An MPI-CUDA approach for hypersonic flows with detailed state-to-state air kinetics using a GPU cluster

Giuseppe Pascazio, Francesco Bonelli

An MPI-CUDA approach for hypersonic flows with detailed state-to-state air kinetics using a GPU cluster

Francesco Bonelli[a,*], Michele Tuttafesta[b], Gianpiero Colonna[c], Luigi Cutrone[d], Giuseppe Pascazio[a]

[a] *DMMM& CEMeC, Politecnico di Bari, via Re David 200, 70125, Bari, Italy*
[b] *Liceo Scientifico Statale "L. da Vinci", Via Cala dell'Arciprete 1 - 76011 Bisceglie (BT), Italy*
[c] *CNR-IMIP, via Amendola 122/D - 70126 Bari (Italy)*
[d] *Centro Italiano Ricerche Aerospaziali (CIRA), Capua, 81043, Italy*

Abstract

This paper describes the most advanced results obtained in the context of fluid dynamic simulations of high–enthalpy flows using detailed state-to-state air kinetics. Thermochemical non-equilibrium, typical of supersonic and hypersonic flows, was modelled by using both the accurate state-to-state approach and the multi-temperature model proposed by Park. The accuracy of the two thermochemical non-equilibrium models was assessed by comparing the results with experimental findings, showing better previsions provided by the state-to-state approach. To overcome the huge computational cost of the state-to-state model, a multiple-nodes GPU implementation, based on an MPI-CUDA approach, was employed and a comprehensive code performance analysis is presented. Both the pure MPI-CPU and the MPI-CUDA implementations exhibit excellent scalability performance. GPUs outperform CPUs computing especially when the state-to-state approach is employed, showing speed-ups, of the single GPU with respect to the single-core CPU, larger than 100 in both the case of one MPI process and multiple MPI process. https://doi.org/10.1016/j.cpc.2017.05.019

Keywords: multi-GPU, GPU cluster, MPI-CUDA, Hypersonic flows, Air state-to-state chemical kinetics, multi-temperature

1. Introduction

Simulations of high–enthalpy flows are needed in order to predict the thermodynamic conditions around space capsule entering planetary atmospheres [1] or around long-range hypersonic air-breathing vehicles developed for future civil transcontinental flights [2]. A complete and correct knowledge of the key phenomena occurring in these flows can help to improve the structural integrity, the flight control, the aerodynamic efficiency of hypersonic vehicles and to estimate the global cost of space missions and civil flights [3].

At hypersonic speed, a strong shock wave with very high temperature (average of about 10000 K) is formed, activating chemical reactions which produce relevant quantities of atoms, ions and electrons that cannot be neglected in characterizing the flow properties. Moreover, at such high speeds, the relaxation time of the vibrational degrees of freedom and of the chemical reactions are of the same order as the fluid dynamic characteristic time, making the shock wave a system in thermochemical non-equilibrium.

A simple, quick and ubiquitous approach to non-equilibrium is the multi-temperature (MT) model proposed by Park [4], assigning a single temperature for translational and rotational degrees of freedom, considered in thermal equilibrium. On the other end, vibrational levels, following a Boltzmann distribution,

*Corresponding author
Email addresses: bonellifra@alice.it (Francesco Bonelli), micheletuttafesta@libero.it (Michele Tuttafesta), gianpiero.colonna@cnr.it (Gianpiero Colonna), L.Cutrone@cira.it (Luigi Cutrone), giuseppe.pascazio@poliba.it (Giuseppe Pascazio)

are characterized by a different temperature which evolves according to the Landau-Teller law. The chemical model assumes Arrhenius type rate coefficients, function of an effective temperature, calculated as the weighted geometrical mean of translational and vibrational temperatures, to account for higher rates from excited vibrational levels. The model has been completed by adding a term to account for the vibrational energy variation due to chemical reactions, assuming that in reactions producing and consuming molecules, the energy exchanged is that of a vibrational distribution at T_v [5]. However, the strong synergy between highly excited vibrational levels and chemical reactions makes not realistic the assumption of Boltzmann vibrational distribution, resulting in an underestimation or overestimation of multi-temperature rates, depending if the flow is in recombination or dissociation regime [6–9].

Unlike the multi-temperature models, the state-to-state (StS) approach [10–12] is able to determine the distribution of internal states even when it deviates from the Boltzmann one. In this kind of models each vibrational level is regarded as a separate species that evolves under the action of a series of elementary processes which take into account the vibrational-vibrational (VV) and vibrational-translational (VT) energy exchanges. Obviously, the number of species involved is much higher than that of classical multi-temperature models. For example, for an air mixture, without ions and electrons, macroscopic models use five species (N_2 , O_2, NO, N, O) along with three vibrational temperatures and 17 reactions (including vibrational relaxation terms) while the StS model employed in this work uses 118 species (68 vibrational levels for N_2, 47 levels for O_2 plus the species N, O and NO) and a total of about 10000 elementary processes. As a consequence of the large computational resources, the StS approach has been applied only to 1D flow, with very few exceptions [13, 14].

A possible strategy in order to perform 2D and 3D simulations is to develop more accurate reduced models based on the state-to-state vibrational kinetics [7, 15, 16]. Our group is exploring a different approach trying to demonstrate the feasibility of 2D fluid dynamic simulations coupled with detailed state-to-state air kinetics by using the growing computational power of the Graphics Processing Units (GPU) [17, 18]. Indeed, in the last few years there has been a growing interest in exploit the huge computational power of GPUs thanks also to new programming languages such as CUDA C and OpenCL that have facilitated general-purpose computing on GPU (GPGPU). Several works in a variety of scientific fields not only have shown promising speed-up but in several cases have demonstrated that GPUs far exceed CPUs performance [17–34]. GPUs not only beat CPUs in terms of absolute performance but above all in terms of power efficiency. The high efficiency of GPUs is confirmed by the November 2016 Green 500 list [35] where the top two positions are awarded to clusters powered with Nvidia GPUs. Therefore, the most powerful supercomputers of the future, like Summit [36], which is expected in 2018 at Oak Ridge National Laboratory, and Sierra [37], which is expected in 2017-2018 at Lawrence Livermore National Laboratory, will be equipped with GPUs and the use of GPGPU will become mandatory.

The main contribution of the present work is to perform, for the first time, a realistic 2D hypersonic flow simulation coupled with a detailed state-to-state air kinetics using GPUs, extending the previous single-node multi-GPU implementation [17, 18] to efficiently scale across a multiple-nodes GPU cluster by using an MPI-CUDA approach [32, 34, 38, 39].

The work is organized as follows: in Section 2, the reactive fluid dynamic model is presented. Two non-equilibrium thermochemical models for a neutral air mixture, i.e. the detailed StS kinetics and the Park multi-temperature model, are described. In Section 3, the numerical approach used to solve the governing equation, the MPI-CUDA implementation along with the details of the hardware and software employed, and the generic 2D configuration used in this work are given. In section 4, a realistic case study is investigated and a detailed code performance analysis is provided. Finally, the conclusions are summarized.

2. Governing equations and thermodynamic models

In this section the governing equations and the thermodynamic models are summarized, extending the model for pure nitrogen [18] to 5 species neutral air. To assess the relevance of the state-to-state model, the calculations have been performed also with the Park multi-temperature model.

2.1. Governing equations

The flow field was simulated by solving the unsteady reactive Euler equations which in vector form read

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{W}(\mathbf{U}), \tag{1}$$

where $\mathbf{U}$, $\mathbf{F}(\mathbf{U})$, $\mathbf{G}(\mathbf{U})$ and $\mathbf{W}(\mathbf{U})$ are the unknown vector of conserved variables, the fluxes along x and y directions and the source term, respectively. The equations can be solved either in a two-dimensional or in an axial symmetric configuration. The detailed expansion of the Eq. 1 depends on the thermochemical approach employed, i.e., the StS or a generic multi-temperature model. StS and MT models differ mainly in the number of species and in the equation defining the internal energy. Specifically, for a mixture of S chemical components, the s-th one having V_s internal levels, the StS approach considers $N = \sum_{s=1}^{S} V_s$ independent species, determining internal energy from level distribution as a mass weighted averaged of the internal energy of individual levels; on the other hand, in the case of multi-temperature models, $V_s = 1$ and the number of independent species N is equal to the number of chemical components S while the internal energy is calculated from the vibrational temperatures.

The detailed expansion of the equation (1), for a mixture of N species, is [18, 40, 41]

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho_{1,1} \\ \vdots \\ \rho_{1,V_1} \\ \vdots \\ \rho_{S,1} \\ \vdots \\ \rho_{S,V_S} \\ \rho u \\ \rho v \\ \rho e \\ \rho\varepsilon_{vib,1} \\ \vdots \\ \rho\varepsilon_{vib,M} \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho_{1,1}u \\ \vdots \\ \rho_{1,V_1}u \\ \vdots \\ \rho_{S,1} \\ \vdots \\ \rho_{S,V_S}u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \\ \rho\varepsilon_{vib,1}u \\ \vdots \\ \rho\varepsilon_{vib,M}u \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} \rho_{1,1}v \\ \vdots \\ \rho_{1,V_1}v \\ \vdots \\ \rho_{S,1}v \\ \vdots \\ \rho_{S,V_S}v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \\ \rho\varepsilon_{vib,1}v \\ \vdots \\ \rho\varepsilon_{vib,M}v \end{bmatrix} = \begin{bmatrix} \dot{\omega}_{1,1} \\ \vdots \\ \dot{\omega}_{1,V_1} \\ \vdots \\ \dot{\omega}_{S,1} \\ \vdots \\ \dot{\omega}_{S,V_S} \\ 0 \\ 0 \\ 0 \\ \dot{\omega}_{vib,1} \\ \vdots \\ \dot{\omega}_{vib,M} \end{bmatrix}, \tag{2}$$

where $\rho_{s,v}$ is the gas density of the species s in the v state (there are no state in multi-temperature models, i.e. $v = 1$), p is the gas pressure, u and v are respectively the x and y components of the flow velocity, e is the total energy per unit mass, $\varepsilon_{vib,m}$ is the vibrational energy per unit mass of molecule m (considered only in the case of multi-temperature models) and M is the number of molecules. The total density of the component s and the total density are given by $\rho_s = \sum_v \rho_{s,v}$ and $\rho = \sum_s \rho_s$, respectively. $\{\dot{\omega}_{s,v}\}$ are the chemical source terms, whereas $\{\dot{\omega}_{vib,m}\}$ are the vibrational energy source terms needed only in the case of multi-temperature models.

The system (2) is closed by a relation between p and e under the approximation of perfect gas [42]. Following the authors' previous work [18] the relation reads

$$p = (\bar{\gamma} - 1)\left[\rho e - \rho\left(\varepsilon_{vib} + \varepsilon_{chem}\right) - \rho\frac{u^2 + v^2}{2}\right], \tag{3}$$

where ε_{vib} is the total contribution of vibrational energy, whose expression depends on the thermochemical model employed and is given in the following subsections, and ε_{chem} is the total contribution of chemical terms given by

$$\varepsilon_{chem} = 1/\rho \sum_{s=1}^{S} \rho_s h_s^f, \tag{4}$$

where h_s^f is the formation enthalpy per unit mass of component s and $\bar{\gamma}$ is the specific heats ratio of the gas mixture. The mixture specific heat at constant pressure, $\bar{c}_p$, computed on the basis of the degrees of freedom in equilibrium at the gas temperature T, is defined by means of the relation

$$\bar{c}_p = \alpha\, R\,.$$

In the equation above, R is the specific gas constant while α is evaluated as $\alpha = \sum_s \chi_s \alpha_s$ where χ_s is the molar fraction of component s and α_s, considering equilibrium between rotation and translation, is equal to 5/2 for monoatomic components and 7/2 for diatomic components. Using the Mayer relation, $R = \bar{c}_p - \bar{c}_v$, where $\bar{c}_v$ is the mixture specific heat at constant volume, the specific heats ratio of the gas mixture $\bar{\gamma}$ can be written as

$$\bar{\gamma} = \frac{\bar{c}_p}{\bar{c}_v} = \frac{\alpha}{\alpha - 1}. \tag{5}$$

More details on the model can be found in Ref. [18].

2.1.1. State-to-State air kinetics

In this work, the 5 species air mixture, N_2, O_2, NO, N and O, is considered. The molar formation enthalpies, expressed in eV, are $H^f_{N_2} = 0$, $H^f_{O_2} = 0$, $H^f_{NO} = 0.941$, $H^f_N = 4.88195$ and $H^f_O = 2.55764$ [4]. Only the ground state was considered for atomic species (N, O) and for NO, whereas 68 and 47 vibrational levels were considered for N_2 and O_2 molecules, respectively, whose energies are given in [43, 44]. The kinetic model is described in details in [6, 43, 45, 46] and includes two different kind of processes in the vibrational kinetics of neutral air mixture flows:
internal energy exchange (involving atoms, a, and molecules, m)

$$N_2(v) + N_2(w) \leftrightarrow N_2(v-1) + N_2(w+1) \quad \text{(VV)} \tag{6}$$

$$N_2(v) + N_2 \leftrightarrow N_2(v-1) + N_2 \quad \text{(VTm)} \tag{7}$$

$$N_2(v) + N \leftrightarrow N_2(v-\Delta v) + N \quad \text{(VTa)} \tag{8}$$

$$O_2(v) + O_2(w) \leftrightarrow O_2(v-1) + O_2(w+1) \quad \text{(VV)} \tag{9}$$

$$O_2(v) + O_2 \leftrightarrow O_2(v-1) + O_2 \quad \text{(VTm)} \tag{10}$$

$$O_2(v) + O \leftrightarrow O_2(v-\Delta v) + O \quad \text{(VTa)} \tag{11}$$

$$N_2(v) + O_2 \leftrightarrow N_2(v-1) + O_2 \quad \text{(VTm)} \tag{12}$$

$$N_2(v) + O \leftrightarrow N_2(v-1) + O \quad \text{(VTa)} \tag{13}$$

$$O_2(v) + N_2 \leftrightarrow O_2(v-1) + N_2 \quad \text{(VTm)} \tag{14}$$

$$O_2(v) + N \leftrightarrow O_2(v-1) + N \quad \text{(VTa)} \tag{15}$$

$$O_2(v) + N_2(w-1) \leftrightarrow O_2(v-2) + N_2(w) \quad \text{(VV)} \tag{16}$$

and dissociation-recombination (DR) processes, by using the direct dissociation model [10, p. 182]:

$$N_2(v) + N_2 \leftrightarrow 2N + N_2 \quad \text{(DRm)} \tag{17}$$

$$N_2(v) + N \leftrightarrow 2N + N \quad \text{(DRa)} \tag{18}$$

$$O_2(v) + O_2 \leftrightarrow 2O + O_2 \quad \text{(DRm)} \tag{19}$$

$$O_2(v) + O \leftrightarrow 2O + O \quad \text{(DRa)} \tag{20}$$

or, when data for dissociation-recombination process are missing, the ladder climbing (LC) model [10, p. 183]:

$$N_2(v_{max}) + O_2 \leftrightarrow N_2(v_{max+1}) + O_2 \equiv 2N + O_2 \quad \text{(LCm)} \tag{21}$$

$$N_2(v_{max}) + O \leftrightarrow N_2(v_{max+1}) + O \equiv 2N + O \quad \text{(LCa)} \tag{22}$$

$$O_2(v_{max}) + N_2 \leftrightarrow O_2(v_{max+1}) + N_2 \equiv 2O + N_2 \quad \text{(LCm)} \tag{23}$$

$$O_2(v_{max}) + N \leftrightarrow O_2(v_{max+1}) + N \equiv 2O + N \quad \text{(LCa)}. \tag{24}$$

