Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Unstructured Convolutional Autoencoders for Big Data Assimilation

Author:
Maxime Redstone Leclerc

Supervisor:
Rossella Arcucci

Submitted in partial fulfillment of the requirements for the MSc degree in Computing Science of Imperial College London

June 2020

Abstract

Data Assimilation (DA) provides a framework to incorporate observations into forecasting models thereby improving predictions. Many attempts have been made to improve the efficiency and accuracy of this process. Techniques such as Principal Component Analysis (PCA) or Truncated Singular Value Decomposition (TSVD) have been used to reduce the space. However, this de facto entails a loss of information. Fluidity, the state-of-the-art Computational Fluid Dynamics (CFD) software, has a adaptive mesh capability and renders unstructured meshes. This work focuses on applying deep learning techniques and particularly Autoencoders (AEs) on unstructured meshes to reduce the space without any loss of information for a Big Data problem. Testing experiments will be conducted using pollution tracers from a London area.

Contents

Chapter 1

Introduction

Climate change and air pollution are amongst the most pressing challenges to date. Outdoor air pollution will result in four-million deaths per year by 2050 according to current estimates [1]. This represents over ten times the number of deaths from COVID-19 as of June 2020 [2]. A deeper understanding of air flows and pollution transport at pedestrian level (i.e. at micro-scale) is crucial to guarantee the development of sustainable cities in the future.

Modelling is extensively used to obtain accurate predictions of dynamical systems. However forecasts from models continuously diverge from reality as time progresses [3]. Data Assimilation (DA) is a technique used to integrate information provided by environmental observations into a forecasting model. DA has been used in many field and improving the predictions in weather forecasting, air pollution and oceanography for instance is of the upmost importance in light of the climate crisis.

Alongside the immense potential of having more data available comes the difficulty to efficiently analyse and use that information in a Big Data problem. Indeed, within the Data Assimilation process, techniques such as Principal Component Analysis (PCA), Truncated Singular Value Decomposition (TSVD) or Tikhonov regularization are used to make the DA process computable [4] [5] [6]. However, these techniques involve an inevitable loss of information [7]. Autoencoders (AEs) are a powerful technique that have been used to reduce the space in DA but require structured meshes [8]. This involves interpolation which in turn introduces errors.

This project makes the following novel contributions:

- Adapts AEs to unstructured meshes in a DA process.
- Uses AEs to conduct DA as part of a Big Data problem.

This project is conducted as part of the *'Managing Air for Green Inner Cities'* (MAGIC) project which aims at finding *'a cost-beneficial method in which to change the way our cities are developing'* and the *'Health assessment across biological length scales for personal pollution exposure and its mitigation'* (INHALE) project that studies the impact of air pollution on health. This work builds upon previous contributions to the MAGIC project [8] [9]. More specifically, the aim is to concentrate on the most relevant sub-domains identified in [9] (i.e. those that contain the most relevant information) and adapt the AEs used on structured meshes in [8] as the starting point for this project.

The methodology will be tested on data obtained from the MAGIC test site (a 500m radius circle around St George's Circus in South London, UK). This data represents a high resolution map of the air flows and pollution concentrations obtained using Fluidity (`http://fluidityproject.github.io/`), the state-of-the-art computational fluids dynamics (CFD) software.

Chapter 2 of this report addresses the background information required to understand the concept of Data Assimilation while Chapter 3 underlines the fundamental principles of AEs. Chapter 4 introduces the basis of this project.

Chapter 2

Background: Data Assimilation

2.1 Related Works

Data Assimilation (DA) has been in use for the past forty years and applied to multiple fields. It was first used for numerical weather prediction (NWP) [10] [11] [12] but has now been applied to planetary climate [13], oceanography [14], biology [15] and urban pollution modelling [4] (in the context of the MAGIC project).

DA couples forecasting models with newly available observations made over time to provide a 'most accurate estimate' of the current state of the system, the *analysis*. It also quantifies the uncertainty of the estimated state of the system. One can distinguish between two basic types of DA [16]:

- Sequential assimilation that only considers observation made in the past until the time of analysis, which is the case for real-time assimilation systems.
- Non-sequential assimilation or retrospective assimilation that can also consider observations from the future.

The DA process is computationally challenging. The European Centre for Medium-Range Weather Forecasts (ECMWF) uses 25 million observations twice a day to correct the 150 million variables that define the model's virtual atmosphere. This takes as much computer power as the 10-day forecast [17]. This example shows how the DA problem is underconstrained as fewer observations than model parameters exist. This requires an *a priori* guess: the *background* state. The estimation of errors from the observations and the model must be taken into account to provide the most accurate assimilation scheme [18].

Most DA techniques are based on Bayesian probabilistic theories and can be divided into two main strategies [19] [20] [16]:

- Variational DA (VarDA) is based on minimising a cost function that computes the distance between the observations and the model. The solution is found by evaluating the cost function and its gradient iteratively. This method relies on errors described by error covariance matrices. VarDA regroups 3D-VaR and

4D-VaR. The latter is a generalisation of the former for observations that are distributed in time which makes flow-dependent corrections to the first guess trajectory possible.

- Filtering can take many forms depending on the problem at hand. The most well-known methods involved Kalman Filter (KF), Extended Kalman Filter (EKF), Reduced-Rank Kalman Filter (RRKF), Ensemble Kalman Filters (EnKF) and Particle Filter (PF). KF is based on finding the solution with the minimum variance while EKF extends the KF approach to non-linear systems. RRKF approximates the KF approach for large dimensional systems. EnKF is based on ensemble forecasts which produces multiple analysis. As opposed to standard KF the error covariance matrices are not computed but simply sampled from the forecasts which makes it more scalable. PF models non-linear systems and allows for non-Gaussian probability distribution which is not the case for the previous Kalman methods described [17].

In recent years multiple attempts have been made to combine strategies and yield a hybrid approach [21] [22]. Specific factors may influence the use of a particular strategy to solve the problem at hand and some criteria are detailed in [20].

Sequential VarDA provides a more rapid and robust approach than other statistical implementations to incorporate environmental observations in real time. It has been the basis of most operational implementations of DA in NWP centres such as the Met Office [12] and the ECMWF [11]. This is why Section 2.2 introduces the notations and definitions for the VarDA mathematical framework and Section 2.3 guides the VarDA formulation and explains where in the DA process the work of this project intervenes.

2.2 Definitions and Notations

The definitions and notations used are based on [20].

Let $\mathbf{x}_t$ represent the state of the model at time t such that:

$$\mathbf{x}_t \in R^{NP} \tag{2.1}$$

where NP is the number of elements in the state vector. For T time steps, this yields a single matrix:

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T] \in R^{NP \times T} \tag{2.2}$$

One can relate the state $\mathbf{x}_t$ by propagating $\mathbf{x}_{t-1}$ forward by one time step using the non-linear model, $\mathcal{M}_{t-1,t}$.

$$\mathbf{x}_t = \mathcal{M}_{t-1,t}\left[\mathbf{x}_{t-1}\right] + \eta_t \tag{2.3}$$

where η_t represents the model error introduced over time $t-1 \rightarrow t$. This can result from an inaccurate representation of the physics involved.

Let $\mathbf{x}_0^b$ be the background state at time step 0. This contains the *a priori* information mentioned in the previous section. To obtain future background states, the following equation is used:

$$\mathbf{x}_t^b = \mathcal{M}_{t-1,t}\left[\mathbf{x}_{t-1}^b\right] \tag{2.4}$$

Let $\mathbf{y}_t$ represent the observation space of the system at time t such that:

$$\mathbf{y}_t \in R^{Nobs} \tag{2.5}$$

where $Nobs << NP$. The mapping from the state of the system to the observations at time t is done via the non-linear observation operator, $\mathcal{H}_t$:

$$\mathbf{y}_t^o = \mathcal{H}_t\left[\mathbf{x}_t\right] + \epsilon_o \tag{2.6}$$

where ϵ_o and $\mathbf{y}_t^o$ represent the observation error and the real observation at time t respectively. The observation error covariance matrix will be noted $\mathbf{R}_t$.

