Optimization of Flow Past a Moving Deformable Airfoil using Spectral Difference Method

Kui Ou* and Antony Jameson[†]

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

In this paper we examine the low Reynolds number flow past a harmonically plunging, pitching, and deforming airfoil, both individually and in combination. The lift and thrust characteristics of the combined flapping airfoil motion of plunging, pitching and deforming are investigated using a high-fidelity flow solver. The high-order Navier-Stokes solver based on Spectral Difference (SD) method is used to analyze the moving airfoils. An algebraic mesh deformation algorithm is used to accommodate airfoil deformation and movement. The thrust characteristic of a harmonically oscillating and deforming airfoil is investigated. For a deformation function defined by two parameters, i.e. maximum camber A_c and the location of the maximum camber X_c , a particular deformation profile is identified that produces significant thrust.

I. Introduction

Traditionally a wing has been regarded as a lift generating device, with its primary function being to support weight. For natural flyers, such as birds and insects, their wings can sustain their body weights in the air as well as propel themselves in their desired directions. Hence, flapping wing aerodynamics differs from fixed wing aerodynamics in the regard that unsteady motion of the wing need to be considered, and the aerodynamic forces of interest should include thrust, in addition to lift and drag. Research in flapping wing aerodynamics is certainly rooted in our innate desire to understand the wonder of animal flight, but in recent years it is also heavily driven by the mounting interest to design and build micro aerial vehicles, which could potentially find very useful applications in military and civil arena. While micro aerial vehicles are characterized by their definition of small sizes, they are also intent to fly and propel themselves in a similar way, and in the similar Reynolds numbers regime, as the natural flyers. Therefore, a major component of flapping aerodynamics consists of, firstly, understanding the lift and thrust characteristics of unsteady moving wings, and secondly, optimizing and designing wings and wing motions that could lead to efficient flapping wing micro aerial vehicles. This is the objective of the present study.

The complexity of 3D flapping wing aerodynamics should not be underestimated. The low Reynolds number, viscous, unsteady, and usually transitional flow itself is complex enough. This is further complicated by the dynamic multi-degree-of-freedom wing movements, the active control employed by the natural flyers and the passive fluid structure interaction between flow and wing surface. There have been substantial research efforts to develop high fidelity flow solvers for flapping flights, but the task remain very challenging. While combining flapping wing flow analysis and optimization in three-dimension is beyond the scope of our present study, we take a more fundamental approach to investigate the more tractable problems of analyzing and optimizing two-dimensional moving airfoil. Flapping motion of airfoil can generally be described by plunging and pitching motions. Much study, either computational or experimental, has gone into examining the fluid flow past a plunging or pitching or combined plunging-pitching airfoil. The lift and thrust producing capability of certain plunging and pitching motions have been demonstrated by previous studies. However, it is not uncommon for biological flyers or swimmers to change their effective shapes during their locomotion. Therefore, in this paper, we introduce a further degree of freedom by allowing the airfoil to change its effective shape and camber. This, when combined with plunging and pitching motions, will define a more

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

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

complete flapping wing locomotion space. We analyze and attempt to optimize this combined motion in the current paper.

II. Numerical Flow Solver

For the flow solver, we use the high order spectral difference method¹⁻⁴. The SD method, like Discontinuous Galerkin,⁵⁻⁹ is element-wise discontinuous. Within each individual element, collocated solution and flux points are used to store flow solution and flow flux information. With these solution and flux points, one solution polynomial and one flux polynomial can be constructed to represent the flow solution and flux in the local domain spanned by the element. The flux polynomial is one order higher than the solution polynomial such that when it is differentiated the flux divergence is of the same order as the solution polynomial. Therefore, the number of flux points is one more than that of solution points. At the element interface, where two sets of local solution meet each other, the flux across the interface is discontinuous. A method to enforce continuity is to use a Riemann solver, which was developed to handle discontinuity across interface. Such a Riemann solver for enforcing flux continuity includes Roe¹⁸ flux or Rusanov¹⁷ flux. Details of the SD formulation are outlined in the subsequent section. Since the current formulation of the SD method on dynamic meshes is based on algebraic mesh mapping, coordinate transformations and conservation law transformations will be discussed first.

