ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **12** (2023), no.1, 45–61
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.12.1.3

A SCALABLE HYBRID CPU-GPU COMPUTATIONAL FRAMEWORK FOR A FINITE ELEMENT-BASED AIR QUALITY MODEL

Abdoulaye Samaké[1], Mahamadou Alassane[2], Amadou Mahamane[3], and Ouaténi Diallo[4]

ABSTRACT. We propose a scalable computational framework for the hybrid CPU-GPU implementation ofa traffic-induced and finite element-based air quality model. The hybrid computing paradigm we investigate consists in combining the CPU-based distributed-memory programming approach using Message Passing Interface (MPI) and a GPU programming model for the finite element numerical integration using Compute Unified Device Architecture (CUDA), a general purpose parallel computing platform released by NVIDIA Corporation and featured on its own GPUs. The scalability results obtained from numerical experiments on two major road traffic-induced air pollutants, namely the fine and inhalable particulate matter $PM_{2.5}$ and PM_{10}, are illustrated. These achievements, including speedup and efficiency analyses, support that this framework scales well up to 256 CPU cores used concurrently with GPUs from a hybrid computing system.

1. INTRODUCTION

Air pollution is a real concern for public health and the environment. It is nowadays considered as an increased priority for joint action worldwide to fight against its harmful effects. Ambient air pollution comes from natural and anthropogenic

[1]*corresponding author*
2020 *Mathematics Subject Classification.* 65N30, 65Y05, 65Y20, 76M10.
Key words and phrases. Hybrid computing, mathematical modeling, numerical simulation, finite element, air quality.
Submitted: 08.12.2022; *Accepted:* 23.12.2022; *Published:* 03.01.2023.

sources. Natural sources include wildfires, volcanic eruptions and dust storms. Anthropogenic sources are mostly resulting from the combustion of different kinds of fuels. They include motor vehicles, factories, industrial facilities and power plants.

Air pollution is particularly prevalent in urban areas subject to high population density and strong economic and industrial activities. Several illnesses like chronic obstructive pulmonary disease (COPD), schemic heart disease, stroke and lung cancer are associated with ambient air pollution, classified as carcinogenic to humans by the International Agency for Research on Cancer (IARC) [20]. The World Health Organization (WHO) estimates that 4.2 million premature deaths worldwide in 2016 were attributable to ambient air pollution [21]. Low-income and middle-income countries are the most experienced since 91% of the mentioned pollution-related deaths occur in these countries. The affects of air pollution, indoor and outdoor sources combined, on health, human capital and economic development are particularly severe in Africa [6]. In this continent, 16.3% of total deaths, i.e. about 1.1 million fatalities, in 2019 were associated with air pollution [19]. As a result, it appears that air pollution is now the second leading cause of death in Africa, surpassed only by HIV/AIDS.

Efforts have been made at local and global scales for reducing the adverse effects of air pollution. They are mainly based on the implementation of ambient air quality monitoring stations in order to better identify pollution sources and to assist in making decisions on binding policies. In contrast to advanced countries, there are very few of these regulatory grade monitors in use in Sub-Saharan Africa, mainly due to their high acquisition cost on the one hand and the limited financial resources of the countries on the other. This fact argues that this approach is not always adequate for developing countries and it is therefore essential to investigate alternative methods. Credible alternatives to these classical techniques include mathematical modeling and numerical simulation. They are fundamental and efficient tools widely used for understanding and predicting the dynamics of pollutants in the atmosphere [13]. They deal with the transport and diffusion of pollutants in the atmosphere and address the mechanism of chemical reactions and deposition processes. Rapid growth in computational power and recent advances in robust numerical methods have largely contributed to the remarkable success of these tools.

Let us focus now on the city of Bamako, the political and economic capital of the landlocked Sahelian country of Mali. Bamako is built on a territory that represents only 0.02% of the total area of the country but houses more than 12% of the Malian population. It is undoubtedly among the most polluted capitals in West Africa. Indeed, in this capital, the daily mean concentration of PM_{10}, an inhalable particulate matter with a diameter of $10\,\mu\mathrm{m}$ or less, nears $600\,\mu\mathrm{g/m^3}$ while the WHO guideline recommends a maximum daily limit of $45\,\mu\mathrm{g/m^3}$ [1]. In addition, the daily mean concentration of $PM_{2.5}$, the fine particulate matter with a diameter of $2.5\,\mu\mathrm{m}$ or less, in Bamako reaches $165\,\mu\mathrm{g/m^3}$, far above the maximum daily limit recommended by WHO and fixed to $15\,\mu\mathrm{g/m^3}$ [22].

In order to make our contribution, we have proposed a unified framework for the mathematical modeling and numerical simulation of traffic-induced air pollution in Bamako [27], We were interested in a deterministic model, the so-called chemical transport model, characterized by its high accuracy and flexibility and suited for practical and long term simulation-based applications. Performing high spatial resolution simulations using such a complex three-dimensional (3D) model is very challenging and requires high computational power. In this context, we have proposed in [28] a parallel computational framework centered on the CPU-based distributed-memory programming approach using Message Passing Interface (MPI) library and modern C++ features. This work has allowed to accurately reproduce the temporal and spatial dynamics of $PM_{2.5}$ and PM_{10}, two traffic-induced primary pollutants widely used in the survey of air quality, while presenting good speedup and efficiency properties up to 128 CPU cores.