Quantifying error will help assess the data assimilation process. We represent the different sources of uncertainty in the background, observations and analysis using probability density functions.

Let the background error be defined by:

$$\epsilon_b = \mathbf{x}^b - \mathbf{x}^{TrueState} \tag{2.7}$$

yielding a background error covariance matrix $\mathbf{B}_t$:

$$\mathbf{B}_t = \left(\mathbf{x}_t^b - \mathbf{x}_t^{TrueState}\right)\left(\mathbf{x}_t^b - \mathbf{x}_t^{TrueState}\right)^T \tag{2.8}$$

Table 2.1 summarises the notations used.

2.3 VarDA Formulation

VarDA consists in minimizing a cost function in order to get the most accurate representation of the system given the available observations $\mathbf{y}^o$, the physics described in the forecast model, $\mathcal{M}_{t-1,t}$ and the uncertainties. The problem requires to find the state $\mathbf{x}$ for which the following equation holds [20]:

$$\mathbf{x}^{DA} = \arg\min_{\mathbf{x}} J\left(\mathbf{x}\right) \tag{2.9}$$

Symbol	Description	Size
$\mathbf{x}_t$	State vector at time t	NP
$\mathbf{x}_0^b$	Background state at time 0	NP
$\mathbf{X}$	State vector at all times	NP x T
η_t	Model error at time t	NP
$\mathbf{y}_t^o$	Real observations at time t	Nobs
$\mathcal{M}_{t-1,t}$	Non-linear model	NP
$\mathcal{H}_t$	Non-linear observation operator at time t	in:NP out:Nobs$_t$
$\mathbf{B}_t$	Background error covariance matrix	NP x NP
$\mathbf{R}_t$	Observation error covariance matrix	Nobs x Nobs
$\mathbf{H}_t$	Linear observation operator at time t	Nobs$_t$ x NP
$\mathbf{d}_t$	Misfit	Nobs$_t$

Table 2.1: Notations summary where NP is the number of points, Nobs the total number of observations (Nobs$_t$ is the observations at time t) and T the number of time steps.

where $J(\mathbf{x})$ is given by:

$$J(\mathbf{x}) = \alpha \left\|\mathbf{x} - \mathbf{x}^b\right\|^2_{\mathbf{B}_0^{-1}} + \alpha \sum_{t=0}^{T} \|\mathbf{y}_t^o - \mathcal{H}_t[\mathbf{x}_t]\|^2_{\mathbf{R}_t^{-1}} + \alpha \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{M}_{t-1,t}[\mathbf{x}_{t-1}]\|^2_{Q_t^{-1}} \tag{2.10}$$

where α is a regularization parameter. Using $\alpha = 1$ can be interpreted as weighting the observations and the background state in a similar way [23]. In equation 2.10, $\|\mathbf{a}\|^2_{\mathbf{A}^{-1}} \equiv \mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{a}$

This cost function contains three terms:

- First term, $\mathbf{J}_b$, represents the background term. This calculates the discrepancy between the initial guess, $\mathbf{x}^b$ and the system's state after the DA phase.
- Second term, $\mathbf{J}_o$, represents the observation term. This calculates the discrepancy between the actual observations and the predicted state of the system according to the model.
- Third term, $\mathbf{J}_q$, represents the model error term. This calculates the discrepancy between any two model predictions t and $t-1$.

This formulation is the most general VarDA approach and is known as 4D-Var. It uses both three dimensions of space and one dimension of time. In practice, the computation of the above equation is not doable and additional approximations are required.

2.3.1 Incremental VarDA

An incremental VarDA formulation has been developed to deal with non-linear observation operator and/or models to still minimise the cost function. This involves updating the reference state with a perturbation [24].

$$\mathbf{x}_t = \mathbf{x}_t^{\mathrm{b}} + \delta \mathbf{x}_t \tag{2.11}$$

Under the following assumptions [20]:

- *Strong-constraint* 4D-VAR: this assumes the model is perfect so that the state of the system, $\mathbf{x}$, is fully determined by the initial condition, $\mathbf{x}_0$. Under this assumption, the third term, $\mathbf{J}_q$ can be discarded and therefore only the state at time step 0 is used to determine all the subsequent states.
- 4D-Var to 3D-Var: this excludes the time dimension and therefore assumes $\mathbf{M}_{t_1,t_2} = \mathbf{I}\ \forall t_1, t_2$.

linearising the problem around the background state yields [4]:

$$\delta \boldsymbol{x}_0^{DA} = \underset{\delta \boldsymbol{x}_0}{\arg\min} J(\delta \boldsymbol{x}_0) \tag{2.12}$$

$$J(\delta \mathrm{x}_0^{DA}) = \alpha \|\delta \mathrm{x}_0\|^2_{\mathrm{B}_0^{-1}} + \alpha \sum_{t=0}^{T} \|\delta \mathbf{d}_t\|^2_{\mathrm{R}_t^{-1}} \tag{2.13}$$

where $\mathbf{d}$ is the misfit and is defined by:

$$\boldsymbol{d}_t = \boldsymbol{y}_t^o - \boldsymbol{H}_t \boldsymbol{x}_t^b \tag{2.14}$$

and the increment by:

$$\delta \mathbf{d}_t = \mathbf{d}_t - \mathbf{H}_t \delta \mathbf{x} \tag{2.15}$$

where $\mathbf{H}_t$ is the linearised observation operation.

2.3.2 Control Variable Transform

As can be seen in Equation 2.13, calculating $\mathbf{J}(\delta \mathbf{x}_0^{DA})$ involves knowing the matrices $\mathbf{B}_0$ and $\mathbf{R}_t$ explicitely. The Control Variable Transform (CVT) method defines the cost function in terms of *control variables* which yields an error covariance matrix of $\mathbf{I}$ and makes the calculation computable [25]. A schematic representation of the transformation is showed in Figure 2.1 (adapted from [26]).

This is done by letting $\mathbf{B}_0 = \mathbf{V}\mathbf{V}^T$ as this implies $\mathbf{V}^T\mathbf{B}_0^{-1}\mathbf{V} = \mathbf{I}$. This way we get:

$$\mathcal{X}^{DA} = \underset{\mathcal{X}}{\arg\min} J(\mathcal{X}) \tag{2.16}$$

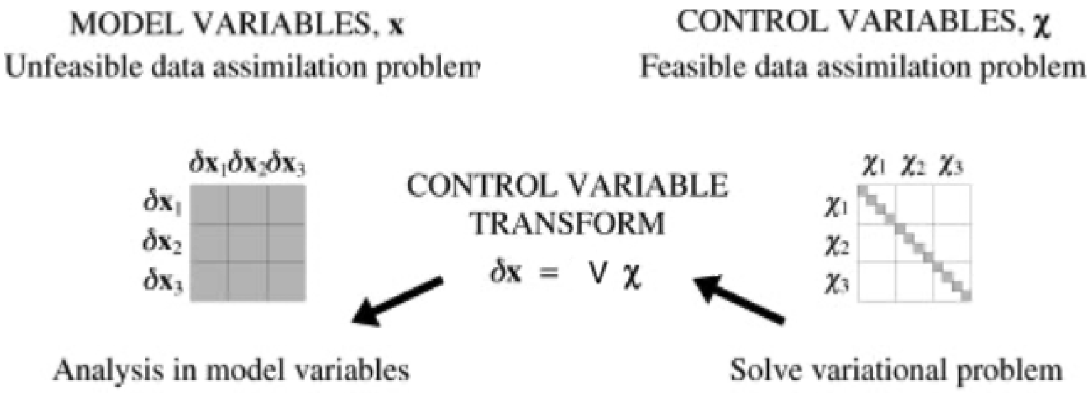

MODEL VARIABLES, x
Unfeasible data assimilation problem
CONTROL VARIABLES, χ
Feasible data assimilation problem
δx1 δx2 δx3
δx1
δx2
δx3
CONTROL VARIABLE TRANSFORM
δx = V χ
χ1 χ2 χ3
χ1
χ2
χ3
Analysis in model variables
Solve variational problem

Figure 2.1: Representation of CVT and impact on covariance matrices. Background error covariance matrix in model variables (left) and control variables (right). Shaded matrix elements represent non-zero elements.

$$J(\mathcal{X}) = \frac{1}{2}\alpha\mathcal{X}^T\mathcal{X} + \frac{1}{2}\alpha\|\mathbf{d} - \mathbf{H}\mathbf{V}\mathcal{X}\|^2_{R_t^{-1}} \tag{2.17}$$

where $\mathcal{X} = \mathbf{V}^{+}\delta\mathbf{x}$ and $\mathbf{V}^{+}$ is the generalised inverse of $\mathbf{V}$ [4]. In most cases, the observation error covariance matrix $\mathbf{R}_t$ is diagonal. This is because there is no reason to assume observation error correlations between independent sensors.