III. 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.

III.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:

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

III.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:

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

The Jacobian of the unsteady transformation gradient is equal to:

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

IV. Conservation Laws Transformations

IV.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

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

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 P_r 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

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

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

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

$$\mathbf{U}^r = J_{T_u} \mathbf{U}^p$$

$$\mathbf{F}^r = J_{Tu}(\mathbf{F}^p \frac{\partial X}{\partial x} + \mathbf{G}^p \frac{\partial X}{\partial y} + \mathbf{U}^p \frac{\partial X}{\partial t}) \quad (12)$$

$$\mathbf{G}^r = J_{Tu}(\mathbf{F}^p \frac{\partial Y}{\partial x} + \mathbf{G}^p \frac{\partial Y}{\partial y} + \mathbf{U}^p \frac{\partial Y}{\partial t})$$

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

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

IV.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

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

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

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

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

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

V. Formulation of Spectral Difference Method on Deforming Meshes

The standard formulation of the spectral difference (SD) method on fixed meshes can be found in.⁴ 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

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

V.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 1.¹⁶ 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 methods 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,

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

V.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:

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

V.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 (¹⁷ or ¹⁸ with entropy fixing approach like ¹⁹) 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.

VI. Mesh Blending Method for Moving Deforming Boundaries

The grid deformation strategy implemented in the current study is first introduced by Morton, Melville, and Visbal.^{10,11} 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.

VI.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,¹² 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.

VI.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 al¹³ 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.

(a) Airfoil O-mesh

(b) Deformed O-mesh

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

For further details on the formulation, analysis, accuracy and application of the SD scheme on dynamic meshes, the reader can refer to.^{14,15}

VII. Flow Results

In this section, we present and discuss the following aspects. Firstly, we examine unsteady flow past an airfoil that deforms its camber line at different reduced frequencies while keeping its thickness distribution unchanged. We show through a simple example that by choosing a suitable combination of the deformation frequency and amplitude, the drag coefficient of the airfoil is subsequently reduced. Hence flexing an airfoil along its camber line can lead to a thrust producing mechanism. Having demonstrated this, we move onto discussing an airfoil undergoing more complex flapping motion. The flapping motion of interest consists of a coordinated plunging and pitching motion. The flow conditions and the flapping motions are selected based on previous study. The current setup has been investigated by experiments as well as computational simulations. We first performed the same flow simulation and later incorporated the deforming motion into the plunging and pitching motion. The benefit of airfoil deformation will be further investigated.

Figure 2. Examples of deformed mesh

VII.A. Deforming Airfoil

The simulation is performed at low Mach number ($Mach=0.1$) and low Reynolds number ($Re=100$). The airfoil is a symmetric naca0012 airfoil. The chord length is 0.5. The deformation of the airfoil camber line and the related mesh motion are prescribed analytically according to the following equation:

$$y(x, t) = \sin(2\pi ft)(A \cos(2\pi x/L) - A) \quad (33)$$

where f is the frequency of deforming, A is one half of the amplitude of deformation, and L is the wavelength of the sinusoidal function that determines the spatial profile of the camber line.

We performed two simulations with the airfoil camber deforming at two different reduced frequencies. In the first case, we selected $f = 0.5$, $A = 0.025$, and $L = 0.35$. This corresponds to a Strouhal number of 0.25. In the second case, we selected $f = 0.4$, $A = 0.05$, and $L = 0.35$. This corresponds to a Strouhal number of 0.4. The two results are compared in Figure 3 and 4. From the pressure and Mach contours, we observe that a more pronounced camber deformation and a higher reduced frequency lead to greater instantaneous pressure fields and flow momentums.

The aerodynamic drag coefficients for the two cases are shown in Figure 5. On average, deforming the airfoil at higher reduced frequency reduces the aerodynamic drag.

VII.B. Plunging and Pitching Airfoil

The combined plunging and pitching naca0012 airfoil has been investigated experimentally by Anderson et al.²⁰ The experiment was performed at a water tank at 40,000 Reynolds number. The airfoil was pitching around the one third chord axis, and the time-average thrust coefficient was measured. One of the optimal cases was found with the plunging amplitude of $h_0 = 0.75c$, the pitching motion leading plunging motion by 75° , and mean angle of attack of $\alpha = 15^\circ$.

