HTR-1.3 solver: predicting electrified combustion using the Hypersonic Task-based Research solver

Mario Di Renzo[*]

Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Tolouse 31100, France

Abstract

This manuscript presents an updated open-source version of the Hypersonics Task-based Research (HTR) solver. The solver, whose main features are presented in Di Renzo *et al.* (Comput. Phys. Commun. 255, 2020) and Di Renzo & Pirozzoli (Comput. Phys. Commun. 261, 2021), is designed for direct numerical simulation of reacting flows at high Reynolds numbers. This new version extends the applications of the HTR solver to turbulent combustion in the presence of external electric fields. In particular, a new distributed Poisson solver compatible with heterogeneous architectures has been incorporated in the algorithm to compute the electric potential distribution in bi-periodic configurations. The drift fluxes of the electrically charged species are now included in the transport equations using a targeted essentially non-oscillatory scheme. A verification of these new features of the solver is provided using one-dimensional burner stabilized flames, whereas a three dimensional turbulent flame is utilized to discuss the scalability of the proposed numerical tool.

Keywords: Compressible reacting flows, GPUs, High-order numerics, Chemi-ionization, Ion-wind

PROGRAM SUMMARY

Program Title: Hypersonics Task-based Research solver
CPC Library link to program files:
Developer's respository link: `https://github.com/stanfordhpccenter/HTR-solver.git`
Code Ocean capsule:
Licensing provisions: BSD 2-clause
Programming language: Regent, C++, and CUDA
Journal reference of previous version:

- Di Renzo, M., Fu, L., & Urzay, J. (2020). HTR solver: An open-source exascale-oriented task-based multi-GPU high-order code for hypersonic aerothermodynamics. Computer Physics Communications 255, 107262.

- Di Renzo, M., & Pirozzoli, S. (2021). HTR-1.2 solver: Hypersonic Task-based Research solver version 1.2. Computer Physics Communications 261, 107733.

[*]Corresponding author.
E-mail address: mario.direnzo@cerfacs.fr

Does the new version supersede the previous version?: yes
Reasons for the new version: new features of the solver
Summary of revisions:

- normalization of the multicomponent mixture
- Lu & Law [1] 30 species chemical scheme added to the solver as a mixture model for combustion of methane in air
- TENO-LAD [2] implemented in the solver as an option for the inviscid fluxes discretization
- new FFT-based Poisson solver for bi-periodic problems
- calculation of transport properties using the (n,6,4) interaction theory for charge-neutral species interactions
- new optional module for ion-wind effects on charged species transport
- FFCM-1 [3] provided as a mixture model
- Boivin et al. [4] added as a mixture model for hydrogen combustion

Nature of problem (approx. 50-250 words):
This code solves the compressible multicomponent Navier-Stokes equations at high Mach numbers including finite-rate chemistry and complex chemical species transport. The implemented numerical methods are designed for direct numerical simulations (DNS) of high Reynolds number reacting flows, such as transitional and turbulent hypersonic boundary layers at high enthalpy and turbulent flames.

Solution method (approx. 50-250 words):
This code uses low-dissipation finite difference schemes for the spatial discretization of the conservation equations on Cartesian stretched grids. The time advancement is performed either with an explicit method, when the chemistry is slow and therefore does not introduce additional stiffness in the integration, or with an operator-splitting method that integrates the chemical production rates with an implicit discretization.

Additional comments including Restrictions and Unusual features (approx. 50-250 words):
The HTR solver builds on the runtime Legion [5] and is written in the programming language Regent [6] recently developed at Stanford University. Instructions for the installation of the components are provided in the README.md file enclosed with the HTR solver repository.

References

[1] Lu, T., & Law, C. K. (2008). A criterion based on computational singular perturbation for the identification of quasi steady state species: A reduced mechanism for methane oxidation with NO chemistry. Combustion and Flame, 154(4), 761-774.

[2] Peng, J., Liu, S., Li, S., Zhang, K., & Shen, Y. (2021). An efficient targeted ENO scheme with local adaptive dissipation for compressible flow simulation. Journal of Computational Physics, 425, 109902.

[3] G.P. Smith, Y. Tao, & H. Wang, Foundational Fuel Chemistry Model Version 1.0 (FFCM-1), `http://nanoenergy.stanford.edu/ffcm1`, 2016.

[4] Boivin, P., Jiménez, C., Sánchez, A. L., & Williams, F. A. (2011). An explicit reduced mechanism for H2–air combustion. Proceedings of the Combustion Institute, 33(1), 517-523.

[5] Legion web page: `https://legion.stanford.edu`

[6] Regent web page: `http://regent-lang.org`

1. Introduction

Developing efficient and scalable solvers is an established way to favor the advancement of Computational Fluid Dynamics (CFD). The advent of High Performance Computing (HPC) facilities with heterogeneous architecture, featuring many types of processors and memories, has significantly improved the throughput available to computational scientists on supercomputers but, at the same time, has posed additional difficulties in developing applications that are able to take advantage of these systems. The community of CFD has recently seen the development of many solvers that are targeted to these heterogeneous systems. AFiD-GPU [1] and CaNS-GPU [2] are two examples of GPU compatible open-source incompressible Navier–Stokes solvers that have been recently open sourced to the community. Similarly, few compressible Navier–Stokes solvers have been open sourced in the past few years. For instance, STREAmS [3] and OpenSBLI [4] are finite difference solvers for ideal gas mixtures. The first solver uses CUDA Fortran to generate GPU compatible binaries, whereas, OpenSBLI is written using Python-based code generator that automatically produces CPU and GPU compatible routines. Both these applications rely on static scheduling of the data communications across the nodes based on Massage Passing Interface (MPI) libraries and have proved to be efficient and scalable. However, the difficulties and potential performance issues that can be encountered in static scheduling of the data communications among the nodes and devices are bound to increase as the complexity of the HPC system architectures and the size of the calculations increases toward exascale computing.

An innovative framework designed for exascale computing is provided by the Legion runtime [5–7]. In this framework, applications are represented as a collection of tasks, which are functions that sequentially operate on a specified set of data. This representation allows the runtime to dynamically foresee the data dependencies among tasks, to organize them as a graph, to map this graph on the available computational resources, and to perform all the memory management operations including data communications among devices. In this way, many of the responsibilities of data management are relieved from the user, who can instead focus on developing the functionality of the application. The applicability of this programming model in the field of CFD has been recently shown in finite-difference solvers [8–10] and a discontinuous Galerkin solver [11]. In particular, the Hypersonic Task-based Research (HTR) solver [9, 10] is a finite-difference multicomponent compressible reacting flow solver designed to compute hypersonic flows at high-enthalpies leveraging HPC systems with heterogeneous architecture. One of the innovative aspects of this solver is its implementation with the Regent programming language [12]. This programming language is designed to efficiently produce applications compatible with the Legion programming model and takes care of automatically compiling binaries that are compatible with GPUs and CPUs.

The task-based approach has many features that ease the implementation of complex applications compatible with heterogeneous architectures. Among these features, the composability of task-graphs is particularly advantageous in assembling multi-physics solvers. In fact, task-graphs defined in various modules of a multi-physics solver can be automatically merged into a single task-graph by the Legion runtime. In this way, the runtime manages the various data structures required by each module ensuring correct synchronization of the components to preserve the correctness of the algorithm. This feature of the task-based programming model is exercised by the presented new version of the HTR solver to predict the behavior of electrified turbulent flames.

