51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0839

# Using adjoint-based approach to study flapping wings

Min Xu*, Mingjun Wei†
*New Mexico State University, Las Cruces, NM 88003*

**Adjoint-based methods show great potential in flow control and optimization of complex problems with high- or infinite-dimensional control space. It is attractive to solve an adjoint problem to understand the complex effects from multiple control parameters to a few performance indicators of the flight of birds or insects. However, the traditional approach to formulate the adjoint problem becomes either impossible or too complex when arbitrary moving boundary (e.g. flapping wings) and its perturbation is considered. Here, we use non-cylindrical calculus to define the perturbation. So that, a simple adjoint system can be derived directly in the inertial coordinate. The approach is applied to optimize the vertical motion of an oscillatory cylinder to match a pre-defined downstream velocity profile. The result quickly converges to the analytical solution.**



## I. Introduction

Recently, direct numerical simulation (DNS) contributed successfully in flapping-wing research by providing detail information of flow field [1–4]. With such high-fidelity simulations, it becomes possible for people to obtain almost the full knowledge of the flow, such as vortex structures, force distribution, and the correlation between those qualities. Numerical simulation also provides the convenience of removing biological limitation and alternating configuration to separate/simplify physical problem or to design man-made flying vehicles. However, even the most perfect simulation can only provide good observation or measurement. More insights are expected if we can compute a higher-order quality, the *derivative of such an observation.*

To move from a vague idea of obtaining the *derivative of flow field* to a well-defined mathematical problem, we need to define a clear objective (i.e. cost function) as a function of control parameters. In the current study of flapping flexible wings, there are almost infinite choices of wing shape and size, flapping trajectory, material property, and so on; on the other hand, there are only a handful of objectives, such as lift/drag performance, maneuverability, instability, and robustness. Such a huge input space makes any direct approach for sensitivity study extremely expensive to cover the entire parametric space. However, when the sensitivity analysis is based on adjoint calculation, the scenario is reversed. The cost to compute the sensitivity by adjoint method is based on the size of outputs (instead of inputs) in the original physical problem. For a single objective (cost function), sensitivity information for all input parameters with infinite space-time choices can be obtained by one single adjoint computation. Consequently, the total cost to obtain all sensitivities is independent of the number of control parameters. Therefore, adjoint-based method is efficient and suitable in analyzing the problem with large input space and small number of objectives, which is the case in the study of flapping wings.

Previously, the adjoint-based method has been used in many researches including sensitivity and shape optimization of airfoils [5–10], topology optimization for channels [11], minimization of dispersion of microchannel bends [12], error estimation for discontinuous Galerkin approximations of hyperbolic systems [13], studies of drag reduction in a turbulent channel flow [14], analysis of mean flow refraction effects on sound radiated by localized sources in jets [15], receptivity prediction in nonparallel flows [16], control of unsteady compressible flow properties [17–19], model reduction [20, 21], jet noise control [22, 23], and many others.

In this paper, we are extending the adjoint-based method for the optimization of problems with dynamical moving and morphing boundary/shape. The method is then applied on the optimization of oscillatory cylinders with regular and arbitrary motion.

*Research Assistant, Department of Mechanical and Aerospace Engineering
†Associate Professor, Department of Mechanical and Aerospace Engineering, Senior Member AIAA

## II. Numerical simulation in physical space

An accurate numerical simulation plays a key role in the proposed work and provides its own many challenges in issues such as moving boundary and fully-coupled fluid-structure interaction. By their distinctive characters, fluids and solids are conventionally treated separately as distinct materials and in different computational domains [24–27]. As the result, the body-fitted mesh resolving the fluid parts needs to move and deform accordingly with the fluid-soild interface. The grid regeneration can be very expensive and sometimes fragile in large deforming. The interpolation of variables to the new mesh can also be computationally expensive [26]. Therefore, several approaches using non-body-fitted mesh become attractive because of the simplicity in resolving moving body or body with complex geometries. These approaches include distributed Lagrange multiplier [28], immersed boundary method [1, 29–38], and immersed-interface method [39–41]. Among all these, immersed boundary method stands out by its relatively simple implementation and outstanding flexibility in handling moving and complex boundary/geometry. Starting from the original idea proposed by Peskin [31] to use body force to mimic solid boundary, different variations of immersed boundary (a.k.a. embedded boundary) approaches have been developed [33]. For flapping wing problems, most immersed boundary approaches apply for prescribed motion and deformation [1, 33, 34, 36–38]. When fluid-structure interaction is involved, normal approach is to solve fluid flow and solid structure separately in two solvers, and the information exchanges continuously between fluid and solid domain at the interface [4]. However, since fluid and solid are treated by two distinct solvers, the convergency by iterations is not guaranteed. Based on Zhao et. al's monolithic algorithm to solve fluid-solid system [42] (similar idea was suggested in Boffi et. al's work [43]), we recently developed an approach to handle flapping flexible wings in the same spirit of immersed boundary method [3]. The basic idea is to solve fluid motion and solid motion (including both prescribed and fully-coupled) monolithically in a combined equation in globally Eulerian framework, where the solid wing structure is embedded using immersed boundary technique.

