Fluid Dynamics and Co-located Conferences
June 24-27, 2013, San Diego, CA
21st AIAA Computational Fluid Dynamics Conference

Adjoint-Based Optimization of Flapping Kinematics in Viscous Flows

Marnix P. van Schrojenstein Lantman*
Department of Mechanical Engineering, University of Twente, the Netherlands
Krzysztof J. Fidkowski†
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109

We present an adjoint-based method for optimizing flapping motion kinematics in a viscous flow governed by the Navier-Stokes equations. We employ an arbitrary Lagrangian Eulerian formulation, discretized by a high-order discontinuous finite element method. Mesh motion is specified analytically, using parameters that we seek to optimize. Sensitivities are computed using a discrete unsteady adjoint approach, and these sensitivities drive a gradient-based optimization scheme. In this paper we show results for optimizations with a moderate number of parameters in a modal approximation to the kinematics.

I. Introduction

Many works have considered optimization of flapping flight motion. An early successful strategy for preliminary design is wake optimization,[1] which provides the optimal vorticity distribution for three-dimensional flapping flight. This approach has been combined in recent multi-fidelity analysis and design research.[2]

Two-dimensional flapping motion is computationally cheaper than three-dimensional motion and serves as the starting point for most optimization studies. These generally focus on optimization of kinematics,[3–6] although shape deformation has recently been addressed as well.[7,8]

Optimization of full three-dimensional flapping with high-fidelity computational methods has only been addressed recently. Due to the cost of the 3D simulations, most successful approaches so far have coupled with lower fidelity models[9] or have used a relatively small number of parameters.[10] The latter restriction is due to the finite-difference approaches for obtaining gradients used to guide the optimization algorithms.

Recently, adjoint-based methods have been applied to optimizing wing kinematics[11] using a finite volume code. Our work follows along these lines but differentiates itself via the discretization, in our case high-order discontinuous finite elements, and via parametrization of the kinematics, in our case using modal decompositions. The latter feature lets us explore more general flapping motions in the optimization.

In terms of optimization methods, while some gradient-free approaches have been applied,[4,12,13] the majority of flapping-flight optimization work appeals to gradient-based methods such as steepest

*Graduate Student, visiting University of Michigan
†Assistant Professor, AIAA Senior Member

descent. We employ a quasi-Newton approach in combination with a line search in our work. Here, adjoint-based methods allow for efficient computation of gradients of a small number of outputs with respect to a large number of design parameters.

II. Navier-Stokes Solver

A. ALE Mapping

We consider the Navier-Stokes equations in conservation form, using the notation $\mathbf{u}(\vec{x},t) \in \mathbb{R}^s$ for the state vector, $\vec{x} \in \mathbb{R}^d$ for the spatial coordinate, $t \in \mathbb{R}$ for the time, and $\vec{\mathbf{F}}^i$ and $\vec{\mathbf{F}}^v$ for the inviscid and viscous fluxes, respectively. Note that s=4 and d=2 in our work. Figure 1 summarizes the arbitrary Lagrangian-Eulerian mapping[14,15] of these equations from a deforming physical domain to a static reference domain, with mapped quantities in reference space denoted by the subscript X.

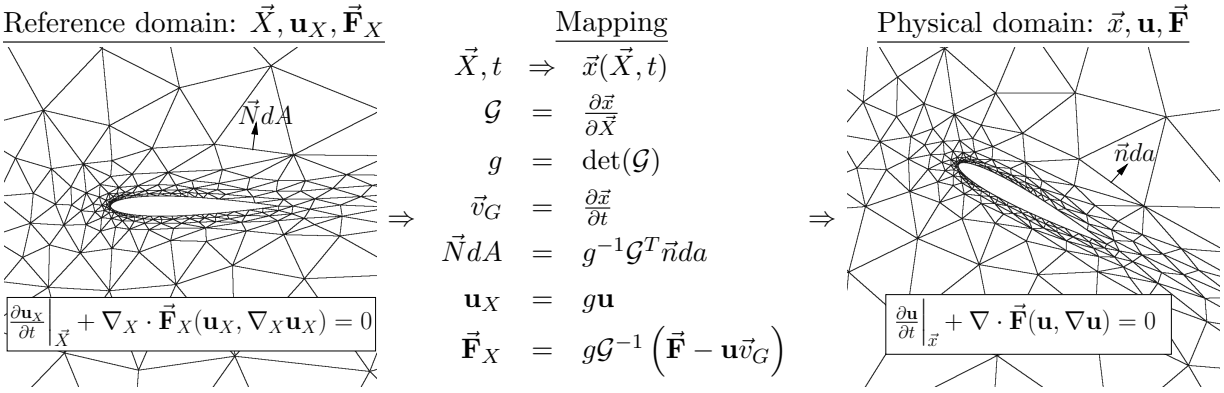


Figure 1. Summary of the mapping between reference and physical domains. The equations are solved on the reference domain, which remains fixed for all time.

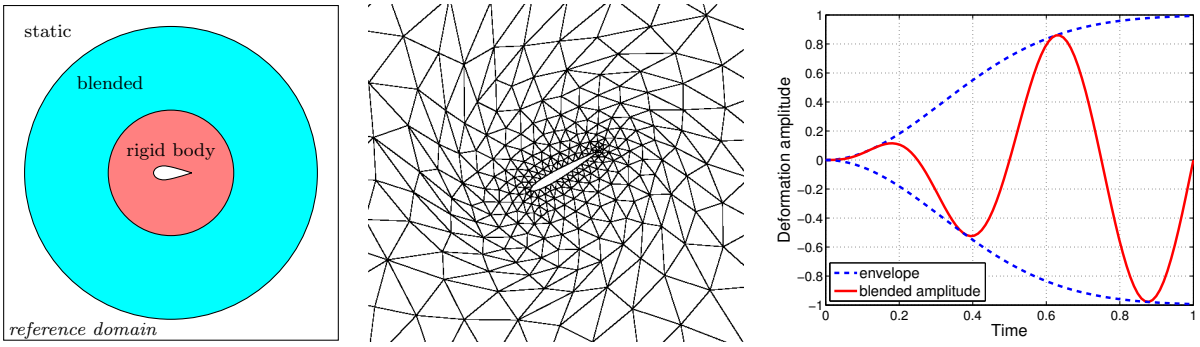
B. Analytical Deformations

Our ALE method employs an analytically-defined mapping between reference and physical domains. This mapping consists of spatially and temporally blended rigid-body motions. The optimization parameters enter into the definition of the rigid-body motion, which could be, for example, a superposition of pitch and plunge motions. We use piecewise-polynomial, radial blending in space,[14] illustrated in Figure 2a, to prevent the motion from affecting the farfield boundary. The resulting spatially-blended motion is given by

$$\vec{x} = b\vec{X} + (1-b)\vec{x}^{\text{rigid}} \tag{1}$$

where b is a spatial blending factor that is a polynomial function of the radial position away from the blending origin, and $\vec{x}^{\text{rigid}}$ is the rigid-body motion. In addition, we employ a temporal blending of the prescribed rigid-body motion at initial times to prevent impulsive start transients. The temporal blending envelope function, illustrated in Figure 2c, is given by

$$f^{\text{temporal}}(t) = 1 - e^{-(t/T_c)^2}, \tag{2}$$



(a) Spatial blending (b) Blended mesh (c) Temporal blending

Figure 2. Snapshots of primal and adjoint solutions.

