# Deep Learning for Realistic Wind Field Prediction in a typical Urban Morphology for Application to Small Unmanned Aerial Systems

Rohit K.S.S. Vuppala * and Kursat Kara†
*School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, USA*

**This works aims to generate realistic wind data in urban spaces, which is essential in developing, testing and ensuring the safe operations of Small Unmanned Aerial Systems (sUAS) using Deep Learning (DL). This provides an alternative to existing turbulence models, specifically aimed at urban air spaces. We devise and utilize a Non-Intrusive Reduced Order Model (NIROM) approach to replicate and realistically predict wind fields in urban spaces. The method uses Large Eddy Simulation data from well-established computational fluid dynamics solvers like OpenFOAM to devise the NIROM. High-fidelity data generated from OpenFOAM is decomposed using Proper Orthogonal Decomposition (POD) into its orthogonal modes and basis. These orthogonal modes obtained over time are trained on specialized Recurrent Neural Networks like Long-Term Short Memory (LSTM) to complete the NIROM formulation. This method combined the traditional reduced order modeling approach with deep learning techniques to devise a framework for easy building and application of Machine Learning (ML) based Reduced Order Models (ROMs). A typical urban morphology subject to the wind is chosen and considered a test case for demonstrating the method.**

## I. Introduction

Unmanned Aerial Systems(UAS) have seen unprecedented growth over the past few decades. While initially designed and operated for military applications, with its availability in various configurations like heavier or lighter than air, fixed-wing, and rotorcraft, UAS has also been a ubiquitous presence in various civilian applications. It has penetrated into the domain of civilian operations for various applications like agricultural imaging[1], marine litter mapping[2], cartography, disaster management[3], urban land usage mapping[4], urban traffic mapping [5] and urban delivery, catering services[6]. The combination of ongoing urban expansion and ever-increasing interest in using Unmanned Aerial Vehicles has created the need to use Unmanned Aerial Systems for urban applications. Unlike UAS applications in other domains, their application in urban airspace is characterized by significant size and weight restrictions on them due to expectations of a dense urban air mobility ecosystem. Small Unmanned Aircraft Systems have rapidly evolved to tackle these limitations, but they remain susceptible to external factors like wind, gusts, and turbulent wakes in urban wind fields. In recent years urban wind fields have also been studied for various configurations like individual buildings[7], street canyons [8] or over limited areas using Computational Fluid Dynamics techniques[9, 10]. Researchers have also investigated the influence of wind on flight planning and trajectory[11, 12] to develop various control strategies to minimize the effects. However, testing these algorithms and methods with realistic wind data is still a significant challenge.

With advances in computing systems to perform efficient parallelization and develop consistent-stable numerical schemes, researchers can now simulate and analyze complex real-world processes. It is possible to compute complex flow fields with a high degree of accuracy and even collect flow-field data which cannot be gathered experimentally. Using Direct Numerical Simulations (DNS), the range of spatial and temporal scales of turbulent flows of interest could be resolved to obtain high-resolution data. However, in practice, Large Eddy Simulations (LES) are used to decrease the extremely prohibitive computational cost associated with DNS. But, Large Eddy Simulations could still be prohibitive in settings where large degrees of freedom need to be resolved or, when repeated, time-bound evaluations are necessary. Various Reduced Order Modeling (ROM) and surrogate modeling techniques have been proposed to tackle this challenge[13–15]. With the advent of machine learning/deep learning, a new class of Machine Learning based Reduced Order Models (ML-ROMs) have emerged. Since then, ML-ROMs have been widely used in many applications

*Graduate Research Assistant, Student Member AIAA
†Assistant Professor, kursat.kara@okstate.edu; Senior Member AIAA.

like process simulation and optimization [16], flow control [13], and fluid flows[17]. Machine Learning based Reduced Order Models (ROMs) use an efficient, robust methodology to generate Non-Intrusive Reduced Order Models for complex dynamical systems. While traditional ROMs have been used previously for various applications, prior knowledge of the governing equations or operator forms was required, especially for unsteady physics. Specialized neural networks like LSTM eliminate this need for generating ROMs for unsteady physics, which propagate the information learned from data in time. Moreover, other deep learning techniques like Convolutional Auto Encoders[18–20] allow the spatial information to be learned by the neural networks by applied self-learned convolutional filters on the training data have been useful for generating efficient NIROMs, similar to Proper Orthogonal Decomposition based methods[21–23].

In this work, we devise a ROM using deep learning and based on Proper Orthogonal Decomposition for flow predictions. A typical urban morphology is chosen, and Large Eddy Simulation data generated from a computational fluid dynamics solver is used to train the ML-ROM. We demonstrate the method for its utilization in making predictions in time. This, in effect, provides a methodology to generate realistic wind data needed for UAS testing and validation without expensive numerical simulations at all instances in time.

