

Research Article

Parallel Implementation and Scalability Results of a Local-Scale Air Quality Model: Application to Bamako Urban City

Abdoulaye Samaké , Amadou Mahamane , and Ouaténi Diallo 

Université des Sciences, des Techniques et des Technologies de Bamako, BPE 423, Bamako, Mali

Correspondence should be addressed to Abdoulaye Samaké; abdoulaye.samake@usttb.edu.ml

Received 22 November 2021; Accepted 7 March 2022; Published 26 March 2022

Academic Editor: Fernando Simoes

Copyright © 2022 Abdoulaye Samaké et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a computational framework for the parallel implementation of a local-scale air quality model described by an advection-diffusion-reaction partial differential equation, the so-called equation of reactive dispersion. The temporal discretization of the model is carried out using the forward Euler scheme. The spatial discretization is achieved using the finite element method. The strategy used for the parallel implementation is based on the distributed-memory approach using the message-passing library MPI. The simulations are focused on two road traffic-related air pollutants, namely particulate matters PM_{2.5} and PM₁₀. The efficiency and the scalability of the parallel implementation are illustrated by numerical experiments performed using up to 128 processor cores of a cluster computing system.

1. Introduction

Air pollution is nowadays an increasingly serious global issue for human health and the environment with significant economic consequences. It is particularly exacerbated in the urban context due to the high population density and strong concentration of pollution sources [1]. Air pollutants refer to any substance in the air that could, at high enough concentrations, harm human health and animals and damage the environment. They may consist of solid particles, liquid droplets, gases, or combinations of these forms. Air pollutants are classified into two main types: primary pollutants and secondary pollutants. The primary pollutants are directly emitted into the atmosphere from identifiable sources, which can be either natural, such as wildfires or volcanic eruptions, or anthropogenic such as industrial activities or traffic emissions. They include carbon monoxide (CO), sulfur dioxide (SO₂), nitrogen oxides (NO_x), particulate matter (PM), ammonia (NH₃), volatile organic compounds (VOCs), and toxic metals. As for secondary pollutants, they are not directly emitted but are formed as a result of chemical interactions between primary pollutants within the atmosphere. Secondary pollutants include ozone

(O₃), nitrogen dioxide (NO₂), sulfur trioxide (SO₃), sulfuric acid (H₂SO₄), nitric acid (HNO₃), hydrogen peroxide (H₂O₂), and particulate matter (PM).

Air pollution is associated with a large spectrum of acute and chronic health effects [2] such as stroke, heart disease, chronic obstructive pulmonary disease, lung cancer, acute respiratory infections, skin irritations, eye and ear inflammations, and nose and throat (ENT) disorders. The World Health Organization (WHO) reports that 4.2 million premature deaths worldwide every year are due to exposure to ambient air pollution [3]. The International Agency for Research on Cancer (IARC), the specialized cancer agency of the WHO, has classified outdoor air pollution as carcinogenic to humans [4], Group 1. Faced with the health and environmental issues and the economic cost of air pollution, policies are gradually being implemented for controlling and regulating air quality at the local, regional, and global levels. In more advanced countries, considerable resources are devoted to the measurement and monitoring of ambient air pollutant concentrations at discrete stations of target areas using continually evolving technologies, including air quality sensors. Since these monitors are often expensive and sometimes difficult to access, especially for developing

countries, it is therefore challenging to sufficiently network the target area with them in order to better evaluate the air quality index and make decisions to mitigate its effects.

The modeling and simulation of the motion and dispersion of air pollutants are alternatives that offer good results [5]. These are common tools on which the study and forecasting of air quality are traditionally based. Air pollution modeling is one of the most important and challenging scientific problems [6] often used to support decisions in air quality assessment and management. It covers the transport and diffusion of pollutants in the atmosphere, their dry and wet deposition, and chemical reactions. It also depends on pollutant properties, meteorological conditions, emission data, and terrain parameters.

This work focuses on the modeling and simulation of the road traffic-related air pollution in Bamako, the political and economic capital of the landlocked West African country of Mali (see Figure 1). This city represents 0.02% of the national territory and concentrates 12.46% of the Malian population, with a density of 9062 inhabitants per km² [7]. It is home to more than 70% of the economic activities of Mali, which makes it the main industrial and commercial crossroads of the country. This strong concentration of economic activities, associated with rapid population growth and increasing urbanization, is a major driving force of air pollution in Bamako. In addition, Bamako is geographically built in a basin surrounded by hills which favor the imprisonment of pollutants and make the city even more vulnerable. Despite significant improvements in engine technology, the traffic flow is the main pollution source in urban areas [1], particularly in Bamako where the road infrastructure is very poor and the road transport fleet is obsolete. A large number of diesel vehicles and the poor quality of fuels cause traffic congestion and the emission of numerous pollutants.

These factors mentioned above mean that Bamako, like most African capitals, is a very polluted city where the annual concentration of PM10, an inhalable particulate matter with a diameter of 10 μm or less, can reach 333 μg/m³ [8] with daily peaks exceeding 600 μg/m³ while the air quality guidelines [9] of the WHO recommend a maximum daily mean concentration of 45 μg/m³. Another harmful pollutant widely used as an indicator of air pollution levels in cities is PM2.5, a fine particulate matter with a diameter of 2.5 μm or less. The daily averaged concentrations of PM2.5 over October 2020, obtained from the hourly based PM2.5 data collected by a regulatory grade air monitor installed on the U. S. Embassy compound of Bamako, are plotted in Figure 2. It follows from this figure that these concentrations, with a peak reaching 165 μg/m³, far exceed 15 μg/m³ [10], the maximum value recommended by the air quality guidelines of the WHO as well as the U.S. Environmental Protection Agency (EPA) daily standard level fixed to 35 μg/m³.