The prediction of the interaction between external electric fields and hydrocarbon flames is very challenging from a numerical point of view. In fact, though this technology could lead to significant improvements in active control of combustion, there are very few numerical investigations that have studied it. Predictive calculations of the impingement of an electric field on a flame involve phenomena occurring in a very broad spectrum of spacial and temporal scales and need complex transport models and chemical mechanisms. For this reason, most of the numerical studies present in the literature are based on one-dimensional calculations [13–17]. These studies have been utilized to assess the accuracy of transport and chemical production models. Two-dimensional calculations have been carried out using flamelet progress variable models [18–20], reduced chemistry [21, 22], and detailed chemistry consisting of about 60 species [23, 24]. Very recently, Belhi *et al.* [25] presented a three-dimensional calculation of a laminar Bunsen flame impinged by an electric field performed using a skeletal mechanism. Even though significant advances in the numerical methods for the simulation of electrified flames have been accomplished [26], the numerical prediction of turbulent flames impinged by electric field still appears very challenging even in academic configurations. In this work, an updated version of the HTR solver is presented with the intent of providing an efficient, opensource, and scalable tool that can be used to predict the behavior of turbulent electrified flames in bi-periodic geometries like 2D and 3D temporal reacting mixing layers.

The rest of the manuscript is organized as follows. Section 2 provides a description of the mathematical formulation utilized in the solver. The dimensionless formulation of the transport equations implemented in the new version of the solver is presented in Section 3. The numerical methods deployed to compute the electric drift fluxes of the charged species and the electric potential distribution are described in Section 4. Section 5 presents two validation tests performed on one-dimensional flames to assess the accuracy of the presented solver in predicting combustion with and without external electric fields. Scaling tests of the solver, performed both on GPU and CPU based supercomputers, using a three-dimensional turbulent flame impinged by an external electric field are provided in Section 6. Finally, Section 7 gives the concluding remarks on this study.

2. Mathematical formulation

The new version of the HTR solver presented in this manuscript numerically integrates the conservation equations of species partial densities, momentum, and total energy, namely

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot [\rho Y_i (\mathbf{u} + \mathbf{V}_i)] = \dot{w}_i \quad \text{for } i = 1, \dots, N_s, \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \nabla \cdot \bar{\bar{\tau}} - \rho_q \nabla \Phi, \tag{2}$$

$$\frac{\partial (\rho e_0)}{\partial t} + \nabla \cdot (\rho \mathbf{u} H) = \nabla \cdot \left(\lambda \nabla T + \bar{\bar{\tau}}\mathbf{u} - P\mathbf{u} - \sum_{i=1}^{N_s} \rho_i \mathbf{V}_i h_i \right) - \sum_{i=1}^{N_s} \rho_{q,i} (\mathbf{u} + \mathbf{V}_i) \cdot \nabla \Phi. \tag{3}$$

In these equations, ρ is the mixture density, Y_i are the mass fraction of each of the N_s species considered in the mixture model, $\mathbf{u}$ is the flow velocity vector, P is the static pressure, T is the static temperature, and e_0 is the total energy. The system of transport equations is closed with the equation of state for a multicomponent mixture of ideal gases,

$$P = \rho \mathrm{R}^0 T / \overline{W}, \tag{4}$$

where R^0 is the universal gas constant and $\overline{W} = (\sum_{i=1}^{N_s} Y_i/W_i)^{-1}$ is the mean molecular weight based on the individual weights W_i of each species, and with the Gauss law,

$$\epsilon_0 \nabla^2 \Phi + \rho_q = 0, \tag{5}$$

where Φ is the local electrostatic potential and ϵ_0 is the dielectric permittivity of vacuum. In this formulation, $\rho_q = \sum_{i=1}^{N_s} \rho_{q,i}$ is the electric charge density of the mixture obtained by the summation of the partial electric charge densities

$$\rho_q = \rho N_A e \frac{\mathcal{S}_i Y_i}{W_i}, \tag{6}$$

where N_A is the Avogadro number, e is the elementary electric charge value, and $\mathcal{S}_i$ is the number of elementary charges of the i-th species ($\mathcal{S}_i = 0$ for neutral species, $\mathcal{S}_i > 0$ for positive ions, and $\mathcal{S}_i < 0$ for negative ions and electrons).

The symbol $\overline{\overline{\boldsymbol{\tau}}}$ denotes, in the momentum conservation equation (2) and in the total energy conservation equation (3), the viscous stress tensor

$$\overline{\overline{\boldsymbol{\tau}}} = \mu \left[\nabla \mathbf{u} + \nabla \mathbf{u}^T - 2 \left(\nabla \cdot \mathbf{u} \right) \overline{\overline{\mathbf{I}}}/3 \right], \tag{7}$$

where $\overline{\overline{\mathbf{I}}}$ is the identity tensor and μ is the dynamic viscosity of the mixture. The stagnation energy is defined with the symbol $e_0 = e + |\mathbf{u}|^2/2$, with e being the specific internal energy of the mixture defined as

$$e = h - P/\rho = \sum_{i=1}^{N_s} Y_i h_i - P/\rho, \tag{8}$$

where h is the specific enthalpy of the mixture, and h_i is the partial specific enthalpy of species i given by

$$h_i = h_{i,\mathrm{ref}} + \int_{T_{\mathrm{ref}}}^{T} c_{p,i}(T') dT'. \tag{9}$$

In Eq. (9), $h_{i,\mathrm{ref}}$ is a reference value of the specific enthalpy taken at the reference temperature T_{ref}, whereas $c_{p,i}$ is a temperature-dependent specific heat of species i at constant pressure, which is evaluated using either the 7-coefficient or the 9-coefficient, depending on availability, NASA polynomials tabulated in [27]. The diffusion velocity vector $\mathbf{V}_i$, appearing in the species conservation equation (1), is defined as

$$\mathbf{V}_{\boldsymbol{i}} = -D_i \nabla \left(\ln X_i \right) - \mathcal{S}_i k_i \nabla \Phi + \sum_{j=1}^{N_s} Y_j \left[D_j \nabla \left(\ln X_j \right) + \mathcal{S}_j k_j \nabla \Phi \right]. \tag{10}$$

The first two terms on the right-hand side of equation (10) correspond, respectively, to a Fickian flux and to the electric drift flux [28, 29]. The third term on the right-hand side of equation (10) is a mass corrector that enforces the conservation of mass in the formulation [30, 31]. X_i, D_i, and k_i are the molar fraction, the mixture-averaged mass diffusivity, and the mixture-averaged electric mobility of species i, respectively.

The chemical production rates $\dot{w}_i$ are evaluated at runtime using Arrhenius based reactions contained in one of the multiple chemical reaction mechanisms provided with the solver.

Most of the transport properties mentioned in this section are evaluated using the algorithm described in Di Renzo *et al.* [9]. In particular, the dynamic viscosity is evaluated as function of the local dynamic viscosity of each component for the gas mixture using Wilke's rule [32]. Similarly, the thermal conductivity of the mixture, λ, is computed by averaging the local thermal conductivities in accordance with the formulation described in Mathur *et al.* [33].

The formulation in Bird *et al.* [34] is applied to compute the mixture-averaged mass diffusivity. The individual binary diffusivities D_{ij} and dynamic viscosities of neutral species are evaluated as a function of the local temperature and pressure from collision integrals based on the Stockmayer potential [35, 36].