## II. Methodology

In this section, we discuss our approach starting with generating Large Eddy Simulation data and, subsequently a proper orthogonal decomposition-based formulation of the ML-ROM used in this work. In Section II.A we briefly describe the governing equations and turbulence closure for the LES method used along with details on the solver used for generating the data. Section II.B describes the details of the method used for generating the Machine Learning based ROM used.

### A. LES simulation

The incompressible fluid flow solver OpenFOAM[24] based on the finite volume method is used for generating the computational data inside the flow domain. PIMPLE algorithm, which combines the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations)[25] algorithm and the PISO (Pressure-Implicit with Splitting of Operators) algorithm[26] is used for computation. All inviscid terms and the pressure gradient are approximated with second-order accuracy. A first-order implicit Euler method is used for time integration together with the dynamic adjustable time stepping technique to guarantee a local Courant number less than 0.8. A Geometric-Algebraic Multi-Grid (GAMG) method is used for solving linear systems with a local accuracy of $1e^{-}5$ for all dependent variables at each time step.

*1. Governing Equations*

The continuity equation and momentum equation for the flow of an incompressible viscous fluid in Cartesian coordinates are given by:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{2}$$

Filtering the above equations and after simplification, we obtain:

$$\frac{\partial \overline{u_j}}{\partial x_j} = 0 \tag{3}$$

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu\frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} \tag{4}$$

However, it is impossible to determine the quantity $\frac{\partial \overline{u_i u_j}}{\partial x_j}$, but the quantity $\frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j}$ is known. We make a substitution and letting $\tau_{ij} = \frac{\partial \overline{u_i u_j}}{\partial x_j} - \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j}$ results in:

$$\frac{\partial \overline{u_i}}{\partial t} + \overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu\frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{5}$$

where,
$u_i (i = 1, 2, 3)$ represents the components of velocities,$\tau_{ij}$ is the sub-grid scale stress tensor.

Combined with the Boussinesq hypothesis, the sub-grid stress can be written as:

$$\tau_{ij} = -2\nu_t \overline{S}_{ij} + \frac{1}{3}\tau_{kk}\delta_{ij} \tag{6}$$

where,
$\nu_t = \frac{\mu_t}{\rho}$, $\mu_t$ is the sub-grid scale turbulent viscosity coefficient.
$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)$

*2. Turbulence closure*

In this work, we use the Wall-Adaptive Local Eddy-viscosity model (WALE) Sub Grid Scale closure presented in [27]. Compared with the Smagorinsky subgrid-scale (SGS) model, the WALE model considers the effects of turbulent wall surface effect and momentum transfer. The sub-grid scale turbulent viscosity is zero in the pure shear flow region, which ensures the accuracy of the flow field in the near-wall laminar flow region. The sub-grid scale turbulence viscosity coefficient is given by:

$$\nu_t = (C_w \Delta)^2 \frac{\left(\overline{S}^d_{ij}\overline{S}^d_{ij}\right)^{\frac{3}{2}}}{\left(\overline{S}_{ij}\overline{S}_{ij}\right)^{\frac{5}{2}} + \left(\overline{S}^d_{ij}\overline{S}^d_{ij}\right)^{\frac{5}{4}}} \tag{7}$$

where,
$C_w = 0.325$ and $\Delta$ is the filter scale determined by the lengths of the element in x,y,z directions and $\overline{S}^d_{ij}$ is computed using the relations below,

$$\overline{S}^d_{ij} = \frac{1}{2}\left(\overline{g}^2_{ij} + \overline{g}^2_{ji}\right) - \frac{1}{3}\delta_{ij}\overline{g}^2_{kk} \tag{8}$$

here,

$$\overline{g}_{ij} = \frac{\partial \overline{u_i}}{\partial x_j} \tag{9}$$

**B. Machine Learning based Reduced Order Model Methodology**

As outlined in Algorithm 1, we initially obtain the modal coefficients in time by performing a Proper Orthogonal Decomposition on the flow-field fluctuation data. Fluctuation data is calculated by subtracting the mean flow from LES data. A relative information content (RIC) index is computed for all the modes obtained. An optimal number of modal coefficients are chosen based on a user-specified threshold for training on a Recurrent Neural Network. RNNs contain cyclic or recurrent connections that enable them to continuously learn characteristics from a series of data and predict future outcomes. Closely following the methodology used in [21], we use Long Short-Term Memory (LSTM) neural networks, better suited for learning long-term dependencies in the input data. After training the network for the specified number of instances in time, model coefficients are predicted for future instances. These predictions are then used with the POD basis previously generated for the fluctuation field. We then compute the flow by adding the mean flow field from the neural network predictions to the fluctuation field.