The objective of this paper is to present an extension of this existing CPU-based parallel framework to a hybrid computational approach that consists in combining Central Processing Units (CPUs) and Graphics Processing Units (GPUs) to obtain better computing performance by using more computational resources on the one hand, and on the other hand, to allow the achievement of simulations with finer spatial resolutions for more accurate knowledge on the urban air pollution patterns.

This paper is organized as follows. Section 2 describes the atmospheric chemical transport model we are interested in. Section 3 focuses on the spatial discretization and the time integration of the model. Section 4 deals with the hybrid CPU-GPU implementation of the model. In Section 5, we present the scalability results and discuss their attainment. The main conclusions are summarized in Section 6.

2. Model Description

Let Ω be a bounded domain of $\mathbb{R}^3$ and $\partial\Omega$ the boundary of Ω. We consider a partition of $\partial\Omega$ of the form $\partial\Omega = \partial\Omega_{\mathrm{G}} \cup \partial\Omega_{\mathrm{H}} \cup \partial\Omega_{\mathrm{L}}$, where $\partial\Omega_{\mathrm{G}}$ denotes the ground boundary of Ω, $\partial\Omega_{\mathrm{H}}$ is the upper limit of Ω and $\partial\Omega_{\mathrm{L}}$ represents the lateral boundaries of Ω. Let c be the vector field of concentrations, where the ith component ϕ_i denotes the scalar concentration field of air pollutant tagged i. The spatial and temporal dynamics of the concentration ϕ_i in the domain Ω over the time interval $(0, T)$ are governed by the atmospheric chemical transport model (2.1), already introduced in [28].

$$
\text{(2.1)} \qquad \begin{cases} \dfrac{\partial \phi_i}{\partial t} + \nabla \cdot (\mathbf{u}\phi_i) - \nabla \cdot (\mu_i \nabla \phi_i) - \chi_i(c) & \\ \quad + \Lambda_i(\mathbf{x}, t)\phi_i = S_i(\mathbf{x}, t) & \text{in} \quad \Omega \times (0, T) \\ \phi_i(\mathbf{x}, 0) = c_{in} & \text{in} \quad \Omega \\ \mu_i \nabla \phi_i \cdot \mathbf{n} = v_i^d \phi_i - E_i & \text{on} \quad \partial\Omega_{\mathrm{G}} \times (0, T) \\ \mu_i \nabla \phi_i \cdot \mathbf{n} = 0 & \text{on} \quad \partial\Omega_{\mathrm{H}} \times (0, T) \\ \phi_i \mathbf{u} \cdot \mathbf{n} = \Phi_i & \text{on} \quad \partial\Omega_{\mathrm{L}} \times (0, T) \end{cases}
$$

The parameters and physical fields of the model (2.1) and their units are listed in Table 1. The vector field $\mathbf{n}$ is the unit outward normal vector to the boundary $\partial\Omega$. The variable $\mathbf{x} = (x, y, z)$ describes the spatial dimensions, where z represents the altitude. The variable t denotes the model time.

Table 1: Fields and physical parameters of the model (2.1).

Symbol	Name	Unit
c	concentration of pollutant	kg m^{-3}
$\mathbf{u}$	wind flow speed	m s^{-1}

μ	diffusion coefficient	$\mathrm{m}^2\ \mathrm{s}^{-1}$
χ	chemical sources	$\mathrm{kg\ m}^{-3}\ \mathrm{s}^{-1}$
Λ	scavenging coefficient	s^{-1}
S	source terms	$\mathrm{kg\ m}^{-3}\ \mathrm{s}^{-1}$
E	surface emissions	$\mathrm{kg\ m}^{-2}\ \mathrm{s}^{-1}$
v^d	dry deposition velocity	$\mathrm{m\ s}^{-1}$
c_{in}	initial concentration	$\mathrm{kg\ m}^{-3}$
Φ_i	advective mass flux	$\mathrm{kg\ m}^{-2}\ \mathrm{s}^{-1}$

In the model (2.1), it is assumed that there is no feedback between the flow fields and the pollutants. We suppose that the urban topography is homogeneous and that the flow is incompressible. The maximum altitude, denoted H, we are interested in is assumed to be included in the atmospheric boundary layer [10]. We consider a first-order chemical reaction, where the balance of physical and chemical processes can be written in the form $\chi_i = -\kappa_i \phi_i$, where κ_i denotes the reaction rate. The losses from wet deposition and scavenging are assumed to be negligible, which holds that $\Lambda_i = 0$. For more details about the deposition and scavenging processes and the chemical kinetics, readers can refer to [27]. The wind flow speed $\mathbf{u}$ is obtained from the meteorological data sources of the European Centre for Medium-Range Weather Forecasts (ECMWF). The initial concentration c_{in} and the advective mass fluxes Φ_i from external pollution sources to the computational domain are collected from the Copernicus Atmosphere Monitoring Service (CAMS), implemented by ECMWF.