The main additions to the mathematical formulation of the solver concern the transport properties of the charged species. The mixture-averaged electric mobilities of the heavy ions are determined as

$$k_i = \left(\sum_{j=1}^{N_s} \frac{X_j}{k_{ij}} \right)^{-1} \tag{11}$$

using the local binary electric mobilities k_{ij} [37]. The k_{ij} of heavy ions are computed from the binary mass diffusivities using the Einstein relation $D_{ij}/k_{ij} = k_B T/e$, where k_B is the Boltzmann constant. The (n,6,4) theory is deployed for the calculation of the binary mass diffusivities between heavy ions and neutral species [13, 38]. The electric mobility of free electrons is kept constant in the mixture. In particular, $k_{e^-} = 0.8 m^2/(Vs)$ is considered for the calculations presented in this work. As consequence, the mass diffusivity of the electrons is assumed equal to $D_{e^-} = k_{e^-} k_B T/e$.

The discussed formulation is targeted toward the simulation of weekly ionized gas ($\sum_{i=1}^{N_s} |\mathcal{S}_i| X_i \ll 1$), that are featured by electrified flames. For this reason, the contribution of collisions among charged species to the transport properties is not currently considered in the mathematical formulation of the solver in order to limit the computational complexity. Future developments of the HTR solver will include the effects of charged-charged species collisions in the evaluation of the transport properties and the introduction of more precise formulation for the estimation of free electrons transport properties [39].

3. Dimensionless formulation

The new version of the HTR solver released with this work features a new dimensionless formulation of the transport equations, which diminishes the roundoff errors involved in the calculation of reacting compressible flows. In particular, the reference values utilized for the normalization of velocity, energy, and density read as

$$u^{ref} = \sqrt{\frac{P^{ref}}{\rho^{ref}}}, \quad e^{ref} = \frac{P^{ref}}{\rho^{ref}}, \quad \text{and} \quad \rho^{ref} = \frac{P^{ref}\overline{W}^{ref}}{\mathcal{R}^0 T^{ref}}, \tag{12}$$

where $\overline{W}^{ref}$ is the reference mean molecular weight computed using a reference mixture composition X_i^{ref}. The dynamic viscosity, mass diffusivities, thermal conductivity are normalized with the following reference values

$$\mu^{ref} = \sqrt{\rho^{ref} P^{ref}} L^{ref}, \quad D_i^{ref} = \sqrt{\frac{P^{ref}}{\rho^{ref}}} L^{ref}, \quad \text{and} \quad \lambda^{ref} = \frac{\sqrt{\rho^{ref} P^{ref}} \mathcal{R}^0 L}{\overline{W}^{ref}}. \tag{13}$$

The constant pressure heat capacities are normalized with $c_p^{ref} = \frac{\mathcal{R}^0}{\overline{W}^{ref}}$, whereas $\dot{w}_i^{ref} = \frac{\sqrt{\rho^{ref} p^{ref}}}{L^{ref}}$ is the reference value for the species chemical production rates. The electric mobilities, the charge densities and the electric potential differences are normalized by

$$k_i^{ref} = \sqrt{\frac{\rho^{ref}}{P^{ref}}} \frac{N_a e L^{ref}}{\overline{W}^{ref}}, \quad \rho_q^{ref} = \frac{N_a e \rho^{ref}}{\overline{W}^{ref}}, \quad \text{and} \quad \Delta\Phi^{ref} = \frac{p^{ref} \overline{W}^{ref}}{\rho^{ref} N_a e}. \tag{14}$$

The user has to specify the reference pressure (P^{ref}), the reference temperature (T^{ref}), the reference length (L^{ref}), and a reference mixture composition (X_i^{ref}) to allow the solver to compute all the normalization factors and perform compressible reacting flow simulations, whereby most of the solved variables are of order one.

The main advantage of the implemented normalization method is that the transport equations described in the previous section read in exactly the same form in their dimensional and dimensionless formulations. For this reason, the implementation of the described normalization is very easy in existing solvers and allows users to easily opt in or out from its deployment.

4. Numerical methods

Most of the numerical methods utilized in this new version of the solver are already described in the previous publications related to the HTR solver [9, 10]. The two most relevant additions are related to the Poisson solver that has been included to compute the electrostatic potential from the Gauss law and the discretization method for the charged species flux due to the electric drift.

4.1. Poisson solver for distributed heterogeneous architectures

The parallel Poisson solver released with the new version of the HTR solver attached to this work is formulated to compute the electrostatic potential from the electric charge density distribution in computational domains periodic in x and z directions. The solution strategy that is implemented for this version of the solver consists in:

1. performing the two-dimensional direct Fast Fourier Transform (FFT) of the electric charge density on each of the x-z plane of the computational domain
2. solving the tri-diagonal systems of equations derived from the Gauss law, where the Fourier transform has been applied in the x and z direction, namely

$$\left(k_x^2 + k_z^2\right)\widehat{\Phi} + \frac{d\widehat{\Phi}}{dy} + \frac{\widehat{\rho_q}}{\epsilon_0} = 0, \tag{15}$$

   and where the derivative in y is discretized with a second-order centered scheme. In the previous equation, k_x and k_z are the wavenumbers in the x and z directions, respectively, and the hat over the symbols means that the corresponding quantities are Fourier transformed in the x and z directions. The Thomas algorithm is utilized to solve the tri-diagonal systems of equations at this step. In particular, the algorithm is parallelized in a way that each CPU or GPU thread computes the values of $\widehat{\Phi}$ along each y pencil of the computational grid.

3. performing the two-dimensional inverse FFT on the $\widehat{\Phi}$ computed from the solution of the tri-diagonal systems of equations at the previous point to obtain the instantaneous electric potential field.

The numerical solution obtained with this algorithm is second order accurate in space. It is noteworthy, that the discretization error introduced by this solution procedure is negligible considering the absence of high-wave number fluctuations in the electric potential field distribution.

The current implementation allows the user to specify either Dirichlet boundary conditions with constant value of Φ or a Robin boundary conditions, which imposes only the mean value of Φ, on the $y = y_{min}$ and $y = y_{max}$ planes.

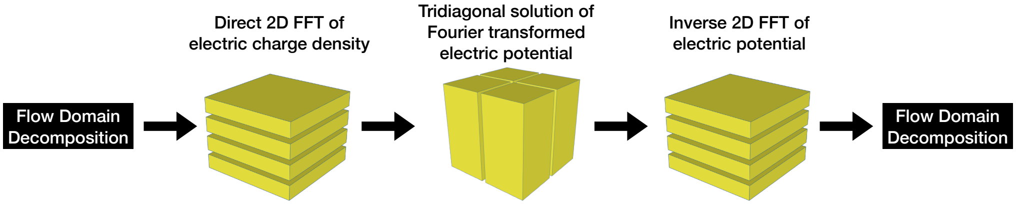

Flow Domain Decomposition
Direct 2D FFT of electric charge density
Tridiagonal solution of Fourier transformed electric potential
Inverse 2D FFT of electric potential
Flow Domain Decomposition

Figure 1: Schematics of the logical data partitions utilized by the Poisson solver algorithm.

