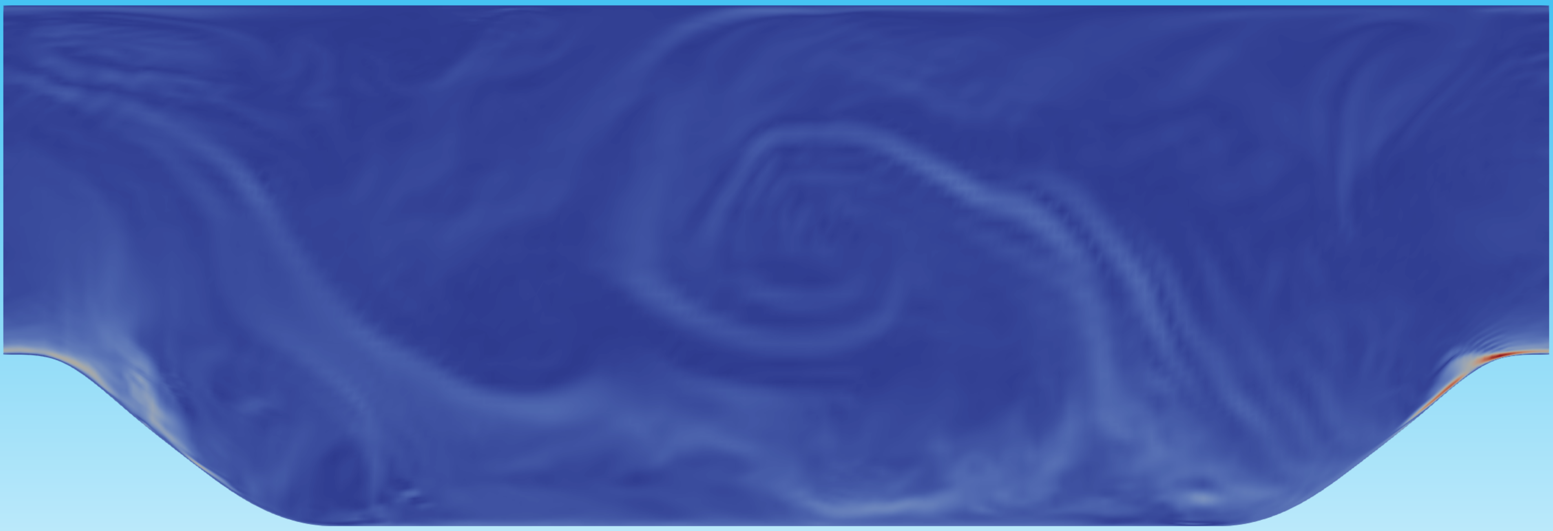
Output error estimation using super-resolution neural networks

A functional-independent approach to adjoint-based error estimation

Thomas Hunter



TU Delft

Output-error estimation using super-resolution neural networks

A functional-independent approach to adjoint-based error estimation

by

Thomas P. Hunter

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday October 14, 2021 at 14:00 PM.

Student number:	4567250	
Project duration:	March 1, 2021 – October 14, 2021	
Thesis committee:	Dr. S. J. Hulshoff,	TU Delft, supervisor
	Prof. Dr. S. Hickel,	TU Delft
	Dr. N. A. K. Doan,	TU Delft
	Dr. E. van Kampen,	TU Delft

Preface

This thesis started without true understanding of the topic I would be researching for the coming months. I had expressed the wish to research topics related to mesh adaptation, however, I had very little knowledge on the topic and was looking forward to making this topic my own. The inability to share this research experience with peers in the facilities of the university made the experience more individualistic, but I am very happy to have been able to find support in the people close to me and my supervisor in our weekly meetings.

I want to thank Dr. Steven Hulshoff for his dedication and support throughout the duration of the thesis. I am delighted that I could share this experience with him, under his supervision in a graduation project for the second time of my young academic career. I have enjoyed the process throughout, from the proposition of research topics, to the final discussion of the results and review. His invaluable feedback during our productive weekly updates greatly helped the course of the thesis. I look forward to starting my time as a PhD candidate, in the AIFluids lab of the Aerospace Faculty at the TU Delft under the supervision of of Dr. Anh Khoa Doan and Dr. Davide Modesti. I have truly enjoyed doing research in both my time as an intern at ONERA and this thesis, and aided by Dr. Steven Hulshoff's contagious enthusiasm for research, I am anticipating this next step in my professional and academic career.

This thesis would not have been possible without the continuous support from my friends and the great time we have shared. I thank you for being for having made my degree a memory I will look very fondly back upon. I would like to thank my girlfriend for her everlasting patience and support when having to listen to my rambling about the thesis. Lastly, I thank my friends and family back in France and Australia for the support during my Bachelor and Master's degrees.

Thomas P. Hunter
Delft, October 7, 2021

Abstract

Numerical simulations and optimisation methods, such as mesh adaptation, rely on the accurate and inexpensive use of error estimation methods. Output-error estimation is the most accurate method; however, it relies on the use of approximations in order to be implemented in practice. The proposed method in this thesis relies on the use of super-resolution neural networks to reconstruct the fine adjoint solution from a computed coarse adjoint solution. The proposed method is compared to reference error estimators on an unsteady Burgers' equation using the method of manufactured solutions, as well as a lid-driven cavity flow. For both of these test cases, it was shown that super-resolution neural networks were able to reconstruct the fine adjoint solution and provide robust and inexpensive output-error estimates at the cost of lower accuracy. Nonetheless, the accurate estimation of the error indicators gives great confidence in the proposed method's ability to perform similarly to the adjoint-weighted residual output-error estimate with a mesh adaptation procedure. A cost metric for the computational overhead of the output-error estimate is proposed. This highlights the superior performance of the lower up-scale ratio super-resolution neural networks due to their higher accuracy and lower computational cost than those with higher up-scaling factors.

Contents

List of Figures

Nomenclature

Abbreviations

AMR Adaptive Mesh Refinement

ANN Artificial Neural Network

ATC Adjoint Transpose Convection

AWR Adjoint-Weighted Residual

BAMG Bi-dimensional Anisotropic Mesh Generator

CAD Computer Aided Design

CCM Computational Cost Metric

CFD Computational Fluid Dynamics

cGAN conditional Generative Adversarial Network

CNN Convolutional Neural Network

CPU Central Processing Unit

DNS Direct Numerical Simulation

DoF Degrees of Freedom

FCNN Fully Convolutional Neural Network

FEM Finite-Element Method

FSI Fluid-Structure Interaction

FVM Finite-Volume Method

GAN Generative Adversarial Network

HIT Homogeneous Isotropic Turbulence

LES Large Eddy Simulation

MMS Method of Manufactured Solution

MSE Mean-Square Error

MUMPS MUltifrontal Massively Parallel Sparse direct Solver

NASA National Aeronautics and Space Administration

PDE Partial Differential Equation

POD Proper Orthogonal Decomposition

QoI Quantity of Interest

RANS Reynolds Averaged Navier-Stokes

RNN Recurrent Neural Network

ROM Reduced Order Models

SGS Sub-Grid Scale

SRNN Super-Resolution Neural Network

SVD Singular Value Decomposition

Symbols - Latin

δJ Output error

f Source term

g Quantity of interest weight function

H Coarse characteristic mesh size

h Fine characteristic mesh size

$(.)_C$ Nearest-neighbour variable interpolation

$(.)_L$ Linear variable interpolation

$(.)_{TL}$ Tri-linear variable interpolation

$(.)_{TQ}$ Tri-quadratic variable interpolation

J Functional

$J'[u_0](\delta u)$ Quantity of interest sensitivity to an arbitrary perturbation δu linearised about the solution u_0

L Differential operator

$R_h(.)$ Strong residual on the fine mesh h

$r'[u_0](\delta u)$ Strong residual sensitivity to an arbitrary perturbation δu linearised about the solution u_0

u Velocity field solution

u_h Discretised velocity field solution

Symbols - Greek

$\partial\Omega$ Computational domain boundary

ϵ_e Error indicator at element e

η Adaptation parameter

ν Kinematic viscosity

Ω Computational domain

Ψ Adjoint vector

θ Snapshot

1
Introduction

1.1. Mesh adaptation

Computational Fluid Dynamics (CFD) has become a vital tool of the modern engineer in the analysis of fluid flows. The decreasing computational cost of numerical simulations in the past decades has enabled the widespread use of CFD from the aerospace industry to healthcare [12]. However, the growing need for ever more accurate and complex fluid dynamic analyses has meant that the use of CFD has remained very specialised and user-intensive. This is emphasised by the large investment required design and analysis tasks on a regular basis. In particular, large transient turbulent problems, such as Large Eddy Simulation (LES), are affected by prohibitively large computational costs limiting their use as a design tool.

Numerical simulations require the generation of a discrete computational support, comprised of cells or elements, that is used to solve fluid equations specific to the problem, for example the Navier-Stokes equations. An increase in the number of elements constituting this computational domain increases the accuracy of modeling of the phenomena of interest. However, this also results in an increase in the cost required to solve the simulation. Moreover, the elements in the computational domain do not contribute equally to the inaccuracy of a simulation. Thus, the computational cost could be reduced by adapting a number of elements only in the regions where it will help the simulation to become more accurate.

Mesh adaptation aims to tackle this issue by maximising the accuracy of the solution of a numerical simulation for a given computational cost. Although the user is free to define the meaning of accuracy for a CFD simulation, the problem is usually solved with respect to an a priori chosen Quantity of Interest (QoI). In the domain of CFD, this can be lift or drag for example. The maximisation of the accuracy can thus be described as an optimisation problem, this is then referred to as goal-oriented mesh adaptation. A key component of this optimisation problem aims at identifying the local contribution of each element to the error in the discrete solution of the QoI. This error in QoI is referred to as the output error. This then enables the correct determination of the optimal number and size of elements to minimise the output error.

1.2. Output error estimation

The main reasons for output errors are the presence of numerical errors and the lack of modeling fidelity, both geometrical and physical [62]. As the physical modeling induced errors result from the choice in fluid flow equations and corresponding models, such as turbulence models, they are independent from the discrete computational support. However, numerical errors and errors resulting from the lack of geometric fidelity can be both reduced by the use of more appropriate spatial and temporal numerical discretisation.

Numerous methods have been devised to estimate the output error. This thesis focuses on the most accurate, which is gradient-based output error estimation, also referred to as adjoint-based error estimation. This relies on the solution of a sensitivity problem associated to the fluid flow equations, the adjoint problem. It is also referred to as the dual problem in literature, in opposition to the primal fluid flow problem. Despite its high accuracy, the linearity of the adjoint problem, and strong mathematical foundations in the determination of the output error, the associated computational cost remains too high for the widespread adoption of adjoint-based error estimation and, thus, goal-oriented mesh adaptation.

The computational overhead required for the use of adjoint-based error estimation stems from the cost of the adjoint solution and the amount of information required to be stored for the solution of unsteady adjoint problems. On the one hand, the former affects primarily steady problems and reflects the need to solve the adjoint problem on a finer computational domain than the fluid equations computational domain. While in practice this is not usually done, it is approximated by the use of prolongation operators. These operators aim to partially reconstruct the fine solution, but also introduce inaccuracies in the error estimation.

On the other hand, the amount of storage required for the solution of the unsteady adjoint problem is a direct result of the adjoint problem representing the sensitivity of a problem. Just as the solution at a given time-step depends on the preceding time-step, the adjoint solution must be solved backwards in time in order to determine the sensitivity of a solution to the discretisation of the computational domain. This means that the instantaneous primal solutions must be stored during the forward solution of the CFD simulation, and recalled during the solution of the backwards in time adjoint problem. This leads to extremely large storage requirements, with current methods writing the primal to memory and loading the instantaneous primal at each time-step, slowing down the solution of the adjoint.

1.3. Optimisation of output error estimation methods

In order to tackle the two main issues mentioned in the preceding section, a number of methods have been proposed to optimise the output error estimation process. Among the methods aimed at reducing the adjoint solution cost, is the use of Artificial Neural Networks (ANN) in order to bypass the solution of the adjoint altogether [17, 77, 81]. This can be achieved by training an algorithm to reconstruct the adjoint solution from the primal solution. However, since the adjoint solution depends on the QoI, this method is fully dependent on the QoI and would require as many Artificial Neural Network (ANN) as QoI, which is limiting in practice.

Another approach consists in mapping the local error contributions to the output error directly from the primal solution [26], making use of ANN to regress non-linear functions too. Neural network-based methods have also included the proposition to compute the error and guide the mesh adaptation process directly from the adjoint solution, without resorting to the use of error estimation [5, 34, 44, 52]. While these methods have been implemented successfully, they remain very specialised, do not generalise well, and behave as 'black-boxes'.

Methods aimed at reducing the storage footprint of the primal solution include the use of Reduced Order Models (ROM) [46] or specifically chosen sub-sampling methods[50, 74]. The former reduces the size of the solution to a superposition of simpler solutions, while the latter represents the solution by points which are judged to be representative of the primal solution. The limitations for these approaches are the inaccuracies introduced by the reconstruction of the primal solution. This in turn can result in inaccurate output error estimation due to an erroneous adjoint solution.

1.4. Research need and outline

Thus, the need for an accurate and cheap adjoint-based output error estimation method is identified. With the advent, and more widespread adoption, of machine learning, it is natural to apply it to output-based mesh adaptation in order to reduce the computational overhead introduced by the estimation of the output error. However, many of the proposed approaches aimed at reducing the overhead computational cost of output-based error estimation using ANN compromise on either the accuracy or the reduction of computational cost.

The proposition in this thesis aims at potentially addressing the two raised computational cost and storage footprint issues through the use of low resolution primal and adjoint solutions, which are solved numerically and then augmented through the use of Super-Resolution Neural Network (SRNN). This would enable the storage of coarser primal solutions, both spatially and temporally [20, 29, 75]. The potential for the use of super-resolution in this regard has been indicated by its application to the reconstruction of turbulent flow information [29, 39, 76]. This has been done successfully for rather simple problems such as Homogeneous

Isotropic Turbulence (HIT) but also attempted for more complex problems with more limited success. The main goal of these studies was to capture information on small scale turbulence such that turbulence models based on these new insights can be devised. They have shown that the energy cascade can be respected using ANN driven super-resolution while more traditional up-sampling methods such as bi-cubic splines have a tendency to be too dissipative [48]. This is a very desirable property when the application of a SRNN to mesh adaptation is considered since the modeling error can be minimised. Moreover, it could also help alleviate the computational cost of the adjoint solution by enabling the solution of the adjoint on a coarse mesh and reconstructing the fine adjoint solution from this coarser reference one. This has the potential to also generalise across QoI and problems since the reconstruction step can be de-correlated from the QoI or problem if the sole reconstruction task is learned.

Due to the super-resolution of the primal having been implemented and shown to be successful in reconstructing the fine primal fields, the primary research objective of the thesis was formulated as:

Research objective

Can super-resolution neural networks enable the accurate reconstruction of adjoint solution for use in output-based error estimation in mesh adaptation?

Using the research objective defined in the previous chapter and the findings in this chapter on related methods aimed at optimising the output-based error estimates, secondary research objectives can be defined. These objectives, shown below, help in testing the hypothesis and forms the framework for the work performed in this thesis. The secondary objectives originate from the need to answer certain questions in order to evaluate the feasibility and performance of using SRNN for output-based error estimation.

1. Evaluate the state-of-the-art output error estimators.
2. Evaluate the choice of the adjoint fine space.
3. Evaluate the state-of-the-art SRNN applied to the adjoint field.
4. Evaluate SRNN in an adjoint-based error estimation method

The first item intended to draw a reference point with current error estimation methods, namely the formulations of [31, 70]. This first effort enables the differences in the methods to be highlighted and the implications they would have on the mesh adaptation routine. Based on this, a choice is formulated for the implementation of a specific error estimation method for the more complex test case.

The second item refers to the difference between the current practice of using higher order elements in the finite element context, and the h-refinement of the fine space. The possibility of p-refined fine spaces does not exist in finite volume discretisation, nor can it be used with the current proposal. It is thus necessary to quantify the consequence of using a finer grid resolution for the fine space, both in terms of accuracy and computational cost.

The evaluation of the capacity of SRNN to reconstruct adjoint fields is then performed. By gaining further understanding of the architecture of the SRNN and the training process, one is able to make informed decisions to maximise its ability to fulfil the primary research objective.

Finally, the last item covers the integration of the proposed method in the actual output error and mesh adaptation routine. The integration must be evaluated in order to be able to propose a framework which is viable and could potentially solve the need from which the research objective arises.

The report aims at providing a relevant example of the capacity of such an approach. As a result, 1D and 2D problems were tackled, ranging from steady to unsteady, and laminar to turbulent. Moreover, previous attempts at using machine learning for the task of predicting the output error and adaptation parameter from the primal solution [17, 19], have uncovered the potential of SRNN to correctly reconstruct the adaptation parameter but not the output error. For this reason, the tests for the hypothesis will focus on the estimation of the output error.

The application to 2D turbulent problems can be tested in the near future since the framework has been created in OpenFOAM. This enabled the identification of certain issues which are addressed in the concluding chapter of this thesis. This implementation has not yet been validated or thoroughly tested but highlights key issues which require further research.

The report's opening chapter synthesises the main developments in mesh adaptation in the past decades. It gives an overview of the current best practices in the domain and highlights the subsequent need for a cheaper approach to estimating output error. Chapter 3 explores in greater depth the best practices with respect to output error estimation and introduces the reference output error estimates used in this thesis. The various research paths used to tackle the limitations are also presented. This leads to the more detailed description of the proposed method as well as the formulation of the secondary research objectives. Chapter 4 validates the implementation of the output error estimates in the FEniCS problem solving environment. The 1D unsteady Burgers' equation test case, documented in chapter 5, describes the implementation and training of the SRNN and evaluates the loss in accuracy of the proposed method as well as the reduction in computational cost of the proposed method. A second test case is proposed where chapter 6 explores the implementation of the proposed method to a lid-driven cavity flow. Finally, chapter 7 closes the report by formulating the various conclusions of the report and the recommendations for the future work to be conducted in order to fully evaluate the proposed method.

2
Mesh Adaptation: Description and Context

In two- and more importantly three-dimensional problems, an increase in resolution leads to the exponential growth in the number of elements, and is referred to as the curse of dimensionality. Mesh adaptation proposes to help alleviate this problem by increasing the resolution of the solution solely where the contribution to the global error is significant. The more widespread adoption of mesh adaptation throughout the aerospace industry is one of the main goals of the coming decade, pushed by National Aeronautics and Space Administration (NASA) [67] and other research institutes [3].

This chapter will introduce mesh adaptation in more depth and introduce the need for computationally cheap and accurate error estimation in both the context of mesh adaptation and CFD more globally. Section 2.1 describes the steps of mesh adaptation. The following sections will explore each step individually. Lastly, the closing section of this chapter will attempt to identify the key integration points of the proposed method to the state-of-the-art mesh adaptation procedures.

2.1. Mesh adaptation steps

Mesh adaptation consists of the steps that an engineer naturally undertakes when attempting to reduce the error in a simulation. In the first of these steps, the CFD problem is solved. Its solution allows conclusions to be made on the current level of error. If it is not satisfactory, one attempts to localise the regions where the error is too large and modify the computational support in these locations. The result is an algorithm such as the one in Figure 2.1.[2, 25, 28, 32]

The steps in mesh adaptation can thus be decomposed as: solution, error estimation, error localisation, adaptation selection and mesh modification. The process can be complex and costly, as a result many different approaches have been proposed to automate it and render it more efficient [3, 9]. The vast majority of the research conducted in this area appears under the banner of Adaptive Mesh Refinement (AMR), however, the term **mesh adaptation** is preferred in this thesis to highlight the importance of also identifying unnecessary mesh refinement and to coarsen the computational support where required.

2.2. Error estimation

There exists two main approaches to estimating the error in a given numerical simulation. The first, known as **metric-based** error estimation, uses features in the flow solution to drive the adaptation routine. These features are described by appropriately chosen metrics. They can either describe a physical variable in the flow, such as the $\mathscr{H}^2$ semi-norm as proposed by Löhner [50], or a chosen variable whose goal is to describe more abstract concepts, such as the wall coordinate. The metric can also seek to represent the target the rate of change of flow variables in space, which is achieved by the use of the Hessian. Metric-based error estimation is computationally cheap but can be extremely inefficient in reducing the output error in the measure of a specific QoI, such as lift, which are not necessarily correlated to the resolution of local flow features [28, 69].

Moreover, the metrics used are dependent on the problem and the chosen QoI to be effective. This entails detailed a priori knowledge of the problem being studied. This implies a lack of generalisation that is sometimes tackled using more complex metrics [9, 57]. The performance of metric-based adaptation has been

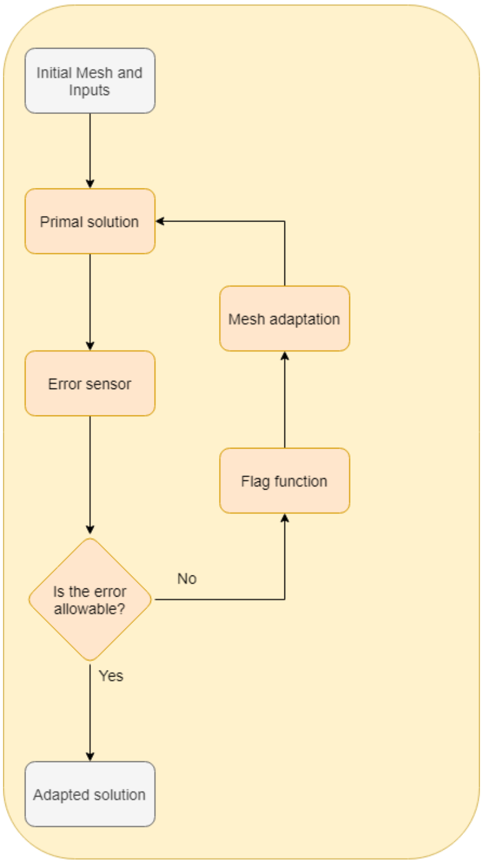


Figure 2.1: Mesh adaptation workflow

assimilated to heuristics, which makes it difficult to justify the added overhead computational cost of using more complex metrics [27]. This is also valid for the more advanced metrics such as dissipation or entropy-based adaptation where the sources of dissipation and entropy generation are identified. For very specific problems these approaches can work well, but they are less efficient than goal-oriented mesh adaptation and do not generalise well.[7]

On the other hand, **output-error estimation** is specifically designed to reduce the error in the prediction of the QoI. Output-based error estimation can be performed by estimating numerical errors or through the solution of the variational problem. In the former, the error related to the discretisation of the problem is targeted. It was shown that they are equivalent to gradient based methods, such as the adjoint-weighted residual, albeit less accurate but also less computationally intensive. These have found applications in LES in particular where the cost of the adjoint solution remains too large [24, 79].

Adjoint-based error estimation is more involved to set up in the first place since it requires the derivation of associated variational problems and is also more computationally expensive [28]. However, it is more accurate and results in the need for coarser meshes and as a result a smaller computational cost since it is able to converge to a QoI value more rapidly. The associated variational problem is referred to as the **adjoint** problem and can be derived from the continuous equations or discretised forms of the problem considered.

This approach enables to effectively predicting the output error and localising the sources of error, and will be the main focus of this thesis. Chapter 3 explores output error prediction methods in greater depth as well as identifying their key limitations justifying the need to develop tools that reduce the computational overhead and induced storage requirements.

2.3. Adaptation parameter

Once the error in the QoI is determined, one must localise the sources of error in order to modify the computational support and reduce the output error. This is performed through the use of an adaptation parameter, also referred to in literature as an **error indicator** or error sensor. This step is essential in the mesh adaptation process but only limited literature is found on the topic.

It is common for error indicators to be derived from the local value of the error estimate and normalised. An interesting area of research consists in the concatenation of the adaptation parameter, mesh adaptation strategy and the mesh mechanics. Such a proposed method is higher-dimensional embedding. This approach makes use of a specific set of parameters that map the metric from the given simulation space into a uniform mesh in higher dimension. This mesh is then projected back into the simulation space leading to an anisotropic mesh.

2.4. Mesh adaptation strategy

This section aims at giving an overview of the some of the mesh adaptations strategies which have been proposed. The first is the use of **rescaling methods**. Arguably the simplest method covered here, yet used throughout, it aims to bound the maxima and minima of the adaptation parameter values in order to prevent undesired results in the mesh adaptation process. It usually makes use of the relation shown in Equation 2.1, in order to map the relative importance of each element to the output error. In this equation, η corresponds to an adaptation parameter, whose aim is to localise the importance of an element relative to the output error. This operation is very sensitive to outliers and can increase the concentration of elements in the centre of the interval. This can be solved through the use of additional regression operations aiming at obtaining a chosen error sensor distribution. However, this leads to additional computational overhead which is not desired. Rescaling methods can be used in combination with other strategies in order to determine the elements to be modified in the mesh adaptation process.[65]

$$\eta^* = \frac{\eta - \eta_{min}}{\eta_{max} - \eta_{min}} \tag{2.1}$$

The **fixed fraction** mesh adaptation strategy differs from the previous method as it identifies a chosen number of cells for the mesh adaptation process. This approach addresses certain issues with rescaling since it is no longer affected by outliers. It also parts with the value of the error sensor and instead only focuses on the ordered values of the error sensor. The main drawback of this method is its inflexibility. It is not able to adapt to widely varying error sensor distributions and can impartially choose to adapt certain cells but not others when both have the same error indicator value.[55]

In an attempt to resolve the limitations observed through the use of these these and other strategies, **constant threshold adaptation** was developed. This method targets all the elements whose adaptation parameter is not within chosen bounds. The main disadvantage is that it can, on the one hand, lead to the adaptation of a very restricted number of cells resulting in a larger number of mesh adaptation iterations. On the other hand, it can attempt to target too large a number of cells which also makes the remeshing step of the adaptation routine very inefficient.