With details being referred to our previous work [3] and Zhao et. al's work [42], the modified momentum equation used in our computation is

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{Re_f}\nabla^2\mathbf{u} + \nabla\cdot(\chi_s\boldsymbol{\tau}_{elas}) + \chi_c f. \tag{1}$$

Using a characteristic function $\chi_s$ defined by

$$\chi_s = \begin{cases} 1 & in\ \ \Omega_s \\ 0 & otherwise \end{cases}, \tag{2}$$

we have specified a "solid region" $\Omega_s$ to apply elastic stress $\tau_{elas}$. The region will be tracked throughout the computation. Similarly, a characteristic function $\chi_c$ is defined for a "control region" $\Omega_c$ where we control/prescribe the desired moving trajectory. In the control region, we use a body force term $f$ to define the moving trajectory following a typical direct-forcing approach [36, 44]. We consider the momentum equation being discretized in time as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\triangle t} = (\mathrm{RHS})^n, \tag{3}$$

where all right-hand-side terms are lumped nominally into (RHS). Here, first-order explicit time difference is used only for easy demonstration, and in practice, we use third-order Runge-Kutta/Crank-Nicolson scheme. The idea of direct-forcing is to use a body force term confined in the control region $\Omega_c$ as

$$f = \begin{cases} -(\mathrm{RHS})^n + \frac{1}{\triangle t}(\mathbf{V}-\mathbf{u}^n) & in\ \ \Omega_c \\ 0 & otherwise \end{cases}, \tag{4}$$

so that, an exact moving trajectory is defined by a prescribed velocity $\mathbf{V}(t)$. In the actual implementation of direct-forcing, interpolations are used to achieve high-order accuracy at boundary [1, 34].

If the "control region" is the same as the "solid region", the motion of all solid cells is prescribed. So, the entire case becomes a problem with prescribed motion and deformation. In fact, in the current paper, the adjoint method is only applied on the optimization of moving objects with prescribed motion. So, we basically turned off the elasticity term and made the control region be the whole solid area, though we used the same code as in [3] for flexible wings.

## III. Formulation and analysis in adjoint space

Typically, there are two approaches (or a hybrid of them) to derive adjoint equations. Based on the order of discretization and adjoint-formulation, they are called respectively continuous (or optimize-then-discretize) and discrete (or discretize-then optimize) approaches. The difference and the pros/cons have been discussed in many occasions [18, 45]. Because of the delicate immersed boundary method used for the simulation of flapping flexible wings, the coefficient matrix required by the discrete approach can be too complex or even be impossible in our case. By choosing the continuous approach, we work directly on the governing equation (e.g. Navier-Stokes equation) without worrying about the numerical details such as finite difference scheme and boundary treatment. At the same time, the continuous approach helps to keep certain "meaning" (e.g. convection, damping, ...) of the adjoint equation. Another strategy being taken here for simplicity is that we do *not* consider the immersed boundary method during the derivation of the adjoint problem even though it becomes part of our modified governing equation. Instead, we consider the flapping wing problem as a moving boundary problem with conventional boundary conditions for solid wall. Once the adjoint equation is derived, we then implement the immersed boundary method as a simulation technique to solve the adjoint problem in the same way as it is implemented for the physical problem. With the above considerations, the process to formulate the adjoint problem is largely simplified, and at the same time the entire methodology avoids the specification of numerical techniques. Thus, the method is general enough to be implemented in a similar manner for simulations using different numerical method.



### III.A. General concept for adjoint-based analysis and optimization

Following loosely the notation by Bewley et al. [14] and later by Wei et al. [23,46], the basic idea of continuous approach for the adjoint equation is shown below. First, a mathematically well-defined cost function $\mathcal{J}(\phi)$ is needed to start the sensitivity analysis or the optimization, where $\phi$ is a vector of control parameters. With the cost function $\mathcal{J}$ being defined, the sensitivity analysis will study the perturbed function $\mathcal{J}'$ that results from an arbitrary perturbation $\phi'$ to the control $\phi$. $\mathcal{J}'$ is defined as a differential of cost function,

$$\mathcal{J}' \equiv \lim_{\varepsilon \to 0} \frac{\mathcal{J}(\phi + \varepsilon\phi') - \mathcal{J}(\phi)}{\varepsilon} \tag{5}$$

Though we can supposedly use the above definition to start the sensitivity analysis by assuming all possible perturbations and computing the resulted changes (e.g. genetic algorithms [47]), the rapid increase of computational cost with the increase of control variables makes this simple idea impossible for most high-dimensional fluid mechanics problems. Alternatively, an adjoint field based on a similar differentiation can be defined for the original problem or the linearly-perturbed problem if the original problem is described by non-linear equations (i.e. Navier-Stokes equations). If the Navier-Stokes equations are symbolically represented as

$$\mathcal{N}(\mathbf{q}) = \mathbf{F}, \tag{6}$$

where the primary variable $\mathbf{q} = [p\ \mathbf{u}]^T$, the perturbed field is then

$$\mathcal{N}'(\mathbf{q}') = \mathbf{F}', \tag{7}$$

where $\mathbf{q}'$ is defined by the same differentiation used for the cost function. The adjoint field therefore can be defined by integration by parts,

$$\langle \mathcal{N}'(\mathbf{q})\mathbf{q}', \mathbf{q}^* \rangle = -\langle \mathbf{q}', \mathcal{N}^*(\mathbf{q})\mathbf{q}^* \rangle + b. \tag{8}$$

With appropriate boundary conditions and right-hand-side terms $\mathbf{F}^*$, the adjoint problem is described by the equation,

$$\mathcal{N}^*(\mathbf{q}^*) = \mathbf{F}^*. \tag{9}$$

Solving the adjoint equation once can achieve the sensitivity of the cost function to all space-time control variables listed in $\phi$. By such a nature of adjoint-based analysis, the increase of computational cost by the increase of dimensions in $\phi$ is negligible. For a single calculation, the time for solving the adjoint equation is normally close to the time for solving the original flow equation.

When optimization is considered, the same sensitivity information provides the gradient $g(\phi)$ to update the control variables iteratively as

$$\phi^{\text{new}} = \phi^{\text{old}} - \alpha g(\phi^{\text{old}}), \tag{10}$$

The cost of the optimization process is mainly determined by the convergence speed as for a typical gradient-based method. Conjugate gradient method and other methods to accelerate the convergence are normally involved.