From an implementation point of view, the direct and inverse FFTs are either performed on CPUs using the library FFTW [40] or on GPUs using the cuFFT library. In order to perform these transformations each node or each GPU, depending on the architecture of the machine, will need to contain some of the x-z data planes that compose the computational domain. Similarly, the solution of the tri-diagonal systems of equations requires each node (or GPU) to store a slab of the domain oriented in the y direction. For this reason, the outlined numerical procedure requires multiple communications and data layouts in order to be correctly executed. An efficient implementation of these procedures would not be trivial in a solver based on MPI. On the other hand, the rich data partitioning system provided by the Legion framework allows one to define the required partitions and data layout constraints, leaving the complex task of scheduling the communications to the runtime. In fact, the two data partitions (x-z planes and y slabs) depicted in the Figure 1 are described at the logical level using the partitioning operators available in Regent. The runtime foresees the data dependencies that exist between the task that perform the Poisson solver and the rest of the HTR solver algorithm and automatically schedules all the data communications needed to correctly advance the solution in time.

The presented algorithm adds a very low overhead in the organization of the task graph that describes the solver algorithm. In fact, only three tasks are needed to describe the entire Poisson solver. At the same time, the large data movements that are required to preserve correctness of this algorithm could create bottlenecks during the execution of the solver. The analysis of potential benefits derived by more complicated task graphs, which describe algorithms where the FFT transformations are executed direction by direction [1] or where different solution strategies are deployed [41], is deferred to future studies.

4.2. Discretization of ion-wind terms

The second addition to the HTR solver algorithm concerns the discretization of the electric drift fluxes of the electrically charged species. These terms in the transport equations, which model the ion-wind generated by the flame, are particularly stiff. In fact, the species electric drift velocity $\mathbf{V}_{d,i} = -\mathcal{S}_i k_i \nabla \Phi$ often exceeds values of thousands of meter per second even when mild electric fields are applied. Such high values of $\mathbf{V}_{d,i}$ pose two main challenges in the numerical integration of the transport equations [18, 19, 23, 26].

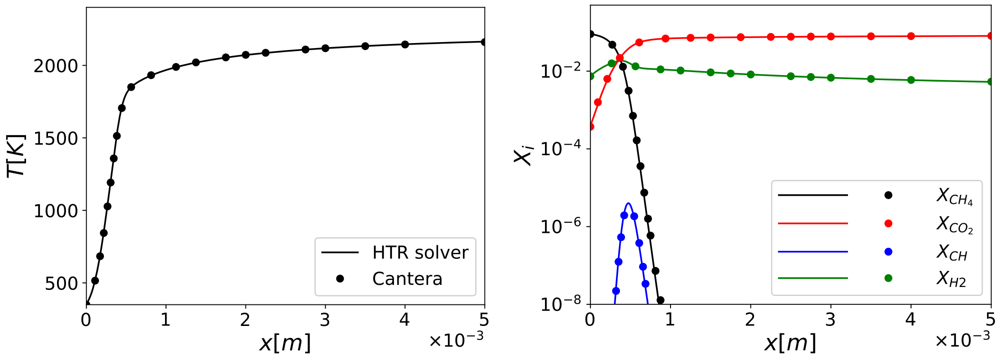
The first challenge in the integration of the charged species transport is related to the stability of the third-order temporal discretization scheme utilized in the HTR solver to advance the transport equations in time [42]. The Courant–Friedrichs–Lewy (CFL) number related to the electric drift velocity reads as $CFL_{d,i} = \Delta t \max(U_{d,i}/\Delta_x, V_{d,i}/\Delta_y, W_{d,i}/\Delta_z)$, where $U_{d,i}$, $V_{d,i}$, and $W_{d,i}$ are the Cartesian components of the vector $\mathbf{V}_{d,i}$, Δt is the time step used in the temporal discretization, and Δ_x, Δ_y, and Δ_z are the computational grid spacing in the three Cartesian directions, respectively. The explicit Runge–Kutta scheme utilized in the HTR solver remains stable for $CFL_{d,i} < 1$, hence very short time steps are required for the integration of electrified flames. These constraints in the temporal advancement could be relaxed by introducing implicit time advancement schemes for the electric drift and chemical production terms. These schemes will need to solve the non-linear problem constituted by the charged species electric drift, chemical production, and by the Gauss law. Considering the number of unknowns and the stiffness of the problem, even Newton-Krylov solvers, which appear to be the best candidates for this task [26], will need a significant number of iterations at each sub-step of the Runge-Kutta algorithm in order to converge. As a result, a significant increase of floating point operations and data communications to be performed within each time step is expected. For these reasons, the numerical efficiency of these complex schemes on distributed heterogeneous computers is still uncertain and will be subject of future investigations.

The second challenge in the numerical description of the ion-wind is due to the hyperbolicity of the electric drift. The high drift velocity developed by the charge species can generate sharp fronts in the molar fraction profiles, which are advected though the domain. For this reason, the spatial discretization scheme has to be capable of capturing the propagation of these sharp discontinuities in the charged species concentration fields. This capability is ensured by the use of sixth-order targetted essentially non oscilatory (TENO) [43] schemes in conjunction with a local Lax-Friederics flux splitting. The numerical procedure for the discretization of these fluxes consist of the following five steps in each of Cartesian direction (the procedure is explained for the x direction because the calculation for the other directions is analogous): i) computing the plus and minus ion-wind fluxes, which in the x direction read as $f^{\pm}_{d,i} = \rho Y_i U_{d,i} \pm \rho Y_i \lambda_i$, where λ_i is the maximum of the absolute value of $U_{d,i}$ across the six points stencil considered in the reconstruction scheme; ii) reconstructing the plus and minus fluxes using a TENO6-A scheme [44] at the intercell locations $i \pm 1/2$ where i are the cell indices in the computational domain; iii) computing the numerical fluxes at the intercell locations as $f_{d,i} = \left(f^{+}_{d,i} + f^{-}_{d,i}\right)/2$; iv) applying the mass corrector in Eq. 10; v) compute the contribution to the flux divergence of the drift fluxes as $df_{d,i}/dx = \left(f^{i+1/2}_{d,i} - f^{i-1/2}_{d,i}\right)/\Delta_x$.

Two types of boundary conditions are imposed for charged species on non-periodic boundaries of the domain. Zero-gradient conditions are imposed on surfaces where locally the normal drift velocity is oriented outward with respect to the computational domain. The molar fraction of the charged species is instead set to zero with a Dirichlet boundary condition if its local normal drift velocity is oriented toward the computational domain.

5. One-dimensional burner stabilized flames

The correctness and appropriateness of the mathematical and numerical formulation of the HTR solver in predicting combustion problems is assessed in this work by computing one-



(a)

(b)

Figure 2: Profiles of mixture temperature (a) and molar fractions (b) obtained by the HTR solver (lines) and by Cantera (symbols) for the electroneutral flame.

dimensional burner stabilized flames of methane and air. Though these testcases are only one dimensional, they have high complexity due to the sharp gradients of temperature and compositions generated in the proximity of the burner. For this reason, the accuracy in the prediction of the coupling between the chemical production and transport of the chemical species and of the temperature is fundamental to correctly capture the expected solution. The accuracy of the HTR solver is assessed in this section by comparing its solutions with the results of the well-known opensource solver Cantera [45].