**Algorithm 1** ROM-LSTM approach

1: Obtain 3D solution data from Large Eddy Simulations for the domain of interest.
2: Compute the fluctuation flow field for the given number of snapshots, i.e. mean-subtracted flow field

$$\overline{u}(x, y, z, t_n) = \frac{1}{N} \sum_{n=1}^{N} u(x, y, z, t_n)$$
$$u'(x, y, z, t_n) = u(x, y, z, t_n) - \overline{u}(x, y, z, t_n)$$

3: Compute the POD basis for the data matrix **A** made up from the data of the snapshot using Singular Value Decomposition.

$$\mathbf{A} = \Phi \Sigma \mathbf{V}$$

Where $\Phi$ is the basis vectors matrix, $\Sigma$ is a diagonal matrix with singular values.
4: Using relative information content of singular values, pick the optimal number of POD modes and basis vectors.
5: Find the modal coefficients using the optimal basis vectors matrix $\Phi_w$ and data matrix $A$,

$$\mathbf{C} = \mathbf{A}^T \Phi_w$$

6: Pre-process the data by scaling and re-arranging data for LSTM training with an appropriate look-back window.
7: Predict the modal coefficients with the trained network for future snapshots.
8: Using the optimal basis vectors calculate the fluctuation field, **U'**

$$\mathbf{U}' = \Phi_w \mathbf{C}^T$$

9: Compute the predicted flow field by adding the mean value to the predicted snapshot data.

# III. Results

## A. Simulation Setup

For this study we setup a set of buildings in a three-dimensional domain, representing a typical urban morphology for Large Eddy Simulation. The set of buildings used are from the publicly available OpenFOAM test case[28]. Initially a block mesh of 30x20x10 cells is generated with three levels of refinement in the region with buildings. Further, numerical grid generation was performed using the built-in snappyHexMesh, from building surface data file. The tool generates a 3-dimensional mesh containing hexahedra (hex) and split-hexahedra (split-hex) automatically from triangulated surface geometries[29]. The final snapped mesh has around 181000cells with 14000 polyhedra, 166000 hexahedra and 1000 prisms. The numerical grid used is depicted in Fig. 1. A constant vertical wind profile of magnitude of 10m/s was chosen at the inlet. The y and z components of the velocity are chosen zero, with flow only in the x-direction.

| Domain size | Specification |
|---|---|
| x-direction | 400m |
| y-direction | 280m |
| z-direction | 140m |