This general formulation of the 3D-Var approach is still ill-conditioned and requires preconditioning.

2.3.3 Minimisation and Preconditioning

The minimization procedure can be done with a variety of techniques such as Gradient descent, Newton and *Quasi-Newton* methods or Limited-Broyden Fletcher Goldfarb Shanno (L-BFGS). The later was used in the past [4] and is considered the fastest for large scale optimisation problems [27]. However, the convergence rate of L-BFGS depends on the conditioning of the numerical problem and therefore on the condition number of the Hessian of $J(\mathcal{X})$ which in turns depends on the condition number of **V**.

To precondition **V**, a variety of methods have been used including PCA, TSVD and Tikhonov regularization [4] [6] [5], all of which involve a lost of information. **This** is the step of the DA process this project will focus on.

Chapter 3

Background: Autoencoders

This project builds upon the work done in [8] and uses it as a starting point for the development of our methodology. It is based on Autoencoders (AEs). AEs are an unsupervised machine learning method that contain three parts as shown in Figure 3.1:

- An Encoder which takes an input, $\mathbf{x}$, and translates it to a reduced space representation, called the *code*, $\mathbf{h}$.
- A Decoder which translates the code, $\mathbf{h}$, back into a reconstruction of the input, $\mathbf{r}$.
- Loss function, $\mathcal{L}$, to compute the discrepancy between the final reconstruction and the original input. The goal being to minimize this discrepancy to yield the most accurate compression-decompression method.

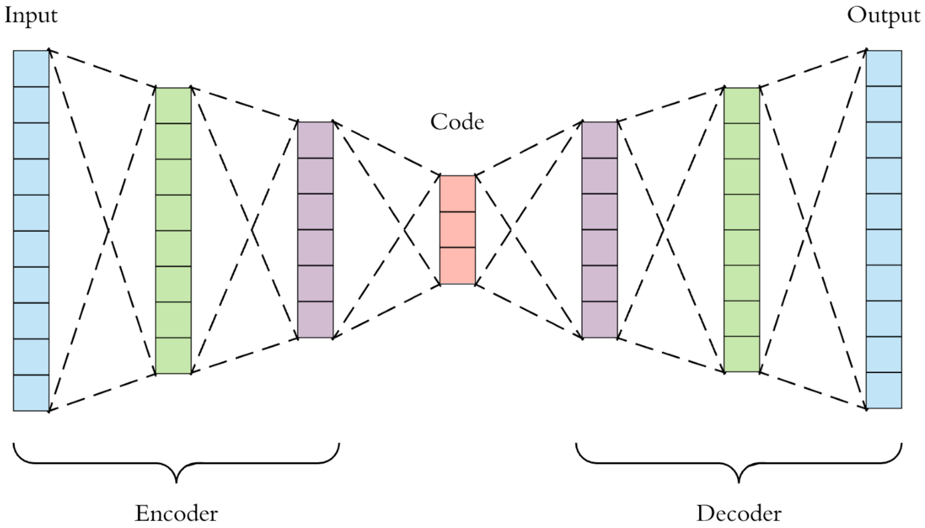


Figure 3.1: Overview of an autoencoder design [28].

There exists multiple variations of the standard, or often called vanilla, AEs such as regularized AEs, sparse AEs, denoising AEs, variational AEs (VAEs) and their precise definitions are detailed in [29]. This technology has been used across multiple fields to reduce the dimensionality of the problems at hand.

In **Geometry Processing**, a combination of VAEs and Convolutional AEs (CAEs) have been studied [30] [31] [32] [33] and prove to yield better results than techniques such as Principal Component Analysis (PCA). In these applications the reduction of the space can be used to represent the deformation of 3D shapes for example.

In **Reduced Order Modelling** (ROM), the complex forecast model (i.e. Fluidity for example) can be replaced with a less complex model using AEs. As during a 4D-DA scheme the model is used at each step of the DA process, this translates to a much more efficient way to perform DA. In particular when compared to ROM conducted with PCA as in [34] [35] [36].

Moreover, in many applications, the problem deals with non-linear patterns which AEs are able to model, which is not the case of PCA techniques that can only create modes that are linear combinations of the inputs [37]. By ensuring the code **h** of the AE has a smaller dimension than the input **x**, AEs are able to focus on the most relevant features in the data [29]. AEs have been used in [8] as part of the DA process and again provided better results than when PCA was used such as in [38] [39]. This motivates the use of AEs in DA.

3.1 Autoencoders in DA

In the context of DA, AEs have been used to produce ROM. The works described in [34] [40] make use of these deep learning tools to improve the robustness of their model's predictions. It is important to note that using these techniques involves a training process which can be time consuming as stated in [34].

AEs have also been used to reduce the space of the DA process in the case of the 3D-VarDA. The author in [8] used CAE to conduct the DA in a *bi-reduced space* which improved both the efficiency and accuracy of the process. *Bi-reduced* refers to the fact that the space is first reduced using the Control Variable Technique (CVT) and then using AEs. They proved autoencoders can produce better results than TSVD and PCA to reduce the space while still preconditioning V. Their 3D-VarDA formulation is given by:

$$\mathbf{w}_l^{DA} = \underset{\mathbf{w}_1}{\arg\min} J\left(\mathbf{w}_l\right) \tag{3.1}$$

$$J\left(\mathbf{w}_l\right) = \frac{1}{2}\mathbf{w}_l^T\mathbf{w}_l + \frac{1}{2}\left\|\mathbf{d}_l - V_l\mathbf{w}_l\right\|_{R_l^{-1}}^2 \tag{3.2}$$

where V_l, $\mathbf{d}_l$ and $\mathbf{w}_l$ represent V, $\mathbf{d}$ and $\mathbf{w}$ in the latent space (i.e. the one produced by the AEs) respectively.

In the literature, various designs for these deep learning techniques are used. Indeed, many parameters come into play such as the number of layers, the number of nodes per layer, the activation functions (Sigmoid, ReLU, PReLU) and the loss function used (Mean Squared Error (MSE), Binary Cross Entropy (BCE)). Better dimensionality reduction has been achieved experimentally using deep autoencoders (multiple layers) [41]. As opposed to more simplistic designs, [8] used state-of-the-art components to create the proposed CAE architecture such as complex residual blocks and attention mechanisms that proved a gain in efficiency using the MAGIC test case. Additionally, the models are readily available via an open-source API on github at: `https://github.com/julianmack/Data_Assimilation` under the MIT license. This is why [8] is taken as the starting point for this project. A succinct description of the models' components is given below for completeness.

3.2 A State-of-the-art AE

Convolution Neural Network (CCN). Although fully-connected AEs have been use to reduce dimensions, they suffer from the *curse of dimensionality* which can yield to 10^9 degrees of freedom. This means that applying large datasets to fully-connected AEs is a) computationally non-viable and b) does not exploit the structure of features in high-dimensional space [40].