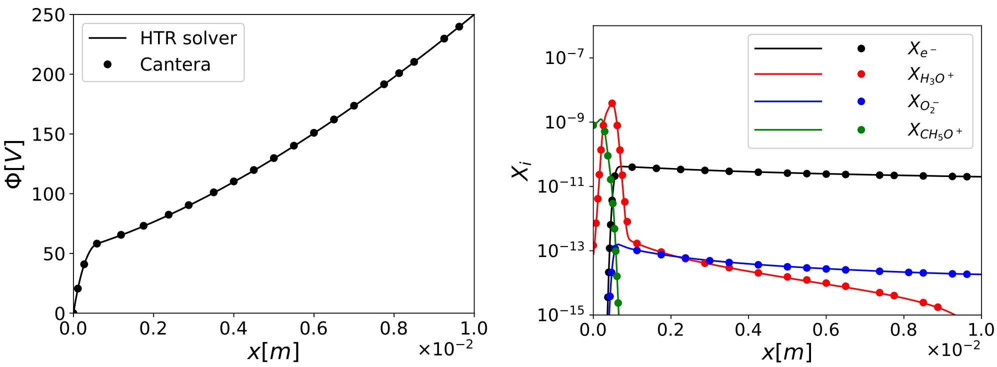
5.1. *Electroneutral flame*

The electroneutral burner stabilized flame utilized to verify the HTR solver in this work is inspired by the setup studied numerically and experimentally by Speelman *et al.* [15]. It consists of an heat flux burner that produces a flat flame. The burner is fed with a stoichiometric mixture of methane and air at ambient pressure, which flows at the laminar flame velocity developed by the same mixture at 298 K. The burner is kept at 350 K to stabilize the flat premixed flame. The computational setup considered in this work consists of a one dimensional domain with a length of 1 cm, which is discretized by 600 grid points. The reacting mixture is modeled using the chemical reaction mechanism composed by 30 species and 184 reactions presented by Lu and Law [46]. The computational solution is initialized with the constant mixture composition and temperature that would be achieved in burnt gasses and left free to evolve until steady state is achieved.

Figure 2 shows a comparison of the solution obtained using the HTR solver and Cantera. The two panels in the figures show the profiles of mixture temperature and the molar fractions of methane, carbon dioxide, CH, and H_2, across the flame. The excellent agreement shown in the pictures demonstrates that the mathematical and numerical formulation of the HTR solver is capable of correctly capturing the combustion phenomena occurring in this flame.

5.2. *Electrified flame*

The premixed flame setup discussed in the previous section has been modified to include the effects of an impinging electric field. Similarly to the experiments in Speelman *et al.* [15], two electrodes are considered in this setup. The first electrode is positioned at the edge of the



(a)

(b)

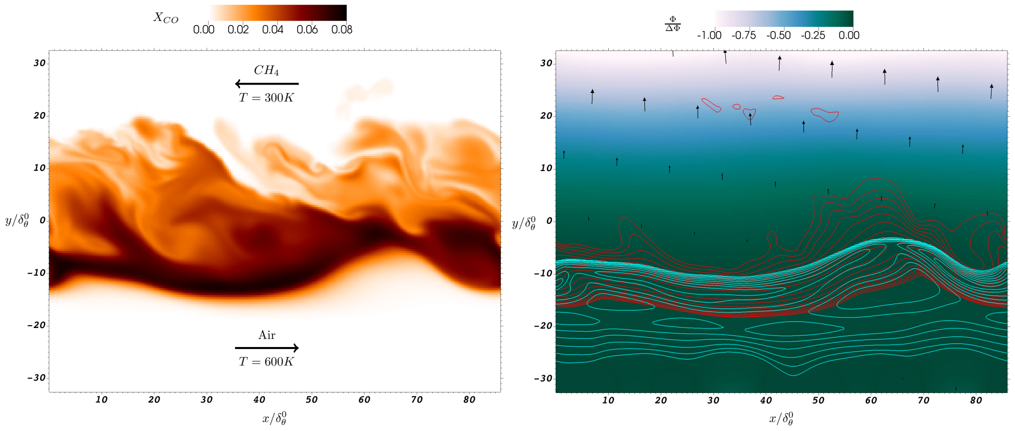
Figure 3: Profiles of electrostatic potential (a) and charged species molar fractions (b) obtained by the HTR solver (lines) and by Cantera (symbols) for the electrified flame.

burner and is grounded. The second electrode is positioned at a distance of 1 cm downstream of the burner exit and is attached to an ideal voltage generator that produces a difference of potential $\Delta\Phi = 250\,\mathrm{V}$. The calculations are executed using a chemical mechanism composed of 43 species, including 4 positive ions (H_3O^+, CHO^+ CH_5O^+, $C_2H_3O^+$), 2 negative ions (OH^-, $O_2{}^-$), and the free electrons, which react with 464 reactions. This chemical scheme, which is provided with this software release, is obtained by the analytic reduction of the detailed combustion mechanism proposed by Ranzi *et al.* [47] supplemented by the chemi-ionization mechanism proposed by Prager *et al.* [13]. The comparison of the temperature and neutral species molar fraction profiles is not included in this manuscript for brevity as these quantities would show the same kind of agreement as in Figure 2. The two panels in Figure 3 show the distribution of the electric potential and of the main charged species molar fractions across the computational domain. The positive polarity of the impinging electric field forces the positive ions to drift toward the burner, while negative ions and electrons are attracted by the downstream electrode. Though the charged species represent only a minor part of the mixture ($X_i \ll 1$), the electric charge that they carry is able to generate significant modifications of the electric potential distribution. In fact, the electric potential distribution, which would have a linear scaling between the two electrodes in the absence of electric charges in the domain, shows significant curvature in the proximity of the burner.

The excellent agreement shown between the results of the HTR solver and of Cantera further demonstrates the appropriateness of the mathematical and numerical formulation proposed in this work to predict the combustion physics even in the presence of electrically charged species and electric fields.

6. 3D temporal mixing layer impinged by an electric field

The capabilities of the developed solver to perform simulations of electrified combustion in high Reynolds number conditions are demonstrated in this work by computing the evolution of a reacting turbulent temporal mixing layer impinged by an external electric field. The examined configuration, which is shown in Figure 4a, consists of two flows at atmospheric pressure with different composition, which mix and react, while flowing in opposite directions. The first gas



(a)

(b)

Figure 4: Contour plots of the instantaneous carbon monoxide molar fraction (a) and normalized electric potential (b) distributions in the reacting temporal mixing layer. In panel b, the plotted vectors are oriented and scaled with the local electric field whereas the red and cyan iso-lines are drawn based on the molar fraction of H_3O^+ and of OH^-, respectively.

