Unsupervised Residual Vector Analysis for Mesh Optimization

Mohammad Zandsalimy* and Carl Ollivier-Gooch†
The University of British Columbia, Vancouver, British Columbia, V6T 1Z4

This work prototypes novel methods to enhance the efficiency of a previous CFD stability improvement approach. The feasibility of residual vector analysis for unstable solution mode identification is studied. Unsupervised machine learning models in the form of outlier detectors are used to identify anomalous vector elements. A novel method is presented to construct synthetic vectors that resemble unstable eigenvectors. Synthetic vectors are used to find vertices for modification and calculate the vertex movement direction and magnitude. The residual vector helps substantially in reducing the computational cost of the optimization algorithm. In comparison to the full eigenanalysis of the Jacobian matrix, residual vector analysis requires much fewer computational resources. This methodology is used for the first time for the stability improvement of finite-volume simulations. It is shown that using residual vector analysis for the identification of the unstable solution modes and problematic cells in the mesh has a similar stabilization performance to using the right eigenvectors.

I. Introduction

The stability analysis and improvement of unstructured finite-volume problems for industrial-scale applications remains a complication of these simulations to date. Such substantial problems often include opposing numerical requirements due to the size and broad spectral content of the solution. This compels engineers to seek a balance between the accuracy, stability, and computational cost of the numerical method. Improving stability at modest computational costs without impacting accuracy shifts this balance in a favorable direction. Consequently, stability analysis has been a fundamental aspect of unstructured finite-volume methods and the subject of different studies in this area [1–3].

In the present study, the Lyapunov stability theory is utilized for stability analysis of the dynamical systems resulting from the discretization of partial differential equations. According to this theory, the eigenanalysis of the Jacobian matrix reveals useful information about the local stability of the linear system in question. The eigenvalues that have a positive real part show the unstable modes of the solution. Such modes can be purely numerical or physically unstable modes. The right eigenvectors associated with the unstable eigenmodes point to the cells of the mesh in which those solution modes have significant weights. The magnitude of such unstable modes grows with solution iteration which will result in the growth of the residual in these cells. The purely real dominant unstable modes exhibit a constant direction in the residual vector during the simulation process. Detecting such anomalies in the residual vector can help identify the problematic cells in the mesh. New vectors can be constructed using the residual, resembling the unstable eigenvectors, to be used for mesh modification to improve stability.

Citro et al. [4] proposed a method based on the minimization of the residual norm at each iteration step with a projection basis to compute the unstable steady states and/or accelerate the convergence to stable configurations. Their algorithm was able to improve the convergence of existing iterative schemes. Farhat et al. [5] focused on the analysis and comparative performance study for nonlinear finite-element models. Their method directly approximates the scalar projections of residuals and Jacobians. For second-order dynamical systems, this method preserves the numerical stability properties of the dynamical system. Xu [6] employed SVD on symplectic matrices and presented a general formula to construct a factorization with the minimal norm and condition number to improve numerical stability.

Further, eigenanalysis has been an effective tool for the investigation of stability in different dynamical systems in the past. Wang and Semlyen [7] successfully applied eigenanalysis to the matrices derived for small signal stability studies of power systems. The resulting matrices in such systems are large, sparse, and non-Hermitian. Lehoucq and Salinger [8] presented an approach for determining the linear stability of the steady state of PDEs on parallel

*Ph.D. Candidate, Mechanical Engineering, The Advanced Numerical Simulation Laboratory.

†Professor, Mechanical Engineering, The Advanced Numerical Simulation Laboratory, AIAA Associate Fellow.

computers. They successfully calculated the eigenvalues with the largest real parts in systems of order up to 4 million. They achieved this goal using Arnoldi's eigensolution method [9] driven by a novel implementation of the Cayley transformation. Morzyski et al. [10] applied the eigensolution to global non-parallel flow stability. They used subspace iteration methods to solve the large-dimensional algebraic eigenvalue problem with the inverse Cayley method for the separation of the eigenvalues of interest. Chen et al. [11] presented a stability analysis method for polynomially-dependent one-parameter systems which is referred to as the eigenvalue perturbation approach. They studied the asymptotic behavior of the critical eigenvalues on the imaginary axis. This behavior determines whether the pure imaginary eigenvalues cross from one half-plane into another. Their approach leads to numerically efficient stability conditions.

Zandsalimy and Ollivier-Gooch [12] presented novel methods to enhance the efficiency of a previous mesh optimization approach [13] that used the eigenanalysis of the Jacobian matrix and was infeasible for the application of industrial simulations. The issues have been addressed in [12] by improving the vertex selection strategies as well as the overall optimization approach. A novel algorithm is also presented therein to remediate the opposing eigenmodes in a finite-volume solution resulting from large problems. These enhancements greatly reduce the computational cost of the presented methodology.

In the present work, the main outlook is shifted from the eigenanalysis of the Jacobian matrix to residual vector analysis. The resulting synthetic solution modes are utilized to identify the problematic cells in the mesh and vertices to move. We also show the feasibility of using the residual vector for movement vector calculation. This novel approach dramatically reduces the computational complexity of CFD stability improvement, which was previously dominated by eigenvalue calculations. We utilize unsupervised anomaly detection algorithms to determine outlier values in the working vectors to isolate the problematic cells. This machine learning module also helps discover the correct non-linear solution iteration to apply our optimization algorithm by identifying anomalous behavior in the residual vector. The results are validated and compared to previous studies which show the improvements achieved through this novel proposal.

II. Background

A. Numerical Simulation Scheme

The conservation equation for an arbitrary dependent variable U which is a function of time and a vector of independent variables $\vec{x}$ is,

$$\frac{\partial U}{\partial t}(t,\vec{x}) + \vec{\nabla} \cdot \vec{F}(U) = f(t,\vec{x}) \tag{1}$$

Here, $\vec{F}$ is the flux vector and f is the source term. The differential equations are written in divergence form and integrated over control volumes. With the application of Gauss's theorem, the volume integration is converted into a surface integral across the boundaries of an arbitrary cell as presented in Equation 2. This numerical procedure is known as the finite-volume method [14–17].

$$\frac{d\bar{U}_i}{dt} = -\frac{1}{|V_i|}\oint_{\partial V_i}\left(\vec{F}\cdot\vec{n}\right)dA + \frac{1}{|V_i|}\int_{V_i} f\,dV = R(\bar{U}) \tag{2}$$

In this equation, V_i refers to the ith cell with a volume of $|V_i|$, $\vec{n}$ is the unit normal vector to the boundaries, R is the residual, and $\bar{U}_i$ is the average conservative property inside the cell,

$$\bar{U}_i = \frac{1}{|V_i|}\int_{V_i} U dV \tag{3}$$

The conservation property is the main advantage over the weak formulation adopted in finite-element methods [18, 19]. The following approach is used to calculate the flux integral in Equation 2 with second-order accuracy;

1) Reconstructing a piece-wise linear solution approximation from the piece-wise constant control volume averages using the linear least-squares method [20].
2) Computing the flux at each quadrature point on the cell's boundaries. We utilize Roe's scheme [21] for inviscid flux calculation.
3) Integrating flux values on Gauss quadrature points.

The boundary conditions are applied weakly using flux values on the boundaries. Crank-Nicolson time advance scheme is utilized for second-order time integration as presented in Equation 4 (the bar notation is dropped for convenience).

$$\frac{\vec{U}^{n+1} - \vec{U}^{n}}{\delta t} = \frac{\delta \vec{U}}{\delta t} = \frac{1}{2}\left(\vec{R}(\vec{U}^{n+1}) + \vec{R}(\vec{U}^{n})\right) \tag{4}$$

in which, $\vec{U} = \{U_1, U_2, \dots, U_k\}$ is the vector of control volume averages and $\vec{R}$ is the residual vector. The linearized from of Equation 4 is,

$$\left(\frac{1}{\delta t} I - \frac{1}{2}\frac{\partial \vec{R}}{\partial \vec{U}}\right)\delta \vec{U} = \vec{R}(\vec{U}^{n}) \tag{5}$$

In this equation, the Jacobian matrix ($\frac{\partial \vec{R}}{\partial \vec{U}}$) can be determined using the finite-difference method or more efficiently by chain rule differentiation [22],

$$A = \frac{\partial \vec{R}}{\partial \vec{U}} = \frac{\partial \text{FluxInt}}{\partial \text{Flux}}\frac{\partial \text{Flux}}{\partial \text{RecSol}}\frac{\partial \text{RecSol}}{\partial \text{RecCoef}}\frac{\partial \text{RecCoef}}{\partial \text{PVars}}\frac{\partial \text{PVars}}{\partial \text{CVars}} \tag{6}$$

in which, FluxInt is the flux integral, Flux are the numerical fluxes, RecSol are the reconstructed solutions at Gauss points, RecCoef are the reconstruction coefficients, PVars are the control volume averages of the primitive variables used in the reconstruction, and CVars are the control volume averages of the conserved variables [22].

B. Solution Stability

The second method of Lyapunov [23] is utilized to determine the stability of a system without explicit integration. According to this method, the rate of change in the Lyapunov function can predict the stability of the solution. Unfortunately, a general and reliable procedure to find Lyapunov functions is not available. For physical systems, however, the total energy stored in the system is often suggested as a Lyapunov function [24]. A concept similar to the kinetic energy of dynamic systems is employed as the Lyapunov function in the present study. The following energy relation is defined on the averaged solution vector U,

$$E = \frac{1}{2}\vec{U}^T \vec{U} \tag{7}$$

For homogeneous systems of degree one (e.g. Advection, Euler, and Navier-Stokes equations) the rate of change in energy is,

$$\frac{dE}{dt} = \vec{U}^T \frac{d\vec{U}}{dt} = \vec{U}^T A \vec{U} \tag{8}$$

Decreasing system energy represents a strictly stable solution.

The Lyapunov theorem of stability implies that the linear time-invariant system $\dot{x} = Ax$ is locally stable if all the eigenvalues of A have negative (or zero) real parts. As a result, the system is unstable if any eigenvalue of A has a positive real part. This theorem can also be used for linear time-varying systems of the form $\dot{x} = A(t)x$. Note that the theorems of Lyapunov offer a procedure to determine stability but not a recipe for finding the energy function (Lyapunov function) [25].

C. Stabilization Scheme

This section provides an overview of the presented stabilization approach by Zandsalimy and Ollivier-Gooch [12]. This approach identifies the vertices whose locations have the largest effect on the unstable eigenmodes in the solution. Changing the location of such vertices along carefully chosen vectors helps improve the stability of the numerical solution or increases the convergence rate. The reader is referred to [12] for the details of the approach. The overall steps required to stabilize a given unstable solution are presented in the following.

Eigenanalysis We must identify eigenvalues in the right open-half of the eigenspectrum for the large, sparse Jacobian matrices arising in CFD problems. This module is the most computationally demanding aspect of the algorithm. Spectral transformation techniques such as the Cayley algorithm are used to isolate the problematic eigenvalues of the Jacobian matrix. The Krylov-Schur technique [26] as implemented in the Scalable Library for Eigenvalue Problem Computations (SLEPc) [27] is utilized to solve the eigenproblem.

Vertex Selection After identifying the unstable solution modes, the next step is to find vertices whose location affects the unstable solution modes the most. A possible solution is to find the gradients of the eigenvalues with respect to the movement of all vertices in different directions and choose the vertices with the largest gradients [13]. This approach, however, is computationally demanding and ideally, we prefer a much less numerically intensive method. It is shown in [12] that the gradient of each eigenmode is highly local to a certain part of the mesh. As a result, only moving a single vertex in the mesh helps to push the unstable eigenvalue with a positive real part to the left open-half of the eigenspectrum. Our approach involves using the right eigenvector associated with each unstable mode. The absolute value of the eigenvector in each cell is added as a selection criterion for each adjacent vertex. Finally, the vertex with the largest selection value is chosen for modification. This new vector points to the vertex with the largest eigenvalue gradient for a given unstable mode. Note that searching the full domain of the solution is not necessary due to the locality of each unstable eigenvector in the solution.

Eigenvalue Gradient The gradients of the unstable eigenvalues with respect to the movement of the selected vertices reveal important information about the correct vertex movement direction and magnitude. The goal is to move the selected vertex in a direction that would result in the unstable eigenvalue being pushed to the left open-half side of the eigenspectrum. Further, the imaginary part of the eigenvalues causes oscillations in the solution which can negatively affect the amplification factor as well as the convergence rate of the simulation. The presented stabilization approach can be readily utilized to mitigate such cases too. The gradients of eigenvalues with respect to mesh coordinates can be computed directly using the finite-difference method. However, this method will require multiple solutions to the eigenproblem which is not ideal. We prefer reducing the need for eigenanalysis as it is the most computationally intensive aspect of the presented approach. Hence, the gradient of an eigenvalue λ with respect to vertex location ζ is estimated using the following equation [28],

$$\frac{d\lambda}{d\zeta} = y^H \frac{dA}{d\zeta} x \tag{9}$$

In this equation, y^H is the Hermitian transpose of the left eigenvector and x is the right eigenvector associated with the unstable eigenvalue λ. In this approach, we only need to solve the eigenvalue problem once. The term $\frac{dA}{d\zeta}$ can be found using the following approximation.

$$\frac{dA}{d\zeta} = \frac{\partial A}{\partial U}\frac{\partial U}{\partial \zeta} + \frac{\partial A}{\partial \zeta} \approx \frac{\partial A}{\partial \zeta} \tag{10}$$

At steady-state conditions, we can assume that the changes in solution with respect to changes in the mesh are negligible. As a result, we can use the presented approximation.

Mesh Modification In the final step, we can utilize the eigenvalue gradient to modify the location of the selected vertex to stabilize the unstable eigenvalue. A truncated Taylor series is used for mesh modification and changing the unstable eigenvalue λ into a new stable value of λ' as follows,

$$\lambda' = \lambda + \frac{d\lambda}{d\zeta}\delta\zeta \tag{11}$$

The goal here is to move the undesirable eigenmodes to the left side of the complex plane. Our experiments show that the eigenvalue with the largest real part dictates the behavior of the residual in the solution. As a result, we aim to move the right-most eigenvalues to the left side of the eigenspectrum so that they are not the dominant values anymore.

The most computationally demanding aspect of the presented algorithm is the solution to the large and sparse eigensystem. For a large sparse matrix $A \in \mathbb{R}^{n\times n}$, computing m eigenvalues requires the following considerations [27],

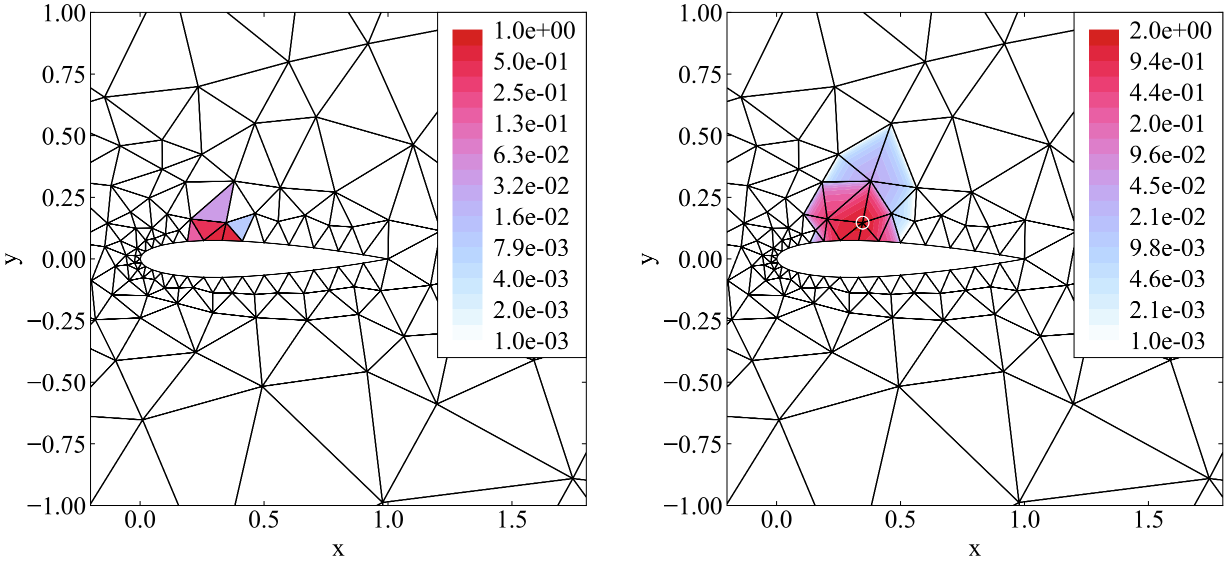
1) Storage of m subspace vectors of length n.
2) Orthogonalization of the basis vectors, with a computational cost of $O(m^2 n)$.
3) Storage of at least one dense projected eigenproblem of size $m \times m$.
4) The solution to the projected eigenproblem with a computational cost of $O(m^3)$.

In our approach, the solution to a handful of eigenvalues on the right side of the eigenspectrum is usually sufficient which are often well-separated from others. This makes finding the eigenvalues of interest less computationally demanding. The present study seeks to prototype new methods to reduce the need for eigenanalysis or remove it completely from the algorithm.

III. Alternative Vertex Selection

Currently, the *Vertex Selection* and *Eigenvalue Gradient* modules depend on the solution to the eigenproblem. The right eigenvectors are solution modes that can be used to identify the problematic areas in the mesh. The eigenvectors associated with the unstable modes point to the cells that contribute the most to instabilities. Alternatively, the residual vector can provide useful information on the unstable modes of the solution. Monitoring the residual vector for anomalies can indicate the problematic areas of the mesh. If the largest unstable mode is a purely real number, the residual value in the problematic cells is expected to grow without phase change. This makes the identification of such modes through the residual vector more feasible. However, if the dominant unstable mode is a pair of conjugate gradient values, the residual will undergo a constant phase change which can hinder our ability to identify instabilities. In general, the residual magnitude growth is related to the real part of the dominant eigenmodes while the oscillatory behavior is due to the imaginary part. Utilizing the residual vector helps tremendously in reducing the computational cost of the optimization algorithm. Unlike the right eigenvectors associated with the unstable modes of the solution which usually are highly local to certain areas in the mesh, the residual vector can have non-negligible values in the areas that are not contributing to instabilities. As a result, a novel approach to building synthetic vectors that point to certain areas in the mesh out of the residual vector is introduced in the present study. In this approach, the maximum value of the residual vector is selected and the corresponding cell is identified. The synthetic vector is constructed with the residual value in the cells that are present in the Jacobian fill of the previously selected control volume with zero in other locations. This increases the chances of identifying a vector that looks like an unstable mode of the solution.

After identifying the appropriate vector x, we can use an approach similar to the work of Zandsalimy and Ollivier-Gooch [12] for vertex selection. In this approach, the absolute values of the vector are plotted on the mesh as shown in Figure 1a. The next step is to use the summation of the vector value in each cell on its vertices as a measure of weight. The selection made through this method usually corresponds to vertices whose locations affect the unstable modes the most. Figure 1b shows the distribution of the weight measure and the vertex that is selected as a candidate for mesh optimization indicated with a white circle. This figure corresponds to the vertex selection procedure in an unstable Euler problem using a synthetic vector built with the residual vector at a certain iteration of the solution. The dominant unstable modes in this problem are pure real values which makes the residual vector have a constant shape, growing over a large number of non-linear solution iterations. The dominant unstable modes in this case dictate the shape of the residual vector and its divergence rate. As depicted, the selected vector is highly local to a certain area in the mesh which echos the findings of [12] about the unstable right eigenvectors.

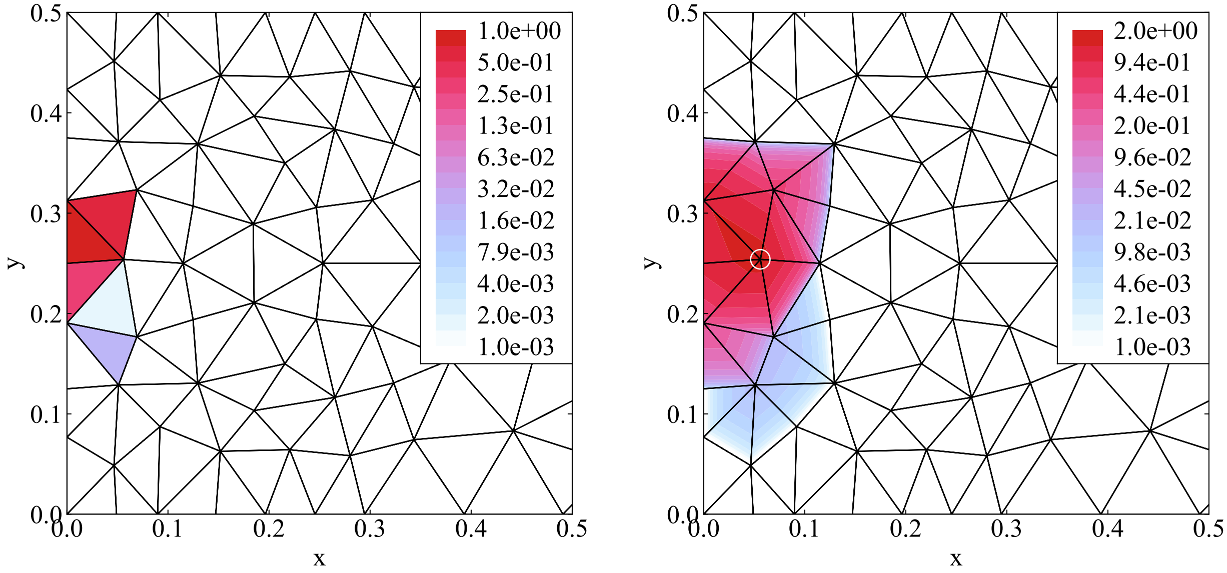


(a) The absolute value of the residual vector

(b) The summation of weights for each vertex

Fig. 1 The vertex selection procedure in an Euler problem

The next experiment is performed on a Burgers problem in a rectangular channel. The absolute values of the synthetic vector built using the residual vector are depicted in Figure 2a. As seen, the selected vector is highly local to a certain area in the mesh which indicates the problematic cells. Figure 2b shows the selection weight measure on the vertices at the same iteration of the problem. In this case, a single vertex is selected for modification through the presented methodology which is indicated with a white circle.



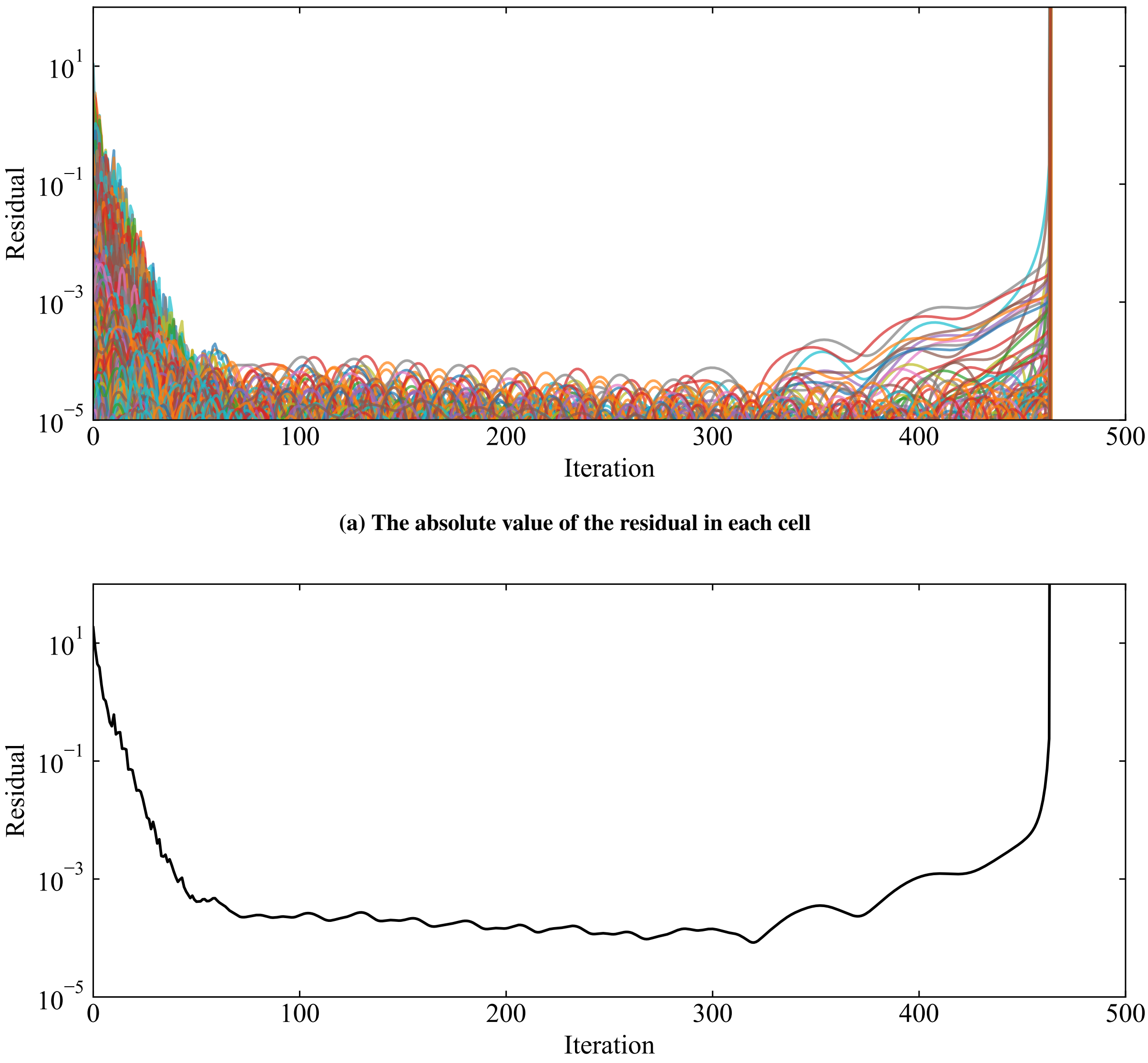
(a) The absolute value of the residual vector

(b) The summation of weights for each vertex

Fig. 2 The vertex selection procedure in a Burgers problem

A. The Residual Vector

If employed correctly, using the residual vector as an alternative measure to identify the problematic areas of the mesh reduces the computational cost tremendously. However, the identification of unstable solution modes in the residual vector can be a perplexing task. The residual vector often changes shape during the non-linear solution iterations. According to our experiments, in cases where there are purely real unstable dominant solution modes, the residual vector keeps a roughly constant shape during its growth phase for several solution iterations. This vector usually points to the areas in the mesh in which those unstable modes are dominant and are causing the residual to grow in magnitude. In cases where the dominant eigenvalues are a pair of conjugate modes, the magnitude growth is complemented by a constant phase change which makes the identification of the problematic areas in the mesh more difficult. Nevertheless, in the former case, anomaly detection models can be utilized to find the cells in which the residual vector exhibits outlier values. Figure 3 shows the L_2 norm of the residual vector as well as the individual residual values in all the cells in the Euler problem around the NACA 0015 airfoil presented in Figure 1. As seen in Figure 3a, after iteration 350 of the solution, some outlier values can be identified which are growing in magnitude. This non-linear iteration also corresponds to the growth in the L_2 norm of the residual vector as seen in Figure 3b.

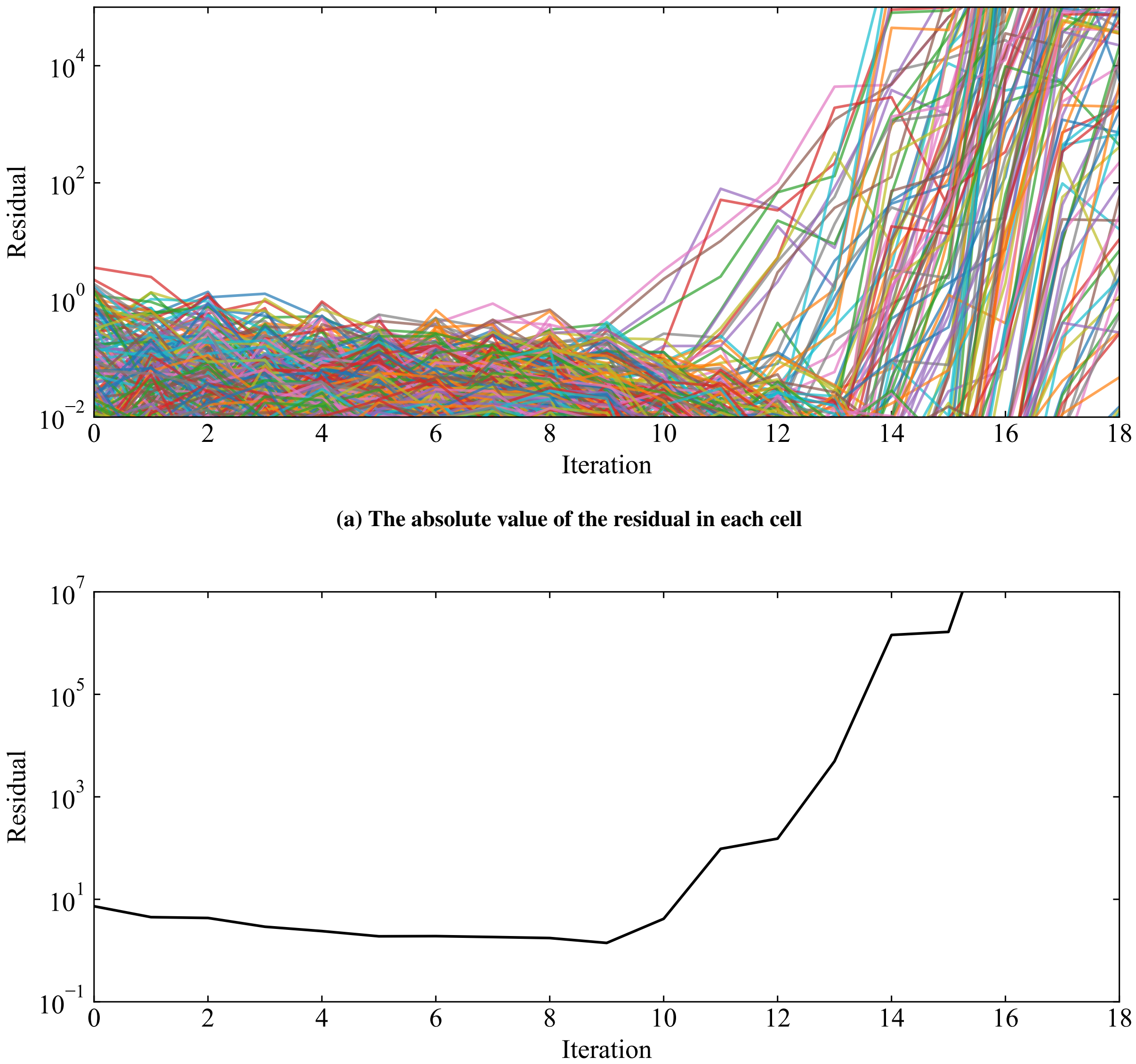


(a) The absolute value of the residual in each cell

(b) The L_2 norm of the residual vector

Fig. 3 The residual history in an Euler problem

Figure 4 shows the L_2 norm of the residual vector as well as the individual residual values in all the cells in the Burgers problem on a rectangular channel presented in Figure 2. As seen in Figure 4a, after iteration 10 of the solution, some outlier values can be identified which are growing in magnitude. This non-linear iteration also corresponds to the growth in the L_2 norm of the residual vector as seen in Figure 4b.



(a) The absolute value of the residual in each cell

(b) The L_2 norm of the residual vector

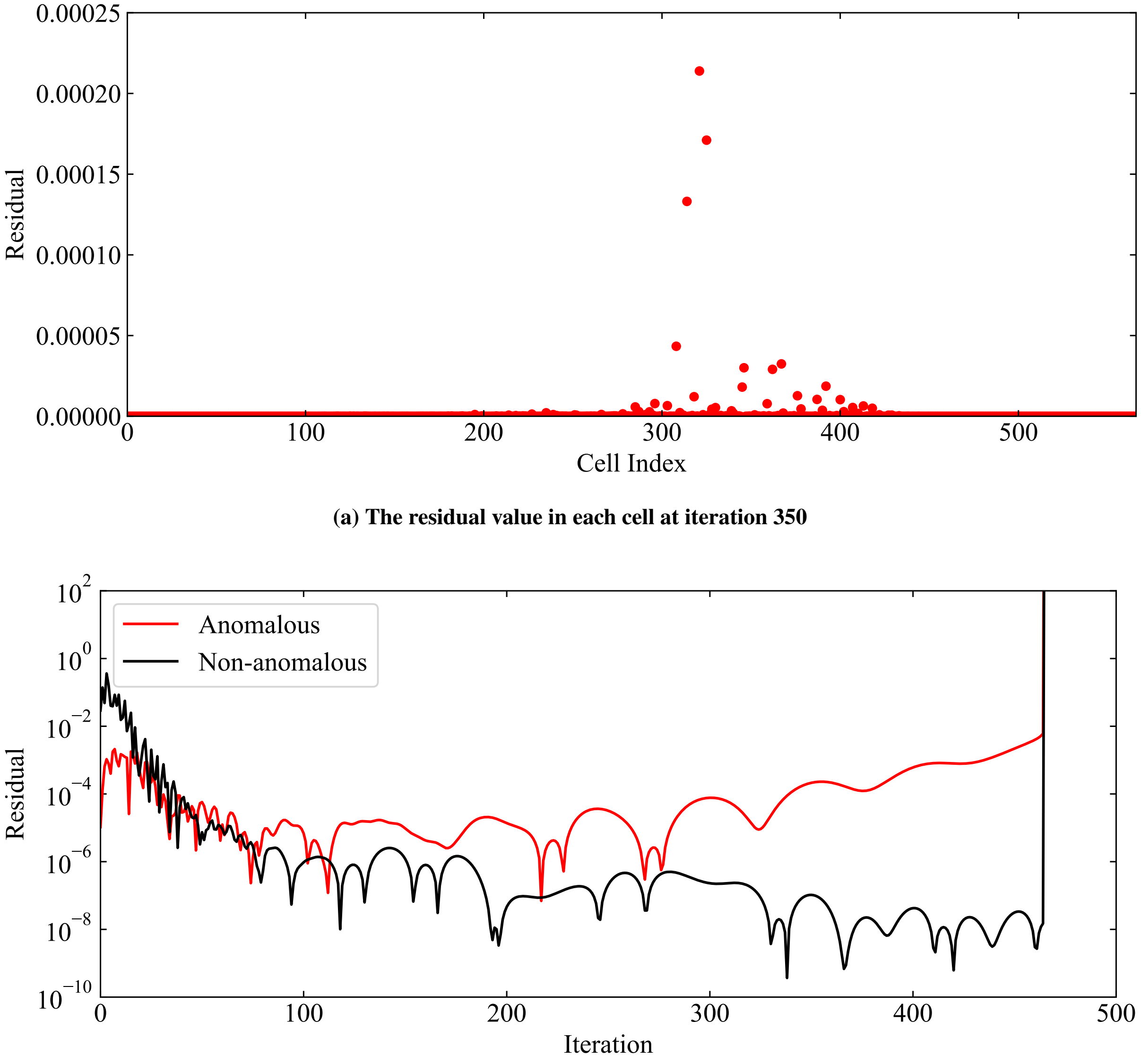
Fig. 4 The residual history in a Burgers problem

The residual vector at the correct non-linear iteration of the solver can have a similar shape to the unstable modes of the solution. However, unlike the right eigenvectors that only point to a single mode of the solution, the residual vector may be a linear combination of different modes. As we know, the unstable solution modes cause the residual to grow in magnitude in the problematic areas of the mesh. As a result, we can extract a synthetic vector from the residual vector which points to the problematic areas in the mesh by detecting the outlier values. In this approach, we analyze the residual vector at every iteration of the solver and look out for outlier values that are growing in magnitude. After selecting a certain iteration in the solution, the control volume corresponding to the maximum residual entry is identified. The next step is to build a synthetic vector that includes the residual values of the selected cell and all the control volumes in its Jacobian fill. This approach isolates the selected vector to only the cells that are present in the Jacobian fill of the cell with the largest residual entry. The unstable mode is growing at a faster rate than all other modes in the solution (usually the magnitude difference is of multiple orders). This means that isolating the search to a certain area of the mesh where the residual vector is maximum in magnitude is a plausible approach to estimating an unstable eigenmode. Furthermore, this process is reversible and in the case where the selected vector does not help with the optimization process, we can go back to select a different vector to do the job. Note that the computational cost of analyzing the residual vector for anomalies is negligible compared to a single non-linear solution iteration which makes it feasible to perform at every iteration. Figures 1a and 2a show synthetic vectors generated using the residual vector in the Euler and Burgers problems, respectively.

B. Anomaly Detection

Anomaly detection has been an important aspect of research in various fields of science and associated applications. Some anomaly detection methods are developed for specific fields while others have more general applications. The main idea is to look for and detect patterns in the data that do not conform to the overall behavior of the problem. Usually, anomalies in the data translate to critical information in a wide variety of applications. In the scope of the present study, anomalies in the residual vector can indicate the unstable solution modes which provide crucial information for solution stability and convergence rate. An anomaly detection problem is a challenging task that requires careful consideration of different factors such as defining the normal behavior that encompasses the overall patterns in the data and defining the deviation necessary to categorize a data point as an outlier. These factors are often domain-specific and can vary greatly for different applications. Detailed surveys on anomaly detection methods and applications can be found in the literature [29–32].

The nature of anomaly detection methods can be categorized into three groups; point anomalies, contextual anomalies, and collective anomalies [32]. In the case of residual values in each cell, specific control volumes might have anomalous values in that specific non-linear iteration while having normal behavior at other iterations. Such patterns can be studied in the form of contextual anomalies in which *time* or *solution iteration* is a contextual attribute that determines the position of an instance on the entire sequence. In this case, the anomalous behavior is identified utilizing the values for the behavioral attributes within a specific context. Here, a data point might be considered anomalous in a given context but the same instance can be considered normal behavior in a different context. The anomalies in the present work are studied both from the temporal as well as spatial points of view. Figure 5a shows the spatial distribution of the residual values inside each cell at the non-linear iteration 350 in the Euler problem presented in Figure 3a. As seen here, there are some outlier values with different spatial behavior than the rest of the cells. Figure 5b shows the time-series behavior of the cell with the largest residual value at iteration 350 (Figure 3a) compared to a cell with non-anomalous behavior. As indicated, the residual value in the non-anomalous cell continues to decrease even after the problematic cell exhibits unstable residual growth. Another aspect of anomaly detection herein is the fact that dominant unstable modes in the solution will have a constant shape during solution iteration and only grow in magnitude. As a result, we are looking for patterns that have a constant normalized shape, contain outliers at each iteration, and grow in magnitude with solution iteration. This novel residual analysis model can be added to the stabilization scheme to remove the need for eigenanalysis which is the most time-consuming module in the previous studies [12, 13].



(a) The residual value in each cell at iteration 350

(b) The residual value at each iteration in the problematic cell versus a non-problematic cell

Fig. 5 The spatial and temporal residual behavior in an Euler problem

The anomalous behavior in different problems is often dynamic which means that new types of anomalies might arise and the points that were anomalous before are not considered outliers anymore compared to the new values. This makes labeling the anomalies beforehand a strenuous and infeasible task. Methods trained in supervised anomaly detection assume the availability of labeled training data while unsupervised methods work with unlabeled data. In the latter case, it is assumed that normal behavior is more prevalent than anomalous patterns. For practical reasons, unsupervised anomaly detection models are utilized in the scope of the present study. The goal is to get labeled outputs from the outlier detection module that categorize the residual value in each cell as normal or an anomaly at each non-linear iteration of the solver (spatial analysis). Individual cells are also scanned for anomalous behavior in time (temporal analysis). We know that the unstable solution modes are highly local to certain areas in the mesh [12]. As a result, certain cells in the solution exhibit anomalous residual growth which indicates the problematic areas in the mesh. Scanning each cell in time to detect the growing trend can help with the identification of the correct unstable modes.

Different anomaly detection methods can be used for outlier classification in the present study; Nearest Neighbor-Based, Clustering-Based, and Statistical anomaly detection techniques are a few examples. Nearest Neighbor-based techniques are the anomaly detection methodology adopted herein. This family of models is data-driven and unsupervised which does not make any assumptions about the normal distribution of the data. This is important in our application as the spatial distribution of the residual values can change dramatically from one iteration to another. There is also the possibility of labeling certain cells as normal or anomalous and utilizing the method in a semi-supervised manner. Further, such methods require tweaking a small number of hyper-parameters for a successful outcome which makes them much easier to work with. This approach, however, can miss the unstable oscillatory solution behavior which is caused by the dominant unstable modes with large imaginary parts. In such cases, the residual value in the problematic cells will undergo constant phase change [33] which makes the job much more difficult for the outlier detector. Furthermore, the computational complexity of basic nearest neighbor methods is $O(N^2)$ [32] which is not ideal. However, it is still an improvement compared to the basic computational complexity of the eigensolver which is of $O(N^3)$ [12].

The next family of methods that are utilized for anomaly detection in the present work is clustering-based techniques. In this approach, the normal data is considered to be a cluster that does not include anomalies. DBSCAN [34] is a well-known method in this family of outlier detectors in which not all points are forced to be members of clusters which saves computational resources. Fixed width clustering can be optimized to have a computational complexity of $O(N\log(N))$ [35] which is a significant improvement over the previous methods. Scikit-learn [36] is an open-source Python library integrating a wide range of such machine learning algorithms. This library is used as the main tool for anomaly detection in the present study which provides an easy-to-use interface without sacrificing much computational performance.

IV. Alternative Gradients

After finding the cells with anomalous residual behavior, the next step is to construct synthetic vectors that resemble the unstable solution modes. Herein, the control volume with the largest residual value is selected and a synthetic vector is constructed with the values inside the cells that are present in the Jacobian row of that particular control volume. Later, this vector is used to find the eigenvalue gradients with respect to mesh vertex movement using Equation 9. Note that, both left and right eigenvectors are assumed to be the same as the newly constructed synthetic vector s. As a result, the eigenvalue gradient can be found as follows,

$$\frac{d\lambda}{d\zeta} = s^T \frac{dA}{d\zeta} s \tag{12}$$

The synthetic vector s in this equation is purely real which results in $s^H = s^T$.

This approach can be repeated throughout the simulation process to mitigate any unstable mode that is detected by looking at the residual vector. In some cases, especially when the residual in the cells in question exhibit oscillatory behavior, the resulting vertex movement vector might increase the magnitude of the residual. In this case, the process can be reverted and the optimization can be performed in a different non-linear solution iteration. The computational cost of each optimization iteration is negligible compared to the cost of a single solution iteration which makes this process feasible.

Table 1 presents the eigenvalue gradients of the dominant unstable eigenmode using the eigenvectors for the Euler and Burgers problems (Figures 1 and 2). This table also presents the gradient of the synthetic vector (constructed using the residual vector) for the two problems at iterations 350 and 11, respectively. The cosine of the angle between the two gradient vectors (found using the eigenvectors and the synthetic vector) is 0.9515 for the Euler problem and 0.9997 for the Burgers problem. As depicted, the gradients calculated using the synthetic vector point in a similar direction as the gradients calculated using the eigenvectors. This proves the accuracy of the presented approach in finding correct movement directions for the selected vertex in the mesh.

Table 1 The eigenvalue gradient with respect to the movement of the suggested vertex

Physics	Vector	Eigenvalue	Gradient w.r.t. x	Gradient w.r.t. y
Euler	Eigenvector	$0.9389 + 0I$	$183.04 + 0I$	$746.38 + 0I$
	Synthetic Vector	NA	$85.06 + 0I$	$137.75 + 0I$
Burgers	Eigenvector	$11.9979 + 0I$	$2460.11 + 0I$	$657.23 + 0I$
	Synthetic Vector	NA	$518.99 + 0I$	$125.74 + 0I$

The eigenvalue gradient $\frac{\partial \lambda}{\partial \zeta}$ can be expanded in real and imaginary parts along direction ζ_i in space as follows,

$$\frac{\partial \lambda}{\partial \zeta_i} = \Re\left(\frac{\partial \lambda}{\partial \zeta_i}\right) + \Im\left(\frac{\partial \lambda}{\partial \zeta_i}\right) I \tag{13}$$

The gradient has a contribution of $M_{\zeta_i} = \Re(\frac{\partial \lambda}{\partial \zeta_i})$ in changing the real value of the unstable eigenmode. Note that our goal is to push the unstable eigenmode to the left side of the eigenspectrum along the real axis. As a result, to change the eigenvalue λ to λ' according to Equation 11 we can use a scaled movement magnitude along ζ_i as follows,

$$\delta_{\zeta_i} = \frac{\lambda' - \lambda}{M_{\zeta_i}} \left(\frac{|M_{\zeta_i}|}{\sum_i |M_{\zeta_i}|}\right) \tag{14}$$

As seen, in this approach, the vertex movement components along different directions in space are equal and only differ by sign. In practice, this means that the components of the gradient from the synthetic vector only need to be accurate enough to have the same sign as the gradient from the eigenvector. This is while in our case the gradient from the synthetic vector is very close to the approach adopted by Zandsalimy and Ollivier-Gooch [12].

Figure 6 shows a contour plot of the real part of the dominant eigenvalue with changes in vertex location in the Burgers problem presented in Figure 2. Here, the location of the vertex indicated with a white circle is changed many times and the resulting Jacobian matrix is analyzed for the dominant eigenmodes. As depicted, for vertex locations on the cyan side, the real part of the dominant eigenmode is negative which means the solution is stable. On the contrary, if the vertex lies on the pink side, the dominant mode is unstable and has a positive real part. As shown here, there is a clear boundary between the unstable and stable regions. Note that many different vertex movement vectors result in the stabilization of the original problem. This large window for vertex movement makes the accuracy of our approximation to the gradient of the eigenvalue less critical.

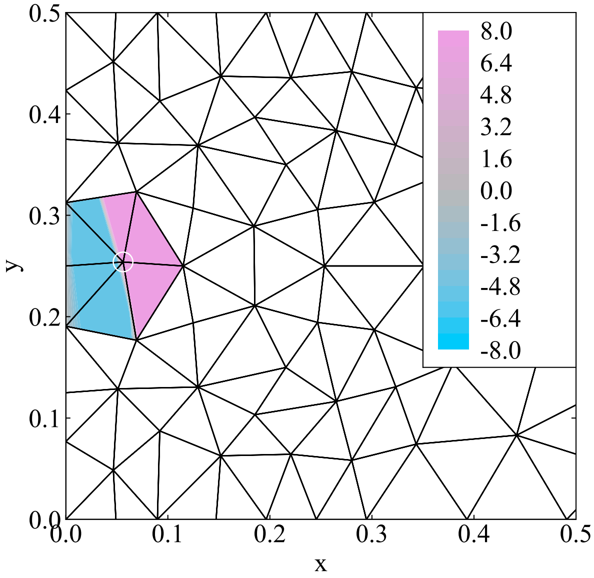


Fig. 6 The real part of the dominant eigenvalue versus vertex location in a Burgers problem

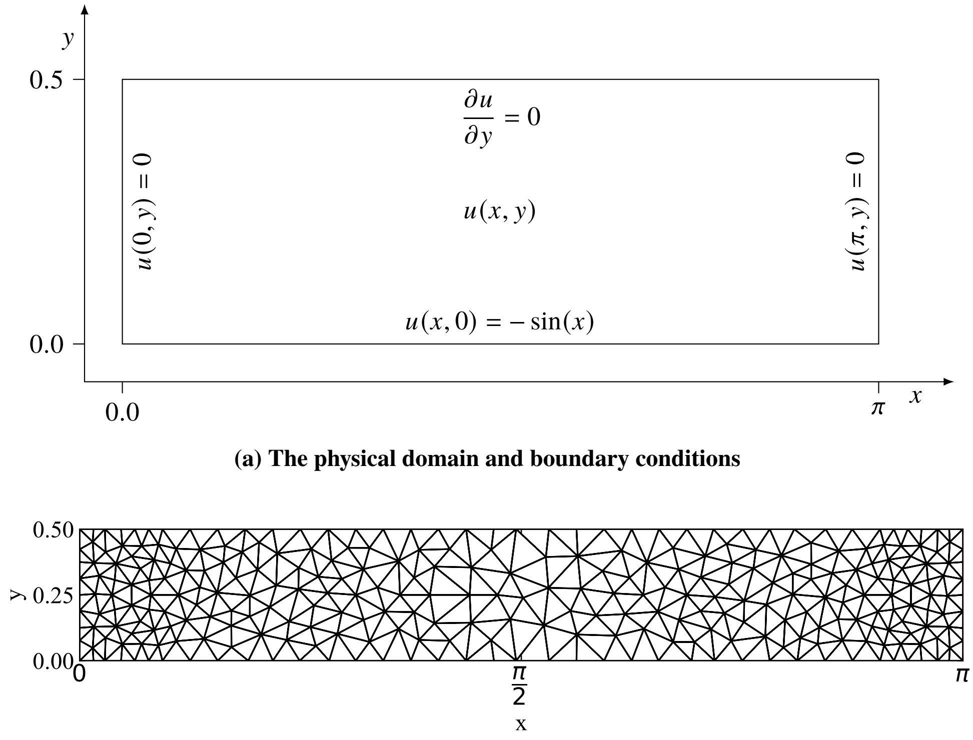
V. Results

This section demonstrates the results obtained from the novel mesh optimization and stability improvement algorithm presented herein. The feasibility of the novel methods is evaluated and the results are gauged against previous studies. Lastly, the contributions to reducing the computational resource requirements are assessed.

A. Mesh Optimization

1. Burgers Test A

The non-linear inviscid Burgers problem, $\frac{\partial u}{\partial y} + u\frac{\partial u}{\partial x} = 0$, is selected as a first test case. This problem is solved on a $\pi \times 0.5$ rectangular channel with the boundary conditions presented in Figure 7a. An example of the unstructured mesh used for the solution of this problem using finite-volume methods is presented in Figure 7b. This mesh is considered high-quality according to the traditional mesh quality guidelines. The solution to this problem is performed using Crank-Nicolson time-stepping method. The original solution on a mesh with 500 control volumes is unstable with the residual history presented in Figure 4b. As seen in Figure 4a, the residual in individual cells contains anomalous values after iteration 10. The residual value inside all the cells at iteration 11 of the solver is presented in Figure 8. The residual vector contains anomalous values as depicted in this figure. The anomaly detection module also selects the four cells with the largest residual values as anomalous control volumes. The application of our novel vertex selection and mesh modification approach at iteration 11 of the solver results in the full stabilization of the solution. The residual history of this problem before and after a single optimization iteration is presented in Figure 9a. This figure also depicts the residual history after optimization of the same problem from Zandsalimy and Ollivier-Gooch [12]. As seen, the convergence history after optimization in the present study closely follows that work. The original and optimized meshes are presented in Figure 9b which shows the location of a single vertex is modified for a stable problem. Figure 9c depicts the eigenspectrum of the semi-discrete Jacobian matrix before and after mesh optimization. As seen here, there is a single pure real unstable mode in the original problem. A single iteration of the novel optimization procedure results in pushing the unstable eigenvalue to the stable region.



(a) The physical domain and boundary conditions

(b) An example of the unstructured mesh

Fig. 7 The domain, boundary conditions, and unstructured mesh for the solution to the Burgers problem [12]

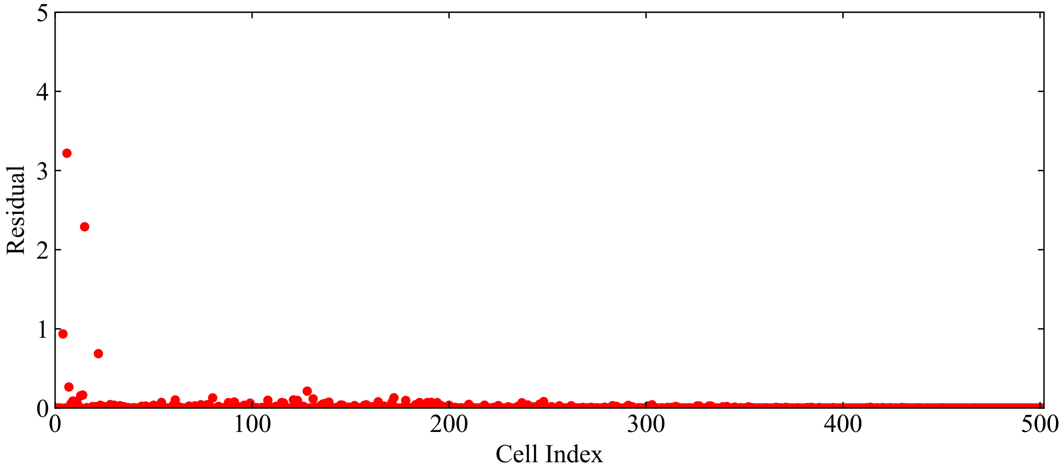
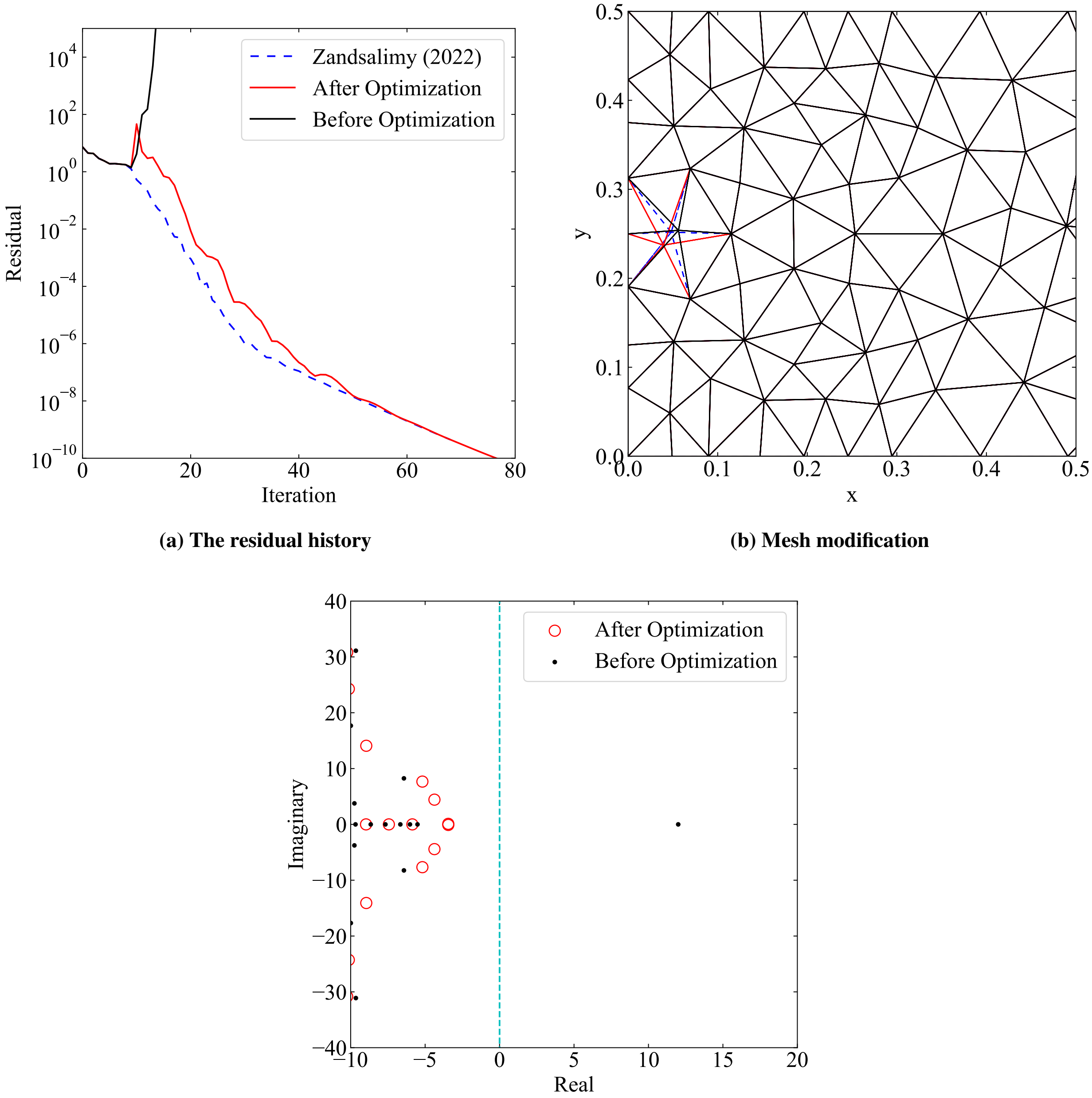


Fig. 8 The residual value in each cell at iteration 11 of a Burgers problem



(a) The residual history

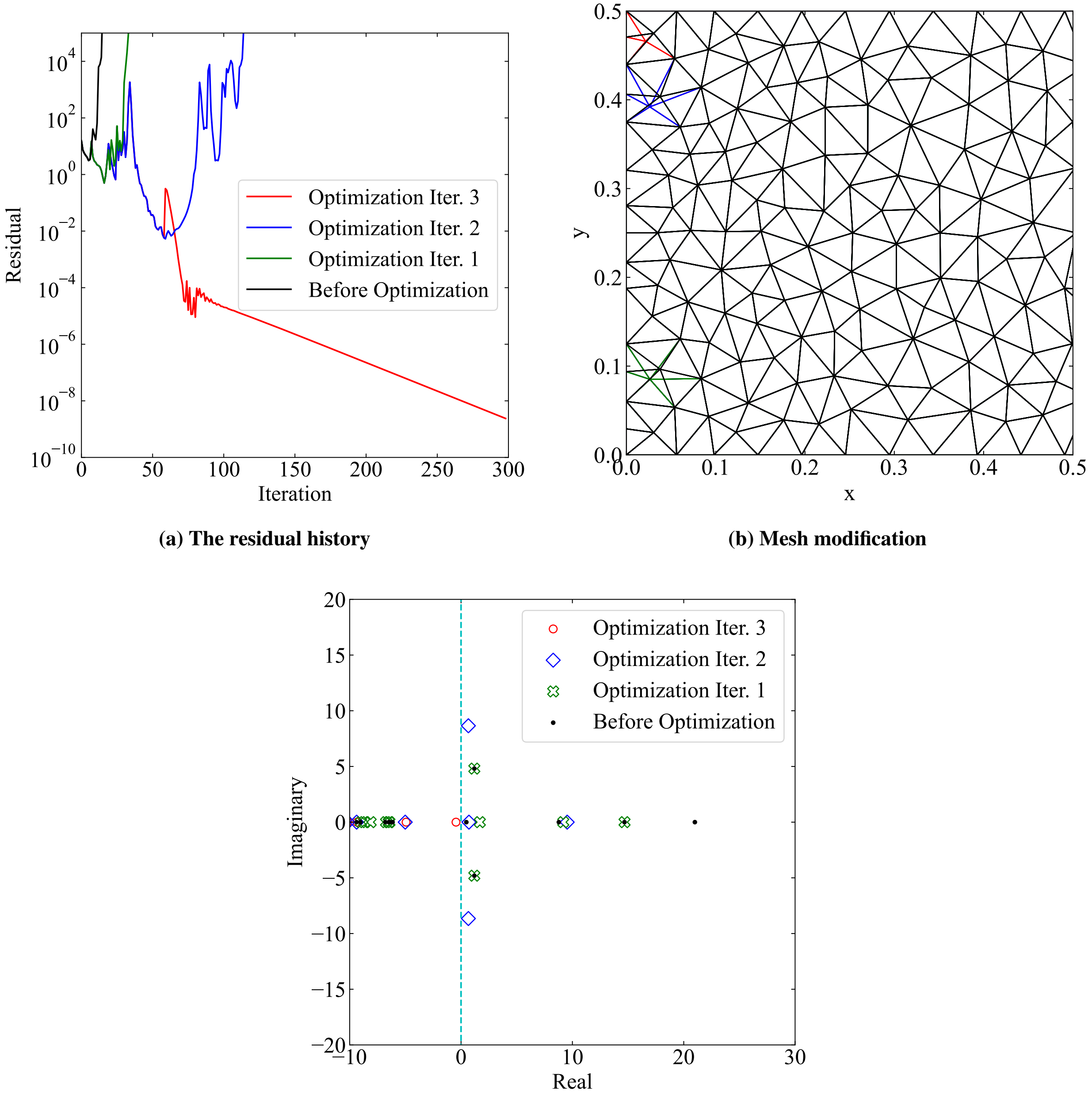
(b) Mesh modification

(c) The eigenspectrum

Fig. 9 Mesh optimization in a Burgers problem on an unstructured mesh with 500 cells

2. Burgers Test B

The next solution to the Burgers problem is performed using Crank-Nicolson time-stepping method on a mesh with 1100 control volumes. This simulation is unstable with the residual history presented in Figure 10a. This problem contains five unstable modes, one of which is a pair of conjugate values. The anomaly detection module shows some outlier values at the non-linear iteration 8 of the solution. We use the presented approach to construct a synthetic vector out of the residual vector at this iteration and perform the optimization, resulting in the residual history presented in green in Figure 10a. The location of a single vertex is modified in this step which is presented in Figure 10b. Figure 10c depicts the eigenspectrum of the original problem as well as after the application of the optimization program. The first iteration of the optimization pushes the dominant unstable mode to the left side of the spectrum as depicted in the eigenspectrum of Figure 10c. However, there are still unstable modes that require remediation. Next, the outlier detection module shows anomalous values in the residual history at iteration 20 of the solver. Performing another iteration of the optimization at this point results in stabilizing another unstable mode as depicted in Figure 10c. However, the simulation is still unstable as there are more unstable modes in the solution. The next iteration of the optimization algorithm is performed at iteration 60 of the solution. This step stabilizes the solution by pushing the remaining 3 unstable eigenvalues to the left side of the spectrum as seen in Figure 10c. As demonstrated in Figure 10b, at each step of the optimization a single vertex is moved to stabilize the corresponding unstable modes.



(a) The residual history

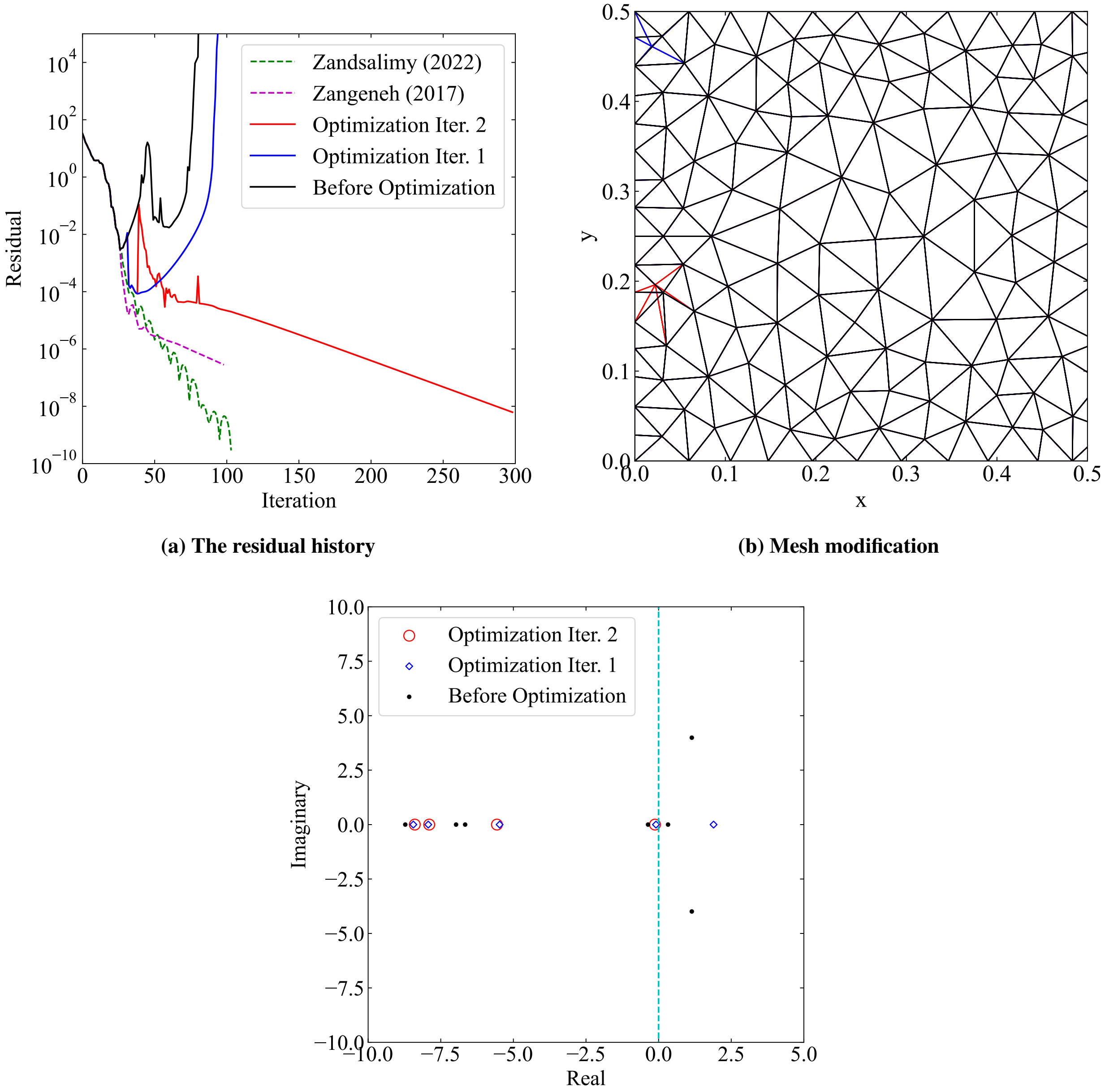
(b) Mesh modification

(c) The eigenspectrum

Fig. 10 Mesh optimization in a Burgers problem on an unstructured mesh with 1100 cells

3. Burgers Test C

Another solution to the Burgers problem is performed using Crank-Nicolson time-stepping method on a mesh with 1400 control volumes. This simulation is unstable with the residual history presented in Figure 11a. This problem contains two unstable modes the dominant of which is a pair of conjugate values. The anomaly detection module shows some outlier values at the non-linear iteration 32 of the solution. Performing the optimization at this non-linear iteration results in the residual history presented in blue in Figure 11a. The modified mesh at this point is presented in Figure 11b indicated with blue. Figure 11c depicts the eigenspectrum of the original problem as well as after the application of the optimization program. The first iteration of the optimization pushes the dominant conjugate pair of unstable modes to the left side of the spectrum as depicted in the eigenspectrum of Figure 11c. However, the single purely real unstable mode still remains in the solution. In the next step, the outlier detection module shows anomalous values in the residual history at iteration 40 of the solver. Performing another iteration of the optimization at this point results in the full stabilization of the problem as depicted in Figure 11c. Figure 11b shows the next vertex location modification in red.



(a) The residual history

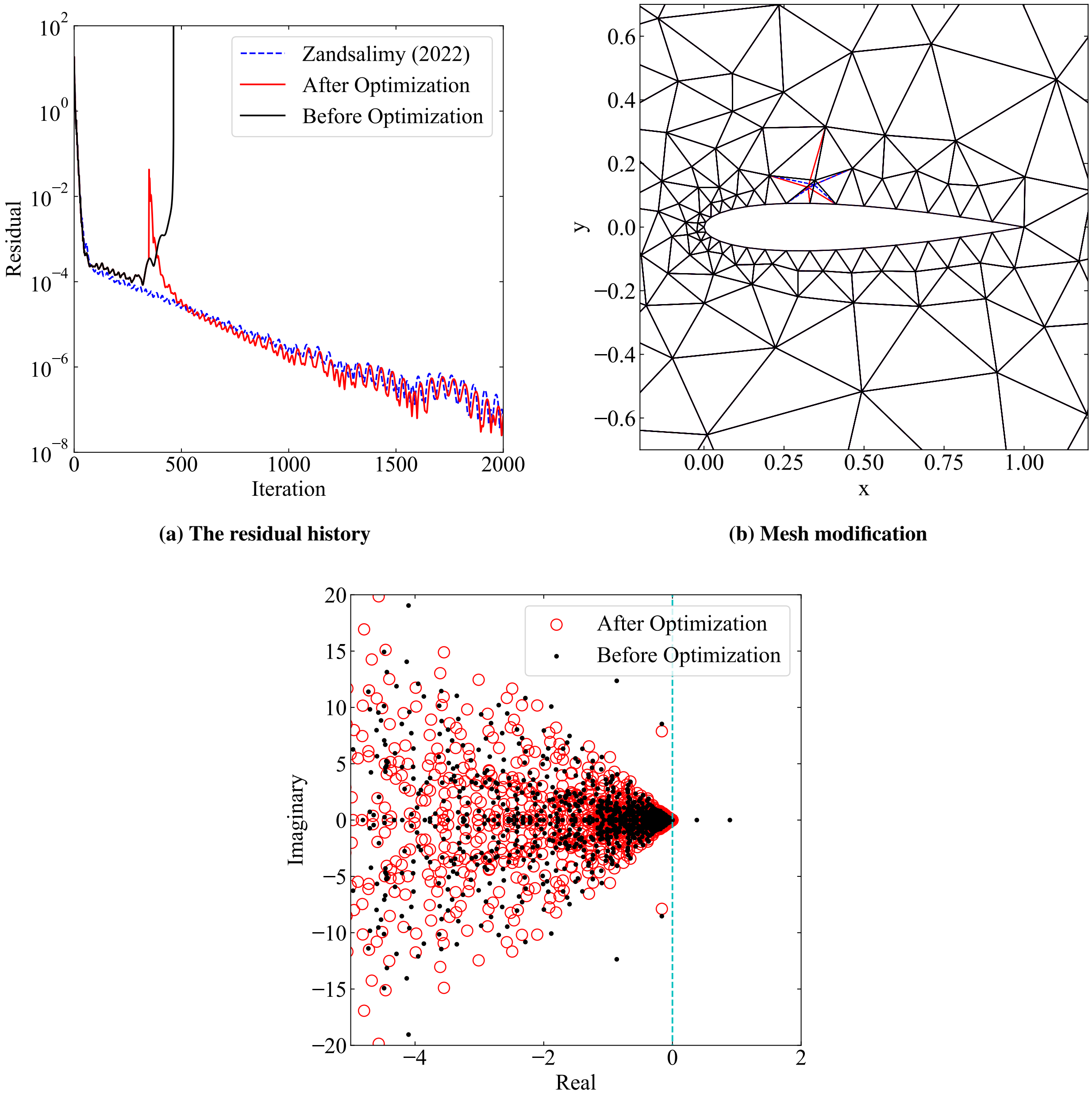
(b) Mesh modification

(c) The eigenspectrum

Fig. 11 Mesh optimization in a Burgers problem on an unstructured mesh with 1400 cells

4. Euler Test A

The Euler problem is selected as the next test case in the present study which is solved around the NACA 0015 airfoil inside a circular domain with a radius of 500 chords. In the initial solution for this problem throughout the paper, density is set to 1.0, velocity in (x, y) direction to $M_\infty(\cos(\alpha), \sin(\alpha))$, and pressure to P_{isen}. In these relations, M_∞ is the free-stream Mach number, α is the angle of attack, and P_{isen} is the resulting pressure during isentropic expansion to M_∞. The first mesh to be tested contains 600 control volumes. The simulation using Crank-Nicolson time-stepping method with $M_\infty = 0.5$ and $\alpha = 0$ is unstable with the residual history presented in Figure 12a. This problem contains two purely real unstable modes. According to the anomaly detection module, outlier values show up in the residual vector at the non-linear iteration 300 of the solver. Performing the optimization at iteration 350 results in the residual history presented in red in Figure 12a. The modified mesh is presented in Figure 12b. Figure 12c depicts the eigenspectrum of the original problem at iteration 350 as well as after the application of the optimization program. The first iteration of the optimization pushes the unstable modes to the left side of the spectrum as depicted. Figure 12a compares the result of mesh optimization from the present study with Zandsalimy and Ollivier-Gooch [12]. As seen, the optimization performed herein stabilizes the dominant unstable mode and the residual history closely follows the result of [12].



(a) The residual history

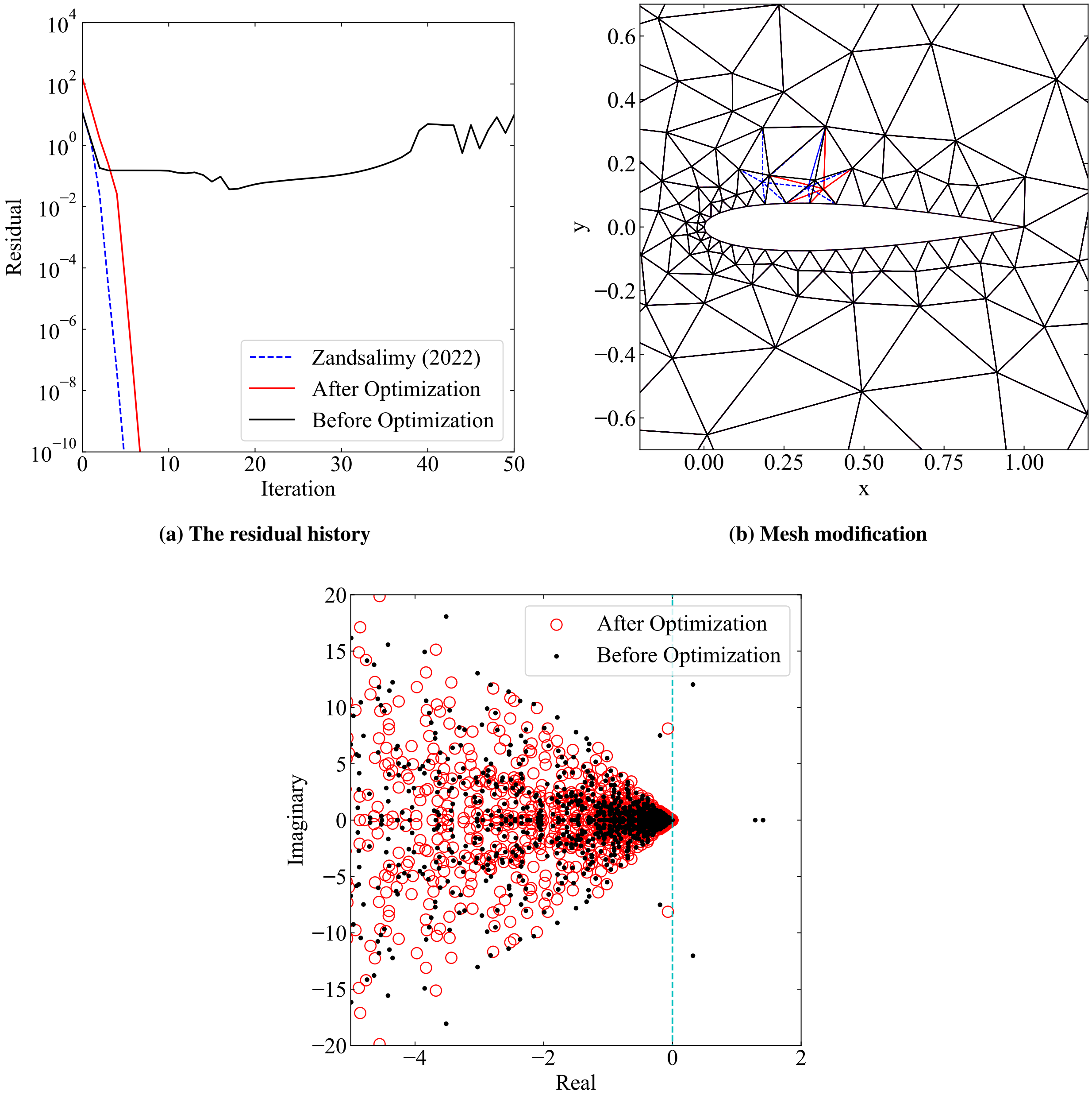
(b) Mesh modification

(c) The eigenspectrum

Fig. 12 Mesh optimization in an Euler problem using Crank-Nicolson time integration on an unstructured mesh with 600 cells

5. Euler Test B

This problem is solved on the same mesh using the Implicit Euler time-stepping method. CFL evolution strategies [37] are utilized to increase the convergence rate of the problem. The original solution is unstable with the residual history presented in Figure 13a. This problem contains three unstable modes one of which is a pair of conjugate eigenvalues. The non-linear iteration 10 of the solver is selected to extract the synthetic vector. Performing the optimization at this point results in the residual history presented in red in Figure 13a. The modified mesh is presented in Figure 13b. As seen, only a single vertex is modified to reach a fully stabilized solution compared to the two vertices in the previous study. Figure 13c depicts the eigenspectrum of the original problem at iteration 9 as well as after the application of the optimization program. The first iteration of the optimization pushes all the unstable modes to the left side of the spectrum and stabilizes the problem. Figure 13a compares the result of mesh optimization from the present study with Zandsalimy and Ollivier-Gooch [12]. As seen here, the convergence rate after optimization is similar to the previous study.



(a) The residual history

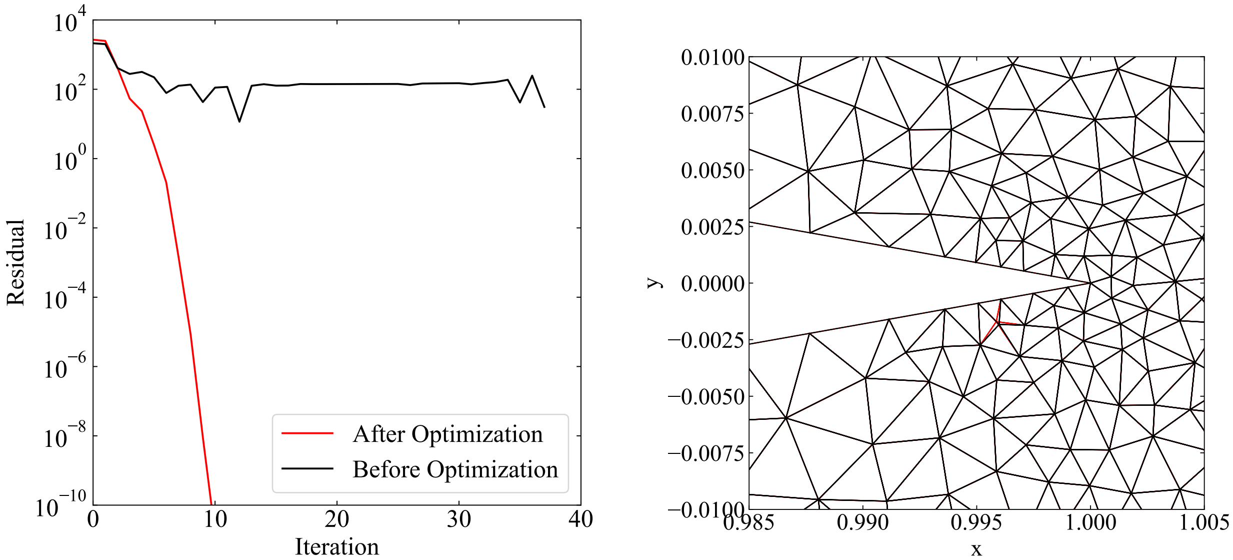
(b) Mesh modification

(c) The eigenspectrum

Fig. 13 Mesh optimization in an Euler problem using Implicit Euler time integration on an unstructured mesh with 600 cells

6. Euler Test C

Another Euler problem is solved using the Implicit Euler time-stepping method on a mesh with 7600 cells. The original solution is unstable with the residual history presented in Figure 14a. The non-linear iteration 20 of the solver is selected to extract the synthetic vector. Performing the optimization at this point results in the residual history presented in red in Figure 14a. The modified mesh is presented in Figure 14b. As seen, only a single vertex is perturbed to reach a fully stabilized solution.



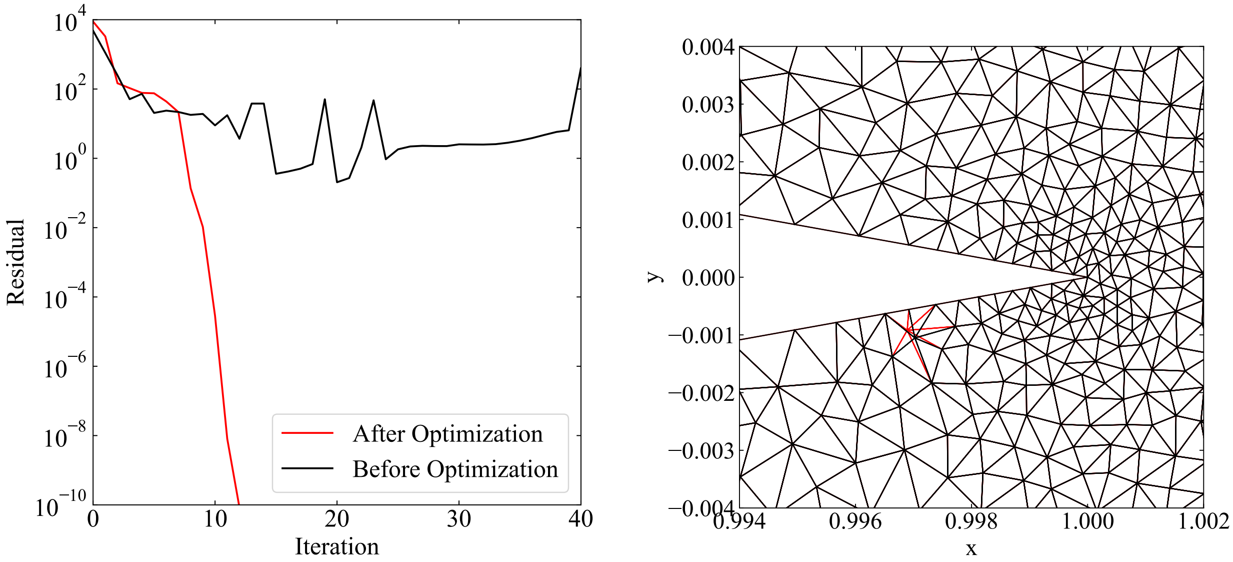
(a) The residual history

(b) Mesh modification

Fig. 14 Mesh optimization in an Euler problem using Implicit Euler time integration on an unstructured mesh with 7600 cells

7. Euler Test D

The same Euler problem is solved using the Implicit Euler time-stepping method on a mesh with 16000 cells. The original solution is unstable with the residual history presented in Figure 15a. The non-linear iteration 20 of the solver is selected to extract the synthetic vector. Performing the optimization at this point results in the residual history presented in red in Figure 15a. The modified mesh is presented in Figure 15b. As seen, only a single vertex is perturbed to reach a fully stabilized solution.



(a) The residual history

(b) Mesh modification

Fig. 15 Mesh optimization in an Euler problem using Implicit Euler time integration on an unstructured mesh with 16000 cells

B. Computational Cost

Figure 16 demonstrates the run-time of different modules in the present work for the solution to an Euler problem in comparison to previous studies. The eigenanalysis module was the most computationally intensive aspect of the work by Zandsalimy and Ollivier-Gooch [12]. This module is eliminated in the present study and replaced with residual vector analysis. The computational time of the flow solver and gradient calculation is the same as before. However, there is a considerable solution-time reduction in the overall computations which is due to the complete elimination of the eigenanalysis module. It is important to note that, unlike the eigenanalysis module, the residual analysis is not the bottleneck of the optimization approach anymore. This is while the residual analysis module is performed in Python programming language which can be much slower than C++. This novel approach to unstable solution mode identification and movement vector calculation enables us to tackle larger problems and improve stability in real-world applications.

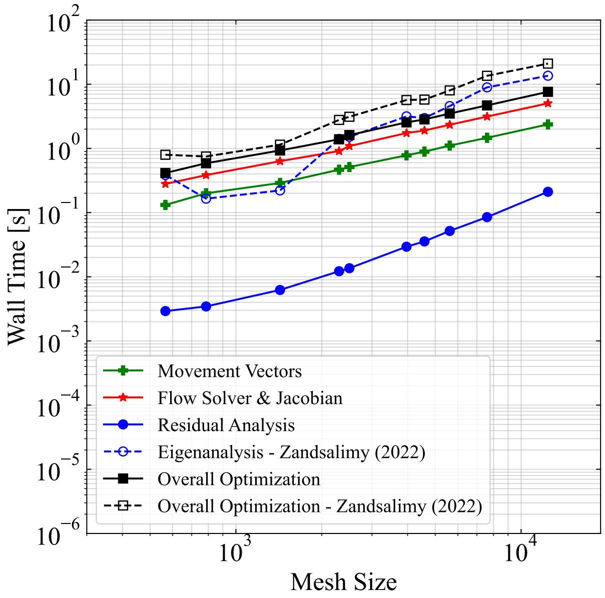


Fig. 16 The optimization run-time of the Euler problem

VI. Conclusion

A novel approach to identifying unstable modes in a finite-volume simulation using the residual vector was presented in the current study. In this approach, a synthetic vector is constructed based on the anomalous behavior of the residual vector. The anomalies in a given residual are evaluated based on their spatial and temporal behavior. The unstable modes of the solution can be identified through outlier values in the residual that point in a constant direction during solution iteration and grow in magnitude. The synthetic vector, if constructed using the residual at a suitable non-linear solution iteration, is shown to be similar to the dominant solutions modes. This helps tremendously in selecting the correct vertices for modification as well as movement vector calculation. We were able to apply the presented stability improvement method to different medium-scale problems without performing the eigenanalysis of the Jacobian matrix. This novel approach to solution mode identification improved the efficiency of our previous method of unstructured finite-volume mesh optimization, reducing the overall computational cost. The new approach was tested on various non-linear simulations and the results were validated against previous studies.

Acknowledgement

This work is co-sponsored by ANSYS Canada and the Natural Sciences and Engineering Research Council of Canada under the Cooperative Research and Development Grant F19-00204.

References

[1] Gustafsson, B., "The Godunov-Ryabenkii condition: The beginning of a new stability theory," *Godunov Methods*, Springer, 2001, pp. 425–443.

[2] Reddy, S. C., and Trefethen, L. N., "Lax-stability of fully discrete spectral methods via stability regions and pseudo-eigenvalues," *Computer Methods in Applied Mechanics and Engineering*, Vol. 80, No. 1-3, 1990, pp. 147–164.

[3] Lax, P. D., and Richtmyer, R. D., "Survey of the stability of linear finite difference equations," *Communications on Pure and Applied Mathematics*, Vol. 9, No. 2, 1956, pp. 267–293.

[4] Citro, V., Luchini, P., Giannetti, F., and Auteri, F., "Efficient stabilization and acceleration of numerical simulation of fluid flows by residual recombination," *Journal of Computational Physics*, Vol. 344, 2017, pp. 234–246. https://doi.org/https://doi.org/10.1016/j.jcp.2017.04.081.

[5] Farhat, C., Chapman, T., and Avery, P., "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International Journal for Numerical Methods in Engineering*, Vol. 102, No. 5, 2015, pp. 1077–1110. https://doi.org/https://doi.org/10.1002/nme.4820.

[6] Xu, H., "An SVD-like matrix decomposition and its applications," *Linear Algebra and its Applications*, Vol. 368, 2003, pp. 1–24. https://doi.org/https://doi.org/10.1016/S0024-3795(03)00370-7.

[7] Wang, L., and Semlyen, A., "Application of sparse eigenvalue techniques to the small signal stability analysis of large power systems," *IEEE Transactions on Power Systems*, Vol. 5, No. 2, 1990, pp. 635–642. https://doi.org/10.1109/59.54575.

[8] Lehoucq, R. B., and Salinger, A. G., "Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers," *International Journal for Numerical Methods in Fluids*, Vol. 36, No. 3, 2001, pp. 309–327. https://doi.org/https://doi.org/10.1002/fld.135.

[9] Arnoldi, W. E., "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of applied mathematics*, Vol. 9, No. 1, 1951, pp. 17–29.

[10] Morzyski, M., Afanasiev, K., and Thiele, F., "Solution of the eigenvalue problems resulting from global non-parallel flow stability analysis," *Computer Methods in Applied Mechanics and Engineering*, Vol. 169, No. 1, 1999, pp. 161–176. https://doi.org/https://doi.org/10.1016/S0045-7825(98)00183-2.

[11] Chen, J., Fu, P., Méndez-Barrios, C.-F., Niculescu, S.-I., and Zhang, H., "Stability Analysis of Polynomially Dependent Systems by Eigenvalue Perturbation," *IEEE Transactions on Automatic Control*, Vol. 62, No. 11, 2017, pp. 5915–5922. https://doi.org/10.1109/TAC.2016.2645758.

[12] Zandsalimy, M., and Ollivier-Gooch, C., "A novel approach to mesh optimization to stabilize unstructured finite volume simulations," *Journal of Computational Physics*, Vol. 453, 2022, p. 110959. https://doi.org/https://doi.org/10.1016/j.jcp.2022.110959.

[13] Zangeneh, R., and Ollivier-Gooch, C. F., "Mesh optimization to improve the stability of finite volume methods on unstructured meshes," *Computers & Fluids*, Vol. 156, 2017, pp. 590 – 601. https://doi.org/https://doi.org/10.1016/j.compfluid.2017.04.020.

[14] Patankar, S., *Numerical Heat Transfer and Fluid Flow*, Series in computational methods in mechanics and thermal sciences, Taylor & Francis, 1980.

[15] LeVeque, R. J., *Numerical Methods for Conservation Laws*, Vol. 3, Springer, 1992.

[16] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer Science & Business Media, 2013.

[17] Eymard, R., Gallouët, T., and Herbin, R., "Finite volume methods," *Handbook of Numerical Analysis*, Vol. 7, 2000, pp. 713–1018.

[18] Zienkiewicz, O. C., Taylor, R. L., Nithiarasu, P., and Zhu, J., *The Finite Element Method*, Vol. 3, McGraw Hill London, 1977.

[19] Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z., *The Finite Element Method: Its Basis and Fundamentals*, Elsevier, 2005.

[20] Chen, L., and Li, R., "An integrated linear reconstruction for finite volume scheme on unstructured grids," *Journal of Scientific Computing*, Vol. 68, No. 3, 2016, pp. 1172–1197.

[21] Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357 – 372. https://doi.org/https://doi.org/10.1016/0021-9991(81)90128-5.

[22] Michalak, C., and Ollivier-Gooch, C., "Globalized matrix-explicit Newton-GMRES for the high-order accurate solution of the Euler equations," *Computers & Fluids*, Vol. 39, No. 7, 2010, pp. 1156 – 1167. https://doi.org/https://doi.org/10.1016/j.compfluid.2010.02.008.

[23] Murray, R., *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 2017.

[24] Bikdash, M. U., and Layton, R. A., "An Energy-Based Lyapunov Function for Physical Systems," *IFAC Proceedings Volumes*, Vol. 33, No. 2, 2000, pp. 81–86.

[25] Sastry, S., *Lyapunov Stability Theory*, Springer New York, 1999, pp. 182–234. https://doi.org/10.1007/978-1-4757-3108-8_5.

[26] Stewart, G. W., "A Krylov–Schur Algorithm for Large Eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, Vol. 23, No. 3, 2002, pp. 601–614. https://doi.org/10.1137/S0895479800371529.

[27] Hernandez, V., Roman, J. E., and Vidal, V., "SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems," *ACM Trans. Math. Softw.*, Vol. 31, No. 3, 2005, p. 351362. https://doi.org/10.1145/1089014.1089019.

[28] Rudisill, C. S., and Chu, Y.-Y., "Numerical Methods for Evaluating the Derivatives of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 13, No. 6, 1975, pp. 834–837. https://doi.org/10.2514/3.60449.

[29] Ahmed, M., Naser Mahmood, A., and Hu, J., "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, Vol. 60, 2016, pp. 19–31. https://doi.org/https://doi.org/10.1016/j.jnca.2015.11.016.

[30] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K., "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, 2014, pp. 303–336. https://doi.org/10.1109/SURV.2013.052213.00046.

[31] Pang, G., Shen, C., Cao, L., and Hengel, A. V. D., "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, Vol. 54, No. 2, 2021. https://doi.org/10.1145/3439950.

[32] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: A Survey," *ACM Computing Surveys*, Vol. 41, No. 3, 2009. https://doi.org/10.1145/1541880.1541882.

[33] Chen, L., "Stability analysis and stabilization of unstructured finite volume method," Master's thesis, University of British Columbia, 2016. https://doi.org/http://dx.doi.org/10.14288/1.0300002, URL https://open.library.ubc.ca/collections/ubctheses/24/items/1.0300002.

[34] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X., "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Transactions on Database Systems*, Vol. 42, No. 3, 2017. https://doi.org/10.1145/3068335.

[35] Khan, K., Rehman, S. U., Aziz, K., Fong, S., and Sarasvady, S., "DBSCAN: Past, present and future," *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 2014, pp. 232–238. https://doi.org/10.1109/ICADIWT.2014.6814687.

[36] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[37] Bücker, H. M., Pollul, B., and Rasch, A., "On CFL evolution strategies for implicit upwind methods in linearized Euler equations," *International Journal for Numerical Methods in Fluids*, Vol. 59, No. 1, 2009, pp. 1–18. https://doi.org/https://doi.org/10.1002/fld.1798.