A High-Order Spectral Difference Method for Fluid-Structure Interaction on Dynamic Deforming Meshes

K. Ou* and A. Jameson †

Aeronautics and Astronautics Department, Stanford University, Stanford, CA 94305

In previous work by the authors, Spectral Difference Method has been formulated in a framework with time-dependent moving deformable meshes. The framework has also been shown to preserve the design accuracy of the underlying high order temporal (explicit Runge-Kutta) and spatial discretization (Spectral Difference) methods. In this study, Spectral Difference method is further extended to deal with fluid-structure interaction problems in a dynamic grid-deforming framework. In particular, we address the technique for blending the mesh in the presence of moving and deforming physical boundaries, the issue of fluid-structure coupling, and the formulation of the structural model. Flows over plunging and pitching airfoils are solved on deforming meshes, and the results are validated with experimental studies as well as previous numerical results with rigidly moving meshes. The solver is then applied to solve a fluid structure interaction problem, which involves an elastic beam interacting with a cylinder vortical wake. Finally, we consider the case of a flow over a finite mass cylinder that is free to oscillate in the cross-flow direction, and examine the dynamics of the flow induced cylinder movement and oscillation.

I. Introduction

High order methods have recently emerged as attractive tools for producing highly accurate results with minimum numerical dissipation. Discontinuous Galerkin (DG)¹⁻⁵ is among the most popular and widely implemented high order method. Other high order methods with the features of being potentially more efficient and easier to implement are emerging. In particular, we focus on spectral difference (SD) method in this paper. Further details on SD method can be found in,⁶⁻¹² while various applications of SD method can be found in.¹³⁻¹⁸

While most of the high order methods applications with DG as well as SD have been concentrated on steady state problems on fixed meshes, recent work has seen the extension of the high order methods to time accurate unsteady problem on dynamic meshes. Spatially and temporally high order DG and SD methods on unstructured dynamic moving deforming meshes have recently been formulated and implemented successfully.¹⁹⁻²²

Accurate aeroelastic simulation has been a subject of ongoing research, with important relevance for many practical engineering problems ranging from blood vessels, bridges, to fighter aircrafts. While DG method has been applied to fluid structure interaction problems,²³ there is relatively little work done in devising high order aeroelastic SD solvers. Building upon the grid-deforming framework of the spatial and temporal high order SD solver,²⁴ we present in this study the SD formulation for the Navier-Stokes equations coupled with structure finite element model for solving the fluid structure interaction on dynamic deforming meshes.

To formulate the SD method for aeroelastic coupling, a strategy for handling mesh and boundary movement need to be devised. In this paper, we review the mesh deforming and blending techniques used for solving the conservation laws on distorted meshes. For arbitrary elastic boundary deformation, which is usually the case for elastic structure elements, we implement the method used by.^{23,29} Using the proposed method, a cubic spline with clamped edges is used for fitting boundary displacements along the structure

*PhD Candidate, Aeronautics and Astronautics Department, Stanford University, AIAA Student Member.

†Thomas V Jones Professor, Aeronautics and Astronautics Department, Stanford University, AIAA Fellow.

element, entailing smooth mesh deformation between the structure and the far field boundary of the flow domain. For the structure solver, we consider a finite element model of a beam based on the theory of elasticity.³¹ The coupling of the fluid solver and structure solver is handled by using a multi-stage explicit Runge-Kutta scheme, without additional inner iteration. At each RK stage, fluid force is passed as an external distributed load to the structure model. The finite element solver subsequently returns the boundary displacements and velocities, which are propagated into the flow in the form of mesh disturbances, represented by the deforming mesh metrics and modulus. To ensure close fluid structure coupling, we use the fourth order, strong stability preserving, five-stage RK4 method.

The rest of the paper is organized as follows: we start by revisiting the method for solving the transformed conservation laws on unsteadily deforming meshes. This is then followed by the formulation of SD method in such a transformed space. Mesh deforming and blending techniques are introduced next, covering both the cases of rigidly moving boundary and elastic deforming boundary. We then present the formulation of the finite element model based on the elastic beam theory. Finally, we demonstrate the solver with a couple of applications in the results section.

II. Steady and Unsteady Coordinate Transformation

Consider two domains, with one being an arbitrary domain in physical space, and the other a cartesian domain in computational space. A transformation, either analytic or numerical, can be formulated to map one domain into the other. The same transformation function is used to transform the governing equations for use in the new computational coordinate system. Depending on whether the two domains are moving relatively to each other, the coordinate transformation can be further classified into steady transformation and unsteady transformation. For completeness, both have been outlined below.

II.A. Steady Coordinate Transformation

For the steady transformation, consider the transformation function $\mathbf{T}_s$, that maps the (X, Y) coordinates in reference space to the (ξ, η) coordinates in computational space:

$$(\xi, \eta) = \mathbf{T}_s(X, Y) \quad (1)$$

and we have:

$$\xi = \xi(X, Y), \quad \eta = \eta(X, Y)$$

Using the chain rule to arrive at the transformation gradient as:

$$\mathbf{G}_{\mathbf{T}_s} = \begin{bmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} \end{bmatrix} \quad (2)$$

The Jacobian of the transformation gradient is equal to:

$$J_{T_s} = \frac{\partial X}{\partial \xi} \frac{\partial Y}{\partial \eta} - \frac{\partial X}{\partial \eta} \frac{\partial Y}{\partial \xi} \quad (3)$$

II.B. Unsteady Coordinate Transformation

When the two coordinate systems are moving relative to one another, the unsteady transformation $\mathbf{T}_u$ is now time dependent. Let's consider the unsteady coordinate transformation between the physical space in (x, y) and the reference space in (X, Y) .

$$(X, Y) = \mathbf{T}_u(x, y, t) \quad (4)$$

and we have:

$$X = X(x, y, t), \quad Y = Y(x, y, t)$$

Again using the chain rule to arrive at the unsteady transformation gradient as:

$$\mathbf{G}_{\mathbf{T}_u} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial y}{\partial X} & 0 \\ \frac{\partial x}{\partial Y} & \frac{\partial y}{\partial Y} & 0 \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & 1 \end{bmatrix} \quad (5)$$

The Jacobian of the unsteady transformation gradient is equal to:

$$J_{Tu} = \frac{\partial x}{\partial X} \frac{\partial y}{\partial Y} - \frac{\partial x}{\partial Y} \frac{\partial y}{\partial X} \quad (6)$$

III. Conservation Laws Transformations

III.A. Navier-Stokes Equation in Untransformed Physical Space

Consider the unsteady compressible 2D Navier-Stokes equations in conservative form written as:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \quad (7)$$

where the conservative variables $\mathbf{U}$ and the Cartesian components $\mathbf{F}(\mathbf{U}, \nabla \mathbf{U})$, and $\mathbf{G}(\mathbf{U}, \nabla \mathbf{U})$ of the flux, which include both the inviscid and viscous flux vectors such that $\mathbf{F} = \mathbf{F}^i + \mathbf{F}^v$ and $\mathbf{G} = \mathbf{G}^i + \mathbf{G}^v$ are given by