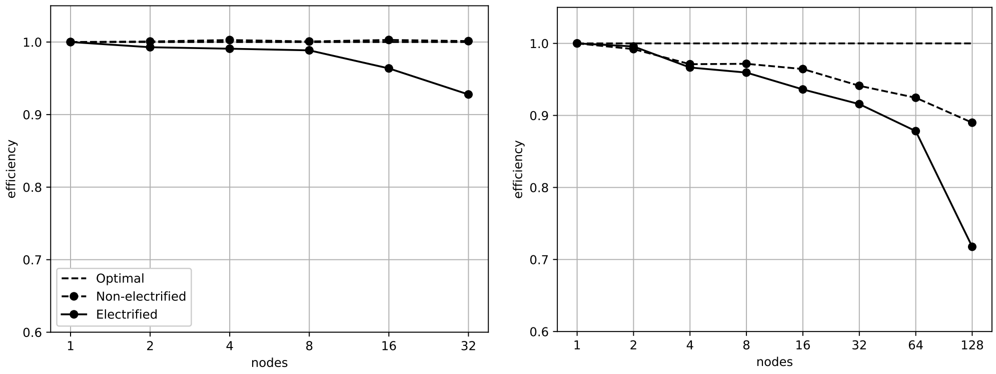
mixture is composed of pure methane at 300 K and has a velocity $-\Delta U/2$ along the x direction. The second mixture is dry air at 600 K and flows with a velocity $\Delta U/2$. The analyzed configuration has a convective Mach number $Ma_c = \Delta U/a_{Air} = 0.2$ and is initialized with a Reynolds number based on the momentum thickness $Re_\theta^0 = \rho_{Air}\Delta U\theta^0/\mu_{Air} = 320$. In these conditions, the density ratio of the two mixtures $\rho_{\mathrm{CH}_4}/\rho_{Air} \sim 1$, therefore the two opposite flows have similar momentum. Thermochemical and velocity fields within the mixing layer are initialized using the solution of a one dimensional mixing layer, whereby the mixture is ignited. Transition to turbulence is favored by a random velocity disturbance based on the Passot-Pouquet spectrum which is imposed around the mixing region at the beginning of the three dimensional simulation [48, 49]. The mixing layer is impinged by an external electric field whose mean directions is aligned with the y axis. The mean intensity of the applied electric field E_0 is determined such that the interaction parameter $\Xi = E_0^2\epsilon_0/\left(\rho_{Air}\Delta U^2\right) = 10^{-5}$.

The computational domain utilized to compute this flow has size $86\theta^0 \times 64.5\theta^0 \times 43\theta^0$ in the streamwise, mixing normal, and spanwise directions, where θ^0 is the momentum thickness computed at the beginning of the calculation. Periodic boundary conditions are considered in the x and z directions, while multicomponent one-dimensional non-reflecting characteristic outflow conditions are imposed at the surfaces $y = \pm 32.25\theta^0$ [50]. Robin boundary conditions are utilized for the electric potential field at the two non-periodic boundaries. The described computation domain is discretized with 256 × 196 × 128 points uniformly distributed along the streamwise, mixing normal, and spanwise directions, respectively.

The examined configuration has a stoichiometric mixture fraction equal to 0.07, entailing the formation of the flame on the air side of mixing layer. For this reason, the reacting layer of the flame, where the molar fraction of CO peaks, is positioned in the bottom half of the figure 4a. In this area, the high temperature developed within the flame locally increases the viscosity and decreases the density of the gas mixture, therefore lowering the local Reynolds number. Hence, the turbulence is localized only on the fuel side of the mixing layer, where the mixture properties favor its formation. At the same time, the chemi-ionization processes lead to the formation of electrically charged species inside the reacting layer of the flame. These charged species drift toward the positive (negative) direction of the y axis, if they are positively (negatively) charged, as shown by the isolines of H_3O^+ and OH^- molar fractions in Figure 4b. The electric drift of the ions generates positive and negative peaks of net charge density around the reacting layer of the flame. These peaks modify the electric potential distribution determining an electro-neutral region around the flame. However, the high mobility of the negative species, and particularly of free-electrons, determines a lower concentration of negative ions in the domain, therefore the electric potential of the electro-neutral region is similar to the positive electrode (bottom end in Figure 4b). For this reason, most of the electric potential difference imposed at the boundaries of the domain is concentrated between the flame and the negative electrode (top end in Figure 4b), where intense electric field is formed as shown by the longer vector arrows drawn in the figure.

The scalability of the presented release of the HTR solver has been assessed for the described electrified reacting mixing layer setup. This configuration is significantly more challenging than the scaling tests of the HTR solver for electroneutral setups previously presented in Di Renzo *et al.* [9] and also proposed in this work for reference. In fact, the coupling of the newly implemented Poisson solver triggers complex data communication patters within the time-advancement algorithm of the transport equations. Two very different facilities have been utilized during these tests in order to assess the performance and portability of the solver on supercomputers with and without GPUs. Marconi 100, which is installed at the HPC center of CINECA in Italy and currently is ranked 11^{th} in the Top500, has been utilized to test the scalability of the solver on multi-GPU machines. In fact, four NVIDIA Volta V100 GPUs and two IBM POWER9 AC922 CPUs are installed on each node of Marconi 100. These components produce a computational power of about 32 TFlops. Conversely, the supercomputer Quartz at Lawrence Livermore National Laboratory has been utilized to test the solver in CPU-based computing architectures. Two 18-cores Intel Xeon E5-2695 are installed on the nodes of this supercomputer. The powerful nodes of Marconi 100 allow the user to consider setups with many more grid points per node than Quartz. For this reason, the same calculation performed on the two machines will have much less communications with larger size on Marconi 100, whereas it will use more nodes which will communicate fewer data on Quartz. These different balances between communications and local floating point operations force the runtime to apply different strategies in order to hide the overhead of data communications. It is noteworthy that the results shown in this section have been obtained by porting the exact same version of the solver on these two different supercomputers without implementing any machine specific tuning. In this way, the results presented in this work are representative of the performance that could be achieved by a new user who installs the software for the first time on similar HPC facilities.

Figure 5 shows the efficiency achieved by the HTR solver during a weak scaling test in electrified and electroneutral conditions. This test has been conducted by assigning a computational grid of the size $128 \times 32 \times 256$ points in the x, y, and z directions to each GPU available on the nodes of Marconi 100 and of the size $32 \times 64 \times 32$ to each of the CPUs available on the nodes of Quartz. The solver shows optimal efficiency for an electrified case up to eight GPU nodes (32 GPUs) for where the about 33 million grid points are solved. The efficiency decreases linearly for larger node counts, while still remaining above 90% for 32 nodes (128 GPUs). At the same time, the non-electrified setup, where the Poisson solver is not deployed, has optimal efficiency for the entire range of problem sizes tested in this study. The largest configuration considered in this scaling tests consists of about 134 million grid points, which correspond to about 6.5 billion degrees of freedom, considering that the chemistry model utilized in this setup involves 43 species. A similar trend of the efficiency is encountered for the scaling test of Quartz. An



(a)

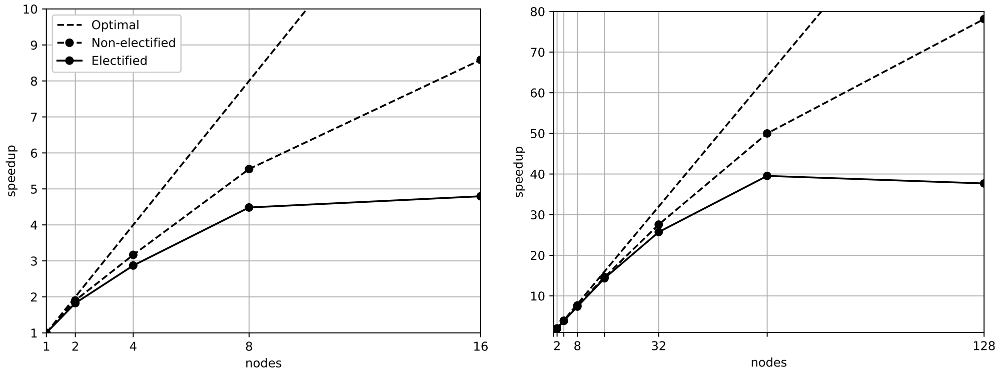
(b)

Figure 5: Efficiency of the HTR solver during a weak scaling performed using a three-dimensional electrified and electroneutral flame configuration on Marconi 100 (a) and on Quartz (b).

efficiency higher than 88% is attained up to 64 nodes in the electrified case, which solve about 8 million grid points. A sudden drop to about 72% is observed for 128 nodes when the time consumed by data communications becomes too large to be hidden by the runtime. The non-electrified setup is in general more efficient than the electrified counterpart whereby an about 89 % is shown for the 128 nodes setup.