In this article, we present a parallel implementation framework and scalability results of a local-scale deterministic and Eulerian-based air quality model in Bamako. We will focus on the parallel implementation aspects and scalability behavior of the computational model used for producing the numerical results presented in [12]. In Section 2, we describe the mathematical model governing the spatiotem-

poral evolution of the concentration of atmospheric pollutants. Section 3 details the temporal and spatial discretization of the model. Section 4 focuses on the parallel implementation and finite element analysis of the model. In Section 5, we present the numerical experiments and computational scalability results of the framework. The main conclusions are summarized in Section 6.

2. Model Description

Let Ω be a bounded computational domain of \mathbb{R}^3 and $\partial\Omega$ the boundary of Ω . We consider a decomposition of $\partial\Omega$ of the form $\partial\Omega = \partial\Omega_G \cup \partial\Omega_H \cup \partial\Omega_L$, where $\partial\Omega_G$ is the ground surface, $\partial\Omega_H$ represents the upper limit boundary of the domain Ω and $\partial\Omega_L$ denotes lateral boundaries. Let c be a vector of concentration fields, where each element c_i corresponds to the scalar field of concentration of the chemical species (pollutant) labelled i in the air. The spatiotemporal evolution of the concentration c_i in the domain Ω over the time interval $(0, T)$ is described by the following advection-diffusion-reaction [13] model:

$$\begin{cases} \frac{\partial c_i}{\partial t} + \nabla \cdot (\mathbf{u}c_i) - \nabla \cdot (\mu \nabla c_i) - \chi_i(c) + \Lambda_i(\mathbf{x}, t)c_i = S_i(\mathbf{x}, t) & \text{in } \Omega \times (0, T) \\ c_i(x, 0) = 0 & \text{in } \Omega \\ \mu \nabla c_i \cdot \mathbf{n} = v_i^{dep} c_i - E_i & \text{on } \partial\Omega_G \times (0, T) \\ \mu \nabla c_i \cdot \mathbf{n} = 0 & \text{on } \partial\Omega_H \times (0, T) \\ c_i \mathbf{u} \cdot \mathbf{n} = 0 & \text{on } \partial\Omega_L \times (0, T) \end{cases} \quad (1)$$

where (i) $c_i = c_i(\mathbf{x}, t)$ is the concentration of pollutant labelled i , (ii) $\mu(\mathbf{x}, t)$ is the diffusion coefficient, (iii) $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the wind velocity (or wind speed), (iv) $\chi_i = \chi_i(\mathbf{x}, t)$ is the chemical source term that describes the chemical reactions, (v) $\Lambda_i = \Lambda_i(\mathbf{x}, t)$ is the scavenging coefficient, (vi) $S_i = S_i(\mathbf{x}, t)$ is the source terms, (vii) $E_i = E_i(\mathbf{x}, t)$ is the surface emissions from road traffic, (viii) $v_i^{dep} = v_i^{dep}(\mathbf{x}, t)$ is the dry deposition velocity, (ix) $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the unit outward normal vector to the boundary $\partial\Omega$, (x) $\mathbf{x} = (x, y, z)$ is the spatial coordinates, where x and y are the variables of the plane surface and z is the altitude, and (xi) t is the physical time. The units in the international system (SI) of these physical fields and parameters are reported in Table 1. The diffusion coefficient parametrization, the dry deposition, scavenging processes, and the chemical kinetics are described in detail in [12]. Model (1) is based on the assumption that there is no feedback between the chemical species and the flow fields, including wind velocity and diffusion coefficient. We assume that the flow is incompressible. This means that the variations in air density are assumed to be constant. We also assume that the urban terrain is homogeneous. The meteorological parameters are strongly influenced by the atmosphere around them. The transport of pollutants takes place mainly in the atmospheric boundary layer (ABL) [14], the lowest part of the troposphere subject to surface effects, since most of the pollution sources are located at the ground level. The ABL is of interest in urban air