Finally, NO dissociation and exchange (Zeldovich) [47, 48] reactions were also taken into account

$$NO + X \leftrightarrow N + O + X \tag{25}$$

$$O_2(v) + N \leftrightarrow NO + O \tag{26}$$

$$N_2(v) + O \leftrightarrow NO + N \tag{27}$$

where X is the generic collision partner, i.e. N_2, O_2 NO, N or O. The rate coefficients of Eqs. (6)-(27) depend only on the gas temperature. The total vibrational energy per unit mass is defined as

$$\varepsilon_{vib} = \frac{1}{\rho} \sum_{s=1}^{S} \sum_{v=1}^{V_s} \rho_{s,v} \varepsilon_{s,v}, \tag{28}$$

where $\varepsilon_{s,v}$ is the molecular energy per unit mass of the given vibrational level. For non-Boltzmann level distributions, the internal temperature cannot be defined uniquely. Here, because the low energy distribution follows a Boltzmann trend, the temperature of the first level will be considered,

$$T_{Vs} = \frac{W_s \left(\varepsilon_{s,2} - \varepsilon_{s,1}\right)}{\Re \ln\left(\dfrac{\rho_{s,1}}{\rho_{s,2}}\right)}, \tag{29}$$

calculated as the temperature of the Boltzmann distribution passing through the ground level and the first excited one, where W_s is the molecular weight of specie s and $\Re$ is the universal gas constant.

2.1.2. Multi-temperature Park's model

The five species Park multi-temperature model considers three dissociation reactions

$$N_2 + X \leftrightarrow 2N + X \tag{30}$$

$$O_2 + X \leftrightarrow 2O + X \tag{31}$$

$$NO + X \leftrightarrow N + O + X \tag{32}$$

and two NO exchange (Zeldovich) reactions

$$\mathrm{NO} + \mathrm{O} \leftrightarrow \mathrm{N} + \mathrm{O_2} \tag{33}$$

$$\mathrm{N_2} + \mathrm{O} \leftrightarrow \mathrm{NO} + \mathrm{N}. \tag{34}$$

The chemical source terms $\{\dot{\omega}_s\}$ were evaluated by using forward rate coefficients and equilibrium constants given in Ref. [4, p. 326, 35]. For the NO exchange reactions the forward rate coefficients and the equilibrium constants were evaluated by using the translational temperature [4, 49], whereas for dissociation reactions, following Park [4, p. 138], a geometrically averaged temperature

$$T_a = T_V^q T^{1-q}, \tag{35}$$

where q is a parameter here assumed equal to 0.5. In the present work a separate T_V was considered for each molecule, solving the corresponding energy transport equation.

The vibrational energy source term of molecule m $\{\dot{\omega}_{vib,m}\}$ can be decomposed in the collisional $\{\dot{\omega}_{LT,m}\}$ and chemical $\{\dot{\omega}_{chem,m}\}$ parts [4, p. 125] [50].

The collisional term describes the energy transfer between the translational and the vibrational degrees of freedom, modeled by the Landau-Teller equation

$$\dot{\omega}_{LT,m} = \rho_m \frac{\varepsilon_{vib,m}(T) - \varepsilon_{vib,m}(T_v)}{\tau_m}, \tag{36}$$

where $\varepsilon_{vib,m}(T)$ is the vibrational energy at equilibrium and τ_m is the corresponding relaxation time. The latter was evaluated by using the Millikan-White expression [4, p. 58][49] plus, in the case of N_2 and O_2, a correction for high temperatures in order to account for the limits in collision cross-sections [4, p. 60][49, 51, 52]. The relaxation time for collisions between molecular species m and the generic partner X given by the Millikan-White expression reads

$$\tau_{m,X}^{MW} = \frac{p_{atm}}{p} \exp[A_{m,X}(T^{-1/3} - B_{m,X}) - 18.42]. \tag{37}$$

where $p_{atm} = 101325$ Pa is the atmospheric pressure. With some exceptions the parameters $A_{m,X}$ and $B_{m,X}$ can be evaluated by using the simple expressions given in Ref. [53]. In this work the values given in Ref. [49, Tab. 1, p. 387] were used. At high temperatures (above 5000 K) the relaxation time given by the Millikan-White expression tends to underestimate the experimental data [4, p. 60][49, 51, 52] becoming (for $T \geq 20000$) shorter than the mean collision time for elastic processes [4, p. 60]. To overcome this problem, the correction proposed by Park was used [4, p. 60][51],

$$\tau_{m,X}^{c} = \frac{1}{N_m \sigma \sqrt{\frac{8\Re T}{\pi \mu_{m,X}}}}, \tag{38}$$

where N_m is the number density of the molecule under consideration, σ is the effective excitation cross section given by $3 \cdot 10^{-17}(50000/T)^2$ cm^2[49] and $\mu_{m,X}$ is the equivalent molecular weight of two colliding particles, i.e. $W_m W_X/(W_m + W_X)$. Thus, the relaxation time for collisions between the molecule m and the generic collision partner X is $\tau_{m,X} = \tau_{m,X}^{MW} + \tau_{m,X}^{c}$ and the mean value was evaluated with a weighted harmonic average [52]

$$\frac{1}{\tau_m} = \frac{1}{N_t} \sum_x \frac{N_x}{\tau_{m,X}}. \tag{39}$$

The chemical $\{\dot{\omega}_{chem,m}\}$ contribution takes into account the preferential removal effect [4, p. 107-108]. Such effect represents the loss and the gain of vibrational energy due to dissociation and recombination process which involve preferentially high vibrational levels. In this work the harmonic oscillator model was

employed, i.e., the energy exchanged in the dissociation process was equally divided by vibrational and translational degrees of freedom [4, p. 107, 126], i.e.

$$\dot{\omega}_{chem,m} = \frac{D_m}{2}\frac{\partial \rho_m}{\partial t}, \tag{40}$$

where D_m is the dissociation energy per unit mass of molecule m. Finally, the vibrational temperatures were evaluated by using the harmonic oscillator model

$$\varepsilon_{vib,m} = \frac{R_m \theta_m}{exp(\theta_m/T_V) - 1}, \tag{41}$$

where θ is the characteristic vibrational temperature equal to 3393 K, 2273 K and 2739 K for the N_2, O_2 and NO molecules, respectively [4, p. 123].

3. Numerical method

In order to solve the system of governing equations (1) an operator-splitting approach [54, 55] was employed. Such approach handles the homogeneous part and the source terms of Eq. (1) in two separate steps. The first step solves the homogeneous system

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{0}, \tag{42}$$

from which the homogeneous unknown vector $\mathbf{U}^{hom}(t + \Delta t)$ is obtained. The second step, starting from $\mathbf{U}^{hom}(t + \Delta t)$, updates the solution of the equations that involves source terms, by solving locally the following system of ordinary differential equations

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{W}(\mathbf{U}^{hom}). \tag{43}$$

The homogeneous part, Eq. (42), represents the Euler equation for a non-reacting flow. As concerns space discretization, the convective fluxes can be solved by using either the Steger and Warming *flux vector splitting* [56] or the AUSMPW+ of Kim et al. [57]. Here, the primitive variables can be reconstructed by using the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) approach [58] in order to get higher accuracy. As far as the time integration concerns, a high-order Runge-Kutta scheme, that can consider both two-step 2nd and three-step 3th order approximations, was employed. More implementation details are given in Refs. [17, 18].

In order to account for the stiffness of source terms Eq. (43) is advanced by using a sub-time step $\Delta t^{(\nu)}$ that is a fraction of the fluid dynamic time step Δt, which is evaluated on the basis of the CFL condition, so that

$$\sum_{\nu=0}^{n-1} \Delta t^{(\nu)} = \Delta t. \tag{44}$$

In the present work the sub-time step width was fixed to $\Delta t^{(\nu)} = \Delta t/n$ where n was chosen depending on the case study. At each sub-step Eq. (43), properly rewritten, is solved by using the Gauss-Seidel iterative scheme with a fixed number of inner iteration that depends on the case study. More implementation details are given in Ref. [18].

3.1. Computational approach

The effort to extend the state-to-state kinetics to simulate complex 2D/3D configurations, by using the aid of GPUs computational power, was shown in [17, 18]. In the first work [17] a GPU version of a Roe's flux difference splitting scheme to solve the 2D compressible Euler equations for a frozen single component flow was presented. A detailed description of the kernel configuration and of the code performance was

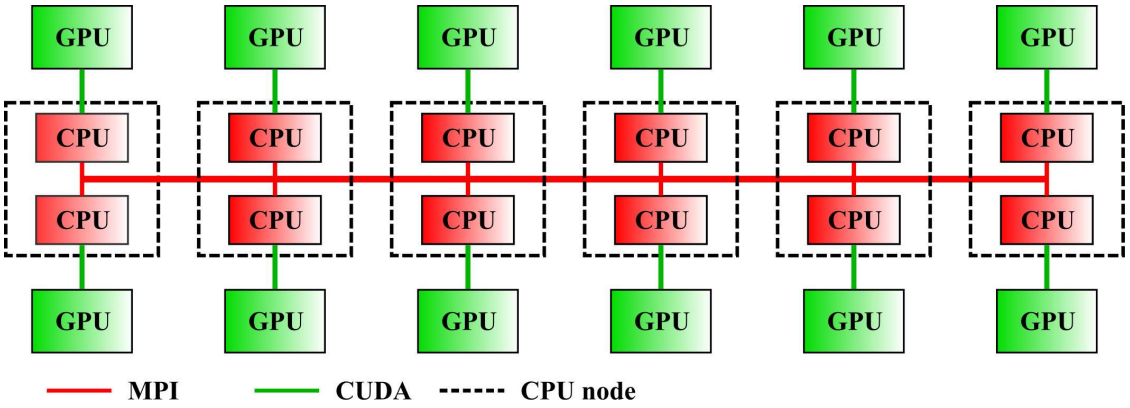


Figure 1: Scheme of the GPU cluster along with the MPI-CUDA paradigm (based on Su et al. [34])

provided. Subsequently [18], a multi-GPU implementation within a compute node was presented and 2D test cases with reacting N-N_2 mixtures were performed. However, the extension of the state-to-state model to an air mixture makes the simulations much more computing demanding due to the increase of number of species and reactions, resulting also in a larger stiffness of the problem.

The previous implementation used peer-to-peer communication to transfer data between GPUs. However, this strategy allows to use only GPUs on the same node. In this work, to scale the application across a multiple-nodes GPU cluster, MPI-CUDA approach was implemented. An all-device (GPU) computational approach is used [34], i.e. all the relevant computations are performed on the device (GPU) whereas initialization, timing, data transfer, synchronization and printing are CPU tasks. As usual the computation starts on the CPU where the MPI execution environment is initialized. A 2D Cartesian topology is created and neighbor relationships are established by using the well know MPI functions (`MPI_Cart_create`, `MPI_Cart_coords`, `MPI_Cart_shift`). Then, the number of GPUs in a node is inquired by using `cudaGetDeviceCount(&devCount)` and each GPU is assigned to an individual MPI process by using `cudaSetDevice(myrank%devCount)` where `myrank` is the rank of the MPI process [34]. After that, the MPI master rank reads the input data and broadcasts them to all the other MPI processes. The computational domain can be partitioned in both x and y directions. Each sub-domain has ghost cells that are used to evaluate the fluxes along a generic boundary edge: boundary conditions are imposed in the ghost cells associated to edges that lie on the borders of the fluid domain; whereas, for internal edges, ghost cells are filled with values corresponding to the fluid cells of adjacent sub-domains. Each MPI process gets a sub-domain which is initialized and the data are copied on the corresponding GPU. The computation on the GPU starts and at the end of each iteration boundary conditions and data transfer between adjacent sub-domains are performed. To transfer data between sub-domains the `cudaMemcpy` function is used to copy data from GPU to CPU then the `MPI_Sendrecv` function transfers the data between CPUs and finally `cudaMemcpy` is used again to copy data from CPU to GPU. Obviously, in order to minimize the time required by communications and not to overload the network, only the data required by the ghost cells are copied and transferred.

3.2. Hardware and Software

High performance computing resources were provided by the 'Politecnico di Bari'. The computations were performed on a 6 node GPU cluster. Each node hosts two Nvidia Tesla K40m, two Intel(R) Xeon(R) CPU E5-2630 v2 2.60 GHz and 64 GB of RAM memory. Communication are carried out by a QLogic QLE7340, with IBA7322 ASIC, Single-Port 40 Gbps 4× QDR Infiniband Host Channel Adapter. The software package was compiled using CentOS 6.6 (Final), GCC 4.9.2, CUDA Toolkit 7.0 and OpenMPI 1.8.5. A scheme of the GPU cluster along with the MPI-CUDA paradigm is shown in Fig. 1.