**Table 1 Domain details**

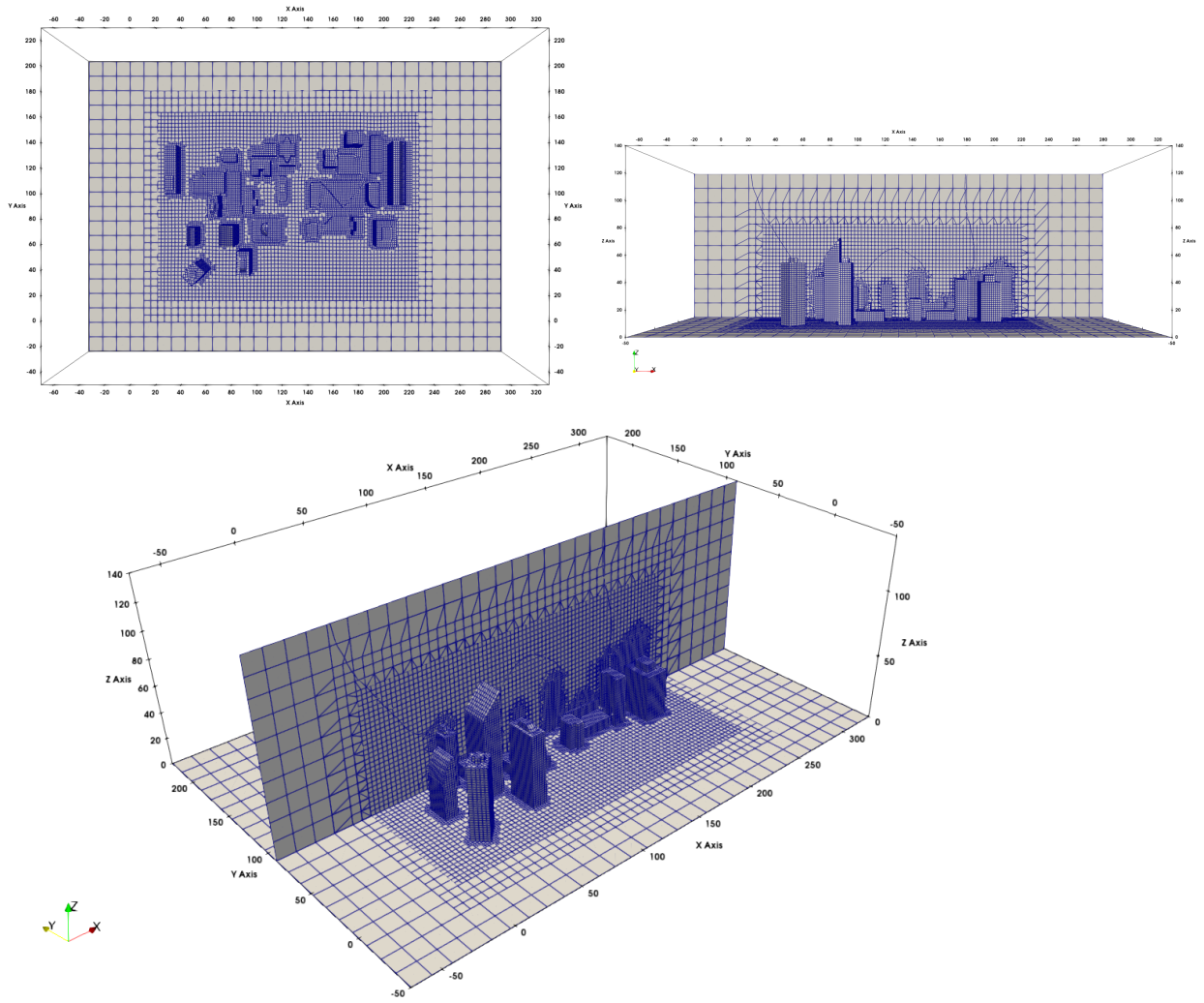



**Fig. 1 Left-Top view (xy-plane); Right-Side view (xz-plane) of the domain; Bottom-Isometric view of the domain**

### B. ML-ROM Results

We use the data from the Large Eddy Simulation using the domain in Fig. 1. The CFD simulation is run till a quasi-stable state is obtained. To simplify the model, only the x-component of velocity $u$ is utilized for this work. The simulation data from this domain at every 1 second (snapshot data) is used for training the ML-ROM. Following Algorithm 1, we pick a threshold for relative information content as 60% giving us 133 modes as shown in Fig 2. The LSTM neural network is trained on the snapshot data for 500 seconds, and predictions are made for 100 more seconds; some more details about the neural network architecture are listed in Table 2.

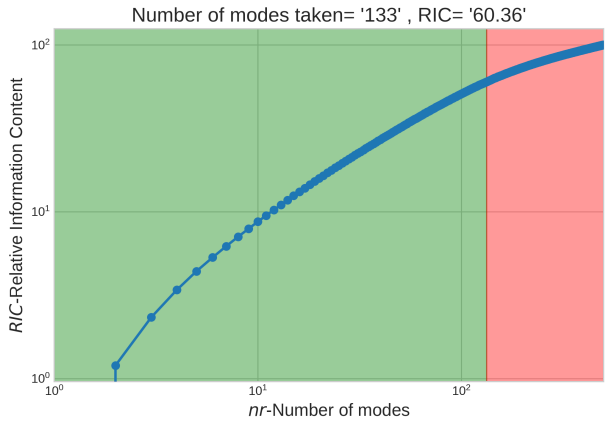



**Fig. 2 Modes and their Relative Information Content; green-modes taken, red-modes neglected**

A comparison is made between the true modal coefficients and ROM-LSTM predictions in Fig. 3 for the first eight modes, selected based on relative information content. We see a close agreement in the predictions made from ROM-LSTM and actual modes from the POD decomposition on the LES data. A mismatch is noticeable in the modal predictions, especially in the amplitude. However, the general trend seems to be similar.