An important aspect in the study of air pollution consists in understanding the essential properties of the pollutants of interest, in particular their characteristic time, also called residence time. It defines the average length of time a pollutant remains in the atmosphere before being removed during chemical reactions or scavenging and deposition processes. The characteristic times of the principal atmospheric species are reported in [31, Fig. 1.7].

The chemical reaction rate κ_i of a species tagged i is related to its characteristic time, denoted τ_i, by the following formula:

$$\kappa_i = \frac{1}{\tau_i} \tag{2.2}$$

Furthermore, the dry deposition velocity of this species can be computed from its characteristic time as follows:

$$v_i^d = H\tau_i, \tag{2.3}$$

where H is the maximum altitude of interest.

3. Model Discretization

The discretization is an essential step for the numerical solution of complex partial differential equations (EDP). It consists in transposing a continuous model into an equivalent discrete model that can be solved using numerical algorithms. The discretization of the model (2.1) will be done in two phases. First, the temporal discretization of the continuous model and then, the spatial discretization of the discrete-time model. The time integration of the model (2.1), already developed in [28], is achieved using the forward Euler method, a basic first-order numerical scheme. Following the assumptions listed in section 2 and introducing a discretization time-step $\Delta t \in \mathbb{R}_+^*$, this integration is presented as follows:

$$\frac{\phi_i^n - \phi_i^{n-1}}{\Delta t} + \mathbf{u}^n \cdot \nabla \phi_i^n - \nabla \cdot (\mu_i \nabla \phi_i^n) + \kappa_i \phi_i^n = S_i^n, \tag{3.1}$$

where $\phi_i^n = \phi_i(\mathbf{x}, t_n)$, $\mathbf{u}^n = \mathbf{u}(\mathbf{x}, t_n)$, $S_i^n = S_i(\mathbf{x}, t_n)$ and $t_n = n\Delta t$, $n \in \mathbb{N}^*$.

The spatial discretization of the model, earlier discussed in [28], is conducted using the finite element method [5, 17]. It is a poweful and general numerical approach that provides an approximate solution of complex problems, expressed in terms of partial differential equations (PDEs), in various fields of science and engineering. It is able to handle complex geometries and allow an insightful error analysis. The finite element method relies on the principle of minimization of the potential energy of the system. The main idea consists in finding an approximation of a problem written in the variational form in an infinite-dimensional Hilbert space V in a finite-dimensional subspace $V_h \subset V$.

Let $\mathcal{T}_\delta$ be a mesh of the domain Ω, where δ denotes the maximum characteristic length of the mesh. We denote $\mathcal{V}_{\Phi_i,\delta}$ the piecewise linear finite element space defined on $\mathcal{T}_\delta$ as follows:

$$\mathcal{V}_{\Phi_i,\delta} = \left\{ v \in C^0(\overline{\Omega}) \,\middle|\, v \text{ is linear on } K, K \in \mathcal{T}_\delta \right\} \bigcap H^1_{\Phi_i, \partial\Omega_{\mathrm{L}}}, \tag{3.2}$$

where $H^1_{\Phi_i,\partial\Omega_{\text{L}}}$ is the Hilbert-Sobolev space of order 1 satisfying the Dirichlet condition on $\partial\Omega_{\text{L}}$, see model (2.1). The space $C^0(\overline{\Omega})$ is the set of continuous functions defined on $\overline{\Omega}$, the closure of Ω. The variational (or weak) form of the problem (2.1) is written: find $\phi_i \in \mathcal{V}_{\Phi_i,\delta}$ such that

$$\int_\Omega \Big(\phi_i^n + \Delta t \mathbf{u}\cdot\nabla\phi_i^n - \Delta t \nabla\cdot(\mu_i\nabla\phi_i^n) + \Delta t\kappa_i\phi_i^n\Big)v = \int_\Omega \Big(\Delta t S_i(\mathbf{x},t) + \phi_i^{n-1}\Big)v, \tag{3.3}$$

for all $v \in \mathcal{V}_{0,\delta}$, where ϕ_i refers to ϕ_i^n. Integrating equation (3.3) by parts and considering the boundary conditions prescribed in (2.1), the following weak form holds: find $\phi_i \in \mathcal{V}_{\Phi_i,\delta}$ such that

$$\int_\Omega \mu_i\Delta t\nabla\phi_i\cdot\nabla v + \Big(\phi_i + \Delta t\mathbf{u}\cdot\nabla\phi_i + \Delta t\kappa_i\phi_i\Big)v - \int_{\partial\Omega_{\text{G}}} \Delta t v_i^d\phi_i v = R(v), \tag{3.4}$$

for all $v \in \mathcal{V}_{0,\delta}$, where

$$R(v) = \int_\Omega \Big(\Delta t S_i + \phi_i^{n-1}\Big)v - \int_{\partial\Omega_{\text{G}}} \Delta t E_i v.$$

