Imperial College London

Department of Earth Science and Engineering

MSc in Environmental Data Science and Machine Learning

Independent Research Project
Project Plan

Optimising sensor location using neural networks applied to air pollution

by

Xuan Zhan

Email: xuan.zhan23@imperial.ac.uk
GitHub username: edsml-xz3323
Repository: https://github.com/ese-msc-2023/irp-xz3323

Supervisors:

Prof Christopher Pain
Dr Claire Heaney

June 14, 2024

Abstract

This study plans to use a combination of the AI4PDEs and AI4Particles methods to analyze airborne pollutant flows, aiming to develop a methodology for linking sensor data to pollution sources. The AI4PDEs method will be used to compute airflow processes using fixed-weight neural networks to solve discretized governing equations, which can save significant computational resources compared to traditional PDE-solving methods. Airborne pollutants are treated as particles in a fluid, and the AI4Particles method will be used to track the trajectories of these particles. This method combines another neural network with fixed-weights and a 3D Lagrangian particle tracking method. Compared to traditional sensitivity analysis methods, it not only saves computational resources but also provides greater interpretability, scalability, and flexibility and is not affected by the way the objective function is defined. The placement of sensors, in terms of time and location, depends on changes in pollutant particle trajectories.

Keywords: AI4PDEs; AI4Particles; Discretised Governing Equations; Lagrangian Particle Tracking

1 Introduction

1.1 Problem Description

Air pollution increases people's risk of illness, especially respiratory diseases and reactions to allergens. Sensors, however, can capture airborne pollutant concentrations and are useful for tracking the source of air pollution. This study links changes in air pollutants to the location and time of day when sensors are placed. This allows detection resources to be used most efficiently to obtain the best possible prediction of the population's exposure to pollution.

1.2 Significance

Research into more efficient adaptive sensor methods can reduce the consumption of computational resources and obtain more intuitive and accurate results, and there is hope that urban air pollution can be effectively controlled. This holds the promise of better air measurement solutions at a low cost. Additionally, by treating air pollutants as a large number of particles, model interpretability is enhanced, making it easier to understand the rationale for placing sensors.

1.3 Review of Existing Work

In 2016, Doctor Fang published an article on designing sensor locations using a sensitivity analysis approach [1], which discusses reducing computational costs by combining Proper Orthogonal Decomposition (POD) and reduced-order modeling [2, 3, 4]. It explores the sensitivity of this approach to different times and locations by setting up a functional objective function [5, 6]. Sensors are placed in areas with high sensitivity. However, this method requires a highly specific objective function, leading to high implementation difficulty and low generalization ability. Subsequently, researchers began to explore the application of machine learning in fluid dynamics [7]. Using fixed-weight neural networks [8], methods for solving PDEs began to appear [9, 10, 11], employing GPUs to accelerate computation [12, 13, 14, 15], which alleviates the problem of computational resources to a certain extent. Currently, this approach is used in models of multiphase flows [10] and DEM models [9]. In the development of Lagrangian particle tracking technology [16], several applications of 3D Lagrangian particle tracking technology in fluid dynamics have emerged [11, 17, 18]. It is anticipated that a more advanced particle tracking technique will be developed by integrating neural networks based on this foundation [19, 20, 21].

1.4 Objectives

1. Combining neural networks and 3D Lagrangian particle tracking methods to implement the AI4Particles method.
2. Combine the AI4Particles and AI4PDEs methods so that the information of air pollutant particles such as position and velocity can be visualized.
3. Design sensor placement programmes based on pollutant change.
4. The model will be tested on real datasets and the results are compared with other methods.

2 Methodology

First, the airflow is simulated using the AI4PDEs method. Then, the AI4DEM technology is used for reference, combined with neural network and 3D Lagrange particle tracking technology to improve the computational efficiency. The ultimate goal is to capture the trajectory and velocity of air pollutant particles.

2.1 Data

The study requires gridded data for the target area, and the current data used for South Kensington is provided by Imperial College London.

2.2 AI4PDEs

This approach solves the discretised governing equations using convolutional neural networks with fixed weights. The weights of each convolutional layer are set to correspond to the coefficients in the discretisation scheme by means of specific discretisation methods, e.g. finite element, second order splitting. It is important for solving processes involving large-scale computation and complex physical reactions.

2.3 Lagrangian particle tracking

This technique is widely used in hydrodynamics to track the trajectory of particles, especially in highly turbulent scenarios, and is capable of continuously tracking the position, velocity, and acceleration of millions of particles.
Implements steps:

1. Acquire particle images: multiple cameras are used to capture images of particles in the system from different angles, and these images are used to analyse particle positions.
2. Find position and velocity: Determine the 3D position and velocity of the particles through advanced iterative particle reconstruction techniques (e.g., Shake-The-Box algorithm).

3. Data assimilation: Data assimilation is performed on the particle data in conjunction with Navier-Stokes equation constraints, while complementing the particle trajectories and velocities.