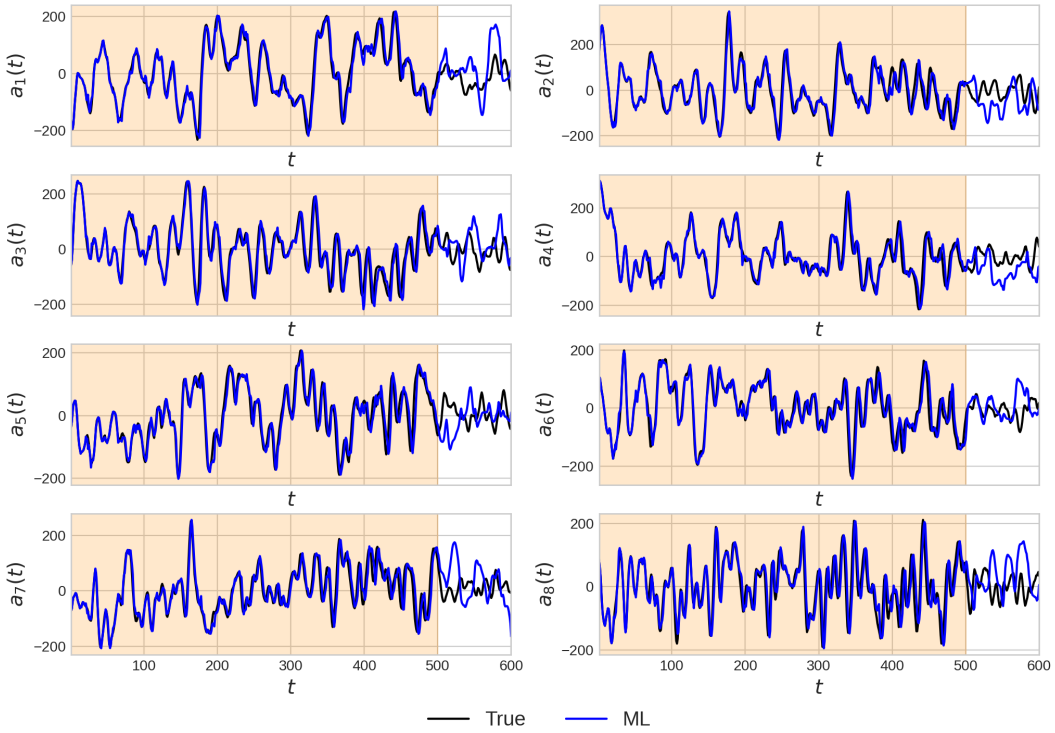



**Fig. 3 Comparison between True and ROM-LSTM (ML) for first eight modes for demonstration; Background colors: Tan/Orange - Training, White - Prediction**

A contour plot of x-directions velocity is depicted for the top view (xy plane), and comparisons are made as shown in Section III.B. A similar comparison is also made for a side view (xz plane) in the center of the domain in Section III.B.

| Parameter | Specification |
|---|---|
| Number of hidden layers | 4 |
| Number of neurons in each hidden layer | 128 |
| Activation function | tanh |
| Lookback time-window | 20 |
| Recurrent dropout | 0.8 |
| Neuron dropout | 0.0 |
| Loss function | MSE |
| Optimiser | ADAM |
| Training-testing ratio | 5:1 |

**Table 2 Neural Network details**

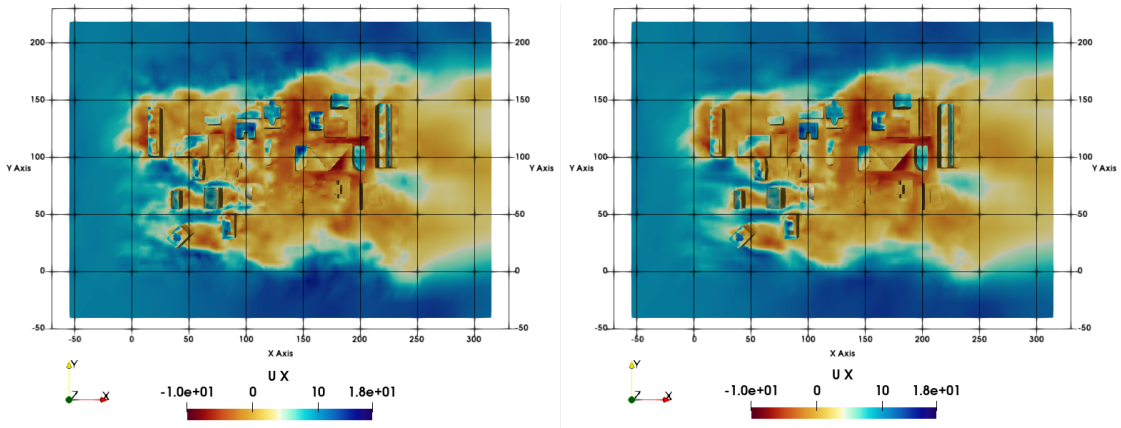
It could be noticed from the comparisons made in the contour plots that the finer structures in the flow are missing. However, the ML-ROM can reasonably predict the larger structures and their movement in the domain. The model's performance deteriorates over time, but overall, the predictions closely resemble the actual flow field.

## IV. Conclusions and Future work

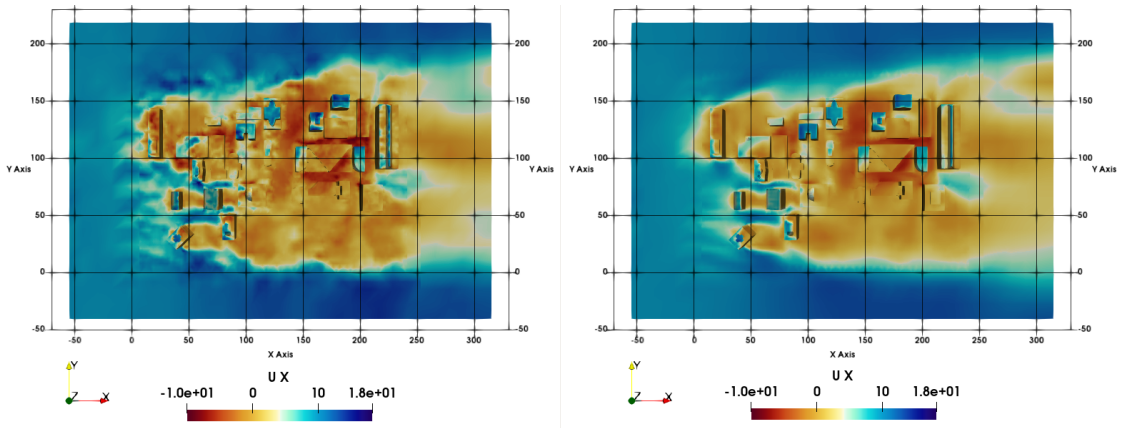
In this work, we utilize deep learning to create a Machine Learning based Reduced Order Model for realistic wind-field prediction. A demonstration of the method is made for a test case with typical urban morphology. We notice a good agreement in the flow field predicted and the actual data from LES simulations. However, while the large structures are well predicted, the finer structures in the flow field are missing. This could be attributed to the modes not utilized for the prediction from Proper Orthogonal Decomposition. The corresponding results could be improved by choosing a higher RIC threshold to incorporate more modes. The work could also be extended using Convolutional Auto Encoders (CAE) or Variational CAE to decompose the data instead of POD, to enable the ML-ROM learning from the spatial orientation of data in the flow domain. Furthermore, using a dynamic basis rather than a constant basis used by POD could reduce the projection error over time and prevent the model predictions from deteriorating quickly.