Figure 6 shows the speedup achieved by the HTR solver during a strong scaling test based on the described electrified reacting mixing layer configuration. This strong scaling test is performed on Marconi 100 and Quartz by associating a computational domain composed of $256\times64\times256$ and 128^3 points, respectively, to an increasing number of nodes. The solver shows an ideal speedup on Marconi100 up to 2 nodes (8 GPUs) for the considered computational grid in the electrified setups where the Poisson solver is deployed. At higher node counts, the reduction of floating point operations assigned to each GPU and the increase in the data communication overhead lead to a degradation of the solver performance and, eventually, to a saturation of the speedup at 8 nodes (32 GPUs). Such saturation is delayed in non-electrified conditions because of the lower data movement. Similarly, the results obtained on Quartz show an almost ideal speedup up to 16 nodes and a saturation at 64 nodes. It is noteworthy that a potential cause of the different scalability limit encountered between GPU-based and CPU-based architectures is the lack of GPU-direct communications in the utilized version of the Legion runtime. This kind of communications significantly augment the efficiency of data transfer among GPUs and will be effortlessly included in the solver when they will be supported by future releases of the Legion runtime.

Significant improvements in the results of weak and strong scaling tests can be also achieved by implementing more sophisticated task mapping techniques. In fact, the Legion runtime offers many different options to modify the way data are laid out in memory and how the tasks are assigned to the available computational resources. Future work will be dedicated to exploring new mapping algorithm that could reduce the data communication overhead and improve the scalability of the solver.



(a) (b)

Figure 6: Speedup of the HTR solver during a strong scaling test performed using a three-dimensional electrified and non-electrified flame configuration on Marconi 100 (a) and on Quartz (b).

7. Conclusions

A new version of the Hypersonic Task-based Research (HTR) solver is presented with this manuscript. This new version contains major updates in the formulation as well as the inclusion of new capabilities. In particular, a dimensionless formulation of the transport equations is introduced in the solver to improve the quality of its numerical solutions. The chosen dimensionless formulation has the main advantage of keeping the same form of the transport equations even for multicomponent flows. For this reason, users can decide to deploy the dimensionless formulation without modifying the implementation of flux reconstruction of the solver. New mixture models suitable for the calculation of methane and hydrogen chemistry with various levels of detail have been included in the provided source files. Additionally, the HTR solver has been upgraded to consider the effects of ion wind on charged species distribution. To this end, a GPU-compatible Poisson solver suitable for bi-periodic computational domains has been implemented in the software framework and interfaced with the algorithm.

The new additions to the solver have been tested using one-dimensional burner stabilized flames. The accuracy of the obtained solutions has been assessed by comparing it with the results of calculations performed in the open-source solver Cantera. The calculations described in this manuscript show that the HTR solver is capable of accurately predict the combustion process and the electrically charged species distribution in the flame. Moreover, the capabilities of the HTR solver to compute three dimensional turbulent flames impinged by external electric fields are shown by performing calculations of a temporal reacting mixing layer. This setup has been used to perform scalability tests of the solver, which has shown a parallel efficiency, measured using a weak scaling test, above 93% on up to 128 GPUs and above 88% up to 64 CPU nodes.

While the new capabilities of the HTR solver can be readily used by new users to investigate the behavior of turbulent flames even in the presence of external electric fields, the presented version offers a solid software base for further developments of this tool. For instance, the current implementation provides a flexible infrastructure where complex descriptions of the transport properties of the reacting species, and particularly of the charged species [16, 39], could be incorporated. On the other hand, future developments of the solver will certainly focus on lifting the constraints on the complexity of the geometry posed by the current numerical methods. In this context, the deployment of curvilinear coordinates formulation [51, 52] of the differential operator utilized to discretize the transport equations together with more general Poisson solver, which will not necessarily rely on fast Fourier transforms, will be investigated.

Acknowledgments

This investigation is funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 898458. The author acknowledges the CINECA award under the ISCRA initiative IsB22_3dSBLI, for the availability of high performance computing resources and support, as well as the computing resources provided by the US Department of Energy, National Nuclear Security Administration under Award Number DE-NA0003968 within the PSAAP III (INSIEME) Program at Stanford University.

References

[1] X. Zhu, E. Phillips, V. Spandan, J. Donners, G. Ruetsch, J. Romero, R. Ostilla-Mónico, Y. Yang, D. Lohse, R. Verzicco, M. Fatica, R. J. Stevens, Computer Physics Communications 229 (2018) 199–210. https://doi.org/10.1016/j.cpc.2018.03.026.

[2] P. Costa, E. Phillips, L. Brandt, M. Fatica, Computers and Mathematics with Applications 81 (2021) 502–511. https://doi.org/10.1016/j.camwa.2020.01.002.

[3] M. Bernardini, D. Modesti, F. Salvadore, S. Pirozzoli, Computer Physics Communications 263 (2021) 107906. https://doi.org/10.1016/j.cpc.2021.107906.

[4] D. J. Lusher, S. P. Jammy, N. D. Sandham, Computer Physics Communications 267 (2021) 108063. https://doi.org/10.1016/j.cpc.2021.108063.

[5] M. Bauer, S. Treichler, E. Slaughter, A. Aiken, Legion: Expressing locality and independence with logical regions, SC '12: International Conference for High Performance Computing, Networking, Storage and Analysis (2012), IEEE.

[6] S. Treichler, M. Bauer, A. Aiken, Realm: An event-based low-level runtime for distributed memory architectures, Parallel Architectures and Compilation Techniques - Conference Proceedings, PACT (2014), Institute of Electrical and Electronics Engineers Inc.

[7] M. Bauer, W. Lee, E. Slaughter, M. Di Renzo, G. Shipman, P. Mccormick, M. Garland, A. Aiken, Scaling Implicit Parallelism via Dynamic Control Replication, Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (2021), Association for Computing Machinery, Republic of Korea.

[8] S. Treichler, M. Bauer, A. Bhagatwala, G. Borghesi, R. Sankaran, H. Kolla, P. S. McCormick, E. Slaughter, W. Lee, A. Aiken, J. Chen, S3D-Legion: An Exascale Software for Direct Numerical Simulation of Turbulent Combustion with Complex Multicomponent Chemistry, Exascale Scientific Applications, Chapman and Hall/CRC, 2017, pp. 257–278.

[9] M. Di Renzo, L. Fu, J. Urzay, Computer Physics Communications 255 (2020) 107262. https://doi.org/10.1016/j.cpc.2020.107262.

[10] M. Di Renzo, S. Pirozzoli, Computer Physics Communications 261 (2021) 107733. https://doi.org/10.1016/j.cpc.2020.107733.

[11] K. Bando, S. Brill, E. Slaughter, M. Sekachev, A. Aiken, M. Ihme, Development of a discontinuous Galerkin solver using Legion for heterogeneous high-performance computing architectures, AIAA Scitech 2021 Forum (2021), American Institute of Aeronautics and Astronautics, Reston, Virginia.