### III.B. Adjoint problem for flapping wings

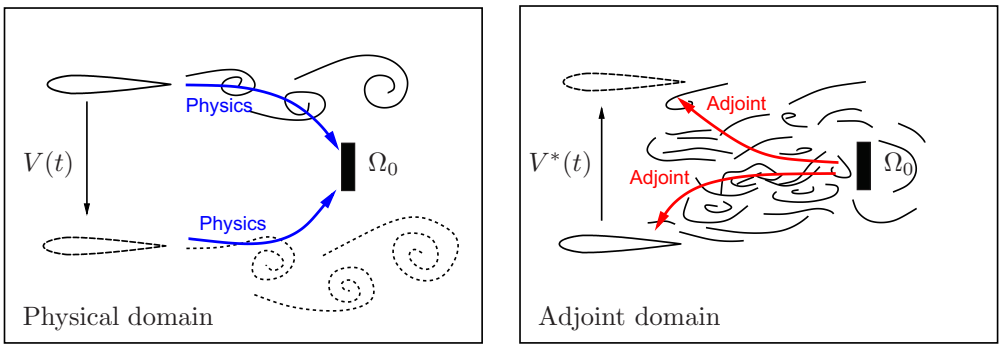


**Figure 1. A demonstration of the adjoint problem being formulated for a slowly plunging airfoil: left) physical space; right) adjoint space.**

For better explanation of the approach, the following discussion will be based on a simple demonstration shown in figure 1. However, the derivation is general enough for any problem with moving and morphing boundary or shape. Here, we consider an airfoil plunging or pitching with prescribed velocity $V(t, \phi, s)$, where $s$ is the arc-length coordinate and $\phi$ is the arbitrary control. The airfoil with flapping motion results in vortex shedding continuously. For demonstration purpose in this section, we pick a simple objective to minimize the overall difference between velocity $\mathbf{u}$ at a downstream region $\Omega_o$ and the target velocity $\mathbf{u}_0$ for a time duration $(0, T)$. A reasonable cost function for this objective is

$$\mathcal{J} = \int_0^T \int_{\Omega_o} |\mathbf{u} - \mathbf{u}_0|^2 d\mathbf{x}dt, \tag{11}$$

By the definition in (5), we have

$$\mathcal{J}' = \int_0^T \int_{\Omega_o} 2(\mathbf{u} - \mathbf{u}_0)\mathbf{u}' d\mathbf{x}dt = \int_0^T \int_{\mathcal{S}} g(\phi)\phi' d\mathbf{x}dt, \tag{12}$$

where the gradient (i.e. sensitivity information) $g(\phi)$ is implied and $\mathcal{S}$ denotes the moving solid surface. The physical problem is described by (6) with

$$\mathcal{N}(\mathbf{q}) = \begin{bmatrix} \dfrac{\partial u_j}{\partial x_j} \\ \dfrac{\partial u_i}{\partial t} + \dfrac{\partial u_j u_i}{\partial x_j} - \nu \dfrac{\partial^2 u_i}{\partial x_j^2} + \dfrac{\partial p}{\partial x_i} \end{bmatrix}, \qquad \mathbf{F} = 0. \tag{13}$$

and the boundary condition at $\mathcal{S}$

$$u_i = V_i \tag{14}$$

Following the same steps introduced earlier, we can derive the adjoint equation. Since the moving boundary is considered conventionally, the adjoint operator (i.e. left-hand-side of the adjoint equation) is similar to the one derived by Bewley et al. for channel flow [14]:

$$\mathcal{N}^*(\mathbf{q}^*) = \begin{bmatrix} \dfrac{\partial u_j^*}{\partial x_j} \\ \dfrac{\partial u_i^*}{\partial t} + u_j \left( \dfrac{\partial u_i^*}{\partial x_j} + \dfrac{\partial u_j^*}{\partial x_i} \right) + \nu \dfrac{\partial^2 u_i^*}{\partial x_j^2} + \dfrac{\partial p^*}{\partial x_i} \end{bmatrix}, \tag{15}$$

where the variables with $(*)$ are the adjoint variables. We notice that the velocity $u_j$ from the physical space remains in the adjoint operator. For this reason, we have to compute the flow equation first in order to even construct the adjoint equation. We also have the boundary term

$$\begin{aligned} b = & \int_{\Omega(t,\phi)} (u_j^* u_j')|_{t=0}^{t=T} d\mathbf{x} + \mathbf{B}_\infty \\ & - \int_0^T \int_{\mathcal{S}} u_i^* \left[ -p' \delta_{ij} + \nu \left( \frac{\partial u'_i}{\partial x_j} + \frac{\partial u'_j}{\partial x_i} \right) \right] n_j ds dt \\ & + \int_0^T \int_{\mathcal{S}} u'_i \left[ p^* n_i + u_i^* u_j n_j + u_j^* u_j n_i + \nu \left( \frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j - u_i^* V_j n_j \right] ds dt \end{aligned} \tag{16}$$

where $\Omega(t,\phi)$ is the computational domain of 'fluid' (for both physical space and adjoint space), $\mathbf{B}_\infty$ is the boundary term at the far-field, $\delta_{ij}$ is Kronecker's delta and $n_i$ is the norm direction of solid surface. It should be marked that the last term in (16) is due to the fact that the fluid domain $\Omega(t,\phi)$ is time-dependent.

Then, boundary and initial conditions are chosen to define the adjoint problem. An appropriate choice can largely simplify (or sometimes remove) the boundary term $b$. For instance, we can choose $u_j^*(t=T)=0$ to remove the time term together with the default condition $u'_j(t=0)=0$, choose $u^*(t)=0$ and $p^*(t)=0$ at the far-field boundary to remove $\mathbf{B}_\infty$ term and choose $u^*(t)=0$ at $\mathcal{S}$ to remove most of the other terms. On the other hand, to keep a consistent computational domain $\Omega$, the location of the boundary should be updated every time step so that, the solid boundary will keep on top of each other all the time for adjoint and physical spaces. The adjoint equation is normally solved backward from time $T$ to 0. For the specific objective in this example, it is convenient to pick the right-hand-side of the adjoint equation as

$$\mathbf{F}^* = \begin{bmatrix} 0 \\ 2(u_i - u_{0i})\chi_o(\Omega_o) \end{bmatrix}, \tag{17}$$

where $\chi_o$ is a characteristic function define as

$$\chi_o = \begin{cases} 1 & in \ \ \Omega_o \\ 0 & otherwise \end{cases}, \tag{18}$$