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix}, \quad \mathbf{F}^i = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{Bmatrix}, \quad \mathbf{G}^i = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{Bmatrix} \quad (8)$$

The Cartesian components $\mathbf{F}^v(\mathbf{U}, \nabla \mathbf{U})$ and $\mathbf{G}^v(\mathbf{U}, \nabla \mathbf{U})$ of the viscous flux vector are given by

$$\mathbf{F}^v = \mu \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} + \frac{C_p}{Pr}T_x \end{Bmatrix}, \quad \mathbf{G}^v = \mu \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ v\tau_{yy} + u\tau_{xy} + \frac{C_p}{Pr}T_y \end{Bmatrix} \quad (9)$$

Here ρ is the density, u and v are the velocity components in x and y directions, p stands for pressure and E is the total energy. The pressure is related to the total energy by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \quad (10)$$

with a constant ratio of specific heat γ . For all test cases in the present study, γ is going to be 1.4 for air. μ is the dynamic viscosity, C_p is the specific heat and Pr stands for Prandtl number. T is temperature which can be derived from the perfect gas assumption. λ is set to $-2/3$ according to the Stokes hypothesis. The stress tensor takes the following form

$$\begin{aligned} \tau_{xx} &= 2u_x + \lambda(u_x + v_y), \tau_{yx} = v_x + u_y \\ \tau_{yy} &= 2v_y + \lambda(u_x + v_y), \tau_{xy} = v_x + u_y \end{aligned} \quad (11)$$

III.B. Navier-Stokes Equations in Transformed Reference Space

In the unsteady case, using the chain rule for differentiation, and define the following new identities

$$\begin{aligned} \mathbf{U}^r &= J_{Tu} \mathbf{U}^p \\ \mathbf{F}^r &= J_{Tu} \left(\mathbf{F}^p \frac{\partial X}{\partial x} + \mathbf{G}^p \frac{\partial X}{\partial y} + \mathbf{U}^p \frac{\partial X}{\partial t} \right) \\ \mathbf{G}^r &= J_{Tu} \left(\mathbf{F}^p \frac{\partial Y}{\partial x} + \mathbf{G}^p \frac{\partial Y}{\partial y} + \mathbf{U}^p \frac{\partial Y}{\partial t} \right) \end{aligned} \quad (12)$$

The governing equation in the new reference coordinate space in the unsteady case still assumes the same conservation law form:

$$\frac{\partial \mathbf{U}^r}{\partial t} + \frac{\partial \mathbf{F}^r}{\partial X} + \frac{\partial \mathbf{G}^r}{\partial Y} = 0 \quad (13)$$

III.C. Navier-Stokes Equations in Transformed Computational Space

In the steady case, again using the chain rule for differentiation, the equations transform into the following non-conservative form

$$\frac{\partial \mathbf{U}^r}{\partial t} + \frac{\partial \mathbf{F}^r}{\partial \xi} \frac{\partial \xi}{\partial X} + \frac{\partial \mathbf{F}^r}{\partial \eta} \frac{\partial \eta}{\partial X} + \frac{\partial \mathbf{G}^r}{\partial \xi} \frac{\partial \xi}{\partial Y} + \frac{\partial \mathbf{G}^r}{\partial \eta} \frac{\partial \eta}{\partial Y} = 0 \quad (14)$$

By defining the following new identities for the steady case

$$\mathbf{U}^c = J_{Ts} \mathbf{U}^r$$

$$\mathbf{F}^c = J_{Ts} \left(\mathbf{F}^r \frac{\partial \xi}{\partial X} + \mathbf{G}^r \frac{\partial \xi}{\partial Y} \right) \quad (15)$$

$$\mathbf{G}^c = J_{Ts} \left(\mathbf{F}^r \frac{\partial \eta}{\partial X} + \mathbf{G}^r \frac{\partial \eta}{\partial Y} \right)$$

The conservation law form of the governing equation in the new computational coordinate space is

$$\frac{\partial \mathbf{U}^c}{\partial t} + \frac{\partial \mathbf{F}^c}{\partial \xi} + \frac{\partial \mathbf{G}^c}{\partial \eta} = 0 \quad (16)$$

IV. Formulation of Spectral Difference Method on Deforming Meshes

For the application of SD method to unsteady moving boundary problems, there are three coordinate systems of interest, i.e. the computational space, the reference space, and the physical space. The computational space is a cartesian domain with standard unit square elements. The reference space can be considered as the physical space at time $t = 0$ when the boundary is initially at rest and has not been displaced. The reference space is in general an unstructured quadrilateral mesh domain. Finally, the physical space is the reference space undergoing a prescribed time dependent rigid or deforming motion. We define these three domains as:

$$D_{phy} \equiv (x, y), \quad D_{ref} \equiv (X, Y), \quad \text{and} \quad D_{cmp} \equiv (\eta, \xi)$$

With three coordinate systems, we need two transformations for the computation of the unsteady flow solutions. The first one involves a stationary transformation between the fixed unstructured reference domain and the fixed cartesian computational domain. The second involves the unsteady mapping of the solutions between the fixed reference domain and the moving deforming physical domain.

$$D_{phy} \xleftrightarrow{T_u} D_{ref} \xleftrightarrow{T_s} D_{cmp}$$

Firstly, consider the stationary transformation between the reference and the computational domain. All elements in the reference domain (X, Y) are transformed into standard square elements $(0 \leq \xi \leq 1, 0 \leq \eta \leq 1)$. The transformation can be written as:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \sum_{i=1}^K \mathbf{M}_i(\xi, \eta) \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \quad (17)$$

where K is the total number of points used to define the physical element, (X_i, Y_i) are the cartesian coordinates of those points, and $M_i(\xi, \eta)$ are the shape functions. For elements with straight edges, K is equal to 4. For elements lying on curved boundaries, 8 points (four mid-edge and four corner points) can define a quadratic representation and 12 points can determine a third-order cubic representation. The metrics and the Jacobian of the transformation can be computed for each element. The governing equations in the

reference domain are then transferred into the computational domain, and the transformed equations take the form we derived earlier:

$$\frac{\partial \mathbf{U}^c}{\partial t} + \frac{\partial \mathbf{F}^c}{\partial \xi} + \frac{\partial \mathbf{G}^c}{\partial \eta} = 0 \quad (18)$$

where

$$\mathbf{U}^c = J_{Ts} \mathbf{U}^r$$

and

$$\begin{bmatrix} \mathbf{F}^c \\ \mathbf{G}^c \end{bmatrix} = J_{Ts} [\mathbf{G}_{Ts}^{-1}] \begin{bmatrix} \mathbf{F}^r \\ \mathbf{G}^r \end{bmatrix} \quad (19)$$

IV.A. Solution Reconstruction

In the standard element, two sets of points are defined, namely the solution points and the flux points. In order to construct a degree $(N - 1)$ polynomial in each coordinate direction, solutions at N points are required.

