REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
08-05-2015	Final Report	6-Apr-2012 - 5-Apr-2015

4. TITLE AND SUBTITLE

Final Report: A FSI Enabled Practical Rotorcraft Flow Simulator for Morphing Blade Innovation

5a. CONTRACT NUMBER
W911NF-12-1-0147

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS

Chen, Chung-Lung, Liu, Gui-Rong

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

University of Missouri - Columbia
310 Jesse Hall

Columbia, MO 65211 -1230

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
62061-MA.15

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

We have developed a fluid-structure interaction (FSI) enabled solver by coupling the Gradient smooth method (GSM) CFD solver and CSD solvers. The research includes three parts: 1) development and improvement of the GSM-CFD solver; 2) development of the FSI coupling code; 3) investigation of transonic shock/boundary-layer flows with a morphing blade.

1) For the GSM-CFD solver, a new matrix-based algorithm and its corresponding data structure have been devised to improve the computational efficiency and compactness of GSM. A new integration scheme has been proposed to

15. SUBJECT TERMS

gradient smoothing method, flow-structure Interaction, flow control, rotor flow and heat transfer enhancement

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Chung-Lung Chen
UU	UU	UU	UU		19b. TELEPHONE NUMBER 573-882-9037

Report Title

Final Report: A FSI Enabled Practical Rotorcraft Flow Simulator for Morphing Blade Innovation

ABSTRACT

We have developed a fluid-structure interaction (FSI) enabled solver by coupling the Gradient smooth method (GSM) CFD solver and CSD solvers. The research includes three parts: 1) development and improvement of the GSM-CFD solver; 2) development of the FSI coupling code; 3) investigation of transonic shock/boundary-layer flows with a morphing blade.
1) For the GSM-CFD solver, a new matrix-based algorithm and its corresponding data structure have been devised to improve the computational efficiency and compactness of GSM. A new integration scheme has been proposed to improve the accuracy of GSM on distorted and hybrid meshes.
2) In the effort of fluid-structure coupling, the codes for FSI pair searching and shape function interpolation have been developed for the data exchange between the fluid part and the structural part. Extensive numerical examples have been provided to show the effectiveness of the proposed methods.
3) In the cases of the transonic shock/boundary layer flow control, we have demonstrated that the morphing layer can effectively weaken the transonic shock at the advancing azimuths. The characteristic of noise could be modified with a morphing layer.
During the execution of this program, five journal papers were published, one has been accepted, and two conference papers have been presented.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
05/08/2015	14.00	Jiangyao Yao, Tao Lin, Gui-Rong Liu, Chung-Lung Chen. A comparative study of time-marching schemes for fluid structure interactions, Vibroengineering PROCEDIA, (11 2014): 170. doi:
07/15/2013	1.00	JIANYAO YAO, G. R. LIU, DONG QIAN, CHUNG-LUNG CHEN, GEORGE X. XU. A MOVING-MESH GRADIENT SMOOTHING METHOD FOR COMPRESSIBLE CFD PROBLEMS, Mathematical Models and Methods in Applied Sciences, (02 2013): 0. doi: 10.1142/S0218202513400046
08/27/2014	6.00	Jianyao Yao, G. R. Liu. A matrix-form GSM–CFD solver for incompressible fluids and its application to hemodynamics, Computational Mechanics, (02 2014): 0. doi: 10.1007/s00466-014-0990-8
08/27/2014	7.00	Junxiang Shi, Tiancheng Xu, Steven R. Schafer, Chung-Lung Chen. Adaptive Control of Shock Waves with a Passively Morphing Layer for Rotating Blades, Journal of Aerospace Engineering, (10 2013): 0. doi: 10.1061/(ASCE)AS.1943-5525.0000394
08/27/2014	8.00	Junxiang Shi, Jingwen Hu, Steven R. Schafer, Chung-Lung (C.L.) Chen. Numerical study of heat transfer enhancement of channel via vortex-induced vibration, Applied Thermal Engineering, (09 2014): 0. doi: 10.1016/j.applthermaleng.2014.05.096
TOTAL:		**5**

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper
08/27/2014	10.00 Steven R. Schafer, Chung-Lung (C.L.) Chen, Junxiang Shi. Vortex-induced vibration for heat transfer enhancement, ASME 2014 International Mechanical Engineering Congress & Exposition. 14-NOV-14, . : ,
08/29/2014	12.00 Junxiang Shi, Jingwen Hu, Steven R. Schafer, Chung-Lung (C.L.) Chen. Heat transfer enhancement of channel flow via vortex-induced vibration of flexible cylinder, ASME 2014 Fluids Engineering Summer Meeting (FEDSM2014). 03-AUG-14, . : ,

TOTAL: **2**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
05/08/2015	13.00	Jianyao Yao, Tao Lin, G. R. Liu, C. L. Chen. An adaptive GSM-CFD solver and its application to shock-wave boundary layer interaction, International Journal of Numerical Methods for Heat and Fluid Flow, 2014 (09 2014)
07/15/2013	2.00	Jianyao Yao, G.R. Liu, C.L. Chen. Adaptive GSM-CFD Techniques for Shock Wave Boundary Layer Interactions of Compressible Turbulent Flows, International Journal of Numerical Methods in Fluids (06 2013)
07/15/2013	4.00	Junxiang Shi, Tiancheng Xu, Steven R. Schafer, Chung-Lung (C.L.) Chen. Adaptive Control of Shock Wave with Passively Morphing Layer for a Rotating Blade, Journal of Aerospace Engineering (06 2013)
07/15/2013	3.00	Junxiang Shi, Steven R. Schafer, Chung-Lung Chen. Investigation of adaptive transonic flow control method by a partitioned fluid-structure interaction solver, Journal of Aircraft (06 2013)
07/16/2013	5.00	Junxiang Shi, Schafer, Steven R., Chung-Lung (C.L.) Chen. Adaptive Shock/Boundary Layer Control with Passively Morphing Layer for Rotor Flows, Aerosol Science and Technology (04 2013)
08/27/2014	11.00	Junxiang Shi, Schafer, Steven R., Chung-Lung (C.L.) Chen. Weakening of Shock Waves with Rotating Adaptive Blades, Journal of Aircraft (08 2014)
TOTAL:		**6**

Number of Manuscripts:

Books

Received		Book
TOTAL:		

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Lin, Tao	0.50	
Zhang, Bo	0.25	
FTE Equivalent:	**0.75**	
Total Number:	**2**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Shi, Junxiang	0.30
Yao, Jianyao	0.30
FTE Equivalent:	**0.60**
Total Number:	**2**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Chen, Chung-Lung	0.07	
Liu, Gui-Rong	0.09	
FTE Equivalent:	**0.16**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Schafer, Steven	0.25	BS
Lipira, Zachary	0.10	BS
FTE Equivalent:	**0.35**	
Total Number:	**2**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	2.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	2.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	2.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):.....	1.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	1.00

Names of Personnel receiving masters degrees

NAME
Total Number:

Names of personnel receiving PHDs

NAME
Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

1 a. University of Cincinnati

1 b. 51 Goodman Dr.
PO Box 210222
Cincinnati OH 452210222

Sub Contractor Numbers (c): C00037615-1
Patent Clause Number (d-1):
Patent Date (d-2):
Work Description (e): Task I. Development of GSM algorithms and software for rotorcraft flowsTask II. Develc
Sub Contract Award Date (f-1): 4/6/12 12:00AM
Sub Contract Est Completion Date(f-2): 4/6/15 12:00AM

1 a. University of Cincinnati

1 b. 311 Braunstein Hall, Clifton Avenue
Cincinnati OH 452210001

Sub Contractor Numbers (c): C00037615-1
Patent Clause Number (d-1):
Patent Date (d-2):
Work Description (e): Task I. Development of GSM algorithms and software for rotorcraft flowsTask II. Develc
Sub Contract Award Date (f-1): 4/6/12 12:00AM
Sub Contract Est Completion Date(f-2): 4/6/15 12:00AM

Inventions (DD882)

Scientific Progress

See Attachment.

Technology Transfer

1. Keep Dr. Roger Strawn at AMRDEC updated with our progress.
2. Visit Dr. Strawn's laboratory on July 18, 2014. C. L. gave a seminar to
describe ARO-funded work and ongoing multiscale modeling, flow control and thermal management research at the University of Missouri.

Foreword

We have developed a fluid-structure interaction (FSI) enabled solver by coupling the GSM-CFD solver and CSD solvers. The research includes three parts: 1) development and improvement of the GSM-CFD solver; 2) development of the FSI coupling code FSIMATE; 3) investigation of transonic shock/boundary-layer flows with a morphing blade.

1) For the GSM-CFD solver, a new matrix-based algorithm and its corresponding data structure have been devised to improve the computational efficiency and compactness of GSM. A new integration scheme has been proposed to improve the accuracy of GSM on distorted and hybrid meshes. The codes for FSI pair searching and shape function interpolation have been developed for the data exchange between the fluid part and the structural part. Different kinds of time-marching schemes have been compared, and the effects of tunable parameters have been investigated. Extensive numerical examples have been provided to show the effectiveness of the proposed methods.

2) In the effort of fluid-structure coupling, a partitioned fluid structure interaction (FSI) enabled simulator named FSIMATE has been developed to resolve the adaptive shock-boundary-surface control with morphing airfoils/blades at complex rotor flows. We have carried out the verification by comparing with results obtained by a commercial solver. The FSI solver can predict the solution as accurately as the commercial solver. A converter program has been developed to customize the output data for aero-acoustic analysis.

3) In the cases of the transonic shock/boundary layer flow control, we have demonstrated that the morphing layer can effectively weaken the transonic shock at the advancing azimuths and has negligible effects at other flow conditions. The characteristic of noise could be modified with a morphing layer.

During the execution of this program (April 2012- April 2015), five journal papers have been published, one has been accepted, and two conference papers have been presented.

Table of Contents

List of Figures and Tables

Statement of the Problem Studied

The proposed research aims to develop a practical, reliable, and easy-to-use solver for the simulation of rotorcraft flows with FSI enabled. Our ultimate goal is to apply this solver to develop a novel passive flow control concept to weaken shock strength, minimize transonic-shock-induced boundary-layer separation, and potentially modify characteristics of shock induced noise, which is critical to the Army's mission.

Technical Approach

a) Gradient smooth method

The matrix-based algorithm designed for gradient approximation is realized by assembling a geometrically dependent sparse matrix $\mathbf{B}$ from the gradient smoothing domains, and calculating the gradient as

$$\frac{\partial \mathbf{U}}{\partial s} = \mathbf{B}\mathbf{U} \tag{1}$$

where $\mathbf{U}$ is the vector storing the field variables. The size and values of the $\mathbf{B}$-matrix depend on the types of the smoothing domain and the integration scheme for GSM. The general edge-based data structure is devised to assemble the $\mathbf{B}$-matrix, and will be introduced in detail in the following sections. According to our observations, the original GSM for nodal gradient approximation cannot perform well on highly stretched hybrid meshes. The accuracy can be improved by increasing the number of integration points on the smoothing domain boundaries.

The interpolation module for FSI simulation are developed using conventional algorithms. To keep the conservation of load transfer, the fluid nodes are projected onto the structural surface, and the values are interpolated based on the shape functions. Similarly, the displacements of the fluid nodes are also interpolated from the nodal displacements of the structural part using the same shape function.

b) Development of FSI enabled solver

Our in-house developed FSI solver, FSIMATE, employs partitioned schemes with an implicit (strong coupling) algorithm [5], which is shown in Figure 1. This algorithm implies that the external flow field and structural response are solved successively during the iteration at each time step until a given convergence criterion is satisfied. We also provide a loose coupling algorithm in FSIMATE. This algorithm implies that the external flow field and structural response are solved successively during the prescribed times steps such as a rotating period until a given convergence criterion is satisfied. In FSIMATE, we have used two different CSD solvers: one is ANSYS, the other one is an open-source FEM software name Calculix.

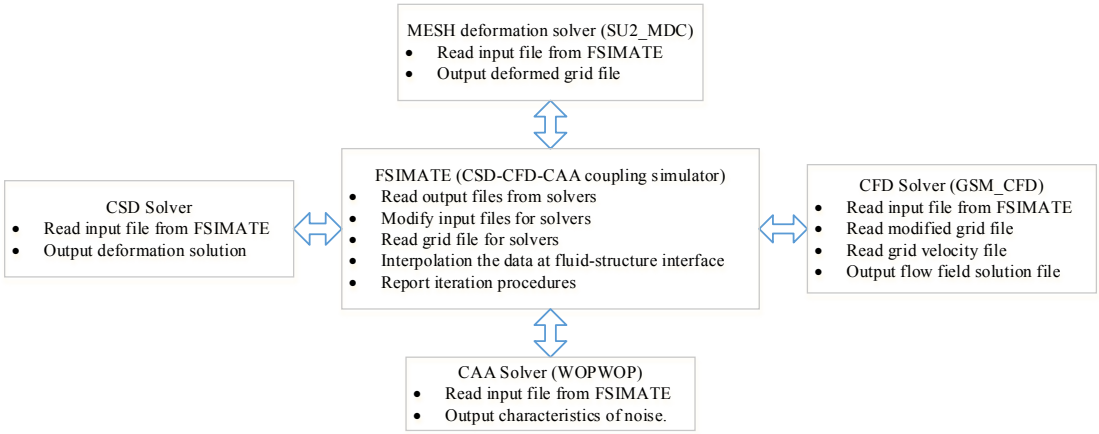
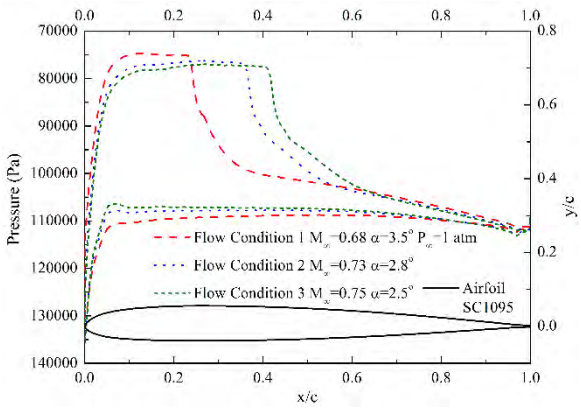


Figure 1. Schematic of FSI solver

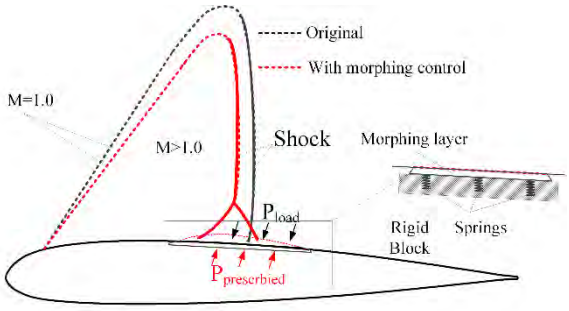
c) Transonic shock control with morphing layer

A local contour bump has been proven to reduce pressure drag due to transonic shock without increasing viscous drag significantly. The shock control bump supersedes the normal transonic shock with nearly isentropic compression or smears it into multiple shocks similar to a λ-shock structure. Unfortunately, the performance of contour bump solutions is highly sensitive to flow conditions. Under unsteady flow conditions, a fixed bump may deteriorate performance, for instance, by reinforcing the shock or boundary layer separation.

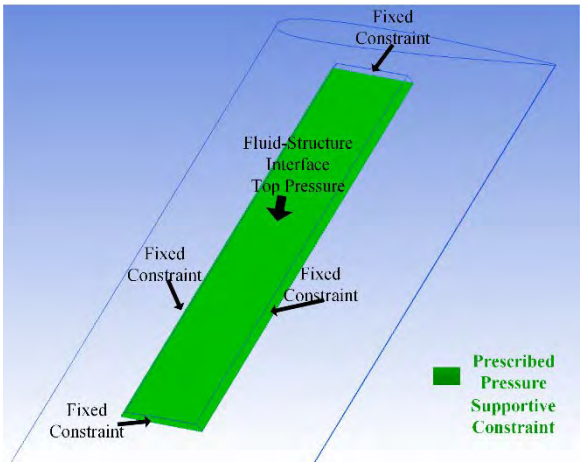
To overcome these shortcomings, a passively morphing layer for shock/boundary layer control of the rotating blade is presented in this work. Figure 2(a) shows the pressure distribution along the SC 1095 airfoil's upper surface under three different flow conditions. The pressure jump is observed at different positions across the normal shock. The morphing function of the blade is achieved by prescribing a constant pressure upon the bottom surface of the blade beneath the shock. This prescribed pressure can be created with the use of springs or some other similar structure, as shown in Figure 2(b). When the prescribed pressure is larger than the pressure load on the upper surface, a layer of deformable material beneath the shock will camber up to form a bump. A fixed rigid block is put under the morphing layer to provide the supportive constraint, which prevents the morphing layer from being pressed down; hence, the morphing layer will retreat to its initial shape when the top pressure exceeds the prescribed pressure. The boundary configurations for the morphing layer on the upper surface of the blade can be found in the Figure 2(c). The top surface of the morphing layer is a fluid-solid interface. The fixed constraints at the four sides prevent chordwise and span-wise movement.



(a) Pressure coefficients of original airfoil under various flow conditions



(b) Adaptive Shock/boundary layer control by morphing layer



(c) Boundary configurations of morphing layer embedded in rotor blade

Figure 2. Scheme of passive shock/boundary layer control via morphing layer

Transonic shock is a major contributor of High Speed Impulsive (HSI) noise. Since we see the formation of transonic shocks along the advancing azimuths in our CFD analysis we know that we will have HSI noise present. It is also noted earlier in this report that the implementation of a morphing layer is effective in reducing the strength of a normal transonic shock by manipulating it into a weaker, λ-structural shock. Thus, it is intuitive to believe that the weakening of the transonic shock will also weaken the acoustic signature produced by the blade. Thus, we have coupled our CFD results from FSIMATE with the computational aeroacoustics (CAA) solver PSU-WOPWOP.

Aeroacoustic analysis is rooted in the Lighthill Acoustic analogy and the Kirchhoff

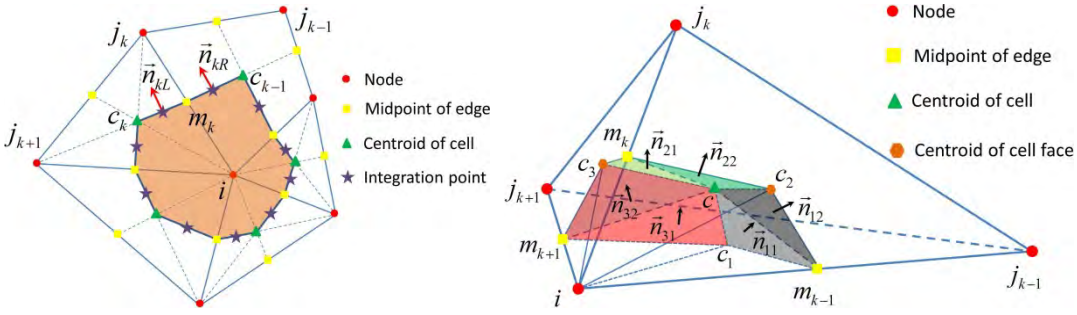
equation. From the Lighthill analogy we receive the Ffowcs Williams – Hawking equation, which forms the basis of the theory behind PSU-WOPWOP, a CAA solver which is utilized in conjunction with FSIMATE. PSU-WOPWOP utilizes Farasssat's Formulations to solve for the noise felt by an observer in the far field.

HSI noise is generated as a result of quadrupole source terms. Quadrupole source terms are sources of noise caused by nonlinear regions in the flow field, i.e. transonic shocks. Quadrupole noise is particularly difficult to analyze, as it typically requires a volume integration which is computationally demanding. However, PSU-WOPWOP allows us to use a specific type of data patch known as a permeable surface. The permeable surface contains all noise sources within it, so by nature it accounts for quadrupole noise sources as well as monopole and dipole sources as well. They are, however, a bit more difficult to use, as the data input required is much more detailed than that for an impermeable surface. As opposed to just requiring geometry and loading data, the permeable surface must contain all of the conservative variables, that is: ρ,ρu,ρv,ρw, and p_gauge. Care should also be taken when choosing the placement/geometry of the permeable surface as well, as it may not exceed the speed of sound, and the grid on it must be sufficient enough to accurately capture the flow variables with high enough resolution.

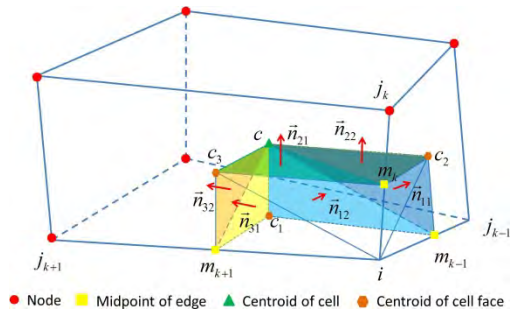
As such, our goal is to also couple the Computation Aeroacoustics (CAA) solver PSU-WOPWOP with FSIMATE to characterize the potential change in the acoustic signature of the blade equipped with the morphing layer, and compare it to a standard blade. A converter program, written in Python, is used to collect the variables from the CFD results and generate the input files for the CAA solver. FSIMATE, used in turn with the converter program, realizes our goal: a CSD-CFD-CAA coupling simulator with FSI enabled.

Summary of the most important results

a) Matrix-based GSM with improved integration schemes



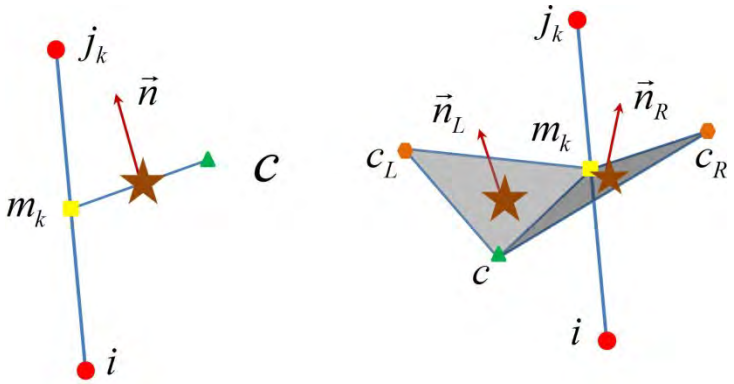
(a) 2D nGSD on hybrid mesh (b) partial 3D nGSD in a tetrahedral element



(c) partial 3D nGSD in a hexahedral element

Figure 3. Illustration of nGSDs

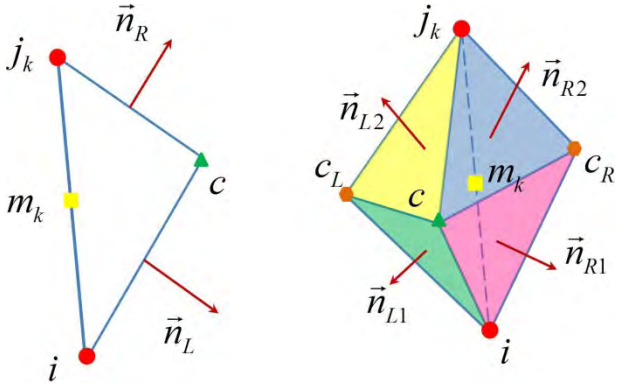
The smoothing domains of the node-associated GSM for 2D and 3D meshes are shown in Figure 3. In the original version of GSM the integration point was located at the midpoint of each edge, denoted as m_k in Figure 3; this scheme is named the mid-edge-point quadrature scheme here. In the improved version the integration points are located at the center of each segment of the smoothing domain boundaries, and it is denoted as the one-point quadrature scheme here. Numerical examples show that the accuracy of the gradient approximation can be greatly improved on the highly stretched hybrid meshes. The smoothing domains for midpoint-associated smoothing domains (mGSD) keep the same form as in our previous reports.



(a) 2D (b) 3D

Figure 4. Edge-based data structure for nGSM

The general edge-based data structures, as shown in Figure 4 and Figure 5, are devised to assemble the geometrically dependent $\mathbf{B}$-matrix. For each edge denoted as $i - j_k$, we need to record the global and local index of the edge and its two end nodes, and the outward unit normal and area for each segment associated with this edge. These data structures are used to construct the element $\mathbf{B}$-matrix by calculating the weight or contribution of each node in the element. Then the global $\mathbf{B}$-matrix can be assembled by accumulating the values of the element $\mathbf{B}$-matrix at corresponding global locations in the matrix.



(a) 2D (b) 3D

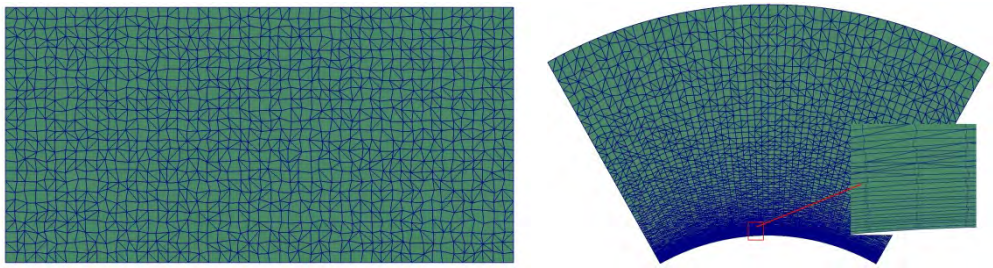
Figure 5. Edge-based data structure for mGSM

b) Results for gradient approximation on distorted and highly stretched meshes

The test meshes for gradient approximation are shown in Figure 6. Both of the meshes are randomly triangulated and distorted, and the second mesh is also stretched near the center of the circle. The linear, harmonic, and exponential variable fields are used as the test functions for the gradient approximation at nodes and edge midpoints.

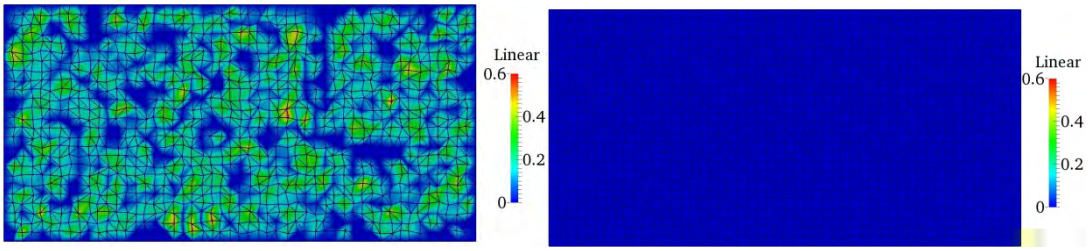
Figure 7 and Figure 8 show the error distributions of nodal gradients on the test meshes. We can observe that the newly proposed one-point quadrature scheme greatly improves the accuracy of the nodal gradient approximations on all three variable fields, especially on distorted and highly stretched meshes, as shown in the figures.

For the gradient approximation at edge midpoints, we only show the results on the stretched meshes where high accuracy is more difficult to achieve. There are two methods to obtain the gradients at the edge midpoint: one is to use the interpolation of the nodal gradients with directional correction (denoted as CAVG), and the other one is to use the gradient smoothing method (denoted as nGSM). Since we have two methods to calculate the nodal gradient, we have three sets of results on each variable field, as shown in Figures 8 through 10. From the results, we can observe that both CAVG-nGSM and mGSM can achieve much better accuracy than CAVG-GG, especially on the highly stretched meshes. We also note that the CAVG-nGSM perform even better than the mGSM, which means satisfactory results at the edge midpoint can be obtained using the interpolation methods provided that the nodal gradients are accurate enough.



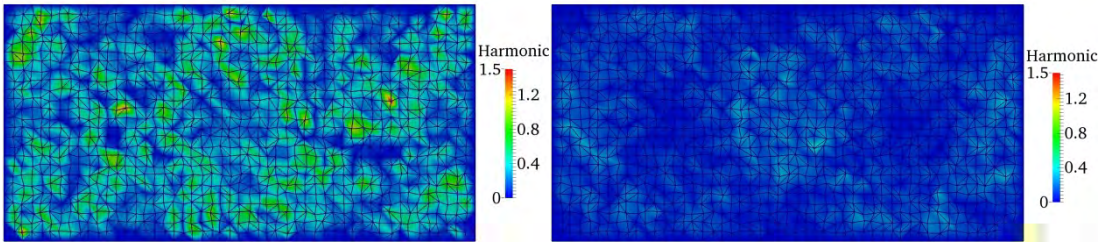
(a) distorted mesh (b) distorted and stretched mesh

Figure 6. Test mesh for gradient approximation



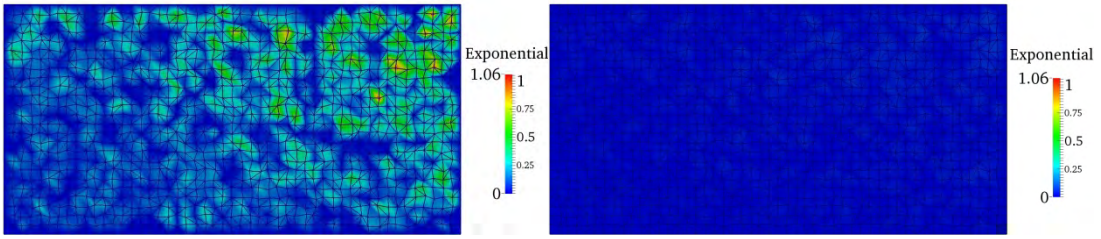
mid-edge-point scheme one-point scheme

(a) linear field $U(x,y)=x+2y$



mid-edge-point scheme one-point scheme

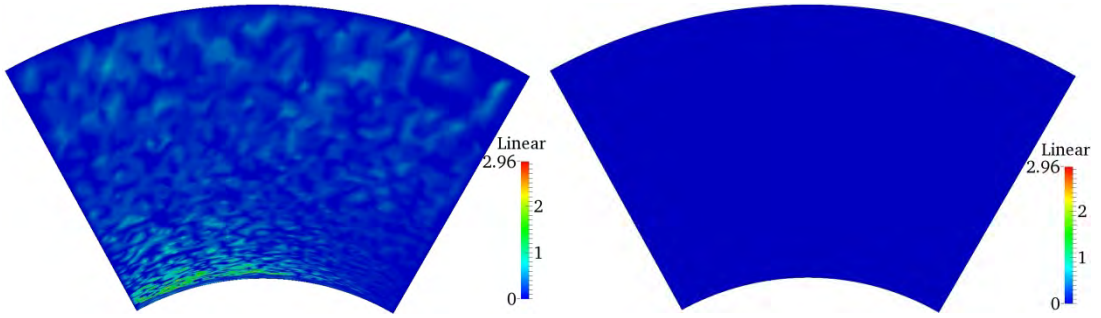
(b) harmonic field $U(x,y)=\sin 2\pi x\cos 2\pi y$



mid-edge-point scheme one-point scheme

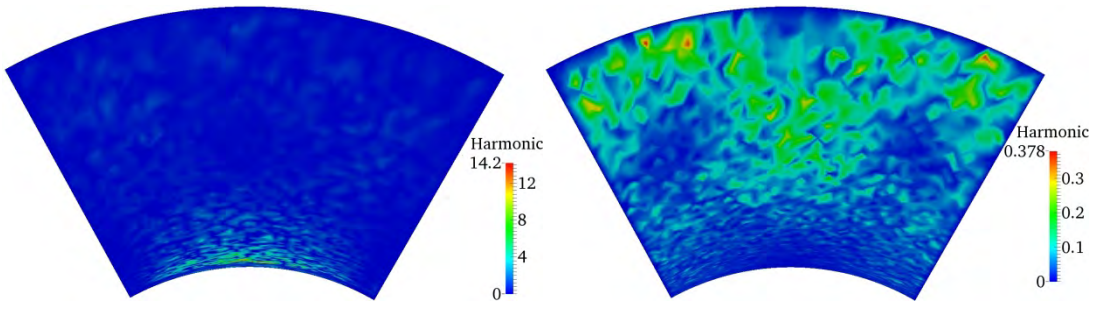
(c) exponential field $U(x,y)=e^{x+2y}$

Figure 7. Results of nodal gradient on distorted hybrid mesh



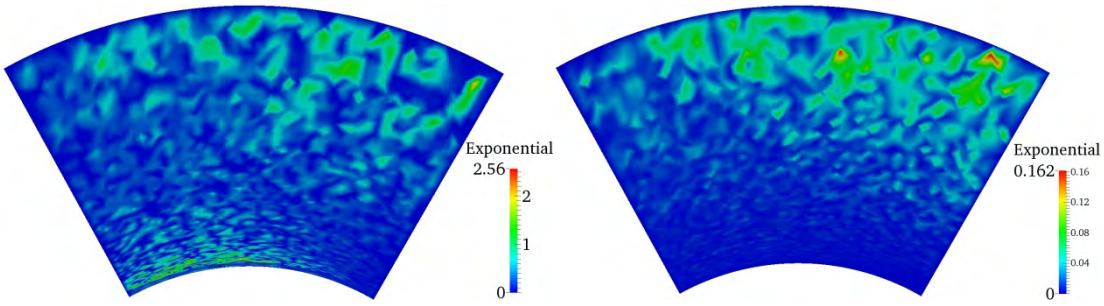
mid-edge-point scheme one-point scheme

(a) linear field $U(x,y)=x+2y$



mid-edge-point scheme one-point scheme

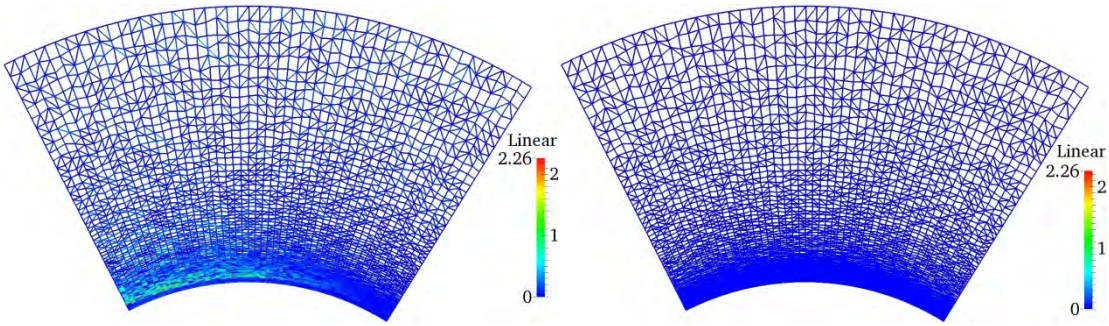
(b) harmonic field $U(x,y)=\sin 2\pi x\cos 2\pi y$



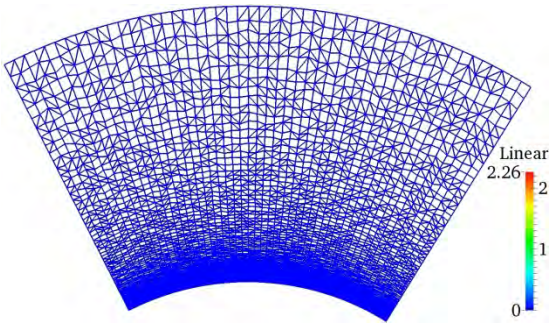
mid-edge-point scheme one-point scheme

(c) exponential field $U(x,y)=e^{x+2y}$

Figure 8. Results of nodal gradient on distorted and stretched hybrid mesh



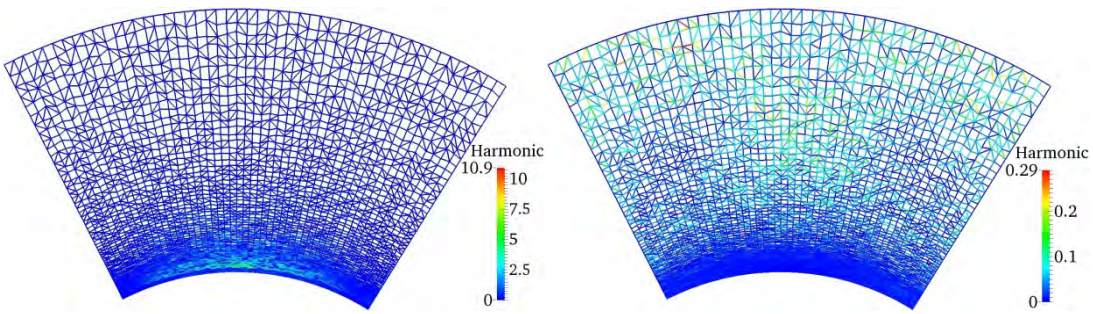
(a) CAVG-GG (b) CAVG-nGSM



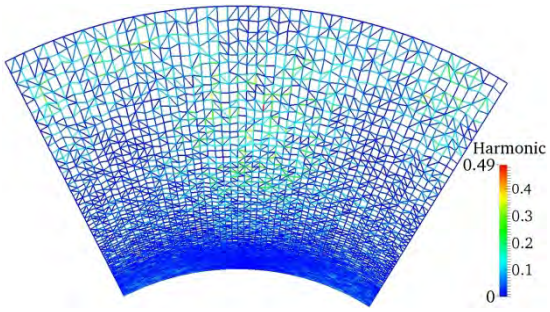
(c) mGSM

Figure 9. Gradient results at edge midpoints on distorted and stretched hybrid mesh for linear field

$$U(x,y)=x+2y$$

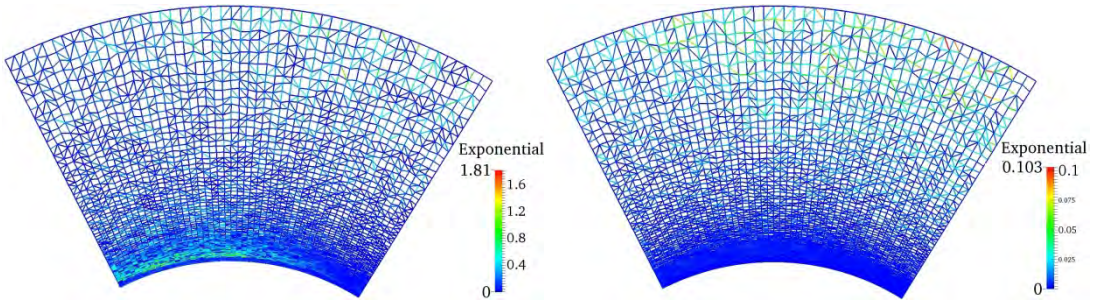


(a) CAVG-GG (b) CAVG-nGSM

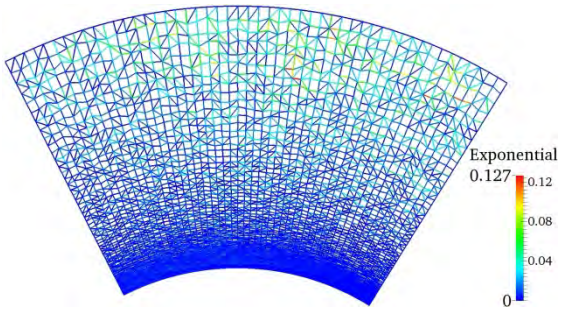


(c) mGSM

Figure 10. Gradient results at edge midpoints on distorted and stretched hybrid mesh for harmonic field $U(x,y)=\sin 2\pi x\cos 2\pi y$



(a) CAVG-GG (b) CAVG-nGSM



(c) mGSM

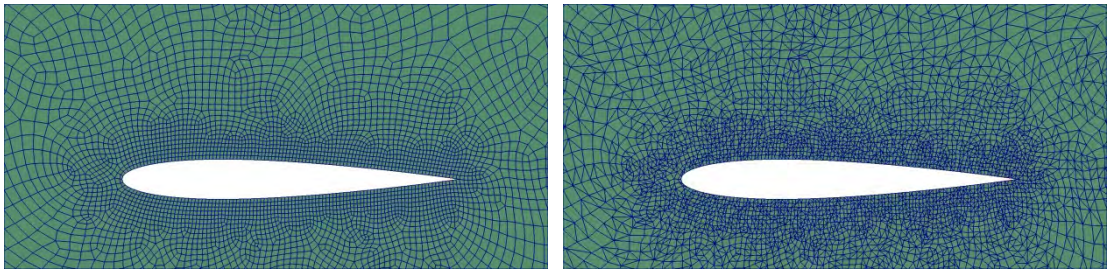
Figure 11. Gradient results at edge midpoints on distorted and stretched hybrid mesh for exponential field $U(x,y)=e^{x+2y}$

c) Inviscid flow results of different schemes

In the GSM-CFD solver the smoothing operation is applied not only to the field variables, such as velocities and pressures, but also to the convective fluxes. The inviscid transonic flow over a NACA 0012 airfoil is used as an example to show the effects of different integration schemes for GSM. The free stream conditions are set as: $Ma_\infty = 0.8$, α=1.25°, $T_\infty = 273.15K$, and $p_\infty = 1.01 \times 10^5 Pa$. Here two sets of meshes with the same number of nodes are used, one is unstructured quadrilateral and the other is a randomly triangulated hybrid mesh, as shown in Figure 12.

The pressure coefficients of the suction side (upper side) of the airfoil obtained on the two test meshes are shown in Figure 13. The reference solution is calculated on a much finer mesh.

We can observe that the nGSM can better capture the shock on both the unstructured mesh and the distorted hybrid mesh. Also, the results obtained using nGSM on the two meshes have smaller differences than the ones obtained using the GG method, thus we can conclude that the nGSM is more robust (insensitive to mesh qualities) than the GG method.



(a) quadrilateral mesh (b) hybrid mesh

Figure 12. Meshes for inviscid flow calculation of NACA 0012 airfoil

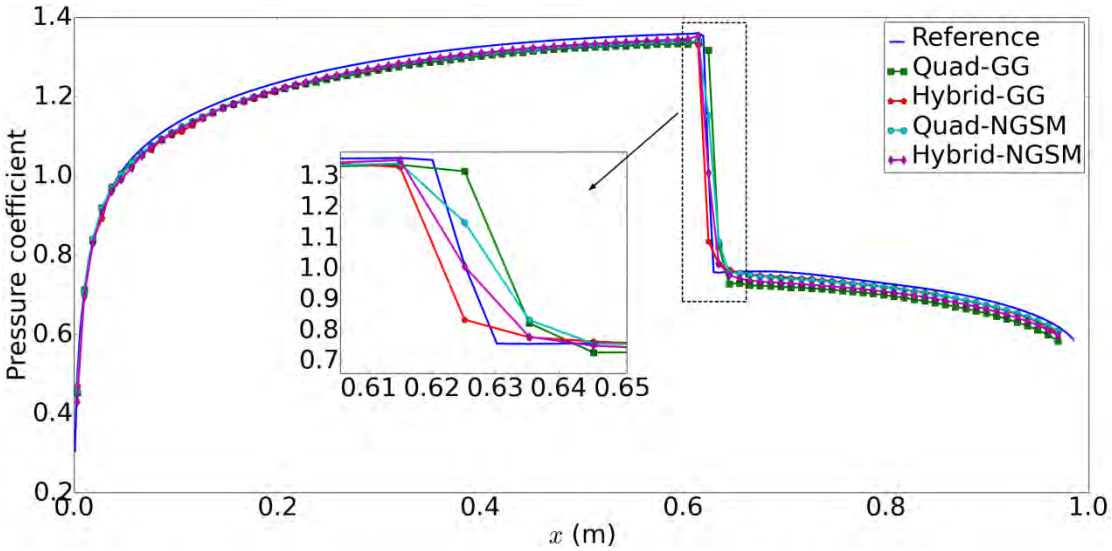


Figure 13. Pressure coefficients obtained using different schemes on quadrilateral and hybrid meshes

d) Acceleration techniques

The multi-grid method and parallel computation are used to speed up the computation and keep the computational time in a reasonable range. So far we have conducted the following tests.

(1) Multi-grid method

The agglomeration multi-grid method is used to generate the coarse girds. It should be pointed out that the GSM and AVG_GRAD_CORRECTED schemes are only applied on the finest grid; the simple average scheme is used for the other levels of the grid. Take the laminar flow over NACA 0012 as an example, the effects of multi-grid technique are shown in Table 1. We can see that the multi-grid can greatly accelerate convergence and reduce the total computational time.

Table 1 Effects of multi-grid technique (only for GSM-CFD)

Grid level	Time/iteration	Total number of iteration	Total computation time(s)

0	0.078	4219	329
1	0.177	1700	302
2	0.195	1092	212
3	0.199	817	163
4	0.203	729	148

(2) Parallel computation

SU2 supports MPICH2 for parallel computation. Here, we take the turbulent flow over an ONERA M6 wing as example. The effects of parallel computation are shown in Table 2. We can see that the speedup of GSM-CFD is not very good under the current software architecture, and we still need to improve the efficiency of parallel computation.

Table 2 Effects of parallel computation (only for GSM-CFD)

No. of cores	1	2	4
Time(s)/iteration	1.025	0.686	0.588

e) Test cases of GSM-CFD

(1) 2D Laminar flow

For viscous flows, the hybrid mesh is used to provide enough resolution at the boundary layer region. The hybrid mesh of a 2D NACA 0012 airfoil is shown in Figure 4. The problem is solved at Mach number (Ma) =0.8, Angle of Attack (AoA) =1.25, P =1.01E5Pa, T =273.15K and Re =5000. The results obtained using different methods on the same mesh are shown in Figure 14 and Figure 15. In this report, three methods, namely the simple average gradient method (AVG_GRAD), the average gradient with directional correction (AVG_GRAD_CORRECTED) and the gradient smoothing method (GSM), are compared. Through the Mach contours shown in Figure 4, we can see that all three methods produce similar results. To further compare the differences, the pressure coefficients (C_P) obtained by the three methods are shown in Figure 5. We can see that the simple average method produces the least accurate results, and the results of the other two methods are quite the same. It should be noted that the most obvious differences in Cp appear at the leading and trailing edge, especially at the leading edge.

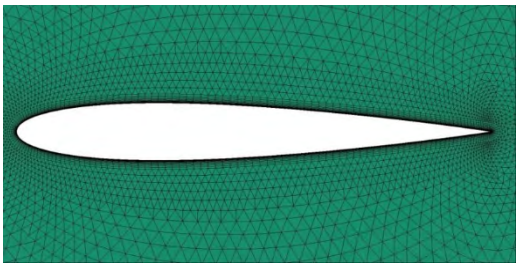
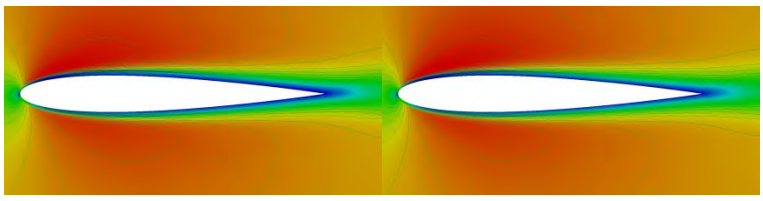
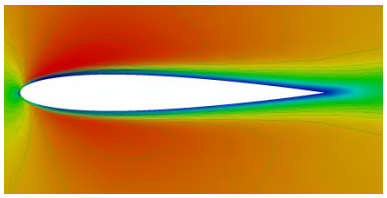


Figure 14. Hybrid mesh of NACA 0012





(a) average gradient (b) average gradient with directional correction (c) GSM

Figure 15. Mach number contours of NACA 0012 with different methods

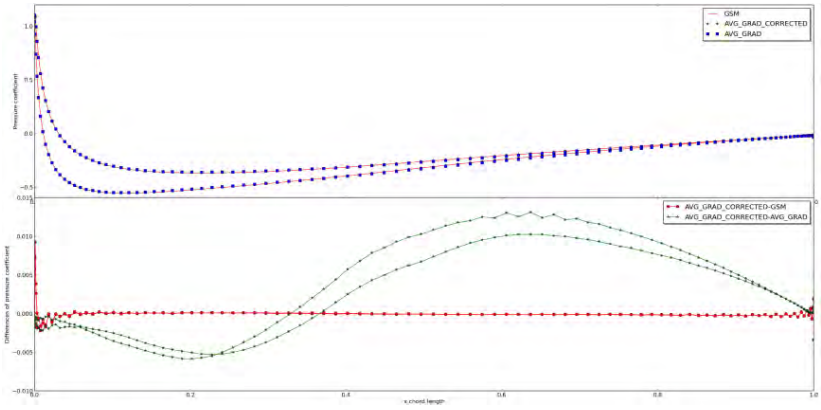


Figure 16. Pressure coefficients of NACA 0012 obtained by different methods

(2) 2D turbulent flow

A RAE2822 airfoil analyzed at Ma =0.729, AoA =2.31 and Re =6.5M is used for the test case for turbulent flow. The hybrid mesh is shown in Figure 7. The pressure coefficients obtained using different methods are shown in Figure 8. We can see that the AVG_GRAD_CORRECTED and GSM-CFD can produce almost the same results, except for differences around the shock region mainly due to the different shock locations captured. However, the results of the simple average scheme AVG_GRAD are not as good as the other two.

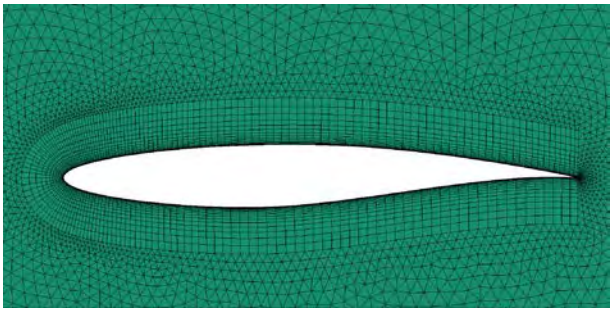


Figure 17. Mesh for RAE 2822 airfoil

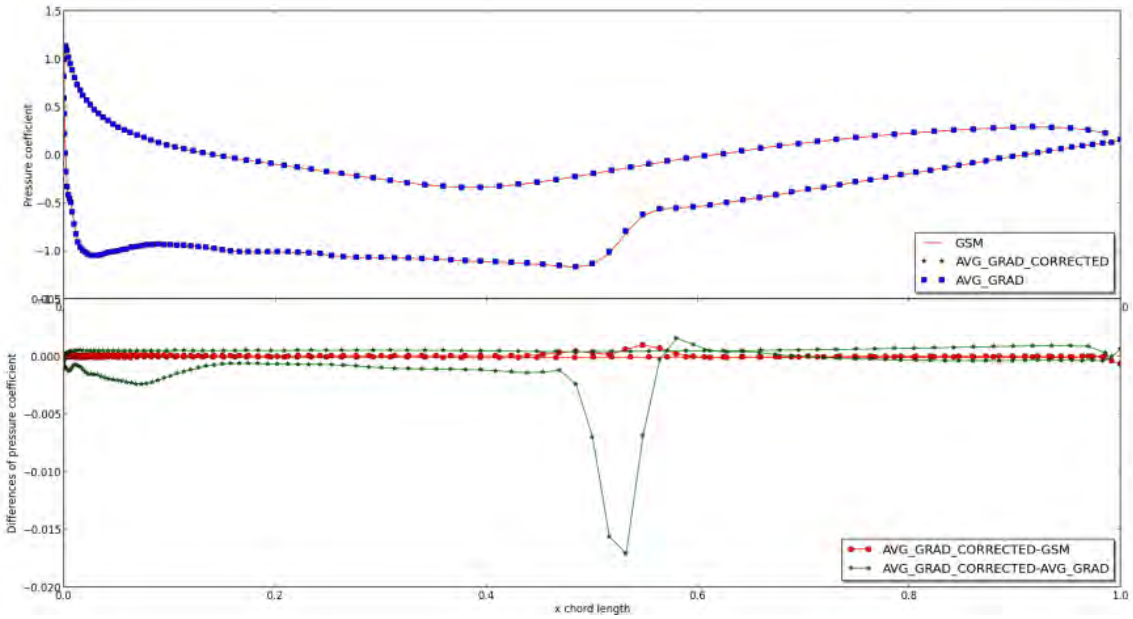


Figure 18. Pressure coefficients of RAE 2822 airfoil

(3) 3D turbulent flow on hexahedral mesh

The ONERA M6 wing is analyzed at Ma=0.8395, α=3.06, T=273.15K, and Re=11.72M. The structured hexahedral mesh of the wing is shown in Figure 9. The contours of the Mach number at different sections of the wing are shown in Figure 19. We can see that the Mach number distributions of the results obtained using two different methods are quite the same, but the values are slightly different. This may be caused by the different shock locations captured. The shock locations also affect the distributions of the pressure coefficient, as shown in Figure 20. We can see that although the C_P distributions are the same, the C_P at exactly the same positions can be very different due to the slight location differences of the shock. However, the overall effects of the wing, the lift and drag, are the same. C_L=0.260493 calculated using GSM-CFD and C_L=0.206945 calculated using AVG_GRAD_CORRECTED, which are both in good agreement with the reference value.

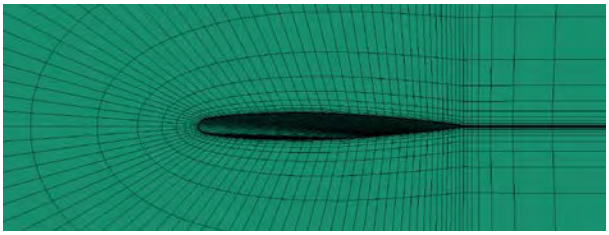
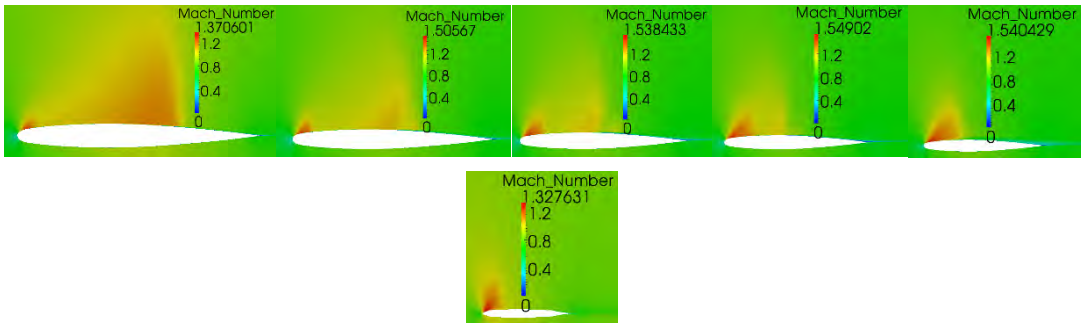
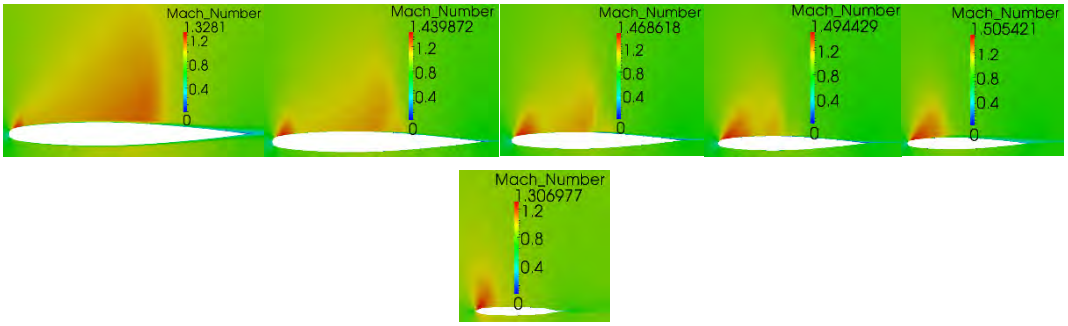


Figure 19. Hexahedral mesh of ONERA M6 wing



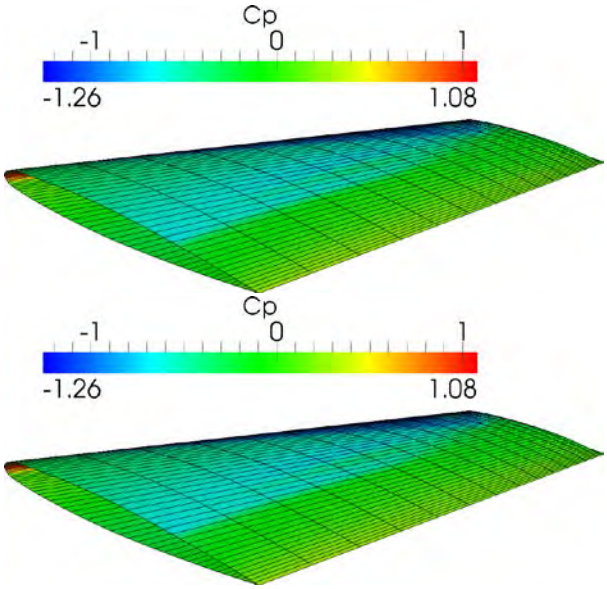
GSM



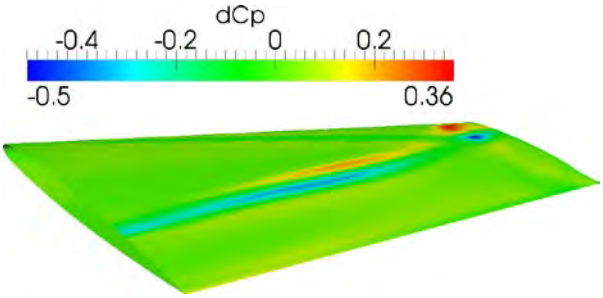
Average gradient with directional correction

(a) y=0.2 (b) y=0.4 (c) y=0.6 (d) y=0.8 (e) y=1.0 (f) y=1.2

Figure 20. Mach number contours of ONERA M6 wing at different locations



(a) Average gradient with directional correction (b) GSM



(c) Contour of differences of Cp at the same position

Figure 21. Pressure coefficient of ONERA M6 wing

4) 3D inviscid flow on tetrahedral mesh

We still use the ONERA M6 wing as the example for tetrahedral mesh. The unstructured tetrahedral mesh is shown in Figure 22. The pressure distributions are shown in Figure 23. The examples of laminar and turbulent flow on tetrahedral or hybrid mesh are still under preparation, and can be provided very soon.

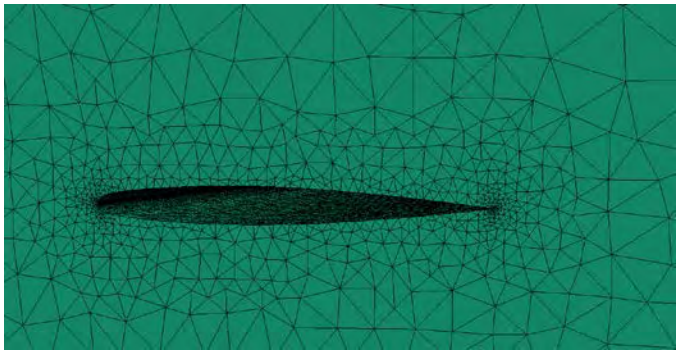


Figure 22. Unstructured tetrahedral mesh of ONERA M6 wing

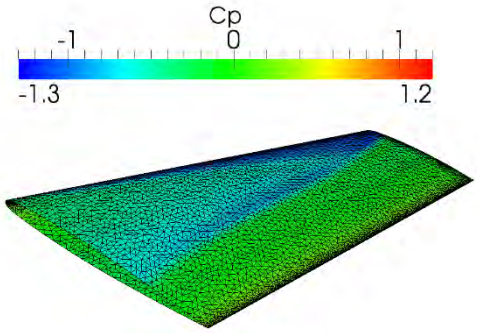


Figure 23. Distribution of pressure coefficients of ONERA M6 wing

f) Interpolation module for FSI simulation

For the data exchange between the fluid and the structure domain, we need to project fluid nodes to the structural surface and find the FSI pair at first. As shown in Figure 24, there may be more than one fluid node associated with one structural element. During the searching and the interpolation process, shape function plays a very important role. If the following conditions are satisfied,

$$\sum_{I=1}^{3} N_I(\chi_i) = 1, \quad N_I(\chi_i) \geq 0 \tag{2}$$

we can determine that the fluid node X_i is associated with the current structural element, where χ_i is the projection coordinate of X_i, and $N_I(\chi_i)$ is the shape function value at structure node I. For the fluid loads interpolation, the interpolated loads at structure node I can be calculated as

$$F_I^{\mathrm{S}} = \sum_{i=1}^{n} N_I(\chi_i) F_i^{\mathrm{F}}(X_i) \tag{3}$$

where n is the number of fluid nodes associated with the structural element. For the structural displacement interpolation, the interpolated displacements at fluid node i can also be calculated using the shape functions as

$$x_i^{\mathrm{F}} = \sum_{I=1}^{3} N_I(\chi_i) x_I^{\mathrm{S}} \tag{4}$$

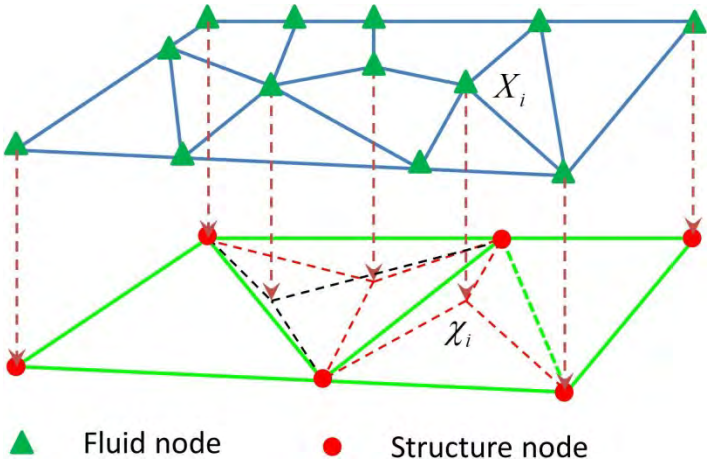


Figure 24. Illustration of data exchange for fluid-structure interaction

g) Time-marching schemes

A simple 2-DOF aeroelastic model, shown in Figure 25, is used as the numerical example to compare various time-marching schemes for the fluid-structure interactions. During the comparison, the results obtained using a monolithic method with a very small time step are used as the reference solutions.

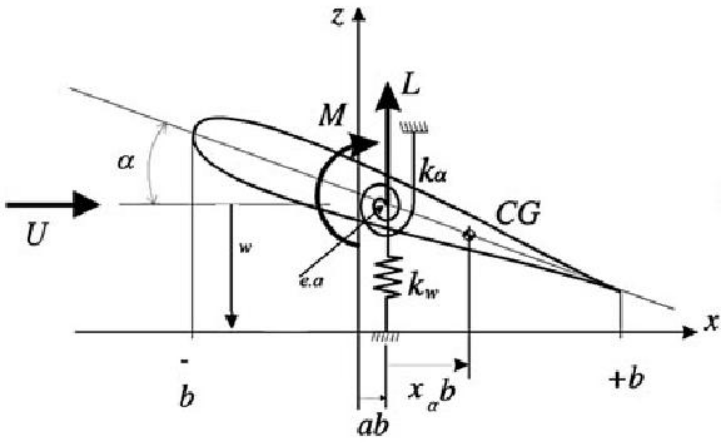


Figure 25. Illustration of 2-DOF aeroelastic model1

Here, five kinds of time-marching schemes are compared in this research:

I. Conventional staggered serial schemes, CSS

II. Iterative staggered serial schemes, ISS

III. Generalized staggered serial schemes, GSS

IV. Generalized staggered serial schemes with half-time offset, GSS0.5

V. Serial staggered schemes with fluid force prediction, FPSS

There are tunable relaxation parameters for structural and fluid results. The structural relaxation or prediction provides a modified configuration for the fluid loads calculation, and the relaxation or prediction for the fluid force provides a modified load for the structural calculation. The relaxations of the structural and fluid results can be written as

$$X_s^{n+1} = (1-\alpha) X_s^{n+1} + \alpha X_s^n \tag{5}$$

[1] A. Abdelkefi, R. Vasconcellos, A.H. Nayfeh, and M.R. Hajj. An analytical and experimental investigation into limit-cycle oscillations of an aeroelastic system. Nonlinear Dyn, 71(1-2):159-173, 2013.

and

$$F_{\mathrm{f}}^{n+1} = (1-\beta) F_{\mathrm{f}}^{n+1} + \beta F_{\mathrm{f}}^{n} \tag{6}$$

to combine the results of two consecutive steps. The displacement predictions in the GSS and GSS0.5 schemes are determined by the kinetic equations as

$$x^{\mathrm{p}} = x^{n} + \alpha_1 \Delta t v^{n} + \alpha_2 \Delta t \left(v^{n} - v^{n-1} \right) \tag{7}$$

For the GSS scheme, we have $\alpha_1 = 1,\ \alpha_2 = 0$ for the first order prediction, and $\alpha_1 = 1,\ \alpha_2 = 0.5$ for the second order prediction. Similarly, for the GSS0.5 scheme, we have $\alpha_1 = 0.5,\ \alpha_2 = 0$ for the first order prediction and $\alpha_1 = 0.5,\ \alpha_2 = 0.125$ for the second order prediction.

Table 3. Response results of different fluid relaxation parameters

Scheme	α	α_2	β	ϵ_h (%)	ϵ_α (%)	ϵ_ϕ (%)
ISS	0		0.5	0.1418	0.0830	8.342
ISS	0		0	-0.2227	-0.1048	5.456
ISS	0		-0.5	-0.2438	-0.1163	5.456
CSS	0		1	95.07	50.28	-52.28
CSS	0		0.5	83.89	45.58	21.33
CSS	0		0	63.79	34.99	-40.73
CSS	0		-0.5	35.37	19.77	-42.18
CSS	0		-1	2.013	2.162	6.900
GSS	1.0	0.5	1	63.87	35.30	-42.18
GSS	1.0	0.5	0.5	34.44	19.37	-42.18
GSS	1.0	0.5	0	-0.4281	-0.2785	5.4561
GSS0.5	0.5	0.125	0	34.81	19.73	-42.18
GSS0.5	0.5	0.125	-0.5	0.5145	0.6045	5.456
GSS0.5	0.5	0.125	0.5	63.64	34.56	-40.70
FPSS	0		0	2.004	2.149	6.900
Reference				0.0121703 m	0.0662163 rad	0.346399 s

Table 4. Response results of different structural relaxation parameters

Scheme	α	α_2	β	ϵ_h (%)	ϵ_α (%)	ϵ_ϕ (%)
ISS	0.5		0.5	-0.1767	-0.07651	8.343
ISS	0		0	-0.2227	-0.1048	5.456
ISS	0.5		0.0	-0.2049	-0.0921	8.343
CSS	0.4889		1.0	-0.2504	1.153	N/A
CSS	0.325		0.0	-0.1532	0.8205	N/A
CSS	0.418		0.5	-0.1818	0.9934	N/A
CSS	0		-1	2.013	2.162	6.900
CSS	0.013		-1	0.1159	0.8527	N/A
GSS	1.0	0.0	0	0.3895	-0.1253	5.456
GSS	1.0	0.5	0	-0.4281	-0.2785	5.456
GSS0.5	0.5	0.0	-0.5	0.6984	0.6294	5.456
GSS0.5	0.5	0.125	-0.5	0.5145	0.6045	5.456
FPSS	0		0	2.004	2.149	6.900
FPSS	-0.02		0	0.06704	1.242	-0.2886
Reference				0.0121703 m	0.0662163 rad	0.346399 s

From the response results, we have the following remarks:

I. The iterative methods (ISS) can always produce results with high accuracy, but require several iterations for each time step, which would be extremely time-consuming for engineering applications.

II. If the parameters are properly chosen, the loosely-coupled schemes can also produce satisfactory response results with much less computational efforts, but are less accurate than

the ISS schemes.

III. With standard options, the GSS and GSS0.5 are more accurate than CSS and FPSS schemes. But when the structural parameter of the FPSS scheme is slightly modified, more accurate results can be obtained. All three kinds of schemes are good options for the time-marching schemes of the FSI simulation of helicopter blades.

h) Validation of FSIMATE

The main FSI solver, FSIMATE, is developed by coupling a CFD solver: GSM_CFD and a CSD solver: ANSYS. As experimental data for a blade with a morphing layer is unavailable, the results of FSIMATE are verified by comparing with those of the commercial software, ANSYS in [3]. The same profile (SC 1095), blade dimensions, mesh, flow conditions, and material properties of the morphing layer are applied. To verify the solutions with the morphing layer, the C_L and C_D (lift and drag coefficients, respectively) at the advancing side, and the maximum deflection of the morphing layer are compared in Figure 26. It can be found that the predictions made by FSIMATE are consistent with those made by the commercial software. Therefore, it is considered that FSIMATE is validated. Figure 27 shows the pressure coefficients, C_P, on the blade's upper surfaces (with/without morphing layer) at the advancing side; the azimuthal angle is from 0º to 180º. It is seen that from 75º to 120º the supersonic regions are significantly affected by the morphing layer. The normal transonic shock has changed to a λ-shock structure due to the bump formed by the morphing layer.

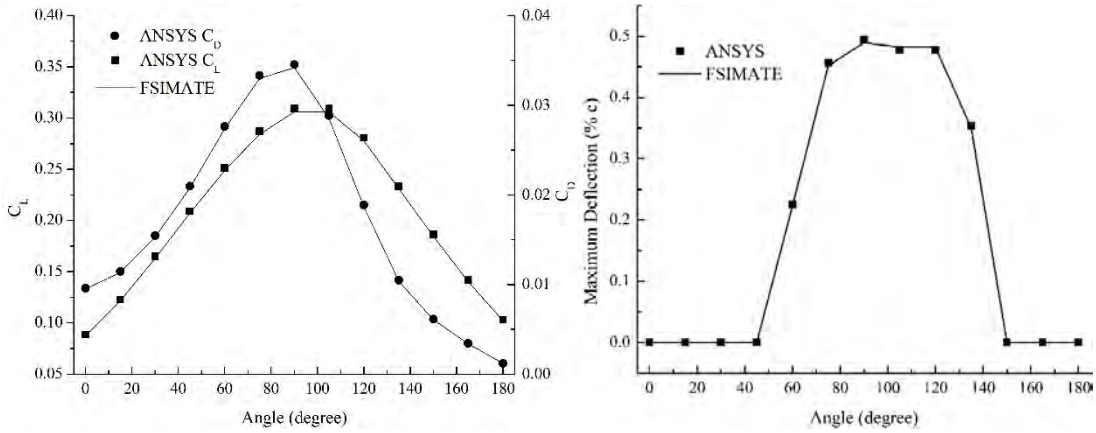
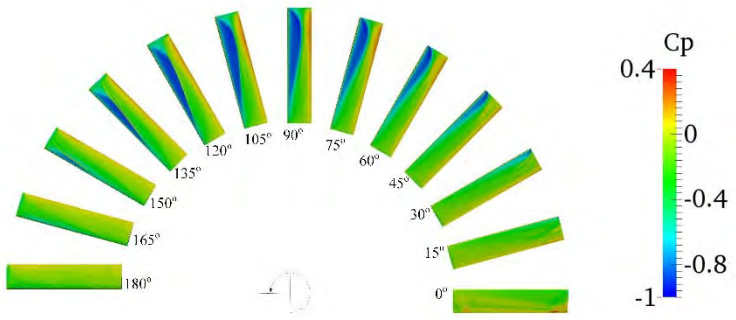


Figure 26. Comparison of C_L, C_D and maximum deflection between ANSYS and FSIMATE (0°-180°)



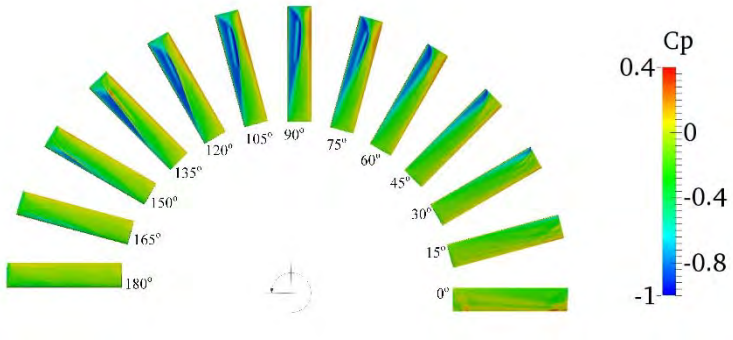
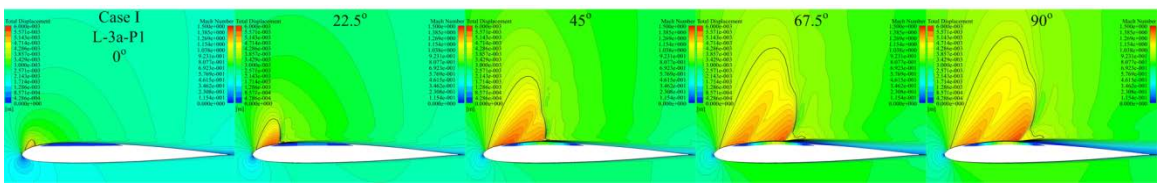


Figure 27. Pressure coefficients on the upper surface of SC1095 without/with morphing layer (0°-180°)

i) Two-dimensional airfoil with different properties of morphing layers

In the investigation of passively adaptive flow control methods, the morphing airfoils (SC1095) at three different span stations are simulated with a constant rotating speed and free-stream velocity. The effects of position, coverage, and material properties of morphing surfaces are studied. Figure 28 shows the Mach number contours of four morphing surfaces with different positions and coverage in the first quadrant of the advancing side: L-3a-P1, L-3b-P1, L-3c-P1 and L-5-P1. Here, L-3a-P1, L-3b-P1, and L-3c-P1 denote the morphing surfaces with the same properties and coverage of 30% chord length and three different positions. L-5-P1 denotes the morphing surface covering 50% of the chord length. Table 5 lists the effects of morphing surface at different azimuthal angles. It can be found that the morphing surfaces do not deform at 0º and 22.5º, because the shock is entirely out of the surface. It is noticed that at 67.5º and 90º, the λ-shock is replaced by two normal shocks for L-3a-P1, and at 45º, L-3b-P1 and L-3c-P1 do not affect the flow fields due to the same reason at 0º and 22.5º. Figure 29 compares C_L/C_D of the above four surfaces:

- L-3a-P1: C_L/C_D is almost the same as original airfoil from 0º to 22.5º, then increases prominently from 22.5º to 67.5º, therein, reaches the maximum improvement of 30% at 45º, then, decreases from 67.5º to 90º, and reaches a maximum deterioration of -6.5% at 90º.
- L-3b-P1, the improvement of C_L/C_D begins from 45˚ and reaches the maximum of 21.5% at 58.5˚, then decreases to 5% at 90˚.
- The effects of L-3c-P1 on the performance of airfoil are simpler than L-3a-P1 and L-3b-P1. C_L/C_D increase from 54˚ and reach the maximum improvement of 13% at 90˚.
- L-5-P1 has a wider effective range of [27.5˚, 90˚] and average improvement (maximum is 14% at 58.5˚) on C_L/C_D.



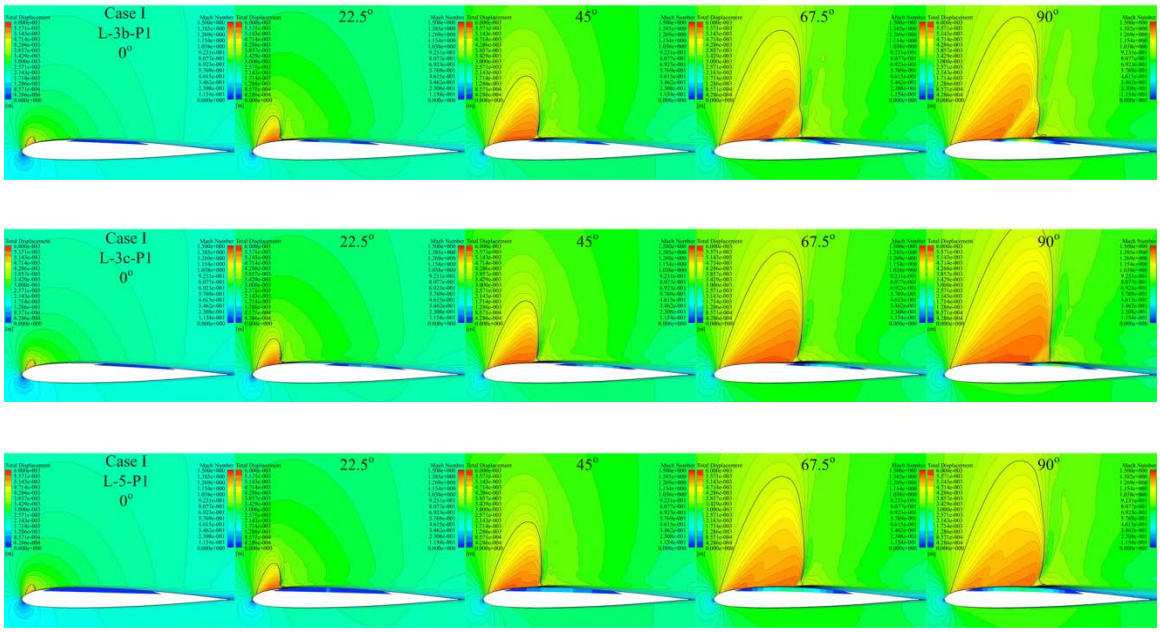


Figure 28. Mach number contours of four morphing airfoils in first quarter of advancing side
Supersonic and subsonic area are separated by a sonic line

Table 5. Effects of morphing surfaces on the transonic shock at different azimuthal angles

	0°	22.5°	45°	67.5°	90°
L-3a-P1			λ-shock	two normal shocks	
L-3b-P1				λ-shock	
L-3c-P1				λ-shock	
L-5-P1			λ-shock		

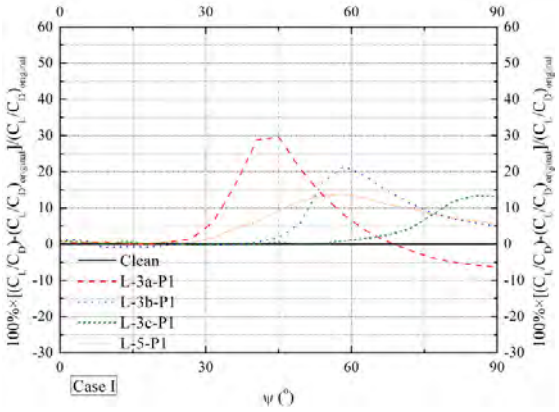


Figure 29. Performance (C_L/C_D) of four morphing airfoils in first quarter of advancing side

j) Morphing control with feedback

To increase the self-adaptive performance of morphing surfaces, we introduce the pressure sensors at the beginning and end of the morphing surface. The sensors will measure the pressure and change the prescribed pressure underneath the morphing surface, and assist the morphing surface to respond to the considerable variation of flow field well. The goal of the feedback is to restrain useless or even detrimental deformation of morphing surface.

We investigate this setup at unsteady rotating flow conditions. In this case, the rotating speed

of the airfoil is Mach 0.65, the forward speed (free stream) is Mach 0.24. Apparently, due to the superposition of two speeds, the position and strength of shock change at different azimuthal angles. As shown in Figure 30, the shock position and strength change significantly at the advancing side. The performance of the bump has been proven to be sensitive to changes in flow conditions. In some cases, for example, the bump precedes the shock and can strengthen the shock and deteriorate the performance of the airfoil. Figure 18 shows the flow control results of the morphing airfoil. At 0° the shock disappears and the morphing surface remains its initial state; at 60° the shock appears but does not move upon the morphing surface, hence the morphing surface does not morph unnecessarily; at 75° a transonic normal shock appears and the morphing surface forms a bump according to the measured pressure; as a result, a λ-shock structure is produced; at 90° the transonic normal shock becomes stronger and the morphing surface keeps the bump shape to control the shock; at 105° the strength of the shock continues to increase, the shock position moves behind the morphing surface, the morphing surface retracts to its initial state to prevent strengthening the shock; at 120° and 135° the shock strength decreases as the position moves forward, the morphing surface forms a bump to control this weaker shock; at 180°the surface does not morph. The results demonstrate that the presence of pressure sensors overcomes the shortcomings of a purely passive morphing surface.

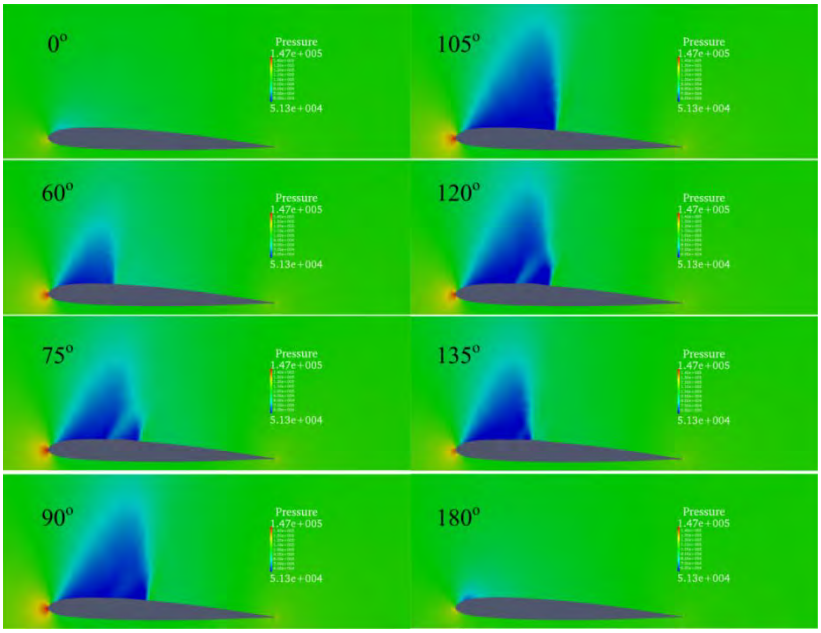


Figure 30. Pressure contours of morphing airfoils as azimuthal angles at forward flow conditions

k) Morphing control for rotor blade at high-speed forward conditions

The rotor blade with a morphing surface is studied at high-speed forward conditions [3]. Figure 31(a) shows the pressure coefficients on the upper surface of the original blade from azimuthal angles of 60° to 135°. From 60° the area covered by the shock region increases as the azimuthal angle increases and reaches a maximum at 90°. At 105° and 120° the shock region does not decrease significantly due to transient influences. At 135° the shock region decreases prominently. The inclined morphing surface is designed based on the fact that the strongest shock generally happens close to the blade tip. Hence, the bump with non-uniform

distribution along the spanwise direction is expected to control the non-uniform shock. Figure 31(b) shows the displacement of morphing blade 4 from 60° to 135°. The inhomogeneous displacement is obvious because of the corresponding distribution of the transonic shock region. The maximum displacement is reached at 105°, close to 0.5%c, which indicates the occurrence of the strongest normal shock. The displacement decreases simultaneously as the shock region shrinks. Figure 32 shows the Mach number contours at an azimuthal angle of 90°. As mentioned in previous work, the bump formed by the morphing surface weakens the normal shock by producing a λ-shock structure, namely an oblique shock and a weaker normal shock. Figure 33 compares the pressure coefficients between the original blade and blades with morphing surface 4-5. At 60° and 135° morphing surfaces weaken the shock insignificantly. At other angles, the shocks at different span stations are well weakened.

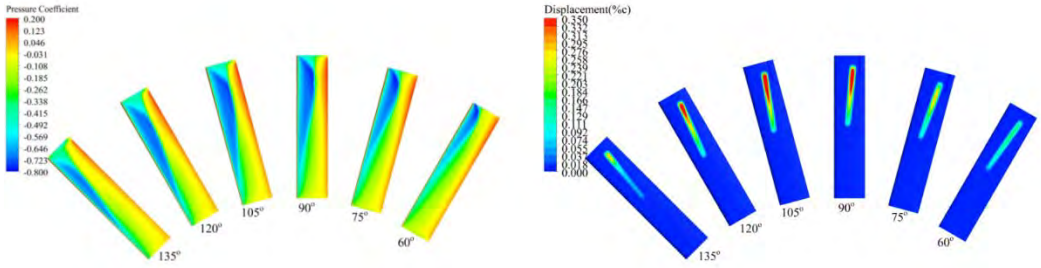


Figure 31. (a) Pressure coefficient on the upper blade at the advancing side (b) Corresponding deformation of morphing surface

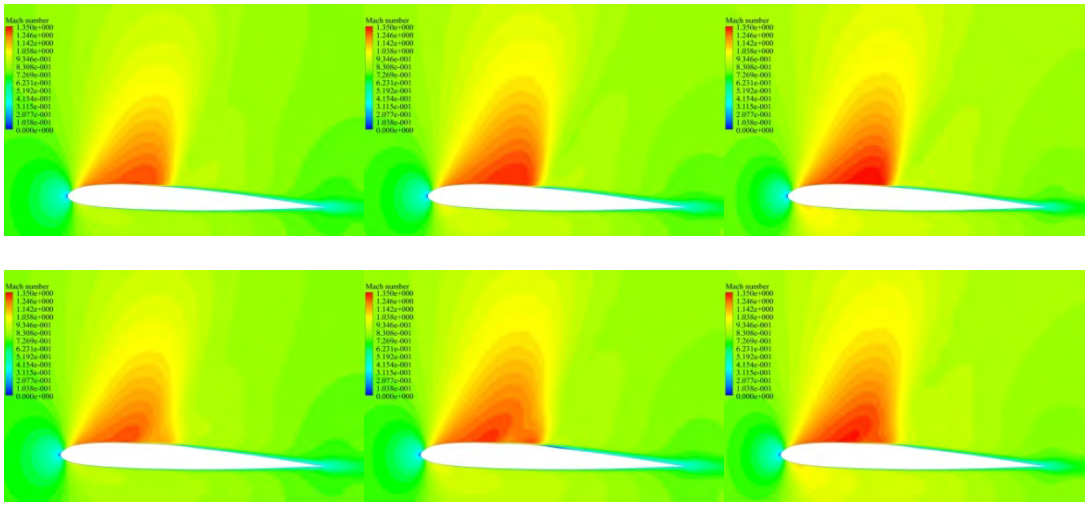


Figure 32. Mach number contours without /with morphing surface at 90°, span station is 5.25/6, 5.5/6, 5.75/6

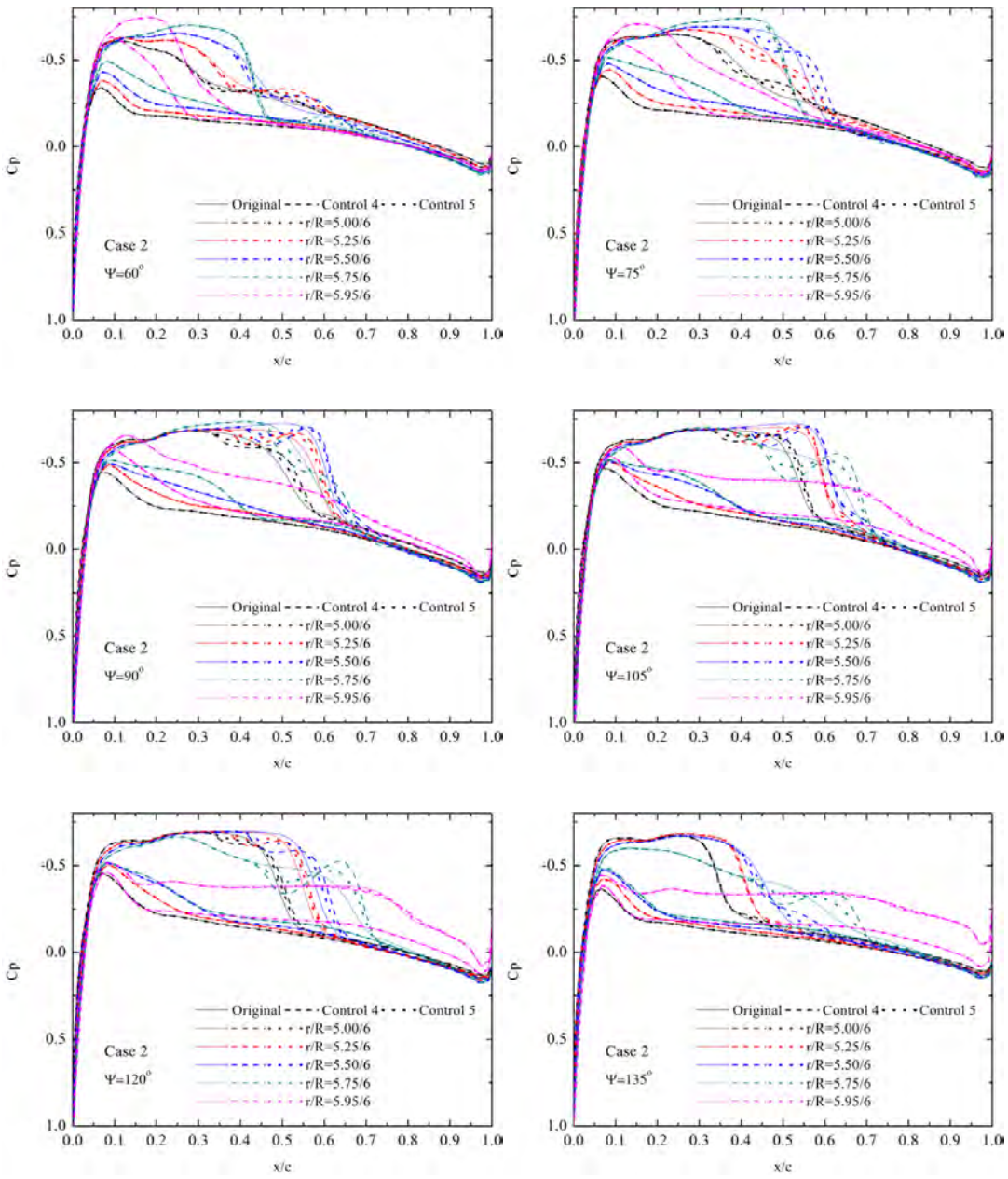
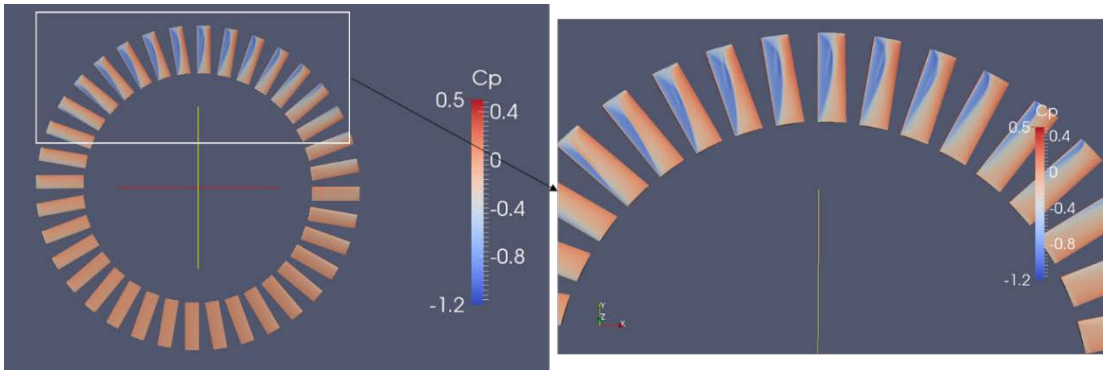


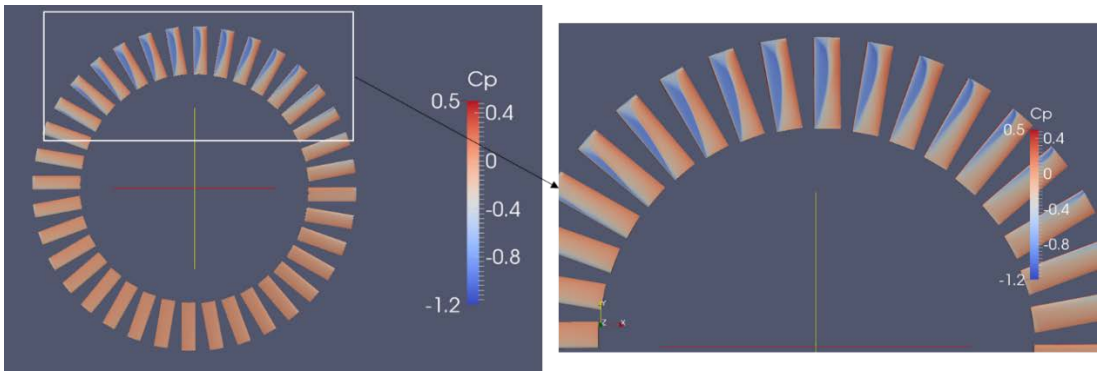
Figure 33. Pressure coefficients as span stations and azimuthal angles for original SC1095, with morphing 4, 5

l) Noise characteristic by FSIMATE with CAA

In this section, we demonstrate the results by CFD-CSD-CAA coupling solver. In the FSI simulation, a physical time step of azimuthal angle=0.5 degree is used. Figure 34 shows the pressure coefficients on the upper surface of the blade from azimuthal angles of 0° to 360°, with the step of 10°. It is seen that from 70° to 120°, the distribution of pressure coefficients are changed due to the morphing layer; at other angles, the pressure coefficients are not changed. In Figure 35, we find that the C_L/C_D is slightly enhanced between the working ranges of morphing layer. In our aeroacousitc simulation the blade data is input at every fifth degree azimuth as loading data. Hence, there will be 72 sets of blade data in one single revolution.



(a) with morphing layer



(b) original blade

Figure 34. Pressure coefficient on the upper blade from azimuthal angles of 0° to 360°

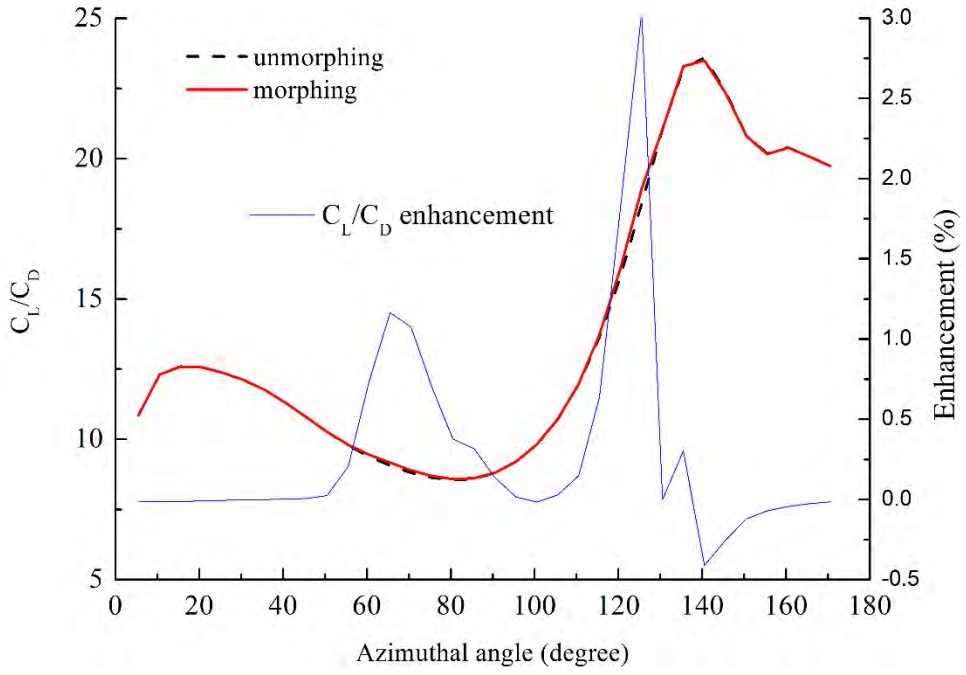


Figure 35. CL/CD enhancement using morphing surface at the advancing side (0°~180°)

In our simulation (CAA) we take two observers, one at 500 meters, and one at 20 meters. The observer is stationary and the blade rotates periodically. This set up follows what would essentially be a wind tunnel set up. The blade is given as a periodic input, with T=0.1879 seconds and the time step being 0.00261 seconds. For frequency analysis, each period is broken into 72 segments where the OSAPL is analyzed, allowing us to output the sound

pressure levels in the time history. Please refer to the namelist given in the appendix for further details.

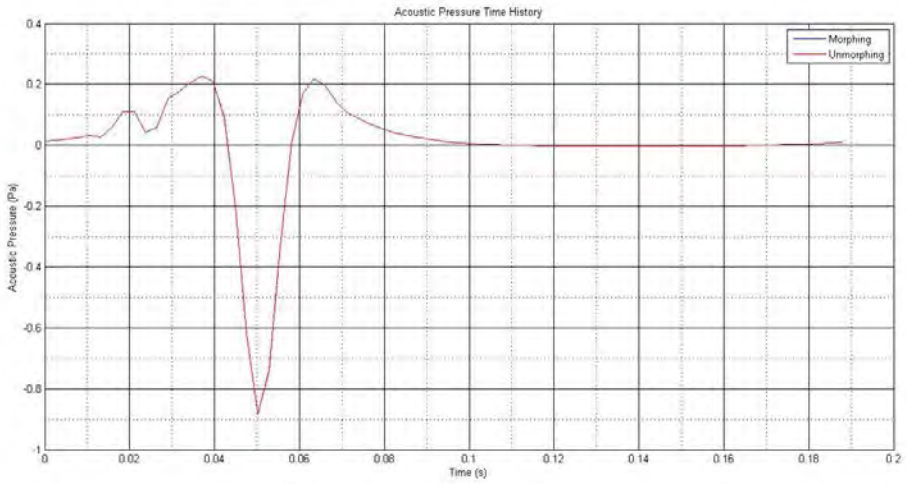


Figure 36. Acoustic pressure (Pa) throughout the time history for both the morphing and unmorphing case (Observer at 500 meters).

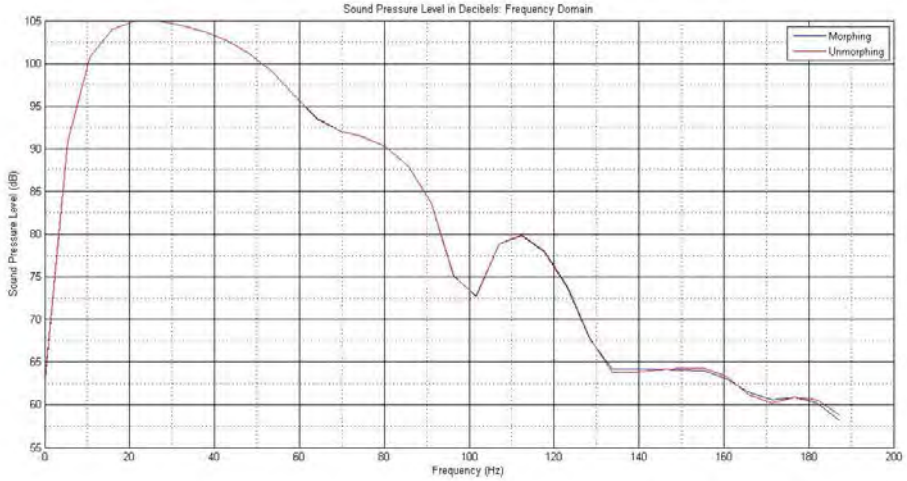


Figure 37. Sound pressure level (dB) in the Frequency domain for the morphing and unmorphing case. (Observer at 500 meters).

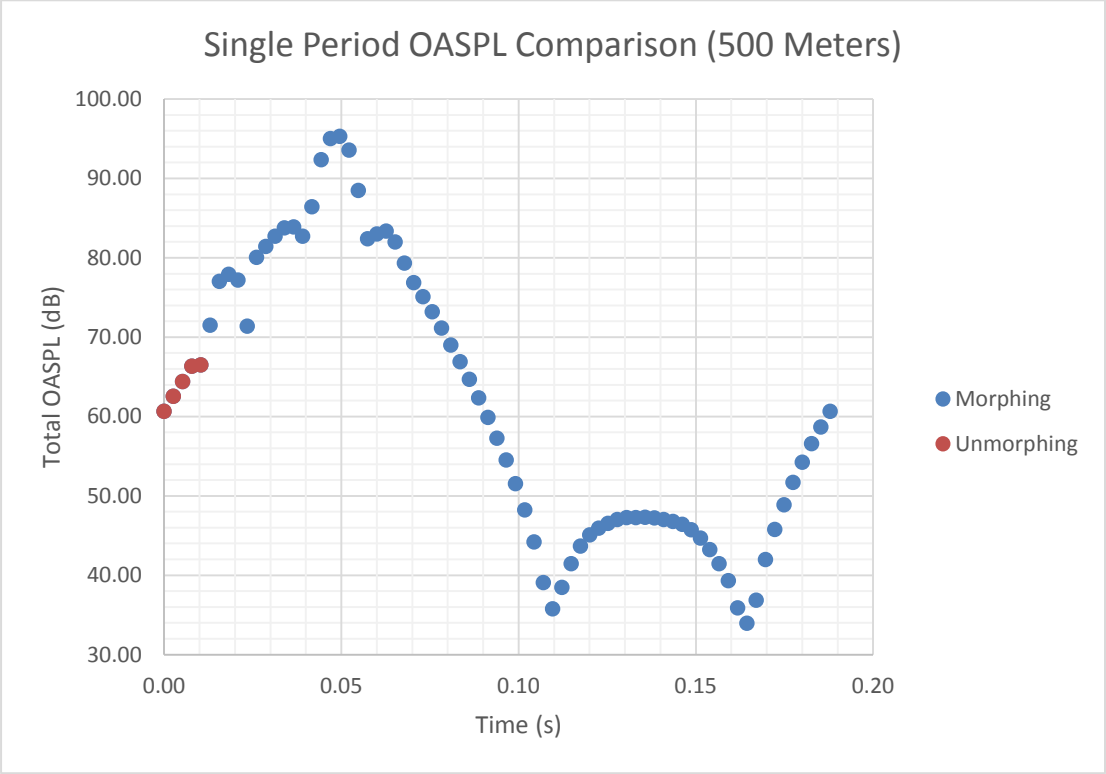


Figure 38. Total overall sound pressure level, calculated in WOPWOP and represented in the observer time domain. Data shown represents one period of data (Observer at 500 meters).

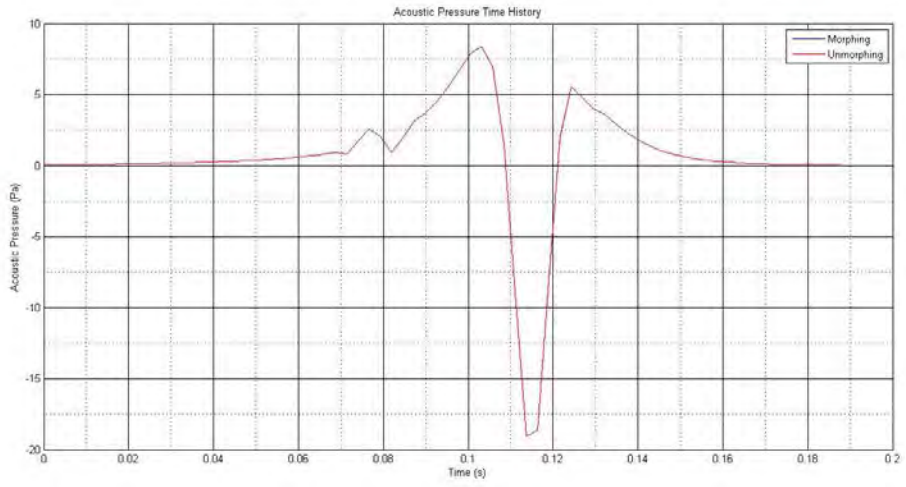


Figure 39. Acoustic pressure (Pa) throughout the time history for both the morphing and unmorphing case (Observer at 20 meters).

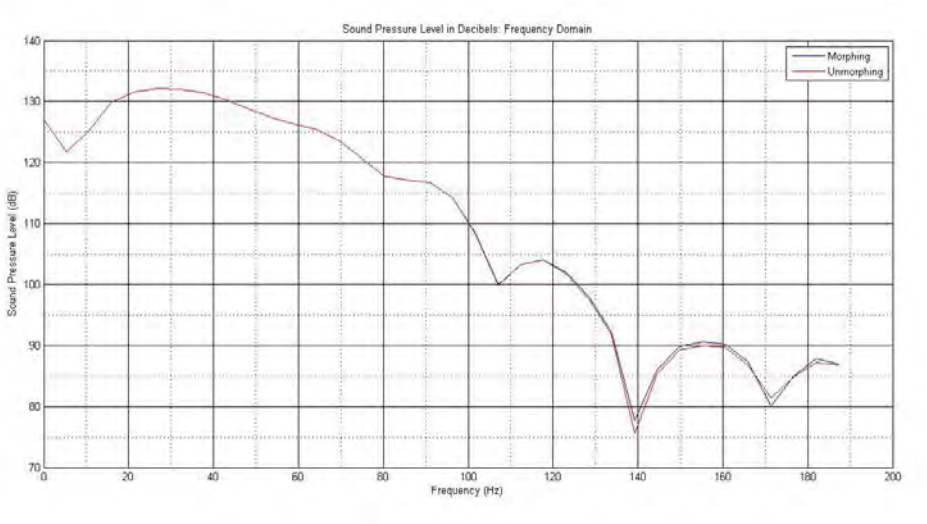


Figure 40. Sound pressure level (dB) in the Frequency domain for the morphing and unmorphing case. (Observer at 20 meters).

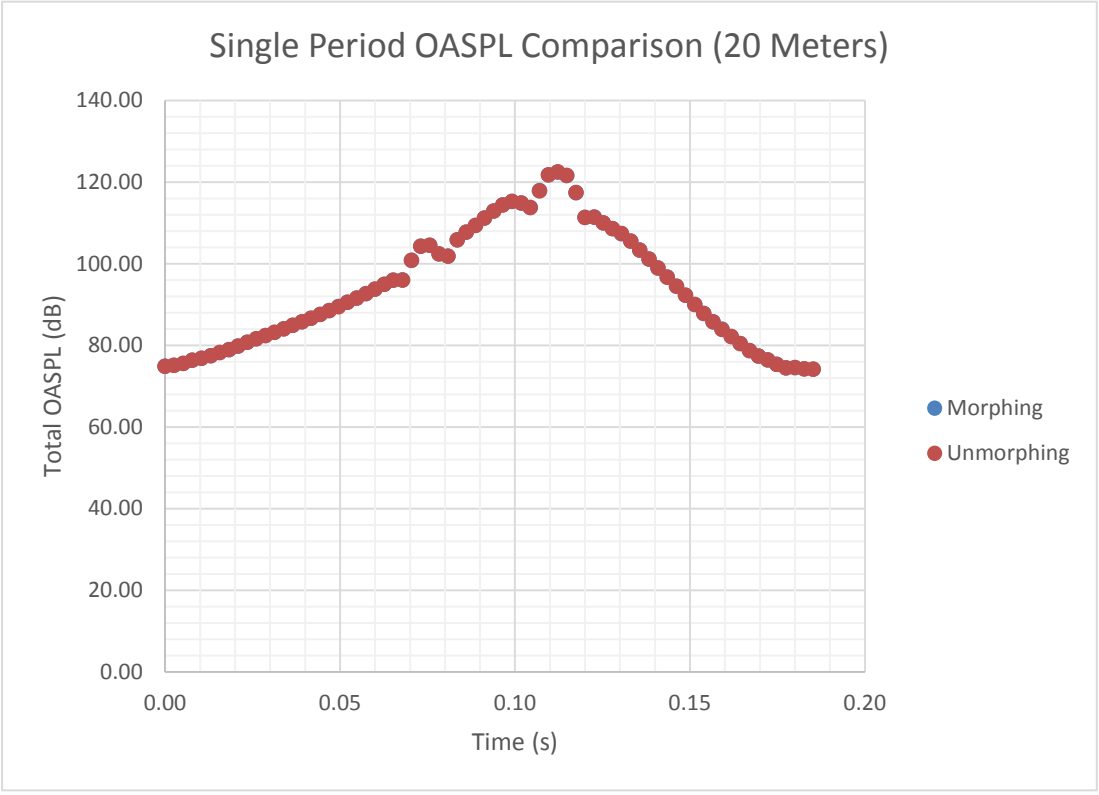


Figure 41 Total overall sound pressure level, calculated in WOPWOP and represented in the observer time domain. Data shown represents one period of data (Observer at 20meters).

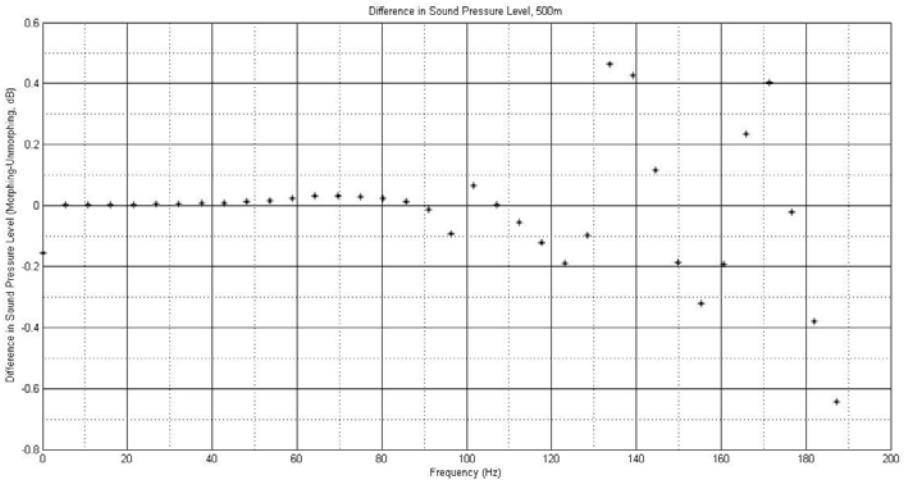


Figure 42. Difference in sound pressure level at 500 m.

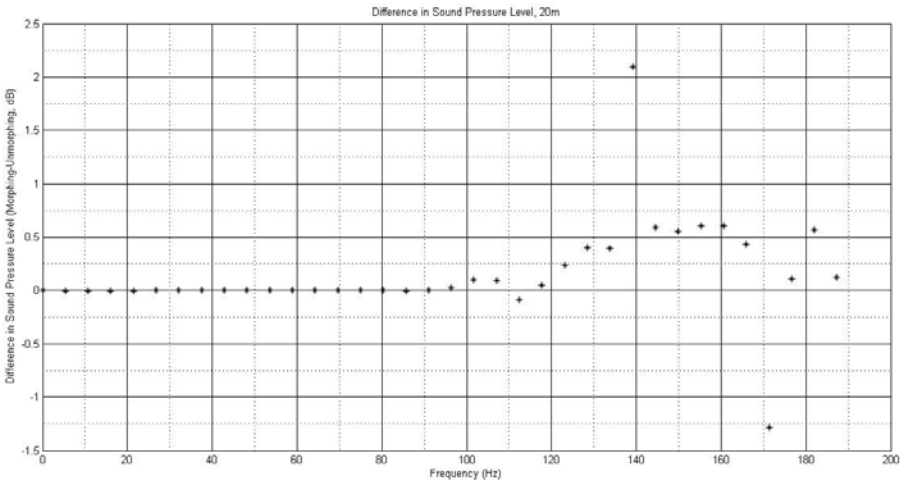


Figure 43. Difference in sound pressure level at 20 m.

Figures 36-41 show the acoustic outputs for observers at 500 meters and 20 meters. One can notice that while the acoustic pressure is of a very similar form in both cases, the sound pressure level plots are notably different.

Figures 42 and 43 show the difference in the sound pressure levels between the morphing and unmorphing blade ($SPL_{morph} - SPL_{unmorph}$) in dB. It can be seen at the difference is very slight at 500 meters, with the maximum difference being about 0.47 dB. A slightly larger negative difference of about -0.64 dB can also be seen at 187Hz. At 20 meters, however, there is a much more noticeable difference. It can be seen that at around 140 Hz there is a large difference between the morphing and unmorphing blades of about 2.09 dB. There is also a large negative difference at 170 Hz, equal to ~-1.25 dB. In this set up, a negative difference is one where the morphing blade is quieter than the unmorphing blade, while a positive difference indicates the morphing blade is louder.

In summary, a notable difference in the sound pressure level of the morphing blade can be seen, however, it is not quieter, in fact, the sound pressure level increases and decreases at different frequencies. The largest difference is around 2 dB. In the case of the immediate local flow there may be some change, however, at the relatively far permeable surface it is likely

that these changes are miniscule in comparison to the total shock effect. Additionally, we do not know if this sound profile is highly representative of the actual noise generated by the rotor flow, since these simulation only accounts for a single blade. Different affects may be seen in multi rotor flows due to the presence of blade tip vortices and BVI noise. We are, however, fully confident that these acoustics have been generated accurately using the coupling code developed in this work. We have hence created a scheme which allows for CFD-CSD-CAA coupling to be performed on rotor flows (or blade flows), which may be a powerful tool for future rotorcraft analysis/design.

m) Development of FSIMATE2

FSIMATE2 is developed to couple an alternative CSD solver: an open-source 3D FEM software, Calculix, with the GSM_CFD solver. For the CFD part, at each node along the wet surface of solids, the normal pressure force and tangential viscous force (only available in viscous flow) migrates to the CSD solver as inputs of external forces. For the CSD part, the deformed geometry coordinates of the wet surface are projected back to the CFD solver. A two-dimensional FSI case is taken as an example here.

The unstructured mesh of a two dimensional SC 1095 airfoil is shown in Figure 44. This compressible flow is analyzed at Mach number (Ma) = 0.75, Reynolds number (Re) = 11.7975M and angle of attack (α) = 2.5. A shock is observed at the upper surface of the airfoil. Both the pressure and Mach number contour, showed in Figures 45 and 46, indicate the shock location. A deformable bump locates at the upper surface of the airfoil and the quadrilateral mesh of it is shown in Figure 47. The upper surface of the bump is exposed to the pressure of the flow field and hence is regarded as the FSI interface. It experience a uniform pressure P = 57550 Pa at the lower surface. The Young's modulus (E) is 2E9 and Possion's ratio is 0.3. The left and right boundaries are fixed in space. A small deformation can be observed due to the almost identical pressure and area of upper and lower surface. The displacement at y direction and Von Mises stress are displayed in Figures 48 and 49, respectively. The concentrate stresses are at the vicinity of the fixed boundary and mid-point of the arch.

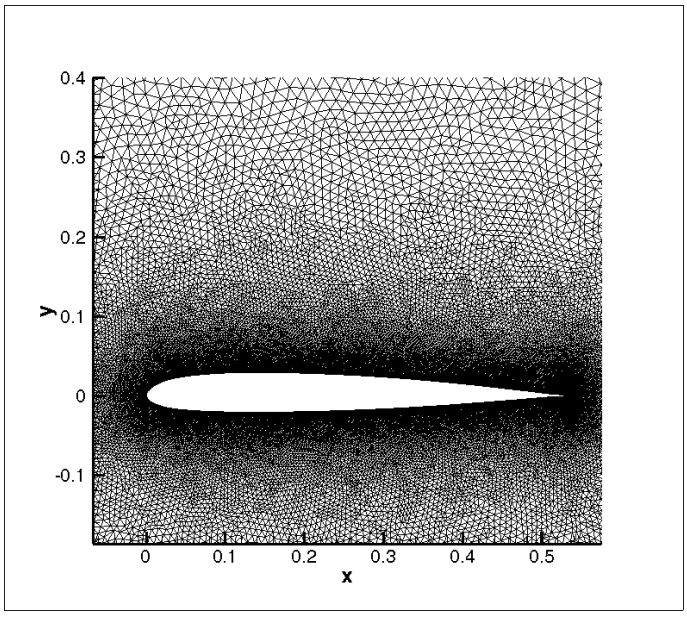


Figure 44. Unstructured mesh

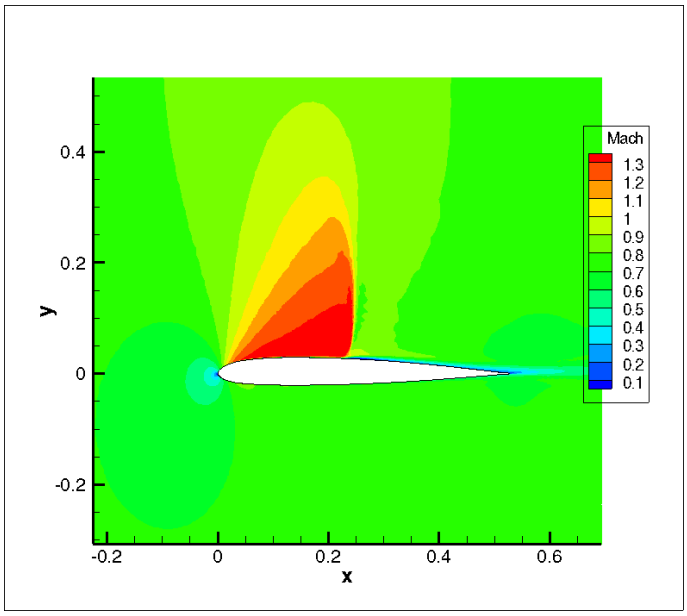


Figure 45. Mach contour at steady flow

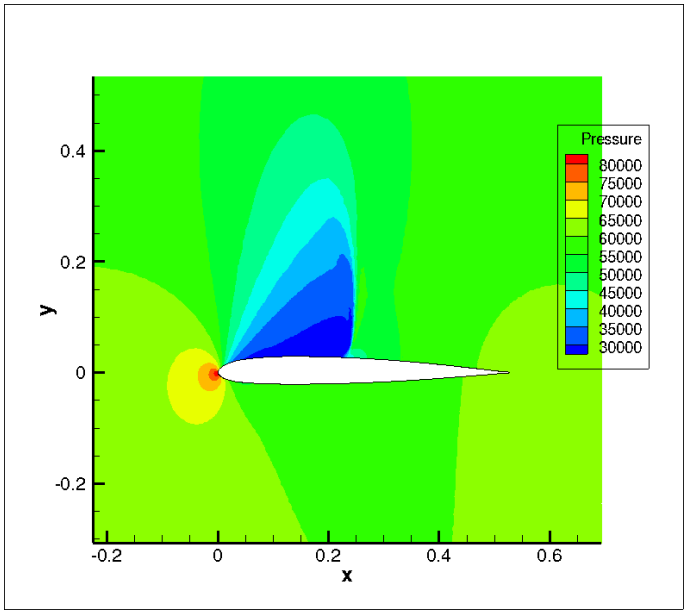


Figure 46. Pressure contour at steady flow

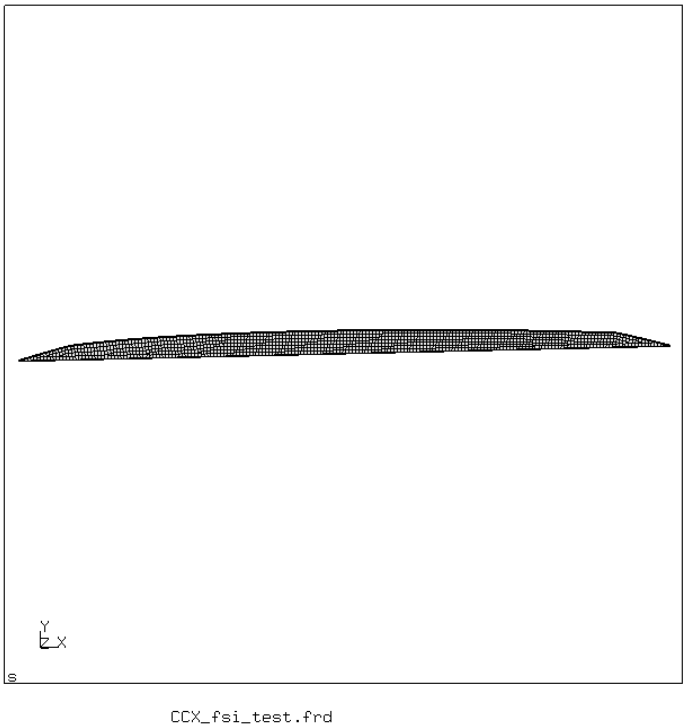


Figure 47. Quadrilateral mesh of the bump structure.

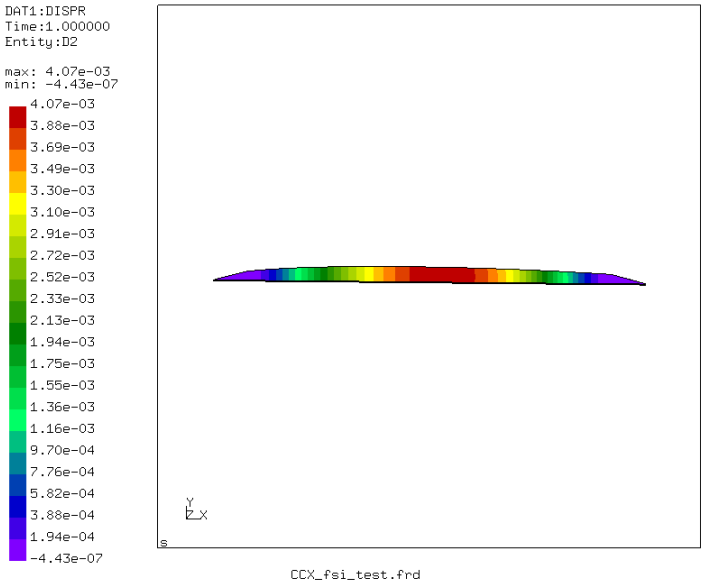


Figure 48. Displacement at y direction

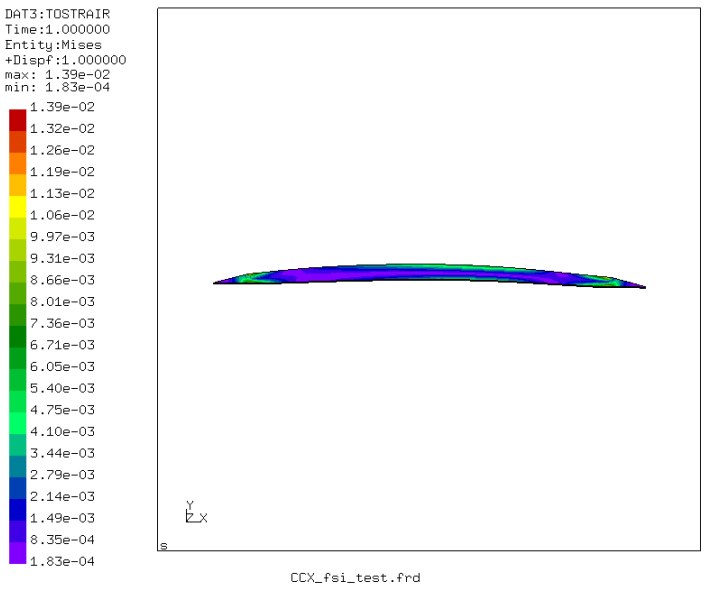


Figure 49. Von Mises stresss distribution on deformed shape.

Bibliography

[1] Jianyao Yao, G. R. Liu, D. Qian, C.L. Chen, and G. X. Xu, “A moving-mesh gradient smoothing method for compressible CFD problems,” Mathematical Models and Methods in Applied Sciences, Vol.23, No.2, 273-305, February 2013.

[2] Jianyao Yao, G.R. Liu. “A matrix-form GSM–CFD solver for incompressible fluids and its application to hemodynamics”, Computational Mechanics, 2014.

[3] Junxiang Shi, Tiancheng Xu, Steven R. Schafer, C.L. Chen, Adaptive Control of Shock wave with Passively Morphing Layer for a Rotating Blade, Journal of Aerospace Engineering, 2014, pp. 04014070

[4] Junxiang Shi, Jingwen Hu, Steven S. Schafer, C.L. Chen, "Numerical study of heat transfer enhancement of channel via vortex-induced vibration." Applied Thermal Engineering 70(1), 2014, 838-845.

[5] Jianyao Yao, Tao Lin, G. R. Liu, C. L. Chen. An adaptive GSM-CFD solver and its application to shock-wave boundary layer interaction. International Journal of Numerical Methods for Heat and Fluid Flow, 2014

[6] Jianyao Yao, Tao Lin, G. R. Liu, C. L. Chen, "A comparative study of time-marching schemes for fluid-structure interactions", Vibroengineering PROCEDIA, Vol. 4, 2014, p. 170-175.

[7] Junxiang Shi, Jingwen Hu, Steven S. Schafer, C. L. Chen, “Heat transfer enhancement of channel flow via vortex-induced vibration of flexible cylinder”, ASME 2014 Fluids Engineering Summer Meeting: August 3-7, 2014, Chicago, USA.

[8] Junxiang Shi, Steven S. Schafer, Chung-Lung C. L. Chen, “Vortex-induced vibration for heat transfer enhancement”, ASME 2014 International Mechanical Engineering Congress & Exposition: November 14-20, 2014, Montreal, Canada.

Appendix

a) GSM-CFD solver

The GSM-CFD solver is based on an open-source CFD code named SU2 and Liu's GSM theory. SU2 is written in C++ and GSM-CFD solver adds the gradient smoothing method (GSM) into it. A new GSM geometry class is created, and instead of using the Green-Gauss or weighted least squares method in gradient computing, a two-point node-based gradient smoothing method (nGSM) is used in GSM-CFD. In the computation of flux across the cell edges, a new edge-based gradient smoothing method (EGSM) is available.
The GSM method can be triggered by setting the configuration file as follows:

NUM_METHOD_GRAM = NGSM
NUM_METHOD_GRAM_VISC = EGSM_VISC
CONV_NUM_METHOD_FLOW=ROE-2POINET_1ST_ORDER (or ROE-2POINET_2ND_ORDER)
VISC_NUM_METHOD_FLOW = GSM

Tips: EGSM_VISC option is not compatible with Menter's shear stress transport (SST) turbulence method. Hence when the value of KIND_TURB_MODEL is SST, other viscous numerical methods should be used here instead of EGSM_VISC.

b) FSIMATE Manual

The FSIMATE code is written in Python to solve a fluid—structure—interaction problem for rotor blades. This code is mainly used to call different solvers and communicate the data between these solvers. Three solvers are included in the framework: computational fluid dynamics (CFD) solver, computational structural dynamics (CSD) solver, and mesh deformation solver. The communication between the solvers is based on ASCII files.

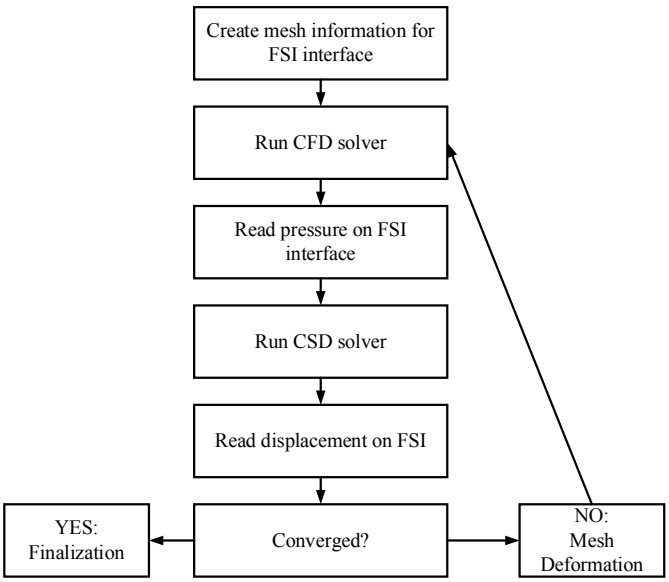


Figure 50. Flow chart of FSIMATE solver

The flow chart is shown in Figure 50.

1) Fsimate.py reads the fsi.cfg that includes all the setup parameter (Detailed comment statements can be found in the file.) and sets up all of the configuration.
2) Fsimate.py reads the cfd grid file and creates the mesh information for the FSI interface, writes to original_cfdnode.txt (global node index) and original_cfdloc.txt (initial coordinates of nodes). This process is very time consuming if the grid file is very large. For example, if there are 5 million elements, this process will cost several hours. For each case, this process only runs once. If it is a restart case, fsimate.py will read the mesh information file.
3) Fsimate.py runs the CFD solver. When it is done, fsimate.py reads the output from the CFD solver and write the pressure load in cfd_pressure.txt.
4) Fsimate.py runs the CSD solver. The pressure and displacement interpolations are processed in the CSD solver. When it is done, fsimate.py reads the output and writes the displacement in external_mesh_motion_total.txt.
5) Fsimate.py calculates the residual between two continuous iterations to determine if the converge criterion is meet. If it is, the results are finalized, if not, it calls the MDC solver to deform the mesh and then jump to step 3). Presently, the criterion is that the difference of the maximum displacement of the morphing layer between two continuous iterations must be less than 10 μm.

It should be noted that it is expected for the user to have the required additional Python libraries installed, such as NumPy and Scipy. The commercial software ANSYS with APDL module is also necessary. To run each case, make sure the required files: grid file, APDL script, bump geometry file, fsi.cfg are located in the folder, then input "python fsimate.py".

Files structure

FSIMATE code includes two files: fsimate.py and fsimate_lib.py. Fsimate_lib.py has all the functions used by fsimate.py. Fsimate.py is the main program.

Python code

Fsimate.py

Fsimate_lib.py

External executable files or script

SU2_PRT — Mesh partition

SU2_CFD — CFD solver

SU2_SOL — Finalize solution

SU2_MDC — Mesh deform

Runfem_3d/Runfem_2d — CSD solver called by APDL script

Input file to fsimate.py

Fsi.cfg — Case configuration

Rot_SC1095.cfg — CFD configuration for SU2 package (SU2_CFD, SU2_PRT, SU2_SOL, SU2_MDC)

Bump.x_t — Geometry file of morphing layer for APDL script

Grid.su2 — CFD grid file

Permeablesurface.txt — Element connectivity of permeable surface for WOPWOP

Output file by fsimate.py and external executable files

Fsi_log.txt — Main log file, information of each sub-step or iteration are recorded in this file.

SU2_CFD.out, SU2_MDC.out, SU2_SOL.out, SU2_PRT.out — SU2 package output if quite mode is on

Runfem.out — CSD solver output

Fsi_MDC_xxxxx_xx.su2 — Deformed mesh, xxxxx denotes the time step number, and xx denotes the parallel grid number. For example, if the time step is 620, the number of parallel cores is 12, there will be 12 files named FSI_MDC_00620_01.su2, FSI_MDC_00620_02.su2, up to FSI_MDC_00620_12.su2.

Fsiload_*.csv, surface_flow_*.csv — Pressure on fsi interface and impermeable surface output by SU2_CFD. It is office excel format.

Original_cfdloc.txt, Original_cfdnode.txt — Node coordinates and connectivity on fsi interface

Cfdloc.txt, cfdnode.txt, CFDpressure.txt — CFD pressure and node information to CSD solver

External_mesh_motion_total_*.txt, New_cor_*.txt — Mesh displacement and new coordinates of fsi interface

csdfacedisp.txt, csdfaceloc.txt, csdloc.txt, csdsolver*, bump*, file* — Temporary files by APDL script

Config_*.cfg — Temporary SU2 configure files for each time step

Flow*.dat, surface_flow*.dat — Finalized flow field in TECPLOT format

Flow*.vtk, surface_flow*.vtk — Finalized flow field in PARAVIEW format

History*.csv History*.plt — CFD output for each time step including CL, CD

Restart_flow_*.dat — Restart data file for continuing calculation of SU2_CFD

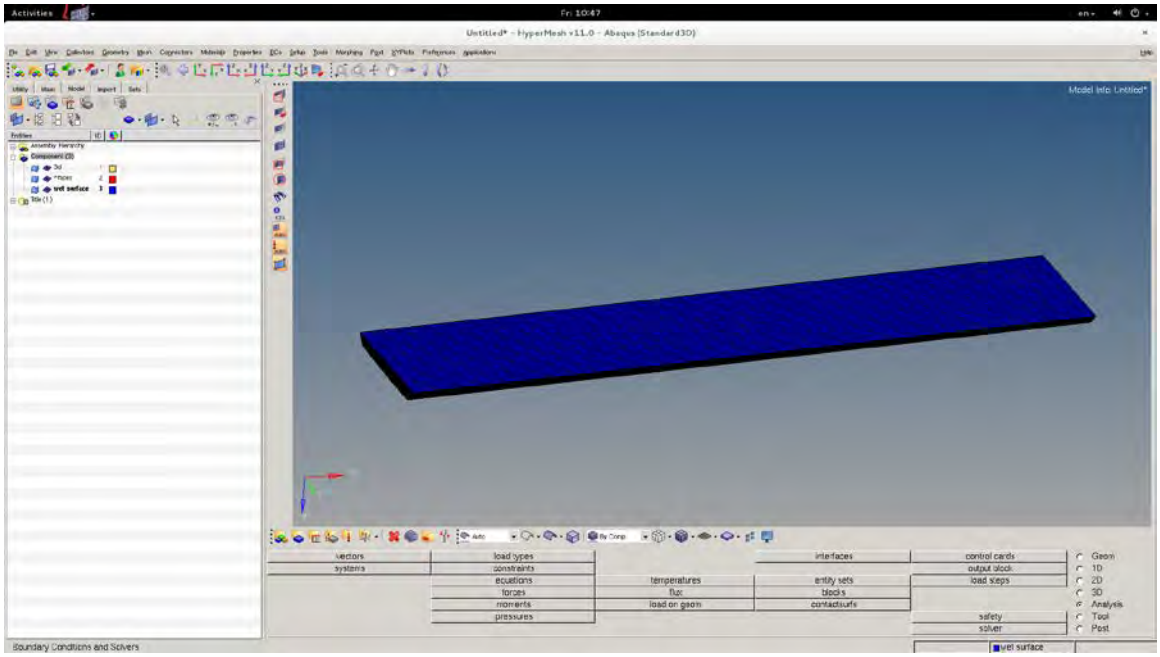
c) FSIMATE2 Manual

FSIMATE2 is an extension of FSIMATE. In FSIMATE2, Calculix is used as the CSD solver. To use FSIMATE2, Calculix is required.

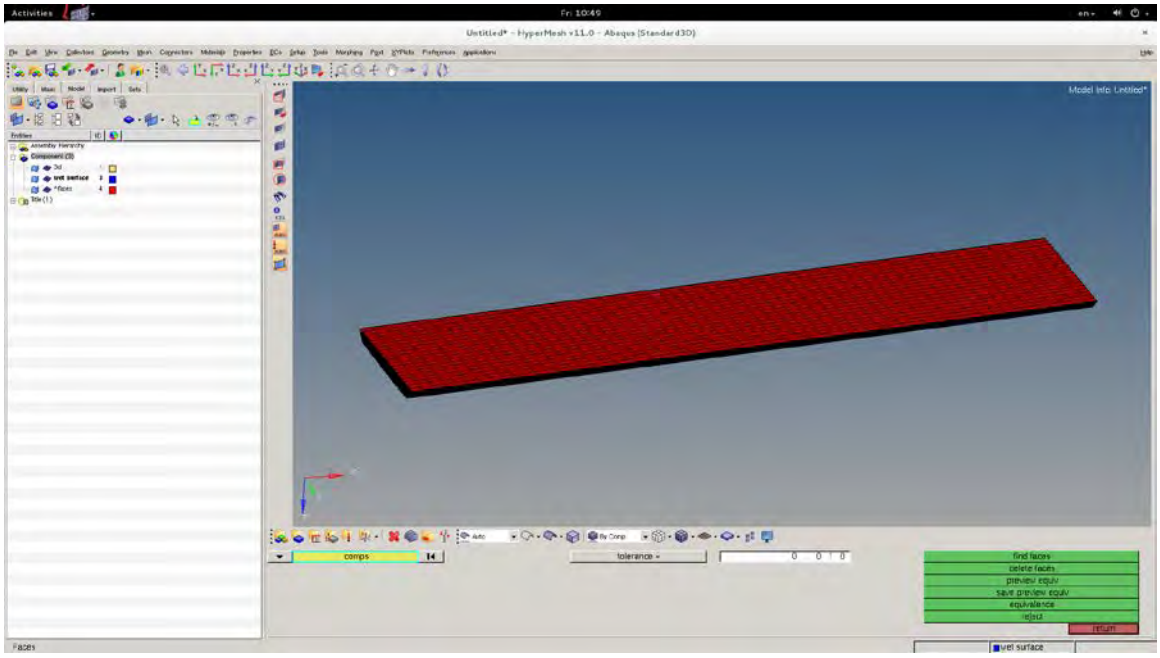
Calculix setup

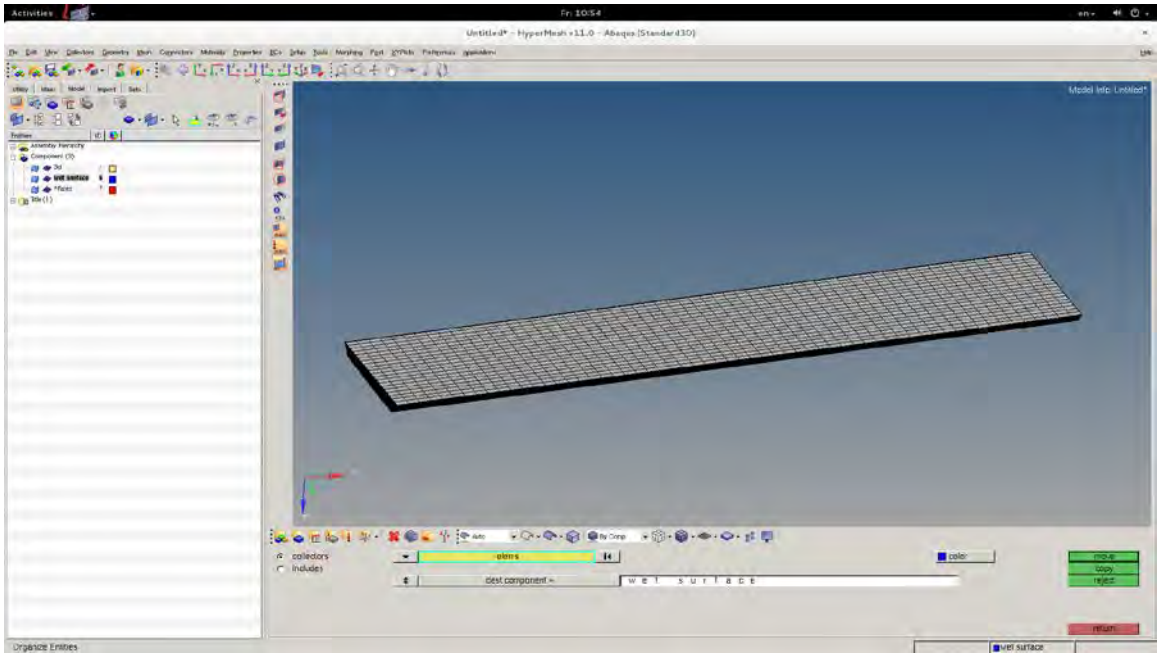
Calculix is a free three-dimensional structural finite element software. The input file is quite similar to that of the renowned commercial structural finite element software: Abaqus. It only requires some minor modifications to the Abaqus input file. The Calculix input file can be created from scratch or more favorably from Abaqus or a mesh tool Hypermesh. The detailed documentation for Calculix can be found on its website: http://www.dhondt.de/ccx_2.8p2.pdf. In order to couple with the GSM-CFD solver, a surface mesh file (faces for 3D and edges for 2D) for wet surfaces of the solids needs to be extracted from the solid mesh file. And the surface mesh file needs to be converted to the su2 format, which is described in the next section. The way of extraction is illustrated below:

Take a 3d solid as example, Hypermesh serves as mesh tool here:



Set the solver deck as Abaqus Standard 3D. Firstly, in the lower section, click faces under Tool. The lower section will appears as follows: select the solid and click find faces. Then all surfaces except the wet surface should be removed in the faces component. In the tool bar, click organize. Select all of the elements on the wet surface and move it to a newly created component named wet surface.





Afterwards, export the solver deck using Abaqus 3D. An abaqus input file will be generated. In the input file, the coordinates for the points and element data for the wet surface need to be transferred to another file with an extension named mesh (e.g. bump.mesh)

Finally, the surface mesh file needs to be modified to follow the su2 mesh format. Remove all the comment lines in the surface mesh file. The mesh file is subdivided into two sections: the point coordinates and the element connectivity. In front of the point coordinates, the dimension of the problem and the number of points are required to be specified. This header can be written as:

NDIME=3
NPOIN=1355

Where NDIME is the value of dimension and NPOIN is the number of points. When it comes to the section of element connectivity, it is necessary to preset the number of elements and vtk type of elements, which can be written as:

NELEM=169
3

Where NELEM gives the number of elements and the value of vtk type of elements immediately appears at the next line. The vtk type of elements is shown in the Table 1.

Table 1

Element type	VTK value
Line	3
Triangular	5
Quadrilateral	9

FSIMATE2

FSIMATE2.py is used as the data exchanger and controller for the FSI process. And an

executable file named SU2_FSI serves as the interpolation module of the corresponding surfaces of solid and fluid zone. The SU2_FSI uses its own configure file.

```
#Inputs declaration
restart = False
output_format = "TECPLOT"
unsteady_scheme = "DUAL_TIME_STEPPING-1ST_ORDER"
fsi_marker = "Bump"
fsi_interface_file = "surface_fsi"
structure_load_file = "bump_load.dat"
displacement_file = "motion.dat"

solid_solver = "ccx_2.7 "
original_grid = "grid.su2"
totalstep = 100
maxitertime = 2

Config_MDC_file  = "MDC_fsi_test.cfg"
Config_CFD_file  = "CFD_fsi_test.cfg"
Config_FSI_file  = "FSI_fsi_test.cfg"
Config_CCX_file  = "CCX_fsi_test"
Mesh_MDC_file    = "MDC_meshout.su2"

run_MDC_file = "run_" + Config_MDC_file
run_CFD_file = "run_" + Config_CFD_file
run_FSI_file = "run_" + Config_FSI_file
rst_file = Config_CCX_file + ".frd"
out_file = "initial.dat"
```

fsi_marker: boundary marker for the wet surface in solid.
Fsi_interface: the file contains all the nodal pressure and viscous force on the FSI interface.
Totalstep: the specified total steps in FSI process for dynamic problem
Maxitertime: the specified local steps for fluid solver during one FSI step.
All options beginning with "Config_" are the input configure file, but only for reference. They reserve all the comments and act as base files. And options beginning with "run_" are the runtime configure file. They are duplicated ones from the base files and abandon all the comments.

d) Aeroacoustic analysis using WOPWOP

Introduction

At high speed forward flight rotorcraft blades will experience numerous different flow conditions, most notably, transonic shock flow along the advancing azimuths and dynamic stall along the retreating azimuths. This stark variation in flow conditions renders many passive flow control methods useless for rotorcraft applications, as the flow control method would degrade the aircraft's performance at any off design flow conditions. However, the morphing bump has been seen to be effective through both transonic shock and dynamic stall regions because of its design, which constrains the presence of the bump to regions where it can be of use. As such, it is proposed as a hopeful candidate for application as a flow control method on rotorcraft.

The majority of the noise generated by rotorcraft blades comes in the form of HSI noise caused during the transonic flow regime experience by the blade throughout the advancing azimuths (0°-180°) by the presence of strong transonic shock waves. However, since the morphing layer affects this flight regime, specifically in terms of shock formation and

strength, it is intuitive to believe that this manipulation of the shock may bring about a change in the acoustic profile of the blade. This could result in decreased sound levels or a drastic alteration in the sound profile of the aircraft, which is an important property concerning the military nature of the SC 1095 blade (which constitutes the majority of the blade geometry used in the UH-60 helicopter).

The field of Aeroacoustics is rooted in the Lighthill acoustic analogy, which is a rearranged set of generalized Navier-Stokes equations. These equations allow for an inhomogeneous wave operator, which allows the user to solve for acoustic variables (i.e. acoustic pressure). This analogy was taken one step further by Ffwocs-Williams and Hawkings, and Farrasat, who have since proposed solutions and numerical methods which allow for more explicit solutions to the Lighthill analogy. Their equations allow for monopole, dipole, and quadrupole source terms to be solved. Monopole and dipole source terms represent the surface source terms, which account for noise generated by the displacement of the fluid by the surface, and by the aerodynamic loading on the surface. Quadrupole source terms are represented as volume integrals, and are typically ignored due to the high computational demands required to solve the volume integral. However, in the case of HSI noise, which is generated due to the presence of shock waves, the quadrupole source term is very important. Hence, a permeable surface must be used. A permeable surface allows for interpolation of all of the noise terms within the surface to be made to the surface itself. The permeable surface permits these noise sources to pass through the surface, unlike an impermeable surface, where the surface itself is the only noise source. Permeable surfaces are more difficult to use, and have special requirements that make their use only valuable when HSI noise is to be considered.

Process

PSU-WOPWOP is a command line style program, which requires a decent background in programming and CL interface to ensure it is used to its full potential. As it is only a CAA solver, and at the lowest level requires inputs for at least the geometry (to solve the thickness noise), and at the most requires a full set of aerodynamic solution variables (in a permeable case), it is always coupled with at least one other program, typically a CFD solver or geometry modeling software. With this in mind, our first goal was to create a working converter program that would allow us to couple PSU-WOPWOP with our CFD outputs from SU2.

The current version of the converter program allows the user to specify flow field file inputs which are then converted into an appropriate WOPWOP input file. The user may specify numerous inputs, and the converter will use these to calculate the number of data surfaces, time steps, time step size, and reference frame. The end result of the converter is a properly formatted WOPWOP input file. Our end goal with this is to create a coupling program that would allow the user to perform a fully coupled CSD-CFD-CAA simulation, assuming they have all of the required solvers (CSD-CFD coupling is handled with FSIMATE). This would be an incredibly powerful tool concerning rotorcraft design, as one could run a full simulation yielding CFD, FSI, and CAA results, which would allow for a nearly full examination of the aircraft's performance. The basic structure of our converter is as follows:

1. The converter reads in and organizes all of the flow data and grid input files, and stores them for use later in the conversion.

2. The normal vectors of the various patches are computed and stored to later be written into the WOPWOP input files.
3. The geometry file is written in WOPWOP format according to the case's needs.
4. The loading file is written in WOPWOP format.

The current converter can create loading and geometry input files in all reference frames, for permeable or non-permeable cases. It is also functional for period, constant, and aperiodic loading data.

The WOPWOP Converter program was written at the University of Missouri by Zachary Lipira and Steve Schafer under the guidance of Dr. Junxiang Shi and Dr. C.L. Chen. This document provides a complete overview of the program that was created to read text files and convert data into binary input files for PSU-WOPWOP_V3.4.2B3308, a computational aeroacoustics solver created by Penn State University. The WOPWOP Converter program was written in compliance with WOPWOP3_User_Manual.pdf and tested for multiple cases to make sure it correctly adhered to WOPWOP's specific input structure.

e) WOPWOP_Converter_2.0

WOWPOP_Converter_2.0 is the second edition of the converter program written in Python. Most of the changes occurred in the fall of 2014 and spring of 2015 and were intended to make the converter more generic, fix key issues, and to add new features. Descriptions of the features and examples of user inputs and configuration files will follow.

The converter is designed to write WOPWOP input files that contain flow data on permeable surfaces. Once again, the geometry does not have to contain permeable surfaces, but the converter expects to read and write flow data that specify the conservation variables along the surfaces. In addition to using the readflow function in reaflow.py to read ".dat" files, WOPWOP_Converter_2.0 is also capable of reading ".vtk" files by using the readvtk function in readvtk.py. It is assumed that the flow data files were written by SU2 and that the related variables needed to be used as input for WOPWOP are called "x", "y", "z", "Grid_Velx", "Grid_Vely", "Grid_Velz", "Pressure", and "Conservative_1" in the flow data files. The density of the flow field is known as "Conservative_1". In contrast to the original converter, WOPWOP_Converter_2.0 can select which flow data files will be read out of a folder according to user input which will be covered in the following sections.

The converter also expects the data that defines the mesh connectivity along all of the surfaces to be located in one file. The readmesh function in readmesh.py is called just once in order to read the mesh file that defines the surface connectivity. WOPWOP does not require flow data or geometry data for the far field from the CFD simulation. The readmesh function will automatically skip over any boundary that is called "FARFIELD". Therefore, it is up to the user to title their far field boundary as "FARFIELD" or to manually adjust the name of the far field boundary in readmesh.py (Line 75) so that it matches their boundary name.

For debugging purposes, WOPWOP will output files that can be used with FieldView. This requires the connectivity of the nodes to be written in a specific format so that FieldView can properly model the geometry. The function connectivity in connectivity.py is designed to fulfill this purpose, allowing the user to validate their results from WOWPOP.

Another new feature is that the user_input.py file was replaced by append_case.py and new_case.py. The append_case.py file is used to get a user's inputs for flow data files that

need to be appended to already existing WOPWOP input files. The new_case.py gets a user's inputs for flow data files that need to be written to new WOPWOP input files. Examples of both cases will be covered in section 1.2.

It is the job of the files geo_unst_header.py, geo_write.py, load_unst_header.py, and load_write.py to actually write all of the data to the files specified by the user. Once the data from a flow data file is read and then written to a file in binary, the flow data is cleared from memory in order to conserve memory usage. The issue with writing input files that contain periodic data has been fixed with WOPWOP_Converter_2.0. This was adjusted in geo_unst_header.py and was noted at the top of the file. The files readvtk.py, new_case.py, and append_case.py are new files that were added to the program. The file main.py was the only other file altered in addition to geo_unst_header.py while all of the other included files were left unchanged.

Examples of Manual User Input

This section is dedicated to describing what is expected as input and to show possible examples. The user should already be familiar with WOPWOP3_User_Manual.pdf in order to fully understand the reasoning behind some of the inputs for the converter. An example input file named "Example of Manual User Input.docx" is located in the folder "Run Converter with Manual User Input" and should be referenced to clarify any questions.

Once main.py is executed, the user will immediately be prompted with a series of inputs as follows:

1) The first input asks whether new geometry and loading input files need to be written for WOPWOP, if existing files need to be appended to, or both. A 1, 2, or 3 must be entered.
2) The user will then be prompted to respond with how many different cases they would like to write new files or append to files. This feature allows someone to create as many input files for WOPWOP as they would like while only having to run the code once. Be sure to enter an integer.
3) The user will then be prompted with a series of statements that require full pathnames to folders that contain the flow data files, folders where files should be written to, where the mesh connectivity file is located, or to existing files where data will be appended to. Examples that cover each of these prompts are located in the example file in lines 15, 19, 27, 34, 94, 97, 100, and 103. Be sure to end the code with "\"when prompted to.
4) When writing new input files for WOPWOP, the program will ask what to name the geometry and loading files. Be sure to use appropriate file extensions. These file names are concatenated to the end of the variables that contain the full pathname of the directory where the files should be written to. Examples that cover this kind of input are in lines 23 and 31of the example file.
5) The way permeable surfaces are input is different in WOPWOP_Converter_2.0. Instead of responding with a "y" or "n", the program will ask for a list of the permeable zones separated by a comma and then give an example. The names of the permeable zones must match up exactly with what is provided in the mesh file that describes the surface connectivity. An example is given in "Example of Manual User Input.docx" in line 42.

6) When prompted to select the data type, only enter a 1, 2, or 3. The WOPWOP user's manual goes into about the different data types and should be consulted for further clarification. Periodic data types are properly handled in WOPWOP_Converter_2.0 and if a 2 is entered to select data of type periodic, the user will be prompted to enter the period of the data in seconds.
7) The next input asks for the type of reference frame. Refer to the WOPWOP user manual for details. Be sure to enter 1, 2, or 3 only.
8) The next few inputs are problem parameters. These problem parameters should match exactly with what was input into the CFD simulation. There are default values in parentheses which can be chosen by pressing return without any other given input. If the user would like to input their own values, they must make sure the units match that given by the default values. The angular velocity of the blade and the number of degrees the blade rotates between flow data files is used to properly calculate the time steps.
9) With the new feature of being able to read ".vtk" files, the user must then select whether ".vtk" or ".dat" files should be read by entering a 1 or 2, respectively.
10) Also, with the new feature of being able to select certain flow data files from a folder, there are three new inputs which ask for the number of the first data file, the number of the last data file, and the number to increment through which flow data files will be read between the first and last. Examples are given in lines 74-77. When inputting the number of the first or last data file to be read, be sure to list all of the numbers in the file's name. It is always assumed that the flow data files starts with "flow_", then contains some numbers, and then ends with the specified file extension. For example, if "flow_00003.dat" was to be read, its number would be 00003.
11) Periodically, the program might ask if you would like to view a list of flow data files. It is expected that the user respond accordingly with either a "y" or "n" for yes or no, respectively.

Additional Notes

The converter is designed to read and write data for triangular and unstructured meshes only. In addition to that, all normal vectors are calculated and input as being node-centered. The converter has proven to work in Windows, although there have been some issues with running it under different Linux operating systems. This could be a rather cumbersome inconvenience since it was written to be coupled with SU2, an open-source program. This issue is still being looked into.

Acknowledge

We acknowledge Professor Brentner at PSU for his guidance to use WOPWOP software.