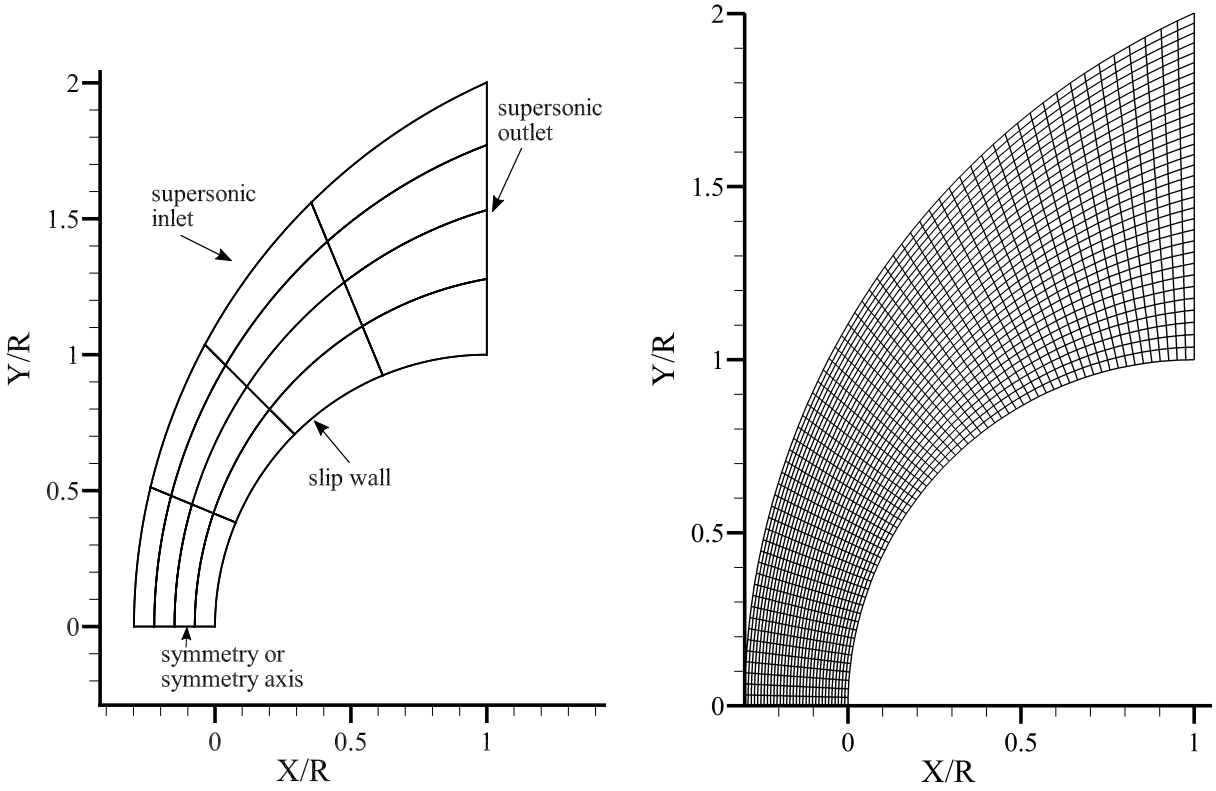


Figure 2: Computational domain, with an example of 4×4 MPI partitioning, along with boundary conditions (left). 32×32 fluid cells computational grid (right).

3.3. Computational domain configuration and partitioning

In order to assess the accuracy and the efficiency of the proposed method, the flow past a sphere in different axisymmetric configurations is considered, as described in the following section. Figure 2 shows the generic computational domain configuration used in this work. On the left panel, the computational domain, with an example of 4×4 MPI partitioning, along with boundary conditions, is given. On the right panel an example of 32×32 fluid cells computational grid is also shown. In the case of hypersonic flows analyzed in this work the left edge of the domain was set up as *supersonic inlet*, the right as *no slip wall*, the top as *supersonic outlet* and bottom as *symmetry or symmetry axis*.

The kernel configuration, defined by the number of *Blocks per grid*, B, and by the number of *Threads per block*, T, was set on the basis of the sensitivity analysis carried out in Ref. [17]. Each module of the code can have a different kernel configuration. The main modules are: `Kinetics`, which performs the integration of the chemical equation (Eq. 43); `BC`, which imposes boundary conditions and performs MPI data transfer; `Calc:mass,`γ`,p,T,`$\mathtt{E_{int}}$, which calculates the mean molecular weight, the mean specific heats ratio, the pressure, the temperature and the total internal energy; `Flux`, which performs the Steger and Warming or the AUSM fluxes calculation; `MUSCL`, which performs the MUSCL reconstruction; `Resid`, which performs the *residuals* calculation; `Update`, which updates the unknown vector of conserved variables ($\mathbf{U}^{hom}$) at the new fluid dynamic time step ($t + \Delta t$); `Ddtime`, which evaluates the fluid dynamic time step (Δt) on the basis of the CFL condition. The details of the kernel configuration for the main modules is given in Tab. 1, where the number of *Blocks per grid* was set as [59]

$$B = \mathrm{MIN}(65535, (N + T - 1)/T) \tag{45}$$

(here N is the number of total threads per kernel) for all modules with the exception of `BC`. Further details can be found in [17, 18].

Table 1: Total threads per kernel (N), *Threads per block* (T) and *Blocks per grid* (B). NVAR is the number of conserved variables, NSPES is the number of independent species, $\mathrm{N_x}$ and $\mathrm{N_y}$ are the fluid cells of the generic MPI sub-domain along the x and y directions, respectively.

Module	N	T	B
`Kinetics`	$\mathrm{Nx} \cdot \mathrm{Ny}$	96	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
$\mathtt{BC}_{\mathrm{left/right}}$	$\mathrm{NSPES} \cdot \mathrm{Ny}$	128	64
$\mathtt{BC}_{\mathrm{top/bottom}}$	$\mathrm{NSPES} \cdot \mathrm{Nx}$	128	64
`Calc:mass,`γ`,p,T,`$\mathtt{E}_{\mathtt{int}}$	$(\mathrm{Nx} + 4) \cdot (\mathrm{Ny} + 4)$	96	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
$\mathtt{Flux}_{\mathrm{i}}$	$(\mathrm{Nx} + 1) \cdot \mathrm{Ny}$	128	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
$\mathtt{Flux}_{\mathrm{j}}$	$\mathrm{Nx} \cdot (\mathrm{Ny} + 1)$	128	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
$\mathtt{MUSCL}_{\mathrm{i}}$	$\mathrm{NVAR} \cdot (\mathrm{Nx} + 1) \cdot \mathrm{Ny}$	128	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
$\mathtt{MUSCL}_{\mathrm{j}}$	$\mathrm{NVAR} \cdot \mathrm{Nx} \cdot (\mathrm{Ny} + 1)$	128	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
`Resid`	$\mathrm{NVAR} \cdot \mathrm{Nx} \cdot \mathrm{Ny}$	192	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
`Update`	$\mathrm{NVAR} \cdot \mathrm{Nx} \cdot \mathrm{Ny}$	96	$B = \mathrm{MIN}(65535, (N + T - 1)/T$
`Ddtime`	$\mathrm{Nx} \cdot \mathrm{Ny}$	128	$B = \mathrm{MIN}(65535, (N + T - 1)/T$

4. Results

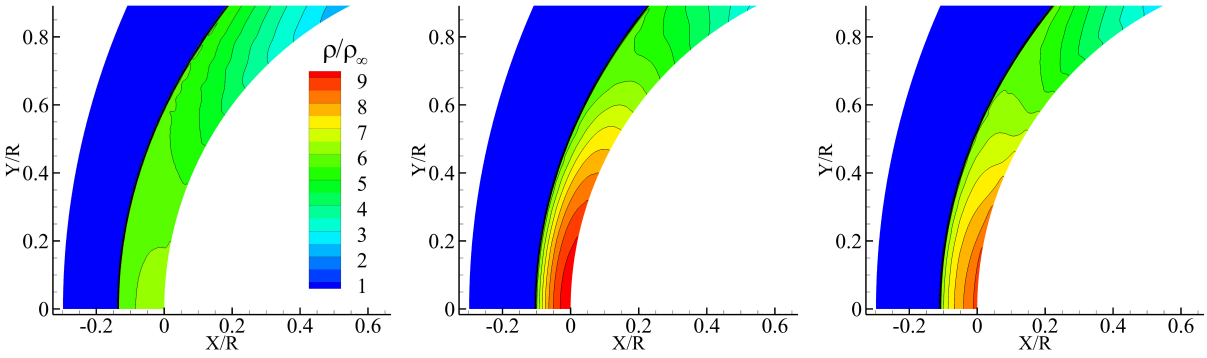
4.1. Model validation

Nowadays, the design of hypersonic vehicles is based on predictions provided by CFD codes. An appropriate reference quantity to validate the correctness of non-equilibrium models is the stand-off distance. The latter can be considered a pointer for non-equilibrium phenomena. Indeed, shock waves around hypersonic vehicles generate so high temperatures to activate molecular excitation, dissociation and ionization, transferring energy from translational degrees of freedom and causing a reduction of temperature and affecting the position of the stand–off distance. Excellent predictions of the stand-off distance can be obtained for 'ideal' or 'frozen' flow conditions and for thermochemical equilibrium conditions [60]. On the contrary, predictions are much more difficult for hypersonic flows, in the presence of thermochemical non-equilibrium.

The multi-temperatures models developed so far show good predictive capability, in terms of stand-off distance, at high hypersonic speeds but tend to reduce their precision in the intermediate regime [60, 61]. The less accuracy occurs, especially, when the molecules are vibrationally excited but chemical reactions are almost frozen [61]. These conditions can be a good reference in order to assess the theoretical correctness and the accuracy of complex models, such as the StS model considered in this work, that are devised from fundamental chemico-physical theories. At the same time, the StS model, being able to predict the real distribution of internal states, can be helpful to gain deep insights into the non-equilibrium phenomena that occur at such conditions.

In this scenario, the experimental measurements carried out by Furudate et al. [61] and by Nonaka et al. [60] were considered. These experiments investigate the flow past a sphere in the intermediate hypersonic regime and, as claimed and established by the authors [60], are more reliable compared to those obtained in the 1960's by Lobb [62]. The measurements were performed in uncontaminated dry air, consisting of $\mathrm{Y_{N_2}} = 0.767$ and $\mathrm{Y_{O_2}} = 0.233$ (Y stands for mass fraction), by using hemispheres with radius equal to 7, 14 and 15 mm, free-stream velocity and pressure ranging respectively between 2.44 and 3.85 km/s and between $5.6 \cdot 10^2$ and $2.0 \cdot 10^4$ Pa. As argued by Nonaka et al. [60], a binary scaling parameter (ρR) is used in order to reproduce the Damköhler and Reynolds numbers of different free-stream conditions. Specifically, the case considered here is characterized by the following parameters (see Tab. 2): $R = 7$ mm, $u_\infty = 3490$ m/s, $T_\infty = 293$ K and $\rho R = 4 \cdot 10^{-4}$ kg/m^2 [60, row 18 Tab. 1 and Fig. 10]. For the free-stream internal distributions equilibrium conditions were imposed: vibrational temperatures were posed equal to the translational temperature in the Park model, whereas a Boltzmann distribution of internal states at the gas temperature was assumed when the StS model is employed.

As concerns the computational setup, the grid used includes 228×392 fluid cells. Such grid was chosen after a mesh refinement study that showed a good grid independence of the results. The time integration is



(a) Frozen (b) Park (c) StS

Figure 3: Normalized density contour plots.

performed by using the three-step third order Runge-Kutta scheme and the primitive variables are reconstructed with the MUSCL scheme that makes the space discretization second order fully upwind. Finally, 4 sub-steps are used to advance in time the chemical equation and 8 inner iterations are employed for the Gauss-Seidel iterative scheme when the Park model is used, whereas 1 sub-step and 32 inner iterations are used with the StS model. The independence of the results from the number of sub-steps and of inner iterations was also verified by comparing the outcomes with those obtained by doubling both numbers.

Figure 3 shows the contour plots of the density, normalized by the free-stream value, obtained by using both the frozen and the two thermochemical non-equilibrium models. The latter models show a quite different shape of the density distribution compared to the one obtained with the frozen condition. In agreement with theoretical observations, chemical reactions cause a larger density increase thus reducing the stand-off distance. Indeed, it is noticeable that the shock layer provided by the frozen assumption is larger that that given by the non-equilibrium models; moreover, the Park model provides larger densities compared to the StS approach.

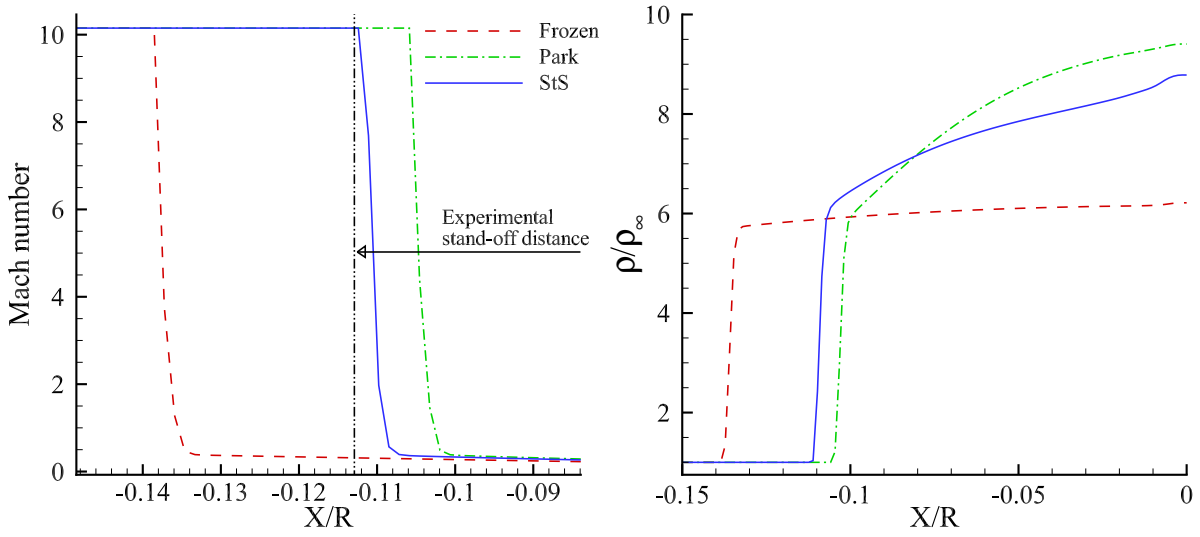
A more quantitative analysis is given in Fig. 4 that shows the stagnation streamline profiles of the Mach number (a) (the experimental value of the stand–off distance is provided for comparison) and of the normalized density (b). As expected, the stand–off distance decreases when chemical reactions are considered. Looking only at the the two non-equilibrium models, the StS approach provides a larger stand-off distance which appears to be in good agreement with the experimental measure. A less good agreement was provided by the Park model which, as already shown in Ref. [61], tends to underestimate such a quantity. The density profiles given in Fig. 4 (b) are consistent with these results: the smaller the stand-off distances, the larger is the density gradient behind the shock layer.

A further proof of the StS model accuracy is given in Figs. 5 (a) and 5 (b). Here, the Mach number contour plots provided by the Park (a) and StS (b) models, along with the experimental shock shape [60, Fig. 10], are given. Once again, the results given by the StS model show a better agreement with the experimental measure.