CNN on the other hand provide a) local connections and b) shared weights. Weight sharing across the local input allows for detection of location-invariant features ([29] Chapter 9). CCN are based on the **convolution** operation which relates the input, **x**, the kernel, **w**, and the feature map, **s(t)** as shown below:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{3.3}$$

which can be extended to more than 1D. The dimension of the feature map can be reduced using *pooling layers* in which a single value is computed from a $n \times m$ filter using the maximum value as shown in Figure 3.2 or the average. In a network, these features maps are obtained one after the other yielding a sequence of feature maps and can be coupled with pooling operations to yield a more complex architecture as shown in Figure 3.3. The decoder has a mirrored architecture of the encoder with transposed-convolution layers or sometimes called deconvolution layers. This ensures the encoder and the decoder are both equally accurate when properly trained.

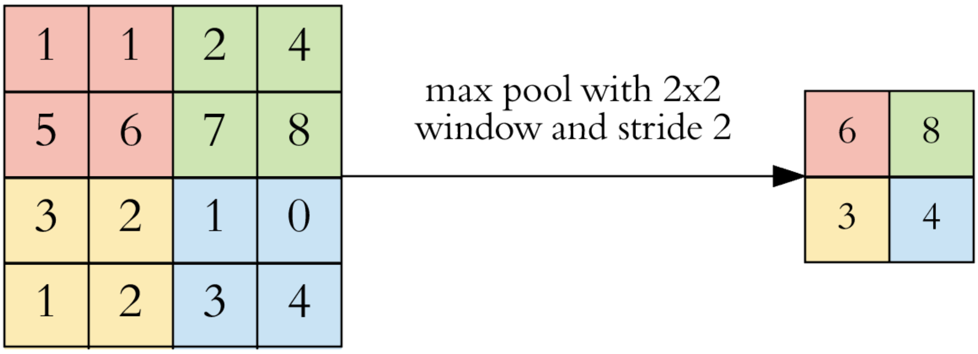


Figure 3.2: Max pooling operation to reduce the space [42].

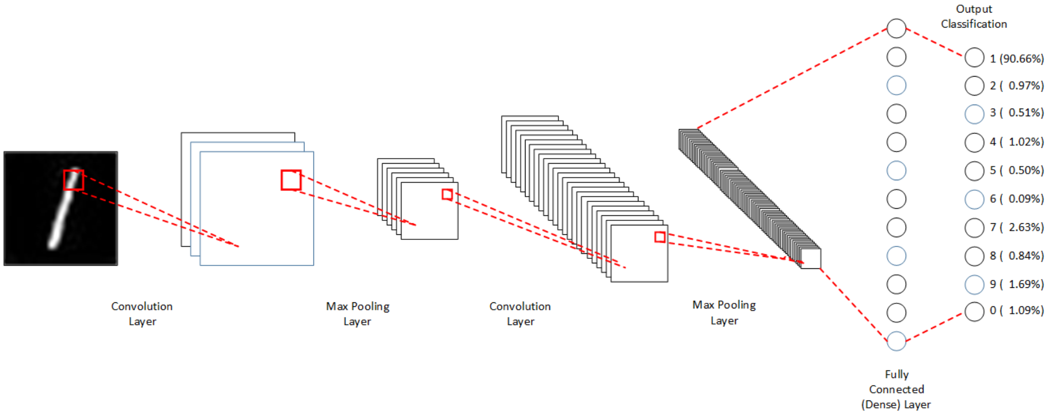


Figure 3.3: Example of a Convolutional Neural Network with multiple operations used in the case of classification [43].

Activation Functions. Activation functions insert non-linearity in the network. Multiple types of activation functions exist and have an impact on the performance of the AEs. The author in [8] investigated the performance of the most common activation functions and concluded that PReLU obtained better performances in the majority of cases during their search for an optimal CAE architecture. PRELU is defined by:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases} \tag{3.4}$$

where a is a learnable parameter. The work in [44] demonstrated PReLUs yielded better results than human performance on image classification.

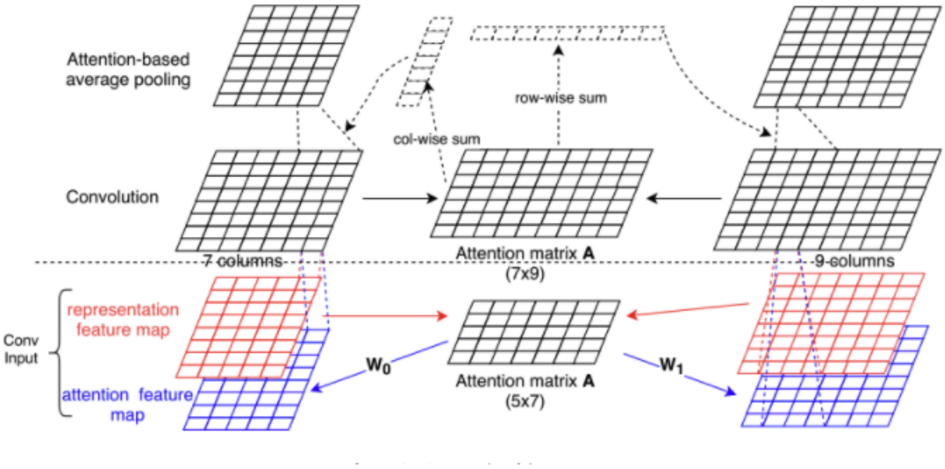


Figure 3.4: Use of attention mechanism inside a CNN architecture [45].

Complex Residual Blocks. Densely Connected Blocks and Convolutional Block Attention Modules (CBAMs) were investigated. The former ensures any feature computed by earlier residual blocks in the network is available to all the subsequent residual blocks. CBAMs exploit channel-wise and spatial attention sequentially. This makes use of an Attention mechanism that forces the network to look at relevant features on the feature map. Indeed, even if CCN exploit local correlation in the data, they assign equal importance all features. Figure 3.4 shows how an attention feature map can be introduced in a CNN architecture (before or after a convolutional feature map) and was investigated in [45].

CCN and Meshes. As Convolutional Autoencoders take equally spaced adjacent states as inputs, the author of [8] interpolated between points which meant the AEs were applied to a structured mesh. This was followed by an up-sampling procedure. The next section explores the potential to apply AEs to unstructured meshes thereby avoiding the error introduced during the interpolation process.

Chapter 4

Contribution

This project is based on the idea to use AEs on unstructured mesh to reduce the space in a DA process of a Big Data problem. In this Section we explain the main points of our approach.

4.1 Unstructured Mesh and Convolutional AE

4.1.1 Fluidity

The model used in this case is Fluidity. Fluidity is a state-of-the-art 3D non-hydrostatic fluid model where the physics behind the model is computed using control volume finite element discretisation methods. It features a mesh-adaptivity capability on unstructured meshes [46]. This means Fluidity is able to modify the accuracy of the solution within certain sensitive regions. Where small-scale physical events take place, the software will be able to provide a higher resolution solution while keeping a coarsed mesh elsewhere. Where high resolution is achieved (i.e. around the observation points), the representative errors of observations is reduced. Additionally, this provides a better picture of developing air flows whose positions are not always known a priori. Fluidity makes use of 3D space and time adaptive meshes which ensures highly active regions and precise locations of interest can be modelled accurately [47]. A visualisation is provided in Figure 4.1. This also provides a better efficiency as it reduces the total computation time [19].