With all the definitions above, the perturbation of cost function can be written as

$$\mathcal{J}' = \int_0^T \int_{\mathcal{S}} u'_i \left[ p^* n_i + \nu \left( \frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right] ds dt. \tag{19}$$

At this moment, (19) is good enough to be used in traditional problems, where either the optimization does not involve boundary shape [23] or the shape optimization is entirely passive and not a function of time [45]. For problems like flapping wings, the boundary and object keeps moving, the optimization needs to arbitrarily change the moving trajectory and deformation in time. For such cases, it is difficult even to define the perturbation at the boundary in traditional approaches. It is noticed that $u'$ is the derivative of $u$ respect to control perturbation in Eulerian frame, while the change of solid boundary shape and speed is defined in Lagrangian frame. The same inconsistence exist in the description of no-slip fluid boundary condition. A straightforward remedy could be using unsteady mapping function and switching the problem back to the traditional situation without moving boundary. However, such mapping function would be too complex to be used in the derivation of adjoint equation [48]. Instead, we use non-cylindrical calculus (a.k.a. tube theory) to solve this problem [48, 49].

First, we define a mapping function $\mathcal{T}(t,\phi,\mathbf{x})$ between the original domain $\Omega(0,\phi)$ and the domain at a later time $\Omega(t,\phi)$, where $\mathcal{T}$ maps boundary to boundary between domains without topological change. A mapping velocity field $\mathbb{V}(t,\phi,\mathbf{x})$ can be defined by

$$\mathbb{V}(t,\phi,\mathbf{x}) = \frac{\partial \mathcal{T}(t,\phi,\mathbf{x})}{\partial t}. \tag{20}$$

It is reasonable to choose the map to follow the solid boundary in motion, which means $\mathbb{V}(t,\phi,\mathbf{x}) = V(t,\phi,s)$ being satisfied at boundary $\mathcal{S}$.

To describe the change of domain from the perturbation of control $\phi$, we define a transverse map $\tilde{\mathcal{T}}(t,\phi,\varepsilon,\mathbf{x}) : \Omega(t,\phi) \to \Omega(t,\phi+\varepsilon\phi')$. Accordingly a transverse velocity field can be defined:

$$Z(t,\phi,\mathbf{x}) = \left.\frac{\partial\tilde{\mathcal{T}}(t,\phi,\varepsilon,\mathbf{x})}{\partial\varepsilon}\right|_{\varepsilon=0}. \tag{21}$$

In the same framework, we define two derivatives: non-cylindrical material derivative and non-cylindrical shape derivative. Non-cylindrical material derivative is a derivative respect to control perturbation in Lagrangian point of view:

$$\dot{f}(t,\phi',\mathbf{x}) \equiv \lim_{\varepsilon\to 0}\frac{f(t,\phi+\varepsilon\phi',\tilde{\mathcal{T}}(t,\phi,\varepsilon,\mathbf{x})) - f(t,\phi,\mathbf{x})}{\varepsilon}, \tag{22}$$

and non-cylindrical shape derivative is a derivative respect to control perturbation in Eulerian point of view:

$$f'(\phi') = \dot{f}(\phi') - Z_i\frac{\partial f}{\partial x_i}, \tag{23}$$

which is similar to the classical definition with transverse velocity $Z$ in place of convective velocity. Using non-cylindrical calculus, we finally are able to relate $u'$ to its well-defined Lagrangian counter-part

$$\dot{u}_i = \dot{V}_i(t,\phi',s) = \frac{\partial V_i}{\partial\phi_l}\phi'_l$$

by

$$u'_i = \dot{u}_i - Z_j\frac{\partial u_i}{\partial x_j}. \tag{24}$$

For a rigid body with translation motion only, at the solid boundary $\mathcal{S}$, the equation reduces to

$$u'_i = \dot{u}_i - Z_j n_j\frac{\partial u_i}{\partial x_k}n_k. \tag{25}$$

Introducing the adjoint variable $Z^*$ accordingly, we have:

$$\int_0^T\int_\Omega \dot{\mathbb{V}}_i Z_i^* d\mathbf{x}dt = \int_0^T\int_\Omega \frac{dZ_i}{dt}Z_i^* d\mathbf{x}dt = \int_\Omega Z_i^* Z_i|_{t=0}^{t=T}d\mathbf{x} - \int_0^T\int_\Omega Z_i^* Z_i\frac{\partial\mathbb{V}_j}{\partial x_j}d\mathbf{x}dt - \int_0^T\int_\Omega\frac{dZ_i^*}{dt}Z_i d\mathbf{x}dt \tag{26}$$

We can set $Z_i^*(t=T)=0$ to remove the time term in the right hand side of above equation and assume the adjoint transverse velocity field following the form: $Z_i^* = \gamma_{\mathcal{S}}^*(\lambda n_i)$. Here $\gamma_{\mathcal{S}}^*$ is the adjoint of trance operator $\gamma_{\mathcal{S}}$ [49], which assigns to every function on $\Omega$ the its boundary values at $\mathcal{S}$. As a result, (26) becomes:

$$\int_0^T\int_{\mathcal{S}}\dot{V}_i\lambda n_i dsdt = -\int_0^T\int_{\mathcal{S}}\left(\frac{d\lambda}{dt} + \lambda\frac{\partial\mathbb{V}_j}{\partial x_j}\right)Z_i n_i dsdt \tag{27}$$

Let $\lambda$ take the following form

$$\frac{d\lambda}{dt} + \lambda\frac{\partial\mathbb{V}_j}{\partial x_j} = -\frac{\partial u_i}{\partial x_k}n_k\left[p^* n_i + \nu\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)n_j\right] \tag{28}$$

where $\frac{\partial\mathbb{V}_j}{\partial x_j}$ can be set to 0 at the solid boundary. Using (19), (25), (27) and (28), we obtain:

$$\mathcal{J}' = \int_0^T\int_{\mathcal{S}}\dot{V}_i\left[p^* n_i + \nu\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)n_j - \lambda n_i\right]dsdt, \tag{29}$$

which provides proper gradient for moving boundary problems.

