A high-order partitioned solver for general multiphysics problems and its applications in optimization

Daniel Z. Huang*
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, U.S.A.

Per-Olof Persson†
Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, U.S.A.

Matthew J. Zahr‡
Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, U.S.A.

A high-order accurate adjoint-based optimization framework is presented for unsteady multiphysics problems. The fully discrete adjoint solver relies on the high-order, linearly stable, partitioned solver introduced in [1], where different subsystems are modeled and discretized separately. The coupled system of semi-discretized ordinary differential equations is taken as a monolithic system and partitioned using an implicit-explicit Runge-Kutta (IMEX-RK) discretization [2]. Quantities of interest (QoI) that take the form of space-time integrals are discretized in a solver-consistent manner. The corresponding adjoint equations are derived to compute exact gradients of QoI, which can be solved in a partitioned manner, i.e. subsystem-by-subsystem and substage-by-substage, thanks to the partitioned primal solver. These quantities of interest and their gradients are then used in the context of gradient-based PDE-constrained optimization. The present optimization framework is applied to two fluid-structure interaction problems: 1D piston problem with a three-field formulation and a 2D energy harvesting problem with a two-field formulation.

I. Introduction

Optimization problems involving multiphysics systems commonly arise in engineering practice, particularly in the context of design or control of physics-based systems. These problems lead to PDE-constrained optimizations. In the literature, a majority of research in PDE-constrained optimization has been focused on a single physical system or steady PDEs, which is sufficient for a large class of problems of interest. However, there is a large class of problems where such analysis is insufficient, such as problems that involves the interactions of multiple physical systems or physical phenomena, which are generally inherently dynamic. Typical examples include flapping flight for Micro-Aerial Vehicles (MAVs) designs [3, 4], optimal combustion control system to maintain stable combustion with low exhaust emissions [5, 6], microscale swimmer designs for drug delivery [7], and wind turbine performance optimization [8, 9] to extract maximum energy. Design and control of these types of systems are challenging considering the coupling effects of multiple physics and the high computational cost due to their unsteady nature. Innovative multiphysics solvers and state-of-art optimization tools are needed to solve such problems.

We first review the high-order, linearly stable, implicit-explicit Runge-Kutta (IMEX-RK) [2] based partitioned solvers for multiphysics problems proposed in [1]. In this framework, a generic multiphysics problem is modeled as a system of n systems of partial differential equations where the ith subsystem is coupled to the other subsystems through a coupling term that can depend on the state of all the other subsystems. This coupled system of partial differential equations reduces to a coupled system of ordinary differential equations via the method of lines where an appropriate

*Institute for Computational and Mathematical Engineering, Durand Building Room 226, Stanford University, Stanford, CA 94305-3035. Email: zhengyuh@stanford.edu. AIAA Student Member.

†Associate Professor, Department of Mathematics, University of California, Berkeley, Berkeley CA 94720-3840. Email: persson@berkeley.edu. AIAA Senior Member.

‡Assistant Professor, Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556-5637. Email: mzahr@nd.edu. AIAA Member.

spatial discretization is applied to each subsystem. The coupled system of ordinary differential equations is taken as a monolithic system and discretized using an IMEX-RK discretization with a specific implicit-explicit decomposition that introduces the concept of a *predictor* for the coupling term. Four coupling predictors are proposed that enable the monolithic system to be solved in a partitioned manner, i.e., subsystem-by-subsystem, and preserve the IMEX-RK structure and therefore the design order of accuracy of the monolithic scheme. The four partitioned solvers that result from these predictors are high-order accurate, allow for maximum re-use of existing single-physics software, and two of the four solvers allow the subsystems to be solved in parallel at a given stage and time step. In [1], we also analyze the stability of a coupled, linear model problem and show that one of the partitioned solvers achieves unconditional linear stability, while the others are unconditionally stable only for certain values of the coupling strength.

Next, we derive the corresponding fully discrete sensitivity and adjoint equations for general optimization problems. Here, we mainly focus on one of the aforementioned partitioned solvers, the weakly coupled Gauss-Seidel predictor based partitioned solver, which has demonstrated its high-order accuracy, numerical stability and software maintainability in many engineering problems. Quantities of interest or objective functions, e.g. energy consumption or the quantities of combustion emission, that take the form of space-time integrals that are discretized in a solver-consistent manner. This ensures the discretization order of quantities of interest exactly matches the PDE temporal discretization. The aforementioned multiphysics partitioned solver becomes the PDE-constraint of the optimization problem. To compute exact gradients of quantities of interest, we need to solve the multiphysics problem, and then either compute the sensitivity of the state variables through forward time-marching or evaluate the adjoint variables through backward time-marching. We can leverage the high-order linear stability property of the partitioned solver, which takes large time steps and therefore reduces the number of time steps and accelerates the time-marching procedure. The optimization solver IPOPT [10] is used to solve the optimization problem based on a nonlinearly constrained interior point method.

The remainder of this paper is organized as follows. In Section II, the governing equations of the multiphysics system, the integral form quantities of interest and their semi-discretizations are introduced. The high-order temporal discretization, based on IMEX-RK schemes, is described in Section III, which leads to a partitioned multiphysics solver. Following this in Section IV, the corresponding fully discrete sensitivity equations and adjoint equations are derived, which deliver the *exact* gradient of the QoIs. Section V demonstrates the approach as applied to two optimization problems: a 1D oscillating piston problem and a 2D airfoil energy harvesting problem. Finally, conclusions are offered in Section VI.

II. Governing multiphysics equations and semi-discretization

Consider a general formulation of a mathematical model describing the behavior of multiple interacting physical phenomena described by the following coupled system of partial differential equations

$$\partial_t u^i = \mathcal{L}^i(u^i,\, c^i,\, x,\, \boldsymbol{\mu},\, t), \quad x \in \Omega^i(c^i,\, \boldsymbol{\mu},\, t), \quad t \in (0,\, T) \tag{1}$$

for $i = 1,\, \dots,\, m$, where m represents the number of physical systems, and boundary conditions are excluded for brevity. The ith physical system is modeled as a partial differential equation characterized by the generalized differential operator $\mathcal{L}^i$ that defines a conservation law or other type of balance law, the state variable u^i that is the solution of the ith physical system on the space-time domain $\Omega^i \times (0,\, T)$, and a *coupling term* c^i that, in general, couples the ith system to the other $m - 1$ systems. In the general case, the differential operator $\mathcal{L}^i$, domain Ω^i, and boundary conditions depend on the coupling term. The coupling term contains quantities usually considered *data* required to define the ith PDE, such as boundary conditions or material properties. In a single-physics setting, these quantities would be prescribed, but in the multiphysics setting they are determined from the state vectors of all m systems, i.e.,

$$c^i = c^i(u^1,\, \dots,\, u^m,\, x,\, \boldsymbol{\mu},\, t). \tag{2}$$

The definition of the coupling term is problem-dependent and special structure in the coupling term can be exploited to create a better partitioned solver. While the form of (1) is specific to first-order temporal systems, it includes equations with higher-order temporal derivatives, assuming they have been re-cast in the first-order form. The spatial domains Ω^i for the individual systems may or may not be overlapping and in many cases are the same, i.e., $\Omega^i = \Omega$ for $i = 1,\, \dots,\, m$. Any of the operators, solution variables, and even the deformed computational domain might depend on the parameter vector $\boldsymbol{\mu}$. The quantities of interest are assumed to be of the integral form,

$$\mathcal{J}(u^1, \ldots, u^m, x, \boldsymbol{\mu}, T) = \int_0^T j(u^1(\tau), \ldots, u^m(\tau), \boldsymbol{\mu}, \tau) d\tau. \tag{3}$$

We introduce the semi-discrete form of the coupled partial differential equations in (1) that arises from applying an appropriate spatial discretization to the ith PDE system individually, which takes the form

$$\boldsymbol{M}^i \dot{\boldsymbol{u}}^i = \boldsymbol{r}^i(\boldsymbol{u}^i, \boldsymbol{c}^i, \boldsymbol{\mu}, t), \quad t \in (0, T) \tag{4}$$

where $\boldsymbol{u}^i(t)$ is the semi-discrete state vector corresponding to the spatial discretization of $u^i(x, t)$, $\boldsymbol{r}^i$ is the spatial discretization of the differential operator $\mathcal{L}^i$ and called the velocity of the ODE system in the remainder of the document, and $\boldsymbol{c}^i$ is the semi-discrete coupling term corresponding to the spatial discretization of $c^i(u^1, \ldots, u^m, x, t)$. In general, the coupling term depends on the semi-discrete state vector of all m systems

$$\boldsymbol{c}^i = \boldsymbol{c}^i(\boldsymbol{u}^1, \ldots, \boldsymbol{u}^m, \boldsymbol{\mu}, t). \tag{5}$$

For convenience, we re-write the system of ordinary differential equations in (4)-(5) as

$$\boldsymbol{M}\dot{\boldsymbol{u}} = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{c}(\boldsymbol{u}, \boldsymbol{\mu}, t), \boldsymbol{\mu}, t), \quad t \in (0, T), \tag{6}$$

where the combined mass matrix is a block diagonal matrix consisting of the single-physics mass matrices

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^1 & & \\ & \ddots & \\ & & \boldsymbol{M}^m \end{bmatrix}$$

and the combined state vector, coupling term, and nonlinear residual are vectors consisting of the corresponding single-physics term, concatenated across all m systems

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}^1 \\ \vdots \\ \boldsymbol{u}^m \end{bmatrix} \qquad \boldsymbol{c}(\boldsymbol{u}, t) = \begin{bmatrix} \boldsymbol{c}^1(\boldsymbol{u}^1, \ldots, \boldsymbol{u}^m, \boldsymbol{\mu}, t) \\ \vdots \\ \boldsymbol{c}^m(\boldsymbol{u}^1, \ldots, \boldsymbol{u}^m, \boldsymbol{\mu}, t) \end{bmatrix} \qquad \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{c}, t) = \begin{bmatrix} \boldsymbol{r}^1(\boldsymbol{u}^1, \boldsymbol{c}^1, \boldsymbol{\mu}, t) \\ \vdots \\ \boldsymbol{r}^m(\boldsymbol{u}^m, \boldsymbol{c}^m, \boldsymbol{\mu}, t) \end{bmatrix}.$$

The total derivative, or Jacobian, of the semi-discrete velocity $D_{\boldsymbol{u}}\boldsymbol{r}$ is expanded as

$$D_{\boldsymbol{u}}\boldsymbol{r} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{c}}\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}, \tag{7}$$

where the individual terms take the form

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial \boldsymbol{r}^1}{\partial \boldsymbol{u}^1} & & \\ & \ddots & \\ & & \frac{\partial \boldsymbol{r}^m}{\partial \boldsymbol{u}^m} \end{bmatrix} \qquad \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{c}} = \begin{bmatrix} \frac{\partial \boldsymbol{r}^1}{\partial \boldsymbol{c}^1} & & \\ & \ddots & \\ & & \frac{\partial \boldsymbol{r}^m}{\partial \boldsymbol{c}^m} \end{bmatrix} \qquad \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial \boldsymbol{c}^1}{\partial \boldsymbol{u}^1} & \cdots & \frac{\partial \boldsymbol{c}^1}{\partial \boldsymbol{u}^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{c}^m}{\partial \boldsymbol{u}^1} & \cdots & \frac{\partial \boldsymbol{c}^m}{\partial \boldsymbol{u}^m} \end{bmatrix}, \tag{8}$$

and the dependencies have been dropped for brevity. The first term in the Jacobian, Eq. (7), is block diagonal and accounts for the direct contribution of a state to its own system while the second term accounts for the coupling between systems.

III. A high-order partitioned solver for multiphysics problems

In this section, high-order partitioned time-integration schemes for multiphysics systems are introduced. In a partitioned sense, individual off-the-shelf single-physics solvers are combined to solve the multiphysics problem, rather than considering the monolithic multiphysics system. However, they tend to be limited to low-order accuracy and have stringent stability requirements. Our partitioned time-integration scheme mitigates most of these issues by combining high-order implicit-explicit Runge-Kutta (IMEX) schemes for the monolithic multiphysics system with a judicious implicit-explicit decomposition that *diagonally* couples the individual systems via a novel predictor for the coupling terms.

A. Background: implicit-explicit Runge-Kutta schemes

Implicit-explicit Runge-Kutta schemes, first proposed in [2, 11], define a family of high-order discretizations for nonlinear differential equations whose velocity term can be decomposed into a sum of a non-stiff $\boldsymbol{f}$ and stiff $\boldsymbol{g}$ velocity

$$\boldsymbol{M}\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t) + \boldsymbol{g}(\boldsymbol{u}, t). \tag{9}$$

The non-stiff $\boldsymbol{f}$ velocity is integrated with an s-stage explicit Runge-Kutta scheme and the stiff term $\boldsymbol{g}$ is integrated with an s-stage diagonally implicit Runge-Kutta scheme. IMEX Runge-Kutta schemes are compactly represented by a double tableau in the usual Butcher notation (Table 1), where $\hat{A}$, $\hat{b}$, $\hat{c}$ defines the Butcher tableau for the explicit Runge-Kutta scheme used for $\boldsymbol{f}$ and A, b, c defines the diagonally implicit Runge-Kutta scheme used for $\boldsymbol{g}$. In this work, we mainly consider IMEX-RK schemes proposed in [12], in which the implicit Runge-Kutta part of these IMEX schemes are L-stable, stiffly-accurate, and have an explicit first stage ($a_{11} = 0$).

Explicit Runge-Kutta coefficients

$$\begin{array}{c|ccccc}
0 & & & & & \\
\hat{c}_2 & \hat{a}_{21} & & & & \\
\hat{c}_3 & \hat{a}_{31} & \hat{a}_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
\hat{c}_s & \hat{a}_{s1} & a_{s2} & \cdots & \hat{a}_{ss-1} & \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_{s-1} & \hat{b}_s
\end{array}$$

Implicit Runge-Kutta coefficients

$$\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & a_{22} & & & \\
c_3 & a_{31} & a_{32} & a_{33} & & \\
\vdots & \vdots & & & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss-1} & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Table 1 Butcher Tableaux for an s-stage implicit-explicit Runge-Kutta scheme.

Consider a discretization of the time domain $[0, T]$ into N_t segments with endpoints $\{t_0, \ldots, t_{N_t}\}$, with the nth segment having length $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \ldots, N_t$. Also, let $\boldsymbol{u}_n$ denote the approximation of the solution of the differential equation in (9) at timestep n, i.e., $\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$. Then, given the explicit $(\hat{A}, \hat{b}, \hat{c})$ and implicit (A, b, c) Butcher tableaux, the s-stage IMEX Runge-Kutta scheme that advances $\boldsymbol{u}_{n-1}$ to $\boldsymbol{u}_n$ is given by

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{p=1}^{s} \hat{b}_p \hat{\boldsymbol{k}}_{n,p} + \sum_{p=1}^{s} b_p \boldsymbol{k}_{n,p}, \tag{10a}$$

$$\boldsymbol{M}\boldsymbol{k}_{n,j} = \Delta t_n \boldsymbol{g}(\boldsymbol{u}_{n,j}, t_{n-1} + c_j \Delta t_n), \tag{10b}$$

$$\boldsymbol{M}\hat{\boldsymbol{k}}_{n,j} = \Delta t_n \boldsymbol{f}(\boldsymbol{u}_{n,j}, t_{n-1} + \hat{c}_j \Delta t_n), \tag{10c}$$

$$\boldsymbol{u}_{n,j} = \boldsymbol{u}_{n-1} + \sum_{p=1}^{j-1} \hat{a}_{jp} \hat{\boldsymbol{k}}_{n,p} + \sum_{p=1}^{j} a_{jp} \boldsymbol{k}_{n,p}, \tag{10d}$$

where $\hat{\boldsymbol{k}}_{n,p}$ and $\boldsymbol{k}_{n,p}$ are the pth explicit and implicit velocity stage, respectively, corresponding to timestep n and $\boldsymbol{u}_{n,p}$ is the approximation to $\boldsymbol{u}_n$ at stage p of timestep n. For each stage j, the nonlinear system of equations in (10b) must be solved to compute the implicit stage $\boldsymbol{k}_{n,j}$. Next, the explicit stage can be computed directly from (10c) since the stage approximation $\boldsymbol{u}_{n,j}$ does not depend on the explicit stage $\hat{\boldsymbol{k}}_{n,j}$. Finally, given the previous timestep and all implicit and explicit stages, the solution at time n is determined from (10a).

B. A partitioned implicit-explicit Runge-Kutta scheme for multiphysics systems

The proposed high-order partitioned scheme for integration of generic time-dependent multiphysics problems of the form (4)-(5) is built on an IMEX Runge-Kutta discretization of the monolithic system. A special choice of implicit-explicit decomposition, along with the introduction of predictors for the coupling term, creates a *diagonal* or *triangular* dependency between the systems and allows the monolithic discretization to be solved in a partitioned manner. The proposed decomposition handles a majority of the relevant physics *implicitly* to leverage the enhanced stability properties of such schemes, while only the correction to the coupling predictor is handled explicitly.

1. Implicit-explicit decomposition and monolithic IMEX Runge-Kutta discretization

To begin our construction, recall the semi-discrete form of the multiphysics system (6) and consider the splitting of the velocity term $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{c}(\boldsymbol{u}, t), t)$ as

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{c}(\boldsymbol{u}, \boldsymbol{\mu}, t), \boldsymbol{\mu}, t) = \boldsymbol{f}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) + \boldsymbol{g}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) \tag{11}$$

where $\tilde{\boldsymbol{c}}$ is an approximation, or *predictor*, of the coupling term $\boldsymbol{c}(\boldsymbol{u}, t)$ and the terms that will be handled explicitly $\boldsymbol{f}$ and implicitly $\boldsymbol{g}$ in the IMEX discretization are defined as

$$\boldsymbol{f}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{c}(\boldsymbol{u}, \boldsymbol{\mu}, t), \boldsymbol{\mu}, t) - \boldsymbol{r}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) \tag{12a}$$
$$\boldsymbol{g}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) = \boldsymbol{r}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t), \tag{12b}$$

where the dependence on the predictor is explicitly included. In general, the predictor depends on the instantaneous state vector $\boldsymbol{u}(t)$ and data $\bar{\boldsymbol{u}}$, likely from the history of the state vector $\{\boldsymbol{u}(\tau) \mid \tau < t\}$

$$\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{c}}(\boldsymbol{u}, \bar{\boldsymbol{u}}, \boldsymbol{\mu}, t). \tag{13}$$

With this decomposition of the velocity of the semi-discrete multiphysics system in (12), the IMEX Runge-Kutta scheme in (10) applied to the monolithic multiphysics system (6) becomes

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{p=1}^{s} \hat{b}_p \hat{\boldsymbol{k}}_{n,p} + \sum_{p=1}^{s} b_p \boldsymbol{k}_{n,p}, \tag{14a}$$
$$\boldsymbol{M}\boldsymbol{k}_{n,j} = \Delta t_n \boldsymbol{g}(\boldsymbol{u}_{n,j}, \tilde{\boldsymbol{c}}(\boldsymbol{u}_{n,j}, \boldsymbol{u}_{n-1}, \boldsymbol{\mu}, t_{n,j}), \boldsymbol{\mu}, t_{n,j}), \tag{14b}$$
$$\boldsymbol{M}\hat{\boldsymbol{k}}_{n,j} = \Delta t_n \boldsymbol{f}(\boldsymbol{u}_{n,j}, \tilde{\boldsymbol{c}}(\boldsymbol{u}_{n,j}, \boldsymbol{u}_{n-1}, \boldsymbol{\mu}, t_{n,j}), \boldsymbol{\mu}, t_{n,j}), \tag{14c}$$
$$\boldsymbol{u}_{n,j} = \boldsymbol{u}_{n-1} + \sum_{p=1}^{j-1} \hat{a}_{jp} \hat{\boldsymbol{k}}_{n,p} + \sum_{p=1}^{j} a_{jp} \boldsymbol{k}_{n,p}, \tag{14d}$$

where the data used in the coupling predictor is taken from the previous timestep. This is the general form of the fully discrete, monolithic multiphysics system where the coupling predictor is unspecified. In the general setting where each coupling predictor depends on the state of all systems, the Jacobian of the coupling predictor is block dense with potentially sparse blocks

$$\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{u}} = \begin{bmatrix} \dfrac{\partial \tilde{\boldsymbol{c}}^1}{\partial \boldsymbol{u}^1} & \cdots & \dfrac{\partial \tilde{\boldsymbol{c}}^1}{\partial \boldsymbol{u}^m} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \tilde{\boldsymbol{c}}^m}{\partial \boldsymbol{u}^1} & \cdots & \dfrac{\partial \tilde{\boldsymbol{c}}^m}{\partial \boldsymbol{u}^m} \end{bmatrix}.$$

This implies the Jacobian of the implicit velocity

$$D_{\boldsymbol{u}}\boldsymbol{g} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} + \frac{\partial \boldsymbol{r}}{\partial \tilde{\boldsymbol{c}}}\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{u}}$$

is also block dense, which highlights the fact that there is coupling across all systems and a monolithic solver is required for the implicit step.

2. Weakly coupled Gauss-Seidel predictor

The Gauss-Seidel-type (triangular) predictors for the multiphysics system assume the individual systems are *ordered* in a physically relevant manner. The preferred ordering is problem-dependent. The weakly coupled Gauss-Seidel-type predictor for the ith system is defined as

$$\tilde{\boldsymbol{c}}^i(\boldsymbol{u}, \bar{\boldsymbol{u}}, \boldsymbol{\mu}, t) = \boldsymbol{c}(\boldsymbol{u}^1, \dots, \boldsymbol{u}^{i-1}, \bar{\boldsymbol{u}}^i, \dots, \bar{\boldsymbol{u}}^m, \boldsymbol{\mu}, t) \tag{15}$$

for $i = 1, \dots, m$. At the fully discrete level, this predictor takes the form

$$\tilde{\boldsymbol{c}}^i(\boldsymbol{u}_{n,j}, \boldsymbol{u}_{n-1}, \boldsymbol{\mu}, t) = \boldsymbol{c}(\boldsymbol{u}^1_{n,j}, \dots, \boldsymbol{u}^{i-1}_{n,j}, \boldsymbol{u}^i_{n-1}, \dots, \boldsymbol{u}^m_{n-1}, \boldsymbol{\mu}, t). \tag{16}$$

In the context of the IMEX-RK discretization in (14a-14d), the ith predictor lags the state of systems $i, \dots, m$ to the previous timestep in the evaluation of the coupling term throughout all stages of the timestep. The IMEX-RK discretization of the multiphysics system in (14a-14d) with this form of the predictor leads to Algorithm 1. In this case, the Jacobian of the coupling predictor is block strictly lower triangular

Algorithm 1 Implicit-Explicit Runge-Kutta partitioned multiphysics scheme: weak Gauss-Seidel predictor

```
for stages $j = 1, \dots, s$ do
    for physical systems $i = 1, \dots, m$ do
        Define stage solution according to (14a): $\boldsymbol{u}^i_{n,j} = \boldsymbol{u}^i_{n-1} + \sum_{p=1}^{j-1} \hat{a}_{jp}\hat{\boldsymbol{k}}^i_{n,p} + \sum_{p=1}^{j} a_{jp}\boldsymbol{k}^i_{n,p}$
        Implicit solve (14b) for $\boldsymbol{k}^i_{n,j}$: $\boldsymbol{M}^i\boldsymbol{k}^i_{n,j} = \Delta t_n \boldsymbol{g}^i(\boldsymbol{u}^i_{n,j}, \boldsymbol{c}^i(\boldsymbol{u}^1_{n,j}, \dots, \boldsymbol{u}^{i-1}_{n,j}, \boldsymbol{u}^i_{n-1}, \dots, \boldsymbol{u}^m_{n-1}, \boldsymbol{\mu}, t_{n,j}), \boldsymbol{\mu}, t_{n,j})$
        Explicit solve (14c) for $\hat{\boldsymbol{k}}^i_{n,j}$: $\boldsymbol{M}^i\hat{\boldsymbol{k}}^i_{n,j} = \Delta t_n \boldsymbol{f}^i(\boldsymbol{u}_{n,j}, \boldsymbol{c}^i(\boldsymbol{u}^1_{n,j}, \dots, \boldsymbol{u}^{i-1}_{n,j}, \boldsymbol{u}^i_{n-1}, \dots, \boldsymbol{u}^m_{n-1}, \boldsymbol{\mu}, t_{n,j}), \boldsymbol{\mu}, t_{n,j})$
    end for
end for
Set $\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{p=1}^{s} \hat{b}_p\hat{\boldsymbol{k}}_{n,p} + \sum_{p=1}^{s} b_p\boldsymbol{k}_{n,p}$
```

$$\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{u}} = \begin{bmatrix} 0 & & & \\ \dfrac{\partial\boldsymbol{c}^2}{\partial\boldsymbol{u}^1} & 0 & & \\ \vdots & \ddots & \ddots & \\ \dfrac{\partial\boldsymbol{c}^m}{\partial\boldsymbol{u}^1} & \cdots & \dfrac{\partial\boldsymbol{c}^m}{\partial\boldsymbol{u}^{m-1}} & 0 \end{bmatrix},$$

which implies the Jacobian of the monolithic implicit system is block lower triangular

$$D_{\boldsymbol{u}^j}\boldsymbol{g}^i = \begin{cases} \dfrac{\partial\boldsymbol{r}^i}{\partial\boldsymbol{u}^i} & i = j \\ \dfrac{\partial\boldsymbol{r}^i}{\partial\boldsymbol{c}^i}\dfrac{\partial\boldsymbol{c}^i}{\partial\boldsymbol{u}^j} & i > j \\ \boldsymbol{0} & i < j. \end{cases} \tag{17}$$

This block lower triangular nature of the monolithic implicit system implies that the individual systems can be solved sequentially beginning with system 1 and yields a partitioned scheme.

The implicit Jacobian of the monolithic implicit system of the weak Gauss-Seidel predictor (17) involves the entire lower triangular portion of the coupling predictor; however, it is *not required* for the implementation. From inspection of Eq. (14c), the implicit phase at stage j for the ith physical system requires the solution of a nonlinear system of equations in the variable $\boldsymbol{u}_{n,j}^{i}$, with $\boldsymbol{u}_{n,j}^{1}, \ldots, \boldsymbol{u}_{n,j}^{i-1}$ available from the implicit solve corresponding to previous physical systems at the current stage. Therefore, only the *diagonal terms* $\frac{D\boldsymbol{g}^i}{D\boldsymbol{u}^i} = \frac{\partial \boldsymbol{r}^i}{\partial \boldsymbol{u}^i}$ of the monolithic implicit Jacobian are required, which shows that the Jacobians of the coupling terms are not required for the weak Gauss-Seidel predictor. This predictor is guaranteed to preserve the design order of the IMEX-RK discretization and possesses stability properties in practice [1].

IV. Fully discrete sensitivity and adjoint method

In this section, we derive the expression for the total derivative of the quantity of interest $\mathcal{J}$ in Eq. (3) with respect to the parameters $\boldsymbol{\mu}$, which is the essence in gradient-based optimization. Since the evaluation of gradients is often the most costly step in the PDE-constraint optimization cycle, using efficient methods that accurately calculate the gradients are extremely important. There are generally two approaches to provide such information: the direct sensitivity approach and the adjoint approach [13]. When the number of parameters is smaller than the number of quantities of interest, the adjoint approach is much cheaper.

A. Solver-consistent discretization of quantities of interest

To maintain high-order accuracy for the optimization, discretization of the quantity of interest Eq. (3) will be done in a solver-consistent manner [14], i.e. the spatial and temporal discretization used for the governing equation will also be used for the quantities of interest. The integral form Eq. (3) can be rewritten as

$$\frac{\partial \mathcal{J}}{\partial t} = j(\boldsymbol{u}(t), \boldsymbol{\mu}, t). \tag{18}$$

Augmenting the semi-discrete governing equations Eq. (6)(11) with this ODE Eq. (18) yields the system of ODEs

$$\begin{bmatrix} \boldsymbol{M} & \\ & \mathbb{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\mathcal{J}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}(\boldsymbol{u}, \tilde{\boldsymbol{c}}, \boldsymbol{\mu}, t) \\ j(\boldsymbol{u}, \boldsymbol{\mu}, t) \end{bmatrix}. \tag{19}$$

Applying the implicit-explicit temporal discretization introduced in Section III yields the fully discrete governing equations and corresponding solver-consistent discretization of the quantity of interest Eq. (18)

$$\begin{aligned} \mathcal{J}^0 &= 0, \\ \mathcal{J}^n &= \mathcal{J}^{n-1} + \Delta t_n \sum_{p=1}^{s} b_p j(\boldsymbol{u}_{n,p}, \boldsymbol{\mu}, t_{n-1} + c_p \Delta t_n). \end{aligned}$$

Finally, the objective functional in Eq. (3) is evaluated at time $t = T$ to yield the solver-consistent approximation

$$J(\boldsymbol{u}_{1,1}, \ldots, \boldsymbol{u}_{n,p}, \boldsymbol{\mu}) = \mathcal{J}^{N_t} = \sum_{n=1}^{N_t} \Delta t_n \sum_{p=1}^{s} b_p j(\boldsymbol{u}_{n,p}, \boldsymbol{\mu}, t_{n-1} + c_p \Delta t_n). \tag{20}$$

B. Direct sensitivity method

Differentiation of the discretized weakly coupled Gauss-Seidel predictor based partitioned scheme expressions in Alg. 1 with respect to $\boldsymbol{\mu}$ gives rise to the fully discrete sensitivity equations. For the jth stage of the nth timestep, the sensitivity equations of the *i*the subsystem write

$$\frac{\partial \boldsymbol{u}^i_{n,j}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{u}^i_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j-1} \hat{a}_{jp} \frac{\partial \hat{\boldsymbol{k}}^i_{n,p}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j} a_{jp} \frac{\partial \boldsymbol{k}^i_{n,p}}{\partial \boldsymbol{\mu}}, \tag{21a}$$

$$\boldsymbol{M}^i \frac{\partial \boldsymbol{k}^i_{n,j}}{\partial \boldsymbol{\mu}} = \Delta t_n \left(\frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} \frac{\partial \boldsymbol{u}^i_{n,j}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n}} \frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}} \right), \tag{21b}$$

$$\boldsymbol{M}^i \frac{\partial \hat{\boldsymbol{k}}^i_{n,j}}{\partial \boldsymbol{\mu}} = \Delta t_n \left(\frac{\partial \boldsymbol{f}^i_{n,j}}{\partial \boldsymbol{\mu}} + \sum_{k=1}^{m} \frac{\partial \boldsymbol{f}^i_{n,j}}{\partial \boldsymbol{u}^k_{n,j}} \frac{\partial \boldsymbol{u}^k_{n,j}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{f}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n}} \frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}} \right), \tag{21c}$$

$$\frac{\partial \boldsymbol{u}_{n}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{u}_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \hat{b}_{p} \frac{\partial \hat{\boldsymbol{k}}_{n,p}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} b_{p} \frac{\partial \boldsymbol{k}_{n,p}}{\partial \boldsymbol{\mu}}, \tag{21d}$$

here the $\tilde{\boldsymbol{c}}^i_{n,j}$ is the weakly coupled Gaussian Seidel predictor in Eq. (16), and its derivative with respect to $\boldsymbol{\mu}$ is

$$\frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{c}^i_{n,j}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{i-1} \frac{\partial \boldsymbol{c}^i_{n,j}}{\partial \boldsymbol{u}^p_{n,j}} \frac{\partial \boldsymbol{u}^p_{n,j}}{\partial \boldsymbol{\mu}} + \sum_{p=i}^{m} \frac{\partial \boldsymbol{c}^i_{n,j}}{\partial \boldsymbol{u}^p_{n-1}} \frac{\partial \boldsymbol{u}^p_{n-1}}{\partial \boldsymbol{\mu}}. \tag{22}$$

By solving the sensitivities of the stage variables $\frac{\partial \boldsymbol{u}_{n,p}}{\partial \boldsymbol{\mu}}$ from Eq. (21a)-Eq. (21d), the derivative of the quantity of interest, Eq. (20), of the multiphysics problem Eq. (1) is written as

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}} = \sum_{n=1}^{N_t} \Delta t_n \sum_{p=1}^{s} \hat{b}_p \frac{\partial j(\boldsymbol{u}_{n,p}, t_{n-1} + c_p \Delta t_n)}{\partial \boldsymbol{u}_{n,p}} \frac{\partial \boldsymbol{u}_{n,p}}{\partial \boldsymbol{\mu}}. \tag{23}$$

Thanks to the partitioned nature of the multiphysics solver, the sensitivities of the stage variables $\frac{\partial \boldsymbol{u}_{n,p}}{\partial \boldsymbol{\mu}}$ can be solved substep-by-substep and subsystem-by-subsystem, the detailed algorithm is presented in Algorithm 2.

Algorithm 2 Direct sensitivity approach

```
for stages $j = 1, \dots, s$ do
    Read stage solution $\boldsymbol{u}^i_{n-1}, \hat{\boldsymbol{k}}^i_{n,p}, \boldsymbol{k}^i_{n,p}$ for $i = 1, \dots, m$ from disk.
    for physical systems $i = 1, \dots, m$ do
        Construct $\frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}}$ based on (23)
        Implicit solve Eq. (21b) for $\frac{\partial \boldsymbol{k}^i_{n,j}}{\partial \boldsymbol{\mu}}$:
        $\left( \boldsymbol{M}^i - a_{jj} \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} \right) \frac{\partial \boldsymbol{k}^i_{n,j}}{\partial \boldsymbol{\mu}} = \Delta t_n \left( \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} \left( \frac{\partial \boldsymbol{u}^i_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j-1} \hat{a}_{jp} \frac{\partial \hat{\boldsymbol{k}}^i_{n,p}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j-1} a_{jp} \frac{\partial \boldsymbol{k}^i_{n,p}}{\partial \boldsymbol{\mu}} \right) + \frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n}} \frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}} \right)$
        Construct $\frac{\partial \boldsymbol{u}^i_{n,j}}{\partial \boldsymbol{\mu}}$ based on (21a)
    end for
    for physical systems $i = 1, \dots, m$ do
        Explicit solve Eq. (21c) for $\hat{\boldsymbol{k}}^i_{n,j}$: $\frac{\partial \boldsymbol{u}^i_{n,j}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{u}^i_{n-1}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j-1} \hat{a}_{jp} \frac{\partial \hat{\boldsymbol{k}}^i_{n,p}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{j} a_{jp} \frac{\partial \boldsymbol{k}^i_{n,p}}{\partial \boldsymbol{\mu}}$
    end for
end for
Construct $\frac{\partial \boldsymbol{u}_{n}}{\partial \boldsymbol{\mu}}$ based on Eq. (21d)
```

C. Adjoint method

The adjoint method provides an efficient alternative to the direct sensitivity method for evaluating the total derivative of the quantity of interest, especially when the number of parameters is large. Before proceeding to the derivation of the adjoint equations, the following definitions are introduced for the fully discrete Implicit-Explicit Runge-Kutta stage equations and state updates (See Alg. 1)

$$
\begin{aligned}
\tilde{\boldsymbol{r}}^i_0(\boldsymbol{u}_0, \boldsymbol{\mu}) &= \boldsymbol{u}^i_0 - \bar{\boldsymbol{u}}^i(\boldsymbol{\mu}),\\
\boldsymbol{q}^i_{n,j}(\boldsymbol{u}^i_{n,j}, \boldsymbol{u}^i_{n-1}, \hat{\boldsymbol{k}}^i_{n,1}, \ldots, \hat{\boldsymbol{k}}^i_{n,j-1}, \boldsymbol{k}^i_{n,1}, \ldots, \boldsymbol{k}^i_{n,j}) &= \boldsymbol{u}^i_{n,j} - \boldsymbol{u}^i_{n-1} - \sum_{p=1}^{j-1} \hat{a}_{jp}\hat{\boldsymbol{k}}^i_{n,p} - \sum_{p=1}^{j} a_{jp}\boldsymbol{k}^i_{n,p},\\
\boldsymbol{R}^i_{n,j}(\boldsymbol{u}^i_{n,j}, \boldsymbol{k}^i_{n,j}, \boldsymbol{\mu}, \tilde{\boldsymbol{c}}^i_{n,j}) &= \boldsymbol{M}\boldsymbol{k}^i_{n,j} - \Delta t_n \boldsymbol{g}(\boldsymbol{u}^i_{n,j}, \tilde{\boldsymbol{c}}^i_{n,j}, \boldsymbol{\mu}),\\
\hat{\boldsymbol{R}}^i_{n,j}(\boldsymbol{u}^i_{n,j}, \hat{\boldsymbol{k}}^i_{n,j}, \boldsymbol{\mu}, \tilde{\boldsymbol{c}}^i_{n,j}) &= \boldsymbol{M}\hat{\boldsymbol{k}}^i_{n,j} - \Delta t_n \boldsymbol{f}(\boldsymbol{u}_{n,j}, \tilde{\boldsymbol{c}}^i_{n,j}, \boldsymbol{\mu}),\\
\tilde{\boldsymbol{r}}^i_n(\boldsymbol{u}^i_n, \boldsymbol{u}^i_{n-1}, \boldsymbol{k}^i_{n,1}, \ldots, \boldsymbol{k}^i_{n,s}, \hat{\boldsymbol{k}}^i_{n,1}, \ldots, \hat{\boldsymbol{k}}^i_{n,s}) &= \boldsymbol{u}^i_n - \boldsymbol{u}^i_{n-1} - \sum_{j=1}^{s} b_j\boldsymbol{k}^i_{n,j} - \sum_{j=1}^{s} \hat{b}_j\hat{\boldsymbol{k}}^i_{n,j},\\
\boldsymbol{p}^i_{n,j}(\boldsymbol{u}^1_{n,j}, \ldots, \boldsymbol{u}^{i-1}_{n,j}, \boldsymbol{u}^i_{n-1}, \ldots, \boldsymbol{u}^m_{n-1}, \tilde{\boldsymbol{c}}^i_{n,j}, \boldsymbol{\mu}) &= \tilde{\boldsymbol{c}}^i_{n,j} - \boldsymbol{c}^i(\boldsymbol{u}^1_{n,j}, \ldots, \boldsymbol{u}^{i-1}_{n,j}, \boldsymbol{u}^i_{n-1}, \ldots, \boldsymbol{u}^m_{n-1}, t_{n,j}, \boldsymbol{\mu}),
\end{aligned}
\tag{24}
$$

for $n = 1, \ldots, N_t$, $i = 1, \ldots, m$ and $j = 1, \ldots, s$. Here $\bar{\boldsymbol{u}}^i(\boldsymbol{\mu})$ is the initial condition, and in this work we use a steady-state solution to start the unsteady simulation.

Since the solution of the fully discretized PDE satisfies the above equations, the QoI can be re-written as

$$
J = J - \sum_{n=0}^{N_t}\sum_{i=1}^{m} \boldsymbol{\lambda}^i_n\tilde{\boldsymbol{r}}^i_n - \sum_{n=1}^{N_t}\sum_{j=1}^{s}\sum_{i=1}^{m} \boldsymbol{\kappa}^i_{n,j}\boldsymbol{R}^i_{n,j} - \sum_{n=1}^{N_t}\sum_{j=1}^{s}\sum_{i=1}^{m} \hat{\boldsymbol{\kappa}}^i_{n,j}\hat{\boldsymbol{R}}^i_{n,j} - \sum_{n=1}^{N_t}\sum_{j=1}^{s}\sum_{i=1}^{m} \boldsymbol{\tau}^i_{n,j}\boldsymbol{q}^i_{n,j} - \sum_{n=1}^{N_t}\sum_{j=1}^{s}\sum_{i=1}^{m} \boldsymbol{\sigma}^i_{n,j}\boldsymbol{p}^i_{n,j}
$$

where $\boldsymbol{\lambda}^i_n$, $\boldsymbol{\kappa}^i_{n,j}$, $\hat{\boldsymbol{\kappa}}^i_{n,j}$, $\boldsymbol{\tau}^i_{n,j}$, and $\boldsymbol{\sigma}^i_{n,j}$ are test variables (also known as adjoint state variables or Lagrange multipliers) that respectively enforce the state ODE system, coupling predictor, and initial conditions in Eq. (24). Total differentiation of the modified QoI (or Lagrangian) leads to

$$
\begin{aligned}
\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial J}{\partial \boldsymbol{\mu}} &+ \sum_{i=1}^{m} \boldsymbol{\lambda}^i_0 \frac{\partial \bar{\boldsymbol{u}}^i}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} \boldsymbol{\kappa}^i_{n,j}\frac{\partial \boldsymbol{R}^i_{n,j}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} \hat{\boldsymbol{\kappa}}^i_{n,j}\frac{\partial \hat{\boldsymbol{R}}^i_{n,j}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} \boldsymbol{\sigma}^i_{n,j}\frac{\partial \boldsymbol{p}^i_{n,j}}{\partial \boldsymbol{\mu}} + \sum_{i=1}^{m}\left[-\boldsymbol{\lambda}^i_{N_t}\frac{\partial \tilde{\boldsymbol{r}}^i_{N_t}}{\partial \boldsymbol{u}^i_{N_t}}\right]\frac{\partial \boldsymbol{u}^i_{N_t}}{\partial \boldsymbol{\mu}}\\
&+ \sum_{n=1}^{N_t}\sum_{i=1}^{m}\left[-\boldsymbol{\lambda}^i_{n-1}\frac{\partial \tilde{\boldsymbol{r}}^i_{n-1}}{\partial \boldsymbol{u}^i_{n-1}} - \boldsymbol{\lambda}^i_n\frac{\partial \tilde{\boldsymbol{r}}^i_n}{\partial \boldsymbol{u}^i_{n-1}} - \sum_{j=1}^{s}\boldsymbol{\tau}^i_{n,j}\frac{\partial \boldsymbol{q}^i_{n,j}}{\partial \boldsymbol{u}^i_{n-1}} - \sum_{j=1}^{s}\sum_{p=1}^{i}\boldsymbol{\sigma}^p_{n,j}\frac{\partial \boldsymbol{p}^p_{n,j}}{\partial \boldsymbol{u}^i_{n-1}}\right]\frac{\partial \boldsymbol{u}^i_{n-1}}{\partial \boldsymbol{\mu}}\\
&+ \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s}\left[-\boldsymbol{\lambda}^i_n\frac{\partial \tilde{\boldsymbol{r}}^i_n}{\partial \boldsymbol{k}^i_{n,j}} - \boldsymbol{\kappa}^i_{n,j}\frac{\partial \boldsymbol{R}^i_{n,j}}{\partial \boldsymbol{k}^i_{n,j}} - \sum_{p=j}^{s}\boldsymbol{\tau}^i_{n,p}\frac{\partial \boldsymbol{q}^i_{n,p}}{\partial \boldsymbol{k}^i_{n,j}}\right]\frac{\partial \boldsymbol{k}^i_{n,j}}{\partial \boldsymbol{\mu}}\\
&+ \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s}\left[-\boldsymbol{\lambda}^i_n\frac{\partial \tilde{\boldsymbol{r}}^i_n}{\partial \hat{\boldsymbol{k}}^i_{n,j}} - \hat{\boldsymbol{\kappa}}^i_{n,j}\frac{\partial \hat{\boldsymbol{R}}^i_{n,j}}{\partial \hat{\boldsymbol{k}}^i_{n,j}} - \sum_{p=j+1}^{s}\boldsymbol{\tau}^i_{n,p}\frac{\partial \boldsymbol{q}^i_{n,p}}{\partial \hat{\boldsymbol{k}}^i_{n,j}}\right]\frac{\partial \hat{\boldsymbol{k}}^i_{n,j}}{\partial \boldsymbol{\mu}}\\
&+ \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s}\left[\frac{\partial J}{\partial \boldsymbol{u}^i_{n,j}} - \boldsymbol{\kappa}^i_{n,j}\frac{\partial \boldsymbol{R}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} - \sum_{k=1}^{m}\hat{\boldsymbol{\kappa}}^k_{n,j}\frac{\partial \hat{\boldsymbol{R}}^k_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} - \boldsymbol{\tau}^i_{n,j}\frac{\partial \boldsymbol{q}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}} - \sum_{p=i+1}^{m}\boldsymbol{\sigma}^p_{n,j}\frac{\partial \boldsymbol{p}^p_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}\right]\frac{\partial \boldsymbol{u}^i_{n,j}}{\partial \boldsymbol{\mu}}\\
&+ \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s}\left[-\boldsymbol{\kappa}^i_{n,j}\frac{\partial \boldsymbol{R}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n,j}} - \hat{\boldsymbol{\kappa}}^i_{n,j}\frac{\partial \hat{\boldsymbol{R}}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n,j}} - \boldsymbol{\sigma}^i_{n,j}\frac{\partial \boldsymbol{p}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n,j}}\right]\frac{\partial \tilde{\boldsymbol{c}}^i_{n,j}}{\partial \boldsymbol{\mu}},
\end{aligned}
\tag{25}
$$

here, we re-arrange these terms, such that the state variable sensitivities are isolated. The adjoint state variables $\boldsymbol{\lambda}^i_n$, $\boldsymbol{\kappa}^i_{n,j}$, $\hat{\boldsymbol{\kappa}}^i_{n,j}$, $\boldsymbol{\tau}^i_{n,j}$, and $\boldsymbol{\sigma}^i_{n,j}$, which have remained arbitrary to this point, are chosen such that the bracketed terms in Eq. (25)

vanish. The adjoint equations are

$$\boldsymbol{\lambda}^i_{N_t} = \mathbf{0}, \tag{26a}$$

$$\boldsymbol{\lambda}^i_{n-1} = \boldsymbol{\lambda}^i_n + \sum_{j=1}^{s} \boldsymbol{\tau}^i_{n,j} + \sum_{j=1}^{s}\sum_{p=1}^{i} {\frac{\partial \boldsymbol{c}^p_{n,j}}{\partial \boldsymbol{u}^i_{n-1}}}^T \boldsymbol{\sigma}^p_{n,j}, \tag{26b}$$

$${\boldsymbol{M}^i}^T \boldsymbol{\kappa}^i_{n,j} = b_j \boldsymbol{\lambda}^i_n + \sum_{p=j}^{s} a_{pj} \boldsymbol{\tau}^i_{n,p}, \tag{26c}$$

$${\boldsymbol{M}^i}^T \hat{\boldsymbol{\kappa}}^i_{n,j} = \hat{b}_j \boldsymbol{\lambda}^i_n + \sum_{p=j+1}^{s} \hat{a}_{pj} \boldsymbol{\tau}^i_{n,p}, \tag{26d}$$

$$\boldsymbol{\tau}^i_{n,j} = \frac{\partial J}{\partial \boldsymbol{u}^i_{n,j}} + \Delta t_n {\frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T \boldsymbol{\kappa}^i_{n,j} + \Delta t_n \sum_{k=1}^{m} {\frac{\partial \boldsymbol{f}^k_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T \hat{\boldsymbol{\kappa}}^k_{n,j} + \sum_{p=i+1}^{m} {\frac{\partial \boldsymbol{c}^p_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T \boldsymbol{\sigma}^p_{n,j}, \tag{26e}$$

$$\boldsymbol{\sigma}^i_{n,j} = \Delta t_n {\frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n,j}}}^T \boldsymbol{\kappa}^i_{n,j} + \Delta t_n {\frac{\partial \boldsymbol{f}^i_{n,j}}{\partial \tilde{\boldsymbol{c}}^i_{n,j}}}^T \hat{\boldsymbol{\kappa}}^i_{n,j}, \tag{26f}$$

for $n = 1, \ldots, N_t$, $i = 1, \ldots, m$ and $j = 1, \ldots, s$. These are the fully discrete adjoint equations corresponding to the multiphysics problem in Eq. (1), discrete quantity of interest $\mathcal{J}$, and parameter $\boldsymbol{\mu}$. Solving the adjoint variables reversely from Eq. (26a)-Eq. (26a), the expression for the gradient in Eq. (25) reduces to

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial J}{\partial \boldsymbol{\mu}} + \sum_{i=1}^{m} \boldsymbol{\lambda}^i_0 \frac{\partial \bar{\boldsymbol{u}}^i}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} \Delta t_n {\frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{\mu}}}^T \boldsymbol{\kappa}^i_{n,j} + \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} \Delta t_n {\frac{\partial \boldsymbol{f}^i_{n,j}}{\partial \boldsymbol{\mu}}}^T \hat{\boldsymbol{\kappa}}^i_{n,j} + \sum_{n=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{s} {\frac{\partial \boldsymbol{c}^i_{n,j}}{\partial \boldsymbol{\mu}}}^T \boldsymbol{\sigma}^i_{n,j}$$

Due to the partitioned nature of the multiphysics solver, the adjoint variables can be solved substep-by-substep and subsystem-by-subsystem, the detailed algorithm is presented in Algorithm 3.

Algorithm 3 Adjoint approach

```
for stages $j = s, \ldots, 1$ do
    Read stage solution $\boldsymbol{u}^i_{n-1}, \hat{\boldsymbol{k}}^i_{n,p}, \boldsymbol{k}^i_{n,p}$ for $i = 1, \ldots, m$ from disk.
    for physical systems $i = m, \ldots, 1$ do
        Explicit solve Eq. (26d) for $\hat{\boldsymbol{\kappa}}^i_{n,j}$: ${\boldsymbol{M}^i}^T \hat{\boldsymbol{\kappa}}^i_{n,j} = \hat{b}_j \boldsymbol{\lambda}^i_n + \sum_{p=j+1}^{s} \hat{a}_{pj} \boldsymbol{\tau}^i_{n,p}$
    end for
    for physical systems $i = m, \ldots, 1$ do
        Set $\widetilde{\boldsymbol{\tau}^i_{n,j}} = \frac{\partial J}{\partial \boldsymbol{u}^i_{n,j}} + \Delta t_n \sum_{k=1}^{m} {\frac{\partial \boldsymbol{f}^k_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T \hat{\boldsymbol{\kappa}}^k_{n,j} + \sum_{p=i+1}^{m} {\frac{\partial \boldsymbol{c}^p_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T \boldsymbol{\sigma}^p_{n,j}$
        Implicit solve Eq. (26c) for $\boldsymbol{\kappa}^i_{n,j}$: $\left({\boldsymbol{M}^i}^T - \Delta t_n {\frac{\partial \boldsymbol{g}^i_{n,j}}{\partial \boldsymbol{u}^i_{n,j}}}^T\right) \boldsymbol{\kappa}^i_{n,j} = b_j \boldsymbol{\lambda}^i_n + \sum_{p=j+1}^{s} a_{pj} \boldsymbol{\tau}^i_{n,p} + a_{jj} \widetilde{\boldsymbol{\tau}^i_{n,j}}$
        Construct $\boldsymbol{\tau}^i_{n,j}$ and $\boldsymbol{\sigma}^i_{n,j}$ based on (26e) and (26f)
    end for
end for
Construct $\boldsymbol{\lambda}^i_{n-1}$ based on (26b)
```

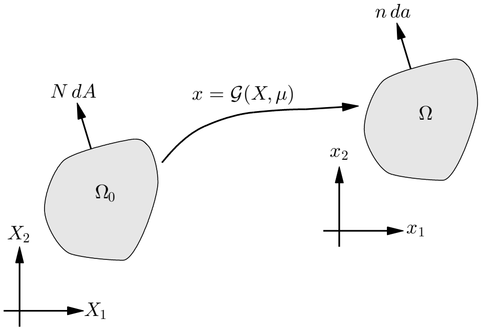

Fig. 1 Mapping between reference and physical domains.

V. Applications

In this section, we demonstrate the proposed high-order optimization procedure on two multiphysics problems: a 1D fluid-structure-mesh three-field coupling piston problem and a 2D fluid-structure two-field coupling foil energy harvesting problem.

A. Governing equations and semi-discretization

1. Compressible fluid flow

The governing equations for compressible fluid flow, defined on a deformable fluid domain $\Omega(\boldsymbol{\mu},\, t)$, can be written as a viscous conservation law

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathcal{F}^{inv}(U) + \nabla \cdot \mathcal{F}^{vis}(U, \nabla U) = 0 \quad \text{in} \quad \Omega(\boldsymbol{\mu}, t), \tag{27}$$

where U is the conservative state variable vector and the physical flux consists of an inviscid part $\mathcal{F}^{inv}(U)$ and a viscous part $\mathcal{F}^{vis}(U,\, \nabla U)$. The conservation law in (27) is transformed to a fixed reference domain Ω_0 by defining a time-dependent diffeomorphism $\mathcal{G}$ between the reference domain and the physical domain; see Figure 1. At each time t, a point X in the reference domain Ω_0 is mapped to $x(X, \boldsymbol{\mu}, t) = \mathcal{G}(X, \boldsymbol{\mu}, t)$ in the physical domain $\Omega(\boldsymbol{\mu}, t)$. The deformation gradient G, velocity v_G, and Jacobian g of the mapping are defined as

$$G = \nabla_X \mathcal{G}, \quad v_G = \frac{\partial \mathcal{G}}{\partial t}, \quad g = \det G. \tag{28}$$

Following the procedure in [14, 15], the governing equation (27) can be written in the reference domain as

$$\frac{\partial U_X}{\partial t} + \nabla_X \cdot \mathcal{F}_X^{inv}(U_X) + \nabla_X \cdot \mathcal{F}_X^{vis}(U_X, \nabla_X U_X) = 0 \quad \text{in} \quad \Omega_0, \tag{29}$$

where ∇_X defines the spatial derivative with respect to the reference domain, conserved quantities and its derivatives in the reference domain are written as

$$U_X = gU, \qquad \nabla_X U_X = g\nabla U_X \cdot G + g^{-1} U_X \frac{\partial g}{\partial X}. \tag{30}$$

The inviscid and viscous fluxes are transformed to the reference domain as

$$\begin{aligned} \mathcal{F}_X^{inv}(U_X) &= g\mathcal{F}^{inv}(g^{-1}U_X)G^{-T} - U_X \otimes G^{-1} v_G, \\ \mathcal{F}_X^{vis}(U_X) &= g\mathcal{F}^{vis}\left(g^{-1}U_X, g^{-1}\left[\nabla_X U_X - g^{-1}U_X \frac{\partial g}{\partial X}\right] G^{-1}\right) G^{-T}. \end{aligned} \tag{31}$$

The governing equations in (29) reduce to the following system of ODEs after an appropriate spatial discretization, such as a discontinuous Galerkin or finite volume method, is applied

$$\boldsymbol{M}^f \dot{\boldsymbol{u}}^f = \boldsymbol{r}^f(\boldsymbol{u}^f,\, \boldsymbol{c}^f,\, \boldsymbol{\mu},\, t), \tag{32}$$

where $\boldsymbol{M}^f$ is the fixed mass matrix, $\boldsymbol{u}^f$ is the semi-discrete fluid state vector, i.e., the discretization of U_X on Ω_0, $\boldsymbol{r}^f(\boldsymbol{u}^f, \boldsymbol{c}^f, \boldsymbol{\mu}, t)$ is the spatial discretization of the transformed inviscid and viscous fluxes on Ω_0, and $\boldsymbol{c}^f$ is the coupling term that might contain information about the domain mapping $\mathcal{G}(X, \boldsymbol{\mu}, t)$. In particular, the coupling term contains the position and velocities of the nodal coordinates of the computational mesh. The domain mapping is defined using an element-wise nodal (Lagrangian) polynomial basis on the mesh with coefficients from the nodal positions and velocities.

2. Simple structure model

In general, the governing equations for the structure will be given by a system of partial differential equation such as the continuum equations in total Lagrangian form with an arbitrary constitutive law. However, in this work, we only consider simple structures like mass-spring-damper systems that can directly be written as a second-order system of ODEs

$$m_s \ddot{u}_s + c_s \dot{u}_s + k_s u_s = f_{ext}(t), \tag{33}$$

where m_s is the mass of the (rigid) object, c_s is the damper resistance constant, k_s is the spring stiffness, and $f_{ext}(t)$ is a time-dependent external load, which will be given by integrating the pointwise force the fluid exerts on the object.

The equations in (33) are re-written in a first-order form, to conform to the notation in this document, as

$$\boldsymbol{M}^s \dot{\boldsymbol{u}}^s = \boldsymbol{r}^s(\boldsymbol{u}^s, \boldsymbol{c}^s, \boldsymbol{\mu}, t). \tag{34}$$

In the case of the simple structure in (33), the mass matrix, state vector, residual, and coupling term are

$$\boldsymbol{M}^s = \begin{bmatrix} m_s & \\ & 1 \end{bmatrix}, \qquad \boldsymbol{u}^s = \begin{bmatrix} \dot{u}_s \\ u_s \end{bmatrix}, \qquad \boldsymbol{c}^s = f_{ext}, \qquad \boldsymbol{r}^s(\boldsymbol{u}^s, \boldsymbol{c}^s) = \begin{bmatrix} f_{ext} - c_s \dot{u}_s - k_s u_s \\ u_s \end{bmatrix}. \tag{35}$$

3. Deformation of the fluid domain

The mesh deformation is generally described by a pseudo-structure driven solely by Dirichlet boundary conditions provided by the displacement of the structure at the fluid-structure interface [16, 17] or a parametrized mapping such as radial basis functions [18–20] or blending maps [15]. Due to different treatments of the mesh deformation, the fluid-structure interaction problem can be formulated as three-field coupling or two-filed coupling problems.

For the first treatment, the governing equations are given by the continuum mechanics equations in total Lagrangian form with an arbitrary constitutive law

$$\begin{aligned} \frac{\partial \bar{p}}{\partial t} - \nabla \cdot P(G) = 0 \qquad & \text{in } \Omega_0 \\ x = x_b \qquad & \text{on } \partial\Omega_0^D \\ \dot{x} = \dot{x}_b \qquad & \text{on } \partial\Omega_0^D, \end{aligned} \tag{36}$$

where $\bar{p}(X, t) = \rho_m \dot{x}$ is the linear momentum, ρ_m is the density, and P is the first Piola-Kirchhoff stress of the pseudo-structure. The deformation gradient G is the mapping that defines the deformation of the reference fluid domain Ω_0 to physical fluid domain $\Omega(t)$. The position and velocity of the fluid domain are prescribed along $\partial\Omega_0^D$, the union of the fluid-structure interface and the fluid domain boundary. The governing equations in (36) reduce to the following system of ODEs after an appropriate spatial discretization, such as the finite element method, is applied and recast in first-order form

$$\boldsymbol{M}^x \dot{\boldsymbol{u}}^x = \boldsymbol{r}^x(\boldsymbol{u}^x, \boldsymbol{c}^x, \boldsymbol{\mu}, t) \tag{37}$$

where $\boldsymbol{M}^x$ is the fixed mass matrix, $\boldsymbol{u}^x(t)$ is the semi-discrete state vector consisting of the displacements and velocities of the mesh nodes, $\boldsymbol{r}^x(\boldsymbol{u}^x, \boldsymbol{c}^x, \boldsymbol{\mu}, t)$ is the spatial discretization of the continuum equations and boundary conditions on the reference domain Ω_0, and $\boldsymbol{c}^x$ is the coupling term that contains information about the motion of the fluid structure interface. This model of the mesh motion leads to a three-field FSI formulation when coupled to the fluid and structure equations.

For the second treatment, the domain mapping $x = \mathcal{G}(X, t)$ is given by an analytical function, parametrized by the deformation and velocity of the fluid-structure interface, that can be analytically differentiated to obtain the

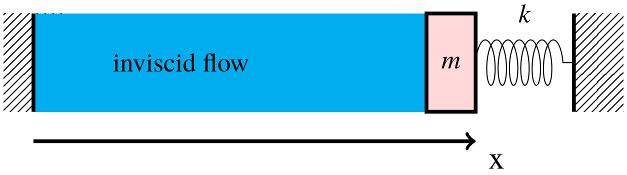

Fig. 2 One-dimensional piston system

deformation gradient $G(X,\, t)$ and velocity $v_G(X,\, t)$. Since the fluid mesh motion is no longer included in the system of time-dependent partial differential equations, this leads to a two-field FSI formulation in terms of the fluid and structure states only.

4. Two-field and three-field fluid-structure coupling

In the three-field fluid-structure interaction setting

$$\boldsymbol{M}^s\dot{\boldsymbol{u}}^s = \boldsymbol{r}^s(\boldsymbol{u}^s,\, \boldsymbol{c}^s,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{M}^x\dot{\boldsymbol{u}}^x = \boldsymbol{r}^x(\boldsymbol{u}^x,\, \boldsymbol{c}^x,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{M}^f\dot{\boldsymbol{u}}^f = \boldsymbol{r}^f(\boldsymbol{u}^f,\, \boldsymbol{c}^f,\, \boldsymbol{\mu},\, t) \tag{38}$$

introduced in [16], the coupling terms have the following dependencies

$$\boldsymbol{c}^s = \boldsymbol{c}^s(\boldsymbol{u}^s,\, \boldsymbol{u}^x,\, \boldsymbol{u}^f,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{c}^x = \boldsymbol{c}^x(\boldsymbol{u}^s,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{c}^f = \boldsymbol{c}^f(\boldsymbol{u}^s,\, \boldsymbol{u}^x,\, \boldsymbol{\mu},\, t). \tag{39}$$

From Eq. (35), the structure coupling term is the external force applied to the structure that comes from integrating the fluid stresses over the fluid-structure interface. The mesh coupling term is the position and velocity of the fluid-structure interface and therefore depends solely on the state of the structure. From Eq. (29)-(30), the fluid coupling term is the position and velocity of the entire fluid mesh and therefore depends on the state of the structure and the mesh.

In the two-field FSI setting

$$\boldsymbol{M}^s\dot{\boldsymbol{u}}^s = \boldsymbol{r}^s(\boldsymbol{u}^s,\, \boldsymbol{c}^s,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{M}^f\dot{\boldsymbol{u}}^f = \boldsymbol{r}^f(\boldsymbol{u}^f,\, \boldsymbol{c}^f,\, \boldsymbol{\mu},\, t) \tag{40}$$

the mesh motion is given by an analytical function and the coupling terms have the following dependencies

$$\boldsymbol{c}^s = \boldsymbol{c}^s(\boldsymbol{u}^s,\, \boldsymbol{u}^f,\, \boldsymbol{\mu},\, t), \quad \boldsymbol{c}^f = \boldsymbol{c}^f(\boldsymbol{u}^s,\, \boldsymbol{\mu},\, t). \tag{41}$$

In this case, the structure coupling term is determined from the fluid and structure state since the external force depends on the traction integrated over the fluid-structure interface. The fluid coupling term, i.e., the position and velocity of the fluid mesh, is determined from the structure state. Finally, the ordering of the subsystems implied in (38) and (40) is used throughout the remainder of this section, which plays an important role when defining the Gauss-Seidel predictors.

B. 1D fluid-structure-mesh three-field coupling piston problem

This proposed optimization procedure is first verified by the canonical FSI model problem: a one-dimensional piston problem (Figure 3). The inviscid fluid is governed by the one-dimensional Euler equations defined on $x \in \Omega(t) = [0,\, 1.0 - u_s]$, where u_s is the displacement of the piston. The fluid flow is the adiabatic gas with constant $\gamma = 1.4$. The fluid is initially at rest $u = 0$ with a density $\rho = 1.0$ and pressure $p = 0.4$. After transformation to the reference domain $\Omega_0 = [0,\, 1]$ following the procedure in Section V.A.1, the equations are semi-discretized by a standard first-order finite volume method using Roe's flux [21] with 100 elements.

The deformation of the fluid mesh is handled by considering the fluid domain to be a pseudo-structure governed by the continuum equations in Eq. (36), restricted to the one-dimensional case with a linear constitutive law and infinitesimal strains assumed

$$\rho_m \ddot{u}_x = E_m \frac{\partial^2 u_x}{\partial X^2} - c_m \dot{u}_x, \tag{42}$$

where $u_x(X,\, t)$ is the mesh displacement vector defined over the reference domain $X \in \Omega_0$ and the density, Young's modulus, and damping coefficient are $\rho_m = 1.0$, $E_m = 1.0$, $c_m = 0.0$, respectively. The governing equation for the mesh deformation is discretized in space using the finite difference method.

Finally, the structure is modeled by a linear mass-spring system as Eq. (33) with piston mass $m_s = 1.0$, spring stiffness $\mu_k = 1.0$, and no damper $c_s = 0$. The piston is initially displaced a distance of $u_s = 0.0$. Once the piston is released, it immediately begins to recede due to the combination of the spring being perturbed from its equilibrium configuration and the flow pressure, which causes a C^0 rarefaction wave near the interface.

The objective function to minimize is set to be the integral of square of the piston displacement till $T = 1.0$

$$\mathcal{J} = \int_0^T u_s^2 dt. \tag{43}$$

The only parameter is the stiffness of the piston μ_k, for verification purpose, an additional constraint $0 \leq \mu_k \leq 10$ is imposed. We should expect that when the stiffness reaches its maximum, the objective function reaches its minimum.

Table 2 shows the objective function and its derivative evaluated in three different ways, using central finite differences with $\epsilon = 10^{-6}$, the direct sensitivity method, and the adjoint method. The results of the direct sensitivity method and the adjoint method are within 10^{-6} of the finite difference results, which verifies the correctness of our current implementation. Moreover, the accuracy of the finite difference method is limited by the "step-size dilemma," therefore the adjoint method and the direct sensitivity method are likely producing more accurate derivatives.

Scheme	$\mathcal{J}$	FD	Direct	Adjoint
IMEX1	5.24027644581e-03	- 6.40416043546e-04	- 6.40416045418e-04	-6.40416045418e-04
IMEX2	5.01357571586e-03	- 5.75379291520e-04	-5.75379340362e-04	-5.75379340362e-04
IMEX3	5.01291619482e-03	-5.75053709945e-04	-5.75053861151e-04	-5.75053861151e-04
IMEX4	5.01291415604e-03	-5.75054676186e-04	-5.75054797593e-04	-5.75054797593e-04

Table 2 1D piston problem: the objective function value and its gradients.

The convergence of the quantities of interest is reported in Figure 3-left, the corresponding convergence of the parameter is reported in Figure 3-right. All IMEX schemes use step size $\Delta t = 0.01$ and lead to convergence in 8 optimization steps. The parameter μ_k converges to its upper bound as expected.

C. 2D fluid-structure two-field coupling foil energy harvesting process

In this section, the high-order, partitioned solver with the optimization framework introduced in this document is applied to find the maximum energy harvesting through flow-induced oscillations of a NACA 0012 foil of length $l = 1$. The two-dimensional energy-harvesting model problem [22] is represented by using a two-field FSI formulation. Consider the mass-damper system in Figure 4, the airfoil is suspended in an isentropic, viscous flow where the rotational motion is a prescribed periodic motion and the vertical displacement u_s is determined by balancing the forces exerted on the airfoil by the fluid and the damper (see Eq. (33)). The airfoil is initially at $\theta(0) = 0$, it matches a prescribed motion for half a period, and then follows a periodic motion, as follows,

$$\theta(t) = \begin{cases} \mu_A \cos(\frac{2t}{T}(\pi + \mu_\phi)), & t < \frac{T}{2}. \\ \mu_A \cos(2\pi f t + \mu_\phi), & t \geq \frac{T}{2}. \end{cases} \tag{44}$$

Here the period $T = 5$ and the frequency is $f = 0.2$.

The fluid is a perfect gas, with the adiabatic gas constant $\gamma = 1.4$, governed by the isentropic Navier-Stokes equations. The isentropic assumption states the entropy of the system is assumed constant, which is tantamount to the flow being adiabatic and reversible. For a perfect gas, the entropy is defined as

$$s = p/\rho^\gamma. \tag{45}$$

The transformed conservation law, as described in Section V.A.1, is discretized with a standard high-order discontinuous Galerkin method using Roe's flux [21] for the inviscid numerical flux and the Compact DG flux [23] for the viscous numerical flux. The DG discretization uses a mesh consisting of 3912 cubic simplex elements ($p = 3$). The second-order

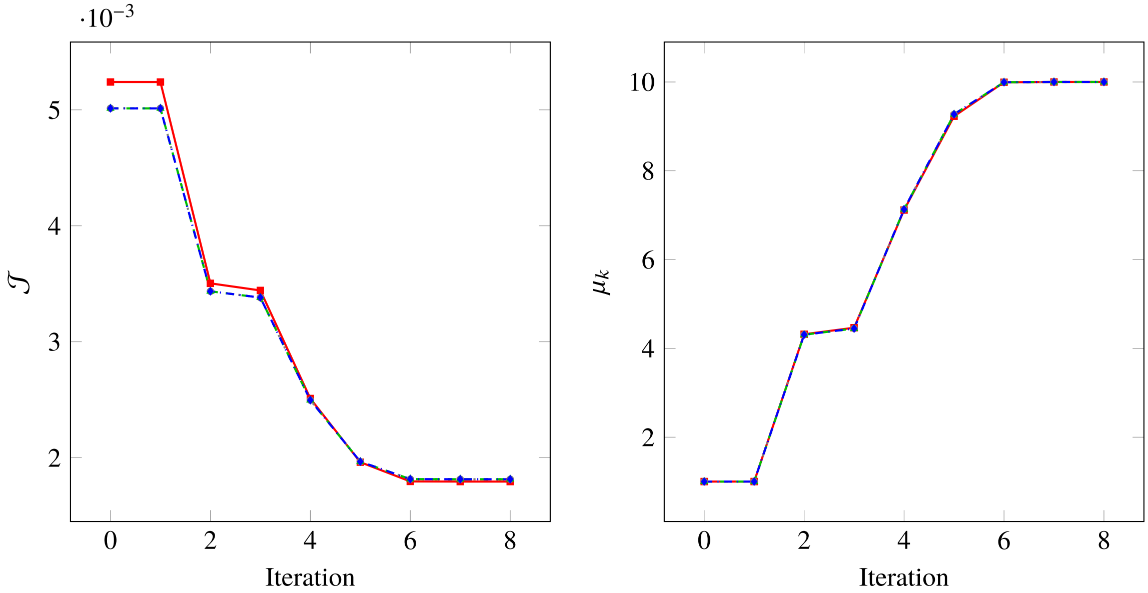

Fig. 3 Convergence of the IMEX1 (—■—), IMEX2 (· · ● · ·), IMEX3 (- -▲- -), and IMEX4 (-·◆·-) schemes when applied to the three-field coupling piston problem: the value of the objective function at each iteration (left); the value of the parameter at each iteration (right).

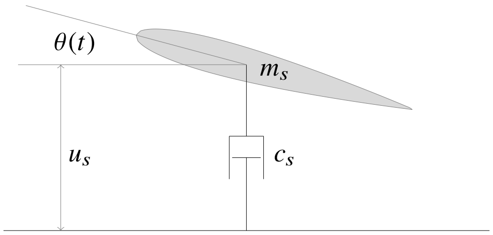

Fig. 4 Foil-damper system

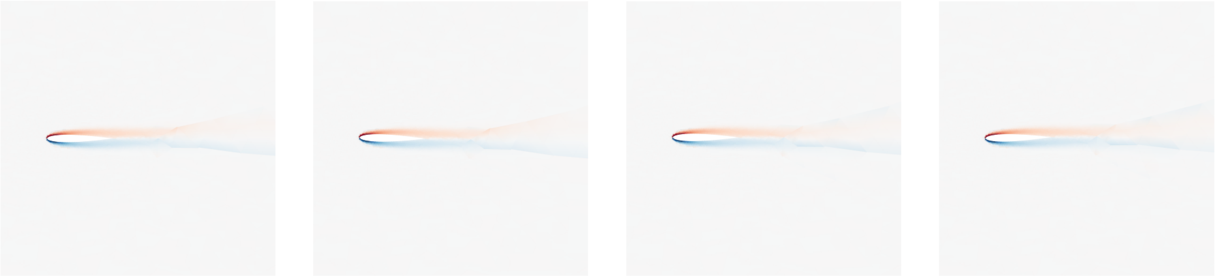

Fig. 5 Airfoil motion and flow vorticity corresponding to foil-damper system under prescribed rotational motion $\theta(t) = \mu_A^{\mathbf{init}} \cos(2\pi f t + \mu_\phi^{\mathbf{init}})$ with frequency $f = 0.2$ at various snapshots in time: $t = T, \frac{5}{4}T, \frac{6}{4}T, \frac{7}{4}T$ (*left*-to-*right*, *top*-to-*bottom*).

ODE in Eq. (33) is the governing equation for the mass-damper system with mass $m_s = 1$, damping constant $c_s = 1$, stiffness $k_s = 0$, and external force given from the fluid as described in Section V.A.2. The mesh motion is determined from the position and velocity of the structure using the blending maps introduced in [15] and identical to that used in Section 5.1 of [14]. IMEX4 is applied for temporal discretization, which matches the expected spatial order of accuracy obtained with polynomials of degree 3.

The objective is to maximize the energy extraction $\mathcal{J} = \frac{1}{T}\int_T^{2T} c_s {\dot{u}_s}^2 dt$ by the device for the second period. The energy injection to maintain the oscillation is defined by $E_\theta = -\frac{1}{T}\int_T^{2T} M_z \dot{\theta} dt$, where M_z is the moment the fluid imparts onto the foil and $\dot{\theta}$ is the rotational speed of the foil. We have linear constraints $-55° \leq \mu_A^{\text{init}} \leq 55°$ for the amplitude parameter and $-\frac{\pi}{2} < \mu_\phi < \frac{\pi}{2}$ for the phase parameter μ_ϕ, and a nonlinear constraint $E_\theta \geq -0.15$ for the energy injection E_θ.

The initial motion is defined by $\mu_A^{\text{init}} = 1°$ and $\mu_\phi^{\text{init}} = 0$. Snapshots of the vorticity field and the motion of the airfoil are shown in Figure 5, and the corresponding energy extraction is close to 0.

For the optimal oscillatory trajectory, the parameters obtained are $\mu_A^{\text{opt}} = 55°$ and $\mu_\phi^{\text{opt}} = -22.95°$. Snapshots of the ensued mesh motion are depicted in Figure 6, snapshots of the vorticity field and motion of the airfoil are shown in Figure 7, and the energy extraction $\mathcal{J} + E_\theta$ by this motion is almost 0.2.

The convergence of the objective function $\mathcal{J}$ and the nonlinear constraint E_θ are reported in Figure 8-left. The convergence of the parameters μ_A and μ_ϕ are presented in Figure 8-right. Initially, the energy harvester extract almost no energy from the fluid without energy injection. However, for the optimal oscillatory trajectory $\theta(t) = \mu_A \cos(2\pi f t + \mu_\phi)$, the injected energy for maintaining the oscillation is $E_\theta = -7.92 \times 10^{-2}$; The energy extracted by the damper is $\mathcal{J} = 2.07 \times 10^{-1}$. The optimized energy harvester can extract $\mathcal{J} + E_\theta = 1.27 \times 10^{-1}$ from the fluid flow, which demonstrate the potential benefits of multiphysics optimization.

VI. Conclusion

We have presented a framework for optimizing unsteady multiphysics systems, based on the high-order, linearly stable, partitioned solver introduced in [1]. An implicit-explicit Runge-Kutta scheme was used for high-order temporal integration with the benefit of achieving accuracy beyond second-order and decoupling all subsystems. Therefore, the corresponding adjoint equations or sensitivity equations can be solved in a partitioned manner, i.e. subsystem-by-subsystem and substage-by-substage. While we did not quantify the benefits of high-order discretizations for these optimization problems, it is still likely that high-order spatial and temporal accuracy allow for smaller mesh size and larger timestep size, which improve the efficiency of function and gradient evaluations in the optimization procedure. Due to the fully discrete adjoint solver, exact gradients are obtained, and the implementation was verified using finite differences. A gradient-based optimizer converged quickly to optional solutions for our examples problems. In future work, the efficiency of the present optimization framework will be studied, and it will be used to better understand the energy harvesting process with multiple airfoils and for the optimization of 3D fluid-structure systems.

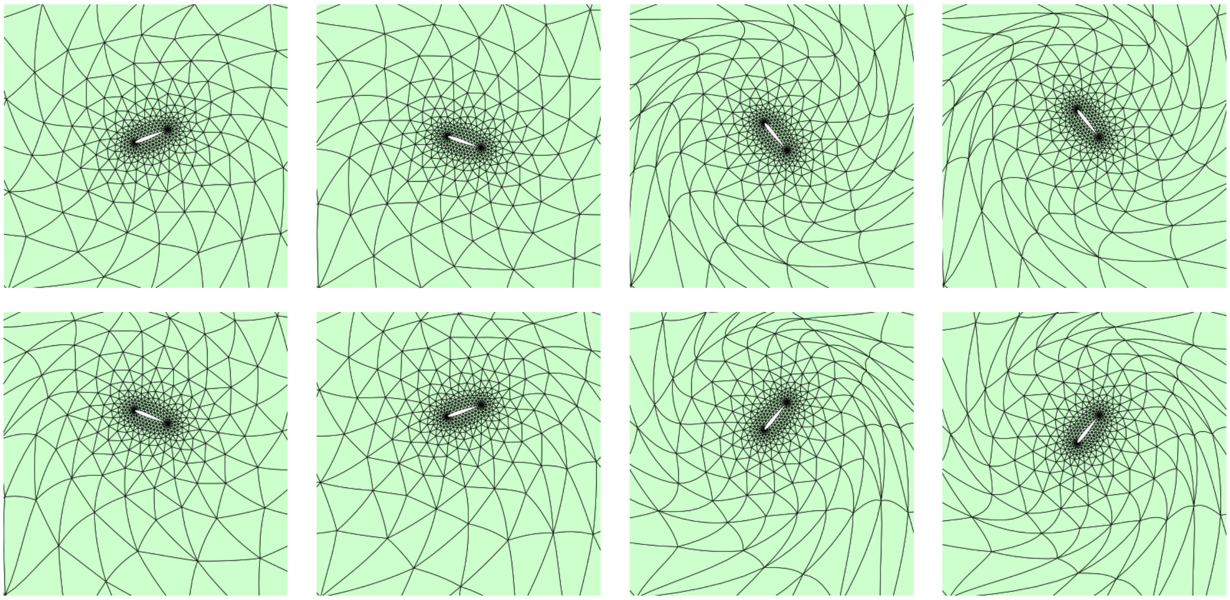

Fig. 6 Airfoil motion and mesh deformation corresponding to foil-damper system under prescribed rotational motion $\theta(t) = \mu_A^{\text{opt}} \cos(2\pi f t + \mu_\phi^{\text{opt}})$ with frequency $f = 0.2$ at various snapshots in time: $t = T, \frac{9}{8}T, \frac{10}{8}T, \frac{11}{8}T, \frac{12}{8}T, \frac{13}{8}T, \frac{14}{8}T, \frac{15}{8}T$ (*left*-to-*right*, *top*-to-*bottom*).

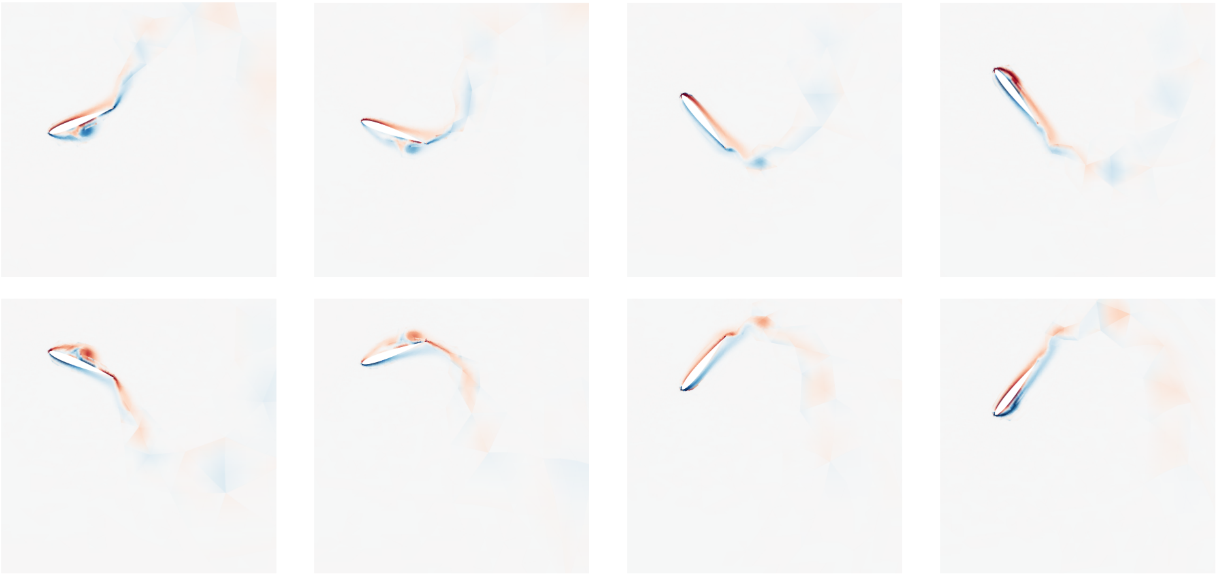

Fig. 7 Airfoil motion and flow vorticity corresponding to foil-damper system under prescribed rotational motion $\theta(t) = \mu_A^{\text{opt}} \cos(2\pi f t + \mu_\phi^{\text{opt}})$ with frequency $f = 0.2$ at various snapshots in time: $t = T, \frac{9}{8}T, \frac{10}{8}T, \frac{11}{8}T, \frac{12}{8}T, \frac{13}{8}T, \frac{14}{8}T, \frac{15}{8}T$ (*left*-to-*right*, *top*-to-*bottom*).

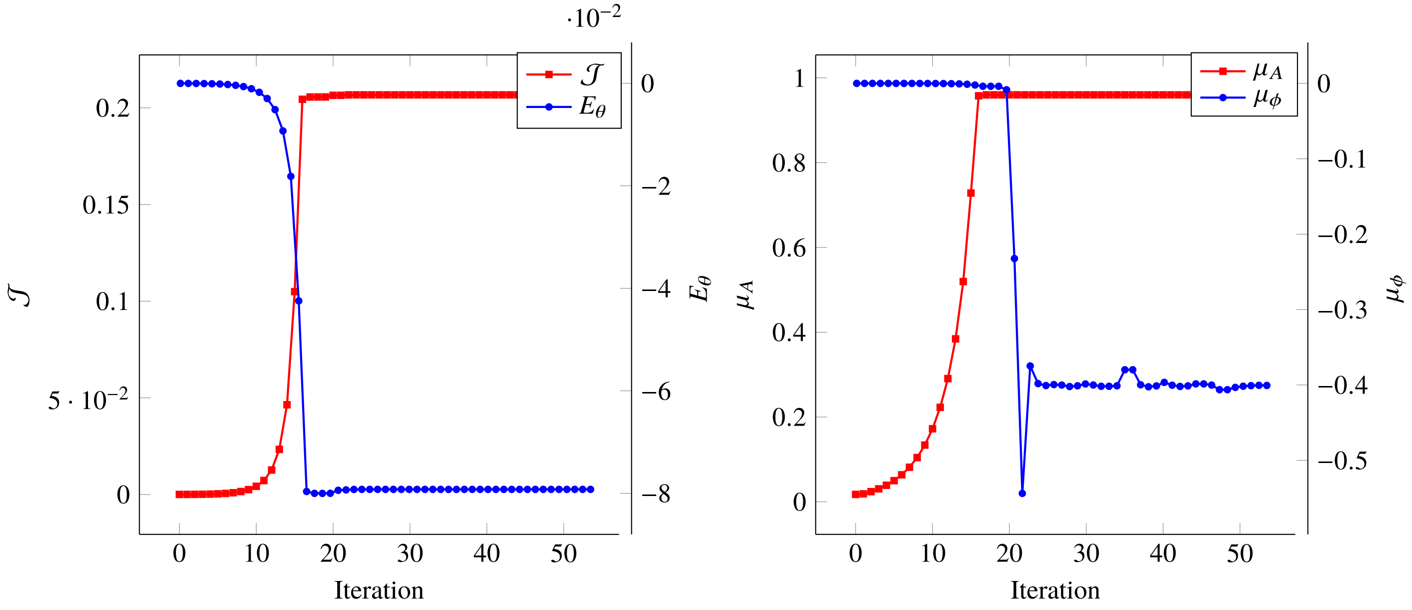

Fig. 8 Convergence of the optimizer for the NACA harvesting problem.

Acknowledgements

This work was supported in part by the Luis W. Alvarez Postdoctoral Fellowship (MZ), by the Director, Office of Science, Office of Advanced Scientific Computing Research, U.S. Department of Energy under Contract No. DE-AC02-05CH11231 (MZ, PP), by the NASA National Aeronautics and Space Administration under grant number NNX16AP15A (MZ, PP), by the Jet Propulsion Laboratory (JPL) under Contract JPL-RSA No. 1590208 (DH) , and by the National Aeronautics and Space Administration (NASA) under Early Stage Innovations (ESI) Grant NASA-NNX17AD02G (DH). 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] Huang, D. Z., Persson, P.-O., and Zahr, M. J., "High-order, linearly stable, partitioned solvers for general multiphysics problems based on implicit-explicit Runge-Kutta schemes," *Computer Methods in Applied Mechanics and Engineering*, 2018.

[2] Ascher, U. M., Ruuth, S. J., and Spiteri, R. J., "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations," *Applied Numerical Mathematics*, Vol. 25, No. 2-3, 1997, pp. 151–167.

[3] Chen, X., Zha, G.-C., and Yang, M.-T., "Numerical simulation of 3-D wing flutