

## Chapter 1

# Energetically optimal flapping wing motions via adjoint-based optimization and high-order discretizations

Matthew J. Zahr and Per-Olof Persson

### Abstract

A globally high-order numerical discretization of time-dependent conservation laws on deforming domains, and the corresponding fully discrete adjoint method, is reviewed and applied to determine energetically optimal flapping wing motions subject to aerodynamic constraints using a reduced space PDE-constrained optimization framework. The conservation law on a deforming domain is transformed to one on a fixed domain and discretized in space using a high-order discontinuous Galerkin method. An efficient, high-order temporal discretization is achieved using diagonally implicit Runge-Kutta schemes. Quantities of interest, such as the total energy required to complete a flapping cycle and the integrated forces produced on the wing, are discretized in a solver-consistent way, that is, via the same spatio-temporal discretization used for the conservation law. The fully discrete adjoint method is used to compute discretely consistent gradients of the quantities of interest and passed to a black-box, gradient-based nonlinear optimization solver. This framework successfully determines an energetically optimal flapping trajectory such that the net thrust of the wing is zero to within 9 digits after only 12 optimization iterations.

## 1.1 Introduction

Flapping flight has been a subject of intense interest and research over the past several decades due to its relevance in designing Micro Aerial Vehicles (MAVs) – unmanned aerial vehicles measuring no more than 15 cm in any dimension, envisioned in a number of civilian and military applications, including surveillance and reconnaissance [32, 43] – and in the understanding of biological systems. The basic goal of any system, whether biological or manmade, that relies on flapping propulsion is to adjust the kinematics of the flapping wing, and possibly its shape, to minimize the energy required to complete a given mission. The problem of determining the flapping kinematics that lead to an energetically optimal motion, while satisfying various mission constraints, leads to a nonlinearly constrained PDE-constrained optimization problem

---

Matthew J. Zahr

Mathematics Group, Lawrence Berkeley National Laboratory and Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, USA. e-mail: mjzahr@lbl.gov

Per-Olof Persson

Mathematics Group, Lawrence Berkeley National Laboratory and Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, USA. e-mail: persson@berkeley.edu

$$\begin{aligned}
& \underset{\mathbf{U}, \boldsymbol{\mu}}{\text{minimize}} && J(\mathbf{U}, \boldsymbol{\mu}) := \frac{1}{T} \int_0^T \int_{\Gamma} j(\mathbf{U}(\mathbf{x}, t), \boldsymbol{\mu}, t) dS dt \\
& \text{subject to} && \mathbf{C}(\mathbf{U}, \boldsymbol{\mu}) := \frac{1}{T} \int_0^T \int_{\Gamma} \mathbf{c}(\mathbf{U}(\mathbf{x}, t), \boldsymbol{\mu}, t) dS dt \leq 0 \\
& && \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) = 0 \quad \text{in } v(\boldsymbol{\mu}, t),
\end{aligned} \tag{1.1}$$

where  $\mathbf{U}(\mathbf{x}, t) \in \mathbb{R}^{N_U}$  is the spatio-temporal solution of the conservation law, i.e., the last constraint in the optimization problem, in the domain  $\mathbf{x} \in v(\boldsymbol{\mu}, t)$ ,  $t \in (0, T]$ ,  $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$  is a vector of parameters controlling the kinematics of the wing,  $T$  is the period of the flapping motion,  $j(\mathbf{U}, \boldsymbol{\mu}, t)$  is the pointwise (in space and time) contribution to energy added to the flow and  $J(\mathbf{U}, \boldsymbol{\mu})$  is the corresponding quantity integrated over space and time, i.e., the time-averaged work done by the surface  $\Gamma(\boldsymbol{\mu}, t)$  on the flow, and  $\mathbf{c}(\mathbf{U}, \boldsymbol{\mu}, t)$  and  $\mathbf{C}(\mathbf{U}, \boldsymbol{\mu})$  are pointwise and integrated, respectively, mission-specific constraints. In the context of MAV design, the constraints will likely correspond to bounds on the thrust, lift, and stability of the vehicle [49, 19].

Due to the unsteady governing equations, most attempts to solve the PDE-constrained optimization problem in (1.1) in the context of flapping flight utilize a *reduced space* approach, also known as nested analysis and design, whereby the state variable  $\mathbf{U}$  is treated as an implicit function of the parameters  $\boldsymbol{\mu}$ , i.e.,  $\mathbf{U}(\boldsymbol{\mu})$  is obtained by solving the (discretized) conservation law. This removes the state variable from the set of optimization variables and eliminates the PDE constraint to reduce the optimization problem in (1.1) to

$$\begin{aligned}
& \underset{\boldsymbol{\mu}}{\text{minimize}} && \mathcal{J}(\boldsymbol{\mu}) := J(\mathbf{U}(\boldsymbol{\mu}), \boldsymbol{\mu}) \\
& \text{subject to} && \mathbf{C}(\boldsymbol{\mu}) := \mathbf{C}(\mathbf{U}(\boldsymbol{\mu}), \boldsymbol{\mu}) \leq 0.
\end{aligned} \tag{1.2}$$

Due to the relatively large expense of high-fidelity methods that model the flow using the Navier-Stokes equations, a number of low- and multi-fidelity methods have been proposed to approximately solve the optimization problem in (1.2) or gain insight into the physics of flapping. Among these low-fidelity methods include: potential flow methods that assume the flow is irrotational, inviscid, and incompressible such as wake only and panel methods [56], lifting line methods, and unsteady vortex-lattice methods [46, 20] that assume the flow is inviscid and incompressible and use global vorticity circulation balance and the Biot-Savart law to construct a 3D velocity field.

While low- and multi-fidelity methods have been popular in the study of flapping flight [24, 7, 46, 9, 20, 19, 56], the need for high-fidelity computational tools has been recognized [43] due to the complex flow features that occur, and are critical for performance, in low Reynolds number flapping. In particular, these flows are highly vortical and subject to separation [22, 44, 2, 6] that will violate many of the critical assumptions of low-fidelity methods [56]. The generation and shedding of a leading-edge vortex, possibly through rapid changes in angle of attack (dynamic stall), have been shown to be important to efficient lift generation [7, 44, 42, 6] and a computational method should possess minimal dissipation to ensure these critical structures are preserved. Furthermore, a realistic study of flapping at scales relevant to the design of MAVs should be performed in three-dimensions due to the importance of three-dimensional effects such as stabilization of the leading-edge vortex [13, 50, 5, 27, 7, 6] and to include out-of-plane flapping kinematics that are relevant to thrust production and control [4].

In this work, we extend the globally high-order method and corresponding fully discrete adjoint method presented in [63] for the discretization and optimization of general nonlinear, unsteady conservation laws to address the challenges of three-dimensional flapping, such as the parametrization of three-dimensional flapping and robust deformation of the three-dimensional geometry. The conservation law on a parametrized, deforming domain is transformed to a fixed domain using an Arbitrary Lagrangian-Eulerian (ALE) formulation and the resulting equations are discretized in space and time using a discontinuous Galerkin method and diagonally implicit Runge-Kutta scheme, respectively. Relevant details are provided on using the ALE formulation to move a *curved* mesh, whereby the reference mesh is taken as straight-sided and the ALE mapping encapsulates the curving as well as the domain deformation. In contrast to most computational approaches that only integrate Quantities of Interest (QoIs), that will eventually define the objective and constraints of the flapping optimization problem, to second-order using the trapezoidal rule, this work discretizes the QoI to exactly the same order as the governing equation using the solver-consistent approach of [63]. High-order methods are an emphasis of this work since they are well-suited to model the highly vortical flow around a flapping

wing due to the small amount of numerical dissipation they introduce [35]. An alternative to high-order methods that has been proposed and demonstrated in the context of flapping to limit numerical dissipation associated with low-order methods is a kinetic energy preserving finite volume scheme [2, 11]. However, we also commit to high-order methods because they have been shown to require fewer spatial degrees of freedom [54, 63] and time steps [30, 61] compared to low-order counterparts.

Given the large computational cost of objective and constraint queries that require high-order Computational Fluid Dynamics (CFD) simulations, and the high-dimensional design space required to sufficiently parametrize three-dimensional flapping, which may include parameters for the flapping kinematics, fixed or actively morphed [20, 63] shape, flexibility of the wing [43, 22, 65, 48, 42, 47], gradient-based optimization methods are used to solve the optimization problem in (1.2) due to their fast convergence properties. Also, since the optimization problems considered in this work involve more parameters than constraints, the gradients of the optimization functionals are computed via the adjoint method since the cost scales very weakly with the number of parameters. Since a black-box optimizer is used to solve the constrained optimization problem in (1.2) with the underlying high-order discretization, the fully discrete variant of the adjoint method is used to ensure the computed gradients are consistent with the functionals to which they correspond.

The proposed numerical method for simulation and optimization of conservation laws on parametrized, deforming domains is used to determine energetically optimal flapping subject to a thrust constraint. The chosen optimization formulation is similar to that studied in [19], which differs from the unconstrained thrust or propulsive efficiency maximization problem that is usually chosen to study optimal flapping [40, 49, 55]. The optimization problem considered in this work is closer to the optimization problem instinctively solved in-flight by biological systems [43] and relevant in the design of MAVs.

The remainder of this document is organized as follows. Section 1.2 introduces the governing conservation law considered in this work, the isentropic Navier-Stokes equations, and an Arbitrary Lagrangian-Eulerian method that transforms it from a deforming, parametrized domain to a fixed one. Section 1.3 introduces the high-order discretization of the conservation law and its quantities of interest, with special attention paid to high-order representation of the geometry in the ALE framework and Section 1.4 introduces the fully discrete adjoint method that was derived in [63]. Finally, Section 1.5 applies this high-order simulation and optimization framework to energetically optimal, three-dimensional flapping flight under lift and thrust constraints and Section 1.6 offers conclusions.

## 1.2 Governing equations

This section presents a formulation of general conservation laws on a *parametrized, deforming domain* using an Arbitrary Lagrangian-Eulerian (ALE) formulation, which summarizes the work in [38]. Given that this work is concerned with energetically optimal flapping flight, the compressible Navier-Stokes equations are taken as the governing equations; however, the primal and adjoint numerical scheme is presented for the case of a general, nonlinear, vector-valued conservation law.

### 1.2.1 Compressible Navier-Stokes equations

The compressible Navier-Stokes equations are written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \quad (1.3)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) = \frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3, \quad (1.4)$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(u_j(\rho E + p)) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(u_i \tau_{ij}), \quad (1.5)$$

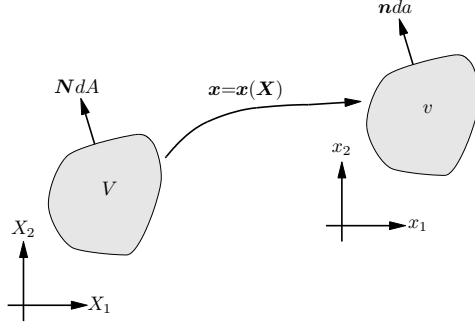


Fig. 1.1: Time-dependent mapping between reference and physical domains.

where  $\rho$  is the fluid density,  $u_1, u_2, u_3$  are the velocity components, and  $E$  is the total energy. The viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \quad \text{and} \quad q_j = -\frac{\mu}{\text{Pr}} \frac{\partial}{\partial x_j} \left( E + \frac{p}{\rho} - \frac{1}{2} u_k u_k \right). \quad (1.6)$$

Here,  $\mu$  is the viscosity coefficient and  $\text{Pr} = 0.72$  is the Prandtl number which we assume to be constant. For an ideal gas, the pressure  $p$  has the form

$$p = (\gamma - 1) \rho \left( E - \frac{1}{2} u_k u_k \right), \quad (1.7)$$

where  $\gamma$  is the adiabatic gas constant. In this work, the entropy is assumed constant, that is to say the flow is adiabatic and reversible. This makes the energy equation redundant and effectively reduces the square system of PDEs of size  $n_{sd} + 2$  to one of size  $n_{sd} + 1$ , where  $n_{sd}$  is the number of spatial dimensions. It can be shown, under suitable assumptions, that the solution of the isentropic approximation of the Navier-Stokes equations converges to the solution of the incompressible Navier-Stokes equations as the Mach number goes to 0 [29, 12, 17].

### 1.2.2 Arbitrary Lagrangian-Eulerian formulation of conservation laws

Consider a general system of conservation laws, defined on a parametrized, deforming domain,  $v(\boldsymbol{\mu}, t)$ ,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) = 0 \quad \text{in } v(\boldsymbol{\mu}, t) \quad (1.8)$$

where the physical flux is decomposed into an inviscid and a viscous part  $\mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) = \mathbf{F}^{inv}(\mathbf{U}) + \mathbf{F}^{vis}(\mathbf{U}, \nabla \mathbf{U})$ ,  $\mathbf{U}(\mathbf{x}, \boldsymbol{\mu}, t)$  is the solution of the system of conservation laws,  $t \in (0, T]$  represents time, and  $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$  is a vector of parameters. This work will focus on the case where the domain is parametrized by  $\boldsymbol{\mu}$ .