where T_c is a blending time constant that could also be an optimization parameter. $f^{\text{temporal}}(t)$ modifies b in Eqn. 1, so that instead of b we use b', where

$$b' = 1 - f^{\text{temporal}} + bf^{\text{temporal}}.$$

C. Spatial and Temporal Discretization

We discretize the transformed Navier-Stokes equations in reference space using a discontinuous Galerkin method in space and backwards differencing in time. Spatially, we approximate the state using order p polynomials on each element of a tessellation of the domain. We use the Roe Riemann solver[16] for convective fluxes and the second form of Bassi and Rebay (BR2)[17] for the viscous treatment. Details on the spatial discretization can be found in previous work.[18,19] We note that the ALE formulation necessitates modifications to the Roe flux to accommodate grid velocity terms, and additional terms in the viscous discretization arising from the transformation of the state gradient.

In time, we use second-order backwards differencing. Specifically, the state at time node $n+1$, $\mathbf{U}^{n+1}$, is found by solving

$$\bar{\mathbf{R}}^{n+1} \equiv \mathbf{M}\frac{3\mathbf{U}^{n+1} - 4\mathbf{U}^{n} + \mathbf{U}^{n-1}}{2\Delta t} + \mathbf{R}(\mathbf{U}^{n+1}) = 0, \tag{3}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{R}$ is the spatial residual vector, and $\bar{\mathbf{R}}^{n+1}$ is the unsteady residual. We will consider time-integral outputs, and we calculate these using the trapezoidal rule – i.e. by approximating the state in between time nodes as linear in time.

III. Motion Optimization

A. Optimization method

We use a gradient-based optimization approach to locate an extremum of an output J, with respect to input parameters, $\boldsymbol{\mu}$. In the present study the output will be an aerodynamics performance scalar, such as imparted impulse, and the inputs will be parameters that govern the motion. We present the optimization as a minimization problem, with the understanding that maximizing J is equivalent to minimizing $-J$.

We employ a quasi-Newton optimization method to find a minimum of the objective function, J. At each optimization iteration, k, this method calculates a search direction based on the gradient and the Hessian matrix of the objective function. The search direction is defined as

$$\mathbf{p}_k = -\mathbf{H}^{-1}\left(\frac{\partial J}{\partial \boldsymbol{\mu}}\right)^T,$$

where $\mathbf{H}$ is the Hessian matrix. A true Newton method computes the Hessian at every function evaluation, while quasi-Newton methods update at every iteration an approximation to the Hessian matrix. There are several methods to update this matrix, and the BFGS formula is used in this work,

$$\mathbf{H}_{k+1}^{-1} = \left[\mathbf{I} - \frac{\mathbf{s}_k\mathbf{y}_k^T}{\mathbf{s}_k^T\mathbf{y}_k}\right]\mathbf{H}_k^{-1}\left[\mathbf{I} - \frac{\mathbf{y}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{y}_k}\right] + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{y}_k},$$

where

$$\mathbf{s}_k = \alpha_k\boldsymbol{p}_k, \quad \mathbf{y}_k = \left(\frac{\partial J}{\partial \boldsymbol{\mu}}\right)_{k+1}^T - \left(\frac{\partial J}{\partial \boldsymbol{\mu}}\right)_k^T.$$

Initially the Hessian is approximated by the identity matrix. With the calculated direction the parameters are updated using the following step,

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k\mathbf{p}_k.$$

The step size, α_k, of the update is determined by a backtracking line search. The starting value for α is 1 and each iteration it is multiplied with a factor until the Armijo rule[20] is satisfied,

$$J(\boldsymbol{\mu}_k + \alpha\mathbf{p}_k) \quad \leq \quad J(\boldsymbol{\mu}_k) + \beta_1\alpha\frac{\partial J}{\partial \boldsymbol{\mu}_k}\mathbf{p}_k.$$

The value of β_1 is set to 10^{-4}. Optimization iterations continue until the L_2 norm of the sensitivity gradient, $\frac{\partial J}{\partial \boldsymbol{\mu}}$ falls below a threshold value.

B. Energy Constraint

We use a constraint to keep the optimized motion away from non realistic/desirable regimes. In previous work[4] this has been done by constraining the total amount of power to sustain the motion of an airfoil. We employ a similar constraint by limiting the total amount of energy deposited into the flow during the course of the simulation. In a conservative scheme, this is equivalent to the time-integrated power required to move the airfoil, since this power ultimately ends up as kinetic or internal energy of the flow. When the time horizon of the simulation is short enough such that energy disturbances created by the airfoil do not leave the domain, the total energy at the final time, $t = T$, is calculated as $\mathcal{E}_T$, where

$$\mathcal{E}_t = \int_\Omega \rho E(t) d\Omega, \tag{4}$$

Ω is the computational domain, and $\rho E(t)$ is the total energy per unit volume at time t. The problem to be solved is,

$$\text{minimize } J(\boldsymbol{\mu}) \text{ subject to } \mathcal{E}_T(\boldsymbol{\mu}) \leq \mathcal{E}_T^{\max}, \tag{5}$$

where $\mathcal{E}_T^{\max}$ denotes the maximum allowed value for $\mathcal{E}_T(\boldsymbol{\mu})$. We transform this constrained optimization problem to one without constraints by applying a penalty function to the optimization problem,

$$\text{minimize } J(\boldsymbol{\mu}) + \frac{\rho}{2}\phi(\boldsymbol{\mu}), \tag{6}$$

where ρ is the penalty factor and $\phi(\boldsymbol{\mu})$ the penalty function defined by

$$\phi(\boldsymbol{\mu}) = \max(0, \mathcal{E}_T(\boldsymbol{\mu}) - \mathcal{E}_T^{\max})^2.$$

Eqn. 6 is solved multiple times with increasing ρ. When ρ approaches infinity the constraint will be satisfied. Solving the exact problem is not feasible due to finite computation time, and hence we deem the optimization converged when the following thresholds have been reached

$$||(\boldsymbol{\mu_{i-1}} - \boldsymbol{\mu_i})|| < c_1 \text{ and } (J(\boldsymbol{\mu_{i-1}}) - J(\boldsymbol{\mu_i})) < c_2. \tag{7}$$

Here $\boldsymbol{\mu_i}$ is the solution of Eqn. 6 subject to ρ_i. Thresholds c_1 and c_2 have been heuristically chosen to be .01.