2.4 AI4DEM

Solve the computational challenges in DEM methods through combining with fixed-weight neural network.

2.4.1 Define the neural network weights

Each convolutional kernel of a neural network depends on the coefficients of physical equations that should be mathematically discretised (e.g. contact force model, Newton's second law).

2.4.2 Traditional Discrete Element Method (DEM)

The DEM simulation stores and continuously updates information about each particle's position, velocity, and contact force with neighbouring particles. Among other things, the translational and rotational motions of the particles are controlled by the following equations:

$$\dot{\vec{v}} = \frac{\vec{F}}{m} + \vec{g}$$
$$\dot{\vec{\omega}} = \frac{\vec{T}}{I}$$

In this model, $\dot{\vec{v}}$ and $\dot{\vec{\omega}}$ represent the particle's linear velocity and angular velocity vectors, respectively. $\vec{F}$ and $\vec{T}$ denote the total force and torque acting on the particle, while m and I are the particle's mass and moment of inertia, respectively. $\vec{g}$ represents the vector of gravitational acceleration.

The contact force is modeled using a spring and viscous damping model. When two particles come into contact, the force exerted on particle i by particle j can be decomposed into a normal force and a tangential force as follows:

$$\vec{f}_{n,j\to i} = \begin{cases} -k\delta_{ij}\vec{n}_{ij} - \eta\vec{v}_{ij,n} & \text{if } \delta_{ij} < (r_i + r_j) \\ 0 & \text{otherwise} \end{cases}$$

Here, δ_{ij} represents the overlap between the particles, k is the stiffness of the spring, η is the coefficient of viscous dissipation, $\vec{n}_{ij}$ is the unit normal vector from particle i to particle j, and $\vec{v}_{ij,n}$ is the relative normal velocity.

2.5 Alternative Approaches

Currently used methods rely on directly established weights, which means that the network may need to be redesigned for different environments. However, designing these weights still requires deep expertise, potentially limiting their popularity among users without relevant professional backgrounds. On the other hand, since the weights of the neural network are determined directly, the cause of the error is difficult to find. Consideration could be given to combining AI4PDEs methods with sensitivity-based methods or improving sensitivity-based methods, which make error analysis easier despite reduced model interpretability.

3 Expected Outcomes

3.1 Outcome files

1. **Research Plan:** Includes research methodology, expected results, and contributions to related fields.
2. **Completed Code:** Utilizes the new method to determine the optimal placement time and location for a specified number of sensors.
3. **Practical Validation Results:** A sensor placement scheme for the South Kensington area of London or somewhere else.
4. **Final Report:** Summarizes the research process, including details of the research methodology, conclusions, and contributions to the field and society.
5. **Slides:** For presentation, illustrating the research process and results.

3.2 Limitations

1. In complex urban environments, changes in weather conditions, car travel rates, and other variables can significantly impact air pollution levels. The use of AI for partial differential equations (AI4PDEs) and AI for particles (AI4Particles) to simulate the flow of air pollutants may not adequately account for these complex factors.
2. The computation and training of neural networks (especially in AI4Particles) consumes a lot of computational resources, especially in large areas or areas with complex terrain.
3. The model parameters or structure need to be adjusted for special areas of the terrain, especially in the treatment of the boundary.

4 Future plan

The work was organised in three phases. In the first phase, the objectives of the project were understood and a research plan was written, and the existing code was piloted. In the second phase, code implementation and report writing will be completed. As a demo workshop is scheduled, preparation for the demo will be started in advance, and the code will be submitted in time to avoid uncontrollable factors. The final report should be submitted 2-3 weeks in advance. In phase 3, do the presentation and IRP end. The schedule is as follows.

4.1 Progress to date

1. (Implemented) Testing the use of the AI4PDEs approach to modelling airflow using data from South Kensington, London. Figure 1 shows the geometry for the South Kensington test case. Figure 2, 3 and 4 shows the airflow in this area with different time steps.

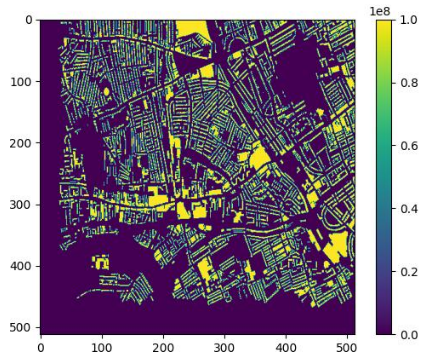

Figure 1: South Kensington

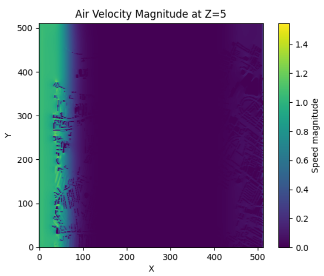

Figure 2: airflow time step=40

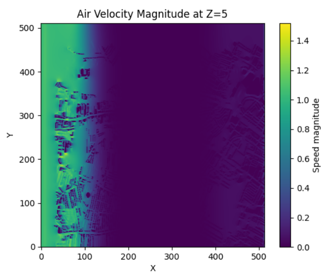

Figure 3: airflow time step=60

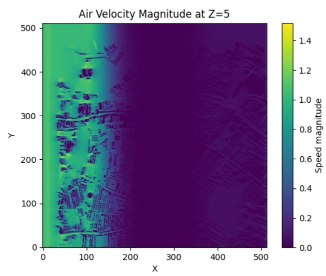

Figure 4: airflow time step=80

References

[1] Fang, Pain, Navon, and Xiao. An efficient goal-based reduced order model approach for targeted adaptive observations. *International Journal for Numerical Methods in Fluids*, 83: 263–275, 2017. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.4265.