The solution points in 1D are chosen to be the Gauss points defined by:

$$X_s = \frac{1}{2} \left[1 - \cos \left(\frac{2s-1}{2N} \cdot \pi \right) \right], s = 1, 2, \dots, N. \quad (20)$$

The flux points were selected to be Legendre-Gauss quadrature points plus the two end points 0 and 1.²⁵ Choosing $P_{-1}(\xi) = 0$ and $P_0(\xi) = 1$, the higher-degree Legendre polynomials can be determined as:

$$P_n(\xi) = \frac{2n-1}{n} (2\xi-1) P_{n-1}(\xi) - \frac{n-1}{n} P_{n-2}(\xi) \quad (21)$$

The locations of these Legendre-Gauss quadrature points are the roots of equation $P_n(\xi) = 0$. They are generally found to be more stable for SD methods¹¹ than the Gauss-Lobatto flux points.

Using the solutions at N solution points, a degree $(N - 1)$ polynomial can be built using the following Lagrange basis:

$$h_i(X) = \prod_{s=0, s \neq i}^N \left(\frac{X - X_s}{X_i - X_s} \right) \quad (22)$$

Similarly, using the fluxes at $(N + 1)$ flux points, a degree N polynomial can be built for the flux using a similar Lagrange basis:

$$l_{i+1/2}(X) = \prod_{s=0, s \neq i}^N \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}} \right) \quad (23)$$

The reconstructed solution for the conserved variables in the standard element is just the tensor products of the two one-dimensional polynomials,

$$\mathbf{U}^c(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \mathbf{U}^c_{i,j} h_i(\xi) \cdot h_j(\eta) \quad (24)$$

IV.B. Flux Reconstruction

From the reconstructed solution $\mathbf{U}^c$ in the computational space, the flux vectors $\mathbf{F}^c(\mathbf{U}^c, \nabla \mathbf{U}^c)$ and $\mathbf{G}^c(\mathbf{U}^c, \nabla \mathbf{U}^c)$ at the flux points in the computational space can be reconstructed through transformations to and from the reference space:

$$\mathbf{U}^c \xrightarrow{J_{Ts}^{-1}} \mathbf{U}^r_{fp} \longrightarrow [\mathbf{F}^r, \mathbf{G}^r]_{fp}^T \xrightarrow{S.T.} [\mathbf{F}^c, \mathbf{G}^c]_{fp}^T$$

where $S.T.$ is the steady transformation operator that maps the reference space flux vectors to the computation space flux vectors, and is represented mathematically as:

$$\begin{bmatrix} \mathbf{F}^c \\ \mathbf{G}^c \end{bmatrix} = J_{Ts} [\mathbf{G}_{Ts}^{-1}] \begin{bmatrix} \mathbf{F}^r \\ \mathbf{G}^r \end{bmatrix}, \text{ where } \mathbf{G}_{Ts} = \begin{bmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} \end{bmatrix}$$

For steady problems, the above mapping and computation will be sufficient. When the actual physical domain is moving in time, a further unsteady transformation need to be introduced between the reference domain and the physical domain to introduce the unsteady perturbations in the physical space into the stationary reference space through the time dependent transformation metrics. The flow diagram now looks as:

$$\mathbf{U}^c \xrightarrow{J_{Ts}^{-1}} \mathbf{U}^r_{fp} \xrightarrow{J_{Tu}^{-1}} \mathbf{U}^p_{fp} \longrightarrow [\mathbf{F}^p, \mathbf{G}^p]_{fp}^T \xrightarrow{U.T.} [\mathbf{F}^r, \mathbf{G}^r]_{fp}^T \xrightarrow{S.T.} [\mathbf{F}^c, \mathbf{G}^c]_{fp}^T$$

where $U.T.$ is the unsteady transformation operator that maps the physical space flux vectors to the reference space flux vectors, and is represented mathematically as:

$$\begin{bmatrix} \mathbf{F}^r \\ \mathbf{G}^r \end{bmatrix} = J_{Tu} [\mathbf{G}_{Tu}^{-1}] \begin{bmatrix} \mathbf{F}^p \\ \mathbf{G}^p \end{bmatrix} - J_{Tu} [\mathbf{G}_{Tu}^{-1}] \begin{bmatrix} \mathbf{U}^p \frac{\partial x}{\partial t} \\ \mathbf{U}^p \frac{\partial y}{\partial t} \end{bmatrix}, \text{ where } \mathbf{G}_{Tu} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial y}{\partial X} \\ \frac{\partial x}{\partial Y} & \frac{\partial y}{\partial Y} \end{bmatrix}$$

Having calculated the fluxes $\mathbf{F}^c_{i+1/2,j}$ and $\mathbf{G}^c_{i,j+1/2}$ at the flux points, the reconstructed flux polynomials take the following form:

$$\begin{aligned} \mathbf{F}^c(\xi, \eta) &= \sum_{j=1}^N \sum_{i=0}^N \mathbf{F}^c_{i+1/2,j} l_{i+1/2}(\xi) \cdot h_j(\eta), \\ \mathbf{G}^c(\xi, \eta) &= \sum_{j=0}^N \sum_{i=1}^N \mathbf{G}^c_{i,j+1/2} h_i(\xi) \cdot l_{j+1/2}(\eta) \end{aligned} \quad (25)$$

IV.C. Riemann Solver for Interface Flux

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, a Riemann solver is employed to compute a common flux at interfaces to ensure conservation and stability. In our case, we have used the Riemann problem solver ([?] or [?] with entropy fixing approach like[?]) to compute the interface fluxes. For moving mesh problem, the Rusanov flux $\mathbf{F}_n$ in the interface normal direction $\mathbf{n}$ in the physical domain is written as:

$$\mathbf{F}_n^p = \mathbf{F}^p \cdot \mathbf{n}^p = \frac{1}{2} [\mathbf{F}_R^p \cdot \mathbf{n}^p + \mathbf{F}_L^p \cdot \mathbf{n}^p - |(\mathbf{V}^p - \mathbf{V}_{mesh}) \cdot \mathbf{n}^p + \bar{c}| (\mathbf{U}^p_R - \mathbf{U}^p_L)] \quad (26)$$

where $\mathbf{F}_R^p$ and $\mathbf{F}_L^p$ are the fluxes at the flux points in the right and left cell interfaces. $\mathbf{U}^p_R$ and $\mathbf{U}^p_L$ are the solutions at the flux points in the right and left cell interfaces. $\mathbf{V}^p$ and $\bar{c}_n$ are the average normal velocity and sound speed at interface flux points. $\mathbf{V}_{mesh}$ is the mesh velocity vector. $\mathbf{n}^p$ is the local outgoing normal at the flux points at the interface in physical space.

To get the outgoing normal in physical space, we use the reference space outgoing normal $\mathbf{n}^r$ and transform it with the transformation matrix as:

$$\mathbf{n}^p = J_{Tu} [G_{Tu}]^{-1} \mathbf{n}^r$$

The fluxes and solutions in physical space that are required for the Riemann solver are obtained from their reference space counter parts through coordinate transformation, as we did previously.

V. Mesh Blending Method for Moving Deforming Boundaries

The grid deformation strategy implemented in the current study is first introduced by Morton, Melville, and Visbal.³⁰ It is an algebraic method that updates the mesh at every time step. This method has the property of preserving grid orthogonality near the surface under substantial deformation, which is very desirable for high Reynolds number viscous flow simulation.

V.A. Rigid Moving Boundary

To retain the orthogonality near the surface where boundary movement starts, the grid lines perpendicular to that surface are rotated and translated as rigid bodies with the movement of the surface.

Define the displaced mesh by $(x, y)^p$, the original undisplaced mesh by $(x, y)^{d.m}$, the translation displacement by (x_t^r, y_t^r) , the rotational angle by θ , and the center of mesh rotation by (x_0^r, y_0^r) , the transformation from the stationary reference mesh to the rigidly deformed mesh can now be represented as

$$\begin{bmatrix} x^{d.m} \\ y^{d.m} \end{bmatrix} = \begin{bmatrix} x^r \\ y^r \end{bmatrix} + \begin{bmatrix} x_t^r \\ y_t^r \end{bmatrix} + \begin{bmatrix} +\cos\theta & +\sin\theta \\ -\sin\theta & +\cos\theta \end{bmatrix} \begin{bmatrix} x^r - x_0^r \\ y^r - y_0^r \end{bmatrix} \quad (27)$$

Far away from the moving boundary, the flow domain is fixed and unchanged, and setting it as $(x, y)^{f.m}$, so that

$$\begin{bmatrix} x^{f.m} \\ y^{f.m} \end{bmatrix} = \begin{bmatrix} x^r \\ y^r \end{bmatrix} \quad (28)$$

In the region between the rigidly displaced mesh and the fixed stationary mesh, a polynomial blending is constructed to provide smooth propagation for the unsteadiness in the mesh from one end to another. Polynomial with zero slope at the end points lead to orthogonality at the boundaries of the deforming region. In this study, a 5th order bending polynomial, as used by Persson et al,²¹ is used. The form of the polynomial is:

$$f_b = 10s^3 - 15s^4 + 6s^5 \quad (29)$$

where $s = d/D$ is the ratio of the arc-length distance, d , of a point to the inner edge of the deforming region to the total width, D , of the deforming region.

The rigidly displaced mesh can then be combined with the fixed stationary mesh through the deforming mesh using the blending polynomial to form the transformed mesh in the unsteady physical domain as:

$$\begin{bmatrix} x^p \\ y^p \end{bmatrix} = f_b \begin{bmatrix} x^{f.m} \\ y^{f.m} \end{bmatrix} + (1 - f_b) \begin{bmatrix} x^{d.m} \\ y^{d.m} \end{bmatrix} \quad (30)$$

An example of a blended mesh for a rotating and translating airfoil in an O-mesh is illustrated in figure 1.

(a) Airfoil O-mesh

(b) Deformed O-mesh

Figure 1. Blended Deforming Mesh in Comparison with its Original Mesh

V.B. Elastic Deforming Boundary

While the above method works well for problems where the entire boundary is moving as a rigid body, a more general and flexible approach is needed for elastic deforming boundary where different points are displaced differently. The method used in the work by Persson et al²³ for deforming membrane is adopted and briefly outlined here.

For an elastic boundary under an externally applied load, the deformation distribution of the boundary can be obtained by solving the structure model. The computed deformation distribution allows us to displace the mesh from one grid point on the boundary to the next, and blend them to the farfield fixed mesh. However, in general the displacement distribution of an elastic boundary has non-vanishing slopes at its two ends, leading to unsmooth transition to its neighbouring mesh. To avoid this, a cubic spline function can be created by sampling the displacements along the elastic boundary and adding a point a small distance forward of the boundary and a second point a small distance aft of the boundary. By specifying zero derivative at those two additional points, the resultant cubic spline will have smooth transition to the surrounding undeformed mesh.

Consider a general deformed elastic boundary, and take the straight line joining the two ends of the boundary as the reference x-axis, the coordinates of the deformed mesh can be represented as:

$$\begin{bmatrix} x^{d.m} \\ y^{d.m} \end{bmatrix} = \begin{bmatrix} x^r \\ y^r \end{bmatrix} + \begin{bmatrix} 0 \\ y_c^d(x^r) \end{bmatrix} \quad (31)$$

where y_c^d is the cubic spline function of the deformation distribution, including the two additional end points for smooth transition.

Once the boundary displacement at every grid point is known and the new mesh coordinates computed as above, the same blending technique used in the previous section can be implemented to smooth the vertical displacement at the boundary to its far field stationary reference mesh. The coordinates of the entire mesh in the deformed physical space can now be written as:

$$\begin{bmatrix} x^p \\ y^p \end{bmatrix} = f_b \begin{bmatrix} x^r \\ y^r \end{bmatrix} + (1 - f_b) \begin{bmatrix} x^r + 0 \\ y^r + y_c^d \end{bmatrix} \quad (32)$$

Lastly, if the straight line joining the two ends of the boundary, which we treat as the reference x-axis, does not coincide with the real x-axis, then additional rotation and translation operations as outlined in the previous section should be carried out.

Figure 2. Cubic-Spline-Blended Deforming Mesh in Comparison with its Original Mesh

V.C. Numerical Validation for Flow Solver

In this section, we performed numerical validation for the SD solver on deforming meshes by comparing our numerical simulations of plunging and pitching airfoils with the available experimental results. The plunging and pitching airfoils problems have previously been studied with rigid mesh displacement.²⁴ We perform the same computations on deforming grids. Nearly identical results have been obtained for simulations performed on rigid mesh and deforming mesh. Both compare very favorably with experiments. The two cases are presented briefly below for completeness. For more details, please refer to.²⁴

Plunging Airfoil in Deformable Domain

We consider a plunge motion with plunging frequency $\omega = 2.46$ and amplitude $h = 0.12c$. This setup is identical to the one with $Sr = 1.5$ in the paper of Jones et al.³² The free-stream Mach number is equal to 0.2. Reynolds number is 1800. Dirichlet boundary condition is used. The SD solver with fourth-order solution polynomial is used for the computation. As shown in figures 3(a), our simulations show that the vortex shedding travels upwards if the first stroke of the airfoil goes downwards, and vice versa. The vortical pattern obtained from simulation agrees very well with the experimental results shown in figure 3(b) obtained in a water tunnel. The computation is able to reproduce the fine structures traveling in the opposite direction in the wake of the airfoil. These fine structures resemble the structures shown in the photograph of the experiment by Jones et al.³²

Figure 3. Computation vs Experiment for Plunging Airfoil

The vortical pattern as well as the lift and drag coefficients obtained in this study using deforming mesh are nearly identical to the previous study using rigidly plunging mesh.

Pitching Airfoil in Deformable Domain

We also studied viscous flow over a pitching NACA0012 airfoil along its quarter chord axis. The simulations were conducted at a significant higher Reynolds number than the previous plunging case. In particular, we aim to validate the simulation with existing experiment results of a pitching airfoil, and test the solver with increased Reynolds number. The simulation Reynolds number, based on the airfoil chord length, is $Re = 12,000$. This is chosen in accordance with the experimental study carried out by Koochesfahani,³³ in which the vortical patterns behind a pitching NACA 0012 airfoil around the quarter chord axis were studied and visualized in a water tunnel. The experiments were performed with the same Reynolds number. The simulations were computed at a small mach number of $M_\infty = 0.12$, at a reduced frequency of $k = \frac{\pi c f}{u_\infty}$, where f is the pitching frequency and c is the chord length. The amplitude of the sinusoidal pitching motion is denoted A . The airfoil starts with a zero mean angle of attack. Two cases have been computed and compared with the corresponding experimental wake pattern. The simulations here were performed with 4th order SD method.

Comparing the vortical patterns, the simulations for the cases with ($A=4$ degree, $k=0.835$) and ($A=4$ degree, $k=3.09$) produce wakes that are very similar to the experimental visualizations. In the early case, as shown in figure 4, the wake assumes a form of undulating vortex sheet. For the latter case, the simulation is able to capture the double-vortex feature which persists a long way downstream, as shown in figure 5.

(a) Computational Vortical Pattern

(b) Experimental Vortical Pattern

Figure 4. Vortical pattern for $A=4\text{deg}$, $k=0.835$

(a) Computational Vortical Pattern

(b) Experimental Vortical Pattern

Figure 5. Vortical pattern for $A=4\text{deg}$, $k=3.09$

As in the previous section, these simulations performed on deforming mesh yield almost the same results as those in the previous study with rigid mesh rotation, hence readers are encouraged to refer to it for further details which are not presented here.

V.D. Euler Vortex on Deforming Mesh

In order to conduct simulation of flow over an elastic flexing beam (presented in the later section) while keeping the channel wall boundaries fixed, the mesh is deformed and blended as discussed previously. To test the mesh deformation algorithm, we solve the euler vortex problem³⁴ on a mesh that deforms in exactly the same way as it would for the channel flow case. The vortex has a radius of 1.5 and strength of 0.5, and has an initial position away from the deforming region. For details of the analytical expression for the euler vortex problem, please refer to.^{21, 24, 34}

The pressure vortex at its initial position and at the position of maximum mesh deformation is shown in figure 6. As can be easily observed from the figure, despite the very weak vortex strength, the pressure contour is undistorted as it passes through the deforming mesh.

(a) Pressure Contour on Initial Undeformed Mesh

(b) Pressure Contour on Deformed Mesh

Figure 6. Euler Vortex Propagating Through Dynamic Deforming Mesh

VI. Structure Model and Fluid Structure Coupling

In this section we outline the structure models that are used in the computation, starting with a spring-mass model that allows simple geometries such as cylinder or airfoil to be modeled, and then following up with the finite element formulation of the elastic beam theory to model the more complex elastic beam bending problems.

VI.A. Moving Cylinder or Airfoil

For the structure model of the cylinder, for example, we use a simplified mass-spring system to represent the dynamic of the cylinder. The cylinder is constrained to move in the vertical direction only, but extension to the streamwise direction is straight forward. A sketch of the structure model is shown in Fig 7. The cylinder movement is driven by the force exerted by the external fluid flow.

Figure 7. Schematic of a Spring-Mass Model for the Cylinder

The governing equation for the simplified structure model is expressed as follows:

$$Ma + Kx = F \quad (33)$$

where M is the mass of the cylinder, K is the stiffness of the spring, F is the external force, which is equal to the lift exerted on the cylinder due to the fluid flow, x is cylinder displacement, and a is the acceleration of the cylinder. The second order dynamic equation can be written as a system of first order differential equations as follows:

$$M\dot{v} + Kx = F \quad (34)$$

$$\dot{x} - v = 0 \quad (35)$$

where v is the velocity of the cylinder. The system of first order equations can then be solved with the fourth order Runge-Kutta method, which is the same time integrator used in the fluid solver.

VI.B. Elastic Beam

To solve the elastic beam bending problem, we consider the theory of elasticity by assuming plane stress or plane strain condition. Interested readers are encouraged to refer to the work of Kwon and Bang,⁷ where greater details are given.

VI.B.1. Governing Equations

For a two-dimensional structural problem, there is a total of eight unknowns, including three stresses, three strains and two displacements. These can be solved by eight equations that govern the force equilibriums, constitutive properties, and kinematics for the structure element.

Equilibrium Equations

Consider a two-dimensional infinitesimal structural element, the force equilibrium in the x direction can be written as

$$(\sigma_x + \frac{\partial \sigma_x}{\partial x})dxdy - \sigma_xdxdy + (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial y})dxdy - \tau_{xy}dxdy + f_xdxdy = 0 \quad (36)$$

which balances pressure stress σ_x , shear stress τ_{xy} , and externally applied force f_x . It can be simplified to

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x = 0 \quad (37)$$

and similarly the y direction equilibrium equation can be expressed as

$$(\tau_{xy} + \frac{\partial \tau_{xy}}{\partial x})dxdy - \tau_{xy}dxdy + (\sigma_y + \frac{\partial \sigma_y}{\partial y})dxdy - \sigma_ydxdy + f_ydxdy = 0 \quad (38)$$

and simplified to the form

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y = 0 \quad (39)$$

Equations (53) and (54) are the equations of equilibrium.

Constitutive Equations

The relationship between the stresses and strains are governed by the constitutive equations. For an isotropic material, the constitutive equations that relate the stresses to strains, assuming the plane stress condition can be written in the matrix form as

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (40)$$

where E is the elastic modulus and ν is the Poisson's ratio. On the other hand, if plane strain is assumed, the constitutive equations are then expressed as

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (41)$$

Kinematic Equations

The kinematic equations relate strains to displacements. In matrix form, this is written as

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \quad (42)$$

where u and v are the displacements in the horizontal and vertical directions.

VI.B.2. Finite Element Formulation

Solution Representation

For the finite element discretization, the discrete solution in each element is locally represented by a shape function l on n solution nodal points u_i or v_i as

$$u(x, y) = \sum_{i=1}^n l_i(x, y)u_i \quad (43)$$

$$v(x, y) = \sum_{i=1}^n l_i(x, y)v_i \quad (44)$$

The derivative of the local solution can then be obtained by taking derivative of the shape function.

$$\frac{\partial u}{\partial x} = \sum_{i=1}^n \left(\frac{\partial l}{\partial x} \right)_i u_i, \quad \frac{\partial u}{\partial y} = \sum_{i=1}^n \left(\frac{\partial l}{\partial y} \right)_i u_i \quad (45)$$

$$\frac{\partial v}{\partial y} = \sum_{i=1}^n \left(\frac{\partial l}{\partial y} \right)_i v_i, \quad \frac{\partial v}{\partial x} = \sum_{i=1}^n \left(\frac{\partial l}{\partial x} \right)_i v_i \quad (46)$$

Galerkin Method

For Galerkin's method, the governing equations (53) and (54) are multiplied by a weighting function, which is chosen to be the same as the shape function l , and the resulting equations are integrated by parts. The resulting equations are expressed in the following form

$$\int_{\Omega} \left(\frac{\partial l}{\partial x} \sigma_x + \frac{\partial l}{\partial y} \tau_{xy} \right) d\Omega = \text{Source} + \text{Boundary Terms in } x \quad (47)$$

$$\int_{\Omega} \left(\frac{\partial l}{\partial x} \tau_{xy} + \frac{\partial l}{\partial y} \sigma_y \right) d\Omega = \text{Source} + \text{Boundary Terms in } y \quad (48)$$

which can be written as a system of equations as

$$\int_{\Omega} \begin{bmatrix} \frac{\partial l}{\partial x} & 0 & \frac{\partial l}{\partial y} \\ 0 & \frac{\partial l}{\partial y} & \frac{\partial l}{\partial x} \end{bmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} d\Omega = \begin{pmatrix} \text{Source} + \text{Boundary Terms in } x \\ \text{Source} + \text{Boundary Terms in } y \end{pmatrix} \quad (49)$$

Using the constitutive and the kinematic equations, we can replace the three stress terms first with the strain terms using material property matrix $[D]$, and then with the displacement terms. The resulting equation in terms of displacements becomes

$$\int_{\Omega} [B]^T [D] \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{pmatrix} d\Omega = \begin{pmatrix} \text{Source} + \text{Boundary Terms in } x \\ \text{Source} + \text{Boundary Terms in } y \end{pmatrix} \quad (50)$$

where the matrix $[B]^T$ has been used to represent the derivative matrix of the shape functions.

Element Stiffness Matrix

From the previous section, the solutions and their derivatives are expressed by the shape functions and nodal values. Hence we can further express the displacement terms in the governing equation by the shape functions and nodal values. Denoting the matrix that consists of the derivative of the shape function by $[B]$ and the nodal displacements by $[d]$, the finite element formulation applied to a local element can be written as

$$\int_{\Omega} [B]^T [D] [B] d\Omega [d] = [B]^T [D] [B] A [d] = [K^e] [d] = \begin{pmatrix} \text{Source} + \text{Boundary Terms in } x \\ \text{Source} + \text{Boundary Terms in } y \end{pmatrix} \quad (51)$$

where A is the element area and $[K^e]$ is called the element stiffness matrix, which is written as

$$[K^e] = \int_{\Omega^e} [B]^T [D] [B] d\Omega = [B]^T [D] [B] A \quad (52)$$

Mass Matrix for Dynamic Analysis

For dynamic problem, the inertia force is added to the equations of equilibrium to obtain the following equations of motions,

$$-\rho \frac{\partial^2 u}{\partial t^2} + \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x = 0 \quad (53)$$

$$-\rho \frac{\partial^2 v}{\partial t^2} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y = 0 \quad (54)$$

Following the same analysis by applying the Galerkin's method to the inertia terms, the resulting mass matrix is written as

$$[M^e] = \int_{\Omega^e} \rho [N]^T [N] d\Omega = \rho [N]^T [N] A \quad (55)$$

where $[N]$ is the matrix of the shape function l .

The governing equation for the elastic problem in terms of finite element formulation can finally be written in terms of the mass and stiffness matrices as

$$[M^e]\ddot{d} + [K^e]d = F \quad (56)$$

where F is the external force including the boundary condition term. The resulting expression resembles the simple form of the equation for the spring-mass model. As before the system of second order dynamic equations can be written as a system of first order differential equations, solved with the fourth order Runge-Kutta method.

VI.C. Numerical Validations for Elastic Beam

In this section, we validate our finite element beam solver by comparing results from existing literature. A benchmark test for fluid structure interaction simulations has been proposed by.³⁶ The test involves flow over a cylinder with an attached elastic beam. The beam deflection test was performed and the result was tabulated, which we used to validate our structure solver. Both static and dynamic analysis of the finite element beam are considered. The structure solver is tested by considering an elastic beam of the following configuration shown in figure 8(a). The material properties are tabulated in table 1.

Material Properties	Static Test 1 Values	Static Test 2 Values	Dynamic Test Values
Poisson's Ratio ν	0.4	0.4	0.4
Density ρ (kg/m^3)	1,000	1,000	1,000
Young's Modulus E	1,400,000	5,600,000	1,400,000
Gravity g (m/s^2)	2	2	2

Table 1. Material Properties of the Elastic Beam

VI.C.1. Finite Element Elastic Beam Static Test

For the static test, the beam is subjected to its own weight under gravity. The tip vertical deflection of the elastic beam is computed and compared with the result from the benchmark test. The comparison is tabulated in table 3, while the deflected beam is shown in figure 8(b). By increasing the number of degree of freedoms in the finite element beam, we obtained a converged solution that agrees reasonably well with the benchmark test result.

Figure 8. Finite Element Results for Beam Deflection Under Gravity

Static Test 1	Benchmark	Current Solver	Current Solver	Current Solver
Vertical Deflection	-0.06603	-0.06738	-0.06779	-0.06779
No. of DOFs	6468	202	1002	1602

Table 2. Comparison of Finite Element Beam Deflection Under Gravity

Static Test 2	Benchmark	Current Solver	Current Solver	Current Solver
Vertical Deflection	-0.01697	-0.01684	-0.01695	-0.01695
No. of DOFs	6468	202	1002	1602

Table 3. Comparison of Finite Element Beam Deflection Under Gravity

VI.C.2. Finite Element Structural Beam Dynamic Test

For the dynamic analysis, we solve the structure equations with the added inertial force. The equations can be written as a system of first order equations, which are then solved with the fourth order Runge-Kutta method. The time history of the beam tip vertical deflection is shown in figure 9. The number of degree of freedoms used is $DOFs = 202$. The time step is set equal to $dt = 0.0001$. The resulting structure dynamic agrees well with the work by Turek and Hron.³⁶

(a) Beam Tip Dynamic by Turek and Hron³⁶

(b) Beam Tip Dynamic with Current Method

Figure 9. Finite Element Result Comparison for Beam Dynamic Under Gravity

VI.C.3. Discussion

While more sophisticated finite element analysis (FEA) solver can be used to model the beam more accurately, the current finite element method based on the work of Kwon and Bang is simple and efficient to implement while providing reasonably accurate results. While the emphasis of the present study is to devise a framework to integrate a finite element structure solver with the high order SD solver to arrive at a general purpose fluid-structure-interaction solver, a more sophisticated FEA solver can be implemented in the future in a straightforward manner.

VII. Fluid Structure Coupling and Time Integration Method

The interaction between the fluid force acting on the solid boundary and the subsequent solid boundary deflection and velocity are handled through multi-stage explicit Runge-Kutta time stepping. At each RK stage, the fluid dynamic force computed from the fluid solver is integrated along the solid boundary. The resultant forcing vector is used to compute the structure deflection and deflection velocities. Subsequently, the flow domain is deformed according to the boundary displacements and its velocities. We do not employ sub-iteration to ensure convergence, but rely on the multi-stage RK method for close coupling. For this reason, the 5-stage Strong Stability Preserving RK4 scheme is used.

VIII. Results

In this section, we present simulation results for two fluid structure interaction problems. In the first case, we consider the configuration of an elastic beam attached to the back of a cylinder, and investigate the interaction of the elastic beam with the wake of the cylinder. We start by studying steady flow over the cylinder-beam structure, before presenting the simulation results for the fluid-structure interaction problem. In the second case, we examined the flow induced oscillation of a free floating cylinder. The finite mass cylinder is given the degree of freedom to move in the cross-flow direction, in reaction to the incoming freestream.

VIII.A. Steady Flow over a Cylinder-Beam

The fluid solver is first tested with channel flow over rigid undeflected cylinder-beam configuration. The inflow boundary condition is set according to the benchmark test such that the inlet velocity has a parabolic profile. With our compressible full Navier-Stokes solver, we have chosen our parameters such that this corresponds to a Mach number of 0.02 and 0.1 respectively. The flows are simulated using 4th order SD method.

Reynolds Number	Average Inlet Speed	Benchmark (D, L)	Current Solver (D, L)
20	0.2	(14.29, 1.12)	(13.62, 1.08)
100	1.0	(136.70, 10.53)	(127.60, 9.55)

Table 4. Comparison of Lift and Drag Forces

The vorticity, Mach, and pressure contours of the steady state solution for the $Re = 100$ case are shown in figure 11. The drag and lift forces are compared with the incompressible results from the benchmark test. The comparison is shown in table 4. The lift and drag forces in both cases are close to the benchmark test results. The deviations are likely due to the difference between a compressible and an incompressible solver. We observe that as the freestream Mach number increases, the difference also increases. Regarding the production of the lift force, we note that the test case has been set up so that the flow is not symmetrical along the center line. The channel is slight wider above the center line than below. The time history of the force coefficients of the channel flow over the rigid cylinder-beam structure is shown in figure 10.

Figure 10. CL and CD Time History for Steady Channel Flow over a rigid Cylinder-Beam Configuration

VIII.B. Fluid Structure Interaction Problem I

For the fluid structure interaction simulation, the elastic beam is modeled using the finite element structural solver formulated previously. The external fluid force along the beam boundary acts as the source term for the beam bending problem. Cubic spline curves are fitted through both the computed beam displacements

Figure 11. Steady Channel Flow over a Cylinder-Beam Configuration

and beam velocities. The spline curves are subsequently used to deform the mesh and to solve the flow equations on the deforming mesh. The elastic modulus and the density of the elastic beam are chosen to be $1.4e^6$ and 1,000 respectively. The density and the pressure of the fluid are set equal to 1 and 71.4 respectively, resulting in a speed of sound of 10. The simulations are computed with 4th order SD method, with the time step $dt = 5e^{-5}$.

Reynolds Number 200 and Mach Number 0.2

We consider the case of a low Reynolds number flow, with $Re = 200$ and $Mach = 0.2$. The time history of the vertical deflection of the beam tip is shown in figure 12 (a). The tip reference point is the mid point of the beam tip, located at (0.35, 0.02). The amplitude of the beam tip deflection is about $d = 0.04$. The flow induced beam flapping frequency is about $f = 1.66\text{Hz}$. Plotted in figure 12 (b) are the time histories of the lift coefficients. The pressure and viscous components of the lift coefficient are plotted along side the total lift coefficient. The cylinder diameter, $d = 0.1$, is used as the reference length. Four instances of the

Figure 12. Tip Deflection and CL Time Histories for the Fluid Structure Interaction Problem. $Re=200$. $Mach=0.2$. Pressure component of CL curve is in dashed blue color. The viscous component is in green dash-dot curve. Total CL is the red solid curve.

mach and pressure contours in a single flapping cycle are plotted in figure 13 and figure 14. Figure 13 (a) to (b) show the velocity fields around the beam as it moves from its top position to the bottom position and finally back to the center position. Figure 14 (a) to (b) show the corresponding pressure fields around the beam. We find that low pressure region develops on the upper surface as the beam deflects downwards, and vice versa. The alternating lift force generated by the flow subsequently leads to a periodic deflection of the elastic beam.

Drag Difference due to Elasticity

Figure 15. Comparison of drag time histories for rigid (left) and elastic (right) beam. Pressure component of CD curve is in dashed blue color line. The viscous component is in green dash-dot curve. Total CD is the red solid curve.

To examine the effect on drag due to structure elasticity, a second simulation is performed with the same flow conditions but with the beam being rigid. The resulting drag coefficient time history is plotted in figure 15 (a), along side the drag coefficient time history of the previous fluid structure interaction case in figure 15 (b). While the average value of the drag has increased by a small amount, about $\Delta C_D = 0.15$, the

instantaneous drag rise due to structure deformation, however, can be quite significant.

VIII.C. Fluid Structure Interaction Problem II

Flow Induced Cross-Flow Oscillation of a Cylinder

Finally we study the flow over a finite mass cylinder that is free to oscillate in the cross-flow direction. Fourth order solution polynomial is used in the SD method. The flow Reynolds number is $Re = 20,000$. The flow Mach number is $M = 0.1$. The cylinder has a diameter of $D = 1.0$, a mass of $m_c = 0.00005 \text{ kg}$ and an initial position of $(x_0, y_0) = (0, 0)$.

Figure 16. Cylinder Movement Evolution

The evolution of the cylinder trajectory is shown in figure 16. The vorticity is shed behind the cylinder as the fluid flows over it. The cylinder initially stays at its original position when the vortical pattern behind the cylinder remains symmetrical, as seen in figure 16(a). However, after a period of time, the symmetry of the vortical pattern is lost, and subsequently a lift is generated on the cylinder which accelerates it upwards. The upward trajectory of the cylinder is shown from figure 16(b)-(e). The cylinder eventually peaked its position at around position 5, and moved back downwards, and settled at a stable position at 6.

After the transient movements shown previously, the cylinder eventually oscillates around a stable position. This is illustrated in figure 16(i)-(l). The lift and drag coefficients time histories are shown in figure 17.

IX. Conclusion

In this paper, we have formulated and implemented a computational aeroelastic solver based on the high order SD method on deforming meshes. We discussed, among other things, the method to deform mesh for flexible surface using cubic spline curve, the formulation of a finite element solver based on elastic theory, and the integration of the structure and fluid solver. Many numerical tests have been conducted to validate the deforming mesh algorithm and the finite element structure solver. In addition, two fluid-

Figure 17. CL and CD Time History for Flow Induced Cylinder Oscillation

structure interaction problems are solved using our present formulation. From those tests, we showed that the solutions on deforming meshes produce results that are nearly identical to fixed meshes. The structure solver yields fairly accurate answers despite its simple formulation. The integration of the fluid and structure solvers is achieved through multi-stage Runge Kutta method, which produces stable coupling of the fluid structure dynamics while at the same time remains very simple and efficient. The current framework can very easily accommodate a different FEA solver without major reformulation.

Acknowledgement

This research work is made possible by the generous fundings from the National Science Foundation and the Air Force Office of Scientific Research under the grants 0708071 and 0915006 monitored by Dr. Leland Jameson, and grant FA9550-07-1-095 by Dr. Bariba Fahroo. The authors would like to thank them for their continuous support.

References

- ¹Cockburn,B., Shu, C.-W, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework", Math. Comput, 52, 1989, 411-435.
- ²Cockburn,B., Lin, S.Y., Shu, C.-W, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems", J. Comput. Phy., 84, 1989, 90-113.
- ³Cockburn,B., Hou, S., Shu, C.-W, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case", Math. Comput, 54, 1990, 545-581.
- ⁴Cockburn,B., Shu, C.-W, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: Multidimensional system", J. Comput. Phy., 141, 1998, 199-224.
- ⁵Hesthaven,J.-S, Warburton, T., "Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications", Springer Verlag, 141, 2008.
- ⁶Kopriva,D.A, Kolas, J.H, "A conservative staggered-grid Chebyshev multidomain method for compressible flows", J. Comput. Phy., 125(1), 1996, 244-261.
- ⁷Liu,Y., Vinokur, M., Wang, Z.J., "Spectral difference method for unstructured grids I: Basic formulation", J. Comput. Phy., 216, 2006, 780-801.
- ⁸Z.J. Wang, Y. Liu, G. May and A. Jameson, "Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations, Journal of Scientific Computing, Vol. 32, No. 1, pp. 45-71 (2007).
- ⁹Y. Sun and Z. J. Wang and Y. Liu, "High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, Communication in Computational Physics, Vol. 2, pp. 310-333 (2007).
- ¹⁰May, G., "The spectral difference scheme as a quadrature-free discontinuous Galerkin method", AICES, 11, 2008
- ¹¹Jameson, A., "A Proof of the Stability of the Spectral Difference Method for All Orders of Accuracy", Journal of Scientific Computing, 2009 (to appear).
- ¹²Van den Abeele, K., Lacor, C., Wang, Z.J., "On the Stability and Accuracy of the Spectral Difference Method", J. Sci. Comput., 37, 2008, 162-188.
- ¹³C. Liang, S. Premasuthan, A. Jameson, "High-order accurate simulation of flow past two side-by-side cylinders with Spectral Difference method", Journal of Computers and Structures, 2009, vol 87, pp. 812-817.
- ¹⁴C. Liang, S. Premasuthan, A. Jameson, Z. J. Wang, "Large Eddy Simulation of compressible turbulent channel flow with Spectral Difference method", AIAA paper, AIAA-2009-402, Orlando, FL.
- ¹⁵C. Liang, A. Jameson, Z. J. Wang, "Spectral Difference method for two-dimensional compressible flow on unstructured grids with mixed elements", Journal of Computational Physics, vol 228, pp 2847-2858, 2009.

- ¹⁶S. Premasuthan, C. Liang, A. Jameson, Z. J. Wang, "A P-multigrid Spectral Difference method for viscous flow", AIAA paper, AIAA-2009-950, Orlando, FL.
- ¹⁷A. H. Mohammad, Z. J. Wang and C. Liang, "LES of turbulent flow past a cylinder using spectral difference method", AIAA applied aerodynamics meeting, 2008, Hawaii. AIAA-2008-7184.
- ¹⁸K. Ou, C. Liang, S. Premasuthan, A. Jameson, "High-Order Spectral Difference Simulation of Laminar Compressible Flow Over Two Counter-Rotating Cylinders", AIAA-2009-3956, 27th AIAA AA Conference, San Antonio, June 2009.
- ¹⁹D.J. Mavriplis, Zhi Yang, "Construction of the Discrete Geometric Conservation Law for High-Order Time Accurate Simulations on Dynamic Meshes", J.Computational Physics 213, 557-573, 2006
- ²⁰D.J. Mavriplis, Cristian R. Nastase, "On the Geometric Conservation Law for High-order Discontinuous Galerkin Discretizations on Dynamically Deforming Meshes", 46th AIAA Aerospace Sciences Meeting, Jan 2008, Reno, Nevada
- ²¹P. O. Persson, J. Bonet, and J. Peraire, "Discontinuous Galerkin Solution of the Navier-Stokes Equations on Deformable Domains", Computer Methods in Applied Mechanics, and Engineering, Volume 198, Issues 17-20, 15 April 2009
- ²²K. Ou and A. Jameson, "On the Temporal and Spatial Accuracy of Spectral Difference Method on Moving Deformable Grids and the Effect of Geometric Conservation Law", 21st Century Challenges in Computational Engineering and Science - An International Symposium on the occasion of the 75th Birthday of Antony Jameson, Princeton, NJ, Nov, 2009
- ²³P.-O. Persson, J. Peraire, J. Bonet, "A High Order Discontinuous Galerkin Method for Fluid-Structure Interaction", 18th AIAA Computational Fluid Dynamics Conference, June 2007. AIAA-2007-4327
- ²⁴K. Ou, C. Liang, A. Jameson, "High-Order Spectral Difference Method for the Navier-Stokes Equations on Unstructured Moving Deformable Grids", AIAA-2010-541, 48th AIAA Aerospace Sciences Meeting, Orlando, Jan 2010.
- ²⁵Huynh, H.T., "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods", AIAA 2007-4079, 18th AIAA CFD Conference, Miami, June 2008.
- ²⁶V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, *J. Comp. Math. Phys. USSR*, 1961
- ²⁷P.L. Roe, Approximate Riemann Solvers, *Journal of Computational Physics*, 1981
- ²⁸A Harten & J.M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws, *Journal of Computational Physics*, 1983
- ²⁹Miguel R. Visbal, Datta V. Gaitonde, "On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes", J.Computational Physics 181, 155-185, 2002
- ³⁰Scott A. Morton, Reid B. Melville, and Miguel R. Visbal, "Accuracy and Coupling Issues of Aeroelastic Navier-Stokes Solutions on Deforming Meshes", Journal of Aircraft, Vol.35, No.5, Sept, 1998
- ³¹Y. W. Kwon and H. Bang, "The Finite Element Method using Matlab", CRC Press, NY, 1997
- ³²K. D. Jones and C. M. Dohring and M. F. Platzer, "Experimental and computational investigation of the Knoller-Betz effect ", AIAA Journal, vol 36, 780-783, 1998
- ³³Manoochehr M. Koochesfahani, "Vortical Patterns in the Wake of an Oscillating Airfoil", AIAA Journal, vol 27, 1989
- ³⁴G. Erlebacher, M. Y. Hussaini, C.-W. Shu, "Interaction of a shock with a longitudinal vortex", J. Fluid Mech. 337 (1997) 129-153.
- ³⁵C. Liang and A. Jameson and Z. J. Wang, "Spectral Difference method for two-dimensional compressible flow on unstructured grids with mixed elements", Journal of Computational Physics, vol 228, 2847-2858, 2009
- ³⁶Stefan Turek and Jaroslav Hron, "Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow", Fluid-Structure Interaction Modelling, Simulation, Optimisation, Springer Berlin Heidelberg, Volume 53, 371-385