4.1.2 Adaptive Mesh and CCN

The unstructured mesh output by Fluidity cannot be fed into the CAE that takes grid-like structure inputs. In [8], their approach consisted in interpolating between the points, which introduces an error, and up-sampling the data. A more precise approach would involve taking advantage of the unstructured mesh and adapting the mesh to the convolutional layers of the AE. The generalization of CNNs to meshes is not trivial but has been done for 3D face representation [30] that based their work on preliminary results by [48] [49] [50]. They propose a Convolutional Mesh

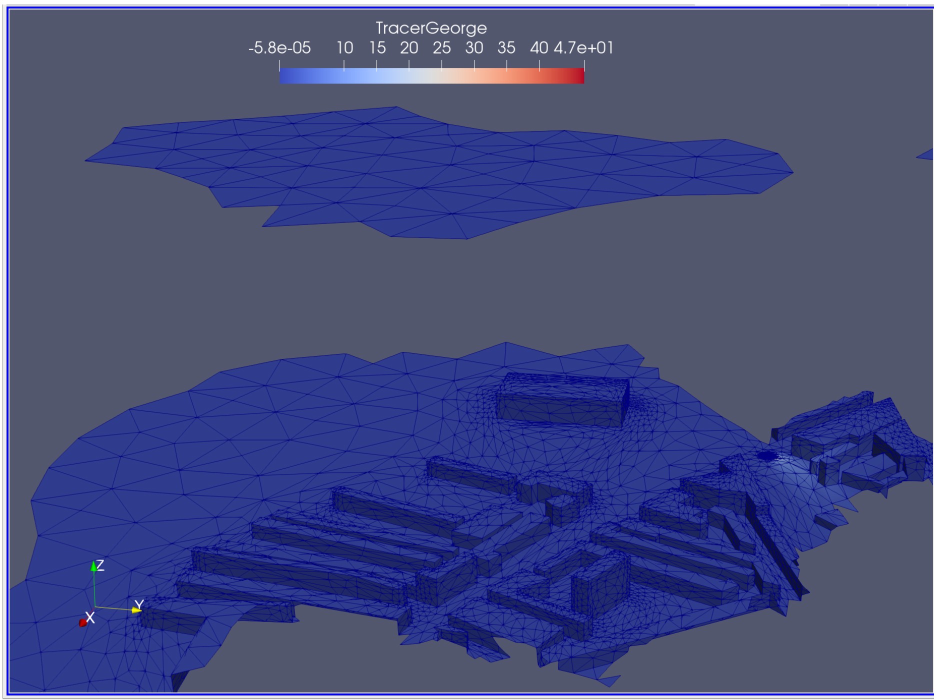


Figure 4.1: Sub-domains 6 and 8 with surface mesh illustrating Fluidity's mesh adaptivity feature.

Autoencoder (CoMA) that takes advantage of novel mesh sampling techniques. More specifically, mesh down-sampling and up-sampling layers were introduced using fast localized convolutional filters defined on the mesh. The authors concluded that their AE performed better than PCA while containing fewer parameters. It is stated that their *'representation models variations on the mesh surface using a hierarchical multi-scale approach and can generalize to other 3D mesh processing applications'*. As opposed to other works that adapted convolutions to meshes [51] [52] [53], the authors of [30] combined mesh convolutions with efficient operators to achieve a reduced dimensional space.

More recently the authors of [32] improved the technique devised in [30] and demonstrated better accuracy by implementing a pooling operations that makes use of the local mesh information.

This project will investigate the use of such a mesh adapted CAE to further reduce the space following the CVT procedure in DA for unstructured meshes. A summary of the improved procedure of [32] is detailed in the next section.

4.1.3 Unstructured Mesh Dimensionality Reduction

Mesh Simplification is achieved based on the method devised in [54] which binds together repeating edges based on *quadric matrices*. That method is slightly modified in [32] as the authors also take into account the edge length in their calculations. The total error is now defined as:

$$E = \overline{\mathbf{v}}_k^{\mathrm{T}} \overline{\mathbf{Q}}_k \overline{\mathbf{v}}_k + \lambda \max \{L_{km}, L_{kn} | m \in \mathcal{N}_i, n \in \mathcal{N}_j, m \neq j, n \neq i\} \tag{4.1}$$

where $\overline{\mathbf{v}}_k^{\mathrm{T}} \overline{\mathbf{Q}}_k \overline{\mathbf{v}}_k$ represents the initial error defined in [54] and L_{km} (resp. L_{kn}) is the edge length between vertex k and m (resp. vertex n). $\mathcal{N}_i$ (resp. $\mathcal{N}_j$) is the set of neighboring vertices of vertex i (resp. j). λ is simply a weight.

The pooling operation is defined by contracting two adjacent vertices to a new vertex as shown in Figure 4.2. This vertex (green) is the result of an averaging of the contracted vertices (red) which allows for a preservation of the mesh information.

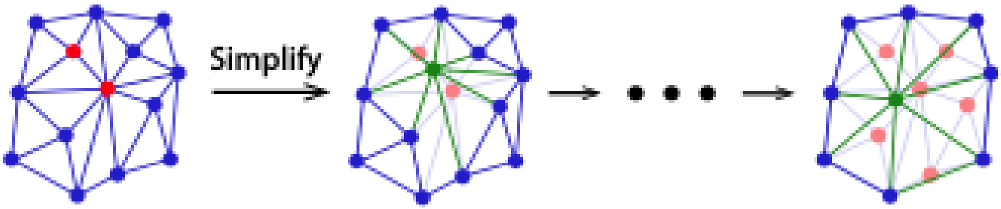


Figure 4.2: Pooling operation on meshes [32].

The lower dimensional mesh can then undergo graph convolution defined by:

$$\mathbf{y} = g_\theta(\mathbf{L})\mathbf{x} = \sum_{h=0}^{H-1} \theta_h \mathbf{T}_h(\tilde{\mathbf{L}})\mathbf{x} \tag{4.2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output matrices respectively. Each row corresponds to a vertex and each column to a feature dimension.

This methodology makes use of Chebyshev polynomials, $\mathbf{T}_h(\tilde{\mathbf{L}}) \in R^{V \times V}$, that allow for faster computation [48]. $\theta \in R^H$ is the polynomial coefficients. $\tilde{\mathbf{L}}$ denotes the normalized Laplacian matrix that is an essential operator in spectral graph analysis. In the Fourier basis, it is diagonal and can represent operations such as filtering. $\tilde{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{max}} - \mathbf{I}$.

Their network, composed of a Variational Mesh AE applied to graphs, combined with new pooling and de-pooling operations was trained using mean squared error (MSE) and performed better than state-of-the-art methods. The authors conclude that their network is highly generalizable. This provides an opportunity to explore the use of AEs directly on unstructured meshes in the context of DA.

4.2 Big Data Problem: Domains and Sub-domains

Adaptive Domain Decomposition Data Assimilation (ADD-DA) combines the Domain Decomposition implemented in Fluidity with the DA model in a simple way which reduces the total computation time. The authors of [19] demonstrated that it was possible to deal with DA of sub-domains instead of the full domain without any loss of information.

If we let the whole domain be represented by $\mathcal{P}(\Omega) = \{\Omega_i\}_{i=1,\dots,s}$ where $\Omega = \{x_j\}_{j=1,\dots,n}$ are discrete spatial domain obtained using adaptive domain decomposition, the incremental ADD-DA formulation becomes:

$$\mathbf{w}_i^{ADD-DA} = \arg\min_{w_i} J_i(\mathbf{w}_i) \tag{4.3}$$

$$\mathbf{w}_i^{ADD-DA} = \arg\min_{w_i} \left(\frac{1}{2}\alpha \mathbf{w}_i^T \mathbf{w}_i + \frac{1}{2}\alpha \left(\mathbf{H}\mathbf{V}_i\mathbf{w}_i - \mathbf{d}_i\right)^T \mathbf{R}_i^{-1} \left(\mathbf{H}\mathbf{V}_i\mathbf{w}_i - \mathbf{d}_i\right) \right) \tag{4.4}$$

where it can be shown that the second term of equation 4.4 is equivalent to that of equation 2.17 using $||\mathbf{a}||^2_{\mathbf{A}^{-1}} \equiv \mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{a}$.

Now the full domain (Figure 4.4) is decomposed, we only consider the most relevant sub-domains. More relevant in this case means sub-domains where the pollution measurements were non-zero for all the time steps available. The author of [9] proved that for our Test Case (detailed in Section 4.3) only sub-domains 6 and 8 are important. An example is shown in Figure 4.3 for the tracer *George* at time step 0 compared to the full domain shown in Figure 4.4. Although these sub-domains (6 and 8) provide the most information, the pollution measurements are very close to zero which leads to a sparse dataset [9].