Finally, the **decreasing threshold** approach has been proposed in order to combine the advantages of the two previous methods. As in the fixed fraction strategy, a constant number of cells is first targeted. This enables the identification of the outliers and lower the output error by refining the elements contributing the most to the error in the first steps. Once this becomes less effective, the threshold triggering the adaptation can be moved in order to modify other cells. This prevents the algorithm from indefinitely attempting to lower the error contribution from certain elements which are insensitive to refinement or coarsening. [55]

2.5. Mesh adaptation mechanics

Many meshing mechanics have been proposed and used in literature. Various distinctions can be made in the approach they use and the reasons they are used. The main goal of these methods is to introduce non-uniformity, which can be achieved through global or local meshing.

Here, a distinction must be made between uniform and isotropic meshes, in opposition to non-uniform and anisotropic meshes. A uniform mesh is constituted of elements, of the same size throughout such as in a Cartesian mesh; whereas an isotropic mesh has elements of the same shape whereas the size can be heterogeneous.

Moreover, a combination of different meshing mechanisms have been used, where the order in which they were applied was also important [32, 72]. Since this further complicates the ability to compare different mesh adaptation frameworks, the scope was limited to the study of output-based error estimation on uniform meshes at different refinement levels.

2.5.1. Global meshing

Among meshing methods, global remeshing approaches have been used extensively throughout literature. These aim at generating a new mesh which complies to certain criteria at every adaptation iteration. This is affordable for simple domains but has limited use in more complex domains where the meshing process produces a non-negligible computational overhead when compared to the cost of solving the primal and/or dual problem.

Metric-driven meshing Remeshing can be based on a number of metrics or scalar values deduced from the error indicator field. A popular approach using global metric-driven remeshing is found in the BAMG [35], MMG [23] and refine-EGADS [33] tools. These make use of two scalars for each element which determine the meshing process. The determination of these scalars is the main issue in the implementation of this approach[26]. The first scalar is an indicator of the element's elongation and the second its orientation.

These scalars are then fed in the meshing algorithms which will use a triangulation method to comply to these constraints provided in the scalar fields. A disadvantage of this method is that in addition to the computational overhead of the creation of the mesh characterisation scalar fields, the entire mesh is modified at each adaptation iteration. However, these meshing methods have the advantage of being highly flexible, to allow the conservation of the geometry and boundary conditions as well as being able to produce highly anisotropic meshes.

Higher-dimensional embedding In a method borrowed from the meshing of Computer Aided Design (CAD) geometries, higher-dimensional embedding has been proposed as an alternative to the other tools presented above [21, 22]. Making use of an isotropic mesh in the higher dimensional space, projected into the computational domain, it enables the fast generation of meshes of different refinement levels. The complexity arises from the choice of higher dimensional space, for instance velocity gradients are proposed by [13–16]. This leads to similar outcomes to metric-based error estimates albeit at a lower cost due to the reduced meshing costs for complex problems.

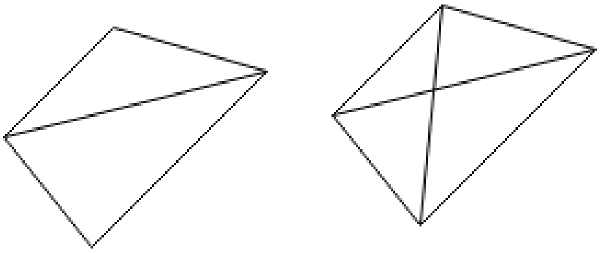


Figure 2.2: Example of edge refinement

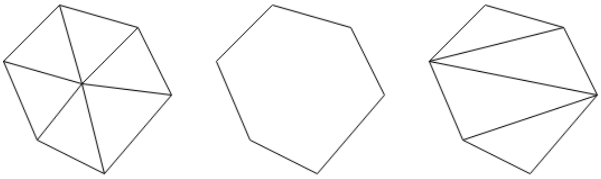


Figure 2.3: Example of edge coarsening

2.5.2. Local meshing methods

To avoid remeshing the computational domain at every adaptation iteration, local meshing methods have also been proposed. The first approach consists of evaluating the adaptation parameter at one or a few neighbouring cells with the goal of equidistributing the error indicator locally. The second consists of removing the element or group of elements and to locally remesh the newly empty space, referred to as a cavity.

Local methods Local mesh methods have been preferred historically since they produce little computational overhead and are similar to the approach an engineer would perform when reacting to the error sensor. Local methods can be split into three categories: h-adaptation, change of nodal connectivity and the r-adaptation [32]. p-adaptation can be singled out as being specific to finite element methods and changes the polynomial order the element used to describe the solution locally.

***h*-adaptation** refers to both refinement or the coarsening of elements. Examples of refinement and coarsening can be found in Figure 2.2 and Figure 2.3 respectively. Special attention should be given to edges where boundary conditions are applied in order to ensure the problem remains well-defined. The most common example found of this method is applied to Cartesian meshes under the names of quadtree in 2D and octree in 3D.

In the case of a **change in nodal connectivity**, elements are concatenated and assembled differently. Criterion have been proposed to drive the application of this method such as the Heron formula proposed in [32]. The risk with this method is that in a non-Euclidean metric, the resulting elements could have a negative shape criterion. The topology of the elements must also be checked to ensure that the resulting elements do not hinder the ability to solve the primal problem. An example of an edge swapping is given in Figure 2.4.

Finally, ***r*-adaptation** can be performed in a few ways. The method illustrated in Figure 2.5 makes the analogy of the edges being a network of springs whose stiffness can be changed in order to move the centre node and as such introduce anisotropy.[6]

Local remeshing-cavity method The last local operation explored in this section is the use of the cavity method. In this approach, the elements in the vicinity of the elements identified by the adaptation parameter are removed. The created cavity is then remeshed automatically either through the use of a Delaunay triangulation, an advancing front or neural network. This method is a compromise between the global meshing

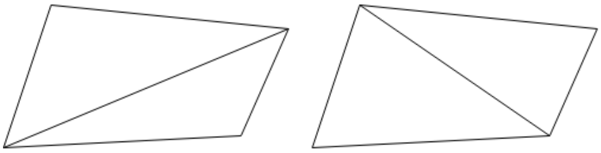


Figure 2.4: Example of edge swapping

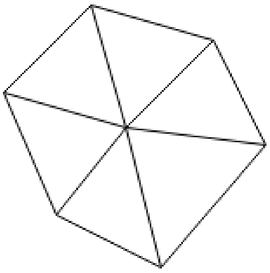


Figure 2.5: Example of r-adaptation

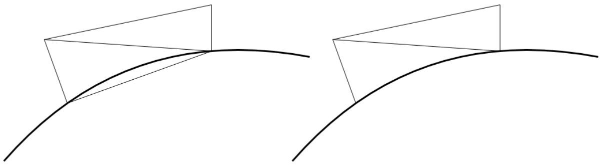


Figure 2.6: Example of a cut-cell

and the local methods and has not been documented extensively. [14, 61, 63]

2.5.3. Cut-cell methods

Cut-cell methods aim at conserving accurate geometrical modelling by using the description of the geometry to shape the neighbouring elements by cutting them as shown in Figure 2.6. The advantage is that the geometry is conserved and limits some of the workarounds which need to be made around the boundary conditions. However, it introduces other problems related to the creation of negative volume and self-intersecting cells. This approach is mostly used in Fluid-Structure Interaction (FSI) problems but has shown the ability to produce good results in mesh adaptation problems, in particular for elements of geometry with large curvature. [80]

2.6. Synthesis and research direction

This chapter details the current state-of-the-art in mesh adaptation with the current developments observed and used in order to formulate the thesis from the opening chapter. The identification of a need for cheap and accurate output-based error estimation for unsteady problems is the main argument for the proposal to introduce SRNN. However, it is important to integrate the proposed method in its context of mesh adaptation from the prototyping and implementation stages. For instance, in the analysis of the output error and the error indicators, the potential consequences on the mesh adaptation strategy and remeshing methods should be considered.

Since the SRNN is designed to reconstruct the fine adjoint from a coarse adjoint solution, the adaptation parameter is here defined as the element output error estimate contribution. However, it is likely that the reconstruction of the adjoint can lead to inaccuracies, in particular in regions near boundaries where certain flow variables can be prescribed. For this reason, large outliers can be expected. This prevents the use of the fixed fraction adaptation strategy, since it is then likely that the same elements are refined, not based on their exact error indicator but based on the inaccurate reconstruction.

Constant threshold or decreasing threshold would be more relevant as this would enable the adaptation of a larger range of elements, on top of the outliers. This is most likely to successfully lead to a reduction in output error.

Due to the nature of SRNN, it is expected that h-adaptation is also the most effective mesh adaptation mechanic. This results from the inability of ANN to represent higher order polynomial elements as would p-adaptation would require. Moreover, the ANN can make use of structured data in order to facilitate the training process [82]. This makes the proposed approach compatible with local remeshing methods, and the combination with cut-cell methods, would enable both geometric fidelity and the conservation of the data structure. This helps formulate the future work to be conducted in order to fully integrate the method proposed in this thesis in a mesh adaptation procedure.

3
Output-based Error Estimation: Description and Limitations

This chapter explores in greater depth the definition and implementation of output-based error estimation methods. Section section 3.1 describes the derivation of the error estimate as well as the corresponding error indicators while section 3.2 describes in greater detail the origin of the limitations of the current implementations of the output-based error estimators.

3.1. Output-based error estimation

Output-based error estimation relies on the evaluation of the output sensitivity to the numerical solution of the CFD problem, also referred to as the **primal** problem. This is obtained through the solution of the adjoint problem to the CFD problem. Various review papers explore this adjoint problem and its mathematical properties in more depth, see [3, 6, 28, 38] to name a few.

3.1.1. Output error

This thesis employs the continuous adjoint for output-based error estimation, and this section will explore the derivation used to obtain the adjoint set of equations from the primal CFD problem. The derivation follows the steps detailed in [38]. The discrete adjoint method will solely be mentioned here as being an alternative which relies on the analysis of the sensitivity of the discretised primal problem to the design variables [27, 54, 74].

The derivation of the continuous adjoint equation starts with the definition of the continuous residual $r(u)$, which is equal to zero when the primal Partial Differential Equation (PDE) is satisfied. This leads to the relation in Equation 3.1.

$$\mathbf{r}(\mathbf{u}) = L\mathbf{u} - \mathbf{f} \tag{3.1}$$

The adjoint operator L^* is defined as the relation between the primal operator and an adjoint set of solutions Ψ through the relation $(L\mathbf{u}, \Psi) = (L^*\Psi, \mathbf{u})$. The adjoint solution can then defined as the sensitivity of a chosen output $J(u)$ with respect to an infinitesimal perturbation in the primal residual requiring:

$$J'(\delta\mathbf{u}) = \int_\Omega \Psi^T \mathbf{r}'(\delta\mathbf{u})\, d\Omega \ \ , \forall\, \delta\mathbf{u} \tag{3.2}$$

Equation 3.2 is referred to as the generalised form of the continuous adjoint equation. Deriving the primal residual sensitivity $\mathbf{r}'(\delta\mathbf{u})$, substituting in Equation 3.2 and performing the integration by parts of each term of the primal PDE enables the derivation of the adjoint operator L^*. The boundary conditions are then determined by associating the various known boundary conditions from the primal problem to the boundary condition terms obtained through integration by parts.

This thesis also considers non-linear problems. For these, the relation Equation 3.2 must be modified to evaluate the sensitivity of the primal residual sensitivity linearised about a given state u_0. This is written as:

$$J'[u_0](\delta\mathbf{u}) = \int_\Omega \Psi^T \mathbf{r}'[u_0](\delta\mathbf{u}) d\Omega \ \ , \forall \delta\mathbf{u} \tag{3.3}$$

Unsteady problems include time variation in both the integral of the change in output and the residual. Due to the discrete time-stepping scheme used here and the application to both finite-volume and finite-element methods, all of the time integrals of the change in output will be approximated using the discretised sum.

The derivation for the error estimation can be found in [38] and results in Equation 3.4, which can be approximated using Equation 3.5 with $\mathbf{u}_h^H$ being the coarse solution injected into the fine space.

$$\delta J_{est} = J_h(\mathbf{u}_h) - J_H(\mathbf{u}_H) \tag{3.4}$$

$$\delta J \approx \delta J_{est} = -\int_\Omega \Psi_h^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega \tag{3.5}$$

Equation 3.5 is referred to as the adjoint-weighted residual approach since the non-zero residual induced by the truncation error is weighted by the adjoint solution. It must be noted that for non-linear problems, a linearisation error is introduced, of order $\mathcal{O}(\delta\mathbf{u}^2)$. This also means that non-linear problems can be far more sensitive to under-resolved solutions of the primal, since this linearisation error can dominate when $\delta\mathbf{u}^2$ is large enough.

All the methods tested in this thesis attempt to approximate Equation 3.5 by alleviating the need to solve the adjoint in the fine space through the use of various prolongation operators aiming at projecting the solution on a coarse space into a fine space.

A common approach when using these prolongation operators is to expand Equation 3.5 into two terms: the computable error and the remaining error. This is shown in Equation 3.6. [31]

$$\underbrace{\delta J_{est}}_{\text{Error}} \approx \underbrace{-\int_\Omega (\Psi_h^H)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Computable Error}} + \underbrace{\int_\Omega (\Psi_h^H - \Psi_h)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Remaining Error}} \tag{3.6}$$

The remaining error term can also be formulated with respect to the adjoint residual as per [71] and shown in Equation 3.7.

$$\underbrace{\delta J_{est}}_{\text{Error}} \approx \underbrace{-\int_\Omega (\Psi_h^H)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Computable Error}} + \underbrace{\int_\Omega (\mathbf{R}_h^\Psi (\Psi_h^H)^T (\mathbf{u}_h - \mathbf{u}_h^H) d\Omega}_{\text{Remaining Error}} \tag{3.7}$$

Finally, a linear combination of both previous formulations can be used. This is found in [28].

3.1.2. Adaptation Parameter

The output error having been estimated, it is necessary to localise the error contribution. This arises naturally from the discrete integration of Equation 3.5. In Equation 3.8, N_H is the number of coarse elements, N_e^k is the number of fine elements comprised in the coarse element. The adjoint-weighted residual is thus integrated over each fine element and the sum over each element results in the output error estimate.

$$\begin{aligned} \epsilon_k &= \sum_{e=1}^{N_e^k} -\int_{\Omega_e} \Psi_{h,e}^T \mathbf{R}_{h,e}(\mathbf{u}_h^H) d\Omega_e \\ \delta J_{est} &\approx \sum_{k=1}^{N_H} \epsilon_k \end{aligned} \tag{3.8}$$

The local error indicator, in the fine space, is then taken to be:

$$\epsilon_e = \left| \int_{\Omega_e} \Psi_{h,e}^T \mathbf{R}_{h,e}(\mathbf{u}_h^H) d\Omega_e \right| \tag{3.9}$$

The coarse space error indicator can be obtained by summing the fine space error indicators for all the fine space elements inside the coarse element. Similar to the construction of the error indicators ϵ_k in the coarse space, error indicators can be constructed using the various decompositions of the output error into computable error and remaining error. They are referred to in this thesis as the **primal residual** form and the **adjoint residual** form. The former requires fewer operations since the primal residual can be reused from the computable error term. On the contrary, the latter requires the additional construction of the fine space adjoint system for the evaluation of the fine space adjoint residual.

However, the adjoint residual error indicator, as shown in Equation 3.12, has been shown to be more robust, even more so when used in a linear combination with the primal residual form as in Equation 3.10. Fidkowski and Darmofal [28] indicate that while the computable error could be used as an error indicator itself, it performs poorly, whereas the remaining error converges at a higher order rate. In practice, and in particular for finite volume applications, it is chosen to adapt on the remaining error, using the formulation from [55] in Equation 3.11, despite having to include the computable error in the evaluation of the output error. [80]

$$\epsilon_e = \frac{1}{2}\left|\int_{\Omega_e} (\Psi_h^H - \Psi_h)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega_e\right| + \frac{1}{2}\left|\int_{\Omega_e} \mathbf{R}_h^{\Psi}(\Psi_h^H)^T (\mathbf{u}_h - \mathbf{u}_h^H) d\Omega_e\right| \tag{3.10}$$

$$\epsilon_e = \left|\int_{\Omega_e} (\Psi_h^H - \Psi_h)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega_e\right| \tag{3.11}$$

$$\epsilon_e = \left|\int_{\Omega_e} \mathbf{R}_h^{\Psi}(\Psi_h^H)^T (\mathbf{u}_h - \mathbf{u}_h^H) d\Omega_e\right| \tag{3.12}$$

3.2. Limitations of output-based error estimation

From the formulations of the output error and error indicators above, a few limitations are observed. Since the cost of solving either the adjoint or the primal on a fine space is too large, the remaining error in the output error formulation must be approximated. This, in practice, is achieved through the use of prolongation operators, whose goal is to reconstruct the fine space variable using an interpolation of the coarse variable onto the fine space. Venditti and Darmofal [71] propose the use of the difference between a (tri-)quadratic projection and a (tri-)linear projection, Equation 3.13, whereas Nemec et al.[55] propose the use of the difference between a linear projection and a constant value projection, Equation 3.14.

$$\epsilon_Q = \left|\int_{\Omega_e} (\Psi_{TQ} - \Psi_{TL})^T \mathbf{R}_h(\mathbf{u}_L) d\Omega_e\right| \tag{3.13}$$

$$\epsilon_L = \left|\int_{\Omega_e} (\Psi_L - \Psi_C)^T \mathbf{R}_h(\mathbf{u}_L) d\Omega_e\right| \tag{3.14}$$

Due to these approximations, the performance of the error estimator in both the output error and the error indicator is hindered. Moreover, these operations, in particular the least-squares reconstruction of the quadratic interpolation, make use of a smoothness assumption which is not valid near discontinuities, a particular area of interest for mesh adaptation in CFD. The physical meaning of the solution is also not guaranteed. Finally, the least-squares problem can become ill-posed which further limits the generality of this approach.

A method aimed at addressing these issues is the use of a few iterations on the coarse adjoint solution injected into the fine space. This usually ensures the physicality of the solution while avoiding the full solution of the adjoint in the fine space[28]. However, this comes at the computational cost of having to evaluate and iterate functions in the fine space, which can be prohibitive.

An issue raised briefly in the preceding section is the potential instability of the adjoint equation. This stems from the apparition of a term which represents the convection of the adjoint field with the primal field. This results in a weak coupling of the adjoint and its derivative which can resonate and lead to the oscillation and divergence of the solution, as well as physical viscosity being destabilising backwards in time. Various approaches have been considered [36, 37, 51], however, they rely on the introduction of additional dissipation in order to curb the growth of the sources of instability. This dissipation can either be formulated in a similar manner to a turbulence model, in a stabilisation term, or through the introduction of additional numerical dissipation. The latter prevents the accurate reconstruction of sharp gradient and led to sub-optimal output-based error estimation [37].

The chosen approach in this thesis consists in identifying the precise sources of instability in the Adjoint Transpose Convection (ATC) term and to dampen them [37]. These terms ensure that the ATC remains diagonally dominant and unconditionally stable based on an eigenvalue analysis. Moreover, the higher order discretisation schemes can be used in these regions of turbulence generation, instead of resorting to lower order more dissipative discretisation schemes.

Finally, since the solution of the unsteady adjoint problem is backwards in time, the storage of the primal solution at every time-step is required. In practice this is achieved in two ways, either the primal is solved in batches which are recalled when the primal information is required for the solution of the adjoint, or a checkpoint method is used.

3.3. Proposed method

The method proposed in this thesis revolves around the reconstruction of the fine adjoint solution Ψ_h from a coarse adjoint solution Ψ_H, which is solved numerically. The fine adjoint definition used in this thesis is the solution of the adjoint solution on a mesh whose refinement corresponds to the super-resolution factor of the SRNN. For example, the fine adjoint solution for a 2x SRNN is the adjoint solution in a once uniformly refined coarse mesh. This definition is consistently applied to the reference output error estimates.

The reconstructed fine adjoint is substituted in Equation 3.5 in order to create the adjoint-based error estimate:

$$\delta J_{SRNN} = -\int_{\Omega} \Psi_{NN}^{T} R_h(\mathbf{u}_h^H) d\Omega \tag{3.15}$$

The error indicators are then defined in a similar manner as in the preceding section by taking the local integral of the adjoint-weighted residual over a coarse element e.

$$\epsilon_{NN,e} = \left| \int_{\Omega_e} \Psi_{NN,e}^{T} \mathbf{R}_{h,e}(\mathbf{u}_h^H) d\Omega_e \right| \tag{3.16}$$

Based on this last relation, one can see that the computational cost savings would be significant compared to the solution of the fine adjoint. However, compared to the practical implementations of the fine adjoint, the cost savings would be lower. The proposed method would outperform the state-of-the-art implementations in its accuracy of the reconstruction of the fine adjoint solution.

It is also possible to predict an increase in the computational cost of the output-based error estimate when using higher up-sampling factors. The error estimate require the evaluation of the error indicators on a finer mesh for higher up-sampling factors, resulting in a larger number of error indicators to evaluate. Moreover, SRNN decreases in accuracy with larger up-sampling factors, resulting in a double penalty when using too high up-sampling factors. This will be analysed in the following chapter, where the impact of the refined space choice will be quantified during the validation of the output-based error estimates.

4

Validation of the Reference Error Estimators

This chapter presents various cases used to validate the implementation of the reference output error estimation methods. This chapter is given particular attention since it came to light that, despite the many attempts in the implementation of the reference error estimators, peculiarities in the chosen problem solving environment led to undesired behaviours. The test cases in chapter 5 and 6 will introduce the use of SRNN for the adjoint-based error estimation.

The chosen problem solving environment is FEniCS[1]. Its high-level interface in both python and C++ makes it very attractive for the prototyping of numerical tools. Particular attention should be given to the interpolation between function spaces of different polynomial orders, which is not accurate in the release used (2019.1) and prevented the comparison of p-refined and h-refined fine adjoint solutions.

A particularity of the implementation relies on the method used to evaluate the strong residual since it is not trivial in FEniCS. Each derivative had to be projected back to the source function space in order to maintain continuity over the element faces. Not resorting to this additional projection led to FEniCS attempting to evaluate the gradient at the element faces which is inherently ill-defined for discontinuous elements function spaces obtained when the gradient of a piece-wise continuous function is considered for instance.

The validation of the error estimation method centres about the use of the Method of Manufactured Solution (MMS) for systems of PDE. This approach takes the reverse approach to the solution of equations in order to evaluate the accuracy of a numerical solver, by prescibing a solution and finding the corresponding PDE. Here, two cases, both one-dimensional, are presented for the validation of the method, as well as a third, two-dimensional, used as verification for the generalisation to more complex and higher dimensional solutions. [17, 45, 56, 66]

The first test is the solution of the **Poisson equation**, also known as the Laplace equation. This equation is used to model diffusive systems, as such it is found in the modelling of the diffusion of heat within materials or the diffusion of kinetic energy to heat in viscous flows. It is one of the components of the Navier-Stokes equations, hence its use here.

The second system used for the validation of the method is the **steady Burgers' equation**. It is constructed using both the diffusion term of the Poisson equation as well as an additional non-linear advection term. This deceptively simple equation can be used in its unsteady form to model the propagation of discontinuities, such as shockwaves, in viscous flows.

Finally, a two-dimensional scalar advection-diffusion equation, from [18], is used. This type of equation is used to model the transport of dissipation of a variable in a flow. For instance, it is the most simple model used for the transport and dissipation of pollution in the atmosphere [43]. This more complex problem also enables the identification of the first limitations of the reference error estimator implementations and justifies certain choices made in the analysis of the proposed method.

[1] `https://fenicsproject.org/`

4.1. Poisson equation - Manufactured solution

The 1D Poisson equation is used here to study the accuracy of the implemented method as well as to help determine the most effective test cases to be used in the evaluation of super-resolved adjoint-based error estimation. The Poisson equation (Equation 4.1) is defined $\forall x \in [0,1]$. The chosen manufactured solution is $u(x) = sin^2(2\pi x)$. This enables the derivation of the required forcing term, f, on the right-hand side of Equation 4.1 as $f = 8\pi^2\left[sin^2(2\pi x) - cos^2(2\pi x)\right]$. Finally, Dirichlet boundary conditions are applied as $u_{left} = sin^2(2\pi 0) = 0$ and $u_{right} = sin^2(2\pi 1) = 0$.

$$-\frac{\partial^2 u}{\partial x^2} = f \tag{4.1}$$

For the use of an output error estimator, the output must also be chosen. Here $J = \int_\Omega u d\Omega$ was used. The exact error can then be evaluated and is computed to be $J_{exact} = 0.5$. This enables the computation of the exact numerical error in the output as well as the error in the error estimation.