[2] F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. E. Farrell, P. A. Allison, and A. J. H. Goddard. A POD reduced-order 4D-Var adaptive mesh ocean modelling approach. *International Journal for Numerical Methods in Fluids*, 60(7):709–732, 2009.

[3] Z. Che, F. Fang, J. Percival, C. C. Pain, O. Matar, and I. M. Navon. An ensemble method for sensor optimisation applied to falling liquid films. *International Journal of Multiphase Flow*, 67:153–161, 2014.

[4] A. Sandu, D. N. Daescu, and G. R. Carmichael. Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part II — numerical validation and applications. *Atmospheric Environment*, 37(36):5097–5114, 2003.

[5] D. N. Daescu and G. R. Carmichael. An adjoint sensitivity method for the adaptive location of the observations in air quality modeling. *Journal of the Atmospheric Sciences*, 60(2): 434–450, 2003.

[6] D. N. Daescu. On the sensitivity equations of four-dimensional variational (4D-Var) data assimilation. *Monthly Weather Review*, 136(8):3050–3065, 2008.

[7] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[8] S. Sanfui and D. Sharma. A three-stage graphics processing unit-based finite element analyses matrix generation strategy for unstructured meshes. *International Journal of Numerical Methods in Engineering*, 121:3824–3848, 2020. doi: 10.1002/nme.6383.

[9] Chen, Heaney, and Pain. Using ai libraries for incompressible computational fluid dynamics. 2024. in preparation.

[10] Chen, Heaney, Gomes, Matar, and Pain. Solving the discretised multiphase flow equations with interface capturing on structured grids using machine learning libraries. *arXiv preprint arXiv:2401.06755*, 2024. URL https://doi.org/10.48550/arXiv.2401.06755.

[11] Phillips, Heaney, Chen, Buchan, and Pain. Solving the discretised neutron diffusion equations using neural networks. *International Journal for Numerical Methods in Engineering*, 124(21):4659–4686, 2023. URL https://doi.org/10.1002/nme.7321.

[12] R. M. Bergmann, K. L. Rowland, N. Radnović, R. N. Slaybaugh, and J. L. Vujić. Performance and accuracy of criticality calculations performed using WARP – a framework for continuous energy monte carlo neutron transport in general 3D geometries on GPUs. *Annals of Nuclear Energy*, 103:334–349, 2017. doi: 10.1016/j.anucene.2017.01.027.

[13] N. Delbosc, J. L. Summers, A. I. Khan, N. Kapur, and C. J. Noakes. Optimized implementation of the lattice boltzmann method on a graphics processing unit towards real-time fluid simulation. *Computers & Mathematics with Applications*, 67:462–475, 2014. doi: 10.1016/j.camwa.2013.10.002.

[14] J. Chan, Z. Wang, A. Modave, J.-F. Remacle, and T. Warburton. Gpu-accelerated discontinuous galerkin methods on hybrid meshes. *Journal of Computational Physics*, 318: 142–168, 2016. doi: 10.1016/j.jcp.2016.04.003.

[15] M.-F. King, A. Khan, N. Delbosc, et al. Modelling urban airflow and natural ventilation using a GPU-based lattice-boltzmann method. *Building and Environment*, 125:273–284, 2017. doi: 10.1016/j.buildenv.2017.08.048.

[16] Wikipedia contributors. Lagrangian particle tracking, 2024. URL https://en.wikipedia.org/wiki/Lagrangian_particle_tracking. Accessed: 2024-06-09.

[17] C. Cierpka, B. Lütke, and C. J. Kähler. Higher order multi-frame particle tracking velocimetry. *Experiments in Fluids*, 54(5):1533, 2013.

[18] C. Cierpka, M. Rossi, R. Segura, and C. J. Kähler. On the calibration of astigmatism particle tracking velocimetry for microflows. *Measurement Science and Technology*, 22 (1):015401, 2010.

[19] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[20] Q. Gao, S. Pan, H. Wang, R. Wei, and J. Wang. Particle reconstruction of volumetric particle image velocimetry with the strategy of machine learning. *Advances in Aerodynamics*, 3:28, 2021.

[21] L. Brandt and F. Coletti. Particle-laden turbulence: progress and perspectives. *Annual Review of Fluid Mechanics*, 54:159–189, 2022.