The existence and uniqueness of the weak solution of (3.4) is ensured by the Lax-Milgram theorem [5], a key tool in finite element analysis. Denoting by Q the left-hand-side form of (3.4), this equation can be written:

$$Q(\phi_i, v) = R(v). \tag{3.5}$$

Let $\mathcal{B} = \Big\{\psi_j\Big\}_{j=1}^{N_\delta}$ be a set of basis functions of $\mathcal{V}_{\Phi_i,\delta}$. The approximate solution $\phi_i \in \mathcal{V}_{\Phi_i,\delta}$ can be expressed as

$$\phi_i = \sum_{j=1}^{N_\delta} \phi_{i,j}\psi_j, \tag{3.6}$$

where $\phi_{i,j}, j = 1,\ldots,N_\delta$, are nodal values. By replacing in (3.5) the trial function ϕ_i by its expression (3.6) and the test function v by the basis elements of $\mathcal{B}$, the following discrete form of the problem holds:

$$\sum_{j=1}^{N_\delta} \phi_{i,j} Q(\psi_j,\psi_k) = R(\psi_k), \quad \forall k = 1,\ldots,N_\delta. \tag{3.7}$$

The discrete problem (3.7) can finally be written as a linear system:

$$A\mathbf{x} = \mathbf{b}, \tag{3.8}$$

where $A \in \mathbb{R}^{N_\delta \times N_\delta}$ and $\mathbf{x}$ and $\mathbf{b}$ belong $\mathbb{R}^{N_\delta}$. The entries of the matrix A are computed by $a_{jk} = Q(\psi_k, \psi_j),\ 1 \leq j, k \leq N_\delta$. The components of the force vector $\mathbf{b}$ are determined by $b_j = R(\psi_j),\ 1 \leq j \leq N_\delta$. The vector solution $\mathbf{x}$ is composed of the unknown coefficients $\phi_{i,j}$.

4. Hybrid CPU-GPU Implementation

4.1. **Background.** Awesome progress in the design of powerful parallel computers and the recent advances in semiconductor manufacturing and multicore technologies have ushered a new era in the field of high-performance Computing (HPC). This consists in practice of aggregating computing resources from multiple servers in such a way as to deliver much higher computing power than standard servers for solving highly demanding and large complex problems in simulation-based applied science and engineering. It is indeed at the core of recent major scientific and technological breakthroughs. The investigation of advanced numerical methods and algorithms that can optimally exploit the full potential of modern computing architectures is a significant challenge and an integral part of academic research in the field of scientific computing.

The CPU-based distributed message passing paradigm, a dominant programming model used to parallelize applications across computing clusters, has been employed in [28] to implement a slightly simplified version of the model (2.1). However, the continued evolution of parallel computing architectures and recent trends in processor and memory technologies, marked by a rapid increase in CPU core count and a decrease in the memory capacity per core, have shown the limitations of this approach with respect to its performance and scalability. As a result, robust and quite successful hybrid approaches, coupling MPI [30] and shared-memory models such as OpenMP [3] and GPU [23], have been introduced as alternatives for overcoming these restrictions. They provide greater flexibility in parallel decomposition in mainstream heterogeneous computing systems and allow a significant reduction in the amount of memory consumed by read-only data structures that are replicated across CPU cores.

GPUs, originally designed to accelerate graphics intensive applications, are currently a promising and most important platform widely used in HPC. The success and attractiveness of this technology in this field is due, in part, to the fact that

GPUs have higher core count and memory bandwidth than CPUs and are much more energy efficient. These features make GPUs more flexible, allowing them to perform far fewer tasks at high speed and with massive data throughput and to support demanding compute workloads far better than CPUs. These advantages provide additional motivation of using the hydrid MPI-GPU programming model [24] in preference to pure MPI for parallel general-purpose computations, including the solution of numerically complex models. However, despite its immense strengths, this cooperative MPI-GPU approach is very challenging since it requires both a advanced knowledge of CUDA [4] or OpenCL [32] APIs and manually managed data communication between distributed CPUs and GPUs using MPI and GPU programming interfaces.

4.2. **Contribution.** In the literature, several works have been dedicated to the use of GPUs in the finite element computations [11, 15, 25, 29]. In General, the main steps involved in the finite element analysis can be recapped as follows: (i) definition of variational formulation (ii) discretization: mesh generation and definition of approximation space (iii) local assembly on each mesh element by numerical integration (iv) global assembly on the entire mesh (v) solution of the resulting linear system.

The most computationally and time consuming steps in finite element analysis are usually the assembly, including local and global, and the solution of the resulting sparse linear system [8]. Significant advances have already been made in the development of robust and scalable solvers [12, 26] for the efficient solution of sparse linear systems. So, we propose in this framework an efficient approach directed on boosting the performance of the finite element assembly by computing numerical integration, inherently parallel, in GPUs.

We first proceed in pre-processing to the generation of the serial computational mesh over the concerned region using Gmsh [7]. We consider a mesh of 4-node tetrahedral elements for three-dimensional linear finite element analysis. This serial mesh will then be partitioned, again in pre-processing, into several subdomains using a useful embedded feature of Gmsh called METIS [14]. The pre-built parallel finite element mesh is fully loaded separately and concurrently by each deployed CPU core, which will only retrieves the partition that fits its local CPU rank enlarged to inter-process (ghost) elements.