Figure 6 (a) shows the stagnation streamline profiles of the translational and vibrational temperatures. In order to give a better view Figs. 6 (b), (c) and (d) show separately the translational, the N_2 and the O_2 vibrational temperatures. As concerns the frozen flow, downstream of the bow shock the temperature remains constant. On the contrary, when the non-equilibrium models are considered, due to molecules

Table 2: Test conditions [60, row 18 Tab. 1 and Fig. 10]

R [mm]	u_∞ [m/s]	T_∞ [K]	Y_{N_2}	Y_{O_2}	ρR [kg/m^2]
7	3490	293	0.767	0.233	$4 \cdot 10^{-4}$



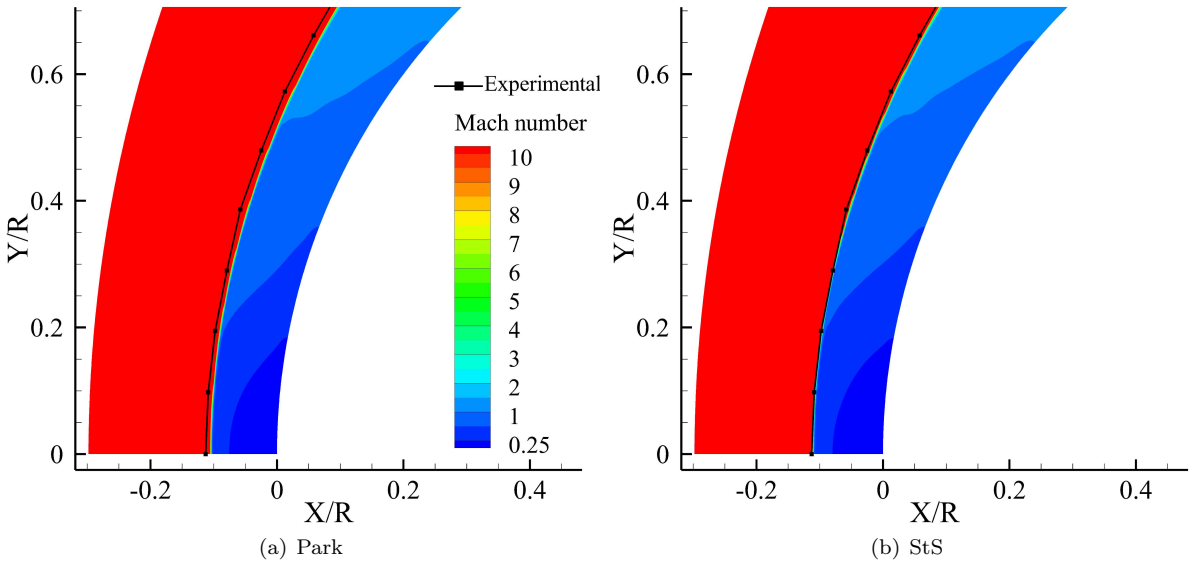
(a) Mach number (b) Density

Figure 4: Stagnation streamline profiles: (a) Mach number along with the experimental stand-off distance [60]; (b) density profiles.

excitation and chemical reactions that adsorb energy, the translational temperatures show a sharp peak just downstream of the shock front, followed by a decrease up to the sphere surface. As far as the vibrational temperatures are concerned, a gain and a loss contribution have to be considered. The gain is due to the transfer of energy from the translational and rotational degrees of freedom to the vibrational one, modelled in the Park approach by using the Landau-Teller equation (Eq. 36). The loss is caused by the preferential removal mechanism [4, p. 107-108]. Just downstream of the shock front there is a relatively quick growth of the vibrational temperatures. The growth rate subsides as the vibrational temperatures approach the translational one. Finally, both vibrational and translational temperatures decrease due to dissociation. The O_2 vibrational temperatures show a sharper increase, compared to the N_2 ones. Oxygen molecules dissociate at lower temperatures than N_2; therefore, the effects of the preferential removal mechanism take place earlier and appear stronger. Finally, the Park model shows smaller vibrational temperatures, likely due to a stronger preferential removal effect modeled here by using the oscillator model [4, p. 107, 126] given in Eq. (40).

The stagnation streamline profiles of the species mass fractions are given in Fig. 7. The results provided by the two reactive models are quite similar. They predict a small consumption of N_2 and a larger O_2 dissociation. However, from a quantitative point of view, the Park model predicts a larger dissociation of both N_2 and O_2 molecules, thus also justifying the smaller translational and vibrational temperatures provided by the same model. Moreover, the Y_{NO} provided by the Park model is almost twice the one given by the StS approach. Finally, both models predicts a nearly zero fraction of atomic nitrogen.

Figures 8 (a) and 8 (b) show the translational temperature contour plot obtained by using the StS and the Park model, respectively. The temperature profiles along the sphere surface are given in Fig. 9. Since the temperature is decreasing, due to expansion, recombination processes prevail on dissociation; this causes higher vibrational temperatures compared to the translational one. However, such non-equilibrium phenomena can affect the global reaction rates, thus explaining the lower accuracy of multi-temperature models, such as the one implemented in this work, that make the assumption that internal states have a Boltzmann distribution and that the chemical rates can be computed as a geometric average of the translational and vibrational temperatures.



(a) Park (b) StS

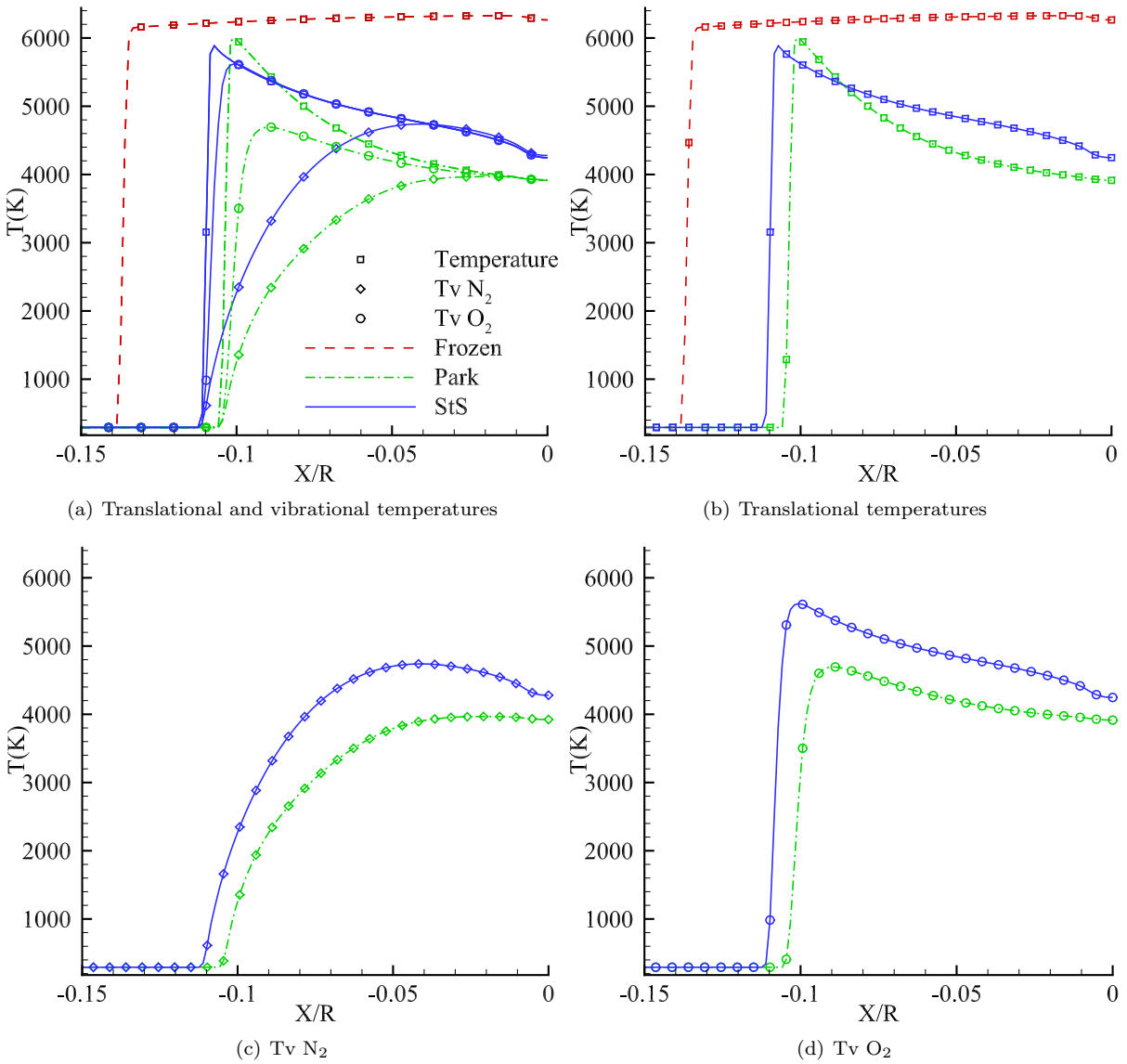
Figure 5: CFD and experimental [60, Fig. 10] shock shape: (a) Park; (b) StS.

4.2. Code performance analysis

In this section a detailed code performance analysis is provided. Both pure MPI-CPU and MPI-CUDA implementations are examined by using the two different reactive non-equilibrium models in addition to the frozen one. The test case consists in a high enthalpy air flow past a sphere with radius $R = 1/2$ inch. The free-stream values of the Mach number, pressure and temperature were set up to 6.14, 2910 Pa and 1833 K, respectively. The free-stream air composition is a mixture of atoms and molecules whose concentrations, in mol/m^3, were set up to $[N_2] = 1.5031 \cdot 10^{-1}$, $[O_2] = 3.9543 \cdot 10^{-2}$, $[NO] = 1.006 \cdot 10^{-3}$, $[N] = 6.5332 \cdot 10^{-11}$, $[O] = 8.4467 \cdot 10^{-5}$. As far as the free-stream internal distributions concern, equilibrium conditions were imposed. Therefore, vibrational temperatures used in the Park model were set up equal to the translational temperature, whereas when the StS model is employed, a Boltzmann distribution of internal states at the gas temperature is imposed. Time integration is performed by using the two-step second order Runge-Kutta scheme and the primitive variables are reconstructed with the MUSCL scheme that makes the space discretization second order fully upwind. Finally, 4 sub-step are used to advance in time the chemical equation (Eq. 43) and 8 inner iterations are employed for the Gauss-Seidel iterative scheme.

Code performance strongly depends on the model employed. In order to show which are the most time-consuming modules, a single GPU profiling performance is given in Fig. 10 for a computational grid of 256×128 fluid cells. The figure shows the execution time of the main modules, normalized with respect to the total execution time, as a function of advection time steps. All cases show a non-constant behavior in the first iterations due to initialization process. As concerns the frozen case the most time-consuming modules are the `Flux + Muscl` and the `Resid + Update` which take almost 60% and 13% of time, respectively. When the chemical kinetics is activated the time taken by the fluid dynamic modules becomes less important or negligible and the `Kinetics` module takes 80% and about 100% of time when the Park and the StS models are employed, respectively. Looking at the absolute computational time per iteration (TpI) it is found that the Park model is an order of magnitude slower than the frozen one and the StS model is three order of magnitude slower than the Park one. Specifically, the time per iteration is equal to $4.72 \cdot 10^{-3} s$, $2.87 \cdot 10^{-2} s$ and $5.11 \cdot 10^{1} s$ for frozen, Park and StS models, respectively. Finally, to give an idea of the computational cost of a full simulation, the number of iterations needed to reach steady state conditions is of the order of ten thousand.

In order to compare the performance of a single GPU against those of a single core of a multicore CPU (single-core CPU), several simulations were carried out by using the two reactive non-equilibrium models



(a) Translational and vibrational temperatures

(b) Translational temperatures

(c) Tv N_2

(d) Tv O_2

Figure 6: Temperatures streamline profiles: (a) Translational and vibrational temperatures; (b) Translational temperatures; (c) N_2 vibrational temperatures, (d) O_2 vibrational temperatures.

in addition to the frozen model and by varying the size of the computational grid. Five grid sizes were investigated starting from a very coarse grid which includes 32×16 fluid cells up to a quite fine grid with 1024×512 fluid cells. The results are summarized in Fig. 11. Specifically, Fig. 11 (a) shows the time per iteration, normalized to the value obtained for the coarsest grid, as a function of the grid size normalized to the coarsest grid size. One would expect that going from a grid size to the next, obtained by doubling the number of fluid cells in each direction, the computational time per iterations should be four times larger than that obtained by using the previous grid. A clear different behavior of the single-core CPU and of the single GPU emerges from the figure. Indeed, while the single-core CPU shows the ideal expected behavior for all the models employed, even with very coarse grid sizes, the single GPU provides a non-linear growth of the computational time and different performance depending on the model used. This behavior was quite predictable and it is something related to the Amdahl's [63, 64] and the Gustafson's laws [63, 65]. The issue