### III.C. Numerical scheme

Immersed boundary method is implemented to solve both forward (from $t=0$ to $t=T$) flow equation and backward (from $t=T$ to $t=0$) adjoint equation in a similar way. A modified third-order Runge-Kutta/Crank-Nicolson scheme is used to match backward. With similar numerical scheme, the computational cost keeps about the same for solving the adjoint field.

# IV. Optimization results

The method is applied on two oscillatory cylinder problems: 1) oscillation with 3 frequencies; 2) arbitrary vertical motion without specific frequency constraint. For both cases, the computational domain is a rectangle area, $0 < x < 10$ and $0 < y < 6$, nondimensionalized by cylinder diameter $D$.

## IV.A. Optimize the amplitude of an oscillatory cylinder

The initial position of the cylinder center is at $(3, 3)$. A uniform $200 \times 120$ Cartesian mesh is used for computation with 125 Lagrangian points for the cylinder. Reynolds number is 100.

The cylinder oscillates vertically with displacement

$$S_1 = 0, \quad S_2 = \sum_{l=1}^{3} A_l \sin(\omega_l t) \tag{30}$$

and the corresponding speed

$$V_1 = 0, \quad V_2 = \sum_{l=1}^{3} \omega_l A_l \cos(\omega_l t) \tag{31}$$

where the subscription 1 and 2 is for $x$ and $y$ direction respectively, $\omega_l$ is angular frequency, and $A_l$ is the amplitude for each frequency. Three frequencies are considered here: $\omega_l = 0.4l\pi$. The control $\phi$ is defined by each amplitude: $\phi = (A_1, A_2, A_3) \in \mathbb{R}^3$. With any initial control (i.e. amplitudes $A_l$), we expect the current approach to optimize the amplitude of cylinder oscillation to match certain target velocity in a downstream region $\Omega_o$ of $4.9 < x < 5.1$ and $2.5 < y < 3.5$. The target velocity $\mathbf{u}_0$ is given by a pre-calculated simulation data with reference control setting $\phi^{(0)}$ for amplitude and other parameters being the same. Such setup essentially provides $\phi^{(0)}$ as an analytical solution to test our optimization algorithm. The cost function is the same as (11). The perturbation of cost function can be derived:

$$\mathcal{J}' = g_l \phi_l' = \phi_l' \int_0^T \int_{\mathcal{S}} \left( \frac{\partial V_i}{\partial \phi_l} - \frac{\partial S_k}{\partial \phi_l} \frac{\partial u_i}{\partial x_k} \right) \left[ p^* n_i + \nu \left( \frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right] ds dt, \tag{32}$$

which gives the gradient

$$g_l = \int_0^T \int_{\mathcal{S}} \left( \frac{\partial V_i}{\partial \phi_l} - \frac{\partial S_k}{\partial \phi_l} \frac{\partial u_i}{\partial x_k} \right) \left[ p^* n_i + \nu \left( \frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right] ds dt. \tag{33}$$

In this case, we choose the reference $\phi^{(0)} = (0.2, 0.2, 0)$, and the initial control $\phi^{(1)} = (0.25, 0.25, 0.25)$. Shown in figure (2), the cost function reduces quickly. The optimized control after 20 iterations is $\phi^{(20)} = (0.2008, 0.1990, -0.002)$, which has only 0.8% difference from the analytical solution $\phi^{(0)}$.

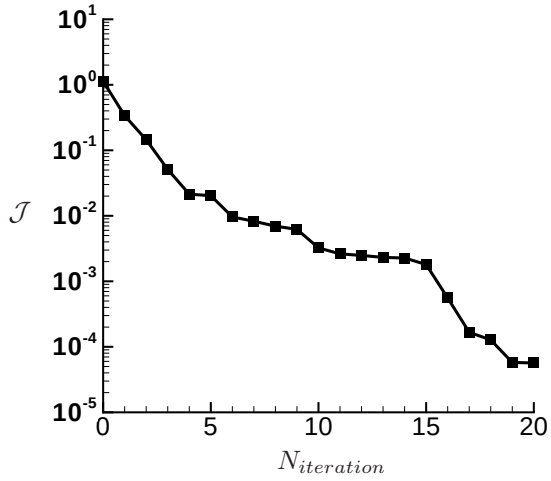


**Figure 2. Reduction of the cost functional $\mathcal{J}$ with the number of iterations $N_{iteration}$.**

Figure 3(a) shows a typical snapshot of flow field with initial control $\phi^{(1)}$. Oscillation at multi-frequency results in complex vortex shedding. Figure 3(b) shows a typical snapshot of the adjoint field in the first iteration, where the contours show nominal vorticity defined by the curl of adjoint variables $u_i^*$. It is shown that the adjoint field initiated from region $\Omega_o$ by the velocity difference. Then, adjoint information travels upstream as the computation is taken backward in time. When the adjoint information reaches the surface $\mathcal{S}$, the integration of adjoint variables provides the gradient to update control.

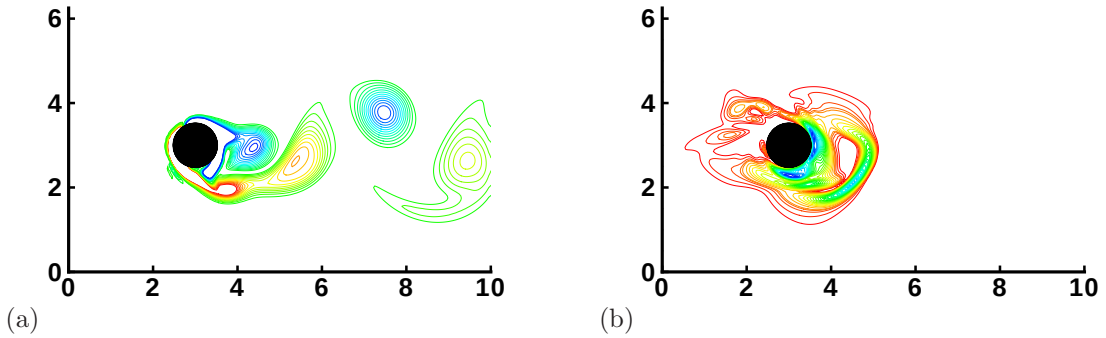


**Figure 3. Typical snapshots from (a) initial flow field (with contours of vorticity) and (b) adjoint field (with contours of nominal vorticity defined by adjoint variables $u_i^*$)**

### IV.B. Optimize arbitrary vertical motion of an cylinder

The real advantage of using adjoint-based method lies in its efficiency for problems with large design space. To test this capability, we remove the constraint of the cylinder moving with several known frequencies. Instead, the cylinder is allowed to move arbitrarily along vertical direction. The corresponding velocity and controls became

$$\begin{aligned} &V_1(t) = 0, \quad V_2(t) = \{V_2(t_1), V_2(t_1), ..., V_2(t_n)\} \\ &\phi = \{V_2(t_1), V_2(t_1), ..., V_2(t_n)\} \in \mathbb{R}^n, \end{aligned} \tag{34}$$

with 4000 degree of freedom when $n = 4000$ is chosen in our case.

For this case, the Reynolds number is 185, and the cost function is similar to (11) but with the integration from $t = T/2$ to $t = T$ only:

$$\mathcal{J} = \int_{T/2}^{T} \int_{\Omega_o} |\mathbf{u} - \mathbf{u}_0|^2 d\mathbf{x} dt, \tag{35}$$

with $T = 40$. The source term $\mathbf{F}^*$ in the adjoint equation (17) is also turned off during $0 < t < T/2$. The perturbation of cost function is:

$$\mathcal{J}' = \int_0^T g(t)\phi'(t)dt = \int_0^T \phi'(t) \int_{\mathcal{S}} \left[ p^* n_2 + \nu \left( \frac{\partial u_2^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_2} \right) n_j - \lambda n_2 \right] ds dt, \tag{36}$$

and the new gradient

$$g(t) = \int_{\mathcal{S}} \left[ p^* n_2 + \nu \left( \frac{\partial u_2^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_2} \right) n_j - \lambda n_2 \right] ds. \tag{37}$$

It is noticed that we extend the time back from $T/2$ to 0 to include effects with time delay.

The reference control $\phi^{(0)}$ is still a regular cosine function

$$\phi^{(0)}(t) = \omega A \cos(\omega t) \tag{38}$$

with $A = 0.2$ and $\omega = 0.4\pi$. The initial control is an elliptic function:

$$\phi^{(1)}(t) = \begin{cases} -\omega A \sqrt{\left(\frac{1}{4}\right)^2 - \left(\frac{t}{T_0} - N - \frac{1}{2}\right)^2} & at \ \ (N + \frac{1}{4})T_0 < t < (N + \frac{3}{4})T_0 \\ +\omega A \sqrt{\left(\frac{1}{4}\right)^2 - \left(\frac{t}{T_0} - N\right)^2} & at \ \ (N - \frac{1}{4})T_0 < t < (N + \frac{1}{4})T_0 \end{cases}, \tag{39}$$

where $N$ is an integer number and $T_0 = 2\pi/\omega$ is the period. However, the control throughout the optimization is an arbitrary function as in (34).

Figure 4(a) shows that, with 17 iterations, the cost function reduces to 0.0094 and it provides an optimal control $\phi^{(17)}$. Shown in figure 4(b), the optimal control matches well with the reference for $20 < t < 35$. For the time before $t = 20$, some improvement is shown from the time delay. At the same time, with adjoint source term being off, it is not expected to match perfectly with the target function in this period. For $t > 35$, it is clearly shown that the time is not enough in this range for the control to take effect because of the distance between the observer location (i.e. $\Omega_o$) and the control location (i.e. $\mathcal{S}$). Typical snapshots for this case are shown in figure 5.

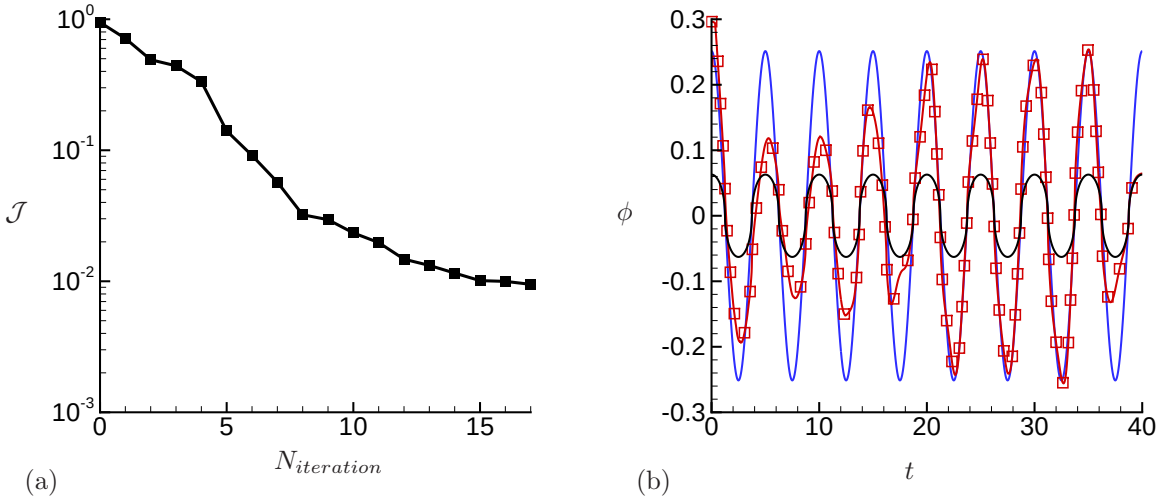


**Figure 4. (a) Reduction of the cost functional $\mathcal{J}$ with the number of iterations $N_{iteration}$. (b) The initial control $\phi^{(1)}$ (—), the optimized control $\phi^{(17)}$ (—) and the reference control $\phi^{(o)}$ (—).**