For the 2D simulations performed using the SD method, a very coarse O-topology mesh with 1,310 cells was used. The simulations were performed with 4th order SD method. The resulting temporal variations of the drag coefficients for various choice of Strouhal numbers are plotted in Figure 6. The experimental result of thrust coefficient as a function of Strouhal number is shown in Figure 7(a) in solid black line. The same problem has been investigated by several authors^{21,22} using various CFD approaches. A rough estimate of the averaged results from these CFD efforts is plotted in Figure 7(a) in solid red line for reference. More detailed results are shown in Figure 7(b). The result obtained by the present SD method is shown in Figure 7(a) in solid blue line. The difference between the simulation results and the experimental data is mostly likely due to the three-dimensional effect which is not taken into account in the 2D simulation. Indeed, performing the same simulations in 3D, by first intruding the airfoil mesh in the spanwise direction, has led to vastly improved result. The 3D results are plotted in Figure 9. Discussing the 3D results is beyond the scope of this paper. Further details can be found in.²³

Figure 3. Comparison of Mach contours for a Naca0012 airfoil deforming with different maximum camber amplitudes at $\text{Mach}=0.1$ and $\text{Re}=100$. Top row

Figure 4. Comparison of Pressure contour for Naca0012 airfoil deforming with different maximum camber amplitudes at $\text{Mach}=0.1$ and $\text{Re}=100$

(a) Drag coefficient for $A=0.025$, $Str=0.25$

(b) Drag coefficient for $A=0.05$, $Str=0.40$

Figure 5. Comparison of drag coefficients for a Naca0012 airfoil deforming with different maximum camber amplitudes at Mach=0.1 and $Re=100$

(a) $Str=0.2$

(b) $Str=0.3$

(c) $Str=0.4$

Figure 6. Drag coefficient time histories for a Naca0012 airfoil pitching and plunging at different Strouhal numbers

(a) Thrust coefficient vs Str number

(b) Thrust coefficient vs Str number

Figure 7. Comparison of thrust coefficients as a function of Strouhal number for a pitching and plunging Naca0012 airfoil with $h_0 = 0.75$, $\alpha_0 = 15^\circ$ and $\Psi = 75^\circ$ at Mach=0.1 and $Re=40,000$

The fourth order simulation results for the $Str=0.4$ case is shown in Figure 8. The instantaneous Mach, pressure and vorticity contours at various instances in a plunging-pitching cycle are illustrated in Figure 8 (a), Figure 8 (b), and Figure 8 (c) respectively.

Figure 8. Mach, pressure and vorticity contours at various instances of a pitching-plunging Naca0012 airfoil during a flapping cycle.

VII.C. Plunging, Pitching, and Deforming Airfoil

In this section we demonstrate that by introducing a suitably selected deforming motion into the existing plunging pitching motion, the aerodynamic thrust can be effectively increased. The choice of the deforming profile, amplitude and frequency is at present ad hoc, and through experiments. A more systematic way of selecting those flapping parameters will be discussed in the later section. The airfoil camber is deformed according to a new function as defined below:

$$y(x, t) = \sin(2\pi ft + \Psi) A e^{\frac{-(x-b)^2}{2c^2}} \quad (34)$$

where f is the deforming frequency, Ψ is the phase angle, A is the amplitude of the deformation, while b and c determine the shape of the camber. Here we have chosen $f = \frac{StU_\infty}{2ch_0}$, $\Psi = 75^\circ$, $A = 0.1$, $b = 0.2$ and $c = 0.25$. The maximum camber is at 50 percent chord.

The time histories of the drag coefficient at increasing Strouhal numbers are plotted in Figure 9. The increase in thrust due to the introduction of camber deformation is clearly observed in the plots. The time averaged thrust coefficient versus Strouhal number curve for the present case is plotted in Figure 10. The improvement in the amount of thrust being produced due to the synchronized camber deformation is quite significant.

The Mach, pressure and vorticity contours for the present case are presented in Figure 11 (a), (b) and (c). The contour levels have been selected to match those in Figure 8.

VIII. Optimal Deformation Parameters

In the previous section, it was found that, for a flapping airfoil, a suitable choice of deformation function can lead to a significant increase in thrust. By exploring the design space of the deformation function, further

Figure 9. Drag coefficient time histories for a Naca0012 airfoil pitching, plunging and deforming at different Strouhal numbers

Figure 10. Comparison of the thrust coefficients vs Strouhal number curves for a moving Naca0012 airfoil at Mach=0.1 and Re=40,000

improvement might be achieved. This section is an effort towards this end. The objective is to investigate the effect of the location of the maximum camber and the effect of the deformation amplitude. This can be done by performing a sequence of simulations with incremental change in the maximum camber location at each fixed deformation amplitude. This process is then repeated for each new choice of deformation amplitude. While this exhaustive search process can seem rather involved, in practice, because the 2D simulations are not very expensive to evaluate, a much improved solution be found very quickly without developing sophisticated optimization strategies. Some results are presented and discussed in the next section.

VIII.A. Optimal Deformation Parameters

To identify a better maximum camber location, X_c , and to investigate the effect of the deformation amplitude, A_c , a number of simulations were performed first. The simulations were performed with the maximum deformed camber at the following 4 locations: 20% chord, 30% chord, 40% chord and 50% chord. At each selected location, the simulations were performed for a range of deformation amplitudes. The results are summarized in Figure 12. The direction for improvement can be identified readily from the figure. It is found that maximum camber location near 30% chord leads to better result. Also a large deformation amplitude is preferred.

Based on results from Figure 12, further simulations were performed at smaller incremental changes of the two design parameters. Curves in Figure 13 (b) and (c) further confirm that $X_c \approx 0.3c$ leads to the best result. This location is actually very close to the location of the pitching axis at one-third chord. Finally, the flapping motion of a pitching, plunging and deforming NACA0012 airfoil with $A_c = 0.5c$ and $X_c = 0.3c$ is depicted in Figure 13 (c). This choice of the deformation parameters improves the thrust coefficient by

Figure 11. Mach, pressure and vorticity contours for a pitching, plunging and deforming Naca0012 airfoil during a flapping cycle.

more than 3 times compared to the case in section VI.

The Mach, pressure and vorticity contours for the $A_c = 0.5c$ and $X_c = 0.3c$ case are presented in Figure 14 (a), (b) and (c). The contour levels are again selected to match those in Figure 8.

VIII.B. Mechanism of Thrust Generation

For this very high thrust flapping motion, some interesting observations on thrust generation can be made by examining the evolution of the streamwise momentum contours. A sequence of streamwise momentum contour pictures is shown in Figure 15. From all the plots in this figure, we can see that a momentum jet is created behind the flapping airfoil, indicating that the airfoil is producing thrust.

The mechanism by which the momentum jet is created is first discussed. By examining the upstroke motion, illustrated in Figure 15 (a) to (e), an important observation is the formation of a strong counter-clockwise rotating vortex at the airfoil's lower surface. The pronounced deformation of the airfoil not only induces the formation of the rotating flow but also helps to lock it in place throughout the upstroke phase. The strong rotating vortex plays a very significant role in contributing to thrust generation. Firstly, the strong rotation of the vortex significantly increases the flow speed near the airfoil. Secondly, the vortex is rotating in a favorable direction. During upstroke, the counter-clockwise rotation results in a large increase in the streamwise velocity below the airfoil. This increase of local flow speed due to the creation of the vortex is evident from the plots. In Figure 15 (e) the large patch of faster moving fluid particle below the airfoil is a direct consequence of the formation of the vortex. Conversely, during downstroke, a clockwise-rotating vortex is formed in the upper surface. The effect is the same, i.e. fluid particles away from the airfoil is drastically sped up by the rotation of the strong vortex. As another mechanism that contributes to creating thrust, Figure 15 (f) to (i) illustrate the beneficial effects of coordinated flapping movements. The well-coordinated pitching and deforming motion allow the aft section of the airfoil to catch the vortex-induced flow at a favorable angle and further accelerate the flow.