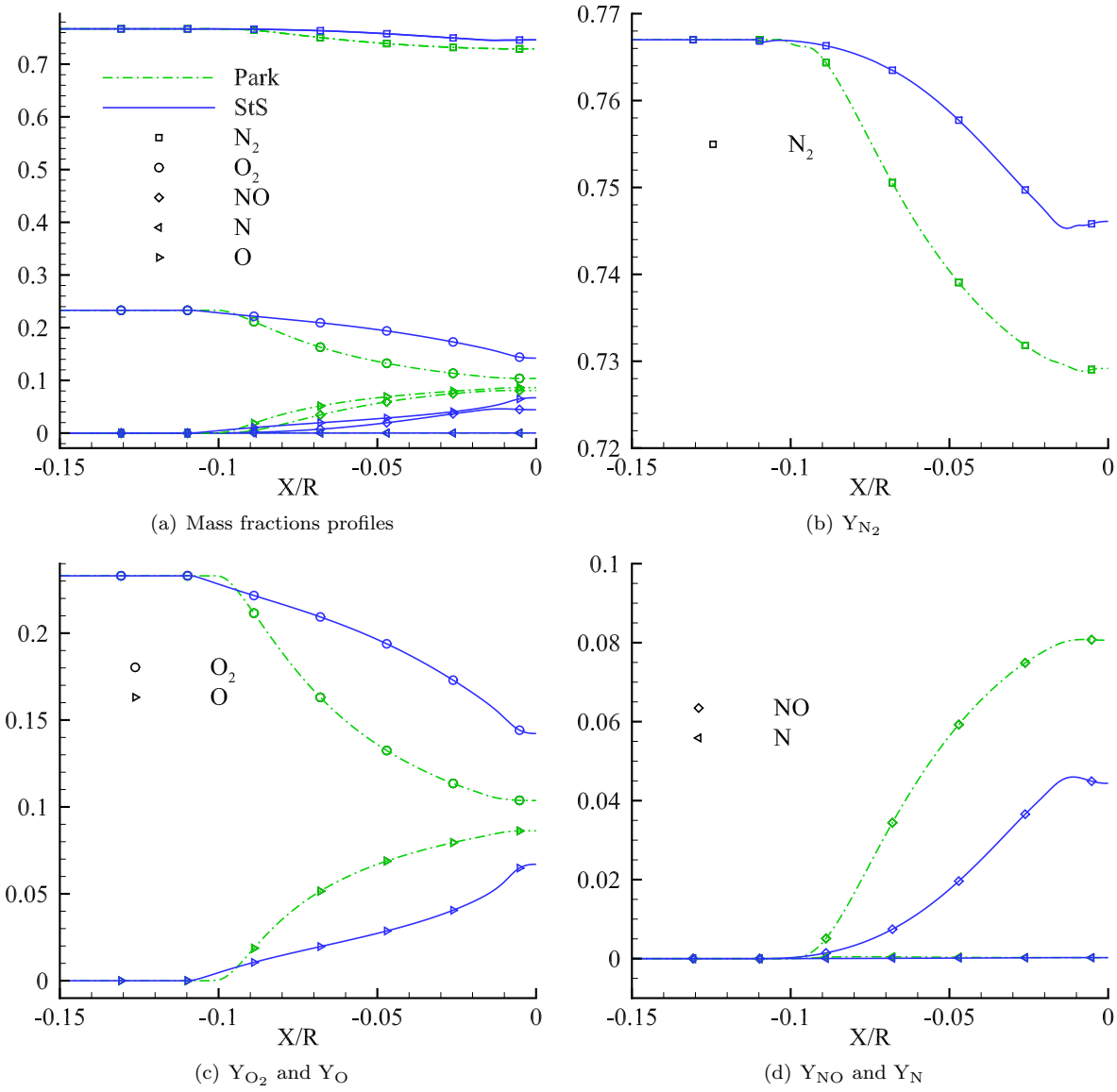
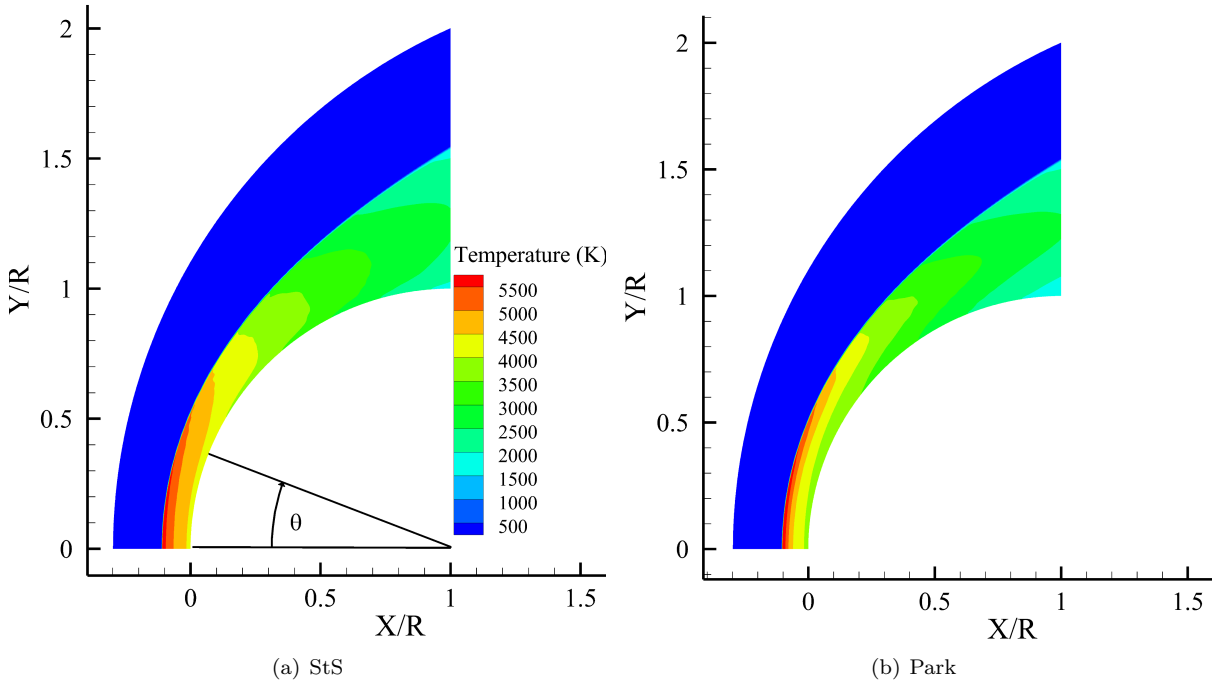


Figure 7: Stagnation streamline profiles: (a) Mass fractions profiles for all chemical species; (b) N_2 mass fraction; (c) O_2 and O mass fractions; (d) NO and N mass fractions.

has been known since the dawn of parallel computing and is related to the fraction of the algorithm that is not parallelizable. The law described in the Amdahl's paper [64] can be summarized by the following relation: speed-up $= 1/(s + p/n)$, where s and p are the not parallelizable and parallelizable fractions of the code and n is the number of processors available. Therefore, if s is even just 1% of the code and, even if the parallelization is ideal, the maximum speed-up cannot be larger than 100. Accordingly, the huge difference between the behavior of the single-core CPU and the single GPU, shown in Fig. 11 (a), is certainly a consequence of Amdahl's law, due to high number of processors available on the GPU. De facto, when the size of the problem is small, s is larger and the use of thousands of cores does not reduce the computational time per iteration. On the contrary, when models that have a larger computational cost are used, the s fraction is smaller and the ideal behavior is obtained for smaller problem sizes, as again shown in Fig. 11 (a). Indeed, looking at the GPU computations the profile related to the StS model grows faster than the ones



(a) StS (b) Park

Figure 8: Translational temperature contour plot: (a) StS; (b) Park.

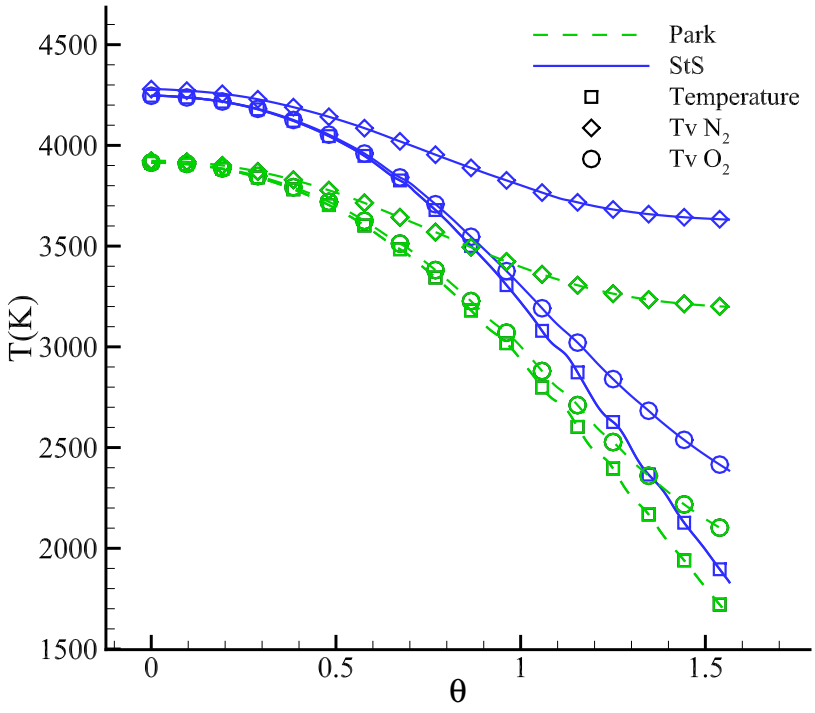
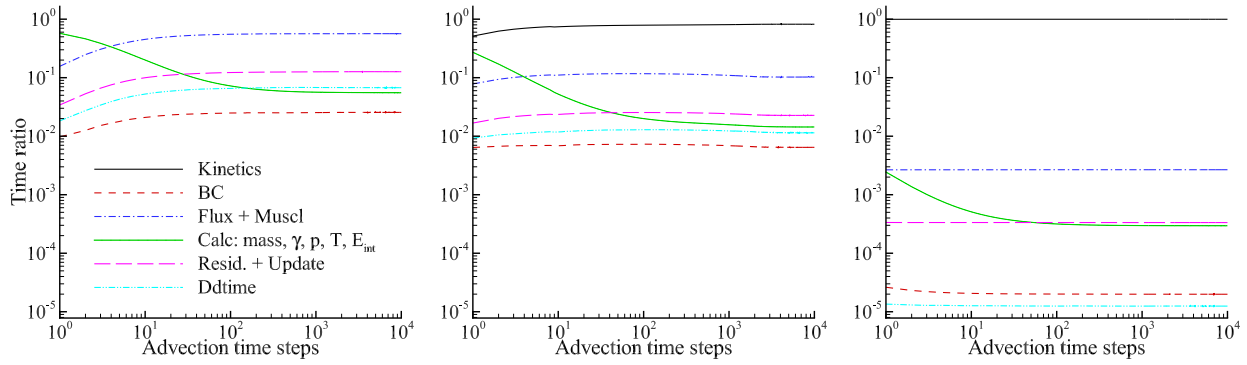
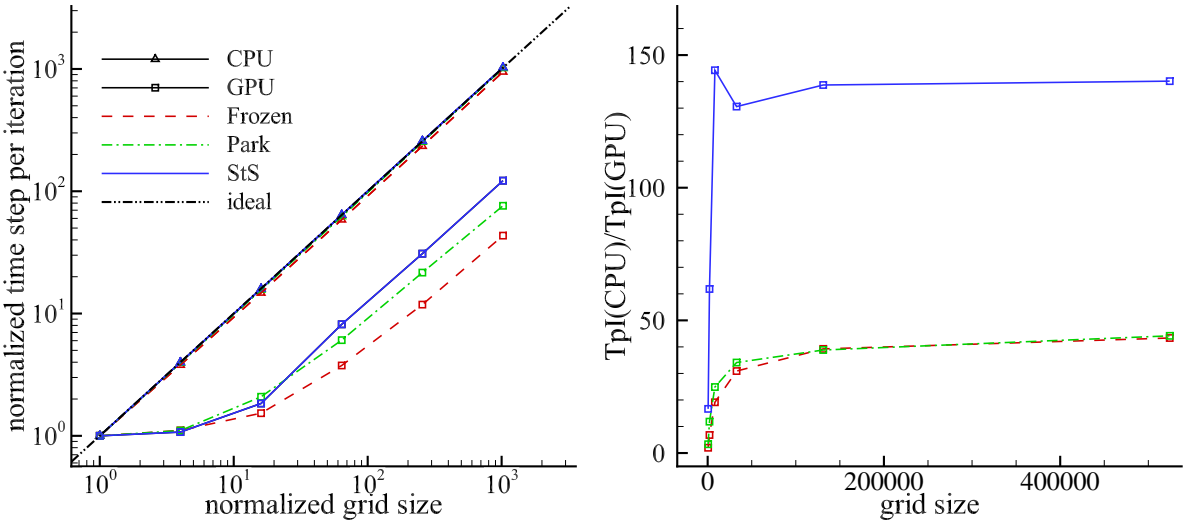


Figure 9: Temperature profiles along the wall as a function of the angle θ for the StS and Park model.



(a) Frozen (b) Park (c) StS

Figure 10: Execution time of the main modules, relative to the total execution time, as a function of advection time steps obtained by using a computational grid which includes 256×128 fluid cells.

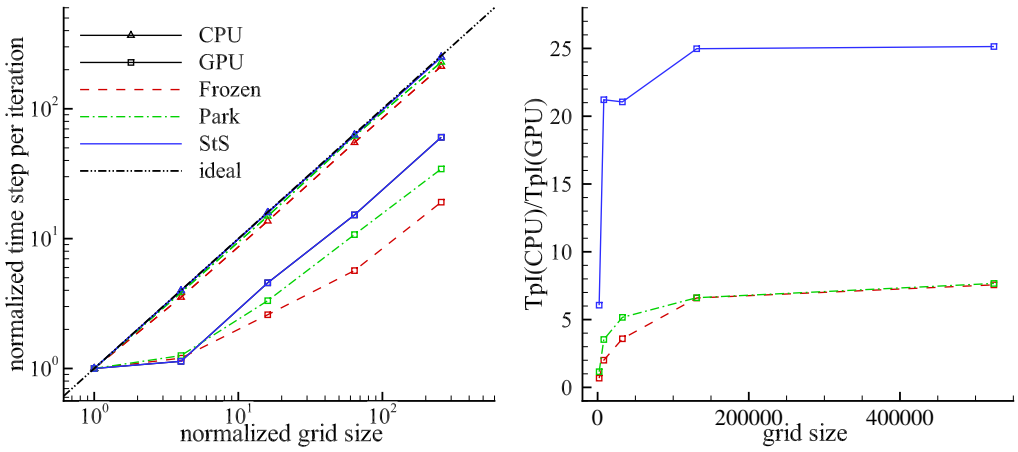


(a) (b)

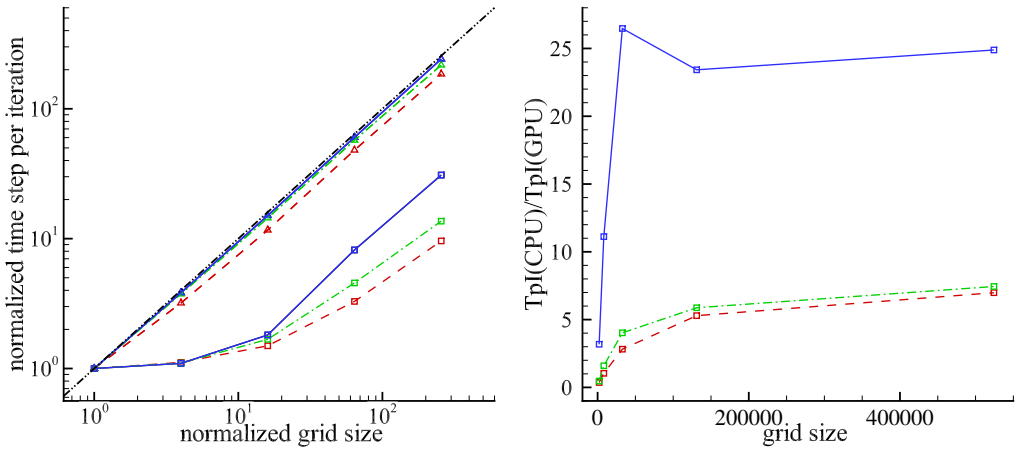
Figure 11: Performance results obtained by using a single-core CPU and a single GPU: (a) time per iteration (TpI) normalized by the one obtained on the coarsest grid as a function of the grid size normalized by the coarsest grid size. (b) TpI(CPU)/TpI(GPU) as a function of the grid size.

obtained with other models. On the opposite side the profile related to the Frozen case has the slowest growth. Such behavior corroborates the more general Gustafson's law [65] because the Amdahl's law [64] doesn't take into account the size of the problem and the complexity of the models. However, as argued by Gustafson it cannot be implicitly assumed that p is independent of n. Conversely, one would expect that the problem size increases with the number of processors available. Hence, what have to be kept constant is the run time rather than the problem size [65]. In light of this observation the use of GPU and, as will be shown later, the use of multi-GPU computing is justified.