The conservation law on the physical, deforming domain  $v(\boldsymbol{\mu}, t) \subset \mathbb{R}^{n_{sd}}$  is transformed into one on a fixed reference domain  $V \subset \mathbb{R}^{n_{sd}}$  through the introduction of a time-dependent diffeomorphism between the physical and reference domains:  $\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t) = \mathcal{G}(\mathbf{X}, \boldsymbol{\mu}, t)$ . In this setting,  $n_{sd}$  is the number of spatial dimensions,  $\mathbf{X} \in V$  is a point in the reference domain and  $\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t) \in v(\boldsymbol{\mu}, t)$  is the corresponding point in the physical domain at time  $t$  and parameter configuration  $\boldsymbol{\mu}$ . The transformed system of conservation laws takes the form

$$\frac{\partial \mathbf{U}_X}{\partial t} \Big|_{\mathbf{X}} + \nabla_X \cdot \mathbf{F}_X(\mathbf{U}_X, \nabla_X \mathbf{U}_X) = 0 \quad \text{in } V \quad (1.9)$$

where  $\nabla_{\mathbf{X}}$  denotes spatial derivatives with respect to the reference variables,  $\mathbf{X}$ . The transformed state vector,  $\mathbf{U}_{\mathbf{X}}$ , and its corresponding spatial gradient with respect to the reference configuration take the form

$$\mathbf{U}_{\mathbf{X}} = g\mathbf{U}, \quad \nabla_{\mathbf{X}}\mathbf{U}_{\mathbf{X}} = g^{-1}\mathbf{U}_{\mathbf{X}}\frac{\partial g}{\partial \mathbf{X}} + g\nabla\mathbf{U} \cdot \mathbf{G}, \quad (1.10)$$

where  $\mathbf{G} = \nabla_{\mathbf{X}}\mathcal{G}$ ,  $g = \det(\mathbf{G})$ ,  $\mathbf{v}_{\mathbf{G}} = \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \mathcal{G}}{\partial t}$ . The transformed fluxes are

$$\begin{aligned} \mathbf{F}_{\mathbf{X}}(\mathbf{U}_{\mathbf{X}}, \nabla_{\mathbf{X}}\mathbf{U}_{\mathbf{X}}) &= \mathbf{F}_{\mathbf{X}}^{inv}(\mathbf{U}_{\mathbf{X}}) + \mathbf{F}_{\mathbf{X}}^{vis}(\mathbf{U}_{\mathbf{X}}, \nabla_{\mathbf{X}}\mathbf{U}_{\mathbf{X}}), \\ \mathbf{F}_{\mathbf{X}}^{inv}(\mathbf{U}_{\mathbf{X}}) &= g\mathbf{F}^{inv}(g^{-1}\mathbf{U}_{\mathbf{X}})\mathbf{G}^{-T} - \mathbf{U}_{\mathbf{X}} \otimes \mathbf{G}^{-1}\mathbf{v}_{\mathbf{G}}, \\ \mathbf{F}_{\mathbf{X}}^{vis}(\mathbf{U}_{\mathbf{X}}, \nabla_{\mathbf{X}}\mathbf{U}_{\mathbf{X}}) &= g\mathbf{F}^{vis}\left(g^{-1}\mathbf{U}_{\mathbf{X}}, g^{-1}\left[\nabla_{\mathbf{X}}\mathbf{U}_{\mathbf{X}} - g^{-1}\mathbf{U}_{\mathbf{X}}\frac{\partial g}{\partial \mathbf{X}}\right]\mathbf{G}^{-1}\right)\mathbf{G}^{-T}. \end{aligned} \quad (1.11)$$

For details regarding the derivation of the transformed equations, the reader is referred to [38].

When integrated using inexact numerical schemes, this ALE formulation does not satisfy the Geometric Conservation Law (GCL) [15, 38]. This is overcome by introduction of an auxiliary variable  $\bar{g}$ , defined as the solution of

$$\frac{\partial \bar{g}}{\partial t} - \nabla_{\mathbf{X}} \cdot (g\mathbf{G}^{-1}\mathbf{v}_{\mathbf{G}}) = 0. \quad (1.12)$$

The auxiliary variable,  $\bar{g}$  is used to modify the *transformed* conservation law according to

$$\left. \frac{\partial \mathbf{U}_{\bar{\mathbf{X}}}}{\partial t} \right|_{\mathbf{X}} + \nabla_{\mathbf{X}} \cdot \mathbf{F}_{\bar{\mathbf{X}}}(\mathbf{U}_{\bar{\mathbf{X}}}, \nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}}) = 0 \quad (1.13)$$

where the GCL-transformed state variables are

$$\mathbf{U}_{\bar{\mathbf{X}}} = \bar{g}\mathbf{U}, \quad \nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}} = \bar{g}^{-1}\mathbf{U}_{\bar{\mathbf{X}}}\frac{\partial \bar{g}}{\partial \mathbf{X}} + \bar{g}\nabla\mathbf{U} \cdot \mathbf{G} \quad (1.14)$$

and the corresponding fluxes

$$\begin{aligned} \mathbf{F}_{\bar{\mathbf{X}}}(\mathbf{U}_{\bar{\mathbf{X}}}, \nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}}) &= \mathbf{F}_{\bar{\mathbf{X}}}^{inv}(\mathbf{U}_{\bar{\mathbf{X}}}) + \mathbf{F}_{\bar{\mathbf{X}}}^{vis}(\mathbf{U}_{\bar{\mathbf{X}}}, \nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}}), \\ \mathbf{F}_{\bar{\mathbf{X}}}^{inv}(\mathbf{U}_{\bar{\mathbf{X}}}) &= g\mathbf{F}^{inv}(\bar{g}^{-1}\mathbf{U}_{\bar{\mathbf{X}}})\mathbf{G}^{-T} - \mathbf{U}_{\bar{\mathbf{X}}} \otimes \mathbf{G}^{-1}\mathbf{v}_{\mathbf{G}}, \\ \mathbf{F}_{\bar{\mathbf{X}}}^{vis}(\mathbf{U}_{\bar{\mathbf{X}}}, \nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}}) &= g\mathbf{F}^{vis}\left(\bar{g}^{-1}\mathbf{U}_{\bar{\mathbf{X}}}, \bar{g}^{-1}\left[\nabla_{\mathbf{X}}\mathbf{U}_{\bar{\mathbf{X}}} - \bar{g}^{-1}\mathbf{U}_{\bar{\mathbf{X}}}\frac{\partial \bar{g}}{\partial \mathbf{X}}\right]\mathbf{G}^{-1}\right)\mathbf{G}^{-T}. \end{aligned} \quad (1.15)$$

It was shown in [38] that the transformed equations (1.13) satisfy the GCL.

### 1.2.3 Uniform flow initial condition

A number of initial conditions can be used to initialize an unsteady CFD simulation, including uniform flow [23, 21], the steady state solution [28, 26, 63], and the state that leads to periodic flow [64]. In this work, the unsteady simulation is initialized from uniform flow for the sake of simplicity. Non-physical transients that result from using an initial condition that is incompatible with the boundary conditions will be dissipated by simulating multiple periods of the flapping motion and only integrating the quantity of interest over the final period. The ALE-transformed state corresponding to uniform flow takes the form

$$\begin{aligned} \mathbf{U}_{\bar{\mathbf{X}}}(\mathbf{X}, \boldsymbol{\mu}, 0) &= g_0(\mathbf{X}, \boldsymbol{\mu})\bar{\mathbf{U}}(\mathbf{X}) \\ \bar{g}(\mathbf{X}, \boldsymbol{\mu}, 0) &= g_0(\mathbf{X}, \boldsymbol{\mu}) \end{aligned} \quad (1.16)$$

where  $\bar{U}(\mathbf{X})$  defines the desired uniform initial condition on the reference domain and  $g_0(\mathbf{X}, \boldsymbol{\mu}) := g(\mathbf{X}, \boldsymbol{\mu}, 0)$  is the determinant of the deformation gradient at time  $t = 0$ .

### 1.3 High-order numerical discretization

This section discusses a globally high-order numerical discretization of the governing equations presented in the previous section. It summarizes the work in [1, 3, 63].

#### 1.3.1 Spatial discretization: discontinuous Galerkin method

To proceed, the second-order system of partial differential equations in (1.12)-(1.13) is converted to first-order form

$$\begin{aligned} \frac{\partial \bar{g}}{\partial t} \Big|_{\mathbf{X}} + \nabla_{\mathbf{X}} \cdot (g \mathbf{G}^{-1} \mathbf{v}_{\mathbf{G}}) &= 0 \\ \frac{\partial \mathbf{U}_{\bar{\mathbf{X}}}}{\partial t} \Big|_{\mathbf{X}} + \nabla_{\mathbf{X}} \cdot \mathbf{F}_{\bar{\mathbf{X}}}(\mathbf{U}_{\bar{\mathbf{X}}}, \mathbf{Q}_{\bar{\mathbf{X}}}) &= 0 \\ \mathbf{Q}_{\bar{\mathbf{X}}} - \nabla_{\mathbf{X}} \mathbf{U}_{\bar{\mathbf{X}}} &= 0, \end{aligned} \tag{1.17}$$

where  $\mathbf{Q}_{\bar{\mathbf{X}}}$  is introduced as an auxiliary variable to represent the spatial gradient of the  $\mathbf{U}_{\bar{\mathbf{X}}}$ . Equation (1.17) is discretized using a standard nodal discontinuous Galerkin finite element method [10, 3], which, after local elimination of the auxiliary variables  $\mathbf{Q}_{\bar{\mathbf{X}}}$ , leads to the following system of ODEs

$$\mathbf{M} \frac{\partial \mathbf{u}}{\partial t} = \mathbf{r}(\mathbf{u}, \boldsymbol{\mu}, t), \tag{1.18}$$

where  $\mathbf{M}$  is the block-diagonal, symmetric, *fixed* mass matrix (state- and parameter-independent),  $\mathbf{u}$  is the vectorization of  $[\mathbf{U}_{\bar{\mathbf{X}}}^T \bar{g}]^T$  at all nodes in the mesh, and  $\mathbf{r}$  is the nonlinear function defining the DG discretization of the inviscid and viscous fluxes. See [63] for an efficient treatment of  $\bar{g}$  that does not lead to an *enlarged* system of ODEs.

To achieve high-order accuracy, the geometry must be represented to high-order, which calls for a curved mesh. Since a curved mesh is usually defined as a nonlinear mapping, e.g., based on nonlinear elasticity or some optimality criteria, applied to an underlying linear or straight-sided mesh, two options exist for defining the ALE mapping. First, the curved mesh can be taken as the reference domain and the ALE mapping must only account for the mapping between the curved mesh and the physical domain. In this case, the ALE mapping takes the form

$$\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t) = \boldsymbol{\varphi}(\mathbf{X}, \boldsymbol{\mu}, t) \tag{1.19}$$

where  $\mathbf{X}$  are coordinates in the domain defined by the *curved mesh* and  $\boldsymbol{\varphi}$  maps the curved mesh into the physical domain. Alternatively, the straight-sided mesh can be taken as the reference mesh and the ALE mapping constructed as a composition of maps that takes the straight-sided mesh into the physical domain with curved boundaries to represent the geometry to high-order. In this case, the ALE mapping takes the form

$$\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t) = \boldsymbol{\varphi}(\boldsymbol{\phi}(\mathbf{X}), \boldsymbol{\mu}, t) \tag{1.20}$$

where  $\mathbf{X}$  are coordinates in the domain defined by the *linear mesh*,  $\boldsymbol{\phi}$  maps the linear mesh to the curved mesh, and  $\boldsymbol{\varphi}$  maps the curved mesh into the physical domain. Even though these options are mathematically equivalent, the latter option is chosen in this work as it leads to a simpler implementation, particularly in the definition of derivative terms required for the adjoint method, but also because all integrals are calculated on straight-sided elements.

This section closes with a discussion of how the domain deformation terms that arise in the ALE formulation will be defined at the semi-discrete level. If the mapping from the reference to physical domain is known analytically, all domain deformation terms, i.e.,  $\mathbf{x}$ ,  $\dot{\mathbf{x}}$ ,  $\mathbf{G}$ ,  $g$ , can be computed exactly and used in (1.15). However, there are many cases where this is not the case, e.g., the domain deformation is the result of a numerical procedure [14, 58, 39, 16]. An alternative that closely aligns with finite element ideology is to interpolate the ALE mapping onto the finite element shape functions and compute spatial gradients by differentiating the shape functions. In this setting, the action of the mapping and its time derivative are computed on the nodal coordinates of the reference mesh, i.e.,

$$\begin{aligned}\mathbf{x}^e(\mathbf{X}, \boldsymbol{\mu}, t) &:= \mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t)|_{\mathbf{X} \in \mathcal{E}_e} = \sum_{i \in \mathcal{N}(e)} N_i(\mathbf{X}) \mathbf{x}_i(\boldsymbol{\mu}, t) \\ \dot{\mathbf{x}}^e(\mathbf{X}, \boldsymbol{\mu}, t) &:= \dot{\mathbf{x}}(\mathbf{X}, \boldsymbol{\mu}, t)|_{\mathbf{X} \in \mathcal{E}_e} = \sum_{i \in \mathcal{N}(e)} N_i(\mathbf{X}) \dot{\mathbf{x}}_i(\boldsymbol{\mu}, t),\end{aligned}\tag{1.21}$$

where  $\mathcal{E}_e$  is element  $e$  in the reference mesh,  $\mathcal{N}(e)$  are the nodes associated with element  $e$ ,  $N_i(\mathbf{X})$  are the DG shape functions on the reference mesh, and  $\mathbf{x}^e$  are the coordinates of the nodes of element  $e$  in the physical domain. An implication of defining the ALE mapping with the DG shape function is the mapping is *discontinuous* between elements, which does not present a problem for the DG method. The expression for the mapping in (1.21) implies that the deformation gradient and its determinant can be easily computed as

$$\mathbf{G}^e(\mathbf{X}, \boldsymbol{\mu}, t) := \mathbf{G}(\mathbf{X}, \boldsymbol{\mu}, t)|_{\mathcal{E}_e} = \sum_{i \in \mathcal{N}(e)} \mathbf{x}_i(\boldsymbol{\mu}, t) \frac{\partial N_i}{\partial \mathbf{X}}(\mathbf{X}), \quad g^e(\mathbf{X}, \boldsymbol{\mu}, t) = \det \mathbf{G}^e(\mathbf{X}, \boldsymbol{\mu}, t).\tag{1.22}$$

Therefore, once the nodal coordinates of the mapping and its time derivatives are known, all the remaining terms directly follow. The implications of such a dependence in the implementation of the adjoint method were discussed in [63] and will be further detailed in Section 1.5.1.

### 1.3.2 Temporal discretization: diagonally implicit Runge-Kutta

The system of ODEs in (1.18) are discretized in time using Diagonally Implicit Runge-Kutta (DIRK) schemes. These schemes are capable of achieving high-order accuracy with the desired stability properties (unlike high-order multistep schemes that are only stable up to second order), without requiring the solution of an enlarged system of equations like general Implicit Runge-Kutta (IRK) schemes (see [36] for an efficient solver for DG-IRK discretizations). DIRK schemes are defined by a *lower triangular* Butcher tableau (Table 1.1) and take the following form when applied to (1.18)

$$\begin{aligned}\mathbf{u}_0 &= \bar{\mathbf{u}}(\boldsymbol{\mu}) \\ \mathbf{u}_n &= \mathbf{u}_{n-1} + \sum_{i=1}^s b_i \mathbf{k}_{n,i} \\ \mathbf{M} \mathbf{k}_{n,i} &= \Delta t_n \mathbf{r}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n),\end{aligned}\tag{1.23}$$

for  $n = 1, \dots, N_t$  and  $i = 1, \dots, s$ , where  $N_t$  are the number of time steps in the temporal discretization and  $s$  is the number of stages in the DIRK scheme. The initial condition,  $\bar{\mathbf{u}}(\boldsymbol{\mu})$ , corresponds to the vectorization of the ALE-transformed uniform flow state in (1.16). The temporal domain,  $(0, T]$  is discretized into  $N_t$  segments with endpoints  $\{t_0, t_1, \dots, t_{N_t}\}$ , with the  $n$ th segment having length  $\Delta t_n = t_n - t_{n-1}$  for  $n = 1, \dots, N_t$ . Additionally, in (1.23),  $\mathbf{u}_{n,i}$  is used to denote the approximation of  $\mathbf{u}_n$  at the  $i$ th stage of time step  $n$

$$\mathbf{u}_{n,i} = \mathbf{u}_{n,i}(\mathbf{u}_{n-1}, \mathbf{k}_{n,1}, \dots, \mathbf{k}_{n,s}) = \mathbf{u}_{n-1} + \sum_{j=1}^i a_{ij} \mathbf{k}_{n,j}.\tag{1.24}$$

From (1.23), a complete time step requires the solution of a sequence of  $s$  nonlinear systems of equation of size  $N_{\mathbf{u}}$ .

$c_1$	$a_{11}$			
$c_2$	$a_{21}$	$a_{22}$		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$c_s$	$a_{s1}$	$a_{s2}$	$\cdots$	$a_{ss}$
	$b_1$	$b_2$	$\cdots$	$b_s$

Table 1.1: Butcher Tableau for  $s$ -stage diagonally implicit Runge-Kutta scheme

### 1.3.3 Solver-consistent discretization of quantities of interest

In this work, quantities of interest that take the form of space-time integrals of nonlinear functions that depends on the solution of the conservation law are discretized in a solver-consistent manner [63], i.e., using the same spatial and temporal discretization used for the conservation law. This ensures the truncation error of the quantities of interest exactly match that of the governing equations.

Consider a quantity of interest of the form

$$\mathcal{F}(\mathbf{U}, \boldsymbol{\mu}, t) = \int_0^t \int_{\Gamma} w(\mathbf{x}, \tau) f(\mathbf{U}(\mathbf{x}, \tau), \boldsymbol{\mu}, \tau) dS d\tau. \quad (1.25)$$

In the context of the optimization problem in (1.1),  $\mathcal{F}$  corresponds to either the objective or a constraint function. Define  $f^h$  as the approximation of  $\int_{\Gamma} w(\mathbf{x}, t) f(\mathbf{U}(\mathbf{x}, t), \boldsymbol{\mu}, t) dS$  using the DG shape functions from the spatial discretization of the governing equations. The solver-consistent spatial discretization of (1.25) becomes

$$\mathcal{F}^h(\mathbf{u}, \boldsymbol{\mu}, t) = \int_0^t f^h(\mathbf{u}, \boldsymbol{\mu}, \tau) d\tau, \quad (1.26)$$

which ensures the spatial integration error in the quantity of interest exactly matches that of the governing equations. Solver-consistent temporal discretization requires the semi-discrete functional in (1.26) be converted to an ODE, which is accomplished via differentiation of (1.26) with respect to  $t$

$$\dot{\mathcal{F}}^h(\mathbf{u}, \boldsymbol{\mu}, t) = f^h(\mathbf{u}, \boldsymbol{\mu}, t). \quad (1.27)$$

Augmenting the semi-discrete governing equations with this ODE (1.27) yields the system of ODEs

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathcal{F}}^h \end{bmatrix} = \begin{bmatrix} \mathbf{r}(\mathbf{u}, \boldsymbol{\mu}, t) \\ f^h(\mathbf{u}, \boldsymbol{\mu}, t) \end{bmatrix}. \quad (1.28)$$

Application of the DIRK temporal discretization introduced in Section 1.3.2 yields the fully discrete governing equations and corresponding solver-consistent discretization of the quantity of interest (1.25)

$$\begin{aligned} \mathbf{u}_n &= \mathbf{u}_{n-1} + \sum_{i=1}^s b_i \mathbf{k}_{n,i} \\ \mathcal{F}_n^h &= \mathcal{F}_{n-1}^h + \Delta t_n \sum_{i=1}^s b_i f^h(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n) \\ \mathbf{M} \mathbf{k}_{n,i} &= \Delta t_n \mathbf{r}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n), \end{aligned} \quad (1.29)$$

for  $n = 1, \dots, N_t$ ,  $i = 1, \dots, s$ , and  $\mathbf{u}_{n,i}$  is defined in (1.24). Finally, the functional in (1.25) is evaluated at time  $t = T$  to yield the solver-consistent approximation of  $\mathcal{F}(\mathbf{u}, \boldsymbol{\mu}, T)$

$$F(\mathbf{u}_0, \dots, \mathbf{u}_{N_t}, \mathbf{k}_{1,1}, \dots, \mathbf{k}_{N_t,s}) := \mathcal{F}_{N_t}^h \approx \mathcal{F}(\mathbf{u}, \boldsymbol{\mu}, T). \quad (1.30)$$



Unlike most methods used in the literature for integrating quantities of interest in time, e.g., trapezoidal rule [33, 57, 34, 25, 51], the proposed method relies on the low-order, intermediate RK stages. These stages are combined in such a way that the temporal integral in (1.26) is approximated to high-order. The dependence of the quantity of interest on these stages must be accounted for in the adjoint equations [63, 64], which will be seen in Section 1.4.1.

## 1.4 Fully discrete adjoint method

### 1.4.1 Fully Discrete, Time-Dependent Adjoint Equations

This section summarizes the work in [63] and begins by posing the adjoint equations corresponding to the fully discrete system of conservation laws in (1.23) and the adjoint method for computing the total derivative of the fully discrete quantity of interest without requiring solution sensitivities,  $\frac{\partial \mathbf{u}_n}{\partial \boldsymbol{\mu}}$  and  $\frac{\partial \mathbf{k}_{n,i}}{\partial \boldsymbol{\mu}}$ . Each of the  $N_{\boldsymbol{\mu}}$  solution sensitivities is the solution of the following linear evolution equations

$$\begin{aligned} \frac{\partial \mathbf{u}_0}{\partial \boldsymbol{\mu}} &= \frac{\partial \bar{\mathbf{u}}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) \\ \frac{\partial \mathbf{u}_n}{\partial \boldsymbol{\mu}} &= \frac{\partial \mathbf{u}_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{i=1}^s b_i \frac{\partial \mathbf{k}_{n,i}}{\partial \boldsymbol{\mu}} \\ \frac{\partial \mathbf{u}_{n,i}}{\partial \boldsymbol{\mu}} &= \frac{\partial \mathbf{u}_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^i a_{ij} \frac{\partial \mathbf{k}_{n,i}}{\partial \boldsymbol{\mu}} \\ \mathbf{M} \frac{\partial \mathbf{k}_{n,i}}{\partial \boldsymbol{\mu}} &= \Delta t_n \frac{\partial \mathbf{r}}{\partial \mathbf{u}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n) \frac{\partial \mathbf{u}_{n,i}}{\partial \boldsymbol{\mu}} + \frac{\partial \mathbf{r}}{\partial \boldsymbol{\mu}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n) \end{aligned} \quad (1.31)$$

for  $n = 1, \dots, N_t$  and  $i = 1, \dots, s$ . These equations are solved forward-in-time and therefore the sensitivity simulation can be performed simultaneously with the primal simulation, which eliminates the need to store the primal solution. However, when  $N_{\boldsymbol{\mu}}$  is large, this approach becomes intractable due to the large number of linear evolution equations that must be solved. To avoid the computational burden of computing the state sensitivities, the adjoint equations corresponding to the functional  $F$  and the corresponding dual variables are introduced to eliminate the state sensitivities from the expression for the total derivative of  $F$  with respect to the parameters,  $\boldsymbol{\mu}$ . From the derivation of the adjoint equations, an expression for the reconstruction of the gradient of  $F$ , independent of the state variables sensitivities, follows naturally. At this point, it is emphasized that  $F$  represents *any* quantity of interest whose gradient is desired, such as the optimization objective function or a constraint.

Let  $\boldsymbol{\lambda}_n$  for  $n = 0, \dots, N_t$  be the adjoint variables corresponding to the state update equation in (1.23) and let  $\boldsymbol{\kappa}_{n,i}$  for  $n = 1, \dots, N_t$  and  $i = 1, \dots, s$  be those corresponding to the stage update equations in (1.23). The adjoint equations are

$$\begin{aligned} \boldsymbol{\lambda}_{N_t} &= \frac{\partial F}{\partial \mathbf{u}_{N_t}}^T \\ \boldsymbol{\lambda}_{n-1} &= \boldsymbol{\lambda}_n + \frac{\partial F}{\partial \mathbf{u}_{n-1}}^T + \sum_{i=1}^s \Delta t_n \frac{\partial \mathbf{r}}{\partial \mathbf{u}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n)^T \boldsymbol{\kappa}_{n,i} \\ \mathbf{M}^T \boldsymbol{\kappa}_{n,i} &= \frac{\partial F}{\partial \mathbf{k}_{n,i}}^T + b_i \boldsymbol{\lambda}_n + \sum_{j=i}^s a_{ji} \Delta t_n \frac{\partial \mathbf{r}}{\partial \mathbf{u}}(\mathbf{u}_{n,j}, \boldsymbol{\mu}, t_{n-1} + c_j \Delta t_n)^T \boldsymbol{\kappa}_{n,j} \end{aligned} \quad (1.32)$$

for  $n = 1, \dots, N_t$  and  $i = 1, \dots, s$  and the expression for  $dF/d\boldsymbol{\mu}$ , independent of state sensitivities, is

$$\frac{dF}{d\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}_0^T \frac{\partial \bar{\mathbf{u}}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^s \boldsymbol{\kappa}_{n,i}^T \frac{\partial \mathbf{r}}{\partial \boldsymbol{\mu}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n). \quad (1.33)$$

Unlike the sensitivity equations in (1.31), the adjoint equations must be solved backward-in-time and the adjoint simulation cannot begin until the primal simulation completes. This implies the entire primal time history, including intermediate stages, must be stored. In our setting, this I/O cost is negligible in comparison to the cost of a linear solve with the Jacobian matrix. Furthermore, in contrast to the sensitivity equations, the derivative of the quantity of interest with respect to the state variable appears as a *forcing* term in (1.32), which requires a separate set of adjoint variables for each quantity of interest whose derivative is sought. In an gradient-based optimization setting, this implies  $N_c + 1$ , where  $N_c$  is the number of state-dependent constraints, adjoint solves are required to compute the gradient of the objective function and all constraint functions. While the number of adjoint solves depends on the number of functionals to differentiate, it is independent of the number of parameters. Since the application in this work is in the regime where  $N_{\boldsymbol{\mu}} > N_c + 1$ , the adjoint method is more desirable.

For the derivation of equations (1.32)-(1.33), the reader is referred to [62, 63]. For the adjoint equations that explicitly enforce time-periodicity of the solution of the partial differential equation, see [64]. From inspection of (1.33), it is clear that the initial condition sensitivity  $\frac{\partial \bar{\mathbf{u}}}{\partial \boldsymbol{\mu}}$  is the only sensitivity term required to reconstruct  $\frac{dF}{d\boldsymbol{\mu}}$ . The derivation of this term for the uniform flow initial condition introduced in Section 1.2.3 is provided in the next section. From the expression for the fully discrete quantity of interest in (1.30), it is clear that  $F$  is *independent* of  $\mathbf{u}_{N_t}$ , which implies

$$\boldsymbol{\lambda}_{N_t} = \frac{\partial F}{\partial \mathbf{u}_{N_t}} = \mathbf{0}. \quad (1.34)$$

Furthermore, the partial derivatives of the fully discrete quantities of interest are

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{u}_n} &= \Delta t_n \sum_{i=1}^s b_i \frac{\partial f^h}{\partial \mathbf{u}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n) & n = 0, \dots, N_t - 1 \\ \frac{\partial F}{\partial \mathbf{k}_{n,j}} &= \Delta t_n \sum_{i=j}^s a_{ij} b_i \frac{\partial f^h}{\partial \mathbf{u}}(\mathbf{u}_{n,i}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n) & n = 1, \dots, N_t, j = 1, \dots, s \end{aligned} \quad (1.35)$$

See [63] for a discussion of the benefits of the fully discrete adjoint framework over the continuous or semi-discrete ones in the context of optimization *or* when a Runge-Kutta temporal discretization is used.

#### 1.4.2 Parametrization of the initial condition

Recall the form of the ALE-transformed uniform flow initial condition in (1.16). Since the physical uniform flow state  $\bar{\mathbf{U}}(\mathbf{X})$  is parameter-independent the sensitivity of the initial condition will be due solely to the sensitivity of the determinant of the deformation gradient. That is,

$$\begin{aligned} \frac{\partial \mathbf{U}_{\bar{\mathbf{X}}}}{\partial \boldsymbol{\mu}}(\mathbf{X}, \boldsymbol{\mu}, 0) &= \bar{\mathbf{U}}(\mathbf{X}) \frac{\partial g_0}{\partial \boldsymbol{\mu}}(\mathbf{X}, \boldsymbol{\mu}) \\ \frac{\partial \bar{g}}{\partial \boldsymbol{\mu}}(\mathbf{X}, \boldsymbol{\mu}, 0) &= \frac{\partial g_0}{\partial \boldsymbol{\mu}}(\mathbf{X}, \boldsymbol{\mu}) \end{aligned} \quad (1.36)$$

The initial condition sensitivity at the semi-discrete or fully discrete level is then the appropriate vectorization of this quantity over the DG mesh.

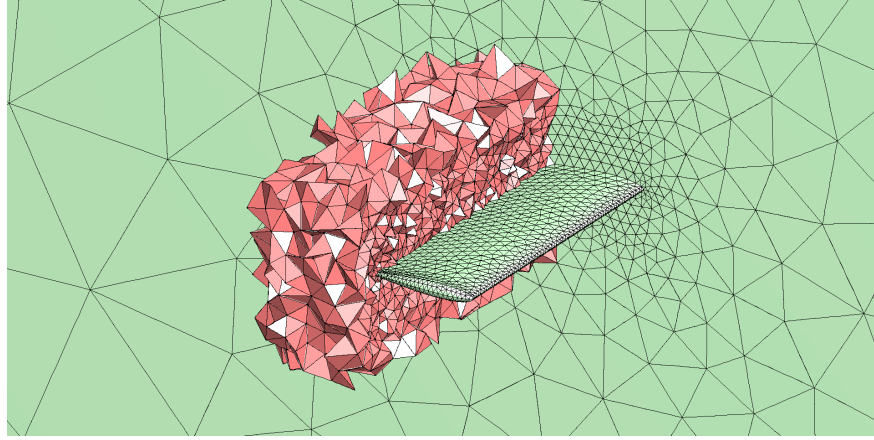


Fig. 1.2: Surface mesh of the wing and the symmetry plane, and some of the tetrahedral elements in the wake. All elements are curved by polynomials of degree  $p = 3$ .

### 1.4.3 Parametrization of the residual and quantities of interest

In addition to the initial condition sensitivity, the equation to reconstruct the total derivative of  $F$  with respect to  $\boldsymbol{\mu}$  requires the partial derivatives of the residual and quantity of interest with respect to the  $\boldsymbol{\mu}$ . For this purpose, we assume the parameter vector  $\boldsymbol{\mu}$  purely controls the domain deformation, e.g., it does not affect the boundary conditions or material properties. Then, given the discussion in Section 1.3.1 that completely defines the ALE map based on its action and the action of its time derivative on the nodes of the mesh, the parameter dependence of the residual and quantity of interest can be written in terms of  $\mathbf{x}(\boldsymbol{\mu})$  and  $\dot{\mathbf{x}}(\boldsymbol{\mu})$ . That is,

$$\begin{aligned}\frac{\partial \mathbf{r}}{\partial \boldsymbol{\mu}} &= \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\mu}} + \frac{\partial \mathbf{r}}{\partial \dot{\mathbf{x}}} \frac{\partial \dot{\mathbf{x}}}{\partial \boldsymbol{\mu}} \\ \frac{\partial f^h}{\partial \boldsymbol{\mu}} &= \frac{\partial f^h}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f^h}{\partial \dot{\mathbf{x}}} \frac{\partial \dot{\mathbf{x}}}{\partial \boldsymbol{\mu}}\end{aligned}\tag{1.37}$$

The form of the ALE map, i.e.,  $\mathbf{x}(\boldsymbol{\mu})$  and  $\dot{\mathbf{x}}(\boldsymbol{\mu})$ , will be described in the next section. Our implementation uses the Maple software [31] to compute all required partial derivatives.

## 1.5 Application to energetically optimal flapping flight

In this section, the high-order numerical discretization of the isentropic, compressible Navier-Stokes equations and corresponding adjoint method are applied to determine the energetically optimal flapping motion of a three-dimensional wing geometry using gradient-based optimization in the low Reynolds number regime of  $Re = 1000$ . For a physically relevant mission, a requirement is placed on the time-averaged thrust, which leads to an optimization problem with a nonlinear constraint. As a result, two adjoint equations must be solved at each optimization iteration to compute the gradient of the objective function and the nonlinear constraint.

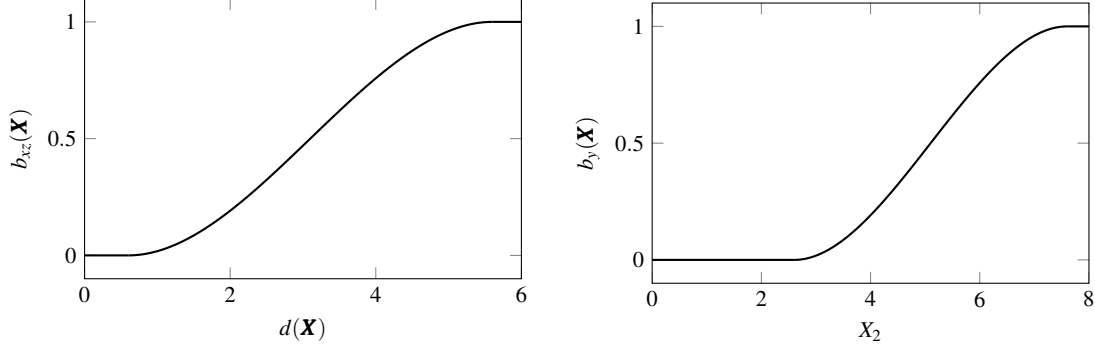


Fig. 1.3: *Left*: Radial blending  $b_{xz}(\mathbf{X})$  corresponding to  $r_1 = 0.6$ ,  $r_2 = 5$ . *Right*: Unidirectional blending  $b_y(\mathbf{X})$  corresponding to  $y_1 = 2.6$ ,  $y_2 = 5$ .

### 1.5.1 Flapping wing geometry and kinematics

The wing geometry considered in this work is an extruded NACA0012 airfoil with a rounded tip to accurately capture three-dimensional effects. In the reference configuration, the NACA0012 airfoil is contained in the  $X_1 - X_3$  plane corresponding to  $X_2 = 0$ , facing the  $-X_1$  direction (flow in the  $+X_1$  direction), and extruded in the  $+X_2$  direction for the span-to-chord ratio of 2. A symmetry plane is included to consider an isolated wing without a fuselage. The fluid domain is discretized using a curved mesh with tetrahedral elements of degree  $p = 3$ . Fig. 1.2 visualizes the mesh and the corresponding geometry is taken as the reference domain in ALE setting.

The flapping motion is parametrized using three angles: the flapping angle  $\theta(\boldsymbol{\mu}, t)$  (rotation about the  $X_1$ -axis), the pitching angle  $\alpha(\boldsymbol{\mu}, t)$  (rotation about the  $X_2$ -axis), and the sweeping angle  $\beta(\boldsymbol{\mu}, t)$  (rotation about the  $X_3$ -axis). The origin of the flapping angle is taken as the intersection of the  $X_2 = s_1$  and  $X_3 = 0$  planes, where  $s_1 > 0$  is a parameter that defines a shoulder away from the symmetry plane. The origin of the pitching and sweeping angles are taken as the intersection of the  $X_1 = 0$ ,  $X_3 = 0$ , and  $X_1 = 0$ ,  $X_2 = 0$  planes, respectively. The combination of these motions takes the form

$$\begin{aligned} x'_1(\mathbf{X}, \boldsymbol{\mu}, t) &= X_1 \cos(\alpha(\boldsymbol{\mu}, t)) + (X_2 - s_1) \sin(\beta(\boldsymbol{\mu}, t)) - X_3 \sin(\alpha(\boldsymbol{\mu}, t)) \\ x'_2(\mathbf{X}, \boldsymbol{\mu}, t) &= s_1 + X_2 \cos(\theta(\boldsymbol{\mu}, t)) \cos(\beta(\boldsymbol{\mu}, t)) - X_3 \sin(\theta(\boldsymbol{\mu}, t)) - X_1 \sin(\beta(\boldsymbol{\mu}, t)) \\ x'_3(\mathbf{X}, \boldsymbol{\mu}, t) &= X_3 \cos(\theta(\boldsymbol{\mu}, t)) \cos(\alpha(\boldsymbol{\mu}, t)) + (X_2 - s_1) \sin(\theta(\boldsymbol{\mu}, t)) + X_1 \sin(\alpha(\boldsymbol{\mu}, t)), \end{aligned} \quad (1.38)$$

where we set the parameter  $s_1 = 0.5$ . While this kinematic description encodes exactly the desired motion of the wing itself, it cannot be applied to the entire fluid domain as it will not preserve the symmetry plane and the rotations will lead to large velocities at the farfield. To avoid these issues, the domain deformation is smoothly blended to zero near the symmetry plane and away from the wing, following the work in [38, 63].

The deformation blending away from the wing is defined as a composition of a radial blending,  $b_{xz}(\mathbf{X})$ , in the  $X_1 - X_3$  plane and a unidirectional blending,  $b_y(\mathbf{X})$ , in the  $+X_2$  direction. These blendings take the form

$$b_{xz}(\mathbf{X}) = \begin{cases} 0 & d(\mathbf{X}) \leq r_1 \\ 1 & d(\mathbf{X}) \geq r_1 + r_2, \\ q\left(\frac{d(\mathbf{X}) - r_1}{r_2}\right) & \text{otherwise} \end{cases}, \quad b_y(\mathbf{X}) = \begin{cases} 0 & X_2 \leq y_1 \\ 1 & X_2 \geq y_1 + y_2, \\ q\left(\frac{X_2 - y_1}{y_2}\right) & \text{otherwise} \end{cases}, \quad (1.39)$$

where  $d(\mathbf{X}) = \sqrt{X_1^2 + X_3^2}$  is the radial distance from the  $X_2$  axis (the axis through the center of the wing in the spanwise direction) and  $q(s) = 3s^2 - 2s^3$  is the cubic blending introduced in [38]. For smoother spatial blendings, the quintic expression  $q(s) = 10s^3 - 15s^4 + 6s^5$  could be used instead. See Fig. 1.3 for the blendings  $b_{xz}(\mathbf{X})$  and  $b_y(\mathbf{X})$  with the values of the blending parameters used in this work:  $r_1 = 0.6$ ,  $r_2 = 5$ ,  $y_1 = 2.6$ ,  $y_2 = 5$ .

Suppose we want to compose two blendings,  $b_1(\mathbf{X})$  and  $b_2(\mathbf{X})$ , in serial, that is, blend a deformed domain  $\mathbf{x}'$  with an undeformed domain  $\mathbf{X}$  via  $b_1(\mathbf{X})$  and blend the result with the undeformed domain via  $b_2(\mathbf{X})$  as follows

$$\begin{aligned}\mathbf{x}'' &= (1 - b_1(\mathbf{X}))\mathbf{x}' + b_1(\mathbf{X})\mathbf{X} \\ \mathbf{x} &= (1 - b_2(\mathbf{X}))\mathbf{x}'' + b_2(\mathbf{X})\mathbf{X}.\end{aligned}\tag{1.40}$$

This can be compactly expressed as a single blending  $b_{12}(\mathbf{X})$  as  $\mathbf{x} = (1 - b_{12}(\mathbf{X}))\mathbf{x}' + b_{12}(\mathbf{X})\mathbf{X}$ , where

$$b_{12}(\mathbf{X}) = b_1(\mathbf{X}) + b_2(\mathbf{X}) - b_1(\mathbf{X})b_2(\mathbf{X}).\tag{1.41}$$

Therefore, the composition of the radial and unidirectional blending in (1.39) leads to a cylindrical blending that takes the form

$$b_{cyl}(\mathbf{X}) = b_{xz}(\mathbf{X}) + b_y(\mathbf{X}) - b_{xz}(\mathbf{X})b_y(\mathbf{X}).\tag{1.42}$$

To ensure the symmetry plane remains motionless, the mapping in (1.38) must smoothly blended to 0 at the  $X_2 = 0$  plane. The blending at the symmetry plane,  $b_{sym}(\mathbf{X})$ , is chosen to be infinitely smooth and the rate of decay decreases with increasing radial distance from the  $X_2$  axis to prevent mesh entanglement, i.e.,

$$b_{sym}(\mathbf{X}) = e^{-(X_2/(s_2+s_3d(\mathbf{X})))^2}.\tag{1.43}$$

The blending parameter  $s_2$  is set to 1 for geometrical considerations since this affects the geometry of the wing during the flapping motion. The blending parameter  $s_3$  is free in the sense that it has little effect on the wing itself and is solely used to improve mesh quality in the fluid domain. A brute force, unidimensional search is performed to determine the value of  $s_3 = 0.3$  that maximizes the mesh quality. See Fig. 1.4 for a plot of  $b_{sym}(\mathbf{X})$  with these blending parameters at various radial positions.

The composition of the cylindrical blending  $b_{cyl}(\mathbf{X})$  and symmetry blending  $b_{sym}(\mathbf{X})$  using the formula in (1.41) leads to the final form of the spatial blending

$$b(\mathbf{X}) = b_{cyl}(\mathbf{X}) + b_{sym}(\mathbf{X}) - b_{cyl}(\mathbf{X})b_{sym}(\mathbf{X})\tag{1.44}$$

and the expression for the deformed domain

$$\mathbf{x}''(\mathbf{X}, \boldsymbol{\mu}, t) = (1 - b(\boldsymbol{\phi}(\mathbf{X})))\mathbf{x}'(\boldsymbol{\phi}(\mathbf{X}), \boldsymbol{\mu}, t) + b(\boldsymbol{\phi}(\mathbf{X}))\boldsymbol{\phi}(\mathbf{X}).\tag{1.45}$$

The above expression uses  $\boldsymbol{\phi}(\mathbf{X})$ , the coordinates in the domain with curved boundaries, in place of  $\mathbf{X}$ , the coordinates in the straight-sided domain, due to the choice discussed in Section 1.3.1 that incorporates the curving of the domain boundaries in the ALE map. Spatial blending of this form ensures the desired physical motion of the body,  $\mathbf{x}'(\mathbf{X}, \boldsymbol{\mu}, t)$ , is exactly achieved near the surface of the wing, there is no deformation far from the surface or at the symmetry plane, and the domain deformation smoothly varies between these extremes.

The expression for the deformed domain,  $\mathbf{x}'(\mathbf{X}, \boldsymbol{\mu}, t)$ , in (1.38) will have a non-trivial deformation and velocity at  $t = 0$ . This may cause difficulty in initializing the simulation from uniform flow as violent transients will result that may prevent convergence of the nonlinear solvers. For this reason, following the work in [51, 63], the deformation is smoothly blended to zero at  $t = 0$  using the infinitely differentiable blending

$$b_t(t) = e^{-(t/T_c)^2}.\tag{1.46}$$

Temporal blendings have also been used in experimental studies involving flapping wings [18], where a quintic blending was used. The final form of the deformed domain is

$$\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t) := (1 - b_t(t))\mathbf{x}''(\boldsymbol{\phi}(\mathbf{X}), \boldsymbol{\mu}, t) + b_t(t)\boldsymbol{\phi}(\mathbf{X})\tag{1.47}$$

and the domain velocity  $\dot{\mathbf{x}}(\mathbf{X}, \boldsymbol{\mu}, t)$  can be computed analytically. It can easily be verified that this temporal blending guarantees  $\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, 0) = \boldsymbol{\phi}(\mathbf{X})$  and  $\dot{\mathbf{x}}(\mathbf{X}, \boldsymbol{\mu}, 0) = \mathbf{0}$ . In this work,  $T_c = T/5$ , where  $T$  is the period of the flapping motion, to ensure  $\mathbf{x}, \dot{\mathbf{x}}$  are effectively equal to  $\mathbf{x}'', \dot{\mathbf{x}}''$  (within 0.1%) by 1/2 a period (see Fig. 1.4). This blending

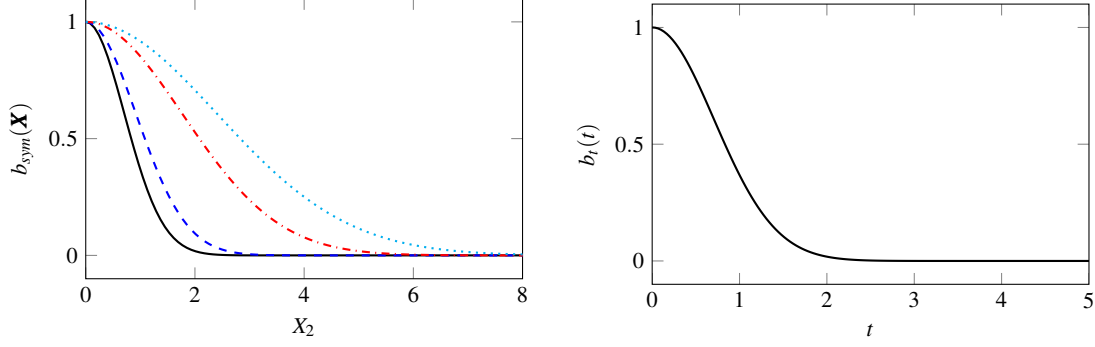


Fig. 1.4: *Left*: Blending at symmetry plane  $b_{sym}(\mathbf{X})$  corresponding to  $s_2 = 1.0$ ,  $s_3 = 0.3$  at radial position  $d(\mathbf{X}) = 0$  (—),  $d(\mathbf{X}) = 1$  (---),  $d(\mathbf{X}) = 5$  (-.-.),  $d(\mathbf{X}) = 8$  (.....). *Right*: Temporal blending  $b_t(t)$  corresponding to  $T_c = 1$ .

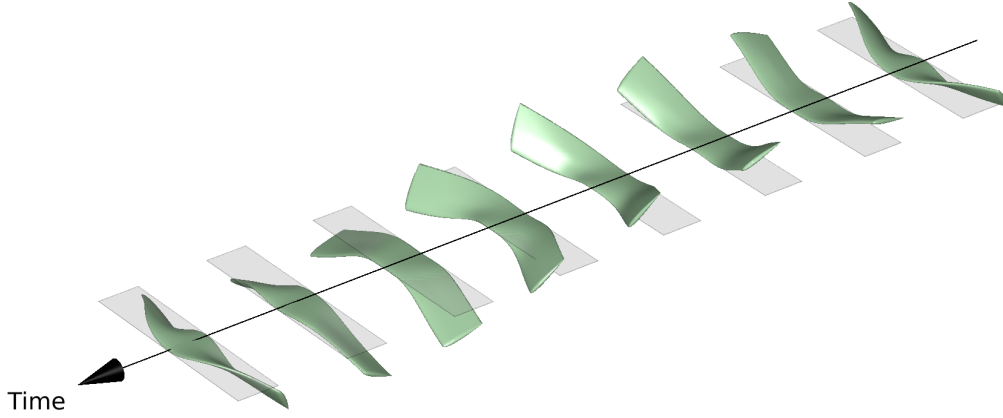


Fig. 1.5: Snapshots of the flapping motion in (1.47) with  $\mu_1 = \mu_4 = \mu_7 = \mu_9 = 0$ ,  $\mu_3 = -\mu_6 = -\pi/3$ ,  $\mu_2 = 60^\circ$ ,  $\mu_5 = -\mu_8 = -25^\circ$ .

limits the transients that result from initializing the flow with incompatible boundary conditions at the viscous wall. Another implication of this temporal blending is that the sensitivity of the initial condition is zero, i.e.,  $\frac{\partial \mathbf{u}_0}{\partial \boldsymbol{\mu}} = 0$ , since  $\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, 0) = \boldsymbol{\phi}(\mathbf{X})$ . Finally, as discussed in Section 1.3.1, once the ALE-mapped domain  $\mathbf{x}(\mathbf{X}, \boldsymbol{\mu}, t)$  and velocity  $\dot{\mathbf{x}}(\mathbf{X}, \boldsymbol{\mu}, t)$  are computed, the remaining quantities required for the ALE formulation of the governing equations, namely  $\mathbf{G}(\mathbf{X}, \boldsymbol{\mu}, t)$  and  $g(\mathbf{X}, \boldsymbol{\mu}, t)$ , can be computed through differentiation of the underlying shape functions, as in (1.22).

Given this kinematic description of the flapping motion in (1.47), all that remains to completely specify the domain deformation and its parametrization is the functional form of the pitching, sweeping, and flapping angles. In this work, these angles are parametrized through a single harmonic function each as

$$\begin{aligned}
 \alpha(\boldsymbol{\mu}, t) &= \mu_1 + \mu_2 \sin(2\pi ft + \mu_3) \\
 \beta(\boldsymbol{\mu}, t) &= \mu_4 + \mu_5 \sin(2\pi ft + \mu_6) \\
 \theta(\boldsymbol{\mu}, t) &= \mu_7 + \mu_8 \sin(2\pi ft + \mu_9),
 \end{aligned} \tag{1.48}$$

where  $f = 1/T$  is the flapping frequency. Even though the flapping frequency is an important design consideration, it will not be taken as a parameter in this work as properly accounting for frequency perturbations in the fully discrete adjoint framework is still a research issue [53] and will be the subject of future work. An example of a typical flapping motion is shown in Fig. 1.5. The same flapping motion is shown in Fig. 1.5-1.8, where the various views of an unstructured volumetric mesh with 10805  $p = 3$  elements are provided to show the impact of the blending and the high-quality elements that are maintained.

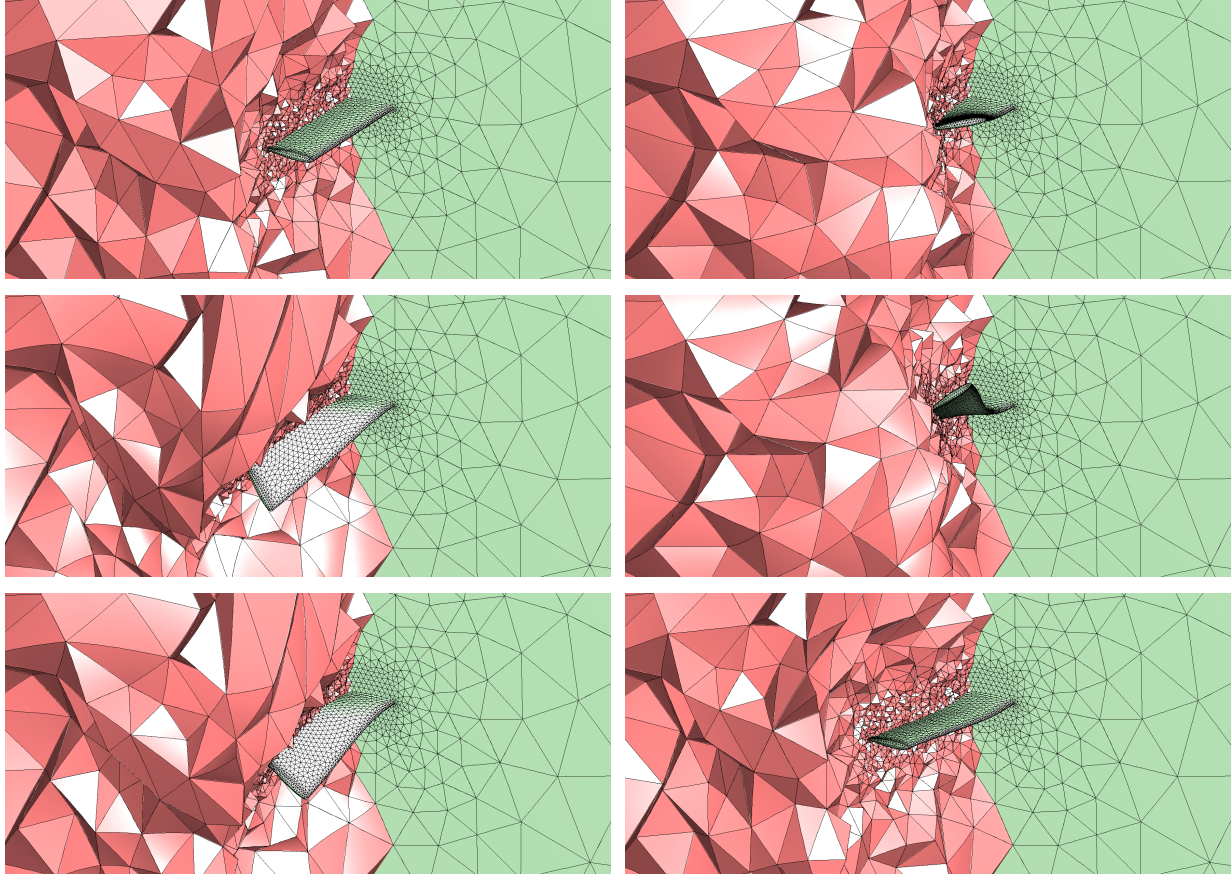


Fig. 1.6: Snapshots of a slice of the volumetric mesh in the  $X_1 - X_3$  and  $X_2 - X_3$  planes corresponding to the flapping motion in (1.47) with  $\mu_1 = \mu_4 = \mu_7 = \mu_9 = 0$ ,  $\mu_3 = -\mu_6 = -\pi/3$ ,  $\mu_2 = 60^\circ$ ,  $\mu_5 = -\mu_8 = -25^\circ$ . The top left figure corresponds to the *curved mesh* with no other deformation applied, i.e.,  $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X})$ . The remaining figures correspond to snapshots (top to bottom, left to right) taken at equally spaced time increments during the second period that correspond to times  $t = 5.0, 6.0, 7.0, 8.0, 9.0$ .

### 1.5.2 Energetically optimal flapping under a thrust constraint

The high-order numerical discretization of the isentropic, compressible Navier-Stokes equations and corresponding adjoint method are applied to determine the energetically optimal flapping motion of the geometry introduced in the previous section using gradient-based optimization techniques in the low Reynolds number regime of  $Re = 1000$ . For a physically relevant mission, requirements are placed on the time-averaged thrust leading to an optimization problem with a nonlinear constraint. As a result, two adjoint equations must be solved at each optimization iteration to compute the gradient of the objective function and nonlinear constraint. From (1.32), it is clear that the linear system that arises at each stage of each time step is the same for each functional; the only difference is the right-hand side, which presents an opportunity to use some fast multiple right-hand side solver [45, 8]; however, this was not done in this work.

The DG-ALE scheme introduced in Section 1.2 is used for the spatial discretization of the system of conservation laws with polynomial order  $p = 3$  (for both the geometry and solution representation) and a diagonally implicit Runge-Kutta scheme for the temporal discretization. The DG-ALE scheme uses the Roe flux [41] for the inviscid numerical flux and the Compact DG flux [37] for the viscous numerical flux. The Butcher tableau for the three-stage, third-order DIRK scheme considered in this work is given in Table 1.2. Since the present study looks to find the energetically optimal flapping motion subject to a constraint on the thrust, the quantities of interest for the optimization problem are the average work done on the fluid by the wing,  $\mathcal{W}(\mathbf{U}, \boldsymbol{\mu})$ , and thrust,  $\mathcal{T}_x(\mathbf{U}, \boldsymbol{\mu})$ , over one flapping period. To

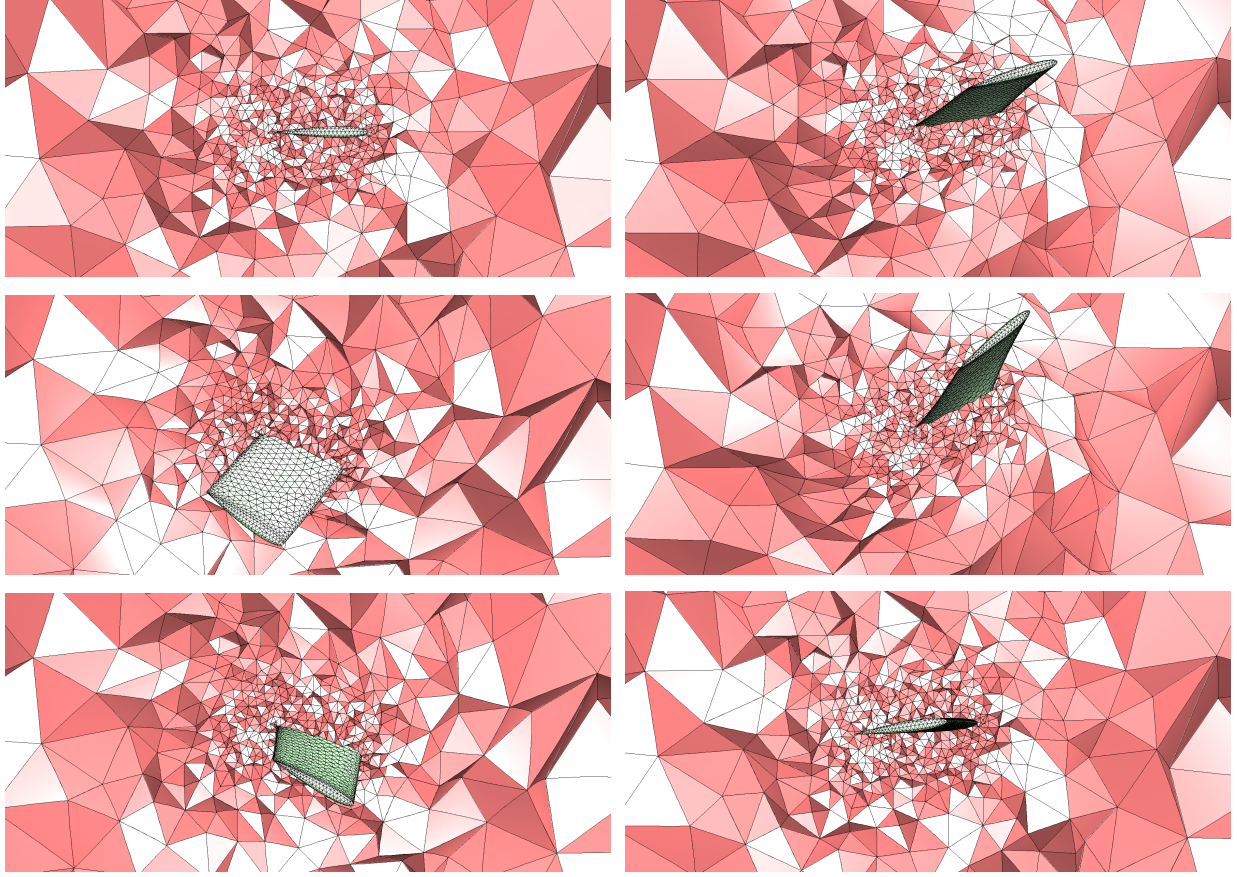


Fig. 1.7: Snapshots of a slice of the volumetric mesh in the  $X_1 - X_3$  plane corresponding to the flapping motion in (1.47) with  $\mu_1 = \mu_4 = \mu_7 = \mu_9 = 0$ ,  $\mu_3 = -\mu_6 = -\pi/3$ ,  $\mu_2 = 60^\circ$ ,  $\mu_5 = -\mu_8 = -25^\circ$ . The top left figure corresponds to the *curved mesh* with no other deformation applied, i.e.,  $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X})$ . The remaining figures correspond to snapshots (top to bottom, left to right) taken at equally spaced time increments during the second period that correspond to times  $t = 5.0, 6.0, 7.0, 8.0, 9.0$ .

$\alpha$	$\alpha$
$\frac{1+\alpha}{2}$	$\frac{1+\alpha}{2} - \alpha$
1	$\gamma$
	$\omega$
	$\gamma$
	$\omega$
	$\alpha$

Table 1.2: Butcher Tableau for 3-stage, 3rd order DIRK scheme [1]

$$\alpha = 0.435866521508459, \gamma = -\frac{6\alpha^2 - 16\alpha + 1}{4}, \omega = \frac{6\alpha^2 - 20\alpha + 5}{4}.$$

ensure the transients that result from initializing the simulation from non-periodic flow (uniform flow in this case) do not pollute the time-averaged quantities, two full periods of the flapping motion are simulated and the quantities are averaged over only the final period. Therefore, the time-averaged quantities are defined as

$$\begin{aligned} \mathcal{W}(\mathbf{U}, \boldsymbol{\mu}) &= -\frac{1}{T} \int_T^{2T} \int_{\Gamma} \mathbf{f}(\mathbf{U}, \boldsymbol{\mu}) \cdot \dot{\mathbf{x}} \, dS \, dt \\ \mathcal{F}_x(\mathbf{U}, \boldsymbol{\mu}) &= -\frac{1}{T} \int_T^{2T} \int_{\Gamma} \mathbf{f}(\mathbf{U}, \boldsymbol{\mu}) \cdot \mathbf{e}_1 \, dS \, dt \end{aligned} \quad (1.49)$$



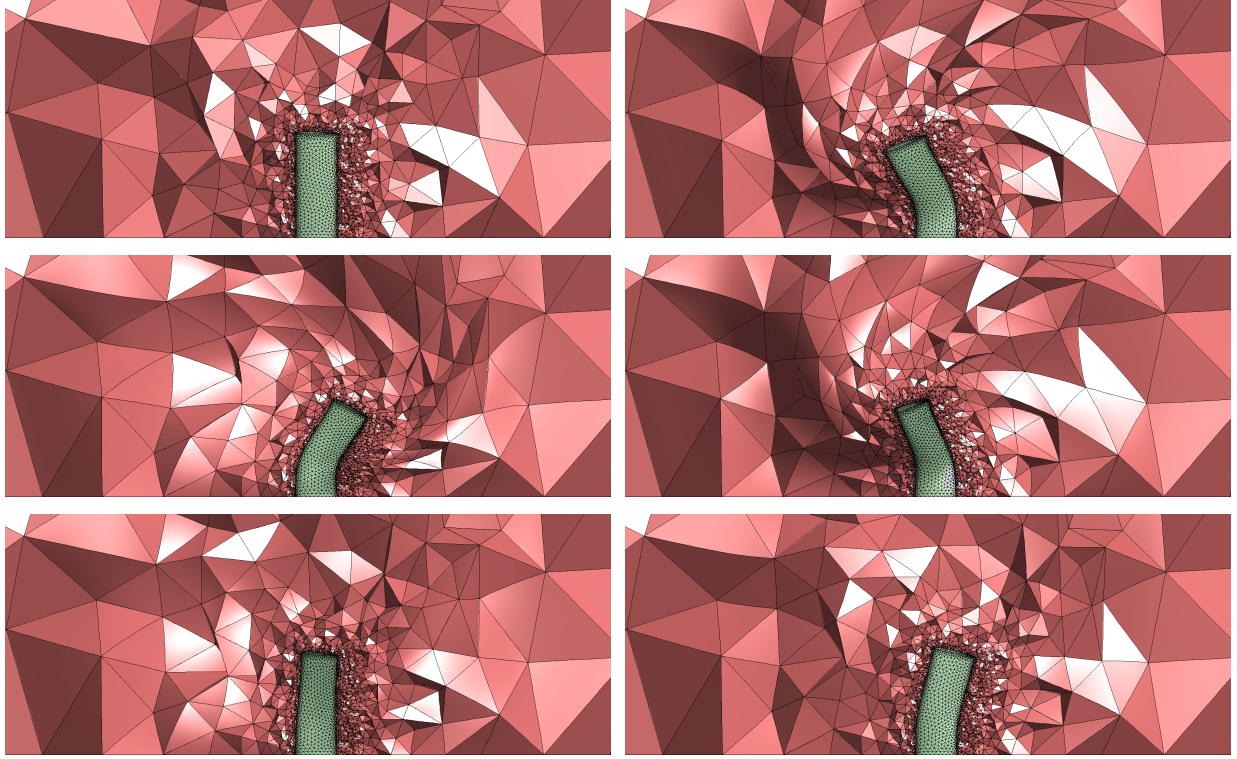


Fig. 1.8: Snapshots of a slice of the volumetric mesh in the  $X_1 - X_2$  plane corresponding to the flapping motion in (1.47) with  $\mu_1 = \mu_4 = \mu_7 = \mu_9 = 0$ ,  $\mu_3 = -\mu_6 = -\pi/3$ ,  $\mu_2 = 60^\circ$ ,  $\mu_5 = -\mu_8 = -25^\circ$ . The top left figure corresponds to the *curved mesh* with no other deformation applied, i.e.,  $\mathbf{x} = \boldsymbol{\phi}(\mathbf{X})$ . The remaining figures correspond to snapshots (top to bottom, left to right) taken at equally spaced time increments during the second period that correspond to times  $t = 5.0, 6.0, 7.0, 8.0, 9.0$ .

where  $\boldsymbol{\Gamma}$  is the surface of the wing,  $\mathbf{f} \in \mathbb{R}^3$  is the force imparted by the fluid on the body,  $\mathbf{e}_i \in \mathbb{R}^3$  is the  $i$ th canonical basis vector, and  $\dot{\mathbf{x}}$  is the velocity of each point on  $\boldsymbol{\Gamma}$ . The negative sign in the definition of the thrust is required since the flow is in the  $+X_1$ -direction and, therefore, a positive thrust is directed in the  $-X_1$ -direction.

The initialization and integration strategy described is a commonly-used and crude approximation to the ideal situation of initializing the simulation with the state that will induce a time-periodic flow, which will ensure the simulation is completely free of unphysical initial transients. A method to initialize a simulation with this state was introduced in [64] as well as the corresponding adjoint method to allow for optimization under a time-periodicity constraint.

Finally, let the fully discrete, high-order approximation of the integrated quantities of interest (DG in space, DIRK in time) in (1.25) be denoted with the corresponding Roman symbol, e.g.,

$W(\mathbf{u}^{(0)}, \dots, \mathbf{u}^{(N_t)}, \mathbf{k}_1^{(n)}, \dots, \mathbf{k}_s^{(n)}, \boldsymbol{\mu})$  is the fully discrete approximation of  $\mathcal{W}(\mathbf{U}, \boldsymbol{\mu})$  and similarly for  $T_x$ . Then the fully discrete optimization problem of interest takes the form

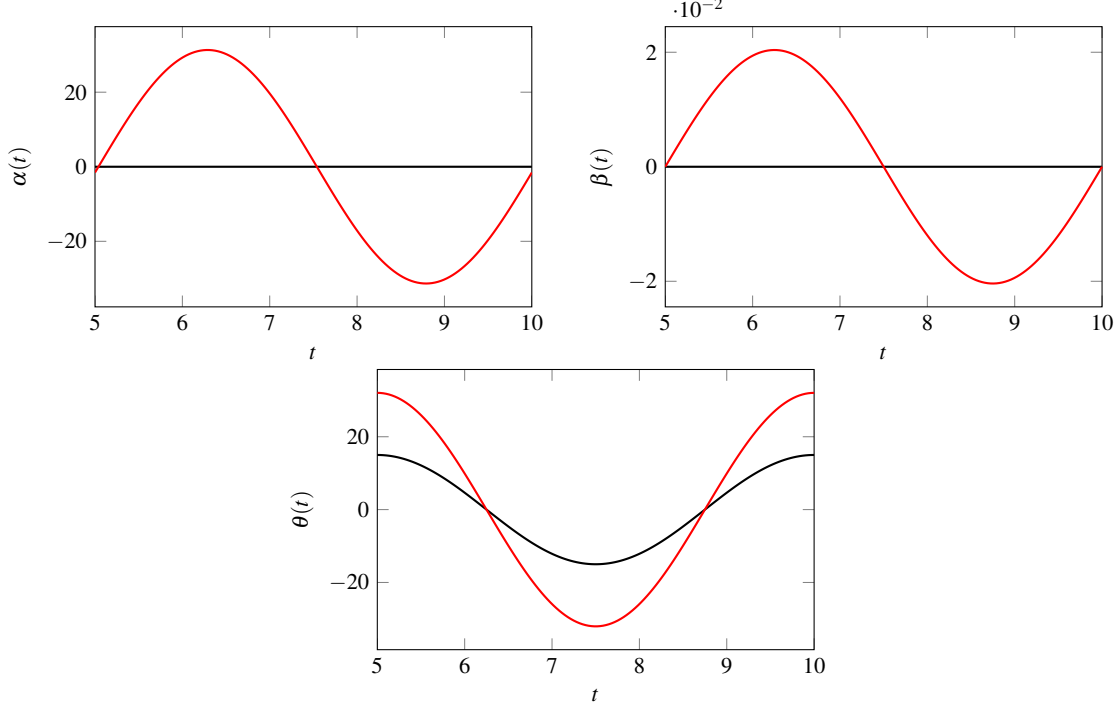


Fig. 1.9: Trajectories of  $\alpha(t)$ ,  $\beta(t)$ , and  $\theta(t)$ , in degrees, at the initial guess (—) and solution (—) of the optimization problem in (1.50).

$$\begin{aligned}
 & \underset{\substack{\mathbf{u}^{(0)}, \dots, \mathbf{u}^{(N_f)} \in \mathbb{R}^{N_u}, \\ \mathbf{k}_1^{(1)}, \dots, \mathbf{k}_s^{(N_f)} \in \mathbb{R}^{N_k}, \\ \boldsymbol{\mu} \in \mathbb{R}^{N_\mu}}}{\text{minimize}} & & W(\mathbf{u}^{(0)}, \dots, \mathbf{u}^{(N_f)}, \mathbf{k}_1^{(1)}, \dots, \mathbf{k}_s^{(N_f)}, \boldsymbol{\mu}) \\
 & \text{subject to} & & T_x(\mathbf{u}^{(0)}, \dots, \mathbf{u}^{(N_f)}, \mathbf{k}_1^{(1)}, \dots, \mathbf{k}_s^{(N_f)}, \boldsymbol{\mu}) \geq \bar{T}_x \\
 & & & \mathbf{u}^{(0)} = \mathbf{u}_0 \\
 & & & \mathbf{u}^{(n)} = \mathbf{u}^{(n-1)} + \sum_{i=1}^s b_i \mathbf{k}_i^{(n)} \\
 & & & \mathbf{M} \mathbf{k}_i^{(n)} = \Delta t_n \mathbf{r}(\mathbf{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n),
 \end{aligned} \tag{1.50}$$

where  $\bar{T}_x$  is a lower bound on the thrust. In this work,  $\bar{T}_x = 0$  is taken to ensure the flapping motion generates sufficient thrust to overcome the induced drag on the wing. In this section, the parameters  $\mu_1 = \mu_4 = \mu_7 = 0$  and  $\mu_9 = \pi/2$  are frozen, which leads to a 5 parameter optimization problem in the all the amplitudes ( $\mu_2, \mu_5, \mu_8$ ) and pitch and sweep phases ( $\mu_3, \mu_6$ ).

The optimization solver used in this work is IPOPT [52], a nonlinearly constrained interior point method. Fig. 1.9 contains the trajectory of  $\alpha$ ,  $\beta$ ,  $\theta$  that define initial guess and solution of the optimization problem in (1.50). The initial guess for the optimization problem is a pure flapping motion, i.e.,  $\alpha(\boldsymbol{\mu}_0, t) = \beta(\boldsymbol{\mu}_0, t) = 0$ . In general, a quality initial guess is important since the solution of non-convex optimization problems, such as this one, are dependent on the starting point. In a practical design setting, the goal is to improve an existing or baseline design, which will usually constitute a reasonable starting guess for the optimizer. Another strategy for generating reasonable initial guesses is to perform homotopy on the thrust constraint. The optimal solution increases the flapping amplitude from  $15^\circ$  to  $32.1^\circ$ , increases the pitch amplitude from  $0^\circ$  to  $31.3^\circ$ , and only incorporates a negligible amount of dynamic sweeping ( $0.02^\circ$ ). The optimal phase angle between the flapping and pitch motions is determined to be  $87.1^\circ$ .

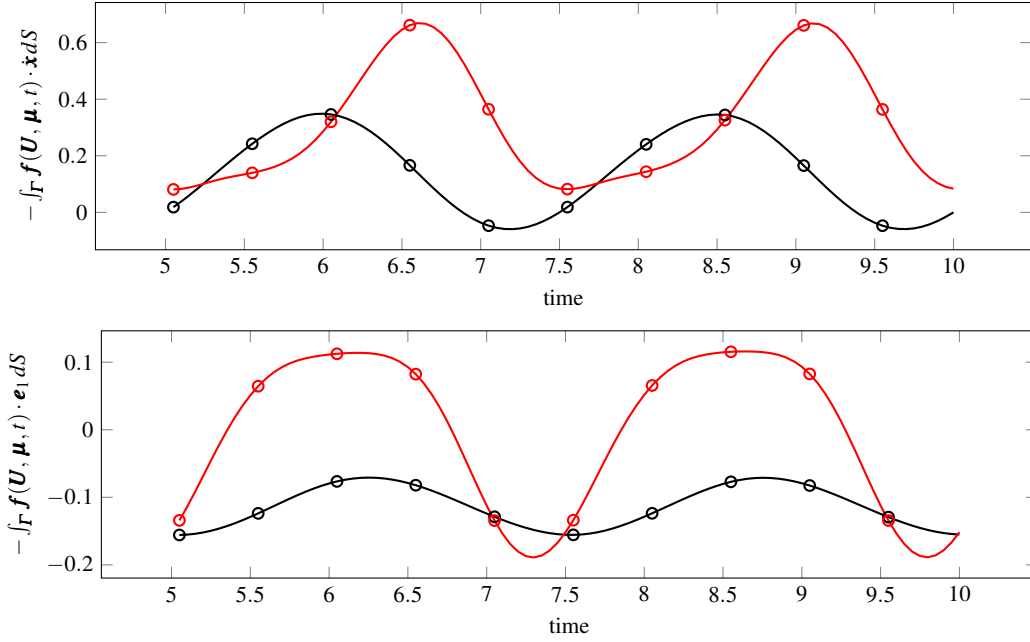


Fig. 1.10: Time of the total power (top) and  $x$ -directed force (bottom) imparted onto the fluid by the airfoil at initial guess ( $\text{---}\circ\text{---}$ ) and optimal solution ( $\text{---}\circ\text{---}$ ).

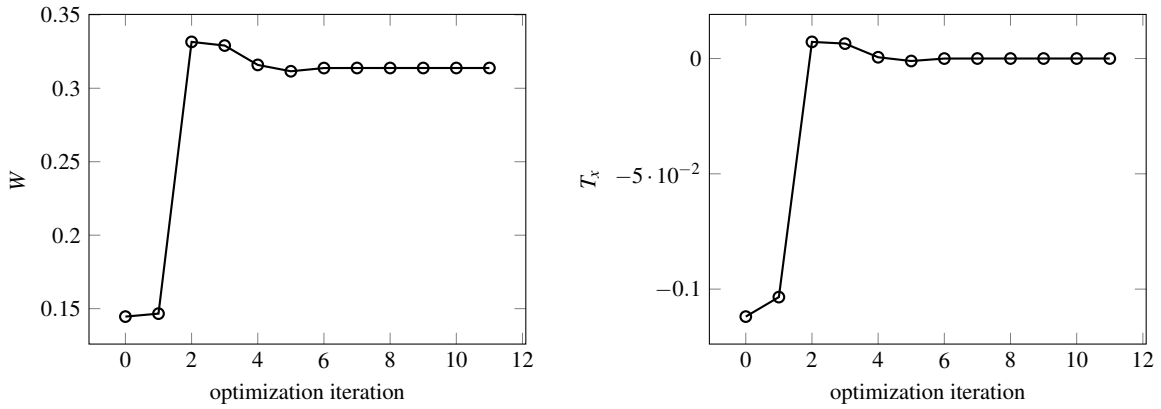


Fig. 1.11: Convergence of quantities of interest,  $W$  and  $T_x$ , with optimization iteration. Each iteration requires a primal and adjoint flow computation to compute the quantities of interest and their gradients, respectively.

The instantaneous quantities of interest for the nominal motion and solution of (1.50) are included in Fig. 1.10. It is clear that the optimal motion requires more work than the nominal motion to overcome the induced drag on the wing and satisfy the thrust constraint.

Fig. 1.11 shows the convergence of the integrated quantities of interest with iterations in the optimization solver. It can be seen that, initially, the thrust constraint is violated and after only 2 optimization iterations, the flapping motion has become sufficient to overcome the induced drag and satisfy the thrust constraint, at the cost of additional energy that must be input to the system. After 10 iterations, the thrust constraint is satisfied and reduction of the work has essentially ceased. At the optimal solution, the thrust constraint is *active* and satisfied to 9 digits of accuracy.

The trajectory of the wing and isosurfaces of the surrounding flow are shown in Fig. 1.12 (nominal) and Fig. 1.13 (optimal). The flow around the nominal trajectory is fairly benign in that there is little flow separation, does not require

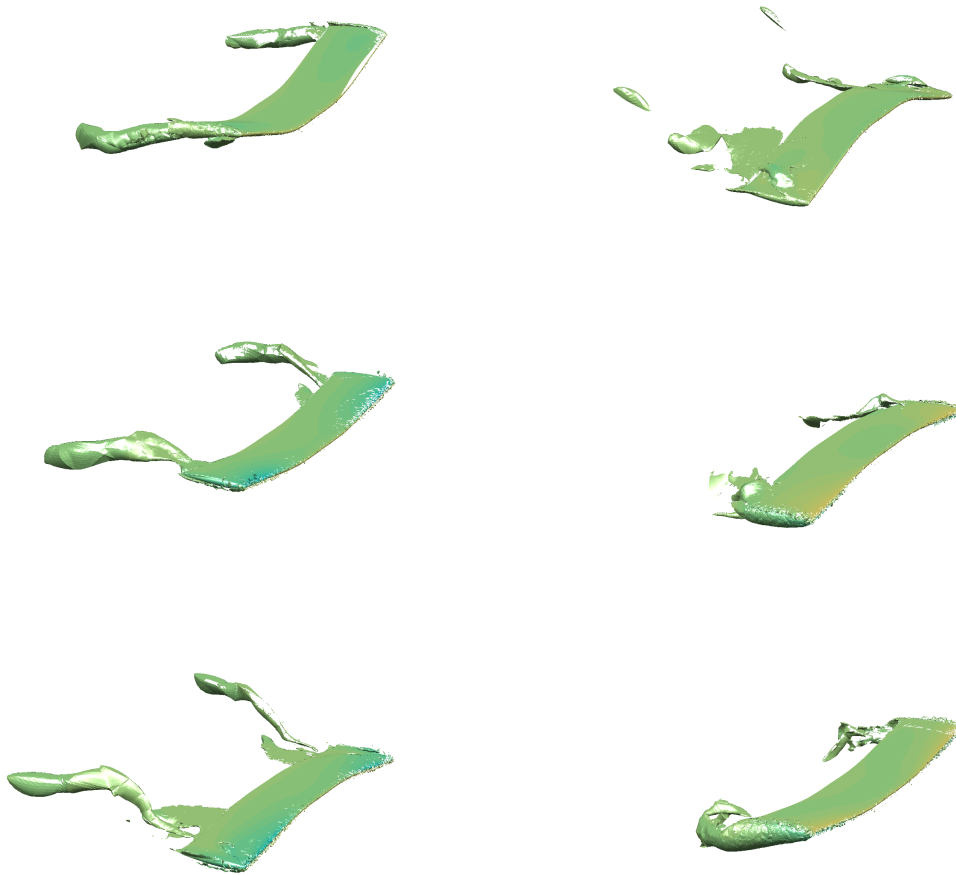


Fig. 1.12: Visualization of the flow field around wing with flapping motion corresponding to the *initial guess* for the optimization problem in (1.50). The color shows the pressure field on the wing surface as well as on an isosurface of the streamwise vorticity. Snapshots (top to bottom, left to right) taken at equally spaced time increments during the second period that correspond to times  $t = 5.0, 5.83, 6.67, 7.5, 8.33, 9.17$ .

much energy, and the generated thrust is not sufficient to overcome the induced drag. In contrast, the optimal trajectory flaps “harder” (larger flapping and pitching amplitudes) in order to generate sufficient thrust to satisfy the constraint. The result is more separation, even though the additional pitching helps streamline the flow, and more required energy.

## 1.6 Conclusion

This work presents a framework for using high-order numerical discretizations to solve optimization problems constrained by deforming domain conservation laws and demonstrates its potential on the large-scale application of determining energetically optimal flapping motions of a three-dimensional wing. The high-order numerical method employs a discontinuous Galerkin spatial discretization and diagonally implicit Runge-Kutta temporal discretization for both the ALE-transformed conservation law and its quantities of interest. The fully discrete adjoint method was used to



Fig. 1.13: Visualization of the flow field around wing with flapping motion corresponding to the *solution* of the optimization problem in (1.50). The color shows the pressure field on the wing surface as well as on an isosurface of the streamwise vorticity. Snapshots (top to bottom, left to right) taken at equally spaced time increments during the second period that correspond to times  $t = 5.0, 5.83, 6.67, 7.5, 8.33, 9.17$ .

compute gradients of quantities of interest to ensure they are discretely consistent and the cost of computing them only scales weakly with the number of parameters. This framework only required 12 iterations when coupled with the nonlinear optimizer IPOPT to solve the relevant problem of finding a thrust-neutral flapping trajectory that minimizes the energy required to complete the motion.

The framework presented is sufficiently general to handle a number of relevant generalizations such as shape optimization of the wing cross-section and planform, more general spline-based parametrizations, and the inclusion of other aerodynamic constraints. The ALE framework is capable of handling completely general domain deformations, which includes static changes to the shape of the wing in a shape-only or combined shape and trajectory optimization setting. A more general parametrization can also easily be included by using a spline-based parametrization of the flapping angles in (1.48) and the expanded design space would likely lead to better designs. Finally, other aerodynamic constraints can easily be incorporated in the optimization problem in (1.50) at the cost of an additional adjoint solve for each additional constraint (that depends on the PDE solution).

While this work is one step toward solving optimization problems of engineering and scientific relevance, further development is required to have an impact in practice. This work has considered a pure fluid problem and treats the structure as rigid, which is not realistic, particularly in the regime of MAVs. Additionally, as noted in [43, 22, 65, 48, 42, 47], more efficient flapping motions may be realized from a flexible structure. As such, extending the high-order discretization and corresponding adjoint method to fluid-structure interaction problems or, more generally, multiphysics problems coupled along an interface, will be the subject of future work. Furthermore, for larger scale applications the cost of repeatedly solving the conservation law becomes a challenge and calls for more efficient solvers such as those developed in [36] or a globally convergent optimization framework that incorporates fast and reliable adaptive reduced-order models [60, 59].

## Acknowledgments

This work was supported in part by the Luis Alvarez Postdoctoral Fellowship by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 (MZ), and by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under contract number DE-AC02-05CH11231 (PP). The content of this publication does not necessarily reflect the position or policy of any of these supporters, and no official endorsement should be inferred.

## References

1. Roger Alexander. Diagonally implicit Runge-Kutta methods for stiff ODEs. *SIAM J. Numer. Anal.*, 14(6):1006–1021, 1977.
2. Yves Alleneau, Matthew Culbreth, and Antony Jameson. A computational framework for low Reynolds number 3d flapping wings simulations. In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 27 – 30 2011.
3. Douglas N Arnold, Franco Brezzi, Bernardo Cockburn, and L Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.
4. Sai K Banala and Sunil K Agrawal. Design and optimization of a mechanism for out-of-plane insect winglike motion with twist. *Journal of Mechanical Design*, 127(4):841–844, 2005.
5. James M Birch, William B Dickson, and Michael H Dickinson. Force production and flow structure of the leading edge vortex on flapping wings at high and low Reynolds numbers. *Journal of Experimental Biology*, 207(7):1063–1072, 2004.
6. Frank M Bos, Bas W van Oudheusden, and Hester Bijl. Wing performance and 3-d vortical structure formation in flapping flight. *Journal of Fluids and Structures*, 42:130–151, 2013.
7. Chris Chabalko, Richard D Snyder, Philip S Beran, and Gregory Parker. The physics of an optimized flapping wing micro air vehicle. In *Proceedings of the 47th AIAA Aerospace Science Meeting Including The New Horizons Forum and Aerospace Exposition*, AIAA Paper, number 2009-801, Orlando, Florida, January 5 – 8 2009.
8. Tony F Chan and Wing Lok Wan. Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 18(6):1698–1721, 1997.
9. Jung-Sun Choi, Liangyu Zhao, Gyung-Jin Park, Sunil K Agrawal, and Raymond K Kolonay. Enhancement of a flapping wing using path and dynamic topology optimization. *AIAA Journal*, 49(12):2616–2626, December 2011.
10. Bernardo Cockburn and Chi-Wang Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001.
11. Matthew Culbreth, Yves Alleneau, and Antony Jameson. High-fidelity optimization of flapping airfoils and wings. In *29th AIAA Applied Aerodynamics Conference*, Honolulu, Hawaii, June 27 – 30 2011.
12. Benoît Desjardins, Emmanuel Grenier, P-L Lions, and Nader Masmoudi. Incompressible limit for solutions of the isentropic Navier–Stokes equations with Dirichlet boundary conditions. *Journal de Mathématiques Pures et Appliquées*, 78(5):461–471, 1999.
13. Michael H Dickinson, Fritz-Olaf Lehmann, and Sanjay P Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422):1954–1960, 1999.
14. Charbel Farhat, Christoph Degand, Bruno Koobus, and Michel Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer methods in applied mechanics and engineering*, 163(1):231–245, 1998.
15. Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.
16. Bradley Froehle and Per-Olof Persson. Nonlinear elasticity for mesh deformation with high-order discontinuous galerkin methods for the navier-stokes equations on deforming domains. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pages 73–85. Springer, 2015.

17. Bradley Michael Froehle. *High-order discontinuous Galerkin fluid-structure interaction methods*. University of California, Berkeley, 2013.
18. Ryan B George, Mark B Colton, Christopher A Mattson, and Scott L Thomson. A differentially driven flapping wing mechanism for force analysis and trajectory optimization. *International Journal of Micro Air Vehicles*, 4(1):31–49, 2012.
19. Mehdi Ghommem, Nathan Collier, Antti H Niemi, and Victor M Calo. On the shape optimization of flapping wings and their performance analysis. *Aerospace Science and Technology*, 32(1):274–292, 2014.
20. Mehdi Ghommem, Muhammad R Hajj, Dean T Mook, Bret K Stanford, Philip S Beran, Richard D Snyder, and Layne T Watson. Global optimization of actively morphing flapping wings. *Journal of Fluids and Structures*, 33:210–228, 2012.
21. Peter A Gnoffo et al. CFD validation studies for hypersonic flow prediction. *AIAA paper*, 1025:2001, 2001.
22. Peter G Ifju, David A Jenkins, Scott Ettinger, Yongsheng Lian, Wei Shyy, and Martin R Waszak. Flexible-wing-based micro air vehicles. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, number 2002-0705, pages 1–5, Reno, Nevada, January 14 – 17 2002.
23. Antony Jameson, Wolfgang Schmidt, Eli Turkel, et al. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA paper*, 1259:1981, 1981.
24. KD Jones and MF Platzer. Numerical computation of flapping-wing propulsion and power extraction. In *35th Aerospace Sciences Meeting and Exhibit*, volume 97, page 0826, Reno, Nevada, January 6 – 9 1997. AIAA.
25. Martin Jones and Nail K Yamaleev. Adjoint based shape and kinematics optimization of flapping wing propulsive efficiency. 43rd AIAA Fluid Dynamics Conference. San Diego, CA, 2013. AIAA 2013-2472, 2013.
26. Bruno Koobus and Charbel Farhat. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170(1):103–129, 1999.
27. Fritz-Olaf Lehmann. The mechanisms of lift enhancement in insect flight. *Naturwissenschaften*, 91(3):101–122, 2004.
28. Michel Lesoinne and Charbel Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer methods in applied mechanics and engineering*, 134(1):71–90, 1996.
29. Chi-Kun Lin. On the incompressible limit of the compressible Navier-Stokes equations. *Communications in partial differential equations*, 20(3-4):677–707, 1995.
30. Karthik Mani and Dimitri J Mavriplis. Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal*, 46(6):1351–1364, 2015/06/22 2008.
31. V Maple. Waterloo maple software. *University of Waterloo, Version*, 5, 1994.
32. James M. McMichael and Michael S. Francis. Micro air vehicles - towards a new dimension in flight. Technical report, DARPA, August 7 1997.
33. Siva K. Nadarajah and Antony Jameson. Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint method. *AIAA Journal*, 45(7):1478–1491, 2007.
34. Eric J Nielsen, Boris Diskin, and Nail K Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA Journal*, 48(6):1195–1206, 2010.
35. Kui Ou, Patrice Castonguay, and Antony Jameson. 3d flapping wing simulation with high order spectral difference method on deformable mesh. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, volume 1316, page 2011, Orlando, Florida, January 4 – 7 2011.
36. Will Pazner and Per-Olof Persson. Stage-parallel fully implicit Runge-Kutta solvers for discontinuous Galerkin fluid simulations. *Journal of Computational Physics*, 2016.
37. Jaime Peraire and Per-Olof Persson. The Compact Discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.
38. Per-Olof Persson, Javier Bonet, and Jaime Peraire. Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1585–1595, 2009.
39. Per-Olof Persson and Jaime Peraire. Curved mesh generation and mesh refinement using Lagrangian solid mechanics. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit*, volume 204, 2009.
40. Ravi Ramamurti and William Sandberg. Simulation of flow about flapping airfoils using finite element incompressible flow solver. *AIAA Journal*, 39(2):253–260, 2001.
41. Philip L Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
42. Wei Shyy, Hikaru Aono, Satish Kumar Chimakurthi, P Trizila, C-K Kang, Carlos ES Cesnik, and Hao Liu. Recent progress in flapping wing aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 46(7):284–327, 2010.
43. Wei Shyy, Mats Berg, and Daniel Ljungqvist. Flapping and flexible wings for biological and micro air vehicles. *Progress in aerospace sciences*, 35(5):455–505, 1999.
44. Wei Shyy, Y Lian, J Tang, H Liu, P Trizila, B Stanford, L Bernal, C Cesnik, P Friedmann, and P Ifju. Computational aerodynamics of low Reynolds number plunging, pitching and flexible wings for mav applications. *Acta Mechanica Sinica*, 24(4):351–373, 2008.
45. Valeria Simoncini and Efstratios Gallopoulos. An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM Journal on Scientific Computing*, 16(4):917–933, 1995.
46. Bret K Stanford and Philip S Beran. Analytical sensitivity analysis of an unsteady vortex-lattice method for flapping-wing optimization. *Journal of Aircraft*, 47(2):647–662, 2010.
47. WB Tay, BW Van Oudheusden, and H Bijl. Numerical simulation of X-wing type biplane flapping wings in 3D using the immersed boundary method. *Bioinspiration & Biomimetics*, 9(3), March 14 2014.
48. Alexandra H Techet. Propulsive performance of biologically inspired flapping foils at high Reynolds numbers. *Journal of Experimental Biology*, 211(2):274–279, 2008.

49. Ismail H Tuncer and Mustafa Kaya. Optimization of flapping airfoils for maximum thrust and propulsive efficiency. *AIAA Journal*, 43(11):2329–2336, 2005.
50. James R Usherwood and Charles P Ellington. The aerodynamics of revolving wings ii. propeller force coefficients from mayfly to quail. *Journal of Experimental Biology*, 205(11):1565–1576, 2002.
51. Marnix P van Schrojenstein Lantman and K Fidkowski. Adjoint-based optimization of flapping kinematics in viscous flows. In *21st AIAA Computational Fluid Dynamics Conference*, 2013.
52. Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
53. Jingyi Wang, Matthew J Zahr, and Per-Olof Persson. Energetically optimal flapping flight via a fully discrete adjoint method with explicit treatment of flapping frequency. In *23rd AIAA Computational Fluid Dynamics Conference*, page 4412, 2017.
54. ZJ Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, HT Huynh, et al. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.
55. David J Willis, Emily R Israeli, Per-Olof Persson, Mark Drela, Jaime Peraire, SM Swartz, and Kenneth S Breuer. A computational framework for fluid structure interaction in biologically inspired flapping flight. In *25th AIAA Applied Aerodynamics Conference*, volume 1, pages 38–59, Miami, Florida, June 25 – 28 2007.
56. David J Willis, Per-Olof Persson, Emily R Israeli, Jaime Peraire, Sharon M Swartz, and Kenneth S Breuer. Multifidelity approaches for the computational analysis and design of effective flapping wing vehicles. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, page 2008, Reno, Nevada, January 7 – 10 2008.
57. Nail Yamaleev, Boris Diskin, and Eric Nielsen. Adjoint-based methodology for time-dependent optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2008.
58. Zhi Yang and Dimitri J Mavriplis. Unstructured dynamic meshes with higher-order time integration schemes for the unsteady navier-stokes equations. *AIAA paper*, 1222(2005):1, 2005.
59. Matthew J. Zahr. *Adaptive model reduction to accelerate optimization problems governed by partial differential equations*. PhD thesis, Stanford University, August 2016.
60. Matthew J Zahr and Charbel Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *International Journal for Numerical Methods in Engineering*, 102(5):1111–1135, 2015.
61. Matthew J. Zahr and Per-Olof Persson. Performance tuning of Newton-GMRES methods for discontinuous Galerkin discretizations of the Navier-Stokes equations. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2013.
62. Matthew J Zahr and Per-Olof Persson. High-order, time-dependent aerodynamic optimization using a discontinuous Galerkin discretization of the Navier-Stokes equations. In *AIAA Science and Technology Forum and Exposition*, San Diego, CA, 2016.
63. Matthew J. Zahr and Per-Olof Persson. An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems. *Journal of Computational Physics*, In review, 2016.
64. Matthew J Zahr, Per-Olof Persson, and Jon Wilkening. A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints. *Computers & Fluids*, Special Issue on USNCCM13 International Symposium on Spectral and High-Order Methods, 2016.
65. Qiang Zhu. Numerical simulation of a flapping foil with chordwise or spanwise flexibility. *AIAA Journal*, 45(10):2448–2457, October 2007.