IV. Adjoint Sensitivities

A. Unsteady Adjoint Solver

For a scalar output J, the adjoint system associated with Eqn. 3 is

$$\left(\frac{\partial \bar{\mathbf{R}}^i}{\partial \mathbf{U}^j}\right)^T \boldsymbol{\Psi}^i + \frac{\partial J}{\partial \mathbf{U}^j} = 0, \tag{8}$$

where i and j index time nodes, $1 \leq i, j, \leq N$ and $\boldsymbol{\Psi}^i$ is the adjoint vector at time node i. At the first time step we use first-order backwards-differencing, so that the adjoint system takes the form,

$$\begin{bmatrix} \frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \bar{\mathbf{R}}^1}{\partial \mathbf{U}^1}\right)^T & -2\frac{\mathbf{M}}{\Delta t} & \frac{1}{2}\frac{\mathbf{M}}{\Delta t} & 0 & 0 \\ 0 & \frac{3}{2}\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \bar{\mathbf{R}}^2}{\partial \mathbf{U}^2}\right)^T & -2\frac{\mathbf{M}}{\Delta t} & \ddots & 0 \\ 0 & 0 & \frac{3}{2}\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \bar{\mathbf{R}}^3}{\partial \mathbf{U}^3}\right)^T & \ddots & \frac{1}{2}\frac{\mathbf{M}}{\Delta t} \\ 0 & 0 & 0 & \ddots & -2\frac{\mathbf{M}}{\Delta t} \\ 0 & 0 & 0 & 0 & \frac{3}{2}\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial \bar{\mathbf{R}}^N}{\partial \mathbf{U}^N}\right)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}^1 \\ \boldsymbol{\Psi}^2 \\ \boldsymbol{\Psi}^3 \\ \vdots \\ \boldsymbol{\Psi}^N \end{bmatrix} + \begin{bmatrix} \frac{\partial J}{\partial \mathbf{U}^1} \\ \frac{\partial J}{\partial \mathbf{U}^2} \\ \frac{\partial J}{\partial \mathbf{U}^3} \\ \vdots \\ \frac{\partial J}{\partial \mathbf{U}^N} \end{bmatrix} = 0.$$

The derivatives used in the adjoint solve are computed in the same way as in the implicit linear solve. The same preconditioner, element-line Jacobi, is used to solve the transposed linear system in the adjoint discretization. Due to the upper-triangular nature of the adjoint system shown above, we solve the system using reverse time-marching. Calculation of the spatial residual Jacobian matrices requires the primal state, which we store to disk at each time iteration in the forward run.

B. Sensitivity Calculation

The adjoint solution gives us the sensitivity of the output J to changes in the residual. For optimization we require the sensitivity of J to parameters, $\boldsymbol{\mu}$. Employing the chain rule,

$$\frac{dJ}{d\boldsymbol{\mu}} = \sum_{i=1}^{N} \left(\boldsymbol{\Psi}^i\right)^T \frac{\partial \bar{\mathbf{R}}^i}{\partial \boldsymbol{\mu}} + \frac{\partial J}{\partial \boldsymbol{\mu}}, \tag{9}$$

where the final partial derivative includes any explicit dependence of the output on the parameters in $\boldsymbol{\mu}$. We note that the sensitivity calculation requires a summation over time nodes, i, which could in principle be done during the adjoint solve. However, for modularity and flexibility in choosing the parameters, we store all of the adjoint states during the adjoint solve and then post-process the adjoints on an as-needed basis. Finally, we obtain the partial derivatives of the unsteady residual with respect to the parameters using finite differences – these are cheap to evaluate relative to an entire flow solution.

V. Results

In this section we present results of our optimization framework. In all cases we try to maximize a transient thrust impulse divided by the simulation time – i.e. the time-averaged thrust. We first demonstrate that convergence is obtained for a simple two parameter case without constraints. We then show the effect of extra parameters combined with the energy constraint. Furthermore we investigate the effect of additional modes in the prescribed airfoil motion.

A. Transient Average Thrust Maximization

For all cases we consider a NACA 0012 airfoil in an initially stagnant fluid. At time $t = 0$ the airfoil begins combined pitching and plunging motion, blended in time according to Eqn. 2. The simulation time, in non-dimensional units, is from $t = 0$ to $t = 5$. Temporal blending is performed with a time constant of $T_c = 1$, and the spatial blending uses a quintic radial polynomial. The rigid-body pitch motion is of the form

$$\theta_{\text{pitch}}(t) = \theta_{\text{offset}} + \sum_{n=1}^{l} A_{\text{pitch},n} \sin(n\omega t + \phi_{\text{pitch},n}), \tag{10}$$

and the rigid-body plunge motion is

$$y_{\text{plunge}}(t) = y_{\text{offset}} + \sum_{n=1}^{l} A_{\text{plunge},n} \sin(n\omega t + \phi_{\text{plunge},n}). \tag{11}$$

In Eqn. 10 and Eqn. 11 l denotes the number of modes in the motion. We prescribe the frequency as $\omega = 0.8\pi$. The pitch motion is centered about $(x, y) = (\frac{1}{3}, 0)$, and the airfoil chord is $c = 1$. Values for θ_{offset} and y_{offset} are 0 and -0.25 respectively. We aim to maximize the average thrust which implies minimizing the average drag. The average drag is obtained by integrating the drag force over the simulation time divided by the simulation time,

$$J = \frac{1}{T} \int_0^T F_x dt, \tag{12}$$

where F_x is the x-component of the force on the airfoil.

B. Optimization of Two Parameters

In this test case we turn to optimize a simple two parameter problem to show that adjoints can be used to optimize an output scalar. This is done by showing the convergence of the output scalar and the parameters.