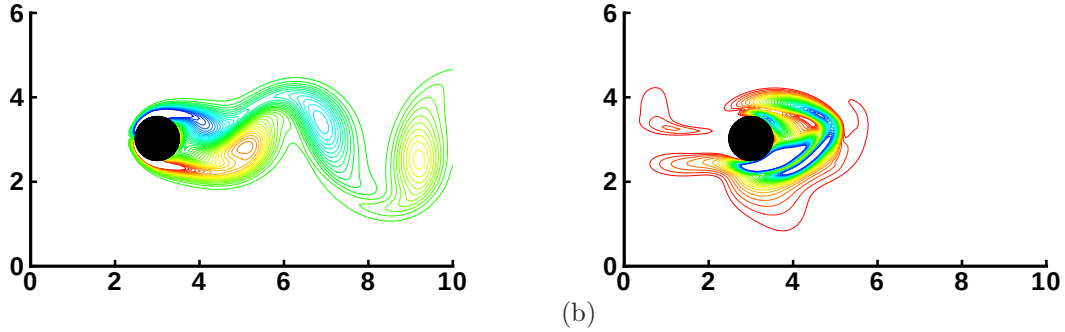


**Figure 5. Typical snapshots from (a) initial flow field (with contours of vorticity) and (b) adjoint field (with contours of nominal vorticity defined by adjoint variables $u_i^*$)**

## V. Conclusion

In this paper, we developed an adjoint-based approach using "non-cylindrical calculus" to study unsteady problems with constantly moving or morphing boundary/shape. By introducing a transverse velocity $Z$ at the boundary, the current approach relates the boundary perturbation in Eulerian framework and Lagrangian framework in a mathematically rigorous manner, while still keeps a relatively simple formulation. The corresponding adjoint problem is well defined and remains in its clean formulation which can be solved by similar algorithm used in forward flow computation. The approach is successfully validated against an analytical solution for the problem of cylinder oscillation at 3 frequencies. Then, to show the capability of handling large design space, the method is tested on a similar oscillatory cylinder but with the vertical motion being arbitrary instead of periodic motion with pre-defined frequencies. The release of constraint increases the degree of freedom to 4000 in our test. Our method shows the same efficiency in the convergency to the target function.

# VI. Acknowledgment

The authors gratefully acknowledge the support from Air Force Office of Scientific Research (AFOSR).

# References

[1]Dong, H., Mittal, R., and Najjar, F., "Wake topology and hydrodynamic performance of low aspect-ratio flapping foils," *J. Fluid Mech.*, Vol. 566, 2006, pp. 309–343.

[2]Dong, H. and Liang, Z., "The wing kinematics effects on performance and wake structure produced by finite-span hovering wings," AIAA Paper 2008-3819, 2008.

[3]Yang, T., Wei, M., and Zhao, H., "Numerical study of flexible flapping wing propulsion," *AIAA J.*, Vol. 48, No. 12, 2010.

[4]Yin, B. and Luo, H., "Effect of wing inertia on hovering performance of flexible flapping wings," *Phys. Fluids*, Vol. 22, No. 111902, 2010.

[5]Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum aerodynamic design using the Navier-Stokes equations," AIAA paper 97-0101, 1997.

[6]Jameson, A., "Re-engineering the design process through computation," AIAA Paper 97-0641, 1997.

[7]Giles, M. B. and Pierce, N. A., "Improved lift and drag estimates using adjoint Euler equations," AIAA paper 99-3293, 1999.

[8]Sung, C. and Kwon, J. H., "Accurate aerodynamic sensitivity analysis using adjoint equations," *AIAA J.*, Vol. 38, No. 2, 2000, pp. 243–250.

[9]Nemec, M. and Zingg, D. W., "Towards efficient aerodynamic shape optimization based on the Navier-Stokes equations," AIAA paper 2001-2532, 2001.

[10]Nemec, M. and Aftosmis, M. J., "Adjoint sensitivity computations for an embedded-boundary Cartesian mesh method," *J. Comput. Phys.*, Vol. 227, 2008, pp. 2724–2742.

[11]Deng, Y., Liu, Z., Zhang, P., Liu, Y., and Wu, Y., "Topology optimization of unsteady incompressible Navier-Stokes flows," *J. Comput. Phys.*, Vol. 230, 2011, pp. 6688–6708.

[12]Mohammadi, B., Molho, J. I., and Santiago, J. A., "Design of minimal dispersion fluidic channels in a CAD-free framework," *Center for Turbulence Research, Proceedings of the Summer Program 2000*, 49–62, 2000.

[13]Larson, M. G. and Barth, T. J., "A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems," $1^{st}$ International Symposium on Discontinuous Galerkin Methods, Newport, R.I., May 1999, 1999.

[14]Bewley, T. R., Moin, P., and Temam, R., "DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms," *J. Fluid Mech.*, Vol. 447, 2001, pp. 179–225.

[15]Tam, C. K. W. and Auriault, L., "Mean flow refraction effects on sound radiated from localized sources in a jet," *J. Fluid Mech.*, Vol. 370, 1998, pp. 149–174.

[16]Dobrinsky, A. and Collis, S. S., "Adjoint methods for receptivity prediction in nonparallel flows," submitted to *Phys. Fluids*, 2000.

[17]Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., and Ulbrich, S., "Optimal control of unsteady compressible viscous flows," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 1401–29.

[18]Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., and Ulbrich, S., "Towards adjoint-based methods for aeroacoustic control," AIAA Paper 2001-0821, 2001.

[19]Collis, S. S., Ghayour, K., Heinkenschloss, M., Ulbrich, M., and Ulbrich, S., "Numerical solution of optimal control problems governed by the compressible navier-Stokes equations," Proceedings of the International Conference on Optimal Control of Complex Structures, G. Leugering, J. Sprekels, and F. Troltzsch (Eds.), Birkhauser Verlag, 2000.

[20]Rowley, C. W., "Model reduction for fluids using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, Vol. 15, No. 3, 2005, pp. 997–1013.

[21]Ahuja, S., Rowley, C. W., Kevrekidis, I. G., Wei, M., Colonius, T., and Tadmor, G., "Low-dimensional models for control of leading-edge vortices: equilibria and linearized models," AIAA Paper 2007-709, 2007.

[22]Wei, M. and Freund, J. B., "Noise control using adjoint-based optimization," 8th AIAA/CEAS Aeroacoustics Conference, Breckenridge, CO, AIAA Paper 2002-2524, June 2002.

[23]Wei, M. and Freund, J. B., "A noise-controlled free shear flow," *J. Fluid Mech.*, Vol. 546, 2006, pp. 123–152.

[24]Hu, H. H., Joseph, D. D., and Crochet, M. J., "Direct simulation of fluid particle mothions," *Theoretical and Computational Fluid Dynamics*, Vol. 3, No. 5, 1992, pp. 285–306.

[25]Shyy, W., Udaykumar, H. S., Rao, M. M., and Smith, R. W., *Computational Fluid Dynamics with Moving Boundaries*, Taylor & Francis, 1996.

[26]Hu, H. H., Patankar, N. A., and Zhu, M. Y., "Direct numerical simulations of fluid-solid systems using the arbitrary Larangian-Eulerian technique," *J. Comput. Phys.*, Vol. 169, No. 2, 2001, pp. 427–462.

[27]Shyy, W. and Liu, H., "Flapping wings and aerodynamic lift: the role of leading-edge vortices," *AIAA J.*, Vol. 45, 2007, pp. 2817–2819.

[28]Glowinski, R., Pan, T. W., Hesla, T. I., and Joseph, D. D., "A distributed Lagrange multiplier/fictitious domain method for particulate flows," *International Journal of Multiphase Flow*, Vol. 25, 1999, pp. 755–794.

[29]Chang, Y. C., Hou, T. Y., Merriman, B., and Osher, S., "A level set formulation of Eulerian interface capturing methods for incompressible fluid flows," *J. Comput. Phys.*, Vol. 124, No. 2, 1996, pp. 449–464.

[30]Fauci, L. J. and Peskin, C. S., "A computational model of aquatic animal locomotion," *J. Comput. Phys.*, Vol. 77, No. 1, 1988, pp. 85–108.

[31]Peskin, C. S., "Flow patterns around heart valves: a numerical method," *J. Comput. Phys.*, Vol. 10, No. 2, 1972, pp. 252–271.

[32]Peskin, C. S., "Numerical analysis of blood flow in the heart," *J. Comput. Phys.*, Vol. 25, No. 3, 1977, pp. 220–252.

[33]Mittal, R. and Iaccarino, G., "Immersed boundary methods," *Ann. Rev. Fluid Mech.*, Vol. 37, 2005, pp. 239–261.

[34]Yang, J. and Balaras, E., "An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries," *J. Comput. Phys.*, Vol. 215, 2006, pp. 12–40.

[35]Ghias, R., Mittal, R., and Dong, H., "A sharp interface immersed boundary method for compressible viscous flows," *J. Comput. Phys.*, Vol. 225, 2007, pp. 528–553.

[36]Zhang, N. and Zheng, Z. C., "An improved direct-forcing immersed-boundary method for finite difference applications," *J. Comput. Phys.*, Vol. 221, 2007, pp. 250–268.

[37]Taira, K. and Colonius, T., "The immersed boundary method: a projection approach," *J. Comput. Phys.*, Vol. 225, 2007, pp. 2118–2137.

[38]Colonius, T. and Taira, K., "A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, 2008, pp. 2131–2146.

[39]LeVeque, R. J. and Li, Z., "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources," *SIAM Journal on Numerical Analysis*, Vol. 18, No. 3, 1997, pp. 709–735.

[40]LeVeque, R. J. and Li, Z., "Immersed interface method for stokes flow with elastic boundaries or surface tension," *SIAM Journal on Scientific Computing*, Vol. 18, No. 3, 1997, pp. 709–735.

[41]Xu, S. and Wang, Z. J., "Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation," *SIAM Journal on Scientific Computing*, Vol. 27, No. 6, 2006, pp. 1948–1980.

[42]Zhao, H., Freund, J. B., and Moser, R. D., "A fixed-mesh method for incompressible flow-structure systems with finite solid deformations," *J. Comput. Phys.*, Vol. 227, 2008, pp. 3114–3140.

[43]Boffi, D., Gastaldi, L., Heltai, L., and Peskin, C. S., "On the hyper-elastic formulation of the immersed boundary method," *Comput. Methods Appl. Mech. Engrg.*, Vol. 197, 2008, pp. 2210–2231.

[44]Mohd-Yusof, J., "Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries," Center for Turbulence Research, Annual Research Briefs – 1997, 1997.

[45]Nadarajah, S. and Jameson, A., "A comparison of the continous and discrete adjoint approaches to automatic aerodynamic optimization," AIAA paper 2000-0667, 2000.

[46]Wei, M., *Jet Noise Control by Adjoint-based Optimization*, Ph.D. thesis, Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2004.

[47]Michalewicz, Z., *Genetic algorithms + data structures = evolution programs*, Springer, 1996.

[48]Protas, B. and Liao, W., "Adjoint-based optimization of PDEs in moving domains," *JCP*, Vol. 227, No. 04, 2008, pp. 2707–2723.

[49]Moubachir, M. and Zolesio, J. P., *Moving shape analysis and control: applications to fluid structure interactions*, Monographs and textbooks in pure and applied mathematics, Chapman & Hall/CRC, 2006.