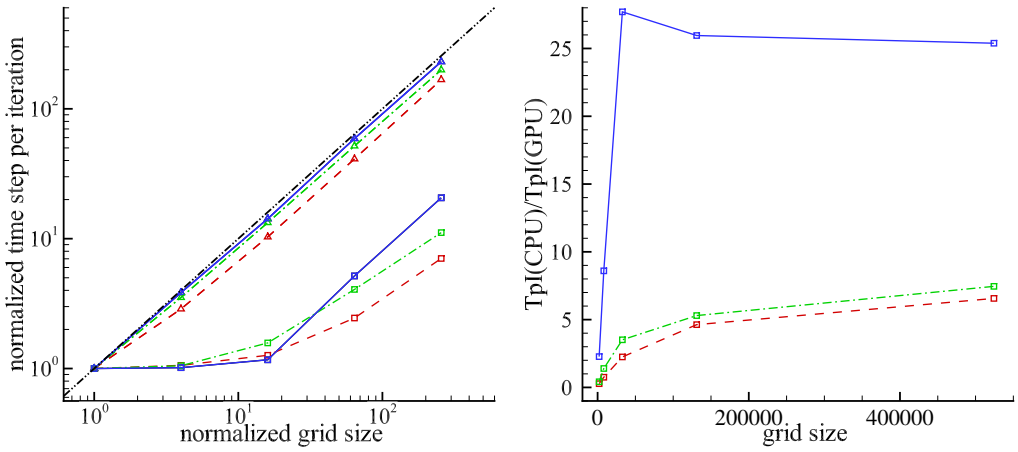
A further comparison between the single-core CPU and the single GPU in terms of reduction of the execution time, here evaluated as the ratio between the time per iteration obtained on the single-core CPU and the one obtained on the single GPU (TpI(CPU)/TpI(GPU)), is shown in Fig. 11 (b). In light of foregoing, it is obvious that for all the models employed the smallest problem size provides the worst TpI(CPU)/TpI(GPU). As concerns the StS model, i.e., the most complex and most time-consuming model,



(a) 1 node

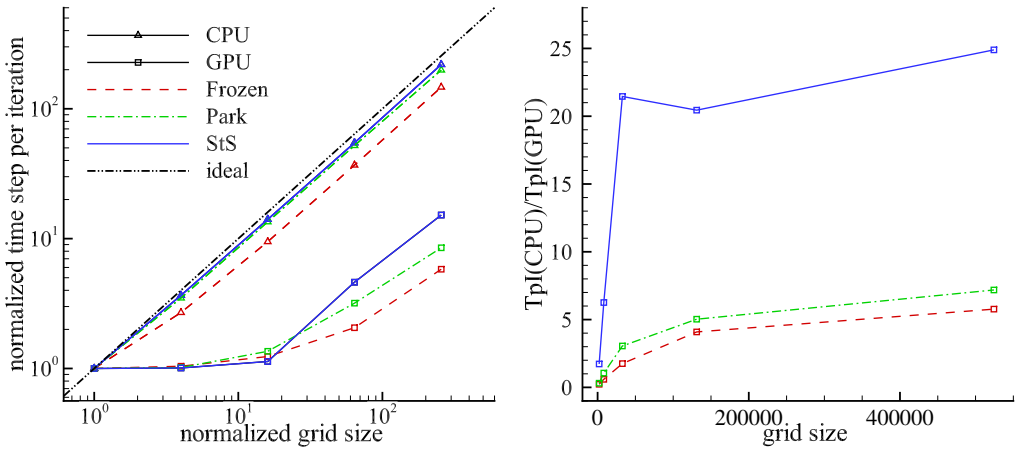


(b) 2 nodes

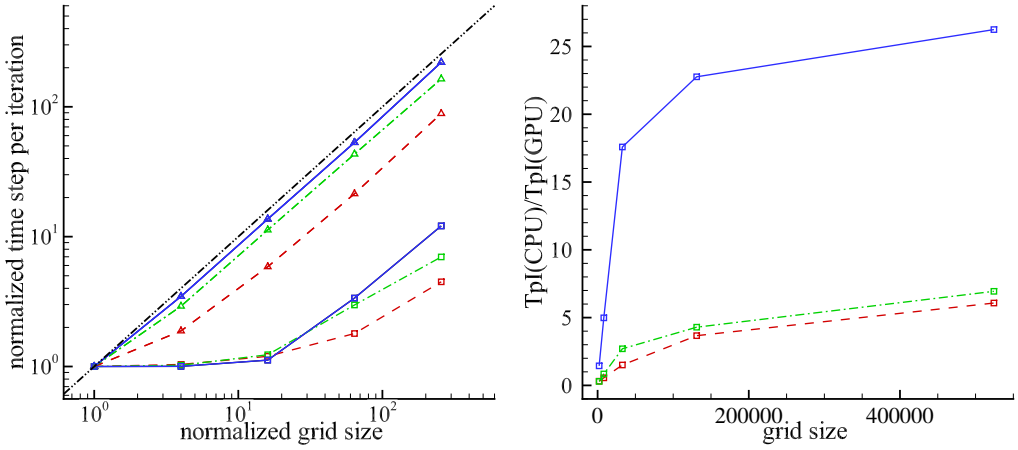


(c) 3 nodes

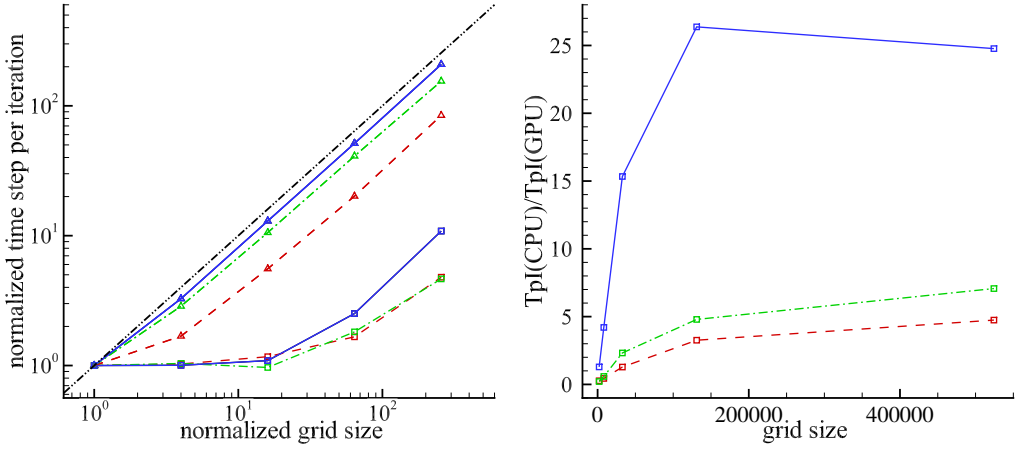
Figure 12: Performance results obtained by using full node computational power (each node has 12 CPU cores and 2 GPUs cards): (left) time per iteration (TpI) normalized by the one obtained on the coarsest grid as a function of the grid size normalized by the coarsest grid size. (right) TpI(CPU)/TpI(GPU) as a function of the grid size.



(a) 4 nodes

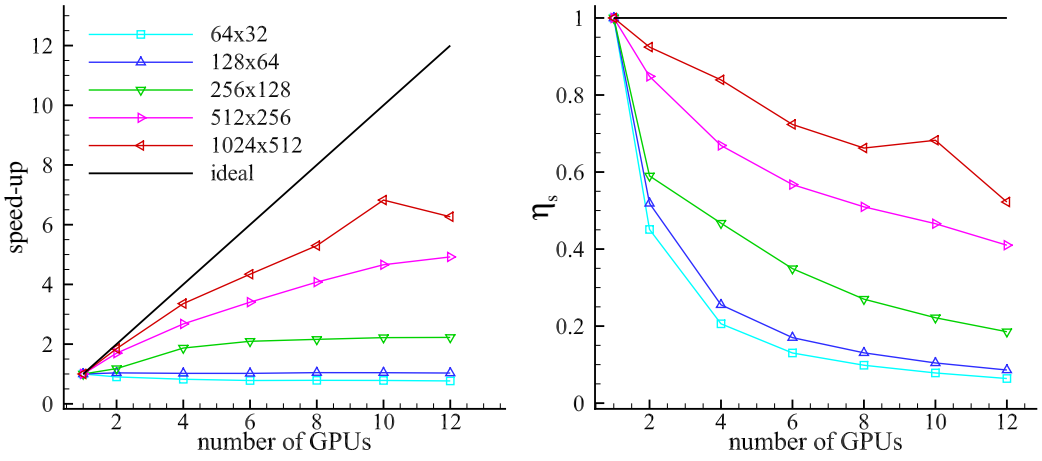


(b) 5 nodes

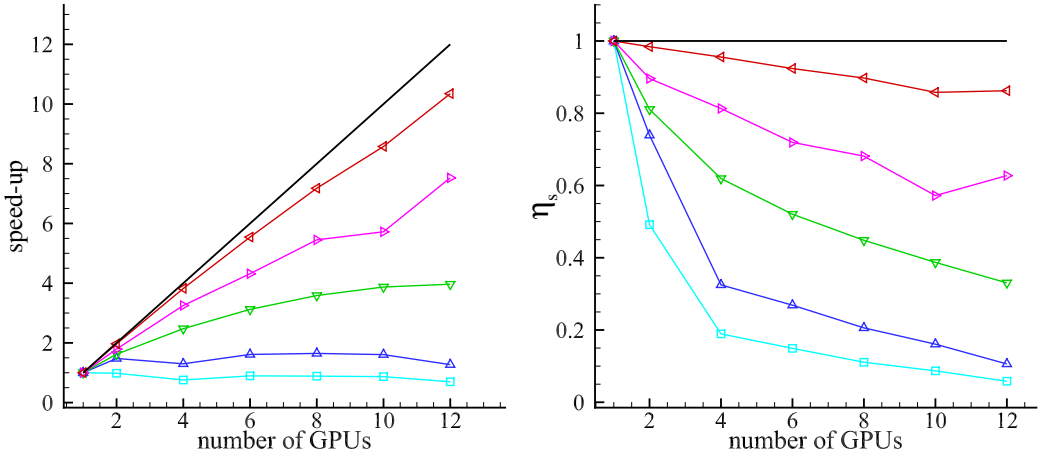


(c) 6 nodes

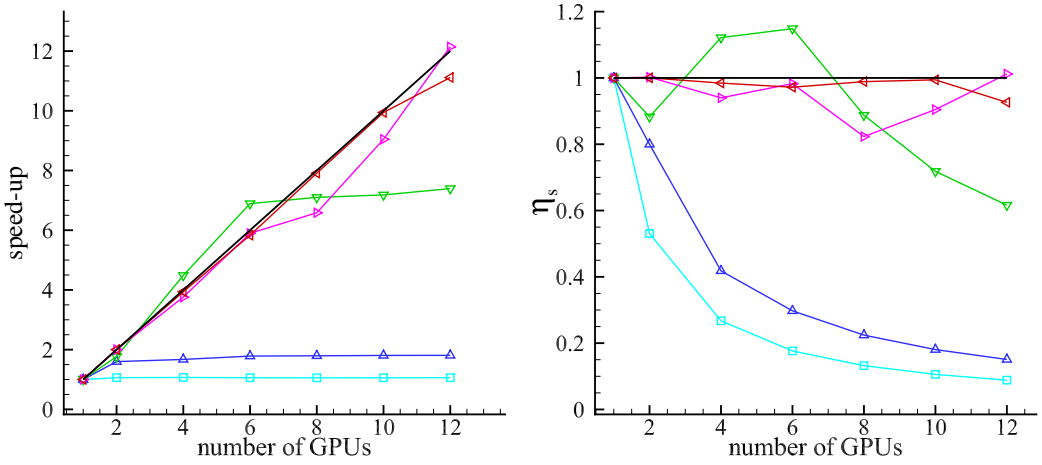
Figure 13: Performance results obtained by using full node computational power (each node has 12 CPU cores and 2 GPUs cards): (left) time per iteration (TpI) normalized by the one obtained on the coarsest grid as a function of the grid size normalized by the coarsest grid size. (right) TpI(CPU)/TpI(GPU) as a function of the grid size.