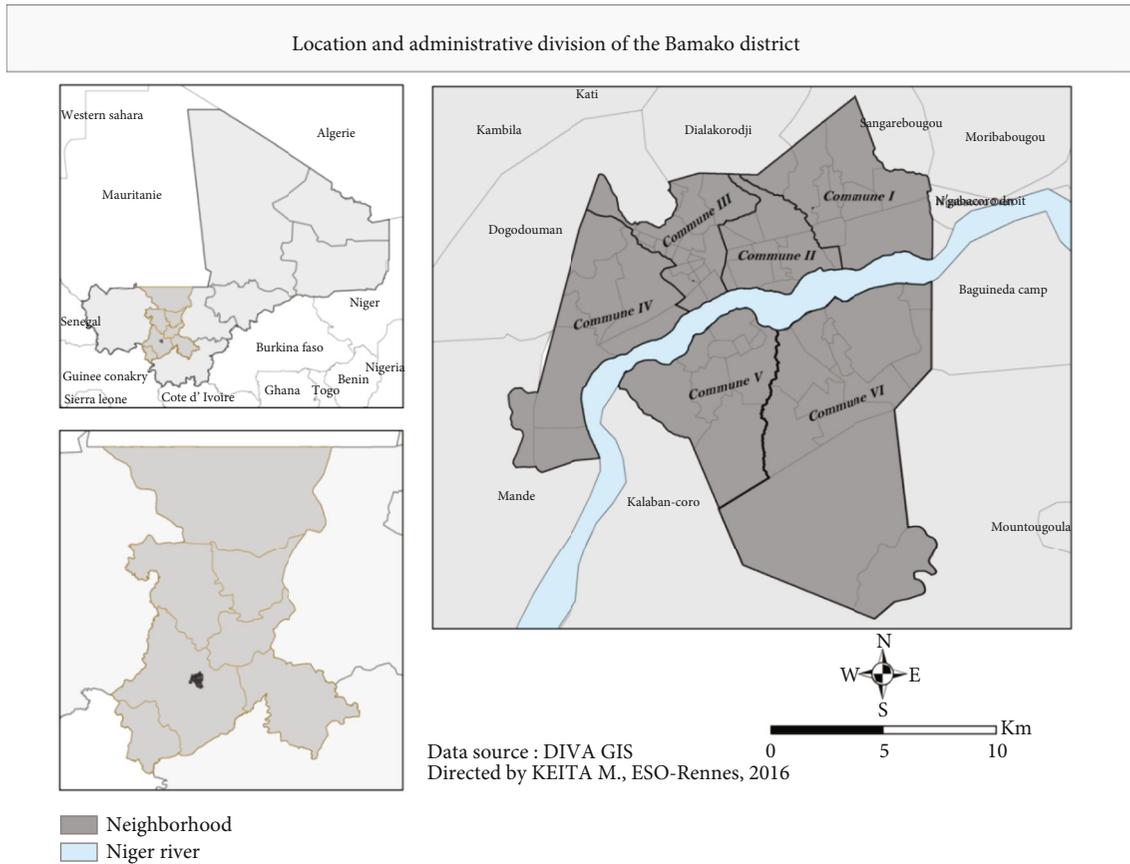


FIGURE 1: Location of the city of Bamako and its administrative division. Source: [11].

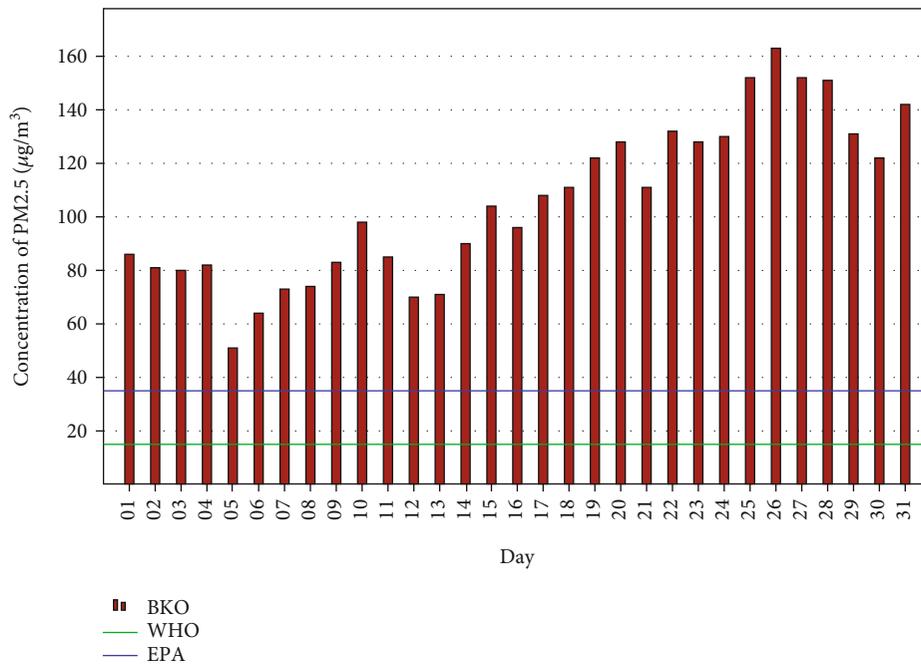


FIGURE 2: Daily averaged concentration of PM_{2.5} in Bamako over October 2020. Source: based on EPA (Bamako Air Quality Monitor - US Embassy) and the World Air Quality Index Project database 2020.

TABLE 1: Fields and physical parameters of the model.

Symbol	Name	Unit
c	Concentration	kg m^{-3}
\mathbf{u}	Wind velocity	m s^{-1}
μ	Diffusion coefficient	$\text{m}^2 \text{s}^{-1}$
v^{dep}	Deposition velocity	m s^{-1}
Λ	Scavenging coefficient	s^{-1}
χ	Chemical sources	$\text{kg m}^{-3} \text{s}^{-1}$
S	Volume emissions	$\text{kg m}^{-3} \text{s}^{-1}$
E	Surface emissions	$\text{kg m}^{-2} \text{s}^{-1}$
H	Maximum altitude	m
κ	Reaction rate	s^{-1}
τ	Characteristic time	s

pollution modeling as it is the atmospheric layer where humans dwell [15]. We will focus on an altitude, which we denote H , included in the ABL layer.

In the rest of this article, we will consider first-order chemical reaction kinetics with a constant coefficient. This means that the balance of the physicochemical transformations in model (1) can be written as $\chi_i = -\kappa_i c_i$, where κ_i is the reaction rate. The characteristic time [16] of the chemical species labelled i , which we denote τ_i , is defined according to the reaction rate by the following relation:

$$\tau_i = \frac{1}{\kappa_i}. \quad (2)$$

This characteristic timescale, also called residence time, represents the average time spent by a representative molecule in the atmosphere before it is removed during loss processes such as dry deposition, wet scavenging, and chemical reactions. The characteristic distance (space scale) over which species can be transported is correlated with the residence time. If the emitted species have short residence time (minutes-hours) in the atmosphere, they cannot be transported long-range, and their effects will therefore be important on a local scale. However, other chemical species with longer residence time (hours-days) can have wider impact zones where regional or continental scales will be needed. The characteristic timescale τ_i associated with the dry deposition in an atmospheric layer of height H for a species labelled i can be computed from its dry deposition velocity as follows:

$$\tau_i = \frac{H}{v_i^{dep}}. \quad (3)$$

3. Model Discretization

The time integration of model (1) is carried out using the forward Euler method. The choice of this classical approach is dictated by the temporal scale of interest, about 10 minutes, and the physical and chemical properties of air pollutants in which we are interested. Taking into account the

assumptions listed in Section 2, model (1) is discretized in time as follows:

$$\frac{c_i^n - c_i^{n-1}}{\Delta t} + \mathbf{u} \cdot \nabla c_i^n - \nabla \cdot (\mu \nabla c_i^n) - \kappa_i c_i^n = S_i(\mathbf{x}, t_n), \quad (4)$$

where $c_i^n = c_i(\mathbf{x}, t_n)$ and $t_n = n\Delta t$, $n \in \mathbb{N}^*$. Δt denotes the time-step of the discretization.

The spatial discretization of the model is achieved using the finite element method [17, 18]. It is considered one of the best-suited methods for computing the approximate solution of complex problems in science and engineering expressed in terms of partial differential equations (PDEs) on regions with complex geometrical configurations. This method is based on the principle of energy minimization and consists in approximating, in a finite-dimensional subspace, a problem written in a variational form in an infinite-dimensional space. The approximate solution is finally computed as a collection of discrete values in the form of components of a solution vector of the linear system resulting from the discretization. Readers who are already familiar with the classical finite element method can directly jump to Section 4.

The main steps involved in the finite element analysis, already recapped in [19], are (i) the problem definition, (ii) the discretization (triangulation and definition of approximation space), (iii) the definition of the variational (or weak) formulation, (iv) the system assembling (local (on each mesh element) and global (on the entire mesh)), and (v) the solution of the resulting linear system.

Let \mathcal{T}_δ be a mesh of Ω , where δ refers to the mesh characteristic length. We denote \mathcal{V}_δ the piecewise linear finite element space defined on \mathcal{T}_δ with a vanishing value at the closed boundary $\partial\Omega_L$. It is defined as

$$\mathcal{V}_\delta = \{v \in C^0(\bar{\Omega}) \mid v \text{ is linear on } K, K \in \mathcal{T}_\delta\} \cap H_{0,\partial\Omega_L}^1, \quad (5)$$

where $H_{0,\partial\Omega_L}^1$ is the Hilbert-Sobolev space of order 1 satisfying a homogeneous Dirichlet condition on the closed boundary $\partial\Omega_L$. $C^0(\bar{\Omega})$ is the set of continuous functions defined on $\bar{\Omega}$, the closure of Ω . The variational form of the problem (1) is written as follows: find $c_i \in \mathcal{V}_\delta$ such that

$$\begin{aligned} \int_{\Omega} (c_i^n + \Delta t \mathbf{u} \cdot \nabla c_i^n - \Delta t \nabla \cdot (\mu \nabla c_i^n) - \Delta t \kappa_i c_i^n) v \\ = \int_{\Omega} (\Delta t S_i(\mathbf{x}, t) + c_i^{n-1}) v \forall v \in \mathcal{V}_\delta, \end{aligned} \quad (6)$$

where c_i refers to c_i^n . Integrating Equation (6) by parts and taking into account the boundary conditions defined in problem (1), the following weak form holds: find $c_i \in \mathcal{V}_\delta$ such that

$$\begin{aligned} \int_{\Omega} \mu \Delta t \nabla c_i \cdot \nabla v + (c_i^n + \Delta t \mathbf{u} \cdot \nabla c_i^n - \Delta t \kappa_i c_i^n) v - \int_{\partial\Omega_G} \Delta t v_i^{dep} c_i v \\ = R(v) \forall v \in \mathcal{V}_\delta, \end{aligned} \quad (7)$$

where

$$R(v) = \int_{\Omega} (\Delta t S_i + c_i^{n-1})v - \int_{\partial\Omega_G} \Delta t E_i v. \quad (8)$$

The Lax-Milgram theorem [17] guarantees the existence and uniqueness of the solution of the weak problem (7), which can also be written as follows:

$$Q(c_i, v) = R(v). \quad (9)$$

Let $\mathcal{B} = \{\phi_j\}_{j=1}^{N_\delta}$ be a set of basis functions of V_δ . The approximate solution $c_i \in V_\delta$ can be expressed as

$$c_i = \sum_{j=1}^{N_\delta} c_{i,j} \phi_j, \quad (10)$$

where $c_{i,j}, j = 1, \dots, N_\delta$ are nodal values. By using Equations (9) and (10), the following discrete form of the problem holds:

$$\sum_{j=1}^{N_\delta} c_{i,j} Q(\phi_j, \phi_k) = R(\phi_k), \forall k = 1, \dots, N_\delta, \quad (11)$$

where $A \in \mathbb{R}^{N_\delta \times N_\delta}$ and $\mathbf{f}, \mathbf{c} \in \mathbb{R}^{N_\delta}$. The entries of the matrix A are computed as $a_{jk} = Q(\phi_k, \phi_j)$. The components of the right-hand-side vector \mathbf{f} are $f_j = R(\phi_j)$, and the vector solution \mathbf{c} is composed of the unknown coefficients $c_{i,j}$. The finite element method presented here refers to the so-called Galerkin approximation [20] since the trial space, where the solution c_i belongs, is the same as the space of test functions. The linear system (11) resulting from this discretization can be solved using either direct solvers like LU factorization [21] or iterative methods like generalized minimal residual [22]. It is often essential to use preconditioning techniques [22], including domain decomposition methods [23], for solving the linear system (11) since it is in general ill-conditioned.

4. Parallel Implementation

Libraries for finite element solutions of problems arising from partial differential equations are commonly used by mathematicians and engineers. These packages provide in general robust and powerful tools for the implementation of advanced numerical methods and support high-performance computing features. There are many open-source and popular libraries that offer these capabilities in the literature: Feel++ [24], FreeFem++ [25], FEniCS Project [26], GetFEM [27], and deal.II [28]. For this project, we develop a new robust framework instead of using existing packages because these are mostly very generic and therefore do not sufficiently support some specific features related to this work. It is therefore an opportunity to make this framework evolve in an independent project using flexible approaches that lead to the intended goals.

Tremendous progress in the design of parallel computers over the last decades and the recent advances in multicore processors and hardware accelerator technologies have opened a new era in high-performance computing for the solution of highly demanding complex problems in simulation-based applied science and engineering. We aim to reproduce the dynamical behaviors of some atmospheric pollutants by solving the air quality model (1) in a computational domain covering the entire region of Bamako (with an area of about 267 km²) using a sufficiently fine spatial resolution. Performing such computations in practice within a reasonable time requires the recourse to parallel computing [29, 30].

In this context, we develop a C++ framework, called AirQMB, using modern C++17 programming features. The structure of the main core of this model code and the parallelization strategy were inspired by the parallel computational framework introduced in [19]. We consider the CPU-based distributed-memory programming approach using the Message Passing Interface (MPI) [31]. The communications between processor cores are handled by using Boost.MPI, a useful Boost library [32] that abstracts layers over the standard MPI for simplifying the user interface.

For the parallel finite element analysis, the first essential step we are interested in is mesh processing. The computational mesh, based on a map extracted from OpenStreetMap [33], is generated in preprocessing over the region of interest by using an open-source mesh generator called Gmsh [34]. The software package METIS [35] is used for partitioning the generated serial mesh so that the number of mesh partitions corresponds to the number of processor cores that will be deployed for numerical simulations. This package is called from the Gmsh interface. The parallel (partitioned) finite element mesh, already built, is loaded independently and concurrently by all processor cores. Each processor core extracts locally only the partition that number corresponds to its local rank plus the interprocess (ghost) elements so that all elements needed for the finite element basis functions are locally available. Therefore, interprocess communications are not required for the parallel finite element assembly.

A crucial step that follows mesh processing refers to finite element discretization. It consists of constructing the set (or table) of degrees of freedom (DOFs) and is achieved independently on the local mesh within each processor core. A local numbering of DOFs is then introduced independently within each processor using a minimum-bandwidth numbering procedure. This local numbering will be connected with the global numbering associated with the entire problem through relationships called local-to-global maps (LtGMs). The construction of these maps requires global communication involving all the deployed processors. The parallel global assembly from the elementary matrices built locally on processor cores is achieved using LtGMs. The serial and parallel algebraic operations involving matrices and vectors, including direct and iterative solvers for the solution of linear systems, are handled by PETSc [36] wrappers.

The input data required for air quality simulations, especially the wind fields and traffic emissions, are uploaded

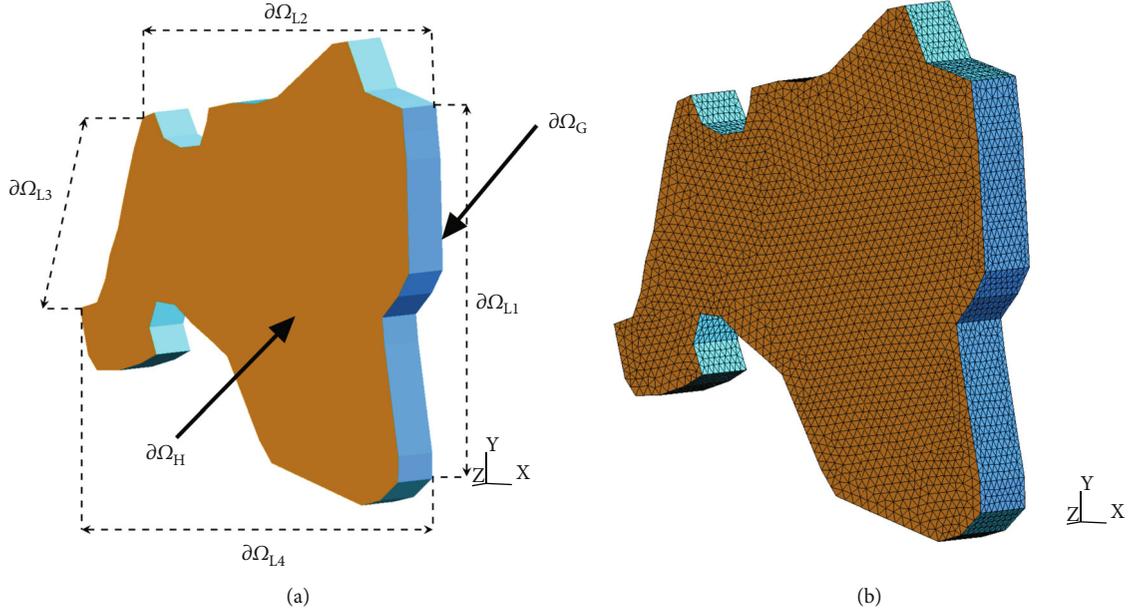


FIGURE 3: Three-dimensional domain and mesh of the urban city of Bamako. (a) The three-dimensional domain of Bamako. (b) Three-dimensional computational mesh.

from files locally in each processor core into the computational framework using the NetCDF [37] library, an interface that supports the creation, access, and sharing of array-oriented scientific data. Once imported, these data are interpolated on the computational mesh using robust and efficient interpolation routines. The simulation environment is initialized by using a useful library called Boost Program Options. This tool makes it easy to parse command-line options for console applications.