The error estimator used here is the exact form of the fine space adjoint-weighted residual estimator as presented in the previous chapter and given in Equation 4.2. This equation takes the fine space solution of the adjoint, as indicated by the subscript h, and multiplies it with the strong primal residual in the fine space of the coarse primal solution projected into the fine space, represented by the notation $(\cdot)^H_h$. In order to evaluate the accuracy of the implementation, the actual fine space adjoint was computed. This is not done in practice since it would require the solution of the fine space primal, meaning that the fine space QoI could be computed directly as well as the exact local error contributions.

$$\delta J \approx \delta J_{est} = -\Psi_h^T R_h(u_h^H) \tag{4.2}$$

In order to compute the adjoint solution, the adjoint PDE must be derived. Following the approach documented in the preceding chapter, the sensitivity of the residual of Equation 4.1 to an arbitrary infinitesimal perturbation δu is derived:

$$r'(\delta u) = -\frac{\partial^2(\delta u)}{\partial x^2}$$

Using this equation to express the relation between this sensitivity and the sensitivity of the QoI, found in Equation 3.3 and integrating by parts leads to Equation 4.3. This equation is defined for all $x \in [0,1]$ with homogeneous boundary conditions.

$$-\frac{\partial^2 \Psi}{\partial x^2} = \frac{dJ}{du} = 1 \tag{4.3}$$

Finally, the validation relies on the analysis of the error estimator with respect to two main parameters: the choice of the discrete elements used for the numerical solution and the choice of the fine function space. As described in the introduction of this chapter, only h-refinement was used.

Element choice The first parameter which was analysed was the choice of finite element family. Here, two types were compared: piece-wise continuous quadratic and cubic elements, denoted from now on as quadratic and cubic respectively. These are the two most simple elements that allow the direct evaluation of $r(u)$.

Approximation of the fine space adjoint Since the proposed method will rely on the use of SRNN to reconstruct the fine space adjoint from the coarse space solution, it must be considered that higher-order elements cannot be used in this reconstruction step since only the value at the vertices of the mesh are considered by the SRNN. For this reason, the fine space adjoint must be projected into a piece-wise continuous linear function space. This enables the impact of using lower order approximations of the fine adjoint to be evaluated. The fine adjoint was computed using the same quadratic and cubic elements, before being projected down to a piece-wise continuous linear space for fine meshes twice and four times finer than the coarse mesh, referred to as 2x and 4x.

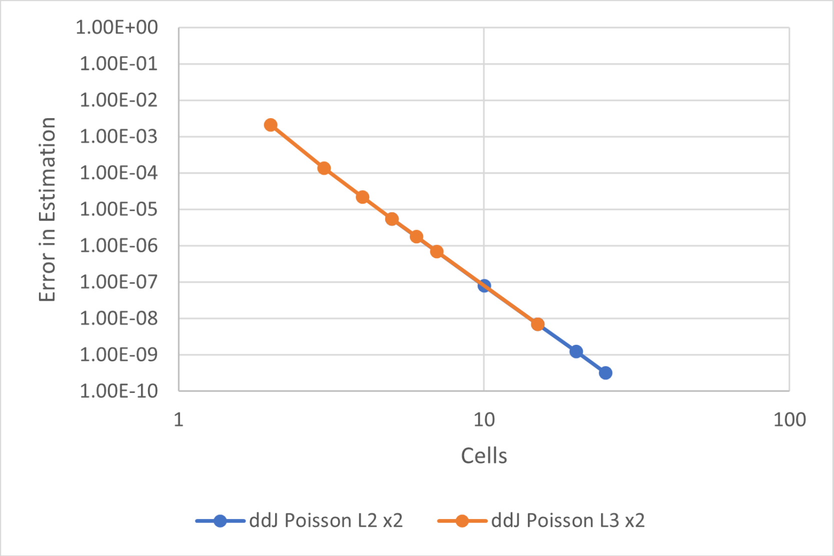


Figure 4.1: Error in output error estimation vs element polynomial order for the Poisson MMS. L2 denotes quadratic elements and L3 cubic.

The results of the error in output error for the quadratic and cubic elements is presented in Figure 4.1. For both families of elements, one can observe that the output error is well predicted across all ranges of refinement of the coarse mesh. This does not come as a surprise since the problem is smooth, without discontinuities, and can be well approximated using both p- and h-refinement, although only the latter is shown here. Note, the convergence rate of the error estimate is the same for both polynomial orders. This would not necessarily be expected.

Figure 4.2 shows the impact of projecting the cubic and quadratic elements into linear elements onto the error in output error estimate. This figure highlights the increase in accuracy gained by the use of a finer adjoint mesh. For quadratic elements, an increase in accuracy of the output error estimation of nearly two orders of magnitude is achieved. For cubic elements, the increase in accuracy is less notable for otherwise well resolved adjoint solutions. However, the convergence rate of the output error estimation accuracy is far higher, when the last point is considered, for the adjoint-based error estimate using the cubic elements. Since this behaviour is not observed for the range of uniform refinement levels considered, this would indicate that the impact of using the linear projected adjoint solution affects primarily the cubic elements' ability to accurately estimate the output error.

4.2. Burgers' equation - Manufactured solution

The one-dimensional Burgers' equation used for this problem is found in Equation 4.4. The manufactured solution considered in this case is $u(x) = 0.5 + sin(x)$ for all $x \in [0,5]$. This leads to the derivation of the boundary conditions $u(0) = 0.5$ and $u(5) = 0.5 + sin(5)$ as well as the forcing term $f = 0.5 \cdot cos(x) + cos(x) \cdot sin(x) + sin(x)$. The considered QoI is the same $J = \int_\Omega u d\Omega$ which leads to $J_{exact} = 3.5 - cos(5)$

$$u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = f \tag{4.4}$$

The same steps are followed in order to derive the adjoint system of equations. First, the residual sensitivity is derived in Equation 4.5. Since the Burgers' equation is non-linear, the residual must be linearised about a chosen primal solution u_0, here the steady primal solution. This is then substituted in Equation 3.3 and integrated by parts, leading to the adjoint PDE Equation 4.6. It must be reminded that the error estimator is less precise for non-linear problems since the linearisation of the adjoint leads to Equation 4.2 being also linearised and leads to a truncation error of $\mathcal{O}(\delta u^2)$.

$$r'[u_0](\delta u) = u_0\frac{\partial(\delta u)}{\partial x} + \delta u\frac{\partial u_0}{\partial x} - \frac{\partial^2(\delta u)}{\partial x^2} \tag{4.5}$$

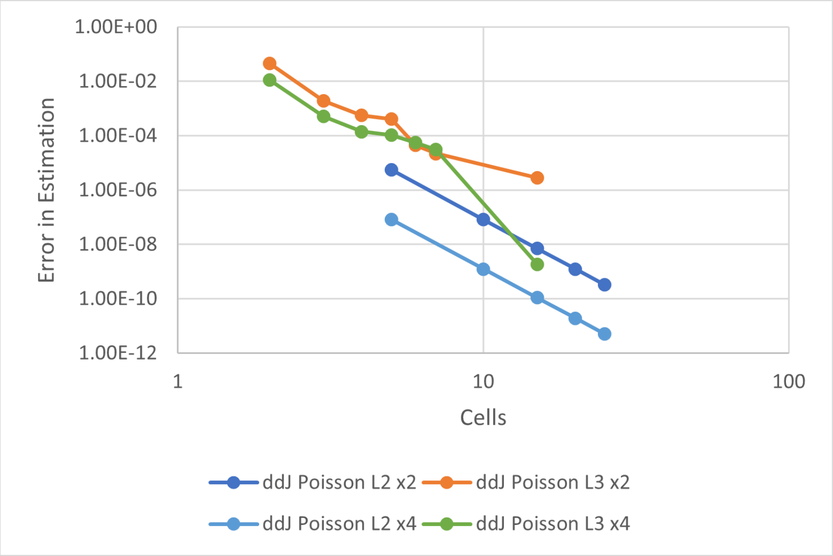


Figure 4.2: Error in output error estimation vs adjoint resolution for the Poisson MMS for solutions projected to linear elements. L2 denotes quadratic elements and L3 cubic.

$$\Psi\frac{\partial u_0}{\partial x} - u_0\frac{\partial \Psi}{\partial x} - \frac{\partial^2 \Psi}{\partial x^2} = 1 \tag{4.6}$$

The same approach is considered as with the Poisson equation. Figure 4.3 shows the rate of accuracy change for both quadratic and cubic elements for twice and four times finer meshes of the adjoint solution. Unlike the previous validation, the error in output error estimation is no longer linear, which may be due to the error introduced in the linearisation of the adjoint equation. However, one can observe that the rate of increase in accuracy for cubic elements converges to the ideal rate as per the Poisson validation case. This rate is not achieved for the 2nd order polynomial elements although their results are more monotonous.

Finally, the use of four time finer adjoint space is not as significant in the increase in accuracy. It does enable the cubic elements to have a more monotonous increase in accuracy across various uniform refinement levels but the actual accuracy is comparable. For the quadratic elements, one can observe an increase in accuracy of about half an order of magnitude. If one takes into account the increase in computational cost induced by using such a refined space, its level of refinement is not justified.

Approximation of the fine space adjoint When the adjoint solution is projected as a piece-wise continuous linear function the consequences are different. This is shown in Figure 4.4. Similarly to the Poisson problem, the quadratic elements seem to benefit far more from the additional refinement in the adjoint solution. Moreover, the additional performance in accuracy given by the finer adjoint solution also results in the quadratic elements being able to outperform higher-order polynomial elements.

This highlights the importance of the ability of the proposed method to reconstruct the fine space adjoint for low order elements. If the SRNN is able to reconstruct the fine adjoint accurately, it can be expected to greatly improve the error estimation and as a result lead to a computationally inexpensive output error estimation method. Since it has been shown that the primal can be reconstructed accurately for turbulent primal flows for up to 8x finer resolution, it is possible to imagine that the adjoint can be reconstructed accurately for at least two to four times finer resolution. This would enable the solution of the adjoint on the same mesh as the primal or even coarser, provided it is sufficiently resolved. This is confirmed by figures 4.3 and 4.4. These figures indicate that the implemented reference output-based error estimates are able to converge to the true output. However, the linearisation of the fine adjoint solution seems to hinder the ability of cubic elements to gain accuracy in their error estimate whereas lower order elements gain significant accuracy.

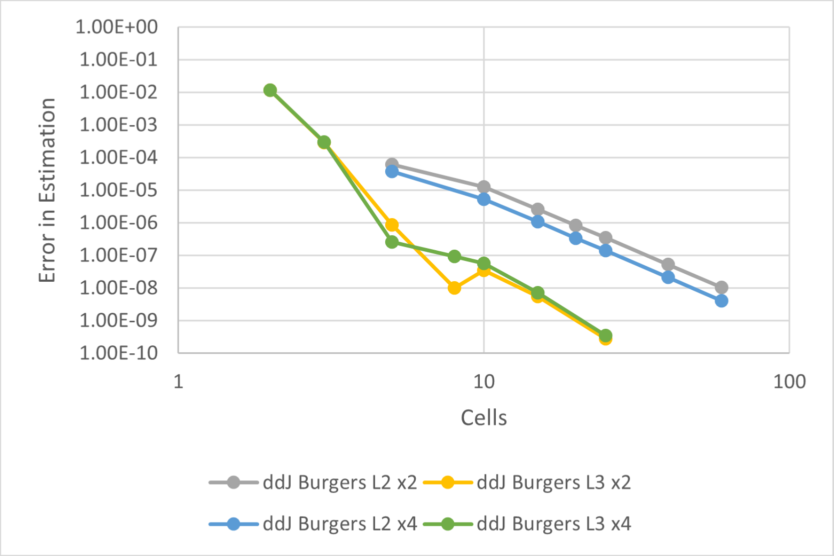


Figure 4.3: Error in output error estimation vs element polynomial order for the Burgers' equation MMS. L2 denotes quadratic elements and L3 cubic.

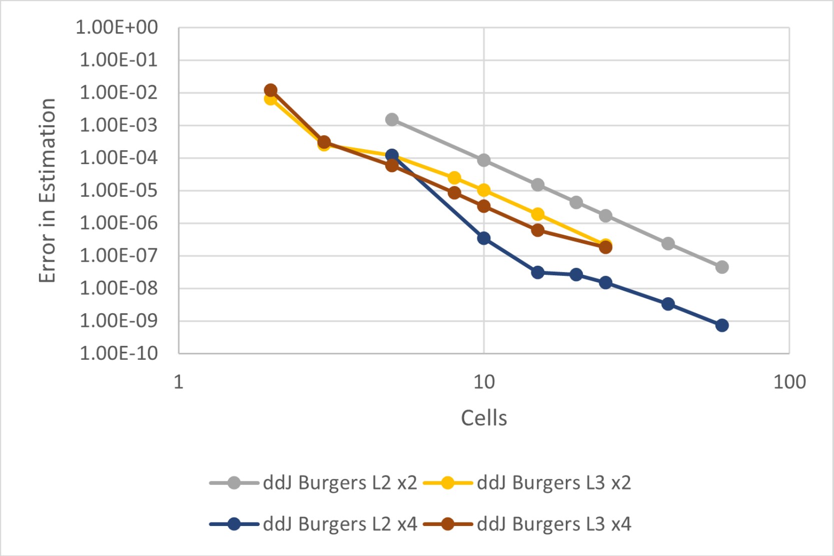


Figure 4.4: Error in output error estimation vs adjoint resolution for the Burgers' equation MMS for for solutions projected to linear elements. L2 denotes quadratic elements and L3 cubic.

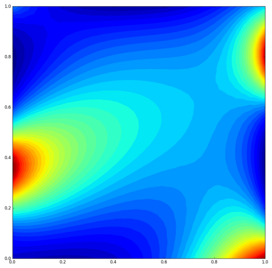


Figure 4.5: Primal solution for the verification scalar advection-diffusion problem for $\nu = 0.1$ and $\alpha = 30\,\text{deg}$

4.3. Scalar advection-diffusion - Verification

The scalar advection-diffusion problem chosen for verification and further investigation of the adjoint-weighted residual error estimate implementation. The case is described in [17]. This problem was used to explore different types of QoI: integrals over boundaries and integrals over a domain. It also allowed the evaluation of the method in a 2D test case. The primal PDE is given in Equation 4.7. It is defined for all x, $y \in [0,1]^2$ with chosen boundary conditions satisfying $u(x,y) = exp\left(0.5sin(-4x+6y) - 0.8cos(3x-8y)\right)$. The two parameters of the PDE are chosen such that they can be described by two variables, as such the convection is described by a vector of unit length at angle $\alpha = 30$ from the x-axis, and the kinematic viscosity $nu = 0.1$. The primal solution is shown in Figure 4.5.

$$\mathbf{v} \cdot \nabla u - \nu \nabla^2 u = 0 \tag{4.7}$$

In [17], the chosen QoI is the integral of the flux $-\nu \nabla u$ over the right boundary, $x = 1$. This is compared here to the QoI related to the resolution of the field $J = \int_\Omega u d\Omega$. The approximated exact QoI for the flux integral and the integral of u are -0.152503 and 1.143833, respectively.

This leads to different continuous adjoint problems, both being described by Equation 4.8. The flux integral, on the one hand, leads to non-homogeneous boundary conditions. The right boundary condition is $u(1,y) = -1$ and the rest are 0 with a forcing term $f = 0$.

On the other hand, the domain integral QoI leads to homogeneous boundary conditions and $f = 1$. The two different adjoint solutions are shown in Figure 4.6.

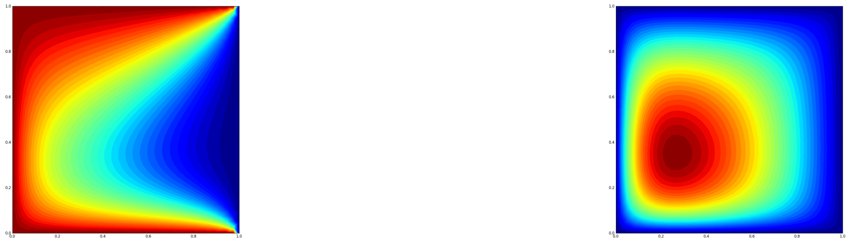
$$-\mathbf{v} \cdot \nabla \Psi - \nu \nabla^2 \Psi = f \tag{4.8}$$

Since it was seen that cubic elements do not lead to a significant increase in error estimator accuracy using the implemented adjoint-weighted residual method, the quadratic elements were chosen here. Moreover, the grid converged value is obtained for a fine primal solution of 320^2 elements, as in [17]. In Figure 4.7, one can observe that for this particular boundary integral QoI, the output error estimate is unable to reach an accuracy below 10^{-6}. This could originate in the inability of quadratic elements to capture the change of behaviour near the wall properly. This may be a result of the adjoint boundary conditions, where the right side corners are singularities since they are both defined to be 0 from one side and 1 from the other. Moreover, when looking at the location of the cell indicators in Figure 4.8, it is apparent that they are dominated by near-boundary cells.

This does not occur with the domain integral QoI, as shown in Figure 4.9. The cell indicators are located near the boundaries too but also throughout the computational domain. This means that the reconstruction of the adjoint in the field is more important than simply near the boundary as per the flux integral.

This also helps highlight that since the reconstruction of the adjoint near boundaries is expected to be difficult near boundaries, the proposed method may have difficulty being compatible with boundary integral QoIs. However, for laminar flows such as the one considered in chapter 6, the near wall region can be accurately reconstructed by a shear layer. Thus, using a priori understanding of the test case would still enable the use of boundary integral QoIs with the proposed method, but may require the use of other reconstruction methods to help the SRNN in the fine adjoint reconstruction task.

Finally, from this verification case, it was possible to identify that domain integral QoIs, are more sensitive to the implementation of the adjoint-weighted residual method. As a result, such QoIs will be pursued in



(a) Adjoint solution of the advection-diffusion verification problem for the boundary integral QoI

(b) Adjoint solution of the advection-diffusion problem for the domain integral QoI

Figure 4.6: Comparison of the flux integral adjoint solution and domain integral adjoint solution for the verification advection-diffusion problem

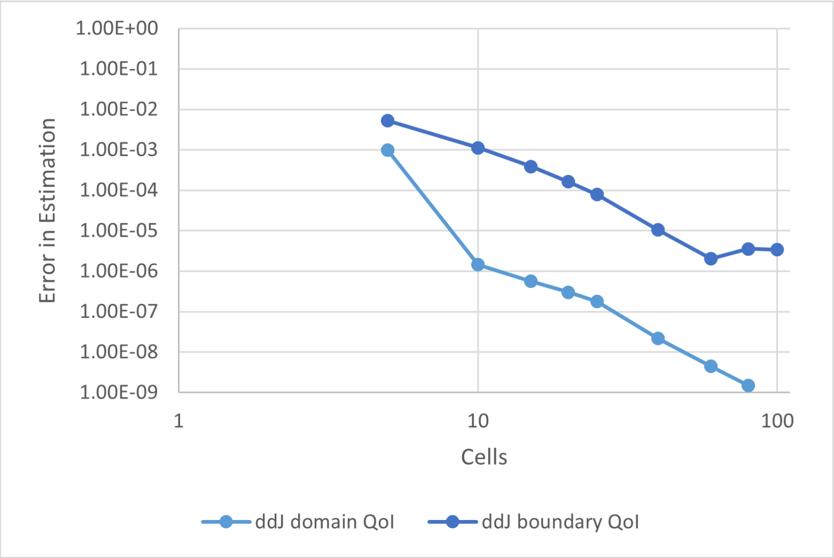


Figure 4.7: Comparison of the accuracy of the output error estimation for the boundary and domain integral QoIs

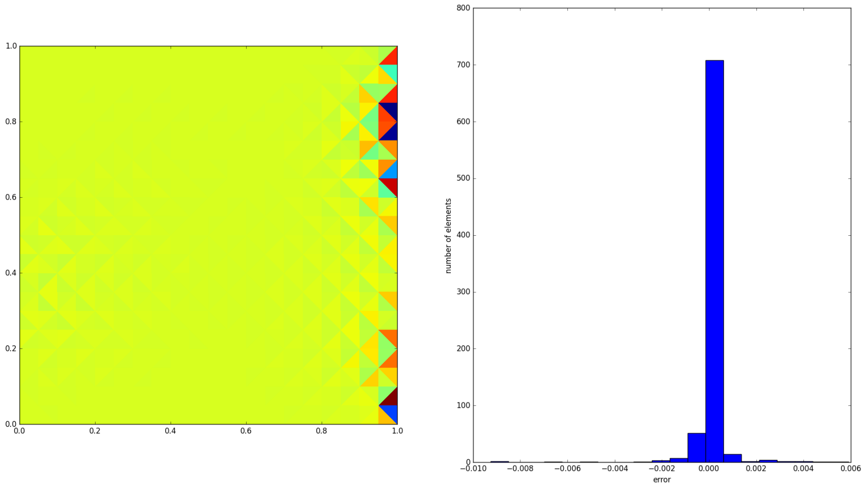


Figure 4.8: Error indicator location and distribution of the advection-diffusion problem for the boundary integral QoI

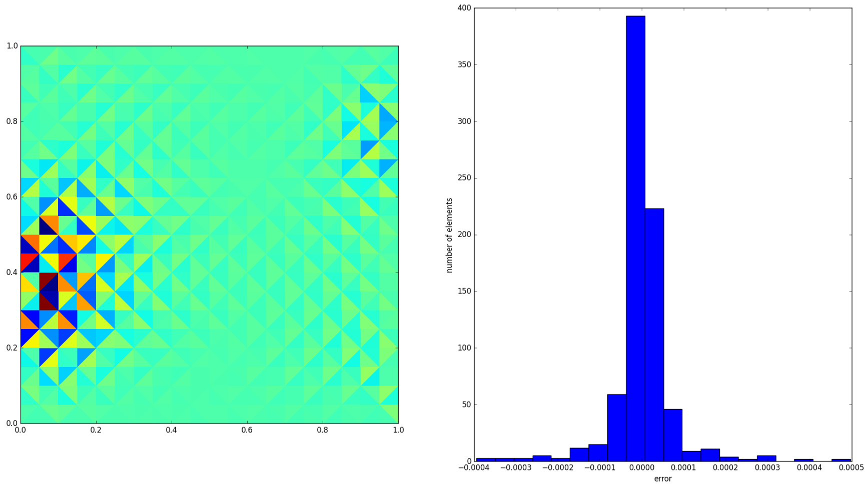


Figure 4.9: Error indicator location and distribution of the advection-diffusion problem for the domain integral QoI

the test cases in the following chapters. Moreover, quadratic elements will be chosen for the discretisation since they seemed to be the most respond the best to the finer space adjoint, an area in which the proposed method could perform well, and since the additional computational cost of using higher-order elements here seems unjustified. In the context of the Navier-Stokes equations, this enables the use of the well-documented Taylor-Hood elements, which are low order and relatively cheap unconditionally stable elements for the solution of the Navier-Stokes equations.

5

SRNN Adjoint Enrichment: Unsteady 1D Burgers' Equation

Building on the experience gained in the steady form of the Burgers' equation in the previous chapter, the unsteady Burgers' equation is considered here. Moreover, a time averaged integrated in both space and time QoI adds to the complexity of the case.

First the fluid equations are presented. This section present both the primal and adjoint equations as well as the discretisation choices which have been made. Section 5.2 then presents the chosen test case. The results of the reference output-based error estimates are then documented in section 5.3. The implementation of the proposed super-resolution output-based error estimate is presented in the following section, an assessed in section 5.5. Finally, a computational cost metric for the output-based error estimates is proposed and subsequently analysed in the closing section of this chapter.

5.1. Fluid equations

As mentioned in the introduction of this chapter, the one-dimensional unsteady Burgers' equation is considered. The equation, expressed as Equation 5.1 defined for all x on the range 0 to 1. The solution u having boundary conditions $u(0,t) = u(1,t) = 0$ and forcing term f. It is the most simple example of a nonlinear unsteady viscous equation and is used throughout literature, notably for the evaluation of discretisation schemes. [4, 56, 59]

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f \tag{5.1}$$

Following the approach described in the previous chapter to obtain the adjoint equation, one must derive $r'[u_0](u)$, shown in Equation 5.2 and equate it to the QoI dependent term on the left-hand side of Equation 3.2. The second and third terms in Equation 5.2 results from the product rule to the $u\frac{\partial u}{\partial x}$ term in the primal equation. The time derivative produces a term similar to the convection term.

$$r'[u_0](u) = \frac{\partial \delta u}{\partial t} + u_0\frac{\partial(\delta u)}{\partial x} + \delta u\frac{\partial u_0}{\partial x} - \nu\frac{\partial^2(\delta u)}{\partial x^2} \tag{5.2}$$

This results in the following adjoint equation:

$$-\frac{\partial \Psi}{\partial t} - u\frac{\partial \Psi}{\partial x} + \Psi\frac{\partial u}{\partial x} - \nu\frac{\partial^2 \Psi}{\partial x^2} = \frac{dJ}{du} \tag{5.3}$$

For the implementation in FEniCS, the equation is expressed in its weak form, obtained by multiplying by a test function of the function space $\mathcal{V}$ which is equivalent to the solution space. Integrating by parts and rearranging yields Equation 5.4. In this relation, it is assumed that there are essential boundary conditions for Ψ.

$$-(\frac{\partial \Psi}{\partial t}, v) - (u\frac{\partial \Psi}{\partial x}, v) - (\Psi\frac{\partial u}{\partial x}, v) + \nu(\frac{\partial \Psi}{\partial x}, \frac{\partial v}{\partial x}) - (\frac{dJ}{du}, v) = 0 \tag{5.4}$$

The problem is discretised using piece-wise quadratic elements. The performance of the output-based error estimates is compared using uniformly refined meshes, in the absence of a mesh adaptation routine. Since the problem is one-dimensional, the growth of the Degrees of Freedom (DoF) due to the refinement is much less than in practical cases. In one dimensional problems the increase in number of DoF is linear. In realistic three-dimensional time-dependent problems, the curse of dimensionality leads to the exponential growth in the number of DoF per refinement level and the inability to uniformly refine the mesh.

However, this simplified case allows for the analysis of the accuracy of the reference error estimators and comparisons to the proposed implementations of SRNN aiming at reconstructing the fine space adjoint Ψ_h from the coarse space adjoint Ψ_H in order to compute the exact value of the output error estimate from Equation 3.5.

In FEniCS a wide range of solvers are available. For non-linear problems, such as the primal problem, a SNES or Newton iterator can be chosen and many Krylov solvers are available. The chosen combination of solver and preconditioner used in this thesis was MUMPS and BT respectively. Although this is a direct solver, it was observed to be more stable for the coupled approach taken here. Additionally, the added computational cost and issues in efficiencies arising from the use of direct solvers were not observed for small scales problems such as the ones studied here.

5.2. Case presentation

To simplify the interpretation of the results, the method of manufactured solutions was used. The solution of this problem was computed on an interval $[0,1]$, $Re = 100$ and no-slip boundary conditions on both ends of the interval, and initialised with a zero field. The chosen manufactured solution is $u(x,t) = sin^2(\pi t) \cdot sin(\pi x)$ [45]. This allows for the simple derivation of the source term for the primal problem and the adjoint problem. The QoI is $\bar{J} = \frac{1}{T}(sin(\pi x), u)_{\Omega,I}$, with the brackets representing the integrated inner product. The primal is solved from $T = 0$ to $T = 20$ seconds. The averaging period is taken from $T = 10$s to $T = 20$s. The true value of the QoI is thus $\bar{J} = 0.25$. The advantage of this QoI is the weighing of the solution near the centre of the computational domain which lessens the contribution of the error of the near-boundary elements in the computation of the adjoint, and as a result in the error estimation.

The adjoint solution is solved backwards in time, with zero final conditions prescribed at $T = 20$s and homogeneous Dirichlet boundary conditions. The adjoint solution is computed from $T = 20$ to $T = 10$ seconds. The forcing term in this problem was obtained by evaluating a Frechet derivative of the objective function. This is obtained by computing the sensitivity of the objective function to a perturbation in the solution as shown in Equation 5.5. Whereas boundary integral QoIs are related to the adjoint problem through the application of boundary conditions, domain integral QoIs introduce a forcing term in the adjoint equation.

$$(g_u, \tilde{u}) := \lim_{\epsilon \to 0} \frac{J(u+\epsilon\tilde{u}) - J(u)}{\epsilon}, \quad \forall u,\ \tilde{u} \in \mathcal{V} \tag{5.5}$$

The QoI was chosen to evaluate the discrepancy between the chosen manufactured solution and the discrete solution obtained. However, being time-averaged, it is necessary to ensure that the temporal discretisation does not introduce errors in the output. In order to reduce the error related to temporal discretisation, a time-step of $dt = 0.001$ seconds was used [45]. This enables to make the assumption that the output error stems from the spatial discretisation and as such only focus on this variable. The problem was discretised spatially using the piece-wise continuous quadratic elements. The various refinement levels used for comparisons, are obtained through the uniform refinement of the interval $[0,1]$.

5.3. Reference error estimator performance

First, the performance of the reference output error estimators is evaluated in order to determine the baseline performance of best practices. The error estimators are thus presented and there relative performance evaluated in this section.

5.3.1. State-of-the-art error estimators

Three error estimation methods were considered. For all three chosen approaches, the fine space information was determined, solving for the actual fine space adjoint problem. In practice this would not be possible due to the associated computational cost, but it does enable the comparison of the true output error and the full capacity of each formulation of the error estimation. The error estimation methods employed in practice rely on the approximation of the fine space information and as a result perform less accurately.

The first error estimation method consists of computing Equation 3.5 directly, relating the output error to the solution of the fine space adjoint through a method known as the adjoint-weighted residual. It is referred to here as the **fine space adjoint** error estimation:

$$\delta J_{est} \approx -\int_{\Omega} \Psi_h^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega$$

This equation requires the solution of the fine space adjoint, which in turn requires the solution of the fine space primal. The projection of the coarse space primal solution was performed by evaluating the coarse solution at every fine space degree of freedom.

The second error estimation method implemented for comparison was the formulation of the error estimation providing the decomposition between computable and remaining error as found in Equation 3.6 [72]. This method uses the primal residual for the remaining error, it is referred here to as the **primal residual** approach:

$$\underbrace{\delta J_{est}}_{\text{Error}} \approx \underbrace{-\int_{\Omega} (\Psi_h^H)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Computable Error}} + \underbrace{\int_{\Omega} (\Psi_h^H - \Psi_h)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Remaining Error}}$$

Finally, the last reference error estimation method used here is the second method to compute the remaining error, Equation 3.7. In this error estimation method, the remaining error is computed using the residual of the coarse space adjoint solution projected in the fine space. This estimator is referred to as the **adjoint residual** approach in this report. For this method, the fine primal must be approximated in practice but the real value was used here in order to compare all the error estimators according to their best theoretical accuracy. The formulation is repeated here for legibility:

$$\underbrace{\delta J_{est}}_{\text{Error}} \approx \underbrace{-\int_{\Omega} (\Psi_h^H)^T \mathbf{R}_h(\mathbf{u}_h^H) d\Omega}_{\text{Computable Error}} + \underbrace{\int_{\Omega} (\mathbf{R}_h^{\Psi} (\Psi_h^H)^T (\mathbf{u}_h - \mathbf{u}_h^H) d\Omega}_{\text{Remaining Error}}$$

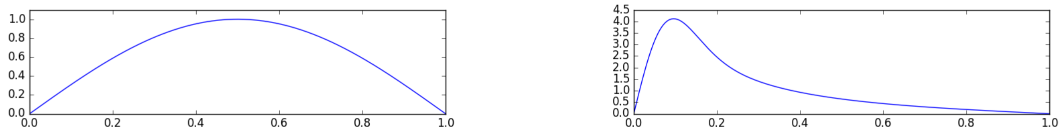
For all of these output error estimation methods, an additional contribution due to the non-zero value of the velocity at the starting time of the time averaging, $T = 20$s, must be added. This is specific to unsteady problems and is formulated as $(\Psi_h^h, u_h^H - u_h)_\Omega|_{t=t_0}$. The fine space adjoint and the primal residual approach give sensibly the same result when the exact fine adjoint is computed, and as such no difference is expected to be observed for these two reference error estimators when the fine adjoint solution is used.

5.3.2. Reference output error estimation

The impact of choosing p-refined and h-refined spaces has been well documented in the past, see [41], and since the FEniCS implementation of the reference output error estimates did not enable to consistently obtain accurate error estimates for the output, it is not pursued here. Nonetheless, the reference error estimates can be compared.

The use of p-refined fine space leads to a lower computational cost since the increase in the number of DoF is less significant than the bisection of elements for instance. However, it is not feasible in methods other than finite element analysis. Most of the CFD is still performed on finite volume codes and as such rely on the use h-refined spaces. Thus the insights gained here using h-refined adjoints are expected to generalise to finite volume methods.

In the following, the accuracy of the output error estimates are compared for various uniform mesh refinement levels. Although this is not strictly representative of the behaviour of the error estimation behaviours in the mesh adaptation process, it is considered to be the best comparison available in the absence of implementation of a mesh adaptation routine. Moreover, this allows for better interpretation of the performance of the output-based error estimates since the mesh adaptation procedure would introduce complex dynamics of cumulative local adaptation. The refinement levels are solely based on h-refinement of quadratic elements based on the preliminary validation performed in chapter 4.



(a) Snapshot of the Burger's equation primal solution at $t = 10.5$s

(b) Snapshot of the Burger's equation adjoint solution at $t = 10.5$s

Figure 5.1: Snapshots of the instantaneous primal and adjoint solutions at $t = 10.5$s for the unsteady Burgers' equation

Similar to the validation case presented in section 4.2, the flow appears to be well resolved at all mesh refinement levels, as shown for example in Figure 5.1. However, the performance of the error estimation methods is disparate, mostly in the coarsely resolved solutions. As expected all reference error estimates are accurate enough in the reconstruction of the output error leading to increasingly accurate error estimations. The adjoint residual formulation appears to be far more accurate in reconstructing the error made between the fine and coarse QoI, as represented by dJ_{H-h}, as shown in Figure 5.2.

The true error between the coarse and fine primal solutions is shown in the grey and highlights the superior ability of the adjoint residual remaining error formulation to accurately estimate the output error, in particular for under-resolved solutions. The fine adjoint and primal residual remaining error formulation give sensibly the same error estimates due to the fact that in this case the fine adjoint is not estimated but solved for.

The error in output-based error estimation is presented in Figure 5.3. It can be observed that the accuracy of the error estimates improves and then levels off. This is attributed to numerical errors and the accumulation of errors added over the thousands of time-steps. This is found to be consistent with the addition of errors of order of magnitude $\mathcal{O}(10^{-16})$ added about 10^3 times. The error in output error with respect to the grid converged value of the QoI indicates the greater accuracy of the adjoint residual formulation. The convergence rate of the error estimate is similar for both methods and for sufficiently fine meshes, near identical. This highlights the trivial nature of the output error estimation in this test case. The non-linearity of the problem does not appear to add much complexity to the error estimators' ability to remain accurate, even at low refinement levels.

Apart from the levelling off of the output error estimate accuracy, the same behaviour as in the validation chapter is observed. During the validation, the projection of the fine adjoint solution in a piece-wise continuous linear function space still enabled the accurate estimation of the output error. Thus, these findings support the idea that a SRNN would be able to accurately estimate the output error. The proposed method would require the sufficiently accurate reconstruction of the fine space adjoint solution.

5.4. Machine learning implementation

One of the simplest, yet robust method for primal solution super-resolution is the use of static Convolutional Neural Networks (CNN). The architecture was inspired by the work of Fukami et al., Liu et al. where it was applied to 2D primal reconstruction. The architecture is thus duly modified for the 1D problem considered here. The architecture is shown in Figure 5.4 for a chosen upscale factor of 4. The Fully Convolutional Neural Network (FCNN) enables the use of inputs and outputs of varying sizes where the amount of reconstructed information is determined by a chosen upscale factor, here referred to as 2x or 4x for up-scale factors of 2 and 4 respectively. It does mean that separate networks need to be constructed for different upscale factors. However, the same raw data-set can be used to generate the low resolution and high resolution training pairs. The particularity of this specific architecture is the use of a PixelShuffle layer in the last layer which aims to

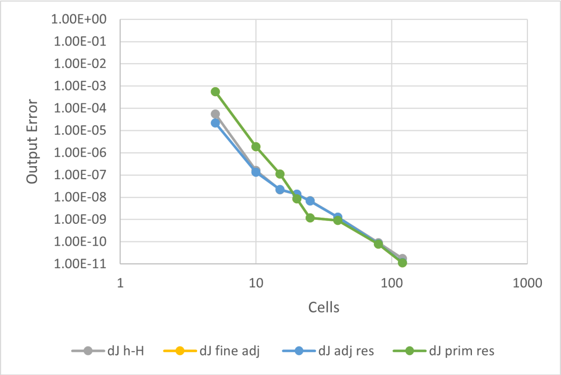


Figure 5.2: output error estimates for the reference error estimators for the Burgers' equation

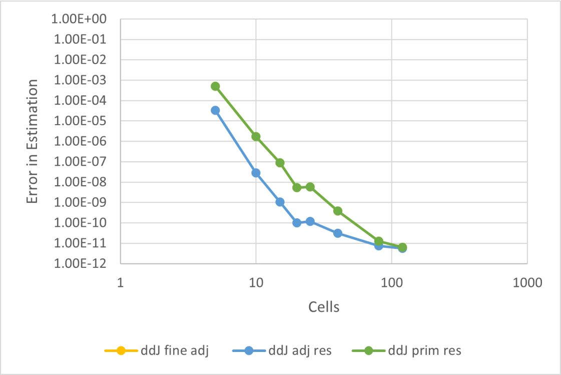


Figure 5.3: Comparison of the error in output error estimate for the Burgers' equation test case with respect to the discrete solution

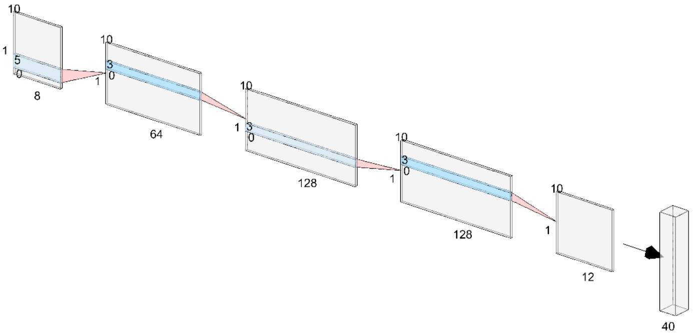


Figure 5.4: Architecture of the 1D super-resolution neural network

interlace the final result with the depth information of the preceding layer [39, 49].

The training pairs were constructed by extracting random samples of the solution between the time interval $T = [10, 20]$, such that both the primal and adjoint states are known. From these samples of differing resolutions, a fixed interrogation window of 20 elements was sampled, the window is then sub-sampled in order to form a high and low resolution pair. This did not yield a diverse and large enough data-set. It was thus chosen to sample the higher resolution solutions randomly for the creation of new training pairs. The data-set was also increased by taking mirrored solutions of the same problem, effectively doubling the number of pairs and attempting to make the x position independent from the u values. The only positional information was thus the relative position of data points. The training pairs were chosen to have three channels in input (x, u, u_a) and one channel in output (u_a).

The training was done with a learning rate of 0.001 since it was found that beyond this, the training performance would no longer improve and yield unsatisfactory results. The loss function is the same as the previously mentioned literature, namely the Mean-Square Error (MSE). This has limitations, which are explored in [48, 68, 83]. These are related to the lack of physical meaning and the inability to evaluate the plausibility of the flow. However, the MSE is a robust indicator and performs well with the existing machine learning frameworks and requires little computational overhead.

The simplicity of the problem led to the rapid convergence in training of the SRNN. Beyond 150 epochs, the convergence of the loss function became chaotic. This can be introduced by a large learning rate, and the beginning of the model over-fitting the training data. Such an example for the loss function can be seen in Figure 5.5. The loss function for the 2x super-resolution Convolutional Neural Network (CNN) can be seen in Figure 5.6.

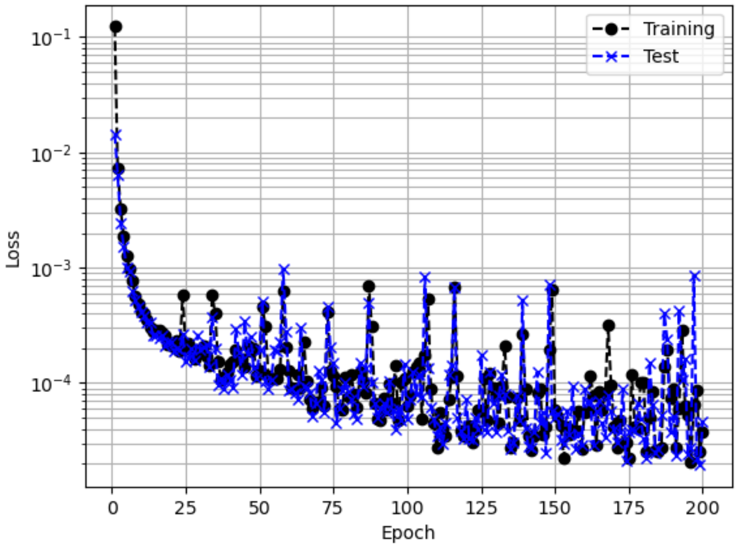


Figure 5.5: Example of the impact of over-fitting and high learning rate on the loss function for the 1D Burgers' equation super-resolution

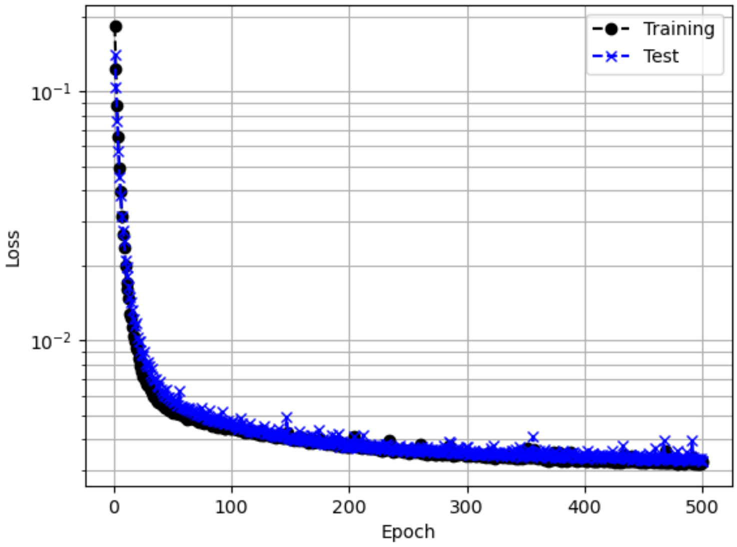


Figure 5.6: Loss for the training of the 1D 2x CNN

5.5. Assessment of the proposed method

Two CNN were compared, the first with an upscale factor of 2 and the second 4. The CNN was found to be able to reconstruct the fine space information from the coarse space primal and adjoint solution. The only major difference observed was due to the bias introduced in the creation of the training data-set. For the higher upscale factor, the input used in training was quite small in size (5 cells up-sampled to 20). This resulted in the network learning to reconstruct only small intervals of the solution. In the learning process, the network was able to minimise the loss function by effectively reconstructing the entire solution on each interval of data. This can be seen in Figure 5.7. This leads to a less smooth solution compared to the exact fine space adjoint. This affects predominantly the higher up-sampling factor SRNN.

In Figure 5.8, one can observe that for low mesh resolution, the method is just as accurate as the reference error estimation methods in estimating the output error. The main observed difference is the different rate of change of the output error estimation. Being shallower for the proposed method, it indicates that it is less accurate for a similar starting primal resolution. Nonetheless, it appears the SRNN is able to reconstruct a sufficient amount of information of the fine adjoint from the coarse adjoint solution in order to estimate the output error and as a consequence drive the mesh adaptation process. The only predictable outcome of implementing this method for this problem would be the requirement for additional mesh adaptation iterations such that the threshold of output error is achieved.

Finally, it can be noted that the proposed method, using 2x super-sampling for low resolution solutions at 5 elements is more accurate or on par with the reference error estimators. This is a desired behaviour since the accurate output error estimation is a notoriously difficult task for under-resolved flows [8, 64] . This is not observed in the higher super-resolution case due to the inaccurate reconstruction of the fine adjoint, resulting from a bias induced in the training process.

The error in output error estimation with respect to the grid converged solution is given in Figure 5.9. In this figure, one can observe that the inaccurate reconstruction of the fine adjoint severely hinders the ability to estimate the output error accurately. Moreover, the shallower rate of convergence is also observed for the 2 times super-resolution. This indicates that despite the small inaccuracies for finer resolution meshes, the output error estimate using SRNN is able to improve its output error estimation, albeit at a slower rate than solving the fine adjoint equations.

It must also be noted that the higher super-resolution factors also induce a higher computational cost since they require the evaluation of the strong residual and projection of the coarse primal solution in a much finer space. This increase computational cost is evaluated in the following section.

Finally, the error indicators, or adaptation parameters can be compared for this test case. Here the fine mesh indicators are shown in figures 5.10 to 5.13. In these figures, one can see that some inaccuracies in the reconstruction of the adjoint. This greatly affects the ability to estimate the output error accurately. Nonetheless, the error indicators are very well reconstructed for both the 2x and 4x CNN. This means that in a mesh adaptation routine, the proposed method is expected to behave very similarly to the reference output error estimators. And this not only solely due to the global behaviour of the reconstruction over the coarse mesh but also at the level of the fine mesh where the error indicators are extremely similar.

The only difference observed is in 4x CNN where the low gradient region on the right of the adjoint solution is reconstructed poorly with the appearance of noise. This can be partly attributed to incomplete training since it is expected that theoretically this can be learned as part of the super-resolution task.

5.6. Computational cost analysis

A metric for the cost of computing the error estimation is compared between all the aforementioned error estimates. The proposed metric is the Computational Cost Metric (CCM), and aims at providing an adimensional indicator of the computational cost of estimating the output error. It is computed according to Equation 5.6. The relation represents an effective operations count by weighting the number of Krylov iterations by the DoF comprising the solution. Since a direct linear solver was used, the problem stiffness on iterative convergence is not accounted for, only the number of times the non-linear solver iteration is called. The metric is adimensionalised by dividing this operations count by the operation count required for the coarse

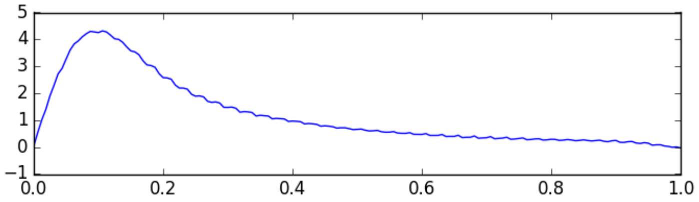


Figure 5.7: Reconstructed fine adjoint solution with an upscale factor of 4 at $t = 10.5$s

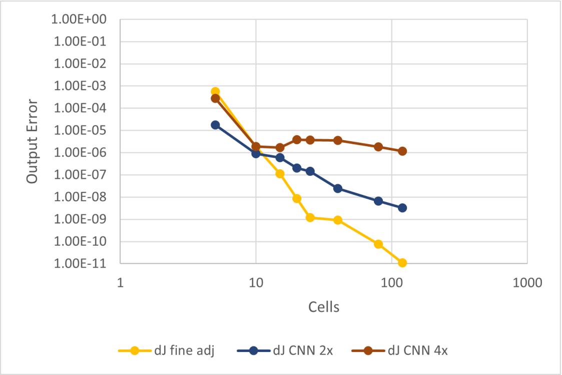


Figure 5.8: Comparison of the output error estimate for the Burgers' equation test case using CNN-based super-resolution

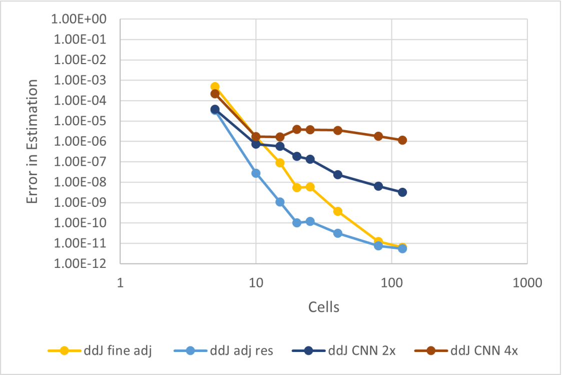


Figure 5.9: Comparison of the error in output error estimate for the Burgers' equation test case using CNN-based super-resolution

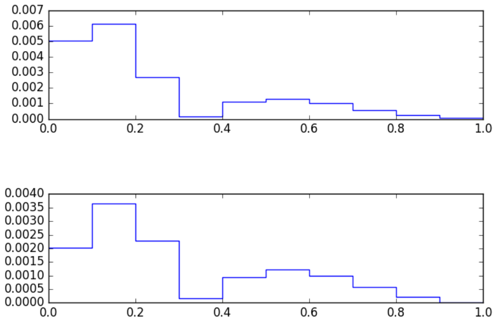


Figure 5.10: Comparison of the error indicators for the Burgers' equation test case using 2x CNN-based super-resolution of the adjoint for $n = 5$ cells. Fine adjoint solution on top and CNN reconstruction on the bottom.

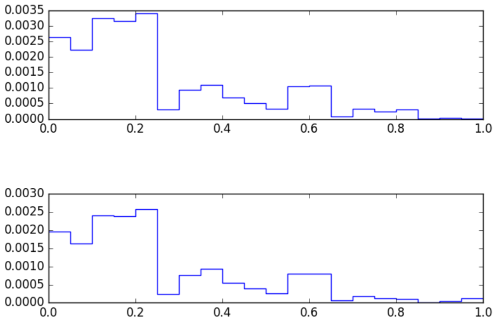


Figure 5.11: Comparison of the error indicators for the Burgers' equation test case using 4x CNN-based super-resolution of the adjoint for $n = 5$ cells. Fine adjoint solution on top and CNN reconstruction on the bottom.

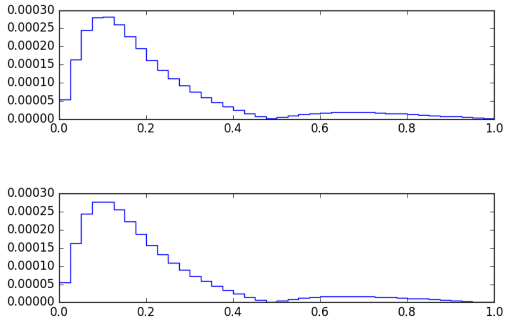


Figure 5.12: Comparison of the error indicators for the Burgers' equation test case using 2x CNN-based super-resolution of the adjoint for $n = 20$ cells. Fine adjoint solution on top and CNN reconstruction on the bottom.

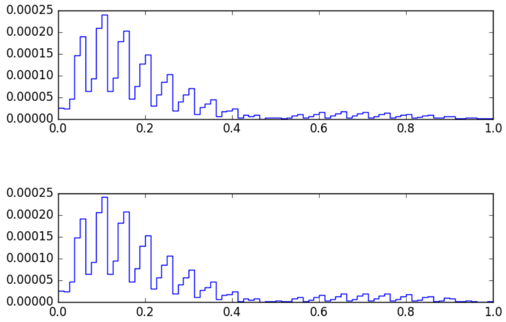


Figure 5.13: Comparison of the error indicators for the Burgers' equation test case using 4x CNN-based super-resolution of the adjoint for $n = 20$ cells. Fine adjoint solution on top and CNN reconstruction on the bottom.

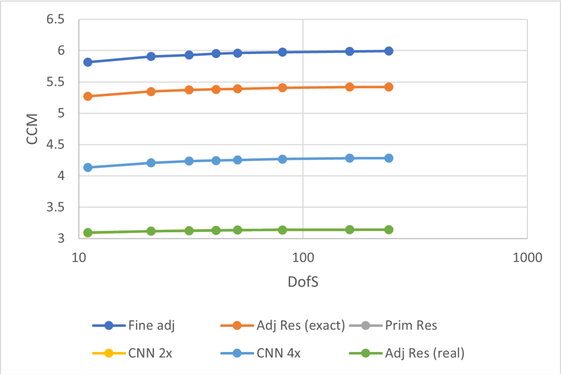


Figure 5.14: CCM versus DoF of the coarse mesh for the unsteady Burgers' equation

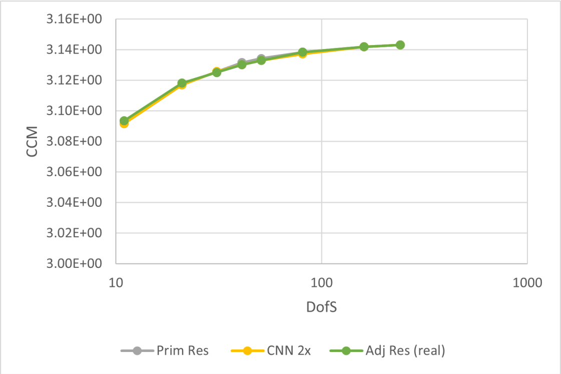


Figure 5.15: Detail of the CCM versus DoF of the coarse mesh for the unsteady Burgers' equation

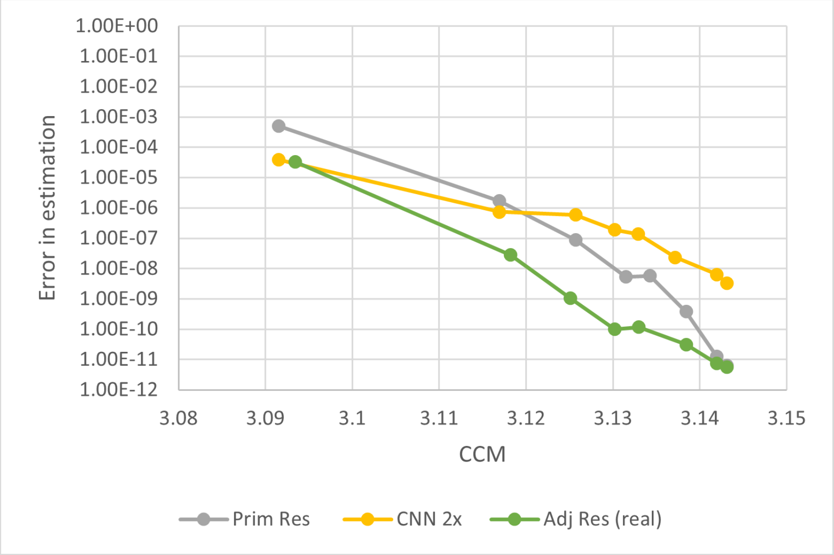


Figure 5.16: CCM versus the error in output error estimation for the unsteady Burgers' equation

primal solution. The CCM then provides an estimate total cost for the estimation of the output error.

$$CCM = \frac{\sum_i n_{iter_i} \cdot \text{DoFs}_i}{n_{iter_{primal}} \cdot \text{DoFs}_{coarse}} \tag{5.6}$$

Figures 5.14 and 5.15 show the increase in computational cost with the increase in the number of degrees of freedom of the coarse mesh. The first figure also shows the superior computational cost required for the solution of the fine problem. For instance, the fine adjoint solution requires the solution of the fine primal and as a result requires the most computational effort to estimate the output error. Similarly, the exact adjoint residual also requires the solution of the fine primal. However, since the fine adjoint is not solved, the CCM is lower. For the SRNN-based output error estimates, the 4x CNN also leads to an increase in computational cost due to the evaluation of the reconstructed adjoint-weighted residual on a far finer mesh, containing approximately four times more DoF than the fine mesh for the other methods.

Figure 5.15 focuses on the cheapest output error estimates. The cost of the adjoint and primal residual output error estimates in this figure are obtained by assuming the primal and adjoint are solved on the coarse mesh and that prolongation operators are used. Using this assumption, the reference output-based error estimates have near equal CCM.

Although the CCM converges to a value with uniform mesh refinement, it must be emphasised that this means the total cost of the primal solution and error estimation becomes proportional to the cost of the primal solution, which is increasing with the mesh refinement.

The CCM is also plotted with respect to the corresponding error estimate accuracy. Since the CCM was found to be superior for the fine adjoint output error estimate and the 4x SRNN, these are not present in this figure. This indicates the relative cost of the output error estimation with respect to the accuracy of the estimation. One can observe the increased computational cost required for the increased accuracy in the output error estimation. Although the CCM is very similar for these three error estimation level at each refinement level, this highlights the inability of the SRNN-based error estimate to provide increasingly accurate output error estimation. For the most refined meshes, the same computational cost yields a less accurate than the reference error estimation methods by more than two orders of magnitude. This, and the cheaper computational cost of the SRNN-based estimate for coarse refinement levels, strengthens the argument that this approach would be very suitable for under-resolved solutions.

In practice, the approximations of the fine space solutions of either the primal or adjoint fields are made using prolongation operators, which further reduce the accuracy of the output error estimation [55]. However, since these prolongation operators are no longer required for the use of the super-resolution reconstruction of the fine field adjoint, it is expect that such an approach could yield superior performance improvements.

The increase in computational overhead induced by the use of higher up-scale factors could be addressed in the future by performing super-resolution to a finer space before sub-sampling this solution to a simple single refinement level. The sub-sampling would not require much additional computational cost but greatly reduce, due to the curse of dimensionality, the cost of projecting the coarse primal into the fine space and evaluate the adjoint-weighted residual on a less refined mesh.

6

SRNN Adjoint Enrichment: Lid-driven Cavity Flow at Re = 250, 500 and 900

The lid-driven cavity flow has been thoroughly studied as a benchmark for assessing numerical methods, verifying code accuracy and implementations [11, 30, 40, 47]. In this case, it is used as a more complex problem to evaluate the performance of the proposed SRNN-based error estimation method. The steady lid-driven cavity flow is considered here.

This chapter first describes the problem and the related adjoint equations in sections 6.1 and 6.2. The performance of the reference output-based error estimates is then evaluated. This enables the comparison of the proposed method, whose implementation is described in section 6.4, to be conducted in section 6.5. The computational cost metric is used to further evaluate the performance of the proposed method. Finally, section 6.7 concludes the chapter and assesses the performance of the proposed use of SRNN for the adjoint solution.

6.1. Fluid equations

In this chapter, the Navier-Stokes equations were considered. This system of PDE describes the motion of a viscous fluid and can be expressed as in Equation 6.1. The first equation represents the conservation of mass whereas the second the conservation of momentum. In the momentum equation, it is apparent that the pressure must now be determined, as well as the velocity field. In practice, this can be solved in two ways, either by coupling the variables in a monolithic solver or by solving each independently and iterating until a solution is found, as in a segregated solver.

$$\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} - \frac{1}{Re}\nabla \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \end{aligned} \tag{6.1}$$

This implementation of the Navier-Stokes solver was performed using the finite element discretisation in the problem solving environment FEniCS[1], using a coupled approach.

As the steady problem is considered here the time dependent term in the momentum equation was set to zero. The BT preconditioner and MUMPS direct Krylov subspace solver were used, being stable for Reynolds numbers below 1200. The added computational cost of using a direct solver was outweighed by the robustness of this approach for the considered range of Reynolds numbers. For simplicity the laminar case is considered here. FEniCS does not include turbulence models in its distribution, however, a development branch in Oasis has implemented a few LES and Reynolds Averaged Navier-Stokes (RANS) models [10].

The implementation of the Navier-Stokes equations in FEniCS requires the formulation of the PDE in their weak form as shown in Equation 6.2. The derivation can be found in literature [58]. In this relation, v and q

[1] `https://fenicsproject.org/`

are test functions associated with the velocity **u** and pressure p.

$$((\mathbf{u}\cdot\nabla)\,\mathbf{u},\mathbf{v})+\left(\frac{1}{Re}\nabla\mathbf{u},\nabla\mathbf{v}\right)-\left(\nabla\cdot\mathbf{v},p\right)-(\mathbf{f},\mathbf{v})+\left(\nabla\cdot\mathbf{u},q\right)=0 \tag{6.2}$$

The continuous adjoint is derived using the same method as in [42, 58, 60]. The sensitivity of the residual of Equation 6.1 to an arbitrary perturbation in the velocity field $\delta\mathbf{u}$ and pressure field δp is considered. However, as in the Burgers' equation, the non-linearity of the Navier-Stokes equations require the linearisation about a given state $\mathbf{u}_0$, here the steady state primal solution. This residual sensitivity is shown in Equation 6.3.

$$\begin{aligned} r'_{mass}[\mathbf{u}_0](\delta\mathbf{u}) &= \nabla\cdot\delta\mathbf{u} \\ r'_{momentum}[\mathbf{u}_0](\delta\mathbf{u}) &= \dot{\delta\mathbf{u}}+(\delta\mathbf{u}\cdot\nabla)\,\mathbf{u}+(\mathbf{u}\cdot\nabla)\,\delta\mathbf{u}-\frac{1}{Re}\nabla\cdot\nabla\delta\mathbf{u}+\nabla\delta p \end{aligned} \tag{6.3}$$

This enables the derivation of the linearised continuous adjoint system of PDE, which results in the form Equation 6.4, where the adjoint variable Ψ is decomposed into its adjoint velocity and pressure terms, Ψ_u and Ψ_p respectively.

$$\begin{aligned} -\nabla\cdot\Psi_u &= \frac{dJ}{dp} \\ -(\mathbf{u}_0\cdot\nabla)\,\Psi_u+(\Psi_u\cdot\nabla)\,\mathbf{u}_0-\frac{1}{Re}\nabla\cdot\nabla\Psi_u-\nabla\Psi_p &= \frac{dJ}{d\mathbf{u}} \end{aligned} \tag{6.4}$$

The derivatives of the QoI are determined in the following section, where the choice of the QoI is made. They are taken to be a Frechet derivative of the chosen QoI. The weak form is derived in a similar manner as the primal solution and is obtained by multiplying by a test function and integration by parts [58]. The problem was solved using Taylor-Hood elements, comprised of piece-wise continuous quadratic elements for the velocity and linear for the pressure.

6.2. Case presentation

The computational domain of the lid-driven cavity flow consists in a square of unit length side where all but one side are walls. The last side is considered to be the lid and, in the steady case, a unit velocity is applied in the tangential directions of the lid. The geometry can be seen in Figure 6.1. The characteristics and history of this problem are thoroughly described in [1, 40]. However, this test case is loosely based on [58], where the choice and derivations of the QoI and associated boundary conditions are well documented, including the derivation of the weak forms. In [58], the QoI is the drag over the lid, which is an indicator of how resolved the field is near the boundary, $J=\int_{\Gamma_D}\left(\frac{1}{Re}\nabla_{\mathbf{n}}\mathbf{u}-p\mathbf{n}\right)\cdot\mathbf{j}_{\Gamma_D}ds$.

The analysis in this chapter differs from the reference paper since it uses a QoI defined as $J=\int_\Omega g\,[1,1]^T\cdot\mathbf{u}d\Omega$. The additional weight $g=exp(-100(x-0.5)^2+(y-0.5)^2$ is used in order to target specifically the resolution in the centre of the cavity. While this is less relevant to aerospace engineers than the measure of drag for instance, it was found that the QoI chosen in [58] targeted the resolution of the near lid too strongly, hindering the ability to evaluate the performance of the SRNN-based error estimates.

This QoI leads to the source terms of the adjoint equation being $\frac{dJ}{dp}=0$ and $\frac{dJ}{d\mathbf{u}}=[1,1]^T\cdot g$. Since the exact value of the QoI is unknown, it was estimated using a grid converged value of the QoI.

In an attempt to address the discontinuous boundary conditions encountered in the usual lid-driven cavity flow. The chosen boundary conditions differ from the usual implementation. The lid velocity is not uniformly distributed across the top boundary and instead follows the distribution from Equation 6.5. This is proposed in [58] with $\delta=0.05$.

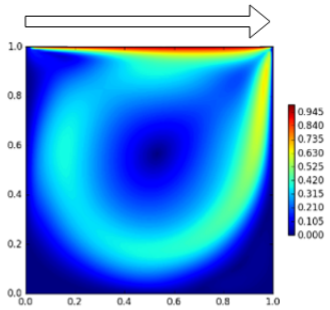


Figure 6.1: Layout of the lid-driven cavity flow

$$s(x) = \begin{cases} 0 & \text{if } x = 0, \\ \frac{1}{1+\exp\left(\frac{x}{\delta-1}+\frac{1}{\delta}\right)} & \text{if } 0 < x < \delta, \\ 1 & \text{if } \delta \leq x \leq 1-\delta, \\ \frac{1}{1+\exp\left(\frac{1}{\frac{(1-x)}{\delta}-1}+\frac{1}{\frac{(1-x)}{\delta}}\right)} & \text{if } 1-\delta < x < 1, \\ 0 & \text{if } x = 1 \end{cases} \tag{6.5}$$

The lid-driven cavity flow was studied for a range of Reynolds numbers from 50 to 1200. This chapter focuses on Reynolds numbers of 250, 500 and 900. The problem leads to the solution of more complex flow features, such as the presence of re-circulation cells. These features increase the difficulty of the reconstruction task for the SRNN. In combination with a QoI describing a flow characteristic about which very little is known, it is expected that this test case is more representative of the realistic behaviour of the reference and SRNN output error estimates.

6.3. Reference error estimator performance

Similarly to section 5.3, an analysis of the reference error estimators was made. This helps in the comparative evaluation of the SRNN-based error estimation, with respect to the reference error estimates.

First, the accuracy of the output-based error estimation is evaluated. This is seconded by the analysis of the error in output-based error estimation. The comparison of the error indicators and the impact of the proposed method on the mesh adaptation process is done in subsection 6.5.3.

6.3.1. Reference output error estimation

The reference error estimates of section 5.3 are also used here. Their output errors are shown in Figure 6.2 to Figure 6.4. These figures also show the QoI difference between the coarse and once refined meshes, $dJ_{h-H} = J_h - J_H$. This quantity is useful for explaining the converging behaviour of the estimations.

At the three Reynolds numbers considered, similar behaviours are observed. The absolute value of the output error is plotted here in order to be able to be presented in a log-log plot. The kinks in the output error convergence can be explained by the QoI being overestimated and underestimated at regular intervals. Nonetheless, the output errors decrease with refinement. This is the expected behaviour and greatly resembles some of the behaviours seen in the unsteady Burgers' equation and the advection-diffusion problem albeit the convergence rate of the output error is smaller. The slope of 2 is observed for the true output error in Figure 6.5. The output error in this figure is plotted against the characteristic cell size dx.

Moreover, one can see that both the primal and adjoint residual formulations are able to estimate quite accurately the trend of the true output error, despite not being able to accurately reconstruct the oscillations in the true output error. The accuracy of the reference error estimators is also noted in Figure 6.6 to 6.8. It appears that while the output error is inaccurate, the error in output error estimation decreases at a similar rate, should a trend-line be drawn.

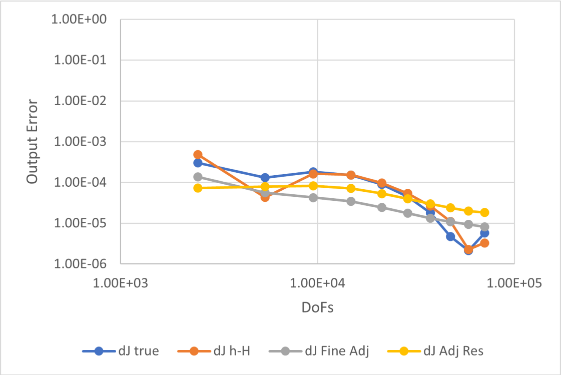


Figure 6.2: Output error for the reference error estimators for the lid-driven cavity at $Re = 250$

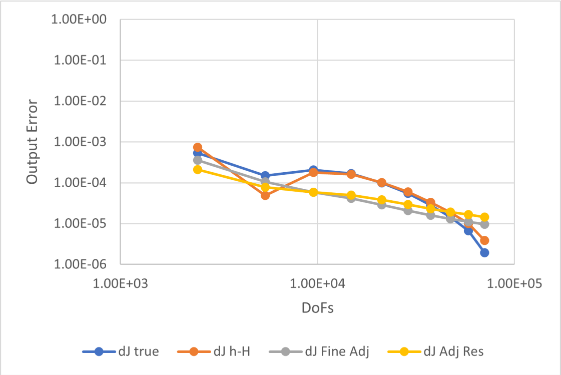


Figure 6.3: Output error for the reference error estimators for the lid-driven cavity at $Re = 500$

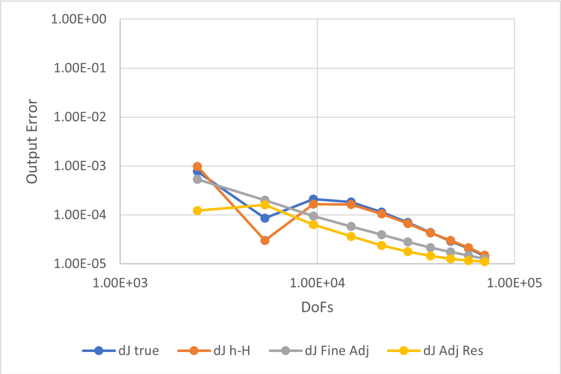


Figure 6.4: Output error versus DoFs for the reference error estimators for the lid-driven cavity at $Re = 900$

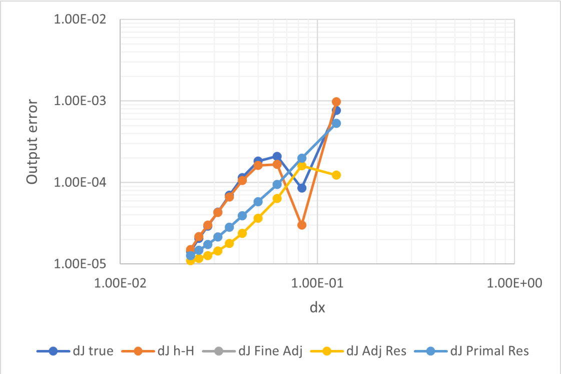


Figure 6.5: Output error versus characteristic cell size for the reference error estimators for the lid-driven cavity at $Re = 900$

It should also be noted that the increase in Reynolds number improves the performance of the reference output error estimates. In the future, it should therefore be investigated whether the performance of the reference error estimators further improves with respect to the QoI for higher Reynolds number flows since the flow in the centre of the cavity appears to become less dependent on the mesh refinement.

Finally, the adjoint residual formulation of the remaining error seems to perform better for lower levels of solution resolution, in particular for $Re = 250$. Beyond a uniformly refined grid of resolution 56^2 it appears that the fine adjoint error estimator, and its approximation the primal residual form, are able to produce a higher level of accuracy in the output-based error estimation. This further strengthens the argument to use a hybrid approach using both the primal and adjoint residual forms proposed in [71]. The computational cost of evaluating both of these residuals must be compared though, and this is partially addressed in the discussion of the CCM in the closing sections of this chapter.

6.3.2. Reference error indicators

Figures 6.9 to 6.12 show the distribution in space and also in magnitude of the adaptation parameters obtained for $Re = 250$ and $Re = 900$ for the fine adjoint-based error estimate. In these figures, the log of the adaptation parameter is taken. This highlights the complex nature of the interaction between the adjoint solution and the strong residual which makes it difficult to determine a priori the areas which require adaptation.

In particular, the region near the lid requires refinement whereas the bottom wall region could benefit from coarsening. This is mostly due to the strong residual being large and small, respectively, in those regions. However, the main flow structures of the primal can still be observed near the right wall. The main contribution of the adjoint appears to be in the centre of the cavity. This would be expected due to the strong focus of the QoI to evaluate the solution in the centre of the cavity.

Finally, these figures also highlight the change in error indicator distribution with increased refinement. For instance, the number of elements whose error indicator is the largest does not seem to decrease signif-

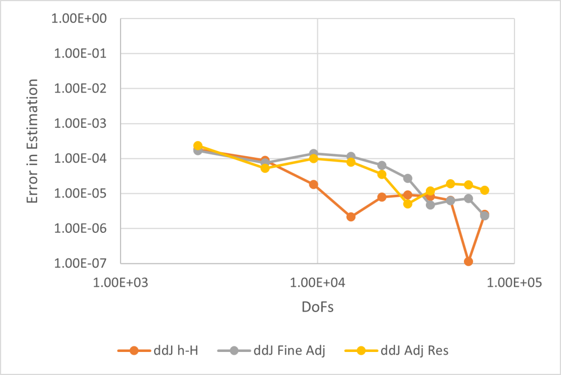


Figure 6.6: Error in output error estimation for the lid-driven cavity at $Re = 250$

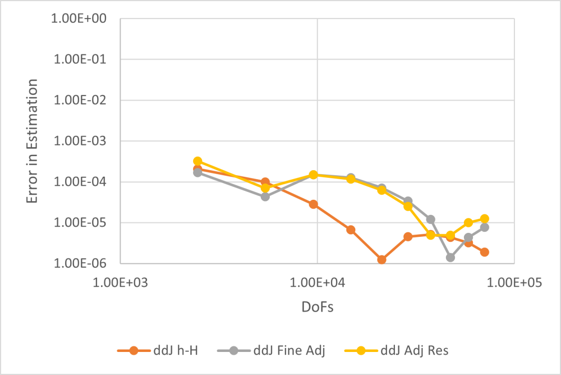


Figure 6.7: Error in output error estimation for the lid-driven cavity at $Re = 500$

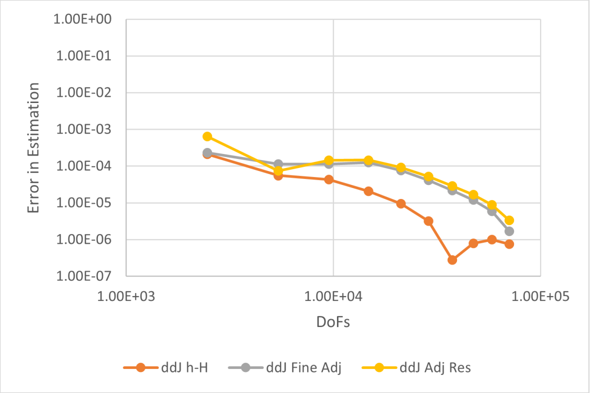


Figure 6.8: Error in output error estimation for the lid-driven cavity at $Re = 900$

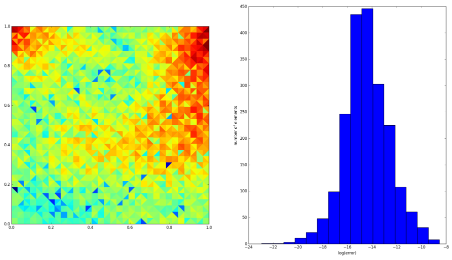


Figure 6.9: Locations and error distribution for the fine adjoint error estimate at $Re = 250$ for $n = 16^2$ elements

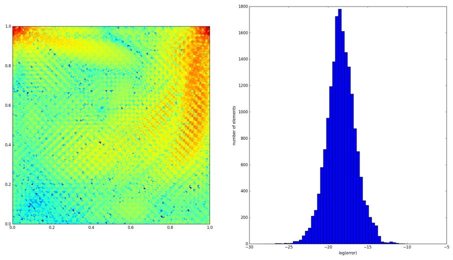


Figure 6.10: Locations and error distribution for the fine adjoint error estimate at $Re = 250$ for $n = 48^2$ elements

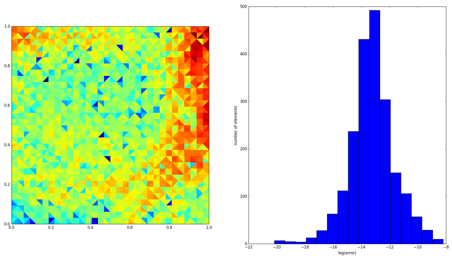


Figure 6.11: Locations and error distribution for the fine adjoint error estimate at $Re = 900$ for $n = 16^2$ elements

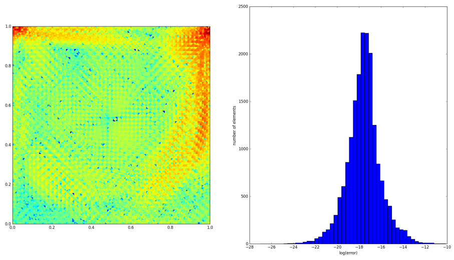


Figure 6.12: Locations and error distribution for the fine adjoint error estimate at $Re = 900$ for $n = 48^2$ elements

icantly on the right tail of the distribution. This would mean that for a number of regions, an increase in refinement does not yield a large decrease in output error contribution. These are considered outliers and make fixed fraction mesh adaptation algorithms less efficient in this particular case.

6.4. Machine learning implementation

Two SRNN architectures were implemented here. The first is the same as the one for the unsteady Burgers' equation [48]. The detailed architecture for the an up-sampling factor of 2 is given in Figure 6.13. The last layer is the same PixelShuffle layer. This step enables the growth of the neural network in depth and better reconstruction, by focusing the conversion of deep data in the neural network to detail in the output in a single step.

This architecture is very simple to train since the main, and only, hyper-parameter to consider is the learning rate of the ADAMS optimiser implemented in PyTorch. This is due to the fact that the momentum decay coefficients of the optimiser can be left at default values of 0.9 and 0.99 without greatly impacting either the learning process or the outcome of the network. The chosen learning rate should be adjusted dynamically in order to train faster. However it was found that a learning rate of 0.0005 worked very well as it both enabled fast training and accurate reconstruction with respect to the MSE loss function. This loss function has been the reference in literature for SRNN due to their roots in image processing. It should be investigated whether using more physics-based loss functions would benefit the training and outcome of the SRNN. The batch size was chosen to be 128 since this would be large enough to have statistically significant batches and make the training faster in order to improve the network more regularly [73]. Moreover, the CNN required significantly less tinkering and epochs to train than the other architecture considered, the cGAN. This increase in robustness in training is expected to lead to a less accurate and generalisable reconstruction than the cGAN [39].

The second architecture considered here was implemented in order to explore the potential benefits from more complex, and more accurate SRNN [39]. The conditional Generative Adversarial Network (cGAN) [53] makes use of two competing ANN: a generator tasked in performing the super-resolution, and a discriminator tasked in determining the validity of the reconstruction. The conditional aspect of this network is that the generator takes in additional information to reconstruct the fine space adjoint, here the Reynolds number

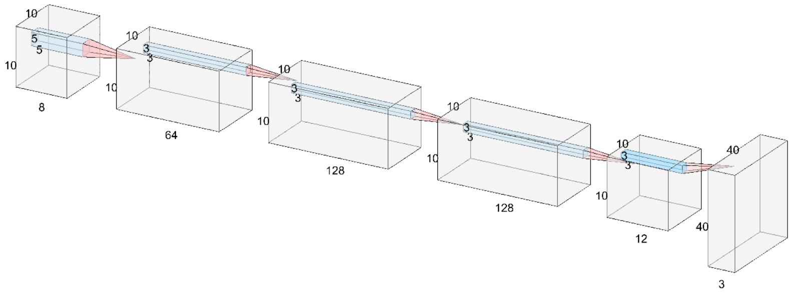


Figure 6.13: Architecture of the 2D CNN

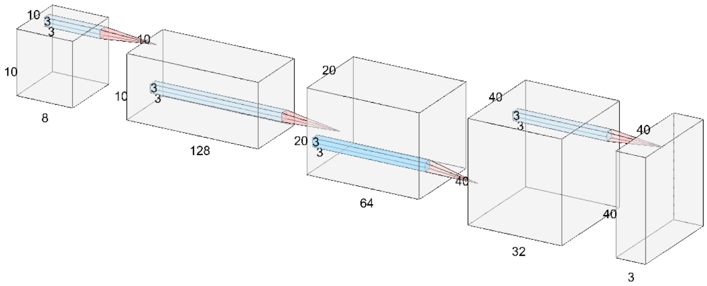


Figure 6.14: Architecture of the 2D cGAN generator

which is referred to as a label. Moreover, the discriminator is also provided with this label as well as the low resolution input that the generator uses. As documented in [39], this approach gives much more accurate reconstructions than the static CNN for HIT, in particular for higher up-sampling factors.

However, this architecture proved much more difficult to train since it requires that both the generator and discriminator are competitive by ensuring that both ANN learn their task at similar rates. This required the use of a far smaller learning rate $lr = 0.00005$ which made the training far slower. In addition to lowering the learning rate, the decay of the momentum gradient was lowered to values of 0.5 and 0.99. This further slowed the training process but ensured that neither the generator or the discriminator were able to outperform the other ANN. The slower training is seen in the need for at least 1000 training epochs in order to reach an MSE loss of $\mathcal{O}(10^{-3})$ whereas the CNN reaches a MSE of $\mathcal{O}(10^{-4})$ in fewer than 400 epochs.

The training data was obtained by solving the steady lid problem at the three Reynolds numbers $Re = 250, 500, 900$ for different levels of refinement with a random distribution of the data of 30% in the testing data-set and 70% in the training data-set. A fixed interrogation window of size 32 was then sampled from the solution. This extract of the solution was subsequently sub-sampled at the chosen super-resolution factor in order to form the low resolution-high resolution training pair. For each solution, four samples were taken in order not to introduce bias towards the amount of data at each refinement level. Moreover, each solution was rotated and taken the symmetry in an attempt to create a larger, more diverse data-set where the ANN does not learn the reconstruction of a specific lid-driven cavity problem but the generalised form.

In hindsight, this latter part was not necessary. While it did produce good results and should be able to generalise better. It greatly increased the complexity of the learning problem and as such led to sub-optimal results for the amount of computing power required to train the SRNN and reconstruct the fine adjoint.

Moreover, the restriction in the number of studied Reynolds numbers did not benefit the training of the cGAN while it did help the CNN to produce better results. The training task was attempted for a larger range of Reynolds numbers from 50 to 1200 and, for this variety and thus richer training data-set, the capacity of

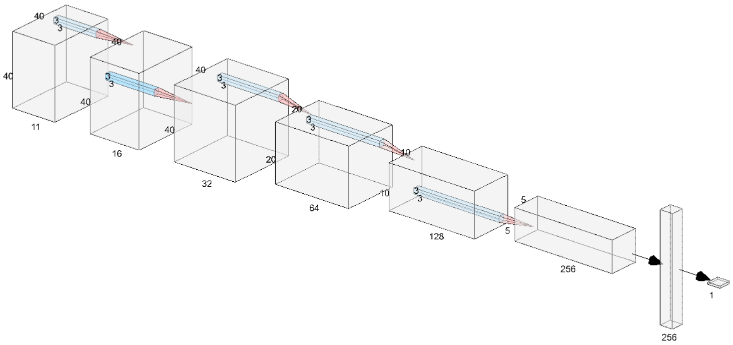


Figure 6.15: Architecture of the 2D cGAN discriminator

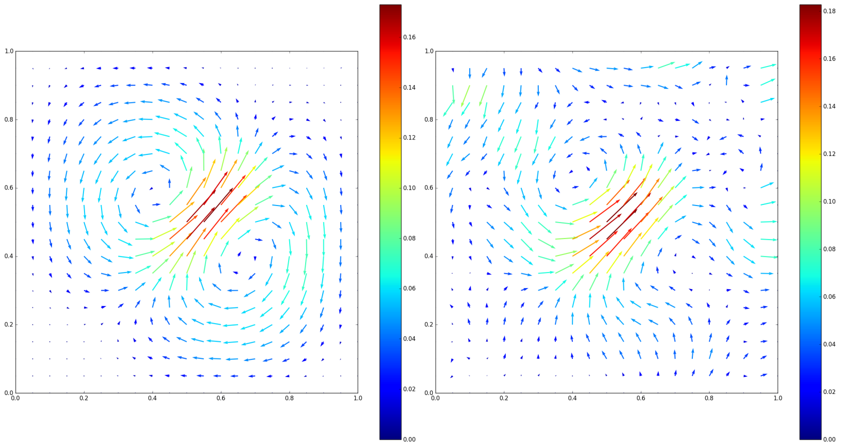


Figure 6.16: Example of the CNN erroneous reconstruction of the fine space adjoint

the cGAN to reconstruct the fine adjoint was significantly better than the CNN. The latter would misidentify certain flow features in the coarse adjoint solution and result in erroneous reconstructions such as the ones shown in Figure 6.16. In the figure, the right side of the cavity is erroneously reconstructed with the adjoint solution seemingly not respecting the no-slip wall boundary conditions.

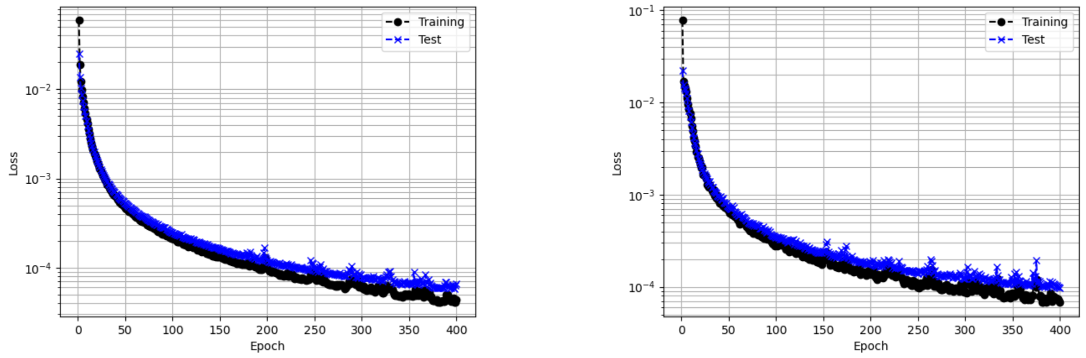
The losses for the CNN and cGAN are shown in figures 6.17 and 6.18, respectively. For the CNN, one can observe that the higher up-scaling factor requires more training in order to reach the same loss level. Moreover, the learning rate should be lowered if one should desire to achieve a lower training and testing MSE loss since similar signs of over-fitting and unproductive training were observed after more than 500 epochs were performed with this learning rate. The signs are similar to the one observed in Figure 5.5.

For the cGAN, the training was more tedious and the plots show the generator and discriminator loss. The actual MSE was measured with respect to batches of the testing data-set and was found to be in $\mathcal{O}(10^{-3})$. While higher than for the CNN, it was desired to compare the two architectures also with respect to the computational cost to train and with respect to this measure, the cGAN is outperformed by the CNN.

A more appropriate loss function could also be chosen, such as the ability to accurately estimate the output error. However, this would require the evaluation of the adjoint-weighted residual in the fine space for each element of the batch which is not trivial and can become computationally intensive. Other cheaper physics-based loss functions could be considered, such as the evaluation of the vorticity or the conservation of mass over the considered domain. This could potentially also enable the network to produce results that are more consistent with the physics of the problem.

6.5. SRNN-based error estimate performance

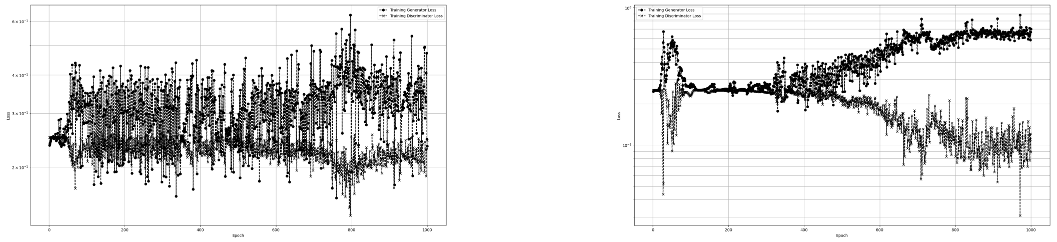
The performance of the proposed method has to be compared on three points: the reconstruction of the fine adjoint task, its impact on the output error estimation, and the impact on the error indicators. While the first



(a) Training losses for the 2x CNN

(b) Training losses for the 4x CNN

Figure 6.17: Losses of the super-resolution CNN for the 2D lid-driven cavity problem



(a) Training losses for the 2x cGAN

(b) Training losses for the 4x cGAN

Figure 6.18: Losses of the super-resolution cGAN for the 2D lid-driven cavity problem

seeks to evaluate the ability of the ANN for the super-resolution tasks, the second provides an estimate for the impact of the more global behaviour of the SRNN in the mesh adaptation process, whereas the ability to identify the main sources of error ensures that the mesh adaptation process remains productive at difference mesh refinement levels.

6.5.1. Fine adjoint reconstruction

First, the fine adjoint reconstruction is evaluated for the four ANN implemented in this chapter. The results are shown in figures 6.19 to 6.22. One can observe that for sufficiently resolved coarse adjoint solutions, all the methods are able to reconstruct the main features of the fine adjoint. The 2x up-scale factor SRNNs are able to more accurately reconstruct the fine adjoint solution for under-resolved solutions. The 4x SRNNalready shows an inability to reconstruct the fine adjoint accurately at a resolution of 16^2 cells

Nonetheless, the amount of information reconstructed for Figure 6.19 is remarkable and it is expected that it could provide an accurate output error estimate since the main features are well reconstructed both in direction and magnitudes. The performance is also expected to degrade for increasing levels of refinement since the construction of the training data-set introduces great bias towards learning from coarse solutions. This is due to the limited size of the interrogation window but also that the highest starting resolution included in the training data-set is 120^2, which is sub-sampled. For the reconstruction task, starting resolutions of 88^2 are considered, meaning that the network is attempting to reconstruct sub-grid scale information that it has never encountered in the training process. This problem is exacerbated for the higher up-sampling ratios.

6.5.2. Output error estimation

The output error estimate and error in output error estimate is given in figures 6.23 and 6.24 for the case $Re = 250$, figures 6.25 and 6.26 for $Re = 500$, and figures 6.27 and 6.28 for $Re = 900$. The accurate estimation of the output error informs the user of the degree of accuracy of the current solution with respect to the finer space solution. In the proposed method, the fine adjoint solution is reconstructed from the coarse adjoint

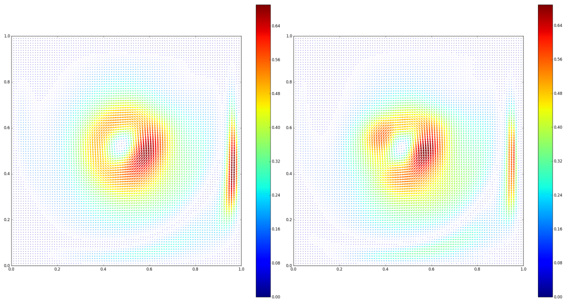


Figure 6.19: Comparison between the reconstructed (right) and fine adjoint solutions (left) for the CNN(2x) at $Re = 900$ and $n = 48^2$ elements

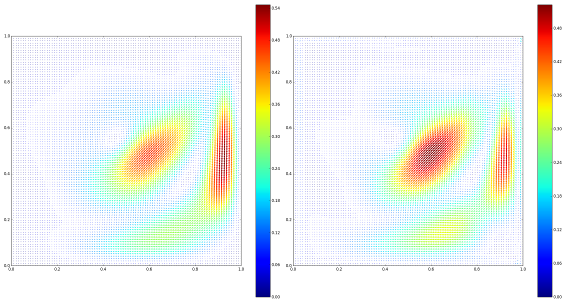


Figure 6.20: Comparison between the reconstructed (right) and fine adjoint solutions (left) for the cGAN(2x) at $Re = 250$ and $n = 48^2$ elements

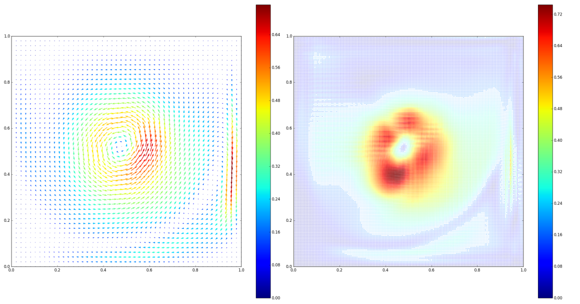


Figure 6.21: Comparison between the reconstructed (right) and fine adjoint solutions (left) for the CNN(4x) at $Re = 900$ and $n = 48^2$ elements

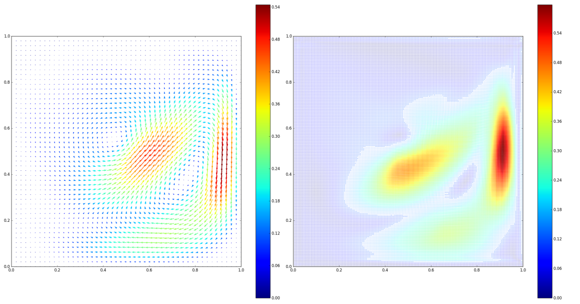


Figure 6.22: Comparison between the reconstructed (right) and fine adjoint solutions (left) for the cGAN(4x) at $Re = 250$ and $n = 32^2$ elements

solution, and then used to weight the strong residual for the output error:

$$\delta J \approx -\int_{\Omega} \Psi_{NN} R_h(U_h^H) d\Omega$$

For the lowest Reynolds number shown here, it can be seen that despite the inaccuracies in the reconstruction of the fine adjoint mentioned in the previous section, the ability to estimate the output error is not hindered. For low resolutions, all methods, apart from the 2x CNN are able to estimate the true output error, within an order of magnitude. In particular the cGANs perform very well for the first three uniform refinement levels. The CNN tend to underestimate the output error, which is not desired since it could lead to the interruption of the mesh adaptation iterations prematurely. Moreover, despite the erroneous fine adjoint reconstruction the 4x cGAN appears to accurately estimate the output error across the range of refinement levels evaluated.

The two main under-performing ANN are the 4x CNN and the 2x cGAN. Throughout the range of meshes, the former greatly underestimates the output error. Moreover, for the unseen range of refinement levels, it is unable to produce output error estimates which outperform the 4x cGAN. The 2x cGAN performs the best out of the four ANN for the first three refinement levels. However, it is then unable to estimate the output error in a sensible way. This could be due to the inaccurate reconstruction of the fine adjoint and its interaction with the residual field which leads to the inaccurate output error estimation. Figure 6.5 highlights the inaccuracies of the output error estimation for the SRNN since the slope of 2 present in the reference methods is not observed. However, the trend for the decrease of output error with increasing refinement is observed and the CNN-based error estimates perform similarly to the fine adjoint error estimate.

When comparing the error in output error estimation in Figure 6.24, the 4x cGAN appears to perform far worse for under-resolved adjoint solutions. This figure further highlights the 2x cGAN inability to provide an accurate output error estimation for fine meshes. Other than this latter method, the three proposed SRNN are able to perform on par with the fine adjoint-weighted residual output error estimate for sufficiently resolved adjoint solutions. It must also be noted that the generalisation to finer scales than those used in training is not successful as they all struggle to achieve an output error estimation accuracy below 10^{-5}.

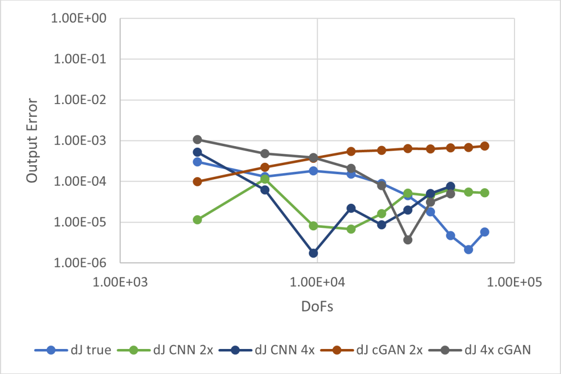


Figure 6.23: Comparison of the output error estimate for the lid-driven test case using SRNN at $Re = 250$

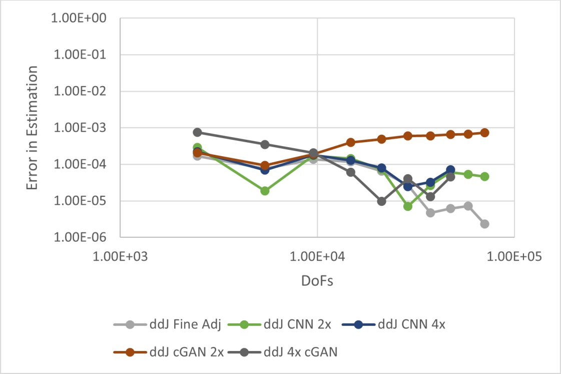


Figure 6.24: Comparison of the error in output error estimate for the lid-driven test case using SRNN at $Re = 250$

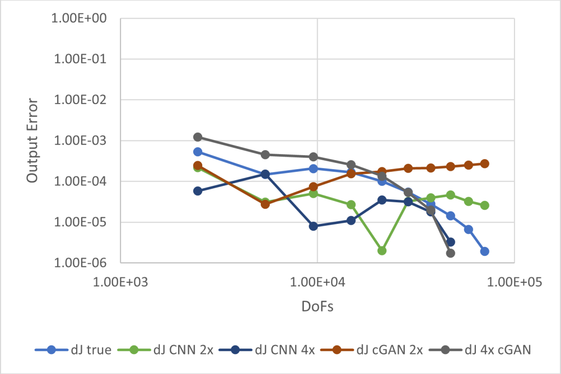


Figure 6.25: Comparison of the output error estimate for the lid-driven test case using SRNN at $Re = 500$

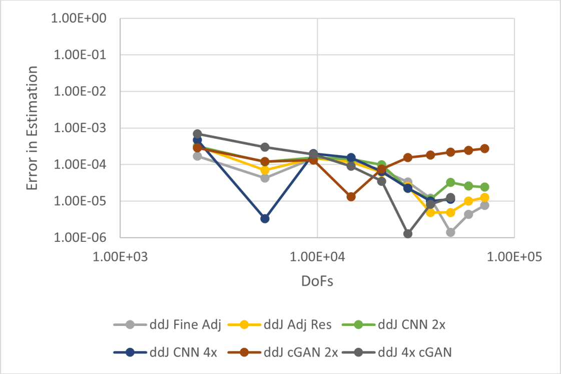


Figure 6.26: Comparison of the error in output error estimate for the lid-driven test case using SRNN at $Re = 500$

For a Reynolds number of 500, much of the same observations can be made, as shown from Figure 6.3 and Figure 6.26. On the one hand, both the 2x SRNN perform very similarly for low mesh resolutions, albeit notably underestimating the output error. However, the CNN is more apt in estimating the output error for finer meshes than its cGAN counterpart. The 4x ANN, on the other hand, perform better for higher resolution solutions, despite not having encountered them in the training process. At the lower resolution, only the cGAN is able to reconstruct the adjoint accurately enough to lead to an accurate output error estimate.

For $Re = 900$, only the 4x cGAN stands out by being able to reproduce the true output error at an accuracy of less than an order of magnitude. This is on par with the reference output error estimates for this case and highlights the sufficiently accurate reconstruction of the adjoint in the regions of high values of residuals. The other SRNN show great limitations in their ability to correctly estimate the output error. They all greatly under-estimate the value of the output error, and although for finer meshes the 4x CNN is able to estimate accurately both the trend and value of the output error, this inability to perform well for under-resolved and sufficiently resolved solutions demonstrates an inaccurate fine adjoint reconstruction.

6.5.3. Cell indicator estimation

It is arguable that the localisation of the sources of output error is more important to the mesh adaptation process than the output error estimation. However, like the output error estimation, it requires the accurate reconstruction of the fine adjoint, in particular the global trend and relative contribution of each element.

The lid-driven cavity problem is dominated by the singular behaviour of the solution on the lid extremities as well as the fractal-like behaviour of the flow patterns in the bottom corners [40]. For this reason, the actual error indicators are dominated by near corner high values, despite the efforts being made in the choice of the QoI to exclude these regions. It appears that the strong residual values are simply very large in these regions, large enough to be flagged as main contributors to the output error. In order to make the plots more insightful, the log of the output error indicators is taken.