(a) Frozen

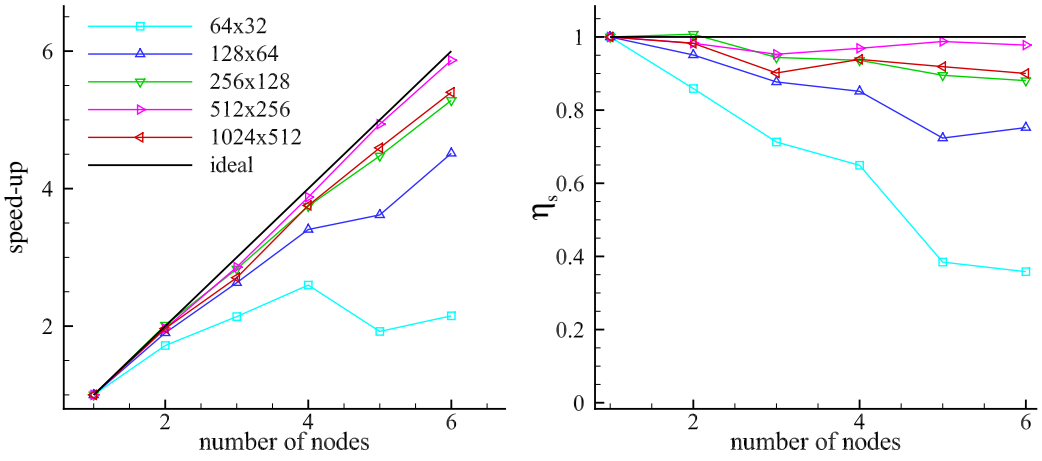


(b) Park

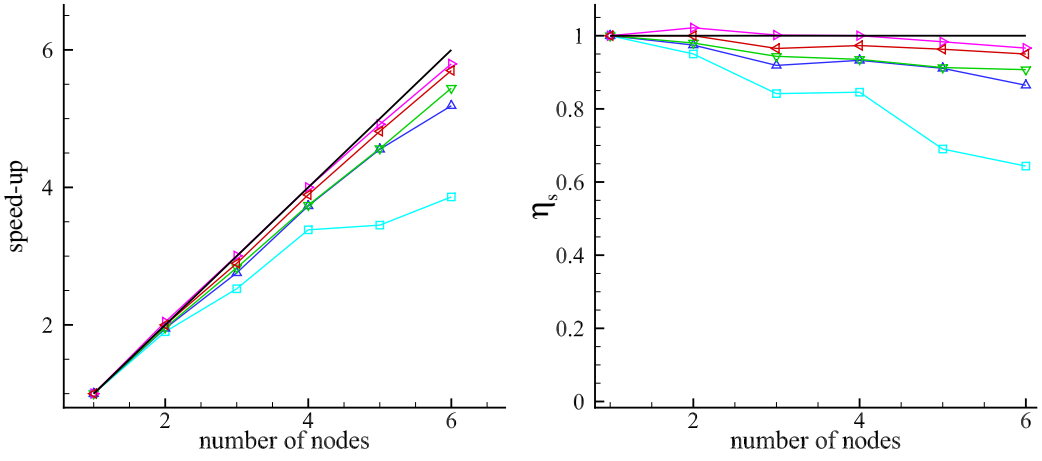


(c) StS

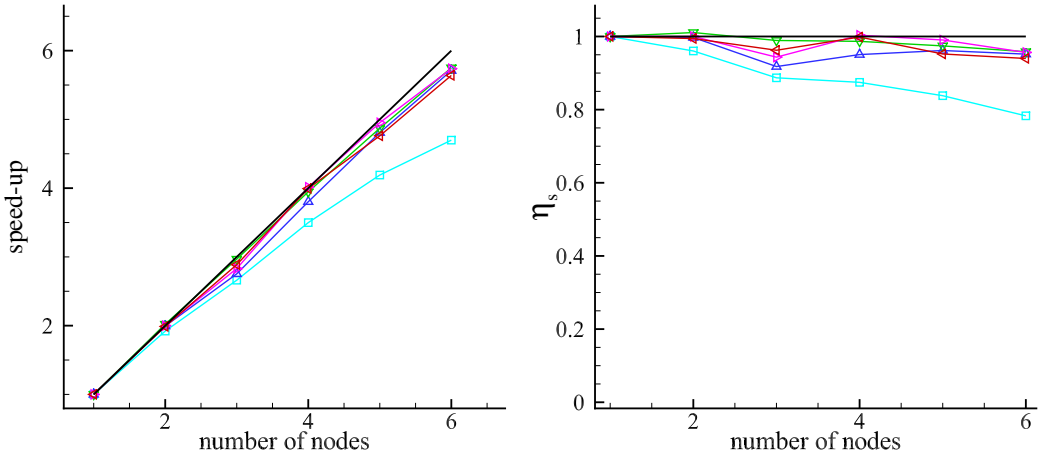
Figure 14: Strong scalability analysis for the MPI-CUDA implementation: (left) speed-up; (right) efficiency.



(a) Frozen

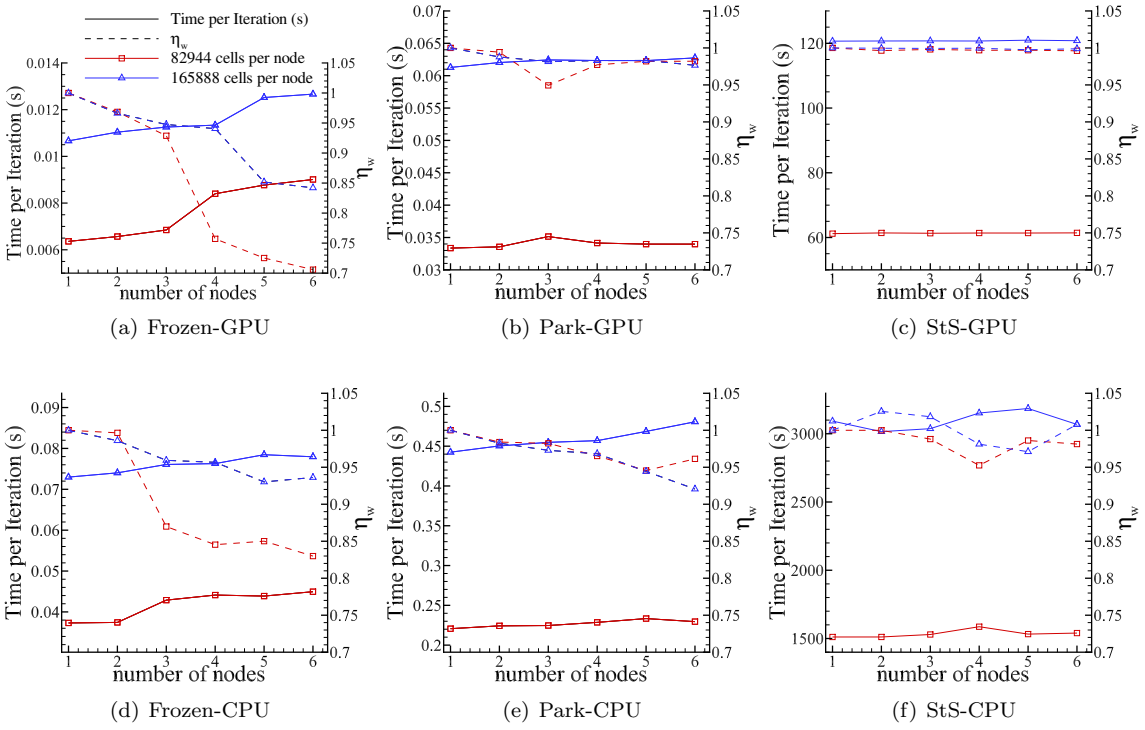


(b) Park



(c) StS

Figure 15: Strong scalability analysis for the pure MPI implementation: (left) speed-up; (right) efficiency.



(a) Frozen-GPU (b) Park-GPU (c) StS-GPU

(d) Frozen-CPU (e) Park-CPU (f) StS-CPU

Figure 16: Weak scalability analysis for the MPI-CUDA (top) and the pure MPI-CPU implementations (bottom).

Table 3: Time per iterations obtained on the computational grid that includes 1024×512 fluid cells by using: a single-core CPU, a single GPU, 72 cores-CPU and 12 GPUs.

Model	single-core CPU TpI (s)	single GPU TpI (s)	6 nodes CPU TpI (s)	6 nodes GPU TpI (s)
Frozen	2.36	$5.45 * 10^{-2}$	$4.12 * 10^{-2}$	$8.70 * 10^{-3}$
Park	$1.59 * 10^{1}$	$3.61 * 10^{-1}$	$2.46 * 10^{-1}$	$3.48 * 10^{-2}$
StS	$1.07 * 10^{5}$	$7.64 * 10^{2}$	$1.70 * 10^{3}$	$6.87 * 10^{1}$

starting from the 128×64 grid size an excellent TpI(CPU)/TpI(GPU) was found with a peak of about 144. As concerns the Park and the Frozen models smaller TpI(CPU)/TpI(GPU), but still appreciable (about 40), were found. To give an idea of the computational cost, the absolute time per iteration obtained on the finest grid by using the different models on both single-core CPU and single GPU are provided in Tab. 3. The time per iteration obtained when the StS model is employed is too large also with the use of a GPU. Therefore, a multi-GPU approach is mandatory.

In order to investigate the MPI parallel performance of both pure MPI-CPU and MPI-CUDA implementations a campaign of simulations were carried out by varying the number of CPUs and GPUs. Specifically, the single node was considered as the smallest unit and the same simulations (with the exclusion of the smallest grid, i.e. 32×16) performed on the single-core CPU and on the single GPU were carried out on one node, two node etc. Figure 12 and 13 show the results varying the number of nodes in the same terms used to obtain the results in Fig. 11. Obviously, this time the CPU computations are also parallel and they will be affected by the Amdahl's law. The effects are smaller compared to those shown by GPU computations but are quite evident running on six node, i.e., on 72 cores. As concerns the GPU computations, in agreement with the Amdahl's law, the ideal behavior is reached with finer grid sizes as the number of nodes is increased.

Again, excellent TpI(CPU)/TpI(GPU) were obtained with the StS model. Specifically, when running on 6 nodes, i.e. 72 CPU's cores and 12 GPUs, a peak of about 26 was found, that means a TpI(CPU)/TpI(GPU) of 156 if the single GPU against the single-core CPU is considered. In Tab. 3 are also given the absolute time per iteration obtained with the finest grid on 6 nodes.

All the data collected in the previous simulations were reworked to make a strong scalability analysis. Here, the speed-up is defined as the ratio between the time per iteration taken on the smallest computational unit, i.e., one GPU and one node-CPU for the MPI-CUDA and pure MPI-CPU implementations, and the one taken when running on n computational units,

$$\text{speed-up} = \frac{\text{TpI(1)}}{\text{TpI(n)}}, \tag{46}$$

whereas efficiency is defined as

$$\eta_s = \frac{\text{speed-up}}{\text{n}} = \frac{\text{TpI(1)}}{\text{n} \cdot \text{TpI(n)}}. \tag{47}$$

Figure 14 and 15 show the speed-up and the efficiency of the MPI-CUDA and of the pure MPI-CPU implementations. In agreement with previous observations the parallel performance improve with increasing problem size and model complexity. As concerns the MPI-CUDA computations the frozen case doesn't show a linear speed-up for any of the grids investigated. Whereas, when the Park model is employed a linear speed-up was found with the finest grid. Finally, with the most complex model, i.e. the StS, excellent speed-up were found starting from the computational grid that includes 256×128 fluid cells. As far as the pure MPI-CPU implementation concerns, due to the much smaller number of cores involved, compared to the GPU case, the results show excellent performance, with speed-up very close to the ideal behaviour, for a wider range of grids and models.

A further proof of the quality of the approach and of the resulting code was obtained by analyzing the weak scalability, which was evaluated by keeping constant the number of computational cells per node. Two different sizes were considered, i.e., 82944 and 165888 cells per node. Figure 16 shows both the time per iteration and the efficiency as a function of the number of nodes for both the MPI-CUDA and the pure MPI-CPU implementations. In the ideal case, the time per iteration is constant by increasing the number of nodes, thus efficiency is defined as

$$\eta_w = \frac{\text{TpI(1)}}{\text{TpI(n)}}. \tag{48}$$

Very good results were obtained when chemical kinetics is activated with a time per iteration that remains fairly constant thus providing efficiencies larger than 95% for almost all cases. Performance deteriorates when the frozen case is considered, especially with the use of multi-GPU. However, the worst efficiency still remains above 70%.

5. Conclusions

This paper demonstrates the accuracy and the feasibility of two-dimensional fluid dynamic computations of thermochemical non-equilibrium flows, typical of supersonic and hypersonic flows, by means of detailed state-to-state (StS) air kinetics by using the aid of multi-GPU computing.

The accuracy of such an approach was demonstrated by showing that the StS outcomes are in better agreement with the experimental findings with respect to those provided by the well established multi-temperature model proposed by Park.

In fact, StS models are devised from fundamentals chemico-physical theories and allow one to gain deep insights into thermochemical non-equilibrium phenomena. The profusion and accuracy of data provided by numerical simulations, using such models, cannot be obtained by experimental measurements and can be useful to improve the design of hypersonic vehicles or as a reference in order to develop or assess simplified models.

Efficient StS computations were made possible by using the emerging GPU technology. Indeed, it is increasingly accepted that GPUs outperform CPUs in terms of both flops and power efficiency and they

represent one of the most promising equipment for future HPC (High Performance Computing) systems. The MPI-CUDA approach implemented in this work allowed to us to efficiently scale the code across a multiple-nodes GPU cluster showing excellent scalability performance. It was found that, in agreement with the Amdahl's and Gustafson's laws, the use of GPUs is justified only with large size problems or when very expensive models, such as the StS one, are employed [32]. In such cases, comparing the single GPU against the single-core CPU performance, excellent speed-ups, up to 156, were found.

Acknowledgement

This research has been supported by grant n. PON03PE-00067-6 APULIA SPACE.

References

[1] S. Surzhikov, High-enthalpy radiating flows in aerophysics, in: G. Colonna, A. D'Angola (Eds.), Plasma Modeling: Methods and Applications, Plasma Physics Series, Bristol: IOP Publishing Ltd, 2016, Ch. 12. doi:10.1088/978-0-7503-1200-4ch12.

[2] F. Bonelli, L. Cutrone, R. Votta, A. Viggiano, V. Magi, Preliminary Design of a Hypersonic Air-breathing Vehicle, in: 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference 11–14 April 2011, San Francisco, California, International Space Planes and Hypersonic Systems and Technologies Conferences, American Institute of Aeronautics and Astronautics, 2011. doi:10.2514/6.2011-2319.

[3] F. Grasso, M. Marini, G. Ranuzzi, S. Cuttica, B. Chanetz, Shock-Wave/Turbulent Boundary-Layer Interactions in Nonequilibrium Flows, AIAA Journal 39 (11) (2001) 2131–2140. doi:10.2514/2.1209.

[4] C. Park, Nonequilibrium Hypersonic Aerothermodynamics, John Wiley & Sons, 1990.

[5] C. E. Treanor, P. V. Marrone, Effect of dissociation on the rate of vibrational relaxation, The Physics of Fluids 5 (9) (1962) 1022–1026. doi:10.1063/1.1724467.

[6] G. Colonna, M. Tuttafesta, M. Capitelli, D. Giordano, Non-Arrhenius NO Formation Rate in One-Dimensional Nozzle Airflow, Journal of Thermophysics and Heat Transfer 13 (3) (1999) 372–375. doi:10.2514/2.6448.

[7] G. Colonna, L. D. Pietanza, M. Capitelli, Recombination-Assisted Nitrogen Dissociation Rates Under Nonequilibrium Conditions, Journal of Thermophysics and Heat Transfer 22 (3) (2008) 399–406. doi:10.2514/1.33505.