5. Numerical Experiments

The numerical simulations presented here are focused on two primary air pollutants commonly used in the survey of air quality. The first one is the fine particulate matter (PM2.5) with an aerodynamic diameter less than or equal to $2.5 \mu\text{m}$ and the second refers to the inhalable particulate matter (PM10) with an aerodynamic diameter less than or equal to $10 \mu\text{m}$. These pollutants, composed of a mixture of solid and liquid particles suspended in the air, are an essential air quality indicator since they are among the most widely distributed traffic-related air pollutants that affect short-term and long-term health.

We assume that the losses caused by deposit and scavenging are negligible, which means that the coefficient $\Lambda = 0$. The maximum altitude is fixed to $H = 10\text{m}$. The physicochemical parameters corresponding to PM2.5 are $\mu = 6.3610^{-6} \text{m}^2 \text{s}^{-1}$, $\kappa = 3.1710^{-6} \text{s}^{-1}$, and $v_i^{dep} = H\kappa = 3.1710^{-5} \text{ms}^{-1}$. The physicochemical parameters for PM10 are given by $\mu = 1.5910^{-6} \text{m}^2 \text{s}^{-1}$, $\kappa = 2.7310^{-5} \text{s}^{-1}$, and $v_i^{dep} = H\kappa = 2.7310^{-4} \text{ms}^{-1}$. The computational mesh covering the city of Bamako with a mean spatial resolution of about 10 m is presented in Figure 3. The number of mesh elements (tetra-

hedra) is approximately equal to 1.7×10^6 . The number of mesh nodes is about 31×10^3 . The physical time-step Δt is fixed to 60 seconds, and the duration of the simulation is set to 24 hours. The first-order finite element approximation is used for the spatial discretization, and the wind data are obtained from the European Centre for Medium-Range Weather Forecasts (ECMWF). The road traffic emission data files used as inputs of the model were generated in pre-processing from traffic simulations performed on the road network presented in Figure 4. In this network, we are interested in the main and most frequented roads that are subject to massive traffic congestions during daily peak hours. The traffic simulations were performed using an open-source road traffic simulation suite called SUMO [38]. The parameters needed as SUMO inputs, such as traffic flow, traffic density, vehicle types, vehicle engine, and mean velocity, are taken from the national databases provided by the Malian department in charge of transport and infrastructure.

The simulations have been performed on the “*Centre de Calcul, Modélisation et Simulation*” (CCMS) of the Faculty of Science and Technology (FST) of Bamako. CCMS is a cluster with ten compute nodes (servers) interconnected by an InfiniBand QDR network. These nodes are divided into two groups of two different machine types. In the first group, composed of five Dell PowerEdge servers, each node has two Intel Xeon Silver 4110 processors with 8 cores (2 threads per core) running at 2.10 GHz and 64 GB of RAM. In the second group, composed of five HPE ProLiant servers, each node has two Intel Xeon E5-2623 v4 processors with 4 cores (2 threads per core) cadenced at 2.60 GHz and 16 GB of RAM. The operating system is Ubuntu Server.

In the context of parallel computing, the computational scalability (or scaling) is a powerful tool widely used for expressing the ability of a given parallel algorithm to best

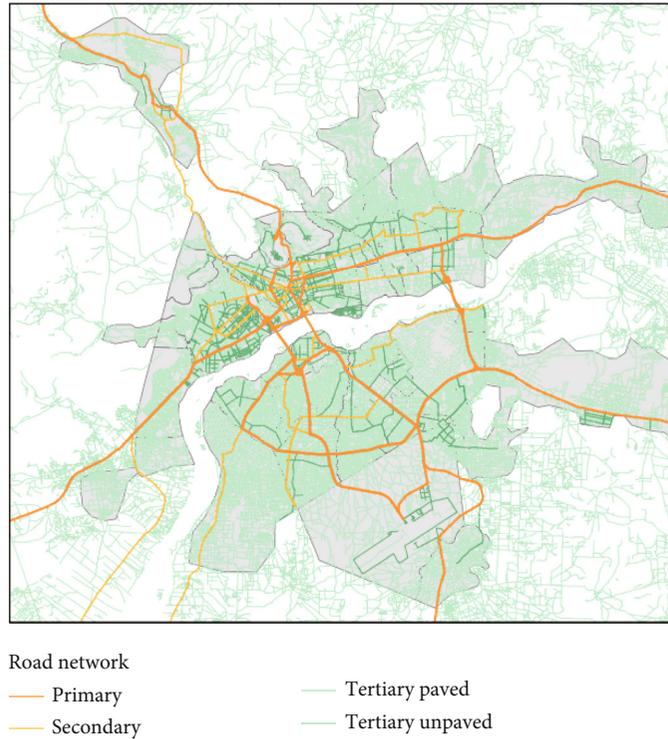


FIGURE 4: Road network of the city of Bamako. Source: [39].

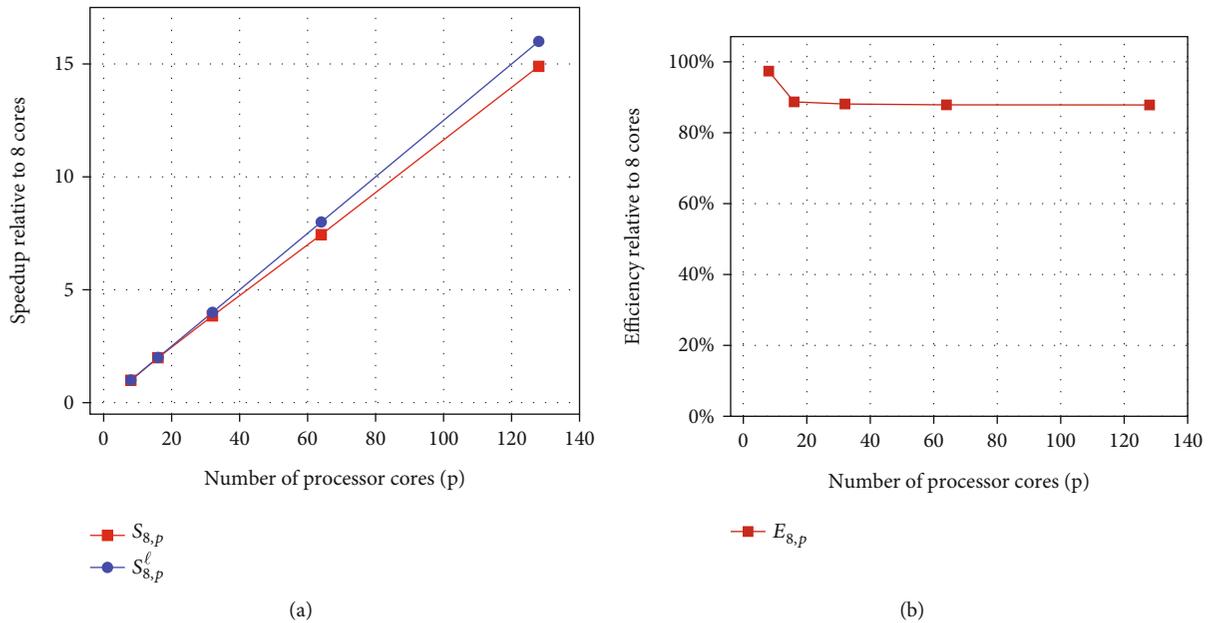


FIGURE 5: Strong and weak scaling analyses. (a) Strong scaling analysis. (b) Weak scaling analysis.

exploit a parallel computing platform. It is related to two common metrics, namely the strong scaling (or speedup) and weak scaling (or efficiency). The speedup is defined as the time spent to run a problem on one processor core divided by the time it takes to run the same problem (fixed size) on p ($p > 1$) processor cores. In other words, let T_1 be the execution time of a problem on a single processor core and T_p be the parallel execution time of the same problem

on p ($p > 1$) processor cores. The speedup with p processor cores is defined as $S_p = T_1/T_p$. The efficiency with p processor cores is computed as $E_p = S_p/p$. The best possible efficiency is $E_p = 1$. It is reached when the speedup is linear, i. e., $S_p = p$. A generalization of these metrics for large-scale computing has led to the introduction of the relative speedup and relative efficiency. Let T_r be the parallel

execution time of a problem on r processor cores and T_p be the parallel execution time of the same problem on p ($r < p$) processor cores. The speedup relative to r processor cores when using p cores is $S_{r,p} = T_r/T_p$. The best possible relative speedup is the linear speedup computed as $S_{r,p}^e = p/r$. The efficiency relative to r processor cores when using p cores is defined as $E_{r,p} = rS_{r,p}/p$.

The strong and weak scaling results for this parallel computational framework when the number of processor cores varies from 8 to 128 are plotted in Figure 5.

Concerning the strong scaling analysis shown in Figure 5 (a), the speedup relative to 8 cores when using 128 processor cores is $S_{8,128} \approx 15.02$ while the linear relative speedup, i.e., the best possible value, is $S_{8,128}^e = 16$. This result corresponds to a performance gain of about 93.88%. According to the weak scaling analysis presented in Figure 5(b), the efficiency relative to 8 processor cores when using 128 processor cores is $E_{8,128} \approx 90.20\%$. As we can see, these weak and strong scaling results clearly show that this framework scales well up to 128 processor cores of a cluster computing architecture.

6. Conclusion

We have presented a computational framework for the parallel implementation of a local-scale air quality model and its application to the urban part of Bamako city. We briefly described the advection-diffusion-reaction partial differential equation that governs the spatiotemporal evolution of the concentrations of air pollutants. The numerical methods used for the time integration and the spatial discretization of the model have been presented. Particular emphasis has been placed on the parallel implementation based on the distributed-memory approach using Message Passing Interface (MPI) and modern C++17 programming features. The simulations were focused on two widely distributed road traffic-related air pollutants, namely particulate matters PM_{2.5} and PM₁₀.

The scalability analyses presented here showed good speedup and efficiency up to 128 processor cores. These results are promising and allow us to reproduce state-of-the-art dynamical behaviors of the atmospheric pollutants specified.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Rectorate of the University of Sciences, Techniques and Technologies of Bamako (USTTB) through project funding. The authors also wish to thank the "Centre de Calcul, Modélisation et Simulation" (CCMS) of

the Faculty of Sciences and Technology (FST) of Bamako for having supported this work by providing scientific support and access to computing resources on its cluster.

References

- [1] F. Murena, A. Di Benedetto, M. D'Onofrio, and G. Vitiello, "Mass transfer velocity and momentum vertical exchange in simulated deep street canyons," *Boundary-Layer Meteorology*, vol. 140, no. 1, pp. 125–142, 2011.
- [2] B. Brunekreef and S. T. Holgate, "Air pollution and health," *The Lancet*, vol. 360, no. 9341, pp. 1233–1242, 2002.
- [3] World Health Organization, "Air pollution," 2020, <http://www.who.int/phe>.
- [4] International Agency for Research on Cancer, "Outdoor air pollution a leading environmental cause of cancer deaths," 2013, <http://www.who.int/phe>.
- [5] S. Kumar and R. Kumar, *Air quality: monitoring and modeling*, BoD–Books on Demand, 2012.
- [6] Á. Leellosy, F. Molnár, F. Izsák, Á. Havasi, I. Lagzi, and R. Mészáros, "Dispersion modeling of air pollutants in the atmosphere: a review," *Open Geosciences*, vol. 6, no. 3, pp. 257–278, 2014.
- [7] INSTAT-ML, "Annuaire statistique du mali," Tech. Rep., Institut National de la Statistique (Mali), Bamako, Mali, 2018, <http://www.instat-mali.org>.
- [8] B. Mondiale, *Profil environnemental des villes de Bamako, Gao, Mopti et Sikasso*, 2011, Numéro du Rapport: 60788-ML.
- [9] World Health Organization, "Ambient air pollution: a global assessment of exposure and burden of disease," Tech. Rep., 2016, <http://www.who.int/phe>.
- [10] World Health Organization, *WHO global air quality guidelines: particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*, World Health Organization, 2021.
- [11] K. Mamadou, *Typologie urbaine et accessibilité géographique potentielle des établissements de santé dits modernes dans le district de Bamako (Mali)*, Espace populations sociétés, 2018, <https://journals.openedition.org/eps/7707>.
- [12] A. Samaké, A. Mahamane, M. Alassane, and O. Diallo, "Advances in the implementation of a mathematical framework for air quality simulation in Bamako," 2021, Submitted.
- [13] B. Sportisse, *Fundamentals in Air Pollution: From Processes to Modelling*, Springer Science & Business Media, 2009.
- [14] R. B. Stull, *An Introduction to Boundary Layer Meteorology*, Atmospheric and Oceanographic Sciences Library. Springer Netherlands, 1988, <https://books.google.ml/books?id=eRRz9RNvNokC>.
- [15] D. A. Vallero, *Fundamentals of air pollution*, Academic press, 2014.
- [16] M. K. Neophytou, D. A. Goussis, M. Van Loon, and E. Mastorakos, "Reduced chemical mechanisms for atmospheric pollution using computational singular perturbation analysis," *Atmospheric Environment*, vol. 38, no. 22, pp. 3661–3673, 2004.
- [17] A. Ern and J. L. Guermond, "Theory and practice of finite elements," vol. 159 of Applied Mathematical Sciences, Springer, 2004, <http://books.google.fr/books?id=CCjm79FbJbcC>.
- [18] D. L. Logan, *A First Course in the Finite Element Method*, Cengage Learning, 2016.

- [19] A. Samaké, P. Rampal, S. Bouillon, and E. Ólason, “Parallel implementation of a Lagrangian-based model on an adaptive mesh in C++: application to sea-ice,” *Journal of Computational Physics*, vol. 350, pp. 84–96, 2017.
- [20] A. Quarteroni and S. Quarteroni, *Numerical Models for Differential Problems*, vol. 2, Springer, 2009.
- [21] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, 2006.
- [22] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
- [23] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, SIAM, 2015.
- [24] C. Prud’homme, V. Chabannes, V. Doyeux, M. Ismail, A. Samaké, and G. Pena, “Feel++: a computational framework for Galerkin methods and advanced numerical methods,” in *ESAIM: Proceedings*, vol. 38, pp. 429–455, Marseille, France, 2012.
- [25] F. Hecht, “New development in freefem++,” *Journal of Numerical Mathematics*, vol. 20, no. 3-4, pp. 251–265, 2012.
- [26] M. Alnæs, J. Blechta, J. Hake et al., “The fenics project version 1.5,” *Archive of Numerical Software*, vol. 3, p. 100, 2015.
- [27] Y. Renard and K. Poullos, “Getfem: automated fe modeling of multiphysics problems based on a generic weak form language,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 47, no. 1, pp. 1–31, 2021.
- [28] D. Arndt, W. Bangerth, B. Blais et al., “The deal. ii library, version 9.3,” *Journal of Numerical Mathematics*, vol. 29, no. 3, pp. 171–186, 2021.
- [29] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, vol. 400, Benjamin/Cummings Redwood City, CA, 1994.
- [30] G. H. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing*, Elsevier, 2014.
- [31] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, “Mpi: the compete reference,” *Computers & Mathematics with Applications*, vol. 31, no. 11, p. 140, 1996.
- [32] S. Koranne, “Boost C++ libraries,” in *Handbook of Open Source Tools*, pp. 127–143, Springer, 2011.
- [33] J. Bennett, *OpenStreetMap*, Packt Publishing Ltd, 2010.
- [34] C. Geuzaine and 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*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [35] G. Karypis and V. Kumar, *Metis, a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, vol. 38, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, 1998.
- [36] S. Balay, K. Buschelman, V. Eijkhout et al., “PETSc users manual,” Tech. Rep. ANL-95/11 revision 2.1.5, Argonne national The Laboratory, 2004.
- [37] R. Rew and G. Davis, “Netcdf: an interface for scientific data access,” *IEEE Computer Graphics and Applications*, vol. 10, no. 4, pp. 76–82, 1990.
- [38] P. A. Lopez, M. Behrisch, L. Bieker-Walz et al., “Microscopic traffic simulation using sumo,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, Maui, HI, USA, Nov 2018.
- [39] M. Mukim, “Bamako urban sector review: an engine of growth and service delivery,” Tech. Rep. The World Bank, 2018.