The task which succeeds consisted in building concurrently in each processor core over its local mesh a table of degrees of freedom (DOFs). It is achieved using a local DOF numbering from a minimum-bandwidth algorithm [16]. A global table of degrees of freedom, distributed across all processor cores and associated with the entire problem, is then built. It is connected to local tables of degrees of freedom through a relation called local-to-global map (LtGM). Creating this map is done once but requires global communication involving all CPU ranks.

Following these first tasks performed completely on the CPUs (hosts), all nodal data related to the local mesh and specific to each CPU rank are transferred from CPUs, on which they reside, to GPUs (device) using functions of CUDA kernel, prior launched asynchronously by the host. Then, the kernel execution starts on the device and the local element matrices generation by numerical integration is fully realized in parallel on GPUs. The GPU parallelization approach used for numerical integration consists in partitioning the finite elements into subsets according to the GPU ressource availability and the number of CUDA warps. The subsets are executed in serial on the device and the numerical integration over the elements in each subset is achieved in parallel within CUDA warps. A finite element will be assigned to each thread within a warp. Once all these calculations are completed on GPUs, element matrices and load vectors data are moved back to CPUs for further operations. The data transfers between the host (main) memory and the device memory are done through the PCI-Express bus. The hybrid algorithm proposed here has been designed to ensure better load balancing and to reduce host-GPU communication patterns, which remain a bottleneck for many heterogeneous applications [9] since the speed of such transfer is much lower than GPU bandwidth.

The parallel assembly of global right-hand side vector and global finite element matrix, created using PETSc [2] wrappers and stored in CSR format, is carried out on CPUs using LtGMs and element matrices and vectors earlier received from GPUs. The parallel algebraic operations involving data structures of matrices and vectors, including solvers and preconditioners for the solution of linear systems, are handled using advanced algebraic interfaces from PETSc wrappers.

5. Numerical Experiments

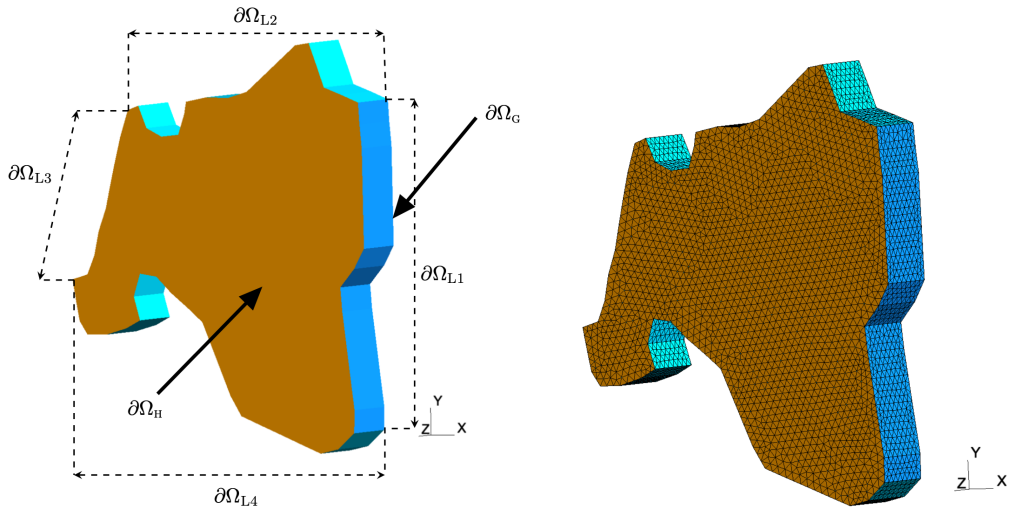
The numerical simulations are performed on two major air pollutants commonly used in air quality survey, namely the fine and inhalable particulate matter $PM_{2.5}$ and PM_{10}. Consisting of a mixture of solid and liquid particles, these pollutants are an important metric since they are among the most common traffic-related air pollutants with both short-term and long-term health impacts.

The maximum altitude of interest we consider is settled to $H = 10\,\mathrm{m}$. The corresponding physico-chemical parameters for PM2.5 are given by $\mu = 6.36 \times 10^{-6}$ $\mathrm{m^2 s^{-1}}$, $\kappa = 3.17 \times 10^{-6}$ $\mathrm{s^{-1}}$ and $v^d = H\kappa = 3.17 \times 10^{-5}$ $\mathrm{ms^{-1}}$. Likewise, the physico-chemical parameters for PM10 are reported by $\mu = 1.59 \times 10^{-6}$ $\mathrm{m^2 s^{-1}}$, $\kappa = 2.73 \times 10^{-5}$ $\mathrm{s^{-1}}$ and $v^d = H\kappa = 2.73 \times 10^{-4}$ $\mathrm{ms^{-1}}$. The three-dimensional computational domain and a mesh of 4-node tetrahedral elements over this domain with an average spatial resolution of 5 m are shown in Figure 1. The spatial discretization of the model is based on the first-order Lagrangian finite element approximation. The meteorological data, including wind speed and wind direction, are collected from European Centre for Medium-Range Weather Forecasts (ECMWF). The initial concentration and the advective mass fluxes from external pollution sources to the computational domain are collected from the Copernicus Atmosphere Monitoring Service (CAMS), implemented by ECMWF.