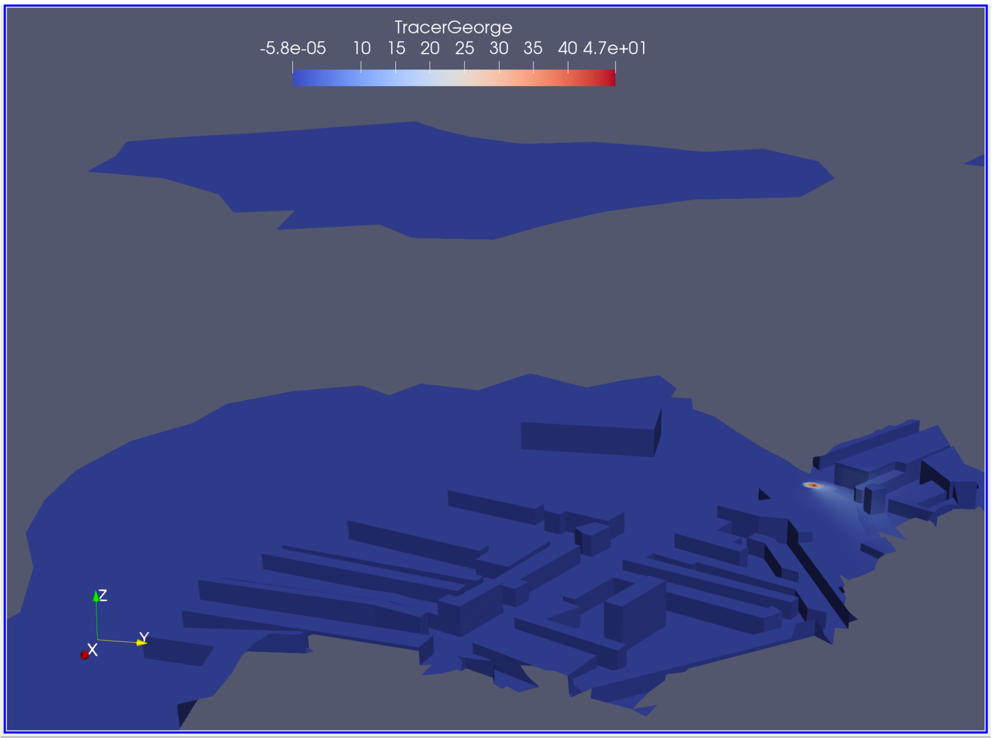


Figure 4.3: Sub-domains 6 and 8 at time step 0 displaying the data for the *George* tracer.

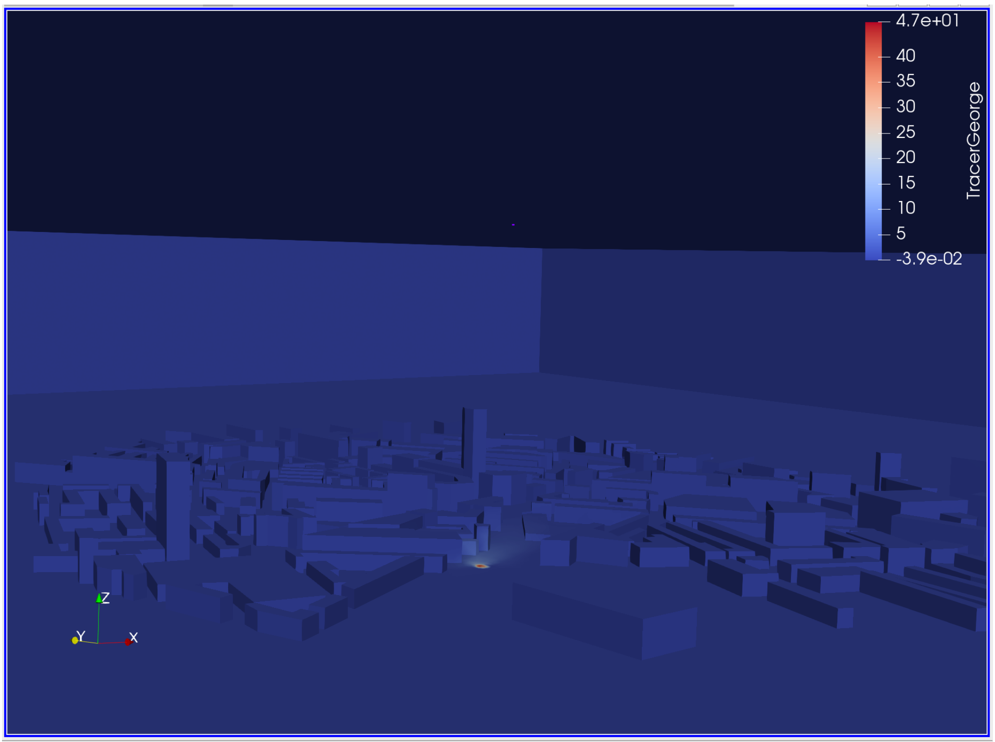


Figure 4.4: Entire domain at time step 0 displaying the data for the George tracer.

Our methodology will be tested on each sub-domain independently. If successful, this means a Big Data problem can be addressed using unstructured convolutional AEs for Data Assimilation by working with sub-domains.

4.3 Test Case

This method will be applied to the data obtained by the MAGIC project. The overall test site represents 14 buildings in an urban environment near London South Bank University (LSBU) in London, UK. The pollution tracers were obtained from sensors placed at the MAGIC test site and computed with the CFD software Fluidity. The domain was decomposed into 10 and 32 sub-domains using Fluidity to allow for parallel execution. The data comprises of 537 time-steps for pollution tracers. These measurements were divided into 9 regions based on location. We will concentrate on the tracer *George* as this was the data used in [9] to identify the most relevant sub-domains.

Sub-domain 6 represents a domain size of 327 x 460 x 250 (meters)

Sub-domain 8 represents a domain size of 806 x 749 x 250 (meters).

The dispersion of the pollution is described by the classic advection-diffusion equation such that the concentration of the pollution is seen as a passive scalar (Equation 4.5).

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (\bar{\kappa} \nabla c) + F \tag{4.5}$$

where $\bar{\kappa}$ represents the diffusivity tensor (m^2/s) and F the source terms ($kg/m^3/s$) (i.e. the pollution generated by a source point). A more detailed description of the Fluidity software implementation can be found in [55] and [56]. The pollution background is represented by a sinusoidal function given by:

$$C(t) = \frac{1}{2}\left(\sin\left(\frac{2\pi t}{T}\right) + 1\right) \tag{4.6}$$

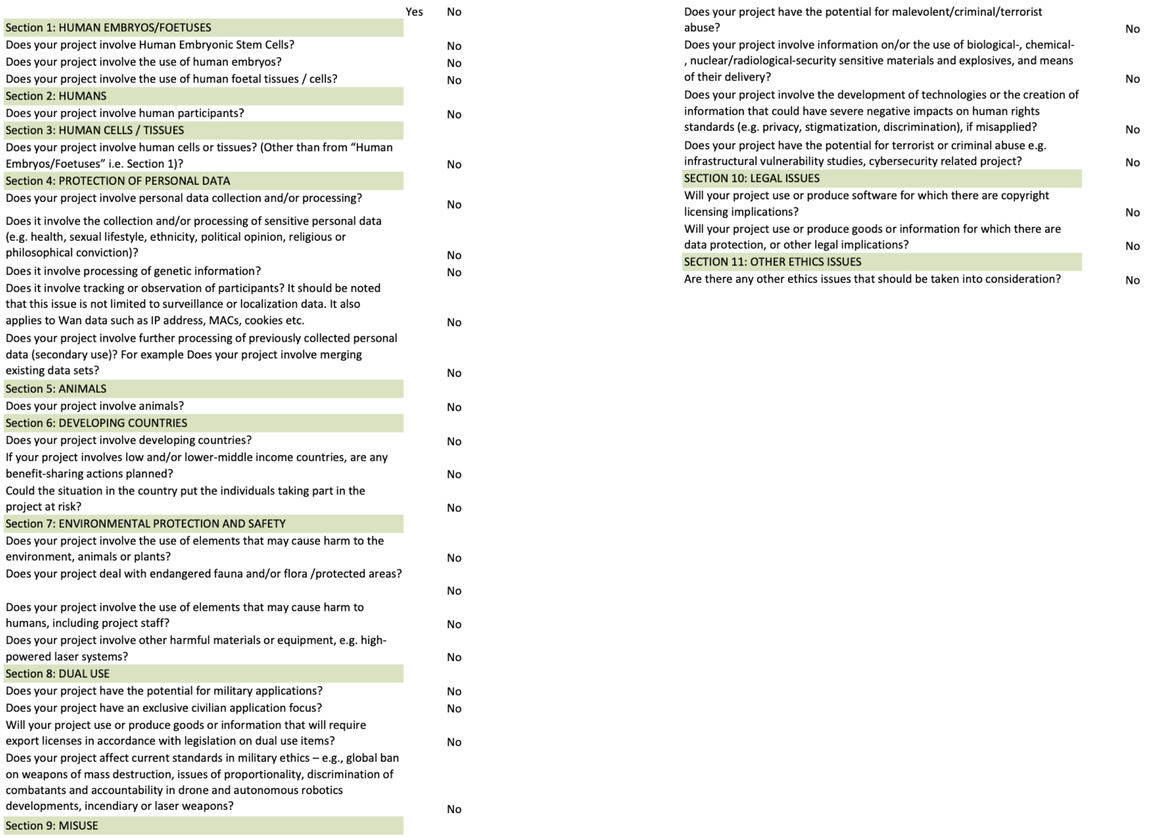
where C, t and T represent the pollution concentration, the time and the period in seconds, respectively. The background pollution equates to the waves of pollution in an urban environment while the tracers are point sources located at traffic intersections and illustrate pollution in a traffic congested junction [57].

4.4 Project Progress

The work done to date involved defining the boundaries of the project and the methods that will be investigated. This required an understanding of the past literature, the mathematical frameworks used and some of the softwares for this project. Following the identification of the relevant techniques that will be used in this project, this report was elaborated in order to give the understanding required to follow the future work that will achieved in the coming months.

A first investigation of the dataset was made in order to characterize the pollution measurements that will be used to test this project's methodology.

An initial consideration of the ethics checklist was done to ensure my understanding of what to consider when working with the data and softwares throughout the duration of the project. This is shown in Figure 4.5. The checklist will be regularly examined to ensure that it provides an accurate representation of the considerations made in a well documented manner.



	Yes	No
Section 1: HUMAN EMBRYOS/FOETUSES		
Does your project involve Human Embryonic Stem Cells?		No
Does your project involve the use of human embryos?		No
Does your project involve the use of human foetal tissues / cells?		No
Section 2: HUMANS		
Does your project involve human participants?		No
Section 3: HUMAN CELLS / TISSUES		
Does your project involve human cells or tissues? (Other than from “Human Embryos/Foetuses” i.e. Section 1)?		No
Section 4: PROTECTION OF PERSONAL DATA		
Does your project involve personal data collection and/or processing?		No
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?		No
Does it involve processing of genetic information?		No
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.		No
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?		No
Section 5: ANIMALS		
Does your project involve animals?		No
Section 6: DEVELOPING COUNTRIES		
Does your project involve developing countries?		No
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?		No
Could the situation in the country put the individuals taking part in the project at risk?		No
Section 7: ENVIRONMENTAL PROTECTION AND SAFETY		
Does your project involve the use of elements that may cause harm to the environment, animals or plants?		No
Does your project deal with endangered fauna and/or flora /protected areas?		No
Does your project involve the use of elements that may cause harm to humans, including project staff?		No
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?		No
Section 8: DUAL USE		
Does your project have the potential for military applications?		No
Does your project have an exclusive civilian application focus?		No
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?		No
Does your project affect current standards in military ethics – e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?		No
Section 9: MISUSE		
Does your project have the potential for malevolent/criminal/terrorist abuse?		No
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?		No
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?		No
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?		No
SECTION 10: LEGAL ISSUES		
Will your project use or produce software for which there are copyright licensing implications?		No
Will your project use or produce goods or information for which there are data protection, or other legal implications?		No
SECTION 11: OTHER ETHICS ISSUES		
Are there any other ethics issues that should be taken into consideration?		No

Figure 4.5: Initial Ethics Checklist for the Project

4.5 Future Work and Timeline

As well as continuing to deepen my understanding, in the next month I will be implementing the technologies discussed above for the test case. Following the initial implementation, I will be testing different scenarios to evaluate the methodology devised in this project. In August I will start writing the final project report to ensure enough time is available to produce a complete and clear document. Provided the proposed methodology yields positive results, incorporating it in a Data Assimilation process or in an Adaptive Domain Decomposition scheme could be considered.

Chapter 5

Conclusion

This project will address the opportunity to use Autoencoders on unstructured meshes to conduct Data Assimilation in a Big Data problem. This will mean the possibility to complement the CVT method and reduce the space further. It could make the overall scheme more efficient without the need to truncate information as is presently the case with the use of PCA or TSVD techniques. The test case represents pollution measurements in Big Cities which can be viewed as a Big Data problem. We make use of Domain Decomposition techniques to work on sub-domains independently which means this methodology could be used in combination with Domain Decomposition to solve a Big Data problem.

Bibliography

[1] J. Lelieveld et al. "The contribution of outdoor air pollution sources to premature mortality on a global scale". In: *Nature* 525.7569 (2015), p. 367. ISSN: 0028-0836.

[2] Esteban Ortiz-Ospina Max Roser Hannah Ritchie and Joe Hasell. "Coronavirus Pandemic (COVID-19)". In: *Our World in Data* (2020). URL: https://ourworldindata.org/coronavirus.

[3] Rodolfo. author Guzzi. *Data Assimilation: Mathematical Concepts and Instructive Examples*. eng. 2016.

[4] Rossella Arcucci et al. "Optimal reduced space for Variational Data Assimilation". eng. In: *Journal of Computational Physics* 379 (2019), pp. 51–69. ISSN: 0021-9991.

[5] Rossella Arcucci et al. "On the variational data assimilation problem solving and sensitivity analysis". In: *Journal of Computational Physics* 335 (2017), pp. 311–326. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.01.034. URL: http://www.sciencedirect.com/science/article/pii/S0021999117300505.

[6] F. Rabier et al. "The ECMWF implementation of three-dimensional variational assimilation (3D-Var). II: Structure functions". In: *Quarterly Journal of the Royal Meteorological Society* 124.550 (1998), pp. 1809–1829. DOI: 10.1002/qj.49712455003. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49712455003. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49712455003.

[7] Dan Gabriel Cacuci, Ionel Michael Navon, and Mihaela Ionescu-Bujor. *Computational methods for data evaluation and assimilation*. CRC press, 2013.

[8] Julian Mack. "Attention-based convolutional Autoencoders for 3D-Variational Data Assimilation". MA thesis. Imperial College London, 2019.

[9] Tolga Hasan Dur et al. "Weak Constraint Gaussian Processes for optimal sensor placement". eng. In: *Journal of Computational Science* 42 (2020). ISSN: 1877-7503.

[10] A. C. Lorenc. "Analysis methods for numerical weather prediction". In: *Quarterly Journal of the Royal Meteorological Society* 112.474 (1986), pp. 1177–1194. DOI: 10.1002/qj.49711247414. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49711247414. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49711247414.

[11] Erik Andersson et al. "The ECMWF implementation of three dimensional variational assimilation. Part III: Experimental results". In: 243 (Jan. 1998), p. 36. DOI: 10.21957/k75n9aqa. URL: https://www.ecmwf.int/node/7752.

[12] Rawlins FR et al. "The Met Office Global 4-Dimensional Data Assimilation Scheme". In: *Quarterly Journal of the Royal Meteorological Society* 133 (Jan. 2007), pp. 347–362. DOI: 10.1002/qj.32.

[13] C. Lee et al. "Demonstration of ensemble data assimilation for Mars using DART, MarsWRF, and radiance observations from MGS TES". eng. In: *Journal of Geophysical Research. Planets* 116.E11 (2011). ISSN: 21699097. URL: http://search.proquest.com/docview/907844162/.