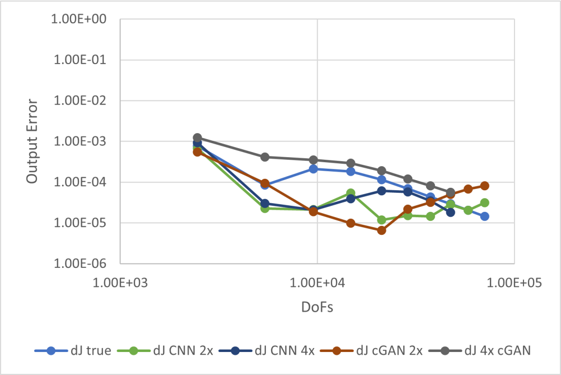


Figure 6.27: Comparison of the output error estimate for the lid-driven test case using SRNN at $Re = 900$

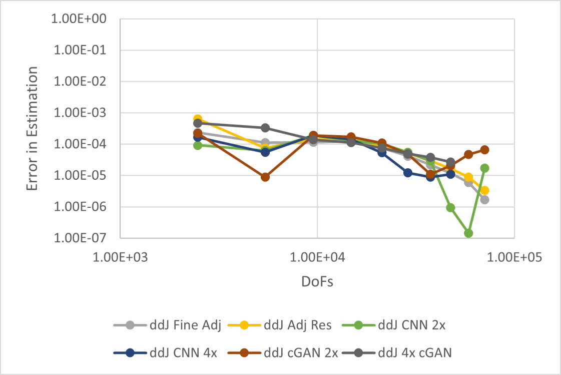


Figure 6.28: Comparison of the error in output error estimate for the lid-driven test case using SRNN at $Re = 900$

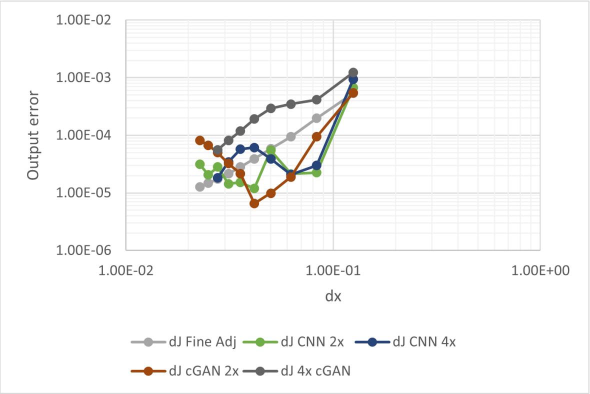


Figure 6.29: Output error versus characteristic cell size for the SRNN error estimators for the lid-driven cavity at $Re = 900$

An under-refined and well refined solution pair will be shown for each tested ANN. Figures 6.30 and 6.31 show the error indicators for the fine adjoint output error estimate and the 2x CNN at $Re = 900$. At this Reynolds number, the features of the adjoint is limited to a main vortical structure in the centre of the cell and a single high adjoint velocity magnitude area near the right boundary. This is recognised in both levels of refinement in the error indicators, on top of the lid-corners, as being areas most sensitive to the flow resolution. These are thus flagged for refinement. The CNN identifies a too large region as being a large contributor to the output error with respect to the reference error estimator.

With a starting mesh of 48^2, similar cells are flagged for refinement. However, a number of other cells in areas contributing less to the output error also are flagged. This refinement level also allows highlighting of the inability of the network to infer flow structures it has not seen in training. This can be seen in the erroneous estimation of the error indicators in the lower right corner, in which an additional circulation bubble appears with refinement.

For a mesh of 16^2, the 4x CNN is unable to accurately estimate the locations of the largest sources of errors, as shown in Figure 6.34. This is related to the erroneous reconstruction of the fine adjoint as previously mentioned. However, beyond this issue, far too many cells are erroneously identified as large contributors to the output error. This would be greatly problematic in a mesh adaptation iteration since this would greatly increase the computational cost of the solution of the problem without necessary resulting in an increase in output accuracy. Moreover, since the upper right corner is still identified correctly as a source of error, it is possible to infer that the primal residual being correct is more important in this case than the accuracy of the fine adjoint.

The finer mesh sheds a different light since a lower level of reconstruction is required. The reconstruction is therefore more accurate, as would be expected from the more accurate output error estimate shown in the previous section. At this refinement level, the problem is no longer the identification of the flow features or the anticipation of flow features appearing in the finer solutions but the accurate reconstruction of the present features. This is not performed extremely well in this case. The centre of the cavity is misidentified as a region whose contribution to the output error is larger than in reality. This behaviour is observed across the range of Reynolds numbers considered.

The 2x cGAN fairs better than its CNN counterpart for the low resolution reconstruction of the fine adjoint for a mesh resolution of 16^2. This results in also a slightly more accurate reconstruction of the error indicators. The main difference occurs in the swirl towards the centre of the cavity where the error contribution is under-estimated. However, the main sources of error are identified both in location and magnitude. Moreover, the example chosen here of $Re = 250$ is significantly more difficult due to the more complex flow structures in the adjoint solution.

The improvement in accuracy generalises well to the finer starting meshes as seen in Figure 6.33. The areas of low contribution to the output error are well identified as being most of the left boundary and the lower left corner in general. Moreover, more complex areas of output error contribution are correctly identified, such as the segments radiating from the left lid extremity. These were not well identified by the CNN and show the ability of the cGAN to outperform the static CNN.

Finally, the performance of the 4x cGAN is shown in Figure 6.36 and 6.37. In the first figure, the error indicators for $Re = 250$ and a mesh of 16^2 cells is shown. In this configuration, The cGAN is able to perform very well in the error indicator estimation, despite a less ideal output-based error estimation and fine adjoint reconstruction. This means that the regions of interest where the strong residual is large are well reconstructed. It identifies the corners of the lid as being the main sources of error and the main swirl in the primal solution to also be the most important area for refinement in order to increase the accuracy of the QoI. This specific case would be very suitable for a mesh adaptation routine, as the error indicators follow the structure one would expect in this problem. Since the QoI is the accurate representation of the velocity field in the centre of the cavity, the upstream field must be well resolved.

Figure 6.37, shows the decrease in performance of the method as refinement levels tend to the upper bound of the resolution seen in the training. Certain features are recreated in the error indicator field that do not exist in reality. The best example is this line at the $y = 0.35$. It is unclear how such an artefact was computed since this discontinuity is not as visible in the adjoint reconstruction. This highlights the interaction between the adjoint and residual fields, which is not trivial and difficult to estimate.

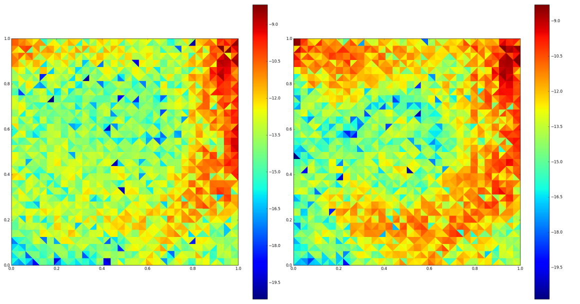


Figure 6.30: Error indicators for the CNN (2x)-based error estimate $Re = 900$ for $n = 16^2$ elements. Fine adjoint on the left and SRNN on the right.

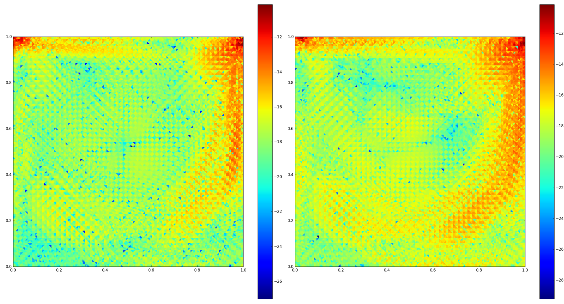


Figure 6.31: Error indicators for the CNN (2x)-based error estimate $Re = 900$ for $n = 48^2$ elements. Fine adjoint on the left and SRNN on the right.

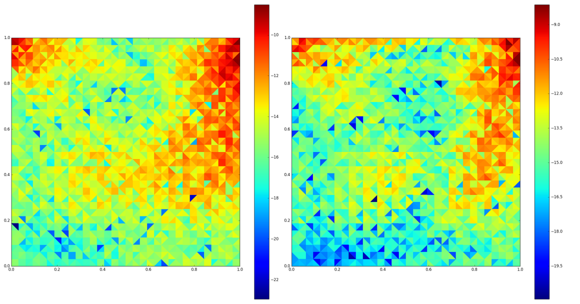


Figure 6.32: Error indicators for the cGAN (2x)-based error estimate $Re = 250$ for $n = 16^2$ elements. Fine adjoint on the left and SRNN on the right.

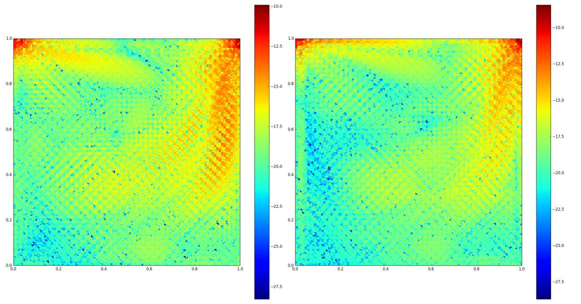


Figure 6.33: Error indicators for the cGAN (2x)-based error estimate $Re = 250$ for $n = 48^2$ elements. Fine adjoint on the left and SRNN on the right.

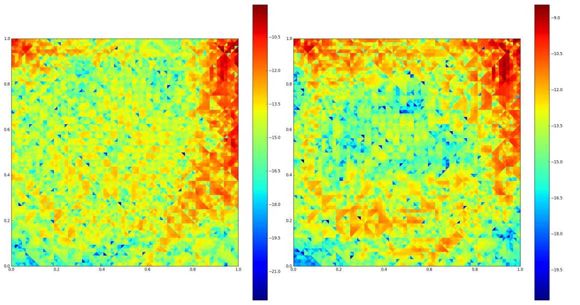


Figure 6.34: Error indicators for the CNN (4x)-based error estimate $Re = 900$ for $n = 16^2$ elements. Fine adjoint on the left and SRNN on the right.

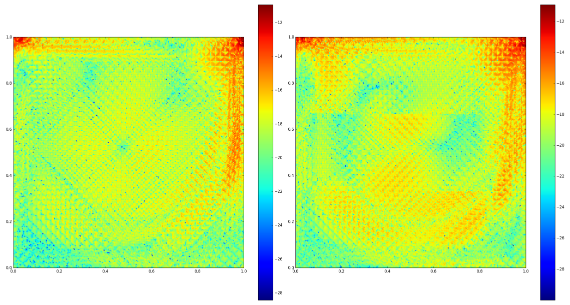


Figure 6.35: Error indicators for the CNN (4x)-based error estimate $Re = 900$ for $n = 48^2$ elements. Fine adjoint on the left and SRNN on the right.

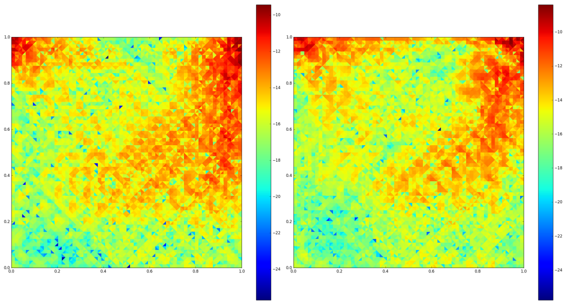


Figure 6.36: Error indicators for the cGAN (4x)-based error estimate $Re = 250$ for $n = 16^2$ elements. Fine adjoint on the left and SRNN on the right.

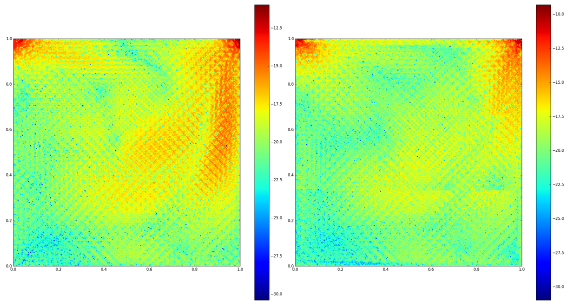


Figure 6.37: Error indicators for the cGAN (4x)-based error estimate $Re = 250$ for $n = 48^2$ elements. Fine adjoint on the left and SRNN on the right.

The analysis of the error indicators is more optimistic than the output error estimation for the use of SRNN of the adjoint. Out of the four demonstrated SRNN, only the 4x methods stand out as having a more difficult time with the estimation of the error indicators. While their performance for meshes of refinement 16^2 is at least on par with the lower upscale factors, this performance rapidly degrades compared to the more consistent behaviour of the 2x networks.

This leads to the conclusion that the 2x SRNN would be suitable for implementing in a mesh adaptation process. While the use of a finer adjoint solution could potentially lead to an increase in accuracy in output error estimation, it is not observed here. Moreover, the additional cost of evaluating a finer adjoint-weighted residual makes it more difficult to justify. The discrepancy of these results and the behaviour observed in [39] is most likely in the different nature of the flow. [39] looks at HIT, while the problem here is anisotropic with clearly identifiable features.

6.6. Proposed error estimate cost analysis

Finally, the same metric as in chapter 5 was used to evaluate the relative cost of SRNN-based output error estimation with respect to the reference error estimates. Whereas the increase in DoF in the previous test case was linear with mesh refinement, it is here quadratic and as such, is expected to further highlight the increase in computational cost in the estimation of the output error when the fine solution is computed, or when the 4x adjoint-weighted residual is evaluated.

Figures 6.38 and 6.39 show the evolution of the CCM with respect to the number of DoF of the coarse problem for $Re = 250$. Similar to the unsteady Burgers' equation, the error estimation methods employing the solution of the primal or adjoint on the fine mesh are significantly more expensive, as shown by the CCM ranging from 4 to 6 in Figure 6.38.

Looking specifically as the cheaper output error estimates, the reference and proposed output error estimation methods have similar CCM, as they are all overlapping in Figure 6.39. It increases greatly with the uniform refinement of the mesh. This means that 2x SRNN-based estimates are more likely to be a compelling case for the implementation in mesh adaptation due to the reduced cost, being on par or cheaper than the current employed methods in literature.

The CCM is also presented with respect to the accuracy of the output error estimate. This is shown in figures 6.40 to 6.42. These figures help highlight the complexity of the error estimation task for each method but also shows the relative increase in computational cost between the presented methods. For the three Reynolds numbers, all output error estimations methods have very similar computation costs and related increase in error estimation accuracy. However, as the mesh is refined, the computational cost of the output error estimation is similar to that of the primal solution. This means that the inaccuracy of the 2x cGAN would lead to a far greater computational cost to reach a similar accuracy level to the other error estimates.

The 2x CNN is able to both provide error estimates of the same accuracy as the reference methods and reduce the computational cost of the error estimation. It is able to reconstruct the fine adjoint output error estimate accuracy at a fraction of the cost. This cost is similar to the methods implemented in practice. However, the approximations used in practice introduce more inaccuracies and as such would lead to a greater computational cost in the form of additional mesh adaptation iterations [55].

6.7. Assessment of the proposed method

The lid-driven cavity flow with a QoI targeting the resolution of the flow in the centre of the cavity proved a very complex QoI to estimate for both reference and SRNN-based error estimates. This highlighted the reduction in cost in using SRNN but also the decrease in accuracy of the output error estimation due to the error in adjoint reconstruction. Most importantly for mesh adaptation though, the error indicators could be well estimated from the SRNN, fine adjoint reconstruction. This supports the idea that the 2x up-sampling methods described here should be implemented in a mesh adaptation method in order to see if this ability to estimate the correct adaptation parameters is reflected outside of uniform mesh refinement.

However since the value of the error indicators is incorrect, it would probably prevent the use certain mesh adaptation algorithms such as the ones based on output error thresholds. Thus, a fixed fraction approach would be most relevant, despite its shortfalls and the mentioned lesser performance it would have in this particular test case. This would most likely also require an architectural change of SRNN, since the mesh adaptation would introduce a non-uniform mesh. This could be addressed through the use of a U-ResNet

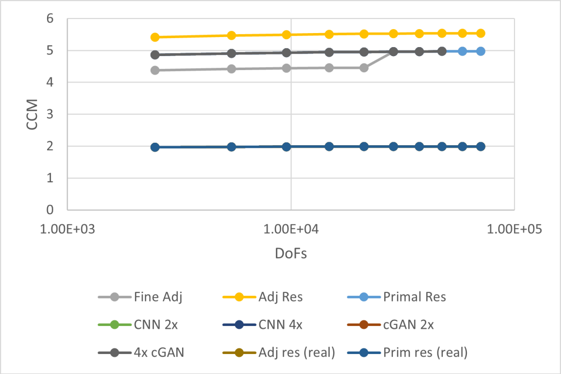


Figure 6.38: Computational cost metric versus DoF for the lid-driven cavity flow at $Re = 250$

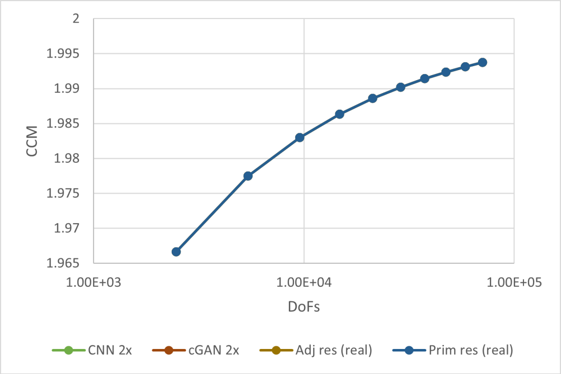


Figure 6.39: Computational cost metric versus DoF for the lid-driven cavity flow at $Re = 500$

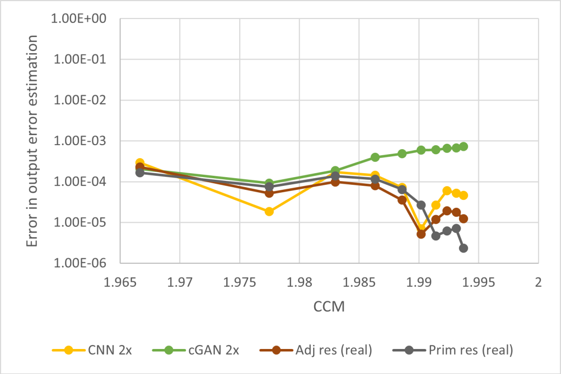


Figure 6.40: Computational cost metric vs error in output error estimate for the lid-driven cavity flow at $Re = 250$

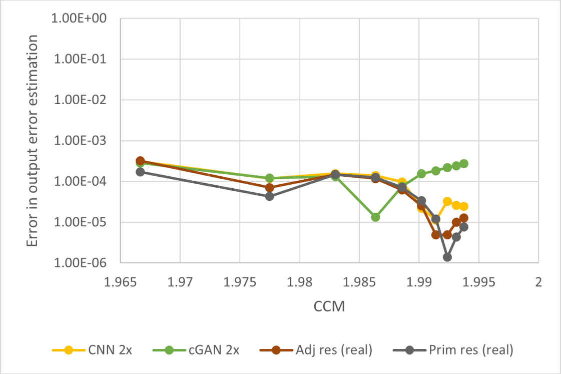


Figure 6.41: Computational cost metric vs error in output error estimate for the lid-driven cavity flow at $Re = 500$

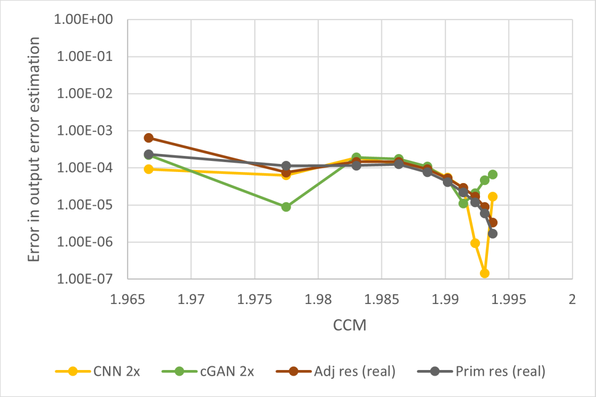


Figure 6.42: Computational cost metric vs error in output error estimate for the lid-driven cavity flow at $Re = 900$

[75, 78], which was specifically developed to tackle unstructured data while retaining the fully convolutional nature of the ANN.

Due to the added difficulty of training the cGAN and the limited advantage of using far finer meshes for the fine adjoint solution, the use of the cGAN is difficult to justify for the task of fine adjoint reconstruction.

Experience was also gained in the ability to construct an SRNN capable of accurately reconstructing the adjoint for the estimation of adaptation parameters. The training process of the CNN in particular showed that it was inexpensive to train,since large amounts of information could be generated from limited number of samples. This was achieved by taking windows of data inside the solution. Moreover, while the use of symmetries and rotations of the data did lead the network to misidentify certain flow features, it also allowed greater generalisation for the unseen mesh refinement levels.

SRNN-based output error estimation is therefore a strong candidate for the reduction in computational cost of adjoint-based error estimation. It appears to be particularly relevant for flows far from prescribed boundary conditions. It could also be very powerful in helping to estimate the error in early mesh adaptation iterations, when the solution is potentially under-resolved.

Finally, the proposed CCM demonstrated the increase in computational cost when higher super-resolution is performed. This increase would make the adoption of 4x SRNN-based error estimation more difficult to justify. However, an approach such as the one proposed in the previous chapter of super-resolution and sub-sampling could enable higher accuracy in the output error estimation, while limiting the computational cost of evaluating the adjoint-weighted residual. For the adjoint reconstruction task, it is thus advised to use lower up-scaling factors and simpler, more robust SRNN architectures such as the 2x CNN.

Nonetheless, the performance in reconstructing the primal solution using SRNN shown in [39] would make the use of 4x cGAN attractive for the reduction of the storage footprint of the unsteady primal solution. In practice, the entire solution is stored during the forward in time solution of the CFD simulation. The individual solutions are then retrieved during the adjoint solution. The SRNN would then enable the reduction of the size of the primal solution by a factor 4^n, with n the dimension of the problem.

7 Conclusion

This chapter aims to provide the conclusion to the findings of this thesis and answer the main research questions as well as the secondary research objectives as formulated in the opening chapters of the report. However, an emphasis is made on the recommendations and future work section, which highlights the necessary steps to be conducted in the evaluation of the proposed method in order to implement it successfully in a mesh adaptation process.

7.1. Conclusion

The thesis highlighted certain limitations in output-based mesh adaptation, in particular related to output error estimation methods. This led to a proposal to use of Super-Resolution Neural Network (SRNN) to potentially solve significant problems in adjoint-based output error estimation: the large storage footprint and the computational overhead. The latter point is the focus of this thesis since the compression and reconstruction of the primal data using ANN has shown already to be effective for primal flows [39]. The evaluation of the use of SRNN for the reconstruction of the adjoint solution thus revolved around two main axes: accuracy and computational cost.

First, FEniCS implementations of manufactured solutions for the Poisson and Burgers' equation were used to validate the output-based error estimate. This led to the choice of higher-order polynomial elements, which were shown to provide a more accurate solution of the field as well as a consistent grid convergence of the primal and adjoint solutions. The validation cases also allowed the testing of certain characteristics of the output error estimate implementation. One such characteristic was the effect of using vertex-valued function spaces to describe the solutions, instead of piece-wise continuous quadratic or cubic elements. This was seen to affect the higher-order elements' ability to provide accurate output error estimates. Consequently, the rest of the results were generated using lower-order quadratic elements (Taylor-Hood elements for the Navier-Stokes case).

In a second step, an advection diffusion problem described in [17] was used to investigate the impact of considering different Quantities of Interest (QoI). The integral of the flux $-\nu\nabla u$ over the right boundary was compared to the integral of the scalar variable u over the complete computational domain. The main effect is on the adjoint system of equations. The flux QoI leads to discontinuous boundary conditions over the boundaries of the computational domain whereas the domain integral introduces a unit source term to the adjoint PDE. The discontinuity in the boundary conditions of the former leads to an ill-definition of the adjoint solution in the right corners. This prevents good estimations of the output error when the fine adjoint solution is coarse, and for a twice more mesh to prevent the error in output error estimation to converging to the grid-refined value of the output error. In [17] this circumvented by using a highly refined mesh. However, this is not possible in practice. In this thesis, it was thus chosen to use global QoI relying over volume integrals rather than boundary integrals.

The tests used to evaluate the proposed SRNN error estimation method were based on the 1D unsteady Burgers' equation and the lid-driven cavity flow. The Burgers' problem, although similar to the validation problem, differed through the inclusion of the time dependence. This also enabled the creation of a larger data-set by sampling randomly in time for the adjoint solution and corresponding instantaneous primal. Although the 4x CNN in the super-resolution task was unable to provide an accurate output error estimate, its lack of accuracy did not reflect on the error indicators. In contrast, the 2x CNN was able to provide a more accurate output error estimate. It is expected that with more thorough training, it could match the fine adjoint output error estimate. For under-resolved solutions, both SRNN were able to provide accurate output error estimates. This could be the most promising in practice, due to the ability of the SRNN to provide sufficiently accurate adjoint fields at low costs.

The second SRNN case was the steady lid-driven cavity. This proved to be far more complex due to the solution of the Navier-Stokes equations, performed with a mixed-element function space, and the need to devise a new QoI in order to limit the impact of the singularities and discontinuous boundary conditions of the primal problem. The chosen QoI was $J = \int_\Omega g\mathbf{u}d\Omega$, with $g = exp(-100[(x-0.5)^2 + (y-0.5)^2])$. This focused on the evaluation of the primal in the centre of the cavity. This QoI proved difficult to estimate for the error estimators. Nonetheless, the SRNN were able to reconstruct the fine adjoint field from the coarse adjoint solution sufficiently to enable the identification of the main adjoint features of the solution. The super-resolution task was particularly difficult in the near-wall regions and for lower Reynolds numbers. In the lower Reynolds number range, the primal solution appeared to be far more dependent on the mesh refinement level since the area of large velocity gradients intersected the area of interest in the QoI. While it was expected that the conditional Generative Adversarial Network (cGAN) would outperform the Convolutional Neural Network (CNN) in the reconstruction task (based on the capacity of each method to super-resolve Homogeneous Isotropic Turbulence (HIT)), this was not observed here. The anisotropic nature of the adjoint solution may be the reason the cGAN could not perform significantly better.

The proposed method showed very promising results in the estimation of the adaptation parameter. In particular for the 2x super-resolution, the main areas contributing to the output error were well identified. In a mesh adaptation procedure, this method is expected to perform well in reducing the output error. The performance of the mesh adaptation using these adaptation parameters will, however, be sub-optimal due to the identification of additional cells requiring refinement which were not found in the fine adjoint output error estimate.