The input traffic emissions data for the model were generated in pre-processing from a model-based traffic simulations on major roads subject to heaviest traffic jams during daily rush hours. Traffic simulations have been released using a microscopic and continuous traffic simulation suite called SUMO [18]. The SUMO input parameters, such as traffic volume, traffic density, vehicle type, vehicle engine, average speed, are extracted from a national database provided by the Malian Department responsible for Transport and Infrastructure.

The numerical simulations have been performed at the *"Centre de Calcul, Modélisation et Simulation"* (CCMS) of the Faculty of Science and Technology (FST) of Bamako. CCMS hosts a cluster of ten compute nodes (servers) connected by an infiniband QDR network. There are two groups of these nodes, each consisting of two distinct machine types. In the first gathering of five Dell PowerEdge servers, each node has two Intel Xeon Silver 4110 processors with 8 cores running at 2.10 GHz and 64 GB of RAM. Still in this group, two nodes are featured with two

NVIDIA Tesla V100 GPUs each. In the second group of five HPE ProLiant servers, each node is equipped with two Intel Xeon E5-2623 v4 processors with 4 cores cadenced at 2.60 GHz and 16 GB of RAM. Based on Volta architecture, NVIDIA Tesla V100 GPU provides 5120 CUDA computational cores over 84 Streaming Multiprocessors (SM) and 32 GB onboard memory. It delivers up to 7 and 14 TFLOPS of peak floating-point performance at single and double precision, respectively.



(a) Three-dimensional domain of Bamako (b) Three-dimensional computational mesh

FIGURE 1. Three-dimensional computational domain and a mesh with an average spatial resolution of $5\,\mathrm{m}$. Source: [28].

The computational scalability is an important tool for expressing the performance of parallel algorithms [28]. It refers to two usual metrics, namely the strong scaling (or speedup) and weak scaling (or efficiency). The speedup is defined as the ratio of the time required to complete a given problem in serial on a single CPU core to the time spent to achieve the same problem in parallel on p ($p > 1$) CPU cores. Let T_1 be the time consumed for running a problem on a single CPU core and T_p be the parallel execution time of the same problem using p ($p > 1$) CPU cores. The speedup with p CPU cores is computed as follows:

$$S_p = T_1/T_p. \tag{5.1}$$

Furthermore, The efficiency with p ($p > 1$) CPU cores is computed as follows:

$$E_p = S_p/p. \tag{5.2}$$

Based on the expression (5.2), it can be shown that the best possible efficiency is reached when the speedup is linear, i.e. $S_p = p$. For large scale computing, the concepts of relative speedup and relative efficiency, a more general approach of the usual metrics of speedup and efficiency, have been introduced [33]. Let T_r be the parallel execution time of a problem on r CPU cores and T_p be the parallel execution time of the same problem on p ($r < p$) CPU cores. The speedup relative to r CPU cores when using p cores is defined as follows:

$$S_{r,p} = T_r/T_p. \tag{5.3}$$

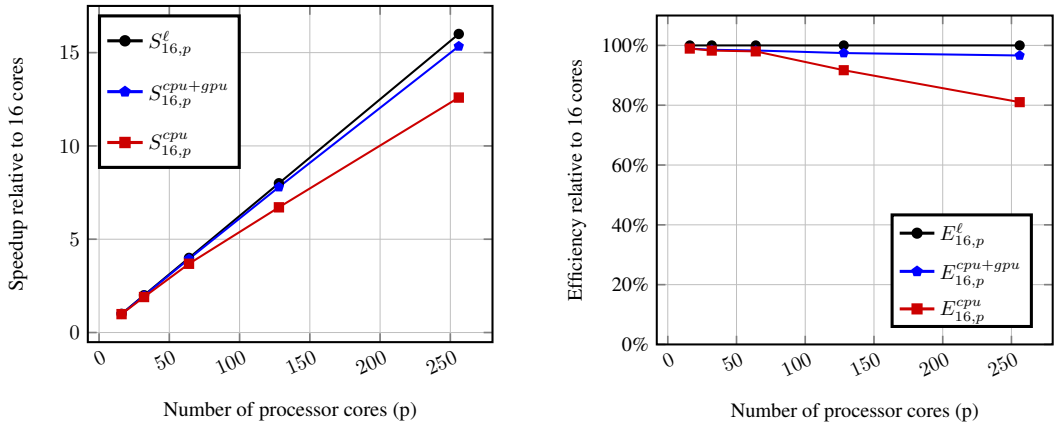
The best possible relative speedup is the linear speedup computed as $S^{\ell}_{r,p} = p/r$. Consequently, the efficiency relative to r CPU cores when using p cores, expressed as a percentage, is defined as follows:

$$E_{r,p} = rS_{r,p}/p. \tag{5.4}$$

From equations (5.3) and (5.4), we denote $S^{cpu}_{r,p}$ and $S^{cpu+gpu}_{r,p}$ the speedup relative to r CPU cores when using p cores (pure MPI) and the speedup relative to r CPU cores when employing p cores and the GPUs (hybrid MPI-GPU), respectively. Similarly, $E^{cpu}_{r,p}$ and $E^{cpu+gpu}_{r,p}$ represent the efficiency relative to r CPU cores when using p cores (pure MPI) and the efficiency relative to r CPU cores when employing p cores and the GPUs (hybrid MPI-GPU), respectively.