1. *Simulation*

The the computational mesh for this case is a coarse grid which is illustrated in Figure 5(a), it consists of 533 triangular elements, curved using cubic geometry approximation on the airfoil. The interpolation order of the elements is $p = 1$ and $N = 100$ timesteps are used. We only consider the first modes in Eqn. 10 and Eqn. 11. For this simulation Eqn. 12 is integrated from 2.5 to T to avoid transient effects, and the energy constraint is not enforced. Only two parameters are varied, the pitch amplitude A_{pitch} and the pitch phase ϕ_{pitch}. We prescribe the plunge amplitude as $A_{\text{plunge}} = 0.25$ and the plunge phase as $\phi_{\text{plunge}} = \pi/2$. Because of the simplicity of this problem we are able to carry out a "brute-force" sweep of the parameter domain. The optimization starts with $\phi_{\text{pitch}} = 0$ and $A_{\text{pitch}} = 0.5$.

2. *Convergence Results*

For the "brute-force" sweep we use a 10×10 grid of parameters, which gives us a contour plot of the output. On this contour plot, given in Figure 3, we overlay the result of the gradient-based optimization. In Figure 4, we show the convergence of the thrust, $-J$, and of the norm of the

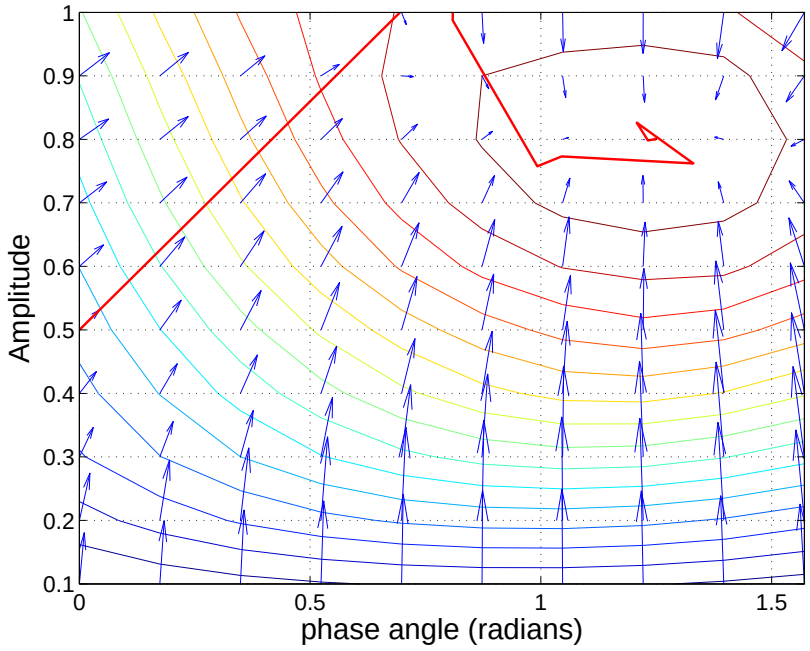
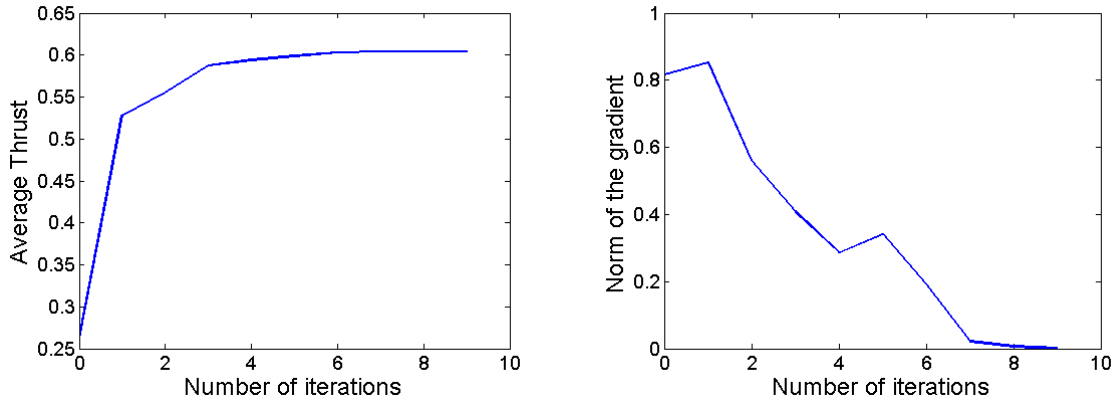


Figure 3. Contour plot of the thrust impulse output for a sweep of parameter values. Results of the optimization, showing convergence to the maximum in a handful of iterations, are overlaid on this contour plot.

thrust gradient, $\| - \frac{\partial J}{\partial \boldsymbol{\mu}} \|$, with optimization iteration. The flattening out of the output and the gradient confirm the ability of the gradient-based optimizer to locate the output extremum. A note must be made that when transient effects are included, thus Eqn. 12 is integrated from 0 to T, the optimum disappears and the optimization does not converge when unconstrained.



(a) Output (b) Sensitivity gradient norm

Figure 4. Convergence results for the output and sensitivity gradient with respect to optimization iterations.

C. Additional Parameters

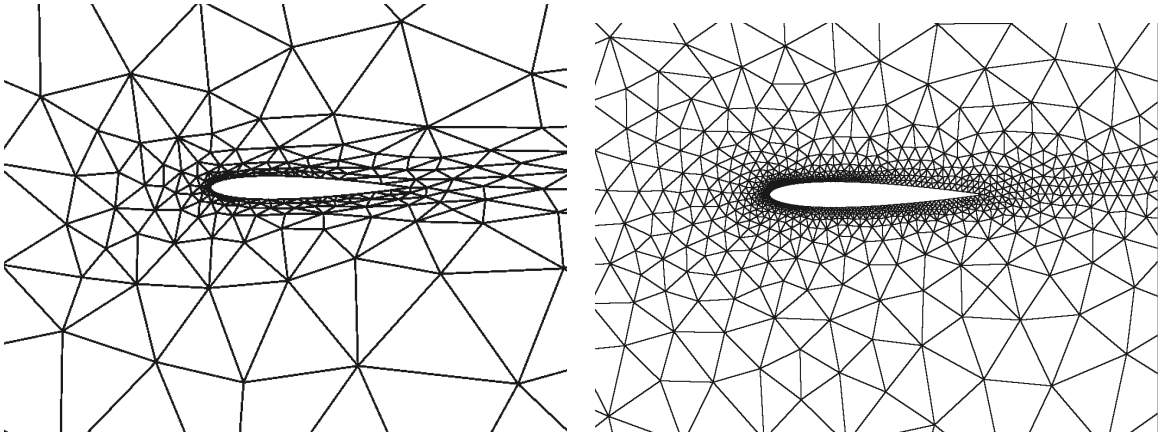
We expand the number of parameters in this test case to a four-parameter optimization problem. The additional parameters might yield nonrealistic motion solutions, where the power requirements may become very large. The energy constraint is thus applied and verified.