## V. Acknowledgements

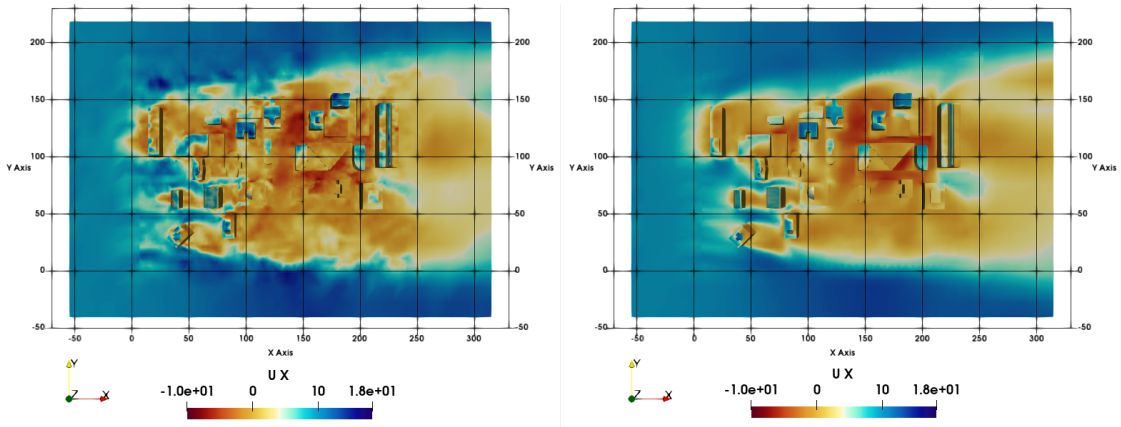
This material is based upon work supported by the National Science Foundation under Grant No. 1925147. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Some of the computing for this project was performed at the High-Performance Computing Center(HPCC) at Oklahoma State University supported in part through the National Science Foundation grant OAC-1531128. We would like to acknowledge high-performance computing support from Cheyenne[30] (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation. This work also used Bridges2 at PSC through allocation MTH220018 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296.