The strong and weak scalability results up to 256 CPU cores, achieved on a computational mesh of about 3.94×10^6 tetrahedral elements, when using pure MPI and hybrid MPI-GPU computing approaches are plotted in Figure 2.

According to the strong scaling analysis presented in Figure 2(a), the speedup relative to 16 cores when using 256 CPU cores is $S^{cpu}_{16,256} \approx 12.77$. This fits a performance gain of about 79.81% compared to the linear relative speedup $S^{\ell}_{16,256} = 16$. This speedup, which is below expectations, has been improved thanks to the use of GPUs, employed together with CPU cores, to achieve a relative speedup of $S^{cpu+gpu}_{16,256} \approx 15.56$. This enhanced speedup corresponds to a gain of about 97.24% compared to the linear relative speedup $S^{\ell}_{16,256}$.



(a) Strong scaling analysis (b) Weak scaling analysis

FIGURE 2. Strong and weak scaling analyses

Regarding the weak scaling analysis shown in Figure 2(b), the efficiency relative to 16 cores when using 256 CPU cores is $E^{cpu}_{16,256} \approx 80.89\%$. However, when using GPUs, together with CPU cores, we obtain an improved relative efficiency of $E^{cpu+gpu}_{16,256} \approx 98.01\%$.

As we can see, the weak and strong scalability results presented here highlight the contribution of GPUs to achieve much better performance for this framework. They support that the proposed framework scales well up to 256 CPU cores, used together with GPUs, from a hybrid computing system.

6. CONCLUSIONS

We presented a scalable computational framework for the hybrid CPU-GPU implementation of a traffic-induced and finite element-based air quality model. The mathematical model governing the temporal and spatial dynamics of air pollutants has been briefly described. The model discretization, including temporal and spatial integration, was developed. The hybrid CPU-GPU implementation, based on the coupling of the CPU-based distributed-memory programming approach using MPI and a GPU programming model for finite element numerical integration using CUDA, has been exhaustively presented. The numerical simulations were performed on two major road traffic-induced air pollutants, namely the inhalable and

fine particulate matter $PM_{2.5}$ and PM_{10}. The meteorological data, including wind speed and wind direction, required for running simulations were collected from European Centre for Medium-Range Weather Forecasts (ECMWF). The model initial data, including concentration and advective mass fluxes were collected from the Copernicus Atmosphere Monitoring Service (CAMS), implemented by ECMWF. The traffic simulations have been released on major roads subject to heaviest traffic jams during daily rush hours using the microscopic and continuous traffic simulation suite SUMO. A national database provided by the Malian Department responsible for Transport and Infrastructure was used to retrieve SUMO input parameters, such as traffic volume, traffic density, vehicle type, vehicle engine, average speed. The scalability results, including speedup and efficiency, obtained from numerical experiments, have been illustrated and discussed. These achievements have clearly demonstrated that the proposed framework scales well up to 256 CPU cores, used together with GPUs, from a hybrid computing system.

ACKNOWLEDGMENTS

This work is supported by the Malian Government through "Fonds Compétitif pour la Recherche et l'Innovation Technologique (FCRIT)". The authors also would like to acknowledge the high-performance computing support from the "Centre de Calcul, Modélisation et Simulation" (CCMS) of the Faculty of Sciences and Technology (FST) of Bamako.

REFERENCES

[1] C. AGRECO: *Révision du profil environnemental du mali*, Tech. Rep. Number: 2014/342864, European Union, Brussels, Belgium, 2014.

[2] S. BALAY, S. ABHYANKAR, M. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, A. DENER, V. EIJKHOUT, W. GROPP, ET AL.: *Petsc users manual*, tech. rep., Argonne National Laboratory, 2019.

[3] B. CHAPMAN, G. JOST, AND R. VAN DER PAS: *Using OpenMP: portable shared memory parallel programming*, MIT press, 2007.

[4] S. COOK: *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*, Newnes, 2012.

[5] A. ERN, J. GUERMOND: *Theory and Practice of Finite Elements*, vol. 159 in Applied Mathematical Sciences, Springer, 2004.

[6] S. FISHER, D. C. BELLINGER, M. L. CROPPER, P. KUMAR, A. BINAGWAHO, J. B. KOUDENOUKPO, Y. PARK, G. TAGHIAN, P. J. LANDRIGAN: *Air pollution and development in africa: impacts on health, the economy, and human capital*, The Lancet Planetary Health, **5** (2021), e681–e688.

[7] C. GEUZAINE, J.-F. REMACLE: *Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, **79** (2009), 1309–1331.

[8] N. S. GOKHALE: *Practical finite element analysis*, Finite to infinite, 2008.

[9] M. GOWANLOCK, B. KARSIN: *A hybrid cpu/gpu approach for optimizing sorting throughput*, Parallel Computing, **85** (2019), 45–55.

[10] X.-M. HU: *Boundary layer (atmospheric) and air pollution | air pollution meteorology*, in Encyclopedia of Atmospheric Sciences (Second Edition), G. R. North, J. Pyle, F. Zhang, eds., Academic Press, Oxford, 2nd ed., 2015, 227–236.