(a) Max. camber at $X_c = 20\%$ chord

(b) Max. camber at $X_c = 30\%$ chord

(c) Max. camber at $X_c = 40\%$ chord

(d) Max. camber at $X_c = 50\%$ chord

Figure 12. Temporal variations of the drag coefficients for different choices of deformation camber locations and amplitudes.

IX. Conclusion

In this study we developed a high-order flow solver that can handle dynamic boundary deformation. We applied this idea to dynamically vary the camber of an airfoil while keeping its thickness unchanged. Numerical experiment of a purely deforming airfoil shows that by increasing the Strouhal number of the deforming motion, effective thrust is produced. This models the similar mechanism that a swimming fish uses to propel itself by flexing its body at an actively controlled amplitude and frequency. Furthermore, we simulated and compared a coordinated plunging and pitching airfoil flapping motion with a coordinated deforming, plunging and pitching motion, and found in this case another evidence that deforming kinematic can effectively boost thrust production. For some good choices of the deformation profile, the improvements in thrust are significant for the flapping airfoil.

X. Acknowledgements

This research has been made possible through the support of the Air Force Office of Scientific Research under grants FA9550-07-1-095 and FA9550-10-1-0418, and the National Science Foundation under grants 0708071 and 0915006.

References

- ¹Kopriva, D.A., Kolas, J.H., "A conservative staggered-grid Chebyshev multidomain method for compressible flows", J. Comput. Phys., 125(1), 1996, 244-261.
- ²Liu, Y., Vinokur, M., Wang, Z.J., "Spectral difference method for unstructured grids I: Basic formulation", J. Comput. Phys., 216, 2006, 780-801.

(a) Deforming amplitude $A_c = 0.3c$

(b) Deforming amplitude $A_c = 0.4c$

(c) Deforming amplitude $A_c = 0.5c$

(d) Naca12 airfoil with a choice of $A_c = 0.5c$ and $X_c = 0.3c$

Figure 13. Temporal variations of the drag coefficients for different choices of deformation camber locations and amplitudes.

³Z.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).

⁴Y. 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).

⁵Cockburn, 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.

⁶Cockburn, 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. Phys., 84, 1989, 90-113.

⁷Cockburn, 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.

⁸Cockburn, B., Shu, C.-W, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: Multidimensional system", J. Comput. Phys., 141, 1998, 199-224.

⁹Hesthaven, J.-S, Warburton, T., "Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications", Springer Verlag, 141, 2008.

¹⁰Miguel 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

¹¹Scott 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

¹²P. 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

¹³P.-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

¹⁴K. 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, AIAA-2010-5032, 40th Fluid Dynamics Conference and Exhibit, Chicago, Illinois, 28 Jun - 1 Jul 2010

¹⁵K. 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.

¹⁶Huynh, H.T., "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods", AIAA 2007-4079, 18th AIAA CFD Conference, Miami, June 2008.

Figure 14. Mach, pressure and vorticity contours for the flapping Naca12 airfoil with a choice of $A_c = 0.5c$ and $X_c = 0.3c$ for the deforming function.

¹⁷V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, *J. Comp. Math. Phys. USSR*, 1961

¹⁸P.L. Roe, Approximate Riemann Solvers, *Journal of Computational Physics*, 1981

¹⁹A Harten & J.M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws, *Journal of Computational Physics*, 1983

²⁰Anderson, J.M., Streitlien, K., Barrett, D.S. and Triantafyllou, M.S. (1998) Oscillating foils of high propulsive efficiency, *J. Fluid Mech.*, 360, pp. 41-72.

²¹Y. Lian, and W.Shyy, "Aerodynamics of Low Reynolds Number Plunging Airfoil under Gusty Environment", AIAA 2007-71,45th AIAA Aerospace Sciences Meeting and Exhibit, 8 - 11 January 2007, Reno, Nevada

²²Lai, C.S.J. and Platzer, F.M. (1999) Jet Characteristics of a Plunging Airfoil, *Aiaa J.*, 37(12).

²³K. Ou and A. Jameson, 'Towards Computational Flapping Wing Aerodynamics of Realistic Configurations using Spectral Difference Method', AIAA Paper, 20th AIAA CFD conference at Hawaii, 2011.

Figure 15. Plots of x momentum contours