[14] M.J Martin et al. "Status and future of data assimilation in operational oceanography". eng. In: *Journal of Operational Oceanography: GODAE OceanView Part 1* 8.sup1 (2015), s28–s48. ISSN: 1755-876X. URL: http://www.tandfonline.com/doi/abs/10.1080/1755876X.2015.1022055.

[15] Eric J Kostelich et al. "Accurate state estimation from uncertain data and models: an application of data assimilation to mathematical models of human brain tumors". eng. In: *Biology direct* 6.1 (2011), p. 64. ISSN: 1745-6150.

[16] F. Bouttier and P. Courtier. *Data Assimilation concepts and methods*. Meteorological Training Course Lecture Series. Mar. 1999.

[17] Massimo Bonavita. *Overview of Data Assimilation Methods*. ECMWF. Training Lecture Notes. 2019.

[18] Stephen A Haben. "Conditioning and preconditioning of the minimisation problem in variational data assimilation". PhD thesis. University of Reading, 2011.

[19] Rossella Arcucci et al. "Adaptive Domain Decomposition for Effective Data Assimilation". In: ().

[20] R. N. Bannister. "A review of operational methods of variational and ensembleâvariational data assimilation". In: *Quarterly Journal of the Royal Meteorological Society* 143.703 (2017), pp. 607–633. ISSN: 0035-9009.

[21] Stephen Penny et al. "A Hybrid Global Ocean Data Assimilation System at NCEP". eng. In: *Monthly Weather Review* 143.11 (2015), pp. 4660–4677. ISSN: 00270644. URL: http://search.proquest.com/docview/1732126085/.

[22] Massimo Bonavita et al. "The evolution of the ECMWF hybrid data assimilation system". In: *Quarterly Journal of the Royal Meteorological Society* 142.694 (2016), pp. 287–303. ISSN: 0035-9009.

[23] Luisa DâAmore et al. “A Scalable Approach for Variational Data Assimilation”. In: *Journal of Scientific Computing* 61 (Nov. 2014). DOI: 10.1007/s10915-014-9824-2.

[24] P. Courtier, J.-N. Thepaut, and A. Hollingsworth. “A strategy for operational implementation of 4D-Var, using an incremental approach”. In: *Quarterly Journal of the Royal Meteorological Society* 120.519 (1994), pp. 1367–1387. DOI: 10.1002/qj.49712051912. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49712051912. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49712051912.

[25] Andrew C. Lorenc. “Development of an Operational Variational Assimilation Scheme (gtSpecial IssueltData Assimilation in Meteology and Oceanography: Theory and Practice)”. In: *Journal of the Meteorological Society of Japan. Ser. II* 75.1B (1997), pp. 339–346. DOI: 10.2151/jmsj1965.75.1B_339.

[26] R. N. Bannister. “A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics”. In: *Quarterly Journal of the Royal Meteorological Society* 134.637 (2008), pp. 1971–1996. DOI: 10.1002/qj.340. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.340. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.340.

[27] Yunhai Xiao, Zengxin Wei, and Zhiguo Wang. “A limited memory BFGS-type method for large-scale unconstrained optimization”. In: *Computers Mathematics with Applications* 56.4 (2008), pp. 1001–1009. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2008.01.028. URL: http://www.sciencedirect.com/science/article/pii/S0898122108001028.

[28] Arden Dertat. “Applied Deep Learning - Part 3: Autoencoders”. In: (2017), Chapter 2.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[30] Anurag Ranjan et al. *Generating 3D faces using Convolutional Mesh Autoencoders*. 2018. arXiv: 1807.10267 [cs.CV].

[31] Qingyang Tan et al. “Mesh-based autoencoders for localized deformation component analysis”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[32] Yu-Jie Yuan et al. “Mesh Variational Autoencoders with Edge Contraction Pooling”. In: *arXiv preprint arXiv:1908.02507* (2019).

[33] Qingyang Tan et al. “Variational autoencoders for deforming 3d mesh models”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5841–5850.

[34] M. Wang et al. “Deep Learning-Based Model Reduction for Distributed Parameter Systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46.12 (2016), pp. 1664–1674.

[35] Kelvin Loh, Pejman Shoeibi Omrani, and Ruud Linden. “Deep Learning and Data Assimilation for Real-Time Production Prediction in Natural Gas Wells”. In: (Feb. 2018).

[36] Sandeep Reddy Bukka, Allan Ross Magee, and Rajeev Kumar Jaiman. “Deep Convolutional Recurrent Autoencoders for Flow Field Prediction”. In: *arXiv* (2020), arXiv–2003.

[37] X. Deng et al. “Deep Principal Component Analysis Based on Layerwise Feature Extraction and Its Application to Nonlinear Process Monitoring”. In: *IEEE Transactions on Control Systems Technology* 27.6 (2019), pp. 2526–2540.

[38] G. Dimitriu and N. Apreutesei. “Comparative study with data assimilation experiments using proper orthogonal decomposition method”. In: vol. 4818. 2008, pp. 393–400. ISBN: 3540788255.

[39] Yanhua Cao et al. “A reduced order approach to four dimensional variational data assimilation using proper orthogonal decomposition”. eng. In: *International Journal for Numerical Methods in Fluids* 53.10 (2007), pp. 1571–1583. ISSN: 0271-2091.

[40] Francisco J. Gonzalez and Maciej Balajewicz. *Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems*. 2018. arXiv: 1808.01346 [math.DS].

[41] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[42] Arden Dertat. *Applied Deep Learning - Part 4: Convolutional Neural Networks*. Nov. 2017. URL: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#9722.

[43] Tjeng Wawan Cenggoro, Tjeng Wawan Cenggoro, and Tjeng Wawan Cenggoro. *Deep Learning for Image Classification with Keras: Step by Step Tutorial*. URL: https://research.binus.ac.id/airnd/2018/03/deep-learning-for-image-classification-with-keras-step-by-step-tutorial/.

[44] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: 1502.01852 [cs.CV].

[45] Wenpeng Yin et al. *ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs*. 2015. arXiv: 1512.05193 [cs.CL].

[46] C.C. Pain et al. “Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations”. In: *Computer Methods in Applied Mechanics and Engineering* 190.29 (2001), pp. 3771–3796. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(00)00294-2. URL: http://www.sciencedirect.com/science/article/pii/S0045782500002942.

[47] Juan Du et al. “Ensemble data assimilation applied to an adaptive mesh ocean model”. In: *International Journal for Numerical Methods in Fluids* 82.12 (2016), pp. 997–1009.

[48] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in neural information processing systems*. 2016, pp. 3844–3852.

[49] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In: *arXiv preprint arXiv:1312.6203* (2013).

[50] Michael M Bronstein et al. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[51] Ayan Sinha, Jing Bai, and Karthik Ramani. "Deep learning 3D shape surfaces using geometry images". In: *European Conference on Computer Vision*. Springer. 2016, pp. 223–240.

[52] Li Yi et al. "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2282–2290.

[53] Haggai Maron et al. "Convolutional neural networks on surfaces via seamless toric covers." In: *ACM Trans. Graph.* 36.4 (2017), pp. 71–1.

[54] Michael Garland and Paul S Heckbert. "Surface simplification using quadric error metrics". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997, pp. 209–216.

[55] R. Ford et al. "A Nonhydrostatic Finite-Element Model for Three-Dimensional Stratified Oceanic Flows. Part I: Model Formulation". In: *Monthly Weather Review* 132.12 (2004), pp. 2816–2831. DOI: 10.1175/MWR2824.1. eprint: https://doi.org/10.1175/MWR2824.1. URL: https://doi.org/10.1175/MWR2824.1.

[56] Imperial College London AMCG. "Fluidity manual v4.1.12". In: (Apr. 2015). DOI: 10.6084/m9.figshare.1387713.v2. URL: https://figshare.com/articles/Fluidity_Manual/1387713.

[57] César Quilodrán Casas, Rossella Arcucci, and Yi-Ke Guo. *Urban Air Pollution Forecasts Generated From Latent Space Representations*. ICLR. 2020.