1. Simulation

We use a finer computational mesh of 2573 triangular elements for this case, as illustrated in Figure 5(b). The interpolation order is now set to $p = 2$ and there are $N = 100$ timesteps. The motion of the airfoil is described in Eqn. 10 and Eqn. 11 with $l = 1$. Now pitch and plunge amplitudes and phases are regarded as parameters. The value for $\mathcal{E}_T^{\max}$ is chosen heuristically to ensure convergence of the solution. The starting values for the parameters are converged values of a coarse grid simulation, as given in Table 1.

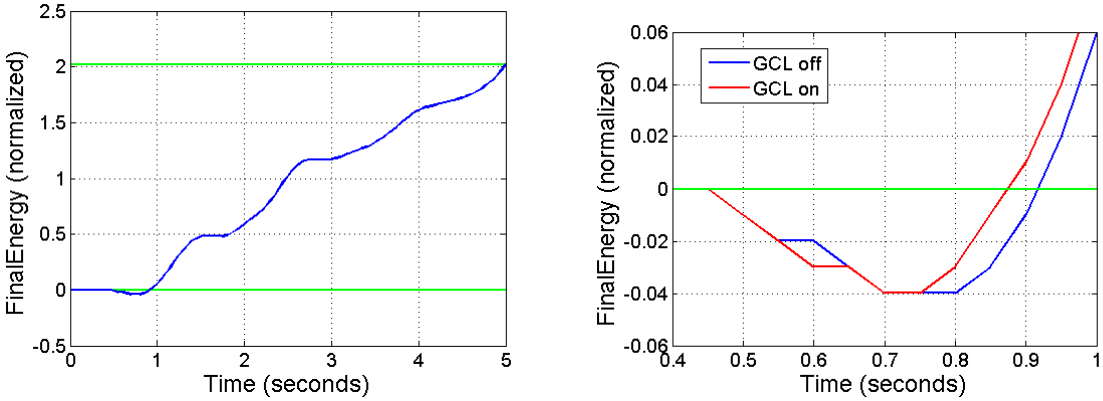
2. Convergence results and constraint verification

The results of optimization exhibit convergence in the average thrust output as a function of the penalty value ρ, as shown in Figure 7(d). The average thrust flattens out for $\rho = 64$ and higher to a value of 0.276. This implies that increasing ρ does not influence the output anymore and thus the energy constraint is satisfied. In Figure 6(a) the value $\mathcal{E}_t - \mathcal{E}_0$ is plotted with the upper and lower boundaries of, respectively, $\mathcal{E}_T^{\max} - \mathcal{E}_0$ and 0. It shows that the energy constraint is indeed satisfied. A note must be made that at the start of the simulation the energy dives below $\mathcal{E}_0$ which is not possible physically nor numerically when using a conservative scheme. The explanation for this is that the scheme is not exactly conservative due to geometric conservation errors in the ALE mapping. To improve the energy conservation, a geometric conservation law (GCL) equation can be used;[14,15] however all optimizations are done without enforcing the GCL to avoid complications in the adjoint calculation. Nevertheless, to demonstrate the effect of enforcing a GCL, in Figure 6(b) we plot results with and without the GCL. Note that turning the GCL on slightly improves the non-physical "dip" in the energy, although it does not eliminate it.



(a) Coarse Mesh (b) Fine Mesh

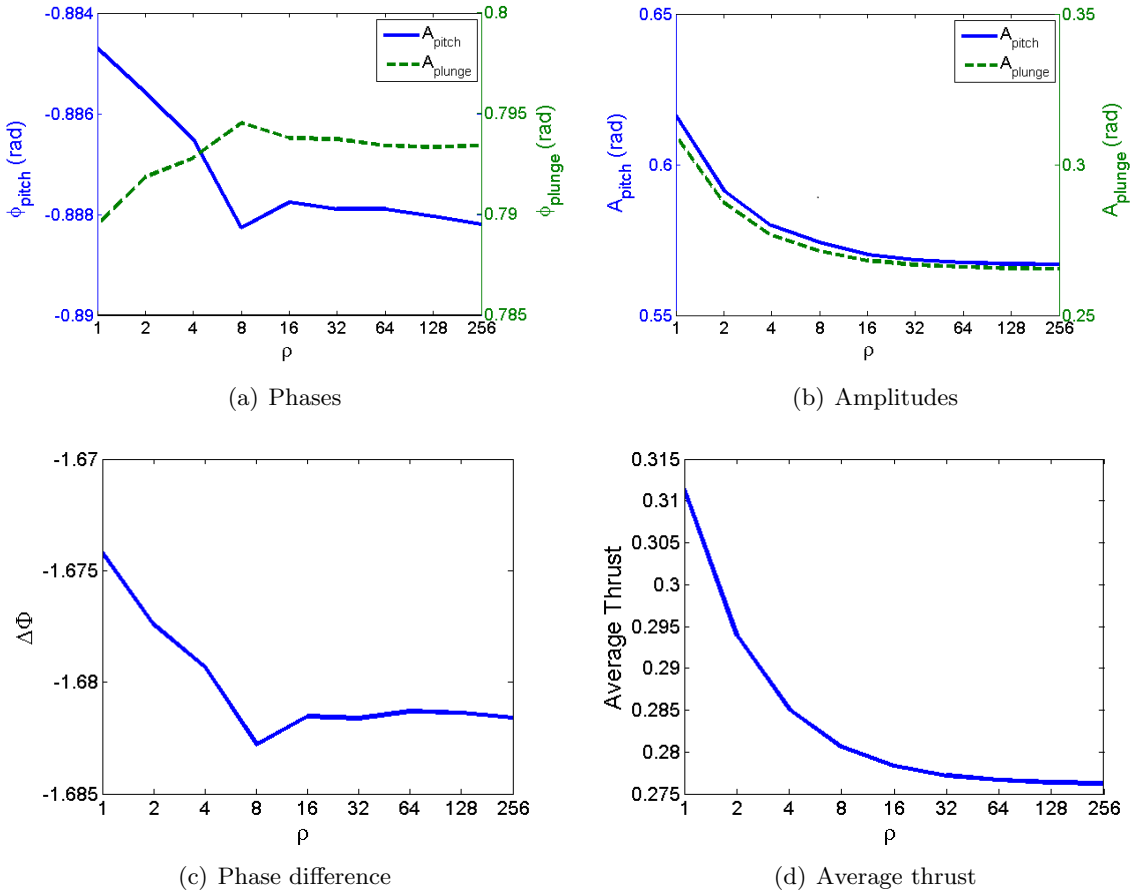
Figure 5. Computational meshes used in the flow solver.



(a) (b)

Figure 6. Total energy inside the computational domain as function of time

In Figure 7(a) the optimized values for the phases are plotted as functions of ρ. In Figure 7(b) this is done for the amplitudes. Clear convergence can be seen for the amplitudes and phases. The phases are correlated: the phase difference, $\Delta\Phi = \Phi_{\text{pitch}} - \Phi_{\text{plunge}}$, is plotted in Figure 7(c). The value approaches approximately -1.68 rad. The optimized parameters are given in Table 1.



(a) Phases (b) Amplitudes

(c) Phase difference (d) Average thrust

Figure 7. Plots of the phases, amplitudes, phase difference and average thrust as function of the penalty factor ρ for the four-parameter case.

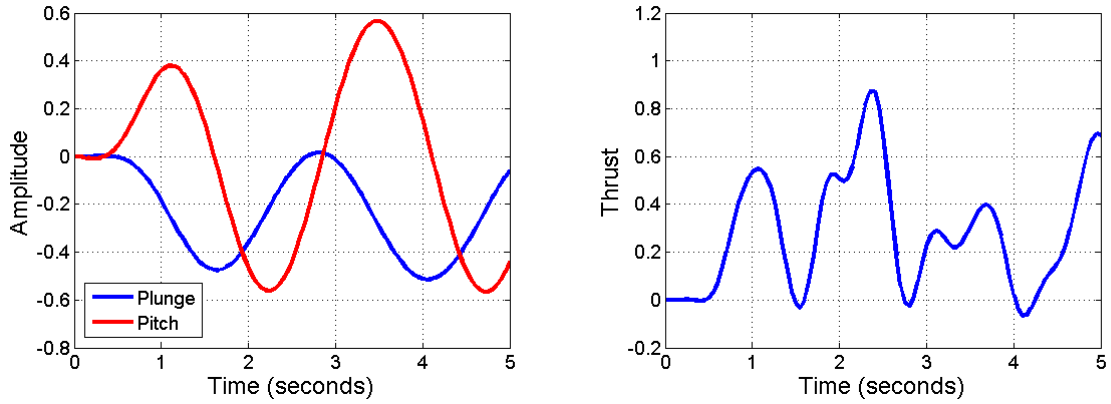
Parameter	Start value	Optimized value
A_{pitch}	0.616	0.567
Φ_{pitch}	-0.884	-0.888
A_{plunge}	0.309	0.265
Φ_{plunge}	0.789	0.793

Table 1.

3. Motion and force analysis

The optimized motion that has been found combined with the resulting thrust on the airfoil is plotted in Figure 8. The peaks in the thrust seem to correspond to the extremum of the pitch motion and the local minima where the thrust is about zero correspond to the extremum of the plunge motion. There are double peaks at around $t = 2$ and $t = 3, 5$. Why this happens can be seen in Figure 9 where snapshots of the entropy in the flow are shown. The entropy shows locations

of vortices. At $t = 1.05$, when the first peak in the thrust plot appears, a leading edge and a trailing edge vortex are being shed creating thrust. The double peak around $t = 2$ seconds can be explained in Figure 9(c) and Figure 9(d). The shedding of leading and trailing edge vortexes are not simultaneous resulting in two peaks: at $t = 3.30$ a leading edge vortex travels from the leading edge to the trailing edge.



(a) Kinematics (b) Thrust

Figure 8. Kinematics of the optimized motion and the resulting thrust as function of time.

D. Additional Modes

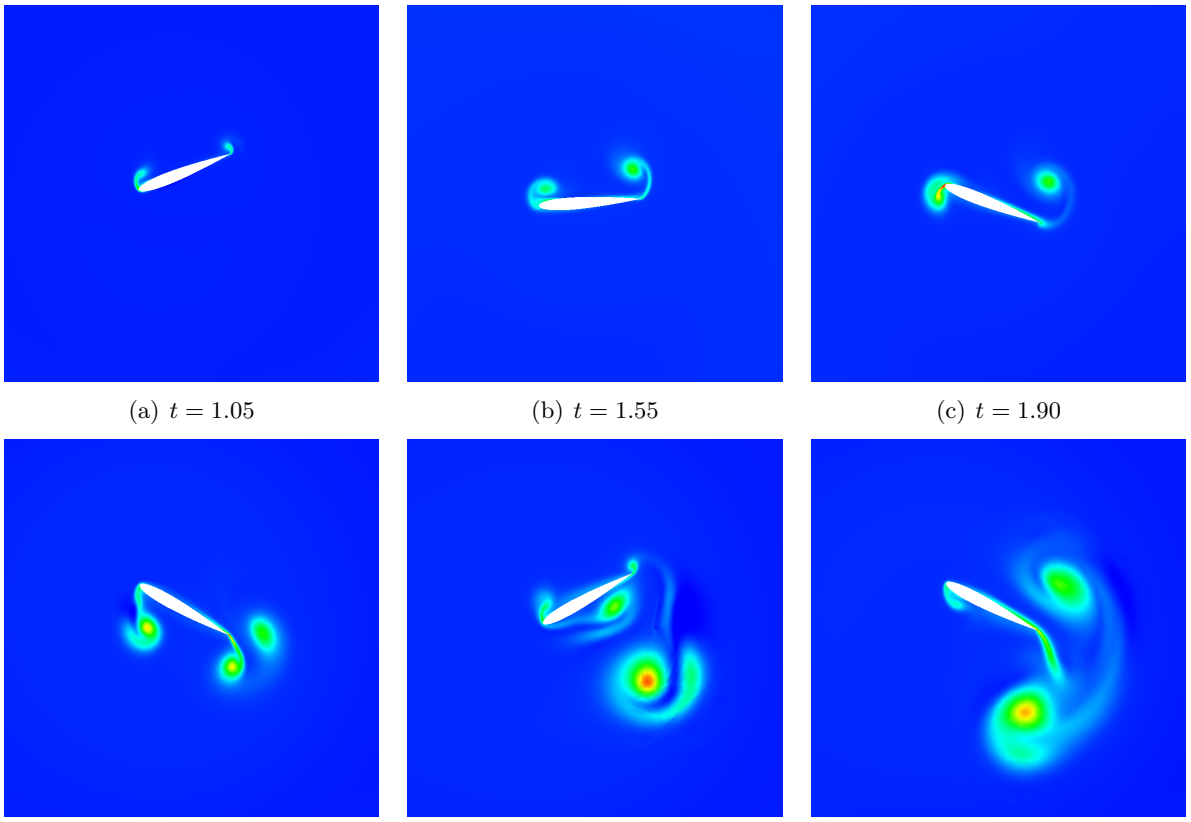
In the previous two cases we have shown that our method can successfully converge certain cases of interest using a few parameters. We now want to investigate what will happen if we increase the number of parameters. The results will be compared with the four parameter case.

1. *Simulation*

The same computational grid is used as in the four parameter case, which is show in Figure 5(b). The interpolation order and number of timesteps are also the same: $p = 2$ and $N = 100$. For the analytical motion we consider Eqn. 10 and Eqn. 11 again, now with $l = 3$ resulting in a motion governed by 12 parameters. The value for $\mathcal{E}_T^{\max}$ is the same as in the previous case. The starting values for the parameters are given in Table 2. These values are chosen according to the parameters from the four-parameter case, based on the expectation that the amplitude of higher modes decay.

2. *Optimization results*

The optimization leads to an average thrust of 0.305. This is 10% higher than the average thrust of 0.276 found in the four parameter case. As expected, adding additional modes thus increases the thrust generation for the same amount of energy expended. The converged optimization values are given in Table 2. The amplitude of each mode decreases with increasing mode number as was expected. Higher amplitudes cost more energy for higher mode numbers compared to lower mode numbers, and hence these are not favored because of the energy constraint in the optimization.



(a) $t = 1.05$ (b) $t = 1.55$ (c) $t = 1.90$

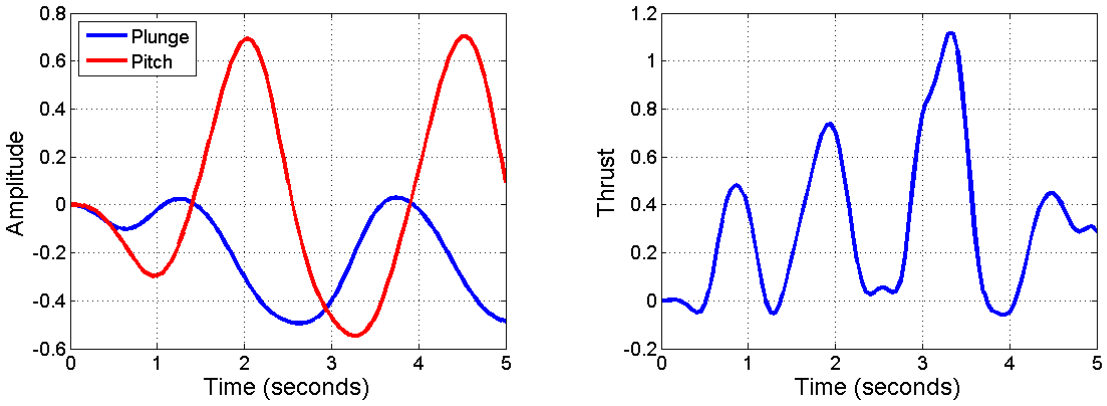
(d) $t = 2.40$ (e) $t = 3.30$ (f) $t = 4.95$

Figure 9. Snapshots of the entropy in the flow.

Parameter	Start value	Optimized value
$A_{\text{pitch},1}$	0.5	0.614
$A_{\text{pitch},2}$	0.1	0.079
$A_{\text{pitch},3}$	0.05	-0.024
$\Phi_{\text{pitch},1}$	1.571	2.857
$\Phi_{\text{pitch},2}$	0	3.942
$\Phi_{\text{pitch},3}$	0	-5.178
$A_{\text{plunge},1}$	0.3	0.268
$A_{\text{plunge},2}$	0.05	0.024
$A_{\text{plunge},3}$	0.001	0.009
$\Phi_{\text{plunge},1}$	0	4.568
$\Phi_{\text{plunge},2}$	0	2.065
$\Phi_{\text{plunge},3}$	0	0.761

Table 2.

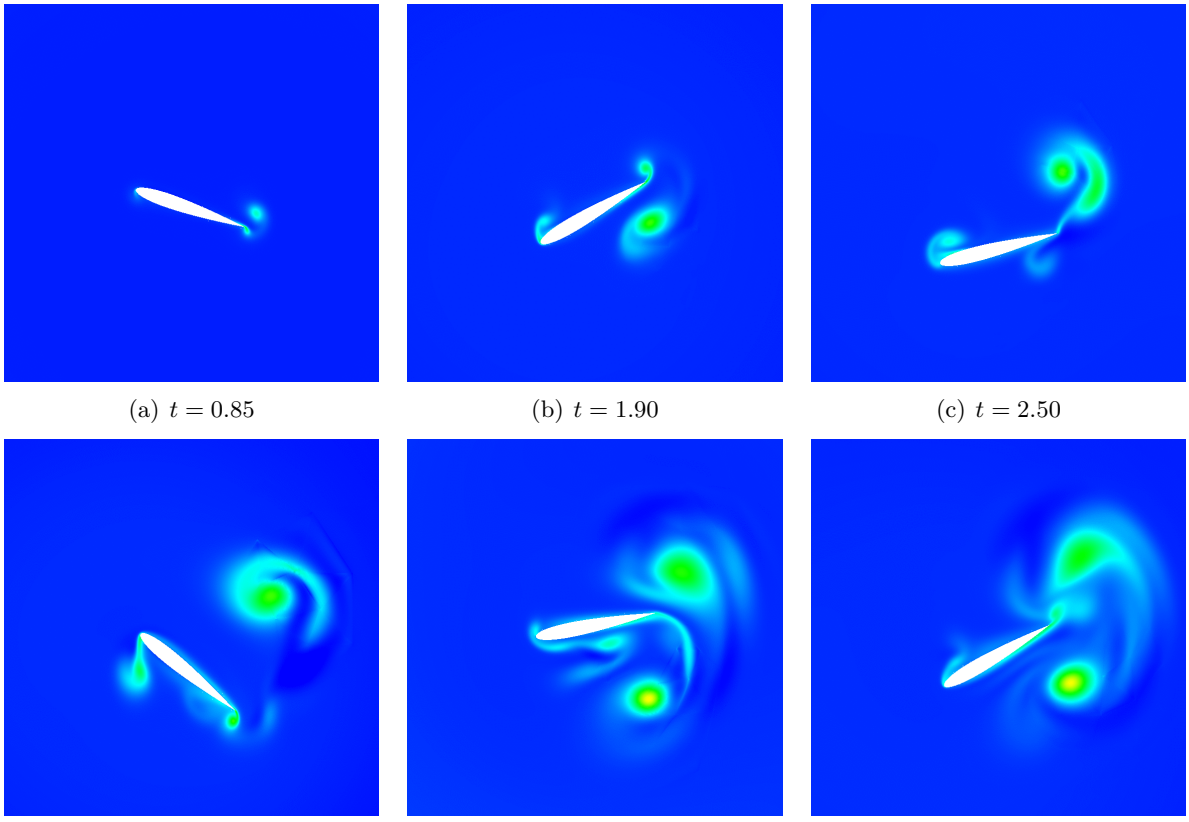
The value of $A_{\text{plunge},1}$ is exactly the same as in the four-parameter case, however the amplitude $A_{\text{pitch},1}$ is higher. In Figure 10(a) the motion of the airfoil has been plotted. Comparing this with Figure 8(a) it can be seen that the overall motion has shifted by about 1 second. The phase difference between the first modes of pitch and plunge is almost the same. The peaks of the pitch in this case are more slender. Extra modes make it possible to increase the pitch amplitudes with the same amount of energy, which results in more thrust. Flow structures can be seen Figure 11,



(a) Kinematics (b) Thrust

Figure 10. Kinematics of the optimized motion and the resulting thrust as function of time.

where snapshots of the entropy are given. The first peak that can be seen in Figure 10 is lower than the second peak. This can be explained by Figure 11(a) where only a vortex at the training edge is being shed, while in Figure 11(b) both the leading and trailing edges vortices are shed. At $t = 2.50$ a counter vortex is under the airfoil.



(a) $t = 0.85$ (b) $t = 1.90$ (c) $t = 2.50$

(d) $t = 3.20$ (e) $t = 3.95$ (f) $t = 4.50$

Figure 11. Snapshots of the entropy in the flow.

VI. Conclusions

An adjoint-based optimization method for a high-order finite element discretization of the Navier-stokes equations on deformable domains has successfully been applied to a flapping airfoil. Constrained by a given amount of energy the pitch and plunge motions of the airfoil have been optimized. The method lets us study a relatively large number of parameters, and hence general motions. Thrust generation at a given energy expenditure is significantly increased by adding higher-frequency sinusoidal modes to the airfoil motion. Next steps include coupling with a structural deformation model and extension to three dimensions, where we have already used the adjoint for error estimation and mesh adaptation.[21]

VII. Acknowledgements

The authors are thankful for support from the University of Twente Internship Program in Mechanical Engineering, and from the University of Michigan.

References

[1]Hall, K. C., Pigott, S. A., and Hall, S. R., "Power requirements for large-amplitude flapping flight," *Journal of Aircraft*, Vol. 35, No. 3, 1998, pp. 352–361.

[2]Willis, D. J., Israeli, E. R., Persson, P.-O., Drela, M., Peraire, J., Swartz, S. M., and Breuer, K. S., "A computational framework for fluid structure interaction in biologically inspired flapping flight," AIAA Paper 2007-3803, 2007.

[3]Tuncer, I. H. and Kaya, M., "Optimization of flapping airfoils for maximum thrust and propulsive efficiency," *AIAA Journal*, Vol. 43, No. 11, 2005, pp. 2329–2336.

[4]Pesavento, U. and Wang, Z. J., "Flapping wing flight can save aerodynamic poser compared to steady flight," *Physical Review Letters*, Vol. 103, No. 118102, 2009, pp. 1–4.

[5]Soueid, H., Guglielmini, L., Airiau, C., and Bottaro, A., "Optimization of flapping airfoils using sensitivity functions," *Computers and Fluids*, Vol. 38, 2009, pp. 861–874.

[6]Dong, H., Liang, Z., and Harff, M., "Optimal settings of aerodynamic performance parameters in hovering flight," *International Journal of Micro Air Vehicles*, Vol. 1, No. 3, 2009, pp. 173–181.

[7]Eldredge, J. D., Toomey, J., and Medina, A., "On the roles of chord-wise flexibility in a flapping wing with hovering kinematics," *Journal of Fluid Mechanics*, Vol. 659, 2010, pp. 94–115.

[8]Ou, K. and Jameson, A., "Optimization of flow past a moving deformable airfoil using spectral difference method," AIAA Paper 2011-3719, 2011.

[9]Persson, P.-O. and Willis, D. J., "High fidelity simulations of flapping wings designed for energetically optimal flight," AIAA Paper 2011-568, 2011.

[10]Culbreth, M., Allaneau, Y., and Jameson, A., "High-fidelity optimization of flapping airfoils and wings," AIAA Paper 2011-3521, 2011.

[11]Jones, M. and Yamaleev, N., "Adjoint-based optimization of flapping wing performance," Seventh International Conference on Computational Fluid Dynamics ICCFD7-2403, 2012.

[12]Milano, M. and Gharib, M., "Uncovering the physics of flapping flat plates with artificial evolution," *Journal of Fluid Mechanics*, Vol. 534, 2005, pp. 403–409.

[13]Hamdaoui, M., Mouret, J.-B., Doncieux, S., and Sagaut, P., "Optimization of kinematics for birds and UAV's using evolutionary algorithms," Proceedings of the World Academy of Science, Engineering, and Technology, 2008.

[14]Persson, P.-O., Bonet, J., and Peraire, J., "Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, 2009, pp. 1585–1595.

[15]Kast, S. M., Ceze, M. A., and Fidkowski, K. J., "Output-adaptive solution strategies for unsteady aerodynamics on deformable domains," Seventh International Conference on Computational Fluid Dynamics ICCFD7-3802, 2012.

[16]Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[17]Bassi, F. and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

[18]Fidkowski, K. J. and Roe, P. L., "An entropy adjoint approach to mesh refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.

[19]Fidkowski, K. J. and Luo, Y., "Output-based space-time mesh adaptation for the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 230, 2011, pp. 5753–5773.

[20]Armijo, L., "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, Vol. 16, No. 1, 1966, pp. 1–3.

[21]Kast, S. M. and Fidkowski, K. J., "Output-based mesh adaptation for high order navier-stokes simulations on deformable domains," *Journal of Computational Physics*, 2013, Accepted.