[11] P. HUTHWAITE: *Accelerated finite element elastodynamic simulations using the gpu*, Journal of Computational Physics, **257** (2014), 687–707.

[12] M. T. JONES, P. E. PLASSMANN: *Scalable iterative solution of sparse linear systems*, Parallel Computing, **20** (1994), 753–773.

[13] K. KARROUM, Y. LIN, Y.-Y. CHIANG, Y. BEN MAISSA, M. EL HAZITI, A. SOKOLOV, H. DELBARRE: *A review of air quality modeling*, MAPAN, **35** (2020), 287–300.

[14] G. KARYPIS, V. KUMAR: *Metis, a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, (1998).

[15] 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** (2010), 7692–7714.

[16] R. LIVESLEY, M. SABIN: *Algorithms for numbering the nodes of finite-element meshes*, Computing Systems in Engineering, **2** (1991), 103–114.

[17] D. L. LOGAN: *A first course in the finite element method*, Cengage Learning, 2016.

[18] P. A. LOPEZ, M. BEHRISCH, L. BIEKER-WALZ, J. ERDMANN, Y.-P. FLÖTTERÖD, R. HILBRICH, L. LÜCKEN, J. RUMMEL, P. WAGNER, E. WIEBNER: *Microscopic traffic simulation using sumo*, in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, Maui, HI, USA, Nov 2018, 2575–2582.

[19] C. J. MURRAY, A. Y. ARAVKIN, P. ZHENG, C. ABBAFATI, K. M. ABBAS, M. ABBASI-KANGEVARI, F. ABD-ALLAH, A. ABDELALIM, M. ABDOLLAHI, I. ABDOLLAHPOUR, ET AL.: *Global burden of 87 risk factors in 204 countries and territories, 1990–2019: a systematic analysis for the global burden of disease study 2019*, The Lancet, **396** (2020), 1223–1249.

[20] W. H. ORGANIZATION: *Outdoor air pollution a leading environmental cause of cancer deaths*, IARC Sci Publ, **161** (2013), 1–177.

[21] W. H. ORGANIZATION: *Ambient air pollution*, sep 2021.

[22] W. H. ORGANIZATION: *WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*, World Health Organization, Geneva, Switzerland, 2021.

[23] J. D. OWENS, M. HOUSTON, D. LUEBKE, S. GREEN, J. E. STONE, J. C. PHILLIPS: *Gpu computing*, Proceedings of the IEEE, **96** (2008), 879–899.

[24] K. RAJU, N. N. CHIPLUNKAR: *A survey on techniques for cooperative cpu-gpu computing*, Sustainable Computing: Informatics and Systems, **19** (2018), 72–85.

[25] J. REN, C. WANG, Y. WANG, R. TIAN: *Scalability tests of a finite element code on hundreds of thousands cores and heterogeneous architecture*, in CCF Conference on High Performance Computing, Springer, 2012, 151–165.

[26] Y. SAAD: *Iterative methods for sparse linear systems*, SIAM, 2003.

[27] A. SAMAKÉ, A. MAHAMANE, M. ALASSANE, O. DIALLO: *A mathematical and numerical framework for traffic-induced air pollution simulation in bamako*, Computation, **10** (2022), 76.

[28] A. SAMAKÉ, A. MAHAMANE, O. DIALLO: *Parallel implementation and scalability results of a local-scale air quality model: Application to bamako urban city*, Journal of Applied Mathematics, 2022.

[29] S. M. H. SEFIDGAR, A. R. FIROOZJAEE, M. DEHESTANI: *Parallelization of torsion finite element code using compressed stiffness matrix algorithm*, Engineering with Computers, **37** (2021), 2439–2455.

[30] M. SNIR, S. OTTO, S. HUSS-LEDERMAN, D. WALKER, J. DONGARRA: *Mpi: The compete reference.*, Computers & Mathematics with Applications, **31** (1996), 140–140.

[31] B. SPORTISSE: *Fundamentals in air pollution: from processes to modelling*, Springer Science & Business Media, Dordrecht, Netherlands, 2009.

[32] J. E. STONE, D. GOHARA, G. SHI: *Opencl: A parallel programming standard for heterogeneous computing systems*, Computing in science & engineering, **12** (2010), 66.

[33] X.-H. SUN, L. M. NI: *Another view on parallel speedup*, in Proceedings of the 1990 ACM/IEEE conference on Supercomputing, 1990, 324–333.

[1,2,3,4] DÉPARTEMENT D'ENSEIGNEMENT ET DE RECHERCHE EN MATHÉMATIQUE ET INFORMATIQUE, FACULTÉ DES SCIENCES ET TECHNIQUES, UNIVERSITÉ DES SCIENCES, DES TECHNIQUES ET DES TECHNOLOGIES DE BAMAKO, BPE 423, BAMAKO, MALI.

Email address: abdoulaye.samake@usttb.edu.ml

Email address: alassanemaiga@yahoo.fr

Email address: moulaye.ahmad@gmail.com

Email address: ouateni@yahoo.fr