(a) 500th snapshot

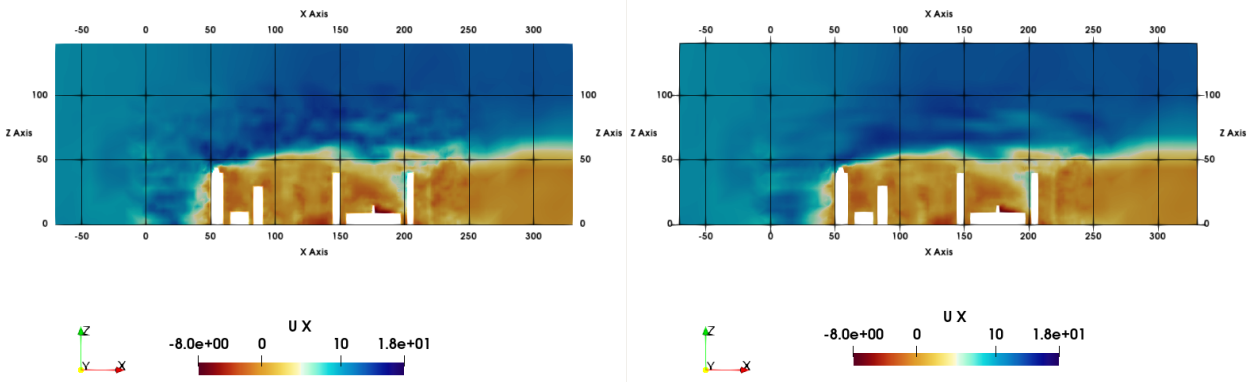



(b) 550th snapshot

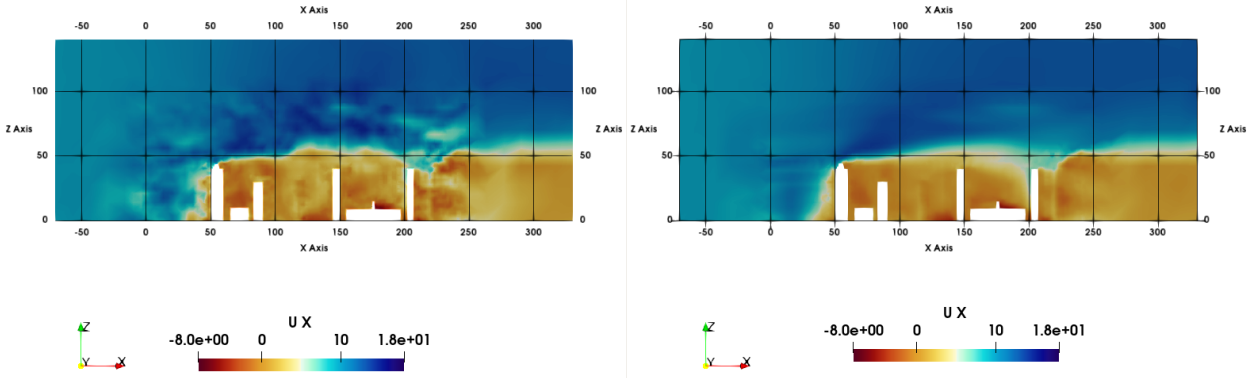



(c) 600th snapshot

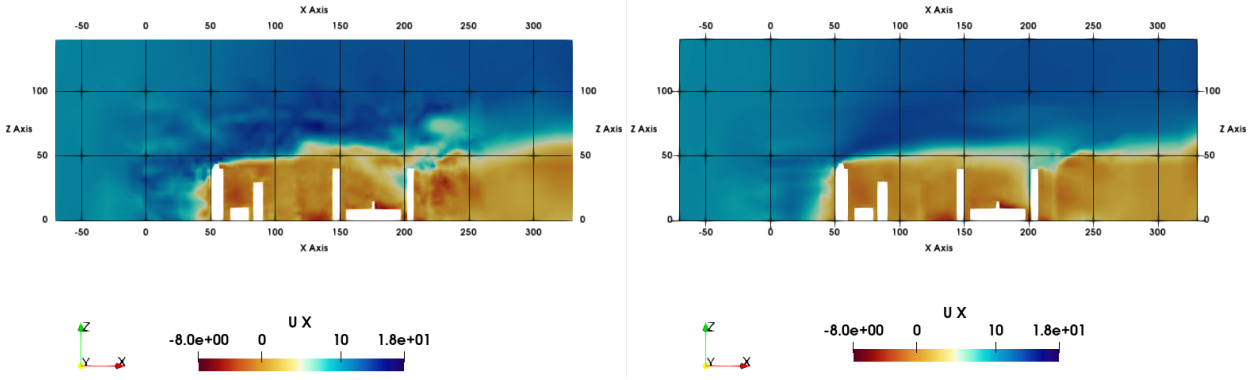
**Fig. 4 x-direction velocity contour for top view of the domain**




(a) 500th snapshot




(b) 550th snapshot




(c) 600th snapshot

**Fig. 5 x-direction velocity contour for side view at center of the domain**

# References


[1] Hassler, S. C., and Baysal-Gurel, F., "Unmanned aircraft system (UAS) technology and applications in agriculture," *Agronomy*, Vol. 9, No. 10, 2019, p. 618.

[2] Gonçalves, G., Andriolo, U., Pinto, L., and Bessa, F., "Mapping marine litter using UAS on a beach-dune system: a multidisciplinary approach," *Science of The Total Environment*, Vol. 706, 2020, p. 135742. doi:https://doi.org/10.1016/j.scitotenv.2019.135742, URL https://www.sciencedirect.com/science/article/pii/S0048969719357377.

[3] de Oliveira Silva, L., de Mello Bandeira, R. A., and Campos, V. B. G., "The use of UAV and geographic information systems for facility location in a post-disaster scenario," *Transportation Research Procedia*, Vol. 27, 2017, pp. 1137–1145.

[4] Grubesic, T. H., and Nelson, J. R., "UAS Platforms and Applications for Mapping and Urban Analysis," *UAVs and Urban Spatial Analysis*, Springer, 2020, pp. 13–29.

[5] Murphy, D. W., and Cycon, J., "Applications for mini VTOL UAV for law enforcement," *Sensors, C3I, information, and training technologies for law enforcement*, Vol. 3577, International Society for Optics and Photonics, 1999, pp. 35–43.

[6] León, C. D., "Drone Delivery? Amazon Moves Closer With F.A.A. Approval," , Aug 2020. URL https://www.nytimes.com/2020/08/31/business/amazon-drone-delivery.html.

[7] Landua, T. R., Vuppala, R. K. S. S., and Kara, K., "Investigation of Airflow around Buildings using Large Eddy Simulations for Unmanned Air Systems Applications," *AIAA SCITECH 2022 Forum*, 2022, p. 1688.