A metric for the comparison of the relative computational cost of a combined primal and adjoint solution was created. This Computational Cost Metric (CCM) was used on both the Burgers' and lid-driven cavity problems. In the former, the performance of the 2x SRNN was on par with the reference estimates for the coarsely refined meshes. However, the CCM did not justify the use of this method for refined meshes due to the lack of accuracy in estimating the output error.

For the lid-driven cavity flow, the main takeaway was the increase in CCM for the 4x SRNN. It performed on par with computing the fine adjoint on a twice finer mesh, since the adjoint-weighted residual had to be evaluated on a mesh containing 16 times more elements than the coarse mesh. The inaccuracy of the 2x cGAN for fine meshes also made the use of cGANs difficult to justify. This was strengthened by the increased difficulty in training the cGANs efficiently.

Nonetheless, the 2x CNN was able to perform as well as the reference error estimators both on its accuracy of the output error estimation and the CCM. This would justify further exploration of the use of simpler and more robust SRNN for the reconstruction of the fine adjoint.

Finally, the results indicate that the proposed method of using super-resolution neural networks for the reconstruction of the adjoint would likely be successful in driving a mesh adaptation process. This is based on the ability of the SRNN-based output error estimates to estimate accurately the error indicators. However, this is accompanied by an error in the output error estimation of about almost an order of magnitude, which could lead to a greater number of mesh adaptation iterations.

7.2. Future work and recommendations

This thesis tested the use of SRNN in the context of adjoint-based error estimation. Biases were found which highlighted the importance of ANN training procedures. Further work could include the comparison of the SRNN-based output error estimates to adjoint-based output error estimates making use of prolongation operators.

The framework for this comparison has been created in OpenFOAM, through the creation of a PISO-based continuous adjoint solver, with unsteady turbulent adjoint stabilisation of the ATC term from [37]. This solver highlighted the need for more robust prolongation operators in OpenFOAM since only nearest-neighbour, linear and inverse distance interpolation are implemented. While this enables the implementation of the method from [55], it is also found to be less accurate than using quadratic or cubic interpolation.

It also raised a number of issues requiring further research. Due to the unstructured nature of the data, a U-ResNet was implemented, but it is far less efficient in training than conventional CNN. This would however also enable the use of the ANN as is during the mesh adaptation process since it would not require additional changes to be made to account for the change in number of degrees of freedom.

One of the issues for the implementation of the proposed method for three-dimensional problems is the growth of the number of weights and biases as the dimension of the mesh increases. 3D FCNN are not expected to be efficient based on the computational cost of training such a complex ANN. It must be explored whether the use of 2D SRNN for each z-layer is more efficient. This is preferred implementation in [39] for HIT.

It was also observed that the reconstruction was cheaper in training and more accurate with fewer variables to reconstruct. This resulted in the use of stream functions to be very attractive for the reconstruction of the primal. Due to the ill-defined meaning of stream functions for turbulence functions, this was not pursued in this thesis and would require additional analysis. However, the creation of a composite scalar field to describe more complex field flow variables could form an interesting area of research.

Moreover, while the thesis mentioned the potential of SRNN to reduce the storage footprint of the primal solution, this has to be implemented in practice in a mesh adaptation procedure in order to also quantify the loss in accuracy resulting from the reconstruction of the primal from a sub-sampled stored solution and the reduction in storage requirements. The cost of the primal reconstruction SRNN could also be lowered by training the ANN on filtered, for LES, or sub-sampled DNS data.

The consequence of sub-sampling in order to create training pairs also has to be investigated. In practice, it is expected that the coarse and fine solutions do not only differ on the frequency of the data in space but also by the frequency of the phenomena they are able to capture. As a result, sub-sampling does not take into account that the inputs to the ANN will differ from the sub-sampled fine solution. A potential solution would be to filter the fine data before sub-sampling it at the correct rate.

Finally, in the training process of either the primal or adjoint SRNN, it was found that the Mean-Square Error (MSE) loss function had great limitations. It lacks physical meaning and while it is effective for computer vision problems, the development of a loss functions that inherits physical meaning could greatly benefit the reconstruction ability of the ANN. An interesting path to investigate would be to use the QoI itself as a loss function or the output error estimate. However, the former would prevent the generalisation of the method to other functionals whereas the latter would lead to a larger computational overhead in the training process of the SRNN. The generalisation of the ANN and sensitivity of the SRNN to hyper-parameters should also be quantified.

Bibliography

[1] Tamer A. AbdelMigid, Khalid M. Saqr, Mohamed A. Kotb, and Ahmed A. Aboelfarag. Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using GPU accelerated code. Alexandria Engineering Journal, 56(1):123–135, 2017. ISSN 11100168. doi: 10.1016/j.aej.2016.09.013. URL `http://dx.doi.org/10.1016/j.aej.2016.09.013`.

[2] Djaffar Ait-Ali-Yahia, Guido Baruzzi, Wagdi G. Habashi, Michel Fortin, Julien Dompierre, and Marie Gabrielle Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part II. Structured grids. International Journal for Numerical Methods in Fluids, 39(8):657–673, 2002. ISSN 02712091. doi: 10.1002/fld.356.

[3] Frédéric Alauzet and Adrien Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. CAD Computer Aided Design, 72:13–39, mar 2016. ISSN 00104485. doi: 10.1016/j.cad.2015.09.005. URL `https://linkinghub.elsevier.com/retrieve/pii/S0010448515001517`.

[4] Jeric Alcala and Ilya Timofeyev. Subgrid-scale parametrization of unresolved scales in forced Burgers equation using Generative Adversarial Networks (GAN). arXiv, pages 1–15, 2020. ISSN 23318422.

[5] Joan Baiges, Ramon Codina, Inocencio Castañar, and Ernesto Castillo. A finite element reduced-order model based on adaptive mesh refinement and artificial neural networks. International Journal for Numerical Methods in Engineering, (September):1–14, 2019. doi: 10.1002/nme.6235.

[6] Timothy J. Baker. Mesh adaptation strategies for problems in fluid dynamics. Finite Elements in Analysis and Design, 25(3-4):243–273, 1997. ISSN 0168874X. doi: 10.1016/S0168-874X(96)00032-7.

[7] Raavi Balasubramanian and III C. Newman. Comparison of adjoint-based and feature-based grid adaptation for functional outputs. International Journal for Numerical Methods in Fluids, 53(10):1541–1569, 2007. ISSN 02712091. doi: 10.1002/fld.1361.

[8] Francesco Bassi, Alessandro Colombo, Andrea Crivellini, Krzysztof J. Fidkowski, Matteo Franciolini, Antonio Ghidoni, and Gianmaria Noventa. Entropy-adjoint p-adaptive discontinuous galerkin method for the under-resolved simulation of turbulent flows. AIAA Journal, 58(9):3963–3977, 2020. ISSN 00011452. doi: 10.2514/1.J058847.

[9] Pierre Benard, Guillaume Balarac, Vincent Moureau, Cecile Dobrzynski, Ghislain Lartigue, and Yves D'Angelo. Mesh adaptation for large-eddy simulations in complex geometries. International Journal for Numerical Methods in Fluids, 65(October 2010):236–253, 2011. doi: 10.1002/fld.4204.

[10] Joakim Bø. Master Thesis: Implementation and assessment of subgrid-scale models for large eddy simulations of incompressible turbulent flows, 2015.

[11] O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. Computers and Fluids, 27(4):421–433, 1998. ISSN 00457930. doi: 10.1016/S0045-7930(98)00002-4.

[12] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. arXiv, 2019. ISSN 23318422.

[13] Philip Claude Caplan, Robert Haimes, David L. Darmofal, and Marshall C. Galbraith. Anisotropic geometry-conforming d-simplicial meshing via isometric embeddings. Procedia Engineering, 203:141–153, 2017. ISSN 18777058. doi: 10.1016/j.proeng.2017.09.798. URL `https://doi.org/10.1016/j.proeng.2017.09.798`.

[14] Philip Claude Caplan, Robert Haimes, David L. Darmofal, and Marshall C. Galbraith. Extension of local cavity operators to 3d + t spacetime mesh adaptation. AIAA Scitech 2019 Forum, (January):1–19, 2019. doi: 10.2514/6.2019-1992.

[15] Philip Claude Caplan, Robert Haimes, and Xevi Roca. Isometric embedding of curvilinear meshes defined on riemannian metric spaces. Lecture Notes in Computational Science and Engineering, 127: 23–36, 2019. ISSN 21977100. doi: 10.1007/978-3-030-13992-6_2.

[16] Philip Claude Caplan, Robert Haimes, David L. Darmofal, and Marshall C. Galbraith. Four-Dimensional Anisotropic Mesh Adaptation. CAD Computer Aided Design, 129:102915, 2020. ISSN 00104485. doi: 10.1016/j.cad.2020.102915. URL `https://doi.org/10.1016/j.cad.2020.102915`.

[17] Guodong Chen and Krzysztof J. Fidkowski. Output-based error estimation and mesh adaptation using convolutional neural networks: Application to a scalar advection-diffusion problem. AIAA Scitech 2020 Forum, 1 PartF(January), 2020. doi: 10.2514/6.2020-1143.

[18] Guodong Chen and Krzysztof J. Fidkowski. Output-based error estimation and mesh adaptation using convolutional neural networks: Application to a scalar advection-diffusion problem. AIAA Scitech 2020 Forum, 1 PartF(February), 2020. doi: 10.2514/6.2020-1143.

[19] Guodong Chen and Krzysztof J. Fidkowski. Output-based adaptive aerodynamic simulations using convolutional neural networks. Computers and Fluids, 223, 2021. ISSN 00457930. doi: 10.1016/j.compfluid.2021.104947.

[20] M. Cheng, F. Fang, C.C. Pain, and I.M. Navon. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. Computer Methods in Applied Mechanics and Engineering, 365:113000, 2020. ISSN 00457825. doi: 10.1016/j.cma.2020.113000. URL `https://doi.org/10.1016/j.cma.2020.113000`.

[21] Franco Dassi, Hang Si, Simona Perotto, and Timo Streckenbach. Anisotropic finite element mesh adaptation via higher dimensional embedding. Procedia Engineering, 124:265–277, 2015. ISSN 18777058. doi: 10.1016/j.proeng.2015.10.138. URL `http://dx.doi.org/10.1016/j.proeng.2015.10.138`.

[22] Franco Dassi, Simona Perotto, Hang Si, and Timo Streckenbach. A priori anisotropic mesh adaptation driven by a higher dimensional embedding. CAD Computer Aided Design, 85:111–122, 2017. ISSN 00104485. doi: 10.1016/j.cad.2016.07.012. URL `http://dx.doi.org/10.1016/j.cad.2016.07.012`.

[23] Cecile Dobrzynski. MMG3D : User Guide. Technical report, INRIA, 2012.

[24] Kevin T. Doetsch and Krzysztof J. Fidkowski. A combined entropy and output-based adjoint approach for mesh refinement and error estimation. AIAA Aviation 2019 Forum, pages 1–21, 2019. doi: 10.2514/6.2019-2951.

[25] Julien Dompierre, Marie Gabrielle Vallet, Yves Bourgault, Michel Fortin, and Wagdi G. Habashi. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part III. Unstructured meshes. International Journal for Numerical Methods in Fluids, 39(8):675–702, 2002. ISSN 02712091. doi: 10.1002/fld.357.

[26] Krzysztof J. Fidkowski and Guodong Chen. A machine-learning anisotropy detection algorithm for output-adapted meshes. AIAA Scitech 2020 Forum, 1 PartF(January), 2020. doi: 10.2514/6.2020-0341.

[27] Krzysztof J. Fidkowski and David L. Darmofal. Output-based error estimation and mesh adaptation in Computational Fluid Dynamics: Overview and recent results. 47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, (January), 2009. doi: 10.2514/6.2009-1303.

[28] Krzysztof J. Fidkowski and David L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. AIAA Journal, 49(4):673–694, 2011. ISSN 00011452. doi: 10.2514/1.J050073.

[29] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Machine learning based spatio-temporal super resolution reconstruction of turbulent flows. Journal of Fluid Mechanics, (2019):1–22, apr 2020. doi: 10.1017/jfm.2020.948. URL `http://arxiv.org/abs/2004.11566http://dx.doi.org/10.1017/jfm.2020.948`.

[30] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of Computational Physics, 48(3):387–411, 1982. ISSN 10902716. doi: 10.1016/0021-9991(82)90058-4.

[31] Michael B Giles, Niles Pierce, and Endre Süli. Computing and Visualization in Science Regular article Progress in adjoint error correction for integral functionals. Computing and Visualisation in Science, 121:113–121, 2004.

[32] Wagdi G. Habashi, Julien Dompierre, Yves Bourgault, Djaffar Ait-Ali-Yahia, Michel Fortin, and Marie Gabrielle Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles. International Journal for Numerical Methods in Fluids, 32(6):725–744, 2000. ISSN 02712091. doi: 10.1002/(SICI)1097-0363(20000330)32:6<725::AID-FLD935>3.0.CO;2-4.

[33] Robert Haimes and Mark Drela. On the construction of aircraft conceptual geometry for high-fidelity analysis and design. 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, (January):1–21, 2012. doi: 10.2514/6.2012-683.

[34] Botros N. Hanna, Nam T. Dinh, Robert W. Youngblood, and Igor A. Bolotnov. Coarse-grid computational fluid dynamic (CG-CFD) error prediction using machine learning. arXiv, 2017. ISSN 23318422.

[35] Frédéric Hecht. BAMG : Bidimensional Anisotropic Mesh Generator. Technical report, University Pierre et Marie Curie, 2006.

[36] Johan Hoffman and Claes Johnson. Stability of the dual Navier-Stokes equations and efficient computation of mean output in turbulent flow using adaptive DNS/LES. Computer Methods in Applied Mechanics and Engineering, 195(13-16):1709–1721, 2006. ISSN 00457825. doi: 10.1016/j.cma.2005.05.034.

[37] Georgios K. Karpouzas, Evangelos M. Papoutsis-Kiachagias, Thomas Schumacher, Eugene de Villiers, Kyriakos C. Giannakoglou, and Carsten Othmer. Adjoint optimization for vehicle external aerodynamics. International Journal of Automotive Engineering, 7(1):1–7, 2016. ISSN 21850992. doi: 10.20485/jsaeijae.7.1_1.

[38] Steven M. Kast. An Introduction to Adjoints and Output Error Estimation in Computational Fluid Dynamics. (May), 2017. URL `http://arxiv.org/abs/1712.00693`.

[39] Hyojin Kim, Junhyuk Kim, Sungjin Won, and Changghoon Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. Journal of Fluid Mechanics, pages 1–34, jul 2020. doi: 10.1017/jfm.2020.1028. URL `http://arxiv.org/abs/2007.15324http://dx.doi.org/10.1017/jfm.2020.1028`.

[40] Hendrik C Kuhlmann and Francesco Romano. The lid-driven cavity. 2020.

[41] Y. L. Kuo and W. L. Cleghorn. The h-, p-, and r-refinements of finite element analysis of flexible slider crank mechanism. JVC/Journal of Vibration and Control, 13(4):415–435, 2007. ISSN 10775463. doi: 10.1177/1077546307076901.

[42] J. Nathan Kutz. Deep learning in fluid dynamics. Journal of Fluid Mechanics, 814:1–4, 2017. ISSN 14697645. doi: 10.1017/jfm.2016.803.

[43] Pierre Lauret, Frédéric Heymes, Laurent Aprin, and Anne Johannet. Atmospheric dispersion modeling using Artificial Neural Network based cellular automata. Environmental Modelling and Software, 85: 56–69, 2016. ISSN 13648152. doi: 10.1016/j.envsoft.2016.08.001.

[44] Jichao Li, Mengqi Zhang, Joaquim R.R.A. Martins, and Chang Shu. Efficient aerodynamic shape optimization with deep-learning-based geometric filtering. AIAA Journal, 58(10):4243–4259, 2020. ISSN 1533385X. doi: 10.2514/1.J059254. URL `https://doi.org/10.2514/1.J059254`.

[45] Xiaodong Li, Steven Hulshoff, and Stefan Hickel. Towards adjoint-based mesh refinement for Large Eddy Simulation using reduced-order primal solutions : preliminary 1D Burgers study. Computer methods in applied mechanics and engineering, 2021.

[46] Yingzhou Li, Jianfeng Lu, and Anqi Mao. Variational training of neural network approximations of solution maps for physical models. Journal of Computational Physics, 409:1–21, 2020. ISSN 23318422. doi: 10.1016/j.jcp.2020.109338.

[47] Zhenquan Li. Accuracy analysis of a mesh refinement method using benchmarks of 2-D lid-driven cavity flows and finer meshes. Journal of Mathematical Chemistry, 52(4):1156–1170, 2014. ISSN 02599791. doi: 10.1007/s10910-014-0334-0.

[48] Bo Liu, Jiupeng Tang, Haibo Huang, and Xi-Yun Lu. Deep learning methods for super-resolution reconstruction of turbulent flows. Physics of Fluids, 32(2):025105, feb 2020. ISSN 1070-6631. doi: 10.1063/1.5140772. URL `https://doi.org/10.1063/1.5140772http://aip.scitation.org/doi/10.1063/1.5140772`.

[49] Yuanyuan Liu, Xiaohui Yuan, Xi Gong, Zhong Xie, Fang Fang, and Zhongwen Luo. Conditional convolution neural network enhanced random forest for facial expression recognition. Pattern Recognition, 84: 251–261, 2018. ISSN 0031-3203. doi: 10.1016/j.patcog.2018.07.016. URL `https://doi.org/10.1016/j.patcog.2018.07.016`.

[50] Rainald Löhner. An adaptive finite element scheme for transient problems in CFD. Computer Methods in Applied Mechanics and Engineering, 61(3):323–338, 1987. ISSN 00457825. doi: 10.1016/0045-7825(87)90098-3.

[51] Paolo Luchini and Alessandro Bottaro. Adjoint equations in stability analysis. Annual Review of Fluid Mechanics, 46(September 2013):493–517, 2014. ISSN 00664189. doi: 10.1146/annurev-fluid-010313-141253.

[52] Larry Manevitz, Akram Bitar, and Dan Givoli. Neural network time series forecasting of finite-element mesh adaptation. Neurocomputing, 63(SPEC. ISS.):447–463, 2005. ISSN 09252312. doi: 10.1016/j.neucom.2004.06.009.

[53] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. arXiv, pages 1–7, nov 2014. URL `http://arxiv.org/abs/1411.1784`.

[54] Siva K. Nadarajah and Antony Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. 38th Aerospace Sciences Meeting and Exhibit, 2000. doi: 10.2514/6.2000-667.

[55] Marian Nemec, Michael J. Aftosmis, and Mathias Wintzer. Adjoint-based adaptive mesh refinement for complex geometries. 46th AIAA Aerospace Sciences Meeting and Exhibit, (January), 2008. doi: 10.2514/6.2008-725.

[56] Steven Nerney, Edward J. Schmahl, and Z. E. Musielak. Analytic solutions of the vector Burgers' equation. Quarterly of Applied Mathematics, 54(1):63–71, 1996. ISSN 0033-569X. doi: 10.1090/qam/1373838.

[57] Nicolas Odier, Adrien Thacker, Maël Harnieh, Gabriel Staffelbach, Laurent Gicquel, Florent Duchaine, Nicolás García Rosa, and Jens Dominik Müller. A mesh adaptation strategy for complex wall-modeled turbomachinery LES. Computers and Fluids, 214:104766, 2021. ISSN 00457930. doi: 10.1016/j.compfluid.2020.104766. URL `https://doi.org/10.1016/j.compfluid.2020.104766`.

[58] N. Offermans, A. Peplinski, O. Marin, and P. Schlatter. Adjoint error estimators and adaptive mesh refinement in Nek5000. 2019. URL `http://www.diva-portal.org/smash/get/diva2:1156594/FULLTEXT01.pdf{%}0Aurn:nbn:se:kth:diva-217498`.

[59] Koji Ohkitani and Mark Dowker. Numerical study on comparison of Navier-Stokes and Burgers equations. Physics of Fluids, 24(5), 2012. ISSN 10706631. doi: 10.1063/1.4719787.

[60] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. International Journal for Numerical Methods in Fluids, 58(8):861–877, 2008. doi: 10.1002/fld.1770.

[61] Alexis Papagiannopoulos, Pascal Clausen, and François Avellan. How to teach neural networks to mesh: Application on 2-D simplicial contours. Neural Networks, 136:152–179, 2021. ISSN 18792782. doi: 10.1016/j.neunet.2020.12.019. URL `https://doi.org/10.1016/j.neunet.2020.12.019`.

[62] Tyrone S. Phillips and Carl Ollivier-Goochy. A truncation error based anisotropic mesh adaptation metric for CFD. AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting, (January):1–10, 2017. doi: 10.2514/6.2017-1947. URL `http://dx.doi.org/10.2514/6.2017-1947`.

[63] Lucien Rochery and Adrien Loseille. P 2 Cavity Operator and Riemannian Curved Edge Length Optimization: a Path to High-Order Mesh Adaptation. In AIAA Scitech 2021 Forum, Reston, Virginia, jan 2021. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-609-5. doi: 10.2514/6.2021-1781. URL `https://arc.aiaa.org/doi/10.2514/6.2021-1781`.

[64] David Roos Launchbury. Unsteady Turbulent Flow Modelling and Applications. Unsteady Turbulent Flow Modelling and Applications, 2016. doi: 10.1007/978-3-658-11912-6.

[65] Christopher J. Roy. Strategies for driving mesh adaptation in CFD (invited). 47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, (January 2009), 2009. doi: 10.2514/6.2009-1302.

[66] Kambiz Salari and Patrick M. Knupp. Code Verification by the Method of Manufactured Solutions. Technical report, Sandia National Laboratories, 2000.

[67] J. Slotnick, A. Khodadoust, J. Alonso, and D. Darmofal. CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences. Nnasa/Cr-2014-218178, (March), 2014.

[68] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N. King. Adversarial super-resolution of climatological wind and solar data. Proceedings of the National Academy of Sciences of the United States of America, 117(29):16805–16815, 2020. ISSN 10916490. doi: 10.1073/pnas.1918964117.

[69] William C. Tyson, Katarzyna Kasia Świrydowicz, Joseph M. Derlaga, Christopher J. Roy, and Eric de Sturler. Improved functional-based error estimation and adaptation without adjoints. 46th AIAA Fluid Dynamics Conference, (June), 2016. doi: 10.2514/6.2016-3809.

[70] David A. Venditti and David L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. Journal of Computational Physics, 176(1):40–69, 2002. ISSN 00219991. doi: 10.1006/jcph.2001.6967.

[71] David A. Venditti and David L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. Journal of Computational Physics, 187(1):22–46, 2003. ISSN 00219991. doi: 10.1016/S0021-9991(03)00074-3.

[72] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. Journal of Computational and Applied Mathematics, 50(1-3):67–83, 1994. ISSN 03770427. doi: 10.1016/0377-0427(94)90290-9.

[73] J.G. Wallwork, D.A. Ham, N. Barral, and M.D. Piggott. Goal-oriented error estimation and mesh adaptation for tracer transport modelling. EarthArXiv, 2021.

[74] M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. International Journal for Numerical Methods in Fluids, 76(11):811–834, 2014. ISSN 10970363. doi: 10.1002/fld.3959.

[75] Xuping Xie, Clayton Webster, and Traian Iliescu. Closure Learning for Nonlinear Model Reduction Using Deep Residual Neural Network. Fluids, 5(1):39, mar 2020. ISSN 2311-5521. doi: 10.3390/fluids5010039. URL `https://www.mdpi.com/2311-5521/5/1/39`.

[76] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow. arXiv, 37(4), jan 2018. doi: 10.1145/3072959.3073643. URL `http://arxiv.org/abs/1801.09710http://dx.doi.org/10.1145/3072959.3073643`.

[77] Mengfei Xu, Shufang Song, Xuxiang Sun, Wengang Chen, and Weiwei Zhang. Machine learning for adjoint vector in aerodynamic shape optimization. arXiv e-prints, page arXiv:2012.15730, 2020. URL `https://ui.adsabs.harvard.edu/abs/2020arXiv201215730X`.

[78] Mengfei Xu, Shufang Song, Xuxiang Sun, and Weiwei Zhang. A convolutional strategy on unstructured mesh for the adjoint vector modeling. Physics of Fluids, 33(3):036115, mar 2021. ISSN 1070-6631. doi: 10.1063/5.0044093. URL `https://aip.scitation.org/doi/10.1063/5.0044093`.

[79] Nail K. Yamaleev. Minimization of the Truncation Error by Grid Adaptation, 2001. ISSN 00219991.

[80] Masayuki Yano, James M. Modisettey, and David L. Darmofal. The importance of mesh adaptation for higher-order discretizations of aerodynamic flows. 20th AIAA Computational Fluid Dynamics Conference 2011, (June):1–19, 2011. doi: 10.2514/6.2011-3852.

[81] Emre Yılmaz and Brian J. German. Conditional generative adversarial network framework for airfoil inverse design. Aiaa Aviation 2020 Forum, 1 PartF:1–17, 2020. doi: 10.2514/6.2020-3185.

[82] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into Deep Learning. 0.16.1 edition, 2021.

[83] Pascal Zille, Thomas Corpetti, Liang Shao, and Chen Xu. Super-resolution of turbulent passive scalar images using data assimilation. Experiments in Fluids, 57(2):1–14, 2016. ISSN 07234864. doi: 10.1007/s00348-015-2104-6.