A clustering algorithm for reduced order modeling of shock waves

2015

Tina R. White
Department of Mechanical Engineering
Stanford University
Stanford, CA 94305
crwhite@stanford.edu

Background

Imagine yourself looking out the window on a commercial flight at the aircraft engine outside. For four years, I worked at Honeywell Aerospace as product design engineer in a research team investigating flutter and forced response issues in turbomachinery for aircraft engines. Flutter is an aeroelastic issue that can cause near-instantaneous and catastrophic part failure. It results from the unsteady interaction of a structure prone to vibration with an airflow. Blade sections can tear away from a rotating component and destroy an engine. Figure 1 shows British Midland Flight 92, a crash precipitated by an aeroelastic issue in 1990. Two blades of the fan are missing - torn away during flight.

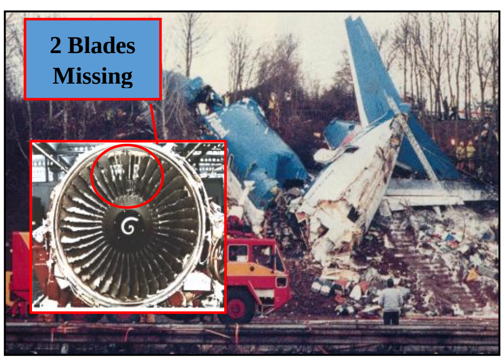

Figure 1: British Midland Flight 92 crash due to aeroelastic issue

A commercial jet cruises at about Mach 0.7 (7/10ths the speed of sound). However, due to the rotation of the gas turbine engine fan and the increasing velocity of the air over the surface of the fan blades, a portion of the airflow over the fan typically experiences supersonic flow conditions and a weak shock wave. Such a flow field is called transonic. Phenomena like transonic flutter in turbomachinery are predicted and mitigated using computational fluid dynamics (CFD), specifically fluid-structure interaction modeling.

A CFD analysis is spatially local and involves computing a differential equation on a large number - often millions - of discrete points throughout a computational domain. Although CFD analyses are a vast improvement over wind tunnel or rig testing alone, they are computationally expensive, which is limiting to the engineer trying to rapidly iterate and optimize a product. Optimizing a fan design while mitigating flutter risk can take months because a full aeroelastic analysis typically takes 5 days even with modern computational tools. While methods exist to increase the speed of a CFD analysis, no commercially available real-time methods accurately predict the aeroelastic behavior of turbomachinery in the region of transonic stall flutter.

Introduction

In reduced order models (ROMs), a dataset already exists with numerous full CFD simulations. From that dataset, the intention of ROMs is to learn how to make good predictions in the future by creating a simple model based on what is already known. The goal of reduced order modelling is to simplify the complex system to run in real time, enabling significant CPU time reductions.

In the first stage of reduced order modeling, the solution vectors (snapshots) are split into sub-regions defined by local basis vectors using k-means clustering, illustrated in Figure 2 below. These clusters are formed based on thousands of snapshots from full CFD simulations. The k-means algorithm identifies where the solution exhibits significantly different features so that these vectors can be assigned to sub-regions. A singular value decomposition (SVD) is computed to reduce the size of the matrices and create a local reduced order basis (ROB). Using these ROBs, a simulation is modeled as a combination of a much smaller number of vectors than the number in the original cluster of snapshots. In the second stage of the method, a residual minimization problem completes the analysis and can compute a solution in real time.

This methodology has been applied at Stanford by the Farhat Research Group [11] and applied on a 1D Burger's Equation and an acceleration study of a transport aircraft. The potential of the method has already been demonstrated for various CFD and fluid-structure interaction problems including aeroelastic problems [4] and problems involving moving shocks. [3]

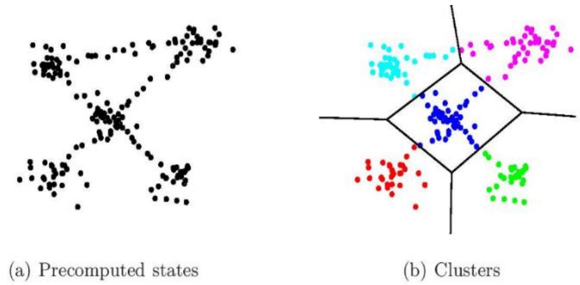

Figure 2: Clustering diagram

Methods

The data from one CFD simulation is placed into one vector called a snapshot. A matrix A is constructed with all pre-computed snapshots along the columns of the matrix. I propose a new method for clustering the data according to Procedure 1.

Procedure 1: Row-Column Clustering and Pre-processing Steps to Create a ROB:

1. Construct matrix A with pre-computed snapshot vectors along columns
2. Apply k-means clustering to columns of A saving column cluster centers and indices
3. Apply k-means clustering to the rows of each submatrix formed by the column clusters of A
4. Perform SVD on submatrices and save the desired number of right singular vectors

The procedure only applies to matrices with single variables saved (velocity for example) but can be expanded to matrices with multiple CFD variables.

Additionally, it's not necessary to run a reduced order model to determine whether the ROB constructed via the procedure is accurate. A representative vector can be chosen from the original snapshot and projected onto the ROB built by the procedure. By plotting this projection and/or calculating the RMS error between the original vector and its projection into the ROB, the quality of the reduced order basis can be ascertained.

Procedure 2: Additional Steps for Projection of a Representative Snapshot Vector V onto its ROB:

5. Use column cluster indices to determine which column cluster contains the vector V
6. Use column cluster indices to reorder the vector of V to correspond to its submatrix
7. Use row indices to split vector along each row cluster of A
8. Perform projection within ROB subspace
9. Use column cluster indices to reorder the vector V back into original matrix order
10. Plot or calculate RMS error between vector and its projection

The implementation of k-means clustering was done in python using cluster.k_means contained in the sklearn library. [8] K-means clustering was implemented twice on the matrix, once on the columns, followed by the rows. Because all terms of both the rows and columns of the original matrix are rearranged to construct the submatrices, the method involves considerable bookkeeping, but is otherwise simply an application of k-means twice. It could be categorized as a form of biclustering.

To emphasize, the end goal of this procedure is not to analyze the dataset. The procedure is a starting point. This pre-processes the data to create an improved reduced order basis that models a complex CFD simulation, which you can use to run previously computationally expensive simulations in real time - for the purpose of either better understanding of the parameters or optimization.

Related Work

Biclustering and co-clustering are well known procedures for a variety of machine learning applications, most notably gene expression. [9] Other work has combined k-means clustering with linear discriminant analysis. [2] However, biclustering algorithms do not appear in literature that output center vectors like those output by the row-column k-means procedure described in the methods section. And these mean vectors are required to complete the reduced order model computation.

Also, different forms of domain decomposition are common in fluid mechanics. The domain is often split along mesh lines evenly for parallel computing. Typically the split for a given domain is determined manually by the engineer considering the particulars of a given design. [6] In this case, what makes it possible to compute sub-domains more generally and independent of mesh is that this procedure is not computing full CFD solutions along mesh lines. The results of CFD simulations are givens. ROMs are computed "off the grid." Therefore, there's more freedom in how to create sub-domains, making the application of machine learning algorithms for domain decomposition for ROMs full of possibilities.

In sum, k-means has been used to cluster subdomains [10] and it has been used to cluster snapshots [1], but little or no research has been done to combine the benefits of the two.

Importantly, one problem encountered consistently in reduced order modeling for CFD is that shock waves are not well modeled by ROBs. A shock wave occurs when a wave moves faster than the local speed of sound. It is characterized by an abrupt, nearly discontinuous change in pressure. Therefore, SVD implementations suffer from unphysical Gibbs' oscillations near the shock wave discontinuity. [5] Capturing the discontinuity and reducing the Gibbs' oscillations is difficult without a full order model in the region of the shock. Some [7] have applied domain decomposition, but they resort to implementing a full order model in the region of the shock in order to resolve it.

While initially the intention of this project was to research and implement adaptive clustering algorithms, I had an interest in clustering along the rows of the CFD domain. When the susceptibility of ROMs to Gibbs' oscillations became apparent, the goal of the project was changed. The row-column clustering procedure was conceived as a form of domain decomposition in order to improve the modeling the shock wave discontinuity. Figures 3 and 4 demonstrate how the shock wave splits the solution into distinct domains of supersonic and subsonic flow.

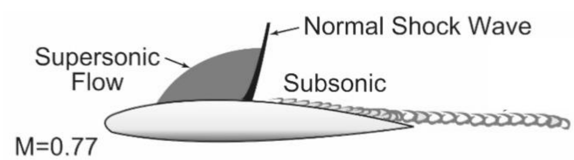

Figure 3: Transonic flow region around an airfoil

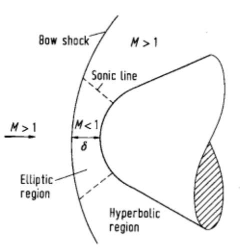

Figure 4: Shock wave preceding a blunt leading edge in supersonic flow

Dataset 1 and Features

The row-column clustering procedure was applied to two test cases: a simple 1D Burger's equation and an oscillating airfoil case.

For a full three-dimensional CFD solution, the data could contain millions of points. At each point in the mesh, many variables are saved. If 1000 snapshots are generated, each containing the variables for all points in the mesh for an analysis, this creates a very large matrix to analyze. Therefore, the first step is to find a simplified problem that can be used for testing, but implement it in a way that allows it to be scaled. The solution of a one-dimensional Burgers' equation is such a case.

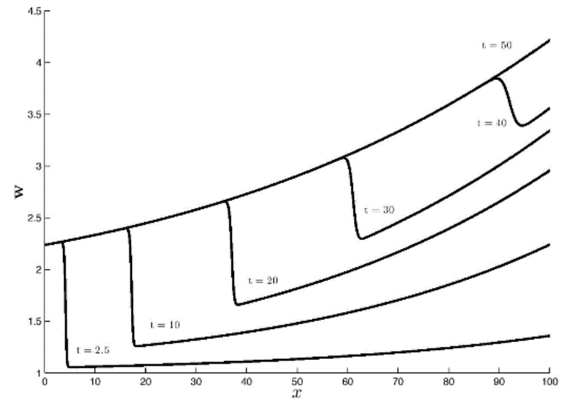

Figure 5: Burger's equation solutions for test case

The one-dimensional Burger's equation test case is a fluids application of an initial-boundary-value problem (IBVP) that models the movement of a shock wave. Figure 5 shows the velocity *w* vs. location *x* as a shock wave moves from left to right in time across the domain. Working with Matt Zahr in the Farhat research group, I obtained a Burger's equation dataset and a python code that runs the one-dimension Burger's problem and implements column clustering using the cluster.k_means function from the python library sklearn. There are 1000 points in the mesh, and one variable, velocity, is saved per coordinate, x. I used a set of 1000 snapshots. Because the problem is much simpler than a full Navier-Stokes simulation, 1000's of snapshots can be generated in a few minutes. Each column represents a different time or different initial or boundary conditions.

Figures 6 and 7 show the results for a simple column-only clustering procedure that splits the data into two clusters. Figure 6 shows the two cluster centers generated for the snapshots. A look back at Figure 3 reveals that the clustering has captured two patterns that characterize the snapshots in time. Figure 7 shows the results calculated for a representative snapshot and the vector projection of that snapshot onto the first basis.

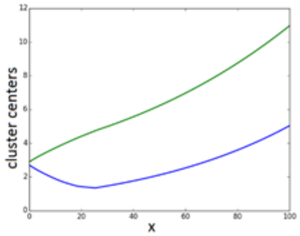

Figure 6: Cluster Centers

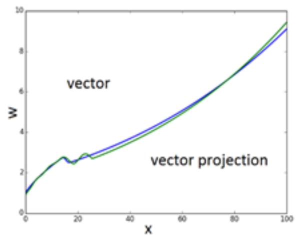

Figure 7: Solution w and its vector projection

Dataset 2 and Features

The second test case for implementing the row-column clustering code was a 2D unsteady CFD solution of an airfoil twisting and flexing in time. A 2D airfoil simulation models a slice of a full 3D CFD simulation of, for example, a wing or turbomachinery blade. The twisting and flexing of the airfoil in time is analogous to the twisting and flexing of a vibrating turbomachinery blade. Figure 8 shows the response of a fluid to the unsteady motion of a blade over time. Figure 9 shows a representative snapshot of this dataset.

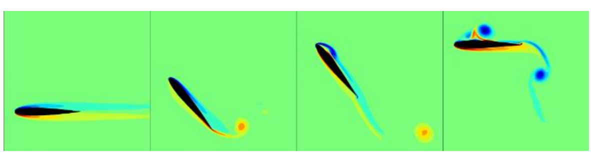

Figure 8: Flow evolution over time as airfoil blade section twists and flexes

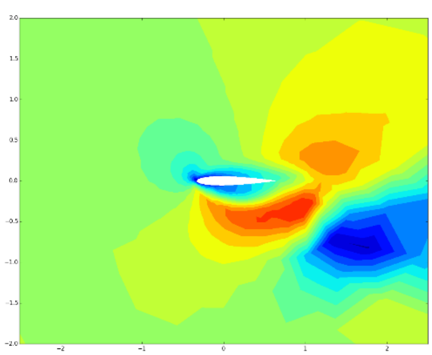

Figure 9: A single 2D CFD simulation snapshot

Results

The implementation of row-column clustering was found to substantially improve the quality of the ROBs for the Burger's equation in the proximity of the shock wave discontinuity. Figure 10 shows the damping of the Gibbs' oscillations for the case of a 10 column clusters on the data with an increasing a number of row clusters from 1 (column-only clustering case) to 10 row clusters.

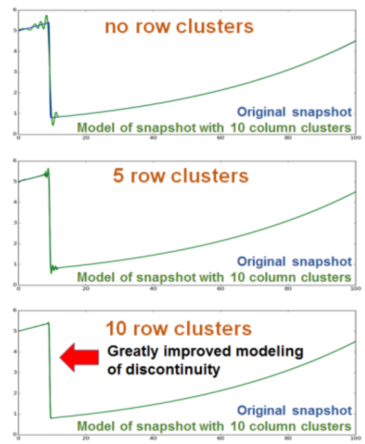

Figure 10: Reduction of Gibbs' oscillations due to row-column clustering

Additionally, because these oscillations are the primary source of the error in the ROB (represented by the RMS difference between a representative vector and its projection onto the ROB), the error is greatly reduced by applying row-column clustering. Figure 11 illustrates the reduction in RMS error as the number of column clusters are increased, given a fixed number of row clusters. Figure 12 illustrates the reduction in RMS error as the number of row clusters is increased, given a fixed number of column clusters. Row-column clustering is clearly an improvement for both conditions.

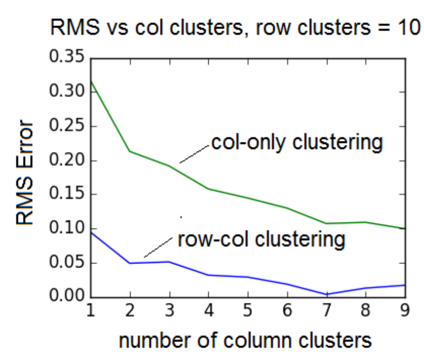

Figure 11: RMS error reduction with increasing column clusters given fixed number of row clusters

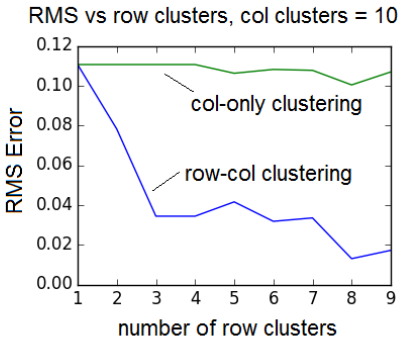

Figure 12: RMS error reduction with increasing row clusters given fixed number of column clusters

Additionally, row column clustering was run on the second dataset. Because of the 2D nature of this type of simulation, the row clustering for a given cluster was possible to visualize. Figure 13 visualizes the row clusters, colored by number, of the particular column cluster for a representative vector. This example in particular may show a case of overfitting, given the multiple clusters used to model a single vortex, yet it also demonstrates that the clusters are partially aligned with flow structures. In this plot, for example, typical flow structures like the free stream, vortices, boundary layer, and stagnation point are captured by the clustering.

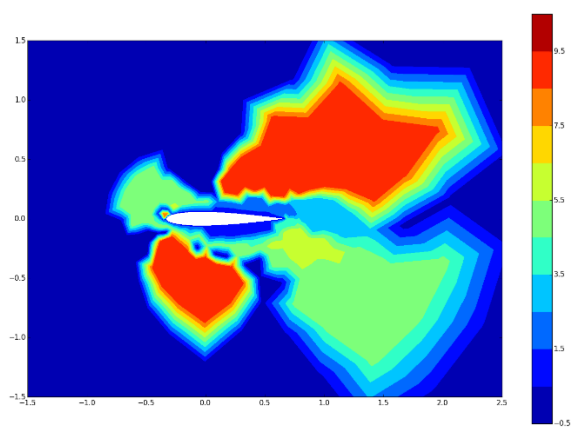

Figure 13: Row clusters colored by number

Finally, a representative snapshot and its projection onto its row-column clustered ROBs were visualized. Figure 14 shows the representative snapshot, its column-only projection and its row-column projection. While there is only slight visual improvement in the modeling, this is to be expected because the dataset does not contain any shock waves. Future work will involve obtaining additional datasets and testing row-column clustering on 2D and 3D datasets with shock waves. Most likely, the benefit was slight also because multiple variables are included in this snapshot matrix, and the procedure has not yet been expanded to accommodate this.

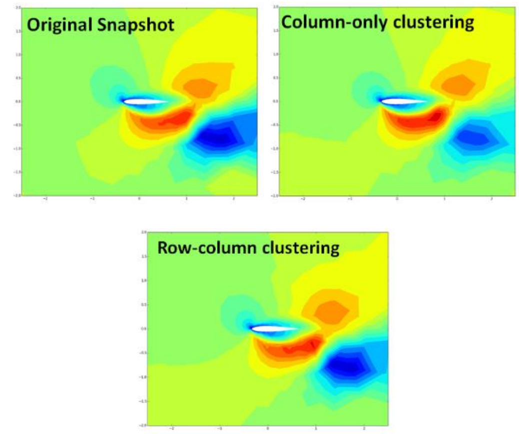

Figure 14: A representative snapshot, its column-only projection, and its row-column projection

Conclusions

The main achievement of this project has been to establish a proof of concept for future work in applying machine learning algorithms like k-means to both snapshot clustering and domain decomposition for reduced order modeling of computational fluid dynamics results. The ability of the row-column clustering method to capture shock waves is a novel contribution. While results are currently limited to the simple 1D Burger's equation case, the method can be implemented easily on far more complex models.

The potential implications are many. CFD for transonic and supersonic flow is used in the design process of many aerospace products, for example, aircraft, engines, and spacecraft. This improvement can have wade ranging impact across the industry.

Future Work

Next quarter I will continue the project as an independent study course with the Farhat Research Group, and potentially submit the work to NIPS. K-means is a very rough form of domain decomposition for CFD solutions, so there is much room for improvement. One limitation is that this implementation of k-means row-column clustering does not take into account the locations of the features in actual space. One proposed improvement is to weight k-means according to the spatial locations of the points.

Also, the row-column clustering procedure must be expanded to accommodate multiple variable types in the original snapshot matrix.

References

1. Amsallem, David, Matthew J. Zahr, and Charbel Farhat. "Nonlinear model order reduction based on local reduced-order bases." *International Journal for Numerical Methods in Engineering* 92.10 (2012): 891-916.

2. Ding, Chris, and Tao Li. "Adaptive dimension reduction using discriminant analysis and k-means clustering." *Proceedings of the 24th international conference on Machine learning*. ACM, 2007.

3. Farhat, Charbel, Michael Lesoinne, and P. Le Tallec. "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity." *Computer methods in applied mechanics and engineering* 157.1 (1998): 95-114.

4. Geuzaine, Philippe, et al. "Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions." *AIAA journal* 41.3 (2003): 363-371.

5. Gottlieb, David, and Chi-Wang Shu. "On the Gibbs phenomenon and its resolution." *SIAM review* 39.4 (1997): 644-668.

6. Gropp, William D., and David E. Keyes. "Domain decomposition methods in computational fluid dynamics." *International journal for numerical methods in fluids* 14.2 (1992): 147-165.

7. Lucia, David J., Paul I. King, and Philip S. Beran. "Domain decomposition for reduced-order modeling of a flow with moving shocks." *AIAA journal* 40.11 (2002): 2360-2362.

8. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12 (2011): 2825-2830.

9. Prelić, Amela, et al. "A systematic comparison and evaluation of biclustering methods for gene expression data." *Bioinformatics* 22.9 (2006): 1122-1129.

10. Smith, Robert E. *Computational fluids domain reduction to a simplified fluid network*. No. TARDEC-22878. ARMY TANK AUTOMOTIVE RESEARCH DEVELOPMENT AND ENGINEERING CENTER WARREN MI, 2012.

11. Washabaugh, Kyle, et al. "Nonlinear model reduction for CFD problems using local reduced-order bases." *42nd AIAA Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, AIAA Paper*. Vol. 2686. 2012.