[8] Carpentieri, M., and Robins, A. G., "Influence of urban morphology on air flow over building arrays," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 145, 2015, pp. 61–74.

[9] Wise, D., Boppana, V., Li, K., and Poh, H., "Effects of minor changes in the mean inlet wind direction on urban flow simulations," *Sustainable cities and society*, Vol. 37, 2018, pp. 492–500.

[10] Toparlar, Y., Blocken, B., Maiheu, B., and Van Heijst, G., "A review on the CFD analysis of urban microclimate," *Renewable and Sustainable Energy Reviews*, Vol. 80, 2017, pp. 1613–1640.

[11] Raza, S. A., "Autonomous UAV control for low-altitude flight in an urban gust environment," Ph.D. thesis, Carleton University, 2015.

[12] Tabassum, A., Vuppala, R. K., Bai, H., and Kara, K., "Variance Reduction of Quadcopter Trajectory Tracking under Stochastic Wind Disturbances," *arXiv preprint arXiv:2104.10266*, 2021.

[13] Noack, B. R., Morzynski, M., and Tadmor, G., *Reduced-order modelling for flow control*, Vol. 528, Springer Science & Business Media, 2011.

[14] Brunton, S. L., and Noack, B. R., "Closed-Loop Turbulence Control: Progress and Challenges," *Applied Mechanics Reviews*, Vol. 67, No. 5, 2015. doi:10.1115/1.4031175, URL https://doi.org/10.1115/1.4031175, 050801.

[15] Rowley, C. W., and Dawson, S. T., "Model reduction for flow analysis and control," *Annu. Rev. Fluid Mech*, Vol. 49, No. 1, 2017, pp. 387–417.

[16] Lang, Y.-d., Malacina, A., Biegler, L. T., Munteanu, S., Madsen, J. I., and Zitney, S. E., "Reduced order model based on principal component analysis for process simulation and optimization," *Energy & Fuels*, Vol. 23, No. 3, 2009, pp. 1695–1706.

[17] Xie, X., Mohebujjaman, M., Rebholz, L. G., and Iliescu, T., "Data-driven filtered reduced order modeling of fluid flows," *SIAM Journal on Scientific Computing*, Vol. 40, No. 3, 2018, pp. B834–B857.

[18] Vuppala, R. K., and Kara, K., "Wind Field Prediction in Urban Spaces for small Unmanned Aerial Systems using Convolutional Autoencoders," *AIAA AVIATION 2022 Forum*, 2022, p. 3605.

[19] Vuppala, R. K. S. S., and Kara, K., "Realistic Wind Data Generation for Small Unmanned Air Systems in Urban Environment using Convolutional Autoencoders," *APS Division of Fluid Dynamics Meeting Abstracts*, 2021, pp. T29–005.

[20] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c., "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, 2015, pp. 802–810.

[21] Vuppala, R. K., and Kara, K., "A non-intrusive reduced order model using deep learning for realistic wind data generation for small unmanned aerial systems in urban spaces," *AIP Advances*, Vol. 12, No. 8, 2022, p. 085020.

[22] Vuppala, R. K., and Kara, K., "A Novel Approach in Realistic Wind Data Generation for The Safe Operation of Small Unmanned Aerial Systems in Urban Environment," *AIAA AVIATION 2021 FORUM*, 2021, p. 2505.

[23] Benner, P., Gugercin, S., and Willcox, K., "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, Vol. 57, No. 4, 2015, pp. 483–531.

[24] Jasak, H., Jemcov, A., Tukovic, Z., et al., "OpenFOAM: A C++ library for complex physics simulations," *International workshop on coupled methods in numerical dynamics*, Vol. 1000, IUC Dubrovnik Croatia, 2007, pp. 1–20.

[25] Caretto, L., Gosman, A., Patankar, S., and Spalding, D., "Two calculation procedures for steady, three-dimensional flows with recirculation," *Proceedings of the third international conference on numerical methods in fluid mechanics*, Springer, 1973, pp. 60–68.

[26] Issa, R. I., "Solution of the implicitly discretised fluid flow equations by operator-splitting," *Journal of computational physics*, Vol. 62, No. 1, 1986, pp. 40–65.

[27] Nicoud, F., and Ducros, F., "Subgrid-scale stress modelling based on the square of the velocity gradient tensor," *Flow, turbulence and Combustion*, Vol. 62, No. 3, 1999, pp. 183–200.

[28] Mohan, R., Sundararaj, S., and Thiagarajan, K. B., "Numerical simulation of flow over buildings using OpenFOAM®," *AIP Conference Proceedings*, Vol. 2112, AIP Publishing LLC, 2019, p. 020149.

[29] "4.4 mesh generation with the snappyhexmesh utility," , ???? URL `https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.4-mesh-generation-with-the-snappyhexmesh-utility`.

[30] Computational, and Laboratory, I. S., "Cheyenne: HPE/SGI ICE XA System (University Community Computing)," , 2017.