[8] E. Josyula, W. F. Bailey, Vibration-dissociation coupling using master equations in nonequilibrium hypersonic blunt-body flow, Journal of Thermophysics and Heat Transfer 15 (2) (2001) 157–167.

[9] E. Josyula, W. F. Bailey, Vibrational relaxation and population depletion of nitrogen in hypersonic flows, sc AIAA 2002-0200 (2002).

[10] M. Capitelli, R. Celiberto, G. Colonna, F. Esposito, C. Gorse, K. Hassouni, A. Laricchiuta, S. Longo, Fundamentals Aspects of Plasma Chemical Physics: Kinetics, Springer, New York, 2016.

[11] M. Capitelli, G. Colonna, G. D'Ammando, V. Laporta, A. Laricchiuta, The role of electron scattering with vibrationally excited nitrogen molecules on non-equilibrium plasma kinetics, Phys. Plasmas 20 (10) (2013) 101609. doi:10.1063/1.4824003.

[12] G. Colonna, L. Pietanza, G. D'Ammando, Self-consistent kinetics, in: G. Colonna, A. D'Angola (Eds.), Plasma Modeling: Methods and Applications, Plasma Physics Series, Bristol: IOP Publishing Ltd, 2016, Ch. 12. doi:10.1088/978-0-7503-1200-4ch8.

[13] L. Cutrone, M. Tuttafesta, M. Capitelli, A. Schettino, G. Pascazio, G. Colonna, 3d nozzle flow simulations including state-to-state kinetics calculation, Proceedings of the XXIX international symposium on rarefied gas dynamics, AIP Conf. Proc. 1628 1154.

[14] D. Giordano, V. Bellucci, G. Colonna, M. Capitelli, I. Armenise, C. Bruno, Vibrationally relaxing flow of n past an infinite cylinder, Journal of thermophysics and heat transfer 11 (1) (1997) 27–35.

[15] A. Guy, A. Bourdon, M.-Y. Perrin, Consistent multi-internal-temperatures models for nonequilibrium nozzle flows, Chemical Physics 420 (11) (2013) 15–24. doi:10.1016/j.chemphys.2013.04.018.

[16] T. E. Magin, M. Panesi, A. Bourdon, R. L. Jaffe, D. W. Schwenke, Coarse-grain model for internal energy excitation and dissociation of molecular nitrogen, Chemical Physics 398 (2012) 90–95. doi:10.1016/j.chemphys.2011.10.009.

[17] M. Tuttafesta, G. Colonna, G. Pascazio, Computing unsteady compressible flows using Roes flux-difference splitting scheme on GPUs, Computer Physics Communications 184 (6) (2013) 1497–1510. doi:10.1016/j.cpc.2013.01.018.

[18] M. Tuttafesta, G. Pascazio, G. Colonna, Multi-GPU unsteady 2D flow simulation coupled with a state-to-state chemical kinetics, Computer Physics Communications 207 (2016) 243–257. doi:10.1016/j.cpc.2016.07.016.

[19] H. P. Le, J.-L. Cambier, L. K. Cole, GPU-based flow simulation with detailed chemical kinetics, Computer Physics Communications 184 (3) (2013) 596–606. doi:10.1016/j.cpc.2012.10.013.

[20] P. Dunzlaff, R. Strauss, M. S. Potgieter, Solving Parker's transport equation with stochastic differential equations on GPUs, Computer Physics Communications 192 (2015) 156–165. doi:10.1016/j.cpc.2015.03.008.

[21] M. Eisenbach, J. Larkin, J. Lutjens, S. Rennich, J. H. Rogers, GPU acceleration of the locally selfconsistent multiple scattering code for first principles calculation of the ground state and statistical physics of materials, Computer Physics Communications 211 (2017) 2 – 7, High Performance Computing for Advanced Modeling and Simulation of Materials.

doi:10.1016/j.cpc.2016.07.013.
URL http://www.sciencedirect.com/science/article/pii/S0010465516301953
[22] W. Jia, J. Wang, X. Chi, L.-W. Wang, GPU implementation of the linear scaling three dimensional fragment method for large scale electronic structure calculations, Computer Physics Communications 211 (2017) 8 – 15, High Performance Computing for Advanced Modeling and Simulation of Materials. doi:10.1016/j.cpc.2016.07.003.
URL http://www.sciencedirect.com/science/article/pii/S0010465516301850
[23] A. Mena, J. M. Ferrero, J. F. R. Matas, GPU accelerated solver for nonlinear reactiondiffusion systems. application to the electrophysiology problem, Computer Physics Communications 196 (2015) 280 – 289. doi:10.1016/j.cpc.2015.06.018.
URL http://www.sciencedirect.com/science/article/pii/S0010465515002635
[24] T. D. Nguyen, GPU-accelerated tersoff potentials for massively parallel molecular dynamics simulations, Computer Physics Communications 212 (2017) 113 – 122. doi:10.1016/j.cpc.2016.10.020.
URL http://www.sciencedirect.com/science/article/pii/S0010465516303393
[25] J. Spiechowicz, M. Kostur, L. Machura, GPU accelerated monte carlo simulation of brownian motors dynamics with {CUDA}, Computer Physics Communications 191 (2015) 140 – 149. doi:10.1016/j.cpc.2015.01.021.
URL http://www.sciencedirect.com/science/article/pii/S0010465515000417
[26] E. Elsen, P. LeGresley, E. Darve, Large calculation of the flow over a hypersonic vehicle using a GPU, Journal of Computational Physics 227 (24) (2008) 10148–10161. doi:10.1016/j.jcp.2008.08.023.
[27] B. Brock, A. Belt, J. J. Billings, M. Guidry, Explicit integration with GPU acceleration for large kinetic networks, Journal of Computational Physics 302 (2015) 591–602. doi:10.1016/j.jcp.2015.09.013.
[28] D. Komatitsch, G. Erlebacher, D. Göddeke, D. Michéa, High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster, Journal of Computational Physics 229 (20) (2010) 7692–7714. doi:10.1016/j.jcp.2010.06.024.
[29] K. E. Niemeyer, C.-J. Sung, Accelerating moderately stiff chemical kinetics in reactive-flow simulations using GPUs, Journal of Computational Physics 256 (2014) 854–871. doi:10.1016/j.jcp.2013.09.025.
[30] P. Norman, P. Valentini, T. Schwartzentruber, GPU-accelerated Classical Trajectory Calculation Direct Simulation Monte Carlo applied to shock waves, Journal of Computational Physics 247 (2013) 153–167. doi:10.1016/j.jcp.2013.03.060.
[31] D. Priimak, Finite difference numerical method for the superlattice Boltzmann transport equation and case comparison of CPU(C) and GPU(CUDA) implementations, Journal of Computational Physics 278 (2014) 182–192. doi:10.1016/j.jcp.2014.08.028.
[32] A. Khajeh-Saeed, J. B. Perot, Direct numerical simulation of turbulence using GPU accelerated supercomputers, Journal of Computational Physics 235 (2013) 241–257. doi:10.1016/j.jcp.2012.10.050.
[33] F. Salvadore, M. Bernardini, M. Botti, GPU accelerated flow solver for direct numerical simulation of turbulent flows, Journal of Computational Physics 235 (2013) 129–142. doi:10.1016/j.jcp.2012.10.012.
[34] C.-C. Su, M. R. Smith, F.-A. Kuo, J.-S. Wu, C.-W. Hsieh, K.-C. Tseng, Large-scale simulations on multiple Graphics Processing Units (GPUs) for the direct simulation Monte Carlo method, Journal of Computational Physics 231 (23) (2012) 7932–7958. doi:10.1016/j.jcp.2012.07.038.
[35] [link].
URL https://www.top500.org/green500/lists/2016/11/
[36] [link].
URL https://www.olcf.ornl.gov/summit/
[37] [link].
URL http://computation.llnl.gov/computers/sierra-advanced-technology-system
[38] M. de la Asunción, J. M. Mantas, M. J. Castro, E. D. Fernández-Nieto, An MPI-CUDA implementation of an improved Roe method for two-layer shallow water systems, Journal of Parallel and Distributed Computing 72 (9) (2012) 1065–1072. doi:10.1016/j.jpdc.2011.07.012.
[39] E. Calore, A. Gabbana, J. Kraus, E. Pellegrini, S. F. Schifano, R. Tripiccione, Massively parallel lattice–Boltzmann codes on large GPU clusters, Parallel Computing 58 (2016) 1–24. doi:10.1016/j.parco.2016.08.005.
[40] G. Colonna, M. Tuttafesta, M. Capitelli, D. Giordano, NO formation in one dimensional air nozzle flow with state-to-state vibrational kinetics: The influence of O2(v)+N=NO+O reaction, Journal of Thermophysics and Heat Transfer 14(3) (2000) 455–456.
[41] G. Colonna, M. Tuttafesta, M. Capitelli, D. Giordano, Influence on dissociation rates of the state-to-state vibrational kinetics of nitrogen in nozzle expansion, in: R. Brun, R. Campargue, R. Gatignol, J.-C. Lengrand (Eds.), 21th International symposium on rarefied gas dynamics, Vol. 2, 1999, pp. 281–288.
[42] M. Capitelli, G. Colonna, A. D'Angola, Fundamental Aspects of Plasma Chemical Physics: Thermodynamics, 1st Edition, Vol. 66 of Atomic, Optical, and Plasma Physics, Springer, New York, 2011.
[43] F. Esposito, I. Armenise, M. Capitelli, N–N_2 state to state vibrational-relaxation and dissociation rates based on quasi-classical calculations, Chemical Physics 331 (1) (2006) 1–8.
[44] F. Esposito, I. Armenise, G. Capitta, M. Capitelli, O-o_2 state to state vibrational–relaxation and dissociation rates based on quasiclassical calculations, Chemical Physics 351 (2008) 91–98.
[45] G. Colonna, L. D. Pietanza, M. Capitelli, Recombination-Assisted Nitrogen Dissociation Rates Under Nonequilibrium Conditions, Journal of Thermophysics and Heat Transfer 22 (2008) 399–406.
[46] M. Capitelli, R. Celiberto, G. Colonna, F. Esposito, C. Gorse, K. Hassouni, A. Laricchiuta, S. Longo, Fundamental Aspects of Plasma Chemical Physics: Kinetics, Vol. 85, Springer Science & Business Media, 2015.
[47] D. Bose, G. V. Candler, Thermal rate constants of the N_2+O→NO+N reaction using ab initio^3 A and ^{3} A potential energy surfaces, The Journal of Chemical Physics 104 (8) (1996) 2825. doi:10.1063/1.471106.

[48] D. Bose, G. V. Candler, Thermal rate constants of the O_2+N→NO+O reaction based on the $A^2$and A^4 potential-energy surfaces , The Journal of Chemical Physics 107 (16) (1997) 6136. doi:10.1063/1.475132.

[49] C. Park, Review of chemical-kinetic problems of future NASA missions, I: Earth entries, Journal of Thermophysics and Heat Transfer 7 (3) (1993) 385–398. doi:10.2514/3.431.

[50] J. Hao, J. Wang, C. Lee, Numerical study of hypersonic flows over reentry configurations with different chemical nonequilibrium models, Acta Astronautica 126 (2016) 1–10. doi:10.1016/j.actaastro.2016.04.014.

[51] C. Park, Problems of Rate Chemistry in the Flight Regimes of Aeroassisted Orbital Transfer Vehicles, in: Progress in Astronautics and Aeronautics, Thermal Design of Aeroassisted Orbital Transfer Vehicles, American Institute of Aeronautics and Astronautics, pp. 511–537. doi:10.2514/5.9781600865718.0511.0537.

[52] C. Park, Two-temperature interpretation of dissociation rate data for N_2 and O_2, in: 26th Aerospace Sciences Meeting Reno, NV, U.S.A., January 11–14, Vol. AIAA–88–0458, 1988. doi:10.2514/6.1988-458.

[53] R. C. Millikan, D. R. White, Systematics of Vibrational Relaxation, The Journal of Chemical Physics 39 (12) (1963) 3209. doi:10.1063/1.1734182.

[54] W. Ran, W. Cheng, F. Qin, X. Luo, GPU accelerated CESE method for 1D shock tube problems, Journal of Computational Physics 230 (2011) 8797–8812. doi:10.1016/j.jcp.2011.08.026.

[55] H. C. Yee, A Class of High–Resolution Explicit and Implicit Shock–Capturing Methods, Technical Memorandum 101088, NASA (February 1989).

[56] J. L. Steger, R. F. Warming, Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods, Journal of Computational Physics 40(2) (1981) 263–293. doi:10.1016/0021-9991(81)90210-2.

[57] K. H. Kim, C. Kim, O.-H. Rho, Methods for the Accurate Computations of Hypersonic Flows: I. AUSMPW+Scheme, Journal of Computational Physics 174 (1) (2001) 38–80. doi:10.1006/jcph.2001.6873.

[58] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, Journal of Computational Physics 32 (1) (1979) 101–136. doi:10.1016/0021-9991(79)90145-1.

[59] J. Sanders, E. Kandrot, CUDA by Example, Addison-Wesley, New-York, 2011.

[60] S. Nonaka, H. Mizuno, K. Takayama, C. Park, Measurement of shock standoff distance for sphere in ballistic range, Journal of Thermophysics and Heat Transfer 14 (2) (2000) 225–229. doi:10.2514/2.6512.

[61] M. Furudate, S. Nonaka, K. Sawada, Behavior of Two-Temperature Model in Intermediate Hypersonic Regime, Journal of Thermophysics and Heat Transfer 13 (4) (1999) 424–430. doi:10.2514/2.6480.

[62] R. Lobb, Experimental Measurement of Shock Detachment Distance on Sphere Fired in Air at Hypervelocities, in: The High Temperature Aspects of Hypersonic Flow – Proceedings of the AGARD–NATO Specialists' Meeting Sponsored by the Fluid Dynamics Panel of AGARD held at the Technical Centre for Experimental Aerodynamics Rhode-Saint-Genèse, Belgium 3–6 April 1962, Vol. 68, 1964, pp. 519–527. doi:10.1016/B978-1-4831-9828-6.50031-X.

[63] P. S. Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, 2011.

[64] G. Amdahl, Validity of the single processor approach to achieving large scale computing capabilities, in: Proceedings of the American Federation of Information Processing Societies Conference (Atlantic City, N.J., Apr. 18-20), Vol. 30 (2), AFIPS Press, Reston. Va., 1967, pp. 483–485.

[65] J. Gustafson, Reevaluating Amdahl's law, Communication of the ACM 31 (5) (1988) 532–533.