[12] E. Slaughter, W. Lee, S. Treichler, M. Bauer, A. Aiken, Regent: A High-Productivity Programming Language for HPC with Logical Regions, SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (2015), Institute of Electrical and Electronics Engineers.

[13] J. Prager, U. Riedel, J. Warnatz, Proceedings of the Combustion Institute 31 (2007) 1129–1137. https://doi.org/10.1016/j.proci.2006.07.141.

[14] N. Speelman, M. Kiefer, D. Markus, U. Maas, L. P. H. de Goey, J. A. van Oijen, Proceedings of the Combustion Institute 35 (2015) 847–854. https://doi.org/10.1016/j.proci.2014.05.067.

[15] N. Speelman, L. P. H. de Goey, J. A. van Oijen, Combustion Theory and Modelling 19 (2015) 159–187. https://doi.org/10.1080/13647830.2014.998712.

[16] J. Han, M. Belhi, F. Bisetti, Combustion Theory and Modelling 19 (2015) 744–772. https://doi.org/10.1080/13647830.2015.1090018.

[17] J. Han, M. Belhi, T. A. Casey, F. Bisetti, H. G. Im, J.-Y. Chen, Proceedings of the Combustion Institute 36 (2017) 1241–1250. https://doi.org/10.1016/j.proci.2016.05.056.

[18] M. Belhi, P. Domingo, P. Vervisch, Combustion and Flame 157 (2010) 2286–2297. https://doi.org/10.1016/j.combustflame.2010.07.007.

[19] M. Belhi, P. Domingo, P. Vervisch, Combustion Theory and Modelling 17 (2013) 749–787. https://doi.org/10.1080/13647830.2013.802415.

[20] M. Di Renzo, P. De Palma, M. D. de Tullio, G. Pascazio, Computers & Fluids 157 (2017) 14–27. https://doi.org/10.1016/j.compfluid.2017.08.024.

[21] M. Belhi, B. J. Lee, F. Bisetti, H. G. Im, Journal of Physics D: Applied Physics 50 (2017) 494005. https://doi.org/10.1088/1361-6463/aa94bb.

[22] A. Sayed-Kassem, A. Elorf, P. Gillon, M. Idir, B. Sarh, V. Gilard, International Communications in Heat and Mass Transfer 122 (2021) 105167. https://doi.org/10.1016/j.icheatmasstransfer.2021.105167.

[23] M. Di Renzo, J. Urzay, P. De Palma, M. D. de Tullio, G. Pascazio, Combustion and Flame 193 (2018) 177–191. https://doi.org/10.1016/j.combustflame.2018.03.001.

[24] M. Di Renzo, G. Pascazio, J. Urzay, Combustion and Flame 205 (2019) 231–240. https://doi.org/10.1016/j.combustflame.2019.04.004.

[25] M. Belhi, B. J. Lee, M. S. Cha, H. G. Im, Combustion and Flame 202 (2019) 90–106. https://doi.org/10.1016/j.combustflame.2019.01.005.

[26] L. Esclapez, V. Ricchiuti, J. B. Bell, M. S. Day, Combustion Theory and Modelling 24 (2020) 194–220. https://doi.org/10.1080/13647830.2019.1668060.

[27] B. J. McBride, M. J. Zehe, S. Gordon, NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species, Technical Report, NASA, 2002.

[28] C. F. Curtiss, J. O. Hirschfelder, The Journal of Chemical Physics 17 (1949) 550–555. https://doi.org/10.1063/1.1747319.

[29] T. Pedersen, R. C. Brown, Combustion and Flame 94 (1993) 433–448. https://doi.org/10.1016/0010-2180(93)90125-M.

[30] T. P. Coffee, J. M. Heimerl, Combustion and Flame 43 (1981) 273–289. https://doi.org/10.1016/0010-2180(81)90027-4.

[31] A. Ern, V. Giovangigli, Multicomponent transport algorithms, volume 24 of *Lecture Notes in Physics Monographs*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[32] C. R. Wilke, The Journal of Chemical Physics 18 (1950) 517–519. https://doi.org/10.1063/1.1747673.

[33] S. Mathur, P. K. Tondon, S. C. Saxena, Molecular Physics 12 (1967) 569–579. https://doi.org/10.1080/00268976700100731.

[34] R. B. Bird, W. E. Stewart, E. N. Lightfoot, Transport phenomena, John Wiley & Sons, Inc., New York, 1960.

[35] J. O. Hirschfelder, C. F. Curtiss, R. B. Bird, Molecular theory of gases and liquids., John Wiley & Sons, 1964.

[36] L. Monchick, E. A. Mason, The Journal of Chemical Physics 35 (1961) 1676–1697. https://doi.org/10.1063/1.1732130.

[37] E. A. Mason, E. W. McDaniel, Transport Properties of Ions in Gases, Wiley, New York, 1988.

[38] J. Prager, Modeling and Simulation of Charges Species in Lean Methane-Oxygen Flames, Ph.D. thesis, 2005.

[39] F. Bisetti, M. El Morsli, Combustion and Flame 159 (2012) 3518–3521. https://doi.org/10.1016/j.combustflame.2012.08.002.

[40] M. Frigo, S. G. Johnson, Astrophysics Source Code Library (2012) ascl–1201.

[41] D. Fortunato, A. Townsend, IMA Journal of Numerical Analysis 40 (2020) 1994–2018. https://doi.org/10.1093/IMANUM/DRZ034.

[42] S. Gottlieb, C.-W. Shu, E. Tadmor, SIAM Review 43 (2001) 89–112.

[43] L. Fu, X. Y. Hu, N. A. Adams, Journal of Computational Physics 305 (2016) 333–359. https://doi.org/10.1016/j.jcp.2015.10.037.

[44] L. Fu, X. Y. Hu, N. A. Adams, Communications in Computational Physics 26 (2019) 311–345. https://doi.org/10.4208/cicp.OA-2018-0145.

[45] D. G. Goodwin, R. L. Speth, H. K. Moffat, B. W. Weber, Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes, https://www.cantera.org, 2021. Version 2.5.1.

[46] T. Lu, C. K. Law, Combustion and Flame 154 (2008) 761–774. https://doi.org/10.1016/j.combustflame.2008.04.025.

[47] E. Ranzi, A. Frassoldati, A. Stagni, M. Pelucchi, A. Cuoci, T. Faravelli, Reduced kinetic schemes of complex reaction systems: Fossil and biomass-derived transportation fuels, volume 46, 2014.

[48] C. Pantano, S. Sarkar, Journal of Fluid Mechanics 451 (2002) 329–371. https://doi.org/10.1017/s0022112001006978.

[49] C. Pantano, S. Sarkar, F. A. Williams, Journal of Fluid Mechanics (2003) 291–328. https://doi.org/10.1017/S0022112003003872.

[50] N. Okong'o, J. Bellan, Journal of Computational Physics 176 (2002) 330–344. https://doi.org/10.1006/jcph.2002.6990.

[51] N. A. Adams, Journal of Fluid Mechanics 420 (2000) 47–83. https://doi.org/10.1017/S0022112000001257.

[52] S. Pirozzoli, Journal of Computational Physics 230 (2011) 2997–3014. https://doi.org/10.1016/j.jcp.2011.01.001.