperature and water mass fraction profiles lie well within the experimental uncertainties. Due to the highly resolved results, this simulation will be published as a reference database for the development of new combustion models.

Acknowledgments

This work was performed on the national supercomputer Cray XC40 Hazel Hen at the High Performance Computing Center Stuttgart (HLRS) and on the computational resource ForHLR II at SCC (KIT) funded by the Ministry of Science, Research and the Arts Baden-Württemberg and DFG (“Deutsche Forschungsgemeinschaft”).

References

- [1] D. Goodwin, H. Moffat, and R. Speth, “Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. version 2.3.0b,” 2017, software available at www.cantera.org.
- [2] T. Zirwes, F. Zhang, J. Denev, P. Habisreuther, and H. Bockhorn, “Automated code generation for maximizing performance of detailed chemistry calculations in OpenFOAM,” in *High Performance Computing in Science and Engineering '17*, W. Nagel, D. Kröner, and M. Resch, Eds. Springer, 2017, pp. 189–204.
- [3] “Suite of nonlinear and differential/algebraic equation solvers,” <http://computation.llnl.gov/casc/sundials>.
- [4] “High performance computing center stuttgart,” www.hlrs.de/systems/cray-xc40-hazel-hen, 2018.
- [5] H. Bonart, “Implementation and validation of a solver for direct numerical simulations of turbulent reacting flows in OpenFOAM,” Bachelor’s Thesis, Karlsruhe Institute of Technology, Germany, 2012.
- [6] B. Gschaider, M. Rhem, P. Seifert, and B. Meyer, “Implementation of an alternative chemistry library into OpenFOAM,” http://powerlab.fsb.hr/ped/kturbo/openfoam/Berlin2008/SessionVA/OSCIC-08_GschaiderRehm.pdf.
- [7] R. Kee, M. Coltrin, and P. Glarborg, *Chemically reacting flow: theory and practice*. John Wiley & Sons, 2005.
- [8] G. Smith, D. Golden, M. Frenklach, N. Moriarty, B. Eiteneer, M. Goldenberg *et al.*, “Gri 3.0 reaction mechanism.”
- [9] J. Li, Z. Zhao, A. Kazakov, and F. L. Dryer, “An updated comprehensive kinetic model of hydrogen combustion,” *International journal of chemical kinetics*, vol. 36, no. 10, pp. 566–575, 2004.
- [10] R. Barlow, S. Meares, G. Magnotti, H. Cutcher, and A. Masri, “Local extinction and near-field structure in piloted turbulent CH₄/air jet flames with inhomogeneous inlets,” *Combust. Flame*, vol. 162, no. 10, pp. 3516–3540, 2015.