Approximate Jacobian Eigenanalysis For Unstructured Mesh Optimization of Finite Volume Simulations

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

This paper presents a novel method for approximate eigenanalysis of large linear systems, with a specific focus on unstable eigenvalues in the Lyapunov sense. The method utilizes residual vector analysis, principal component analysis, and dynamic mode decomposition to identify unstable solution modes efficiently. Outlier detection models are employed to find the problematic cells on the unstructured mesh in the case of residual vector and principal component analysis while the dynamic mode decomposition points directly to the control volumes of interest without such help. By extracting and assembling corresponding rows of the Jacobian matrix, the large system is projected on a new matrix with significantly fewer degrees of freedom, leading to improved computational efficiency. The eigenvalue problem is then solved on this smaller matrix, and the obtained results are utilized for mesh optimization, enhancing the local stability of the dynamic system. By addressing the challenges of computational complexity and automation faced by previous methods, this novel approach offers a comprehensive and automated solution for the eigenanalysis of large linear systems. The potential impact of this method extends to various fields, providing a more efficient eigenanalysis process and opening new avenues for exploring and optimizing complex systems.

I. Introduction

The eigenanalysis of the Jacobian matrix has been utilized as a powerful tool in mesh optimization for the local stability improvement of unstructured mesh finite volume simulations. Recent studies have focused on finding alternative methods to the eigenanalysis module, aiming to reduce computational costs. These methods include outlier detection applications, residual vector analysis, Principal Component Analysis (PCA) of solution vectors, and Dynamic Mode Decomposition (DMD) of solution vectors. Notably, the eigenanalysis module was identified as the most resource-intensive aspect of the mesh optimization approach proposed by Zandsalimy and Ollivier-Gooch [1]. Therefore, eliminating this module from the algorithm could lead to significant computational savings. In this paper, we present a novel approach to achieve a similar reduction in computational requirements by estimating the eigenvalues and eigenvectors of the Jacobian matrix through our proposed method.

Zandsalimy and Ollivier-Gooch [1] presented a novel method for mesh optimization in unstructured finite volume simulations, considerably improving the efficiency of a previous method. They present a new strategy for vertex selection in the mesh and an improved overall mesh optimization approach, resulting in significant efficiency improvements. Zandsalimy and Ollivier-Gooch [2] utilized anomaly detection models to identify outlier values in the residual vector at every iteration of the solver for possible unstable solution mode identification. This method was introduced to replace the eigenanalysis module and was shown to work well on problems with purely real dominant unstable solution modes. Synthetic vectors were formed out of the anomalous residual vectors, resembling the unstable eigenvectors of the solution. Further, Zandsalimy and Ollivier-Gooch [3] utilized a different anomaly detection model on the PCA solution modes for control volume selection and synthetic vector formation. This method was shown to be more effective in the stabilization scheme to replace the eigenanalysis module as the PCA solution modes are more similar in shape to the solution modes than the residual vector. Zandsalimy and Ollivier-Gooch [4] utilized the dynamic mode decomposition of the solution vectors for unstable solution mode identification. DMD provides the time dynamics of the dominant solution modes including the magnitude growth rate as well as the oscillation frequency of each mode at a small computational cost. As a result, it was possible to apply DMD on a collection of the most recent solution vectors at every iteration of the solver. Mesh optimization through this method does not require the eigenanalysis of the large Jacobian matrix or an outlier detection model as the unstable solution modes are detected automatically in the DMD process.

*Ph.D. Candidate, Mechanical Engineering, The Advanced Numerical Simulation Laboratory.
†Professor, Mechanical Engineering, The Advanced Numerical Simulation Laboratory, AIAA Associate Fellow.

This method was also shown to be an effective mesh optimization strategy on unstructured finite volume simulations for the improved stability and convergence rate of the solver.

The current study aims to leverage computationally efficient methods for identifying unstable solution modes in order to extract only the crucial information from the Jacobian matrix. By selecting specific entries from the complete Jacobian, a new matrix with significantly fewer degrees of freedom is constructed, effectively projecting the Jacobian onto a reduced space. Experimental evidence demonstrates that this smaller matrix exhibits similar eigenvalues and eigenvectors as the full Jacobian matrix. Successful implementation of this approach enables us to utilize eigenanalysis of the projected approximate Jacobian in the mesh optimization approach proposed by Zandsalimy and Ollivier-Gooch [1]. This advancement leads to substantial computational savings, as the eigenanalysis module was previously the most resource-intensive component of the methodology. As demonstrated in this work, the novel Jacobian reduction method accurately captures the correct eigenmodes and eigenvectors of the original problem while incurring minimal computational costs. Moreover, this innovative data reduction technique can be applied to any application that involves eigenanalysis of large matrices, provided knowledge of the non-zero pattern of the desired eigenvectors.

II. Background

A. Numerical Simulation Scheme

The cell center finite volume method [5] is utilized for the solution to CFD problems herein with 2^{nd}-order spatial accuracy. Roe's scheme [6] is used for inviscid flux calculation from the piece-wise linear solution approximation reconstructed from the piece-wise constant control volume averages. The high-quality unstructured mesh is produced using the Generation and Refinement of Unstructured, Mixed-Element Meshes in Parallel (GRUMMP) software [7]. The boundary conditions are applied weakly using flux values on the boundaries. The Jacobian matrix can be determined using the finite-difference method or more efficiently by chain rule differentiation [8] as follows,

$$\boldsymbol{A} = \frac{\partial \vec{R}}{\partial \vec{U}} = \frac{\partial \text{FluxInt}}{\partial \text{Flux}} \frac{\partial \text{Flux}}{\partial \text{RecSol}} \frac{\partial \text{RecSol}}{\partial \text{RecCoef}} \frac{\partial \text{RecCoef}}{\partial \text{PVars}} \frac{\partial \text{PVars}}{\partial \text{CVars}} \tag{1}$$

in which, $\vec{R}$ and $\vec{U}$ are the residual and solution vectors, 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 [8].

B. Solution Stability

The Lyapunov stability theory [9], plays a crucial role in determining the stability of linear time-invariant systems. According to this theory, local stability can be established when all eigenvalues of the coefficients matrix possess non-positive real parts. Conversely, the presence of an eigenvalue with a positive real part indicates system instability. Furthermore, the real and imaginary components of each solution mode provide valuable insights into the growth rate and oscillation frequency, respectively.

Recent studies have leveraged this theory to analyze and enhance the stability of finite volume methods for unstructured mesh optimization. By examining the eigenspectrum of a given linear system, unstable solution modes can be identified. Adjusting problem parameters allows modification of the real and/or imaginary parts of the corresponding eigenvalue, thereby improving stability and convergence rate. Consequently, any simulation parameter that significantly influences the eigenvalues of the coefficients matrix can be considered a viable candidate for manipulation. In the context of linear systems resulting from the discretization of Partial Differential Equations (PDEs), examples of such factors include the discretization scheme, reconstruction stencil, time step size, and numerical grid.

C. Mesh Optimization

The stabilization scheme by Zandsalimy and Ollivier-Gooch [1] identifies the mesh vertices whose locations have the largest effect on the unstable eigenmodes. Modification of these vertices can result in improved stability and convergence rate of the numerical simulation. An overview of this approach is presented in the following.

Eigenanalysis In Lyapunov stability theory, the unstable modes can be found on the right half of the eigenspectrum of the Jacobian matrix. Performing eigenanalysis on the Jacobian matrix helps us identify these modes for

further examination. Spectral transformation methods, such as the Cayley algorithm, are employed to isolate the eigenvalues of interest. To solve the eigenvalue problem, the Krylov-Schur technique [10] is utilized, implemented in the Scalable Library for Eigenvalue Problem Computations (SLEPc) [11].

Vertex Selection In the next step, our focus shifts to identifying problematic cells and vertices within the mesh. These control volumes and vertices significantly influence the Jacobian and, consequently, the eigenvalues of the linear system. By analyzing the gradients of eigenvalues with respect to mesh vertex movement, we can pinpoint such vertices. Specifically, we select vertices with the highest gradients of eigenvalues, as they serve as potential candidates for mesh optimization. However, this approach can be computationally demanding due to the need for multiple solutions to the eigenvalue problem, which should be avoided to minimize computational costs.

To address this limitation, Zandsalimy and Ollivier-Gooch [1] proposed an alternative approach that directly utilizes unstable eigenvectors for vertex identification. In this method, the absolute value of an unstable eigenvector in each cell is added as a selection measure for the adjacent vertices. By employing this technique, we can identify the vertex with the largest selection measure, indicating a significant impact on the eigenvalue in question. Experimental results have demonstrated the effectiveness of this approach as a viable proxy for efficient vertex selection. Furthermore, the computational complexity of this method represents a substantial improvement over the eigenvalue gradient-based approach.

Modification Vector To achieve the most favorable change in the corresponding eigenvalue, the selected vertices can be modified using a movement vector. We can determine these vectors by analyzing the gradient of the eigenvalue with respect to mesh movement. Our objective is to move the vertex using the steepest descent method along the gradient map. Calculating the eigenvalue gradients would involve solving the eigenvalue problem multiple times using the finite difference method. However, due to the high computational cost associated with the eigenvalue problem, we have adopted an alternative approach. We estimate the gradient of an eigenvalue λ with respect to the vertex location ζ using [12],

$$\frac{d\lambda}{d\zeta} = \boldsymbol{y}^H \frac{d\boldsymbol{A}}{d\zeta}\boldsymbol{x} \tag{2}$$

In this equation, $\boldsymbol{y}^H$ is the Hermitian transpose of the left eigenvector and $\boldsymbol{x}$ is the right eigenvector associated with the eigenvalue λ. The term $\frac{d\boldsymbol{A}}{d\zeta}$ can be estimated using,

$$\frac{d\boldsymbol{A}}{d\zeta} = \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \zeta} + \frac{\partial \boldsymbol{A}}{\partial \zeta} \approx \frac{\partial \boldsymbol{A}}{\partial \zeta} \tag{3}$$

We can assume that the changes in solution $\boldsymbol{u}$ with respect to changes in the mesh are negligible.

Mesh Modification In this step, the gradients are utilized to identify modification vectors and to change the locations of the selected vertices. A truncated Taylor series is used for this purpose as follows.

$$\lambda' = \lambda + \frac{d\lambda}{d\zeta}\delta\zeta \tag{4}$$

Here, λ is the selected eigenvalue and λ' is the new modified eigenvalue.

D. Computational Complexity

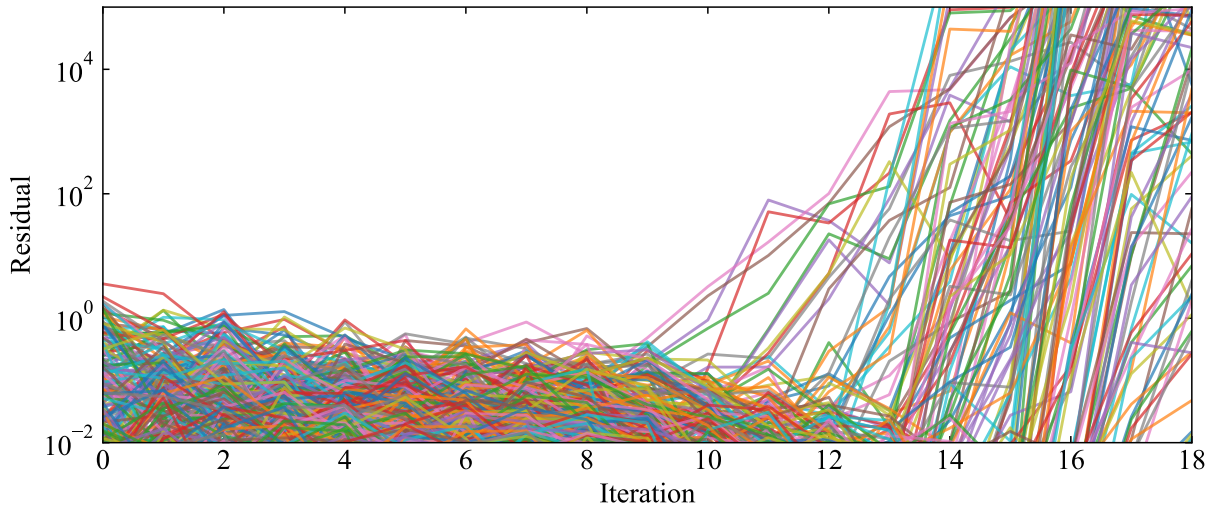
The solution to the large sparse eigenvalue problem is the most computationally demanding module in this mesh optimization approach. For a large sparse matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, computing m eigenvalues requires the following considerations [11],

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 $O(m^3)$.

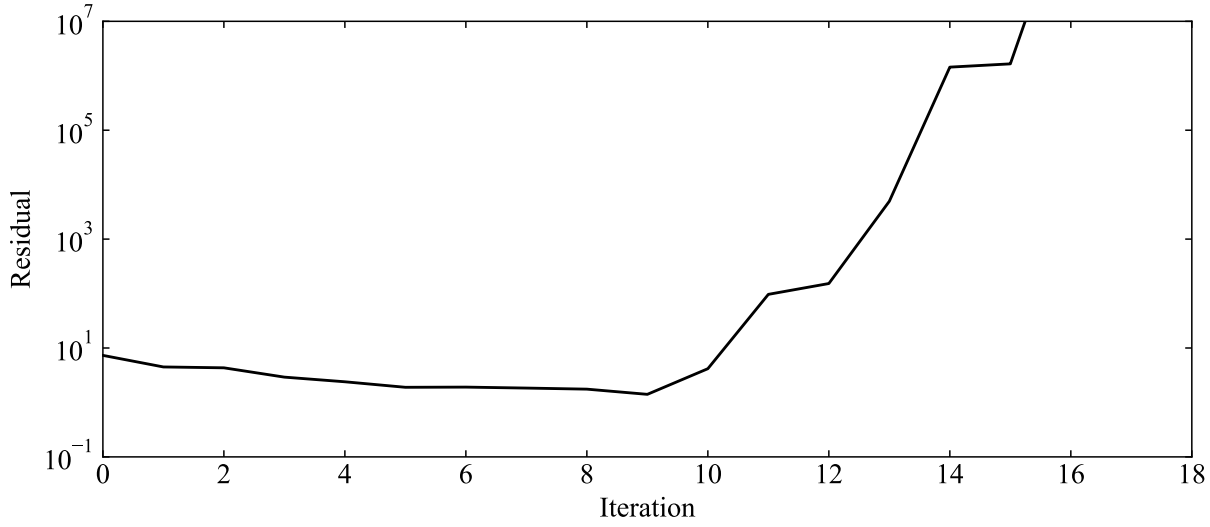
Given the high computational requirements, recent studies have aimed to eliminate the need for eigenanalysis in the optimization procedure. For instance, Zandsalimy and Ollivier-Gooch [2] proposed an approach that utilizes outlier detection models to analyze the residual vector and identify unstable modes in the simulation. Another approach by Zandsalimy and Ollivier-Gooch [3] utilizes the principal component analysis of the solution update vectors for identifying unstable modes. In a different study, Zandsalimy and Ollivier-Gooch [4] employed dynamic mode decomposition of the solution update vectors, which offers a better approximation of the dynamic solution modes compared to previous works.

E. Residual Vector Analysis

According to Zandsalimy and Ollivier-Gooch [2], the residual vector can serve as an alternative metric for vertex identification and movement vector calculation. When the unstable eigenmode is a pure real value, the residual vector exhibits local growth with a constant shape, indicating the dominant solution modes and highlighting problematic areas in the mesh. However, if the dominant eigenmode possesses a non-zero imaginary part, a constant phase change occurs, making it challenging to identify problematic cells in the mesh. Figure 1 illustrates the L_2 norm of the residual vector and individual residual values for all cells in a model Burgers problem. Figure 1a displays the residual history in each cell, with different colors representing different control volumes. Notably, after the 10th solution iteration, certain outlier values begin to increase in magnitude, coinciding with the growth observed in the L_2 norm of the residual vector in Figure 1b.



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



(b) The L_2 norm of the residual vector

Fig. 1 The residual history in a Burgers problem [2]

An outlier detection model is utilized to identify the problematic cells on the mesh using the residual vector at the correct non-linear iteration of the solution. A synthetic vector is formed using the residual value inside the selected cells and their neighbors. This newly formed vector shares a similar shape to the unstable eigenvector in question, but does

not require explicit eigenanalysis of the Jacobian matrix. Further details on this approach can be found in Zandsalimy and Ollivier-Gooch [2]. For the purpose of this study, the focus is solely on identifying problematic control volumes within the mesh.

F. Principal Component Analysis

Applying principal component analysis to a selection of solution update vectors allows us to identify the dominant solution modes. Particularly, when the unstable solution modes are purely real, the dominant PCA modes exhibit outlier values in the problematic cells of the mesh, as demonstrated by Zandsalimy and Ollivier-Gooch [3]. By utilizing an outlier detection model, we can analyze the principal modes derived from the PCA of solution update vectors, aiding in the identification of cells that require modification. This approach also enables the construction of synthetic vectors using the principal components, resembling the unstable solution modes.

PCA is a technique used to extract the most significant information from data by generating a new set of orthogonal vectors known as principal components. This approach is particularly valuable in addressing the computational complexity challenges encountered in the original mesh optimization approach. The principal components in PCA can be obtained through the Singular Value Decomposition (SVD) of a given matrix. SVD is the factorization of a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ into the product $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, where $n > m$. In this factorization, $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{C}^{m\times m}$ are unitary matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix containing the singular values. The columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ represent the left and right singular vectors of $\boldsymbol{A}$, respectively. The SLEPc library [11] provides systematic tools to perform SVD efficiently. By applying SVD to a subset of solution update vectors, significant computational savings can be achieved compared to the eigenanalysis of the Jacobian matrix. For a tall and skinny matrix $\boldsymbol{A} \in \mathbb{C}^{n\times m}$ (with $n \gg m$), the computational complexity of SVD is approximately $O(nm^2)$ [13].

In simulations that include unstable numerical modes, the dominant PCA components exhibit anomalies in certain cells. Consider a model Burgers problem, solved on a mesh with 500 cells using the Crank-Nicolson time-stepping scheme. On this mesh, the solution is unstable and the Jacobian matrix includes a single unstable mode on the right open half of the eigenspectrum. At each iteration of the solution, SVD is performed on the 10 most recent solution update vectors. The largest solution mode at each iteration is presented in Figure 2. As depicted, the solution mode manifests outliers in a few cells near the left boundary at around iteration 11 and after. The outliers amplify with solution iteration, extending to other cells and causing the solution to diverge. This anomalous behavior is generated by the sole unstable numerical mode in the solution. Anomaly detection can be used as a powerful tool in detecting such outlier values to find the correct iteration of the solution to apply the optimization as well as identify the problematic cells for modification.

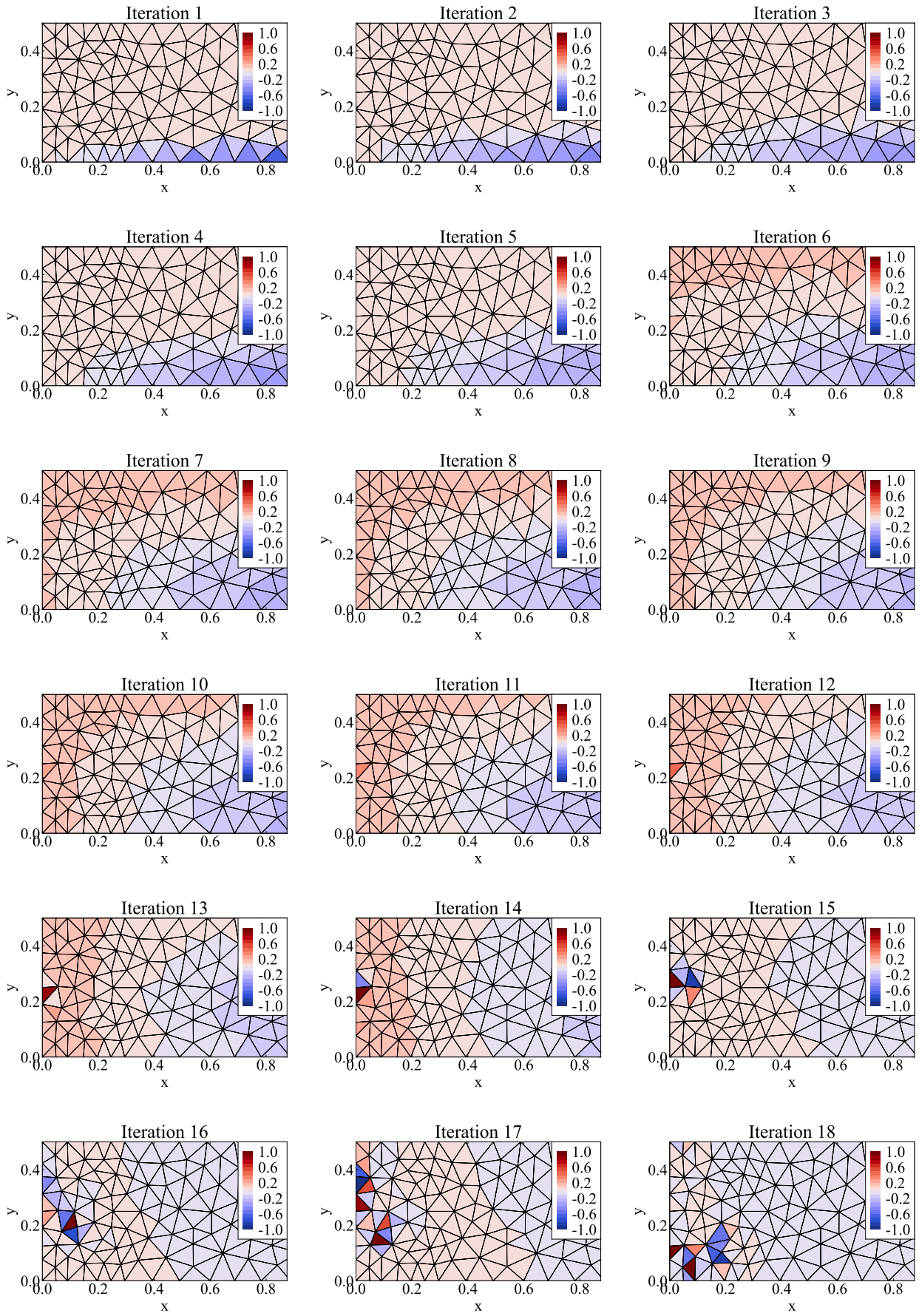
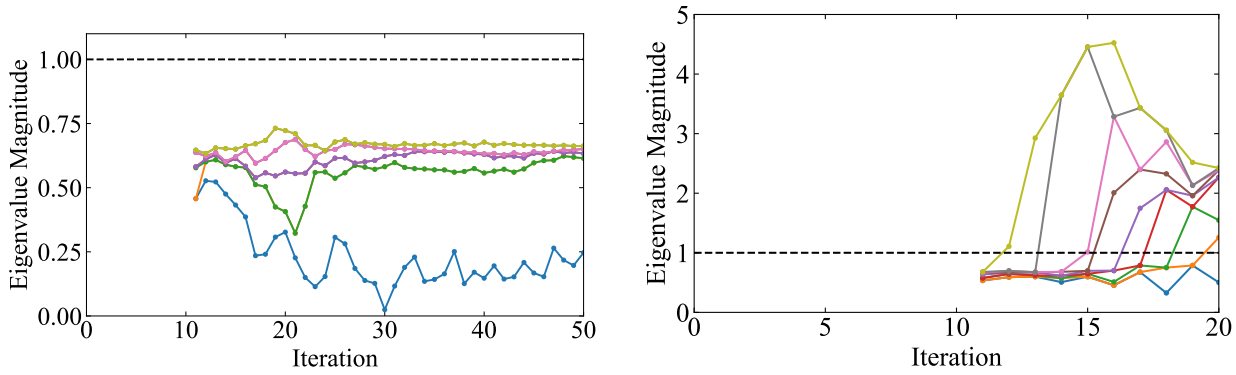


Fig. 2 Largest solution mode contours in a Burgers problem [3]

G. Dynamic Mode Decomposition

Dynamic mode decomposition offers several advantages over principal component analysis. It not only identifies unstable solution modes but also captures their temporal evolution behavior. DMD directly identifies dominant solution modes without the need for machine learning models. In a study by Zandsalimy and Ollivier-Gooch [4], it was demonstrated that DMD eigenvectors can effectively select problematic control volumes on the mesh for modification. This approach holds significant promise for mesh optimization as it eliminates the need for large-scale eigenanalysis, resulting in potential computational savings. DMD operates by computing dynamic modes from a multivariate time series dataset [14]. The eigenvalues of the Koopman matrix provide insights into the dominant solution modes and their temporal behavior. With a computational complexity of $O(nm^2)$, where m denotes the number of solution update vectors and n represents degrees of freedom, DMD offers an efficient mesh optimization approach.

DMD is performed on the latest 10 solution update vectors in two example Burgers problems. The magnitude of the eigenvalues of the Koopman matrix is depicted in Figure 3, where each color corresponds to the evolution of a different eigenvalue over time. In the case of a stable Burgers simulation, shown in Figure 3a, we observe that the magnitudes of the DMD eigenvalues consistently remain below 1.0. However, for an unstable Burgers problem, as depicted in Figure 3b, certain solution modes exhibit magnitudes greater than 1.0. Notably, the DMD eigenvectors associated with these unstable solution modes are localized to the problematic cells on the mesh, providing direct insights into their nature.



(a) Stable Solution **(b) Unstable Solution**

Fig. 3 Eigenvalue magnitude of the Koopman matrix for two example Burgers problems [4]

III. Methodology

The following section outlines the essential steps for a successful mesh optimization application in the current study.

A. Unstable Mode Identification

The process of mesh optimization begins by identifying unstable solution modes. Three methods have been chosen for this purpose: residual vector analysis [2], principal component analysis of solution update vectors [3], and dynamic mode decomposition of solution update vectors [4]. Each of these methods can be independently employed to achieve successful mesh optimization. To establish a benchmark for future research, we conduct tests utilizing all three methods. Once the solution modes are identified, we select and examine the problematic cells on the mesh. Anomaly detection models as discussed by [2, 3] are used for outlier detection and problematic control volume identification in the present study.

B. Approximate Eigenanalysis

First, we identify and extract the corresponding rows from the Jacobian matrix, relating to the selected cells and all the control volumes in their reconstruction stencil. Since numerically unstable modes tend to be localized in specific areas of the mesh, we can estimate the unstable eigenvalues and eigenvectors by performing eigenanalysis on these specific rows of the Jacobian. Essentially, we recognize that the numerically unstable eigenvectors have only a few

significantly non-zero values. Therefore, by starting with this eigenvector to find the corresponding eigenvalue, we effectively multiply the Jacobian matrix with a vector that predominantly consists of zero entries. As a result, we can ignore the less important rows and columns of the Jacobian matrix. We extract the selected rows and the corresponding columns from the Jacobian, projecting the large matrix on a much smaller space that is expected to capture the unstable eigenvalues and essential features of the corresponding eigenvectors.

To demonstrate the process of Jacobian matrix projection, we present an example scenario using the standard eigenvalue problem described in Equation 5. In this context, the Jacobian matrix, denoted by $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, represents the underlying system, while $\boldsymbol{x} \in \mathbb{C}^n$ denotes the eigenvector, and $\lambda \in \mathbb{C}$ corresponds to the eigenvalue. In this study, we propose a novel data projection method to approximate the Jacobian using a new matrix, denoted by $\boldsymbol{B} \in \mathbb{C}^{m \times m}$. Here, m represents the number of non-zero values in the eigenvector $\boldsymbol{x}$. To ensure successful results, prior knowledge of the non-zero pattern in the full eigenvector is necessary. Subsequently, we solve the approximate eigenproblem described in Equation 6, where $\boldsymbol{B}$ represents the reduced Jacobian, $\boldsymbol{z} \in \mathbb{C}^m$ denotes the reduced eigenvector, and the eigenvalue λ remains the same. Finally, the full eigenvector $\boldsymbol{x}$ can be readily reconstructed using the approximate $\boldsymbol{z}$ obtained through this method.

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \tag{5}$$

$$\boldsymbol{B}\boldsymbol{z} = \lambda z \tag{6}$$

An example full eigenvalue problem of Equation 5 is shown in Equation 7. Here, the full Jacobian matrix is presented along with the eigenvector $\boldsymbol{x}$, where zero values are indicated. Due to the presence of zero values in the eigenvector, certain rows and columns in $\boldsymbol{A}$ are irrelevant and can be zeroed out as demonstrated in Equation 8. For instance, the first row of $\boldsymbol{A}$ can be eliminated because the first entry on the right-hand side of the equation is zero. Similarly, the first column of $\boldsymbol{A}$ can be removed because the first value of the eigenvector $\boldsymbol{x}$ on the left-hand side is zero, and during the application of the Jacobian on this vector, the first column of the Jacobian is ignored. The resulting equation can be further reduced to the final form displayed in Equation 9, which is equivalent to Equation 6.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ x_3 \\ x_4 \\ 0 \\ x_6 \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} 0 \\ 0 \\ x_3 \\ x_4 \\ 0 \\ x_6 \\ 0 \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 & a_{36} & 0 \\ 0 & 0 & a_{43} & a_{44} & 0 & a_{46} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{63} & a_{64} & 0 & a_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ x_3 \\ x_4 \\ 0 \\ x_6 \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} 0 \\ 0 \\ x_3 \\ x_4 \\ 0 \\ x_6 \\ 0 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} a_{33} & a_{34} & a_{36} \\ a_{43} & a_{44} & a_{46} \\ a_{63} & a_{64} & a_{66} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \\ x_6 \end{bmatrix} = \lambda \begin{bmatrix} x_3 \\ x_4 \\ x_6 \end{bmatrix} \tag{9}$$

It is crucial to highlight that we are specifically focused on identifying unstable numerical eigenvectors that are localized to specific areas within the mesh. Consequently, these eigenvectors exhibit high sparsity, characterized by a small number of non-zero values. As a result, $n \gg m$ and the approximate eigenproblem will have considerably fewer degrees of freedom in comparison to the full Jacobian. Consequently, the presented optimization application achieves significantly improved computational efficiency.

C. Approximate Eigenvector

After calculating the eigenvectors of the small projected matrix, we can perform the reverse projection process. The zero values are reinserted into their corresponding indices (the rows that were omitted in the previous step) to obtain an approximation of the eigenvectors for the full Jacobian matrix. These approximate eigenvectors can then be utilized in the mesh optimization procedure presented by Zandsalimy and Ollivier-Gooch [1] and summarized in subsection II.C. It is expected that the approximate eigenvectors will align closely with the full eigenvectors of the Jacobian, pointing in similar directions.

To verify this, experiments are performed on different problems, and the resulting approximate eigenvectors are compared with the full eigenvectors. In the first test case, a model Burgers problem is solved on an unstructured mesh with 502 cells using the Crank-Nicolson time-stepping method. The residual vector is analyzed at each iteration of the solver and two outlier values are identified at iteration 12. The Jacobian rows corresponding to these outlier values and all the rows in their reconstruction stencil are selected to assemble the projected Jacobian. The approximate matrix in this case is 9×9 compared to the full Jacobian which is 502×502. The full Jacobian matrix at iteration 12 of the solver contains a single unstable mode $\lambda = 21.6647 + 0I$ and the approximate Jacobian contains a single unstable mode $\lambda = 21.5924 + 0I$ which have a relative difference of 0.33%. The right eigenvectors associated with these two unstable modes are presented and compared in Figure 4. The dot product of these two vectors is 0.99976, indicating a high level of similarity between them.

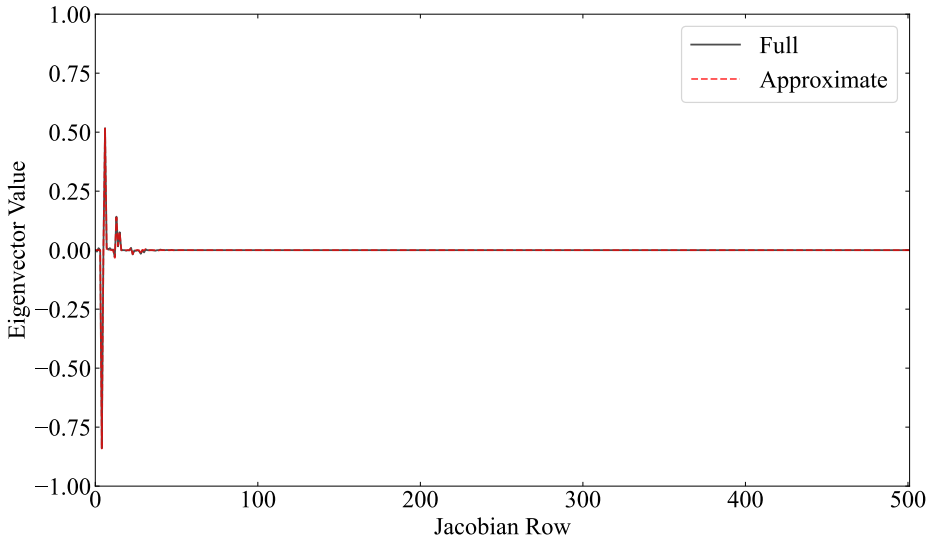


Fig. 4 The approximate and full eigenvectors in a Burgers problem

The next test is performed on a model Euler problem solved on an unstructured mesh with 566 cells using the Implicit Euler time-stepping method. PCA is performed on the solution update vectors and the largest solution mode is analyzed at each iteration for outliers. One outlier is identified at iteration 6 of the non-linear solver on the first solution variable (density). The Jacobian row corresponding to this outlier value and all the rows in its reconstruction stencil are selected to assemble the projected Jacobian. The approximate matrix in this case is 24×24 compared to the full Jacobian which is 2264×2264. The full Jacobian matrix at iteration 6 of the solver contains multiple unstable modes largest of which is $\lambda = 1.2792 + 0I$ and the largest unstable mode in the approximate Jacobian is $\lambda = 1.8903 + 0I$ which have a relative difference of 47.77%. The right eigenvectors associated with these two unstable modes are presented and compared in Figure 5. The dot product of these two vectors is 0.9685, indicating a high level of similarity.

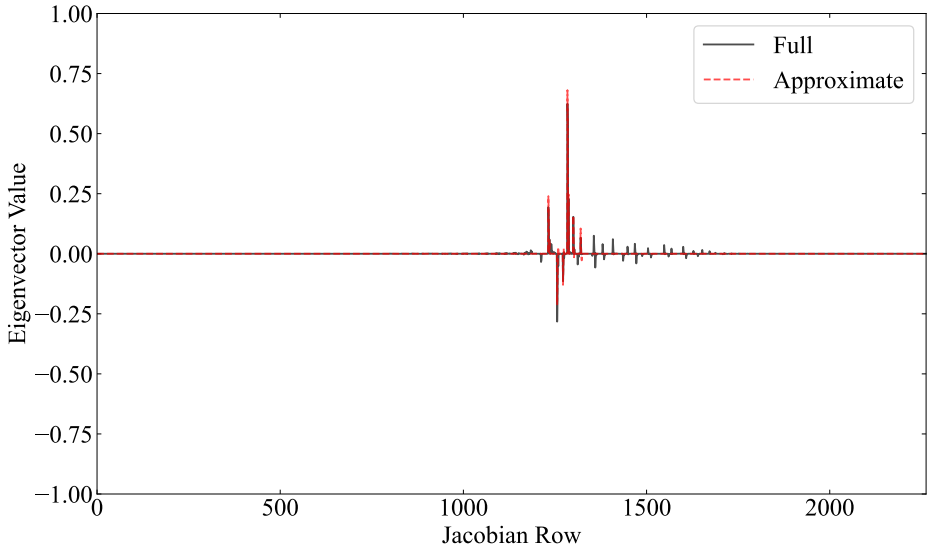


Fig. 5 The approximate and full eigenvectors in an Euler problem

Another Burgers problem is solved on an unstructured mesh with 4478 cells using the Crank-Nicolson time-stepping method. DMD is performed on the latest 10 solution update vectors and the DMD modes are analyzed. At iteration 45 the magnitude of a single DMD mode is larger than 0.9 which is stable but refers to a slow converging mode. The Jacobian rows corresponding to the largest values in the DMD eigenvector and all the rows in their reconstruction stencil are selected to assemble the projected Jacobian. The approximate matrix in this case is 15×15 compared to the full Jacobian which is 4478×4478. The full Jacobian matrix at iteration 45 of the solver contains no unstable modes. However, in this test a slow converging mode $\lambda = -2.6605 + 0I$ exists, and the largest mode in the approximate Jacobian is $\lambda = -2.7858 + 0I$ which have a relative difference of 4.71%. The right eigenvectors associated with these two unstable modes are presented and compared in Figure 6. The dot product of these two vectors is 0.71486. It is important to note that the approximate eigenvector successfully resolves the peaks in the exact eigenvector which is essential in the mesh optimization process.

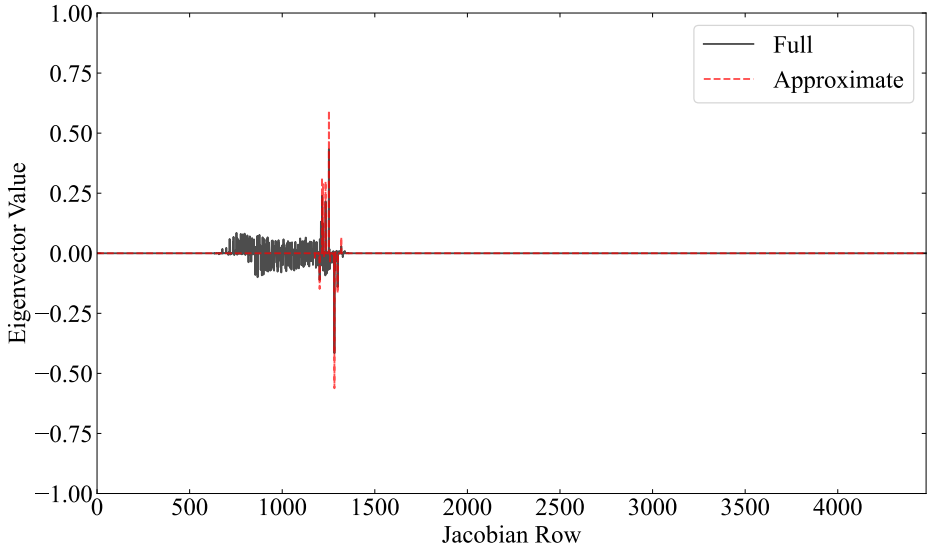


Fig. 6 The approximate and full eigenvectors in a Burgers problem

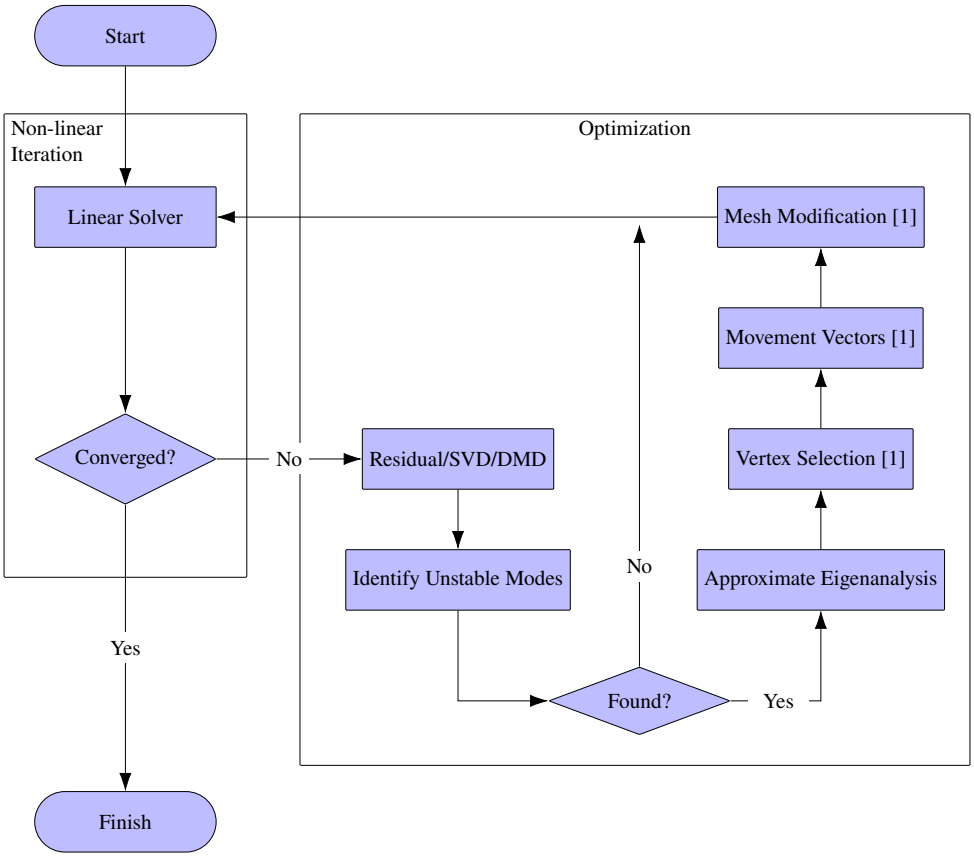


Fig. 7 Overview of the novel stabilization approach

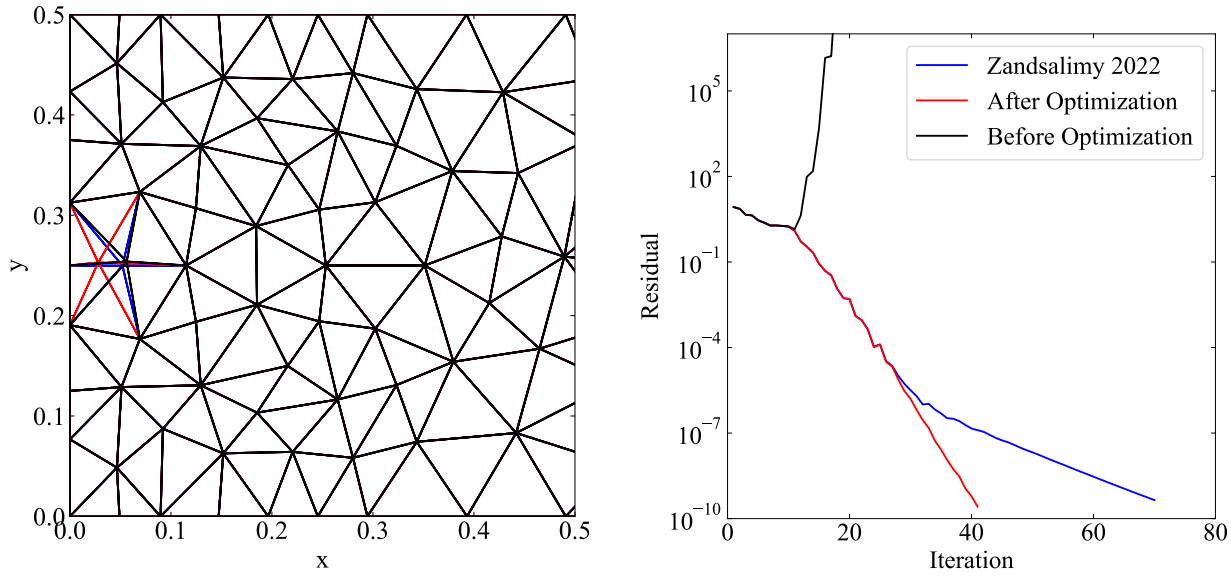
D. Algorithm Summary

Figure 7 illustrates the overall optimization algorithm introduced in this study. During each iteration of the non-linear solver, one of the three methods proposed for identifying unstable solution modes is employed. The first method presented in subsection II.E, involves analyzing the residual vector and utilizing an outlier detection module to identify unstable modes, which can be identified through anomalous values in the residual vector. The second method demonstrated in subsection II.F, utilizes principal component analysis on a collection of recent solution update vectors, and anomaly detection is performed on the PCA modes to identify unstable modes. The third method discussed in subsection II.G, employs dynamic mode decomposition on the solution vectors, allowing for direct identification of unstable modes without anomaly detection. Furthermore, the presented methodology can also be used to identify slow converging modes. If unstable modes (or modes of interest) are detected, the mesh optimization process proceeds to the next step. The approximate Jacobian is constructed using the selected rows and all the rows in their reconstruction stencil. Subsequently, eigenanalysis is performed on the small projected Jacobian, and the mesh optimization continues as described by Zandsalimy and Ollivier-Gooch [1].

IV. Results

This section presents the results obtained from the mesh optimization procedure described in this paper. The initial experiment involves solving a Burgers problem using the Crank-Nicolson time-stepping method on a mesh with 502 cells. This particular problem exhibits instability, characterized by a single unstable mode in its eigenspectrum. To identify any anomalous behavior, the residual vector is subjected to an outlier detection at each iteration of the non-linear solver. Several outliers are detected at iteration 12 of the solver. In Figure 4, we discussed the approximate eigenvector for this problem. The black residual history shown in Figure 8b represents the original unstable solution, which diverges after approximately 20 iterations. However, by employing the mesh optimization process using the approximate eigenvector, we achieve a stable solution as indicated by the red residual history. The original and opti-

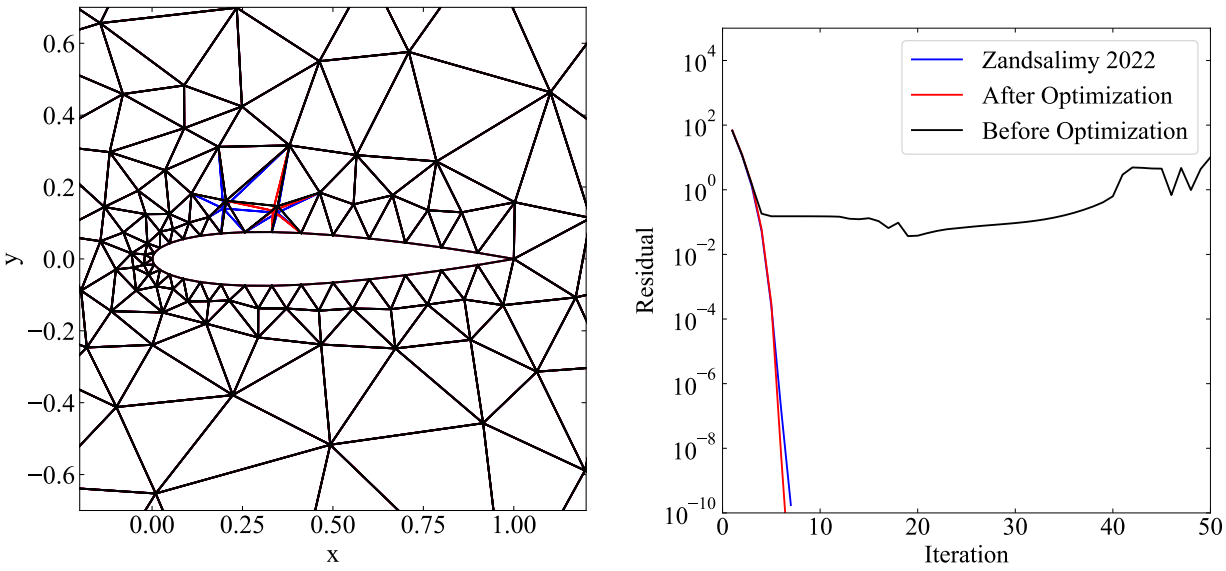
mized meshes are depicted in Figure 8a, highlighting the modification of a single vertex. In comparison to the mesh optimization results obtained by Zandsalimy and Ollivier-Gooch [1], our novel approach achieves a Frobenius residual norm of 10^{-10} in 45% fewer iterations.



(a) Numerical mesh **(b) Residual history**

Fig. 8 Mesh optimization in a Burgers problem with 502 cells

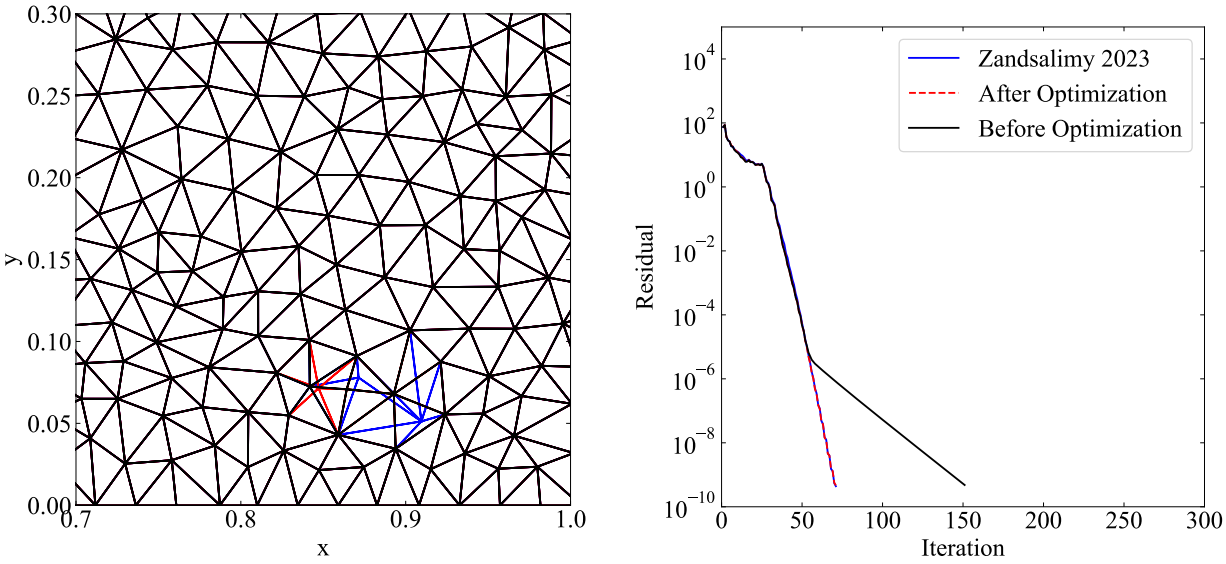
In the next experiment, an Euler problem is solved using the Implicit Euler time-stepping method on a mesh with 566 cells. In this test, the pseudo-transient continuation [15] method is used to systematically change the time-step size for faster convergence in the case of stable problems and to prevent solution blow-up in the case of unstable solutions. SVD is performed on the solution update vectors and the largest solution mode is analyzed for outlier values. The first anomalies are detected at iteration 6 of the solver. The approximate eigenvector for this problem was discussed in Figure 5. In Figure 9b, the black line represents the original unstable problem that fails to converge, while the red line presents the residual history of the stable solution obtained through mesh optimization using the approximate eigenvector. Furthermore, Figure 9a illustrates the modification of a single vertex in the mesh during the process. In comparison to the results by Zandsalimy and Ollivier-Gooch [1], we obtain a similar convergence rate after the optimization process.



(a) Numerical mesh **(b) Residual history**

Fig. 9 Mesh optimization in an Euler problem with 566 cells

The next experiment is performed on a Burgers problem using the Crank-Nicolson time-stepping on a mesh with 4478 cells. This problem is originally stable but there is a single slow-converging mode that hinders the convergence rate substantially. DMD is performed on the latest 10 solution update vectors in the simulation and at iteration 45 a DMD mode is identified to have a magnitude larger than 0.9. In this case, the DMD eigenvector associated with this mode is used to find the cells of interest on the mesh. The projected eigenvector for this problem was discussed in Figure 6. As seen in Figure 10b the original problem is stable and converges at around 150 iterations. The process of mesh optimization using the approximate eigenvector results in a stable solution with the residual history presented in red converging in 50% fewer iterations. Figure 10a shows the location of a single vertex that was modified in the process. In comparison to the mesh optimization presented by Zandsalimy and Ollivier-Gooch [3], we obtain a similar convergence rate.



(a) Numerical mesh **(b) Residual history**

Fig. 10 Mesh optimization in a Burgers problem with 4478 cells

The current study introduces an effective approach to eliminate the need for eigenanalysis of the Jacobian matrix in the mesh optimization process for improved stability of finite volume methods. This end is achieved by identifying the most important rows in the Jacobian matrix and assembling them into an approximate Jacobian with significantly fewer degrees of freedom. Consequently, the computational cost associated with solving the eigenvalue problem on the projected Jacobian is negligible compared to analyzing the full eigenspectrum. To facilitate this process, new computational modules are required to identify problematic areas in the mesh and the corresponding rows in the Jacobian matrix. Notably, the presented optimization algorithm demonstrates favorable computational savings through anomaly detection on the residual vector and PCA solution modes. Additionally, dynamic mode decomposition of the solution update vectors offers a computationally inexpensive method for identifying unstable modes, eliminating the need for anomaly detection. The results obtained show substantial computational savings compared to the method employed by Zandsalimy and Ollivier-Gooch [1].

Figure 11 shows the computational time required for the mesh optimization process in an Euler problem as a function of the number of degrees of freedom. The solid lines represent the run-time of different modules in the current study, while the dashed lines represent the results obtained by Zandsalimy and Ollivier-Gooch [1]. The modules of Movement Vector, Flow Solver, and Jacobian computations are consistent across both studies. However, in the current study, the eigenanalysis module has been replaced with either the Anomaly Detection (on the residual vector or the PCA solution modes) or DMD Analysis modules.

It is noteworthy that the Projected Jacobian Eigenanalysis module exhibits almost a constant run-time regardless of the degrees of freedom, as it is independent of the mesh size. The solid black line represents the overall optimization time in the current study, which is approximately 2.5 times lower than the dashed black line from Zandsalimy and Ollivier-Gooch [1], indicating a significant reduction in the time required for the mesh optimization process.

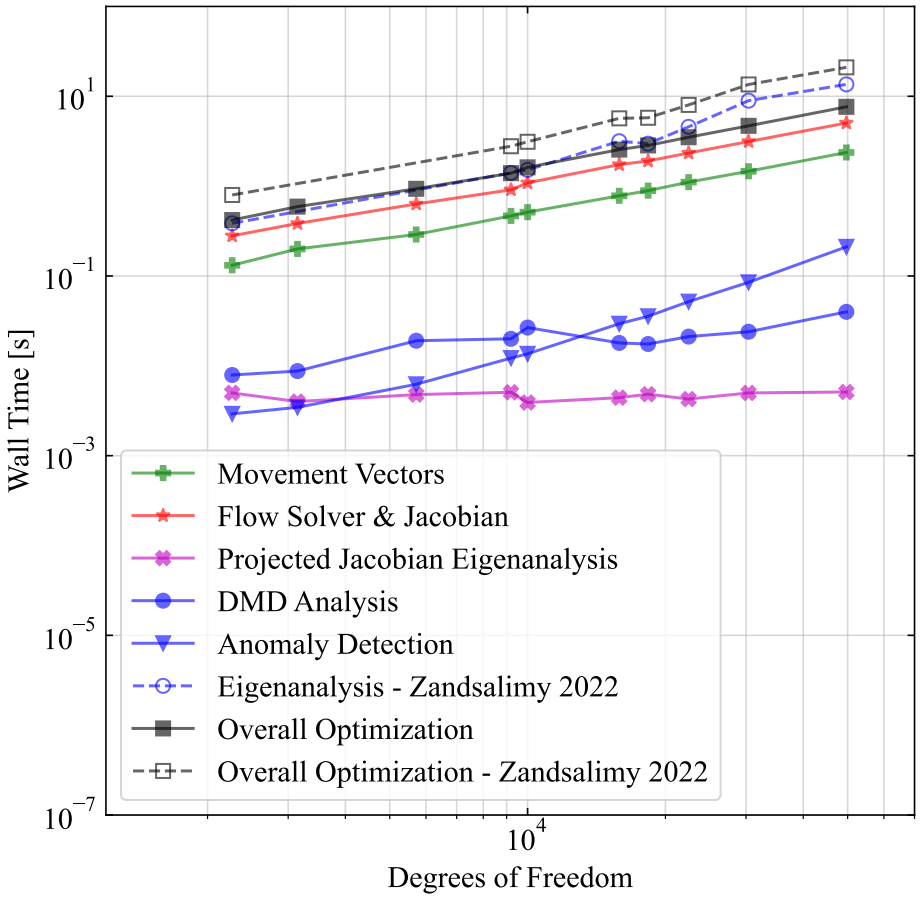


Fig. 11 The optimization run-time of the Euler problem

V. Conclusion

The current study presents a novel method for mesh optimization of finite volume simulations. By targeting local areas in the mesh that contribute to numerical instabilities, the study extracts the corresponding rows and columns from the full Jacobian matrix, forming a much smaller approximate Jacobian. It is demonstrated that the projected matrix accurately captures the unstable solution modes when the problematic cells in the mesh are correctly identified. The key idea behind this technique is that unstable numerical solution modes tend to be highly localized in specific areas of the solution domain. The eigenvectors associated with these modes have only a few non-zero values concentrated in these areas of interest. By applying the Jacobian matrix to such a vector, we obtain a vector with the same non-zero pattern. This allows us to eliminate the rows and columns corresponding to zero entries in the eigenvector, resulting in the same eigenvalue problem.

Eigenanalysis can then be performed on the projected Jacobian, which has significantly fewer degrees of freedom compared to the full Jacobian matrix. In the subsequent step, full eigenvectors are formed based on the approximate eigenvectors, maintaining the same non-zero pattern. The mesh optimization approach can then proceed as usual, where the correct vertices on the mesh are identified, movement vectors are calculated through eigenvalue gradient calculation, and mesh modification is performed. The results presented in this paper demonstrate the feasibility of the proposed approach for mesh optimization in unstructured mesh finite volume simulations. Notably, this method eliminates the need for eigenanalysis of the full Jacobian matrix, which was a significant computational bottleneck in previous studies. Moreover, the current approach enables fully automatic optimization without requiring human intervention. Additionally, the formation of a projected approximate Jacobian is not limited to the implementation described here; other fields working with the reduced dimensionality of large matrices that possess knowledge of the eigenvector non-zero pattern can also benefit from this method.

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] 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.

[2] Zandsalimy, M., and Ollivier-Gooch, C., "Unsupervised Residual Vector Analysis for Mesh Optimization," *AIAA SciTech 2023 Forum*, 2023, p. 0833. https://doi.org/10.2514/6.2023-0833.

[3] Zandsalimy, M., and Ollivier-Gooch, C., "Residual Vector And Solution Mode Analysis Using Semi-Supervised Machine Learning For Mesh Modification And CFD Stability Improvement," *In Preparation, Journal of Computational Physics*, ????

[4] Zandsalimy, M., and Ollivier-Gooch, C., "Dynamic Mode Decomposition For Improved Numerical Stability of Finite Volume Simulations," *Submitted to AIAA SciTech 2024 Forum*, 2024.

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

[6] 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.

[7] Ollivier-Gooch, C., "GRUMMP version 0.7.0 user's guide," *Department of Mechanical Engineering, University of British Columbia*, 2016.

[8] 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.

[9] Khalil, H. K., "Lyapunov Stability," *Control Systems, Robotics and Automation – Volume XII: Nonlinear, Distributed, and Time Delay Systems-I*, 2009, p. 115.

[10] 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.

[11] 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. 351–362. https://doi.org/10.1145/1089014.1089019.

[12] 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.

[13] Vasudevan, V., and Ramakrishna, M., "A Hierarchical Singular Value Decomposition Algorithm for Low Rank Matrices," *CoRR*, Vol. abs/1710.02812, 2017.

[14] Schmid, P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, Vol. 656, 2010, p. 5–28. https://doi.org/10.1017/S0022112010001217.

[15] Ceze, M., and Fidkowski, K. J., "Constrained pseudo-transient continuation," *International Journal for Numerical Methods in Engineering*, Vol. 102, No. 11, 2015, pp. 1683–1703. https://doi.org/https://doi.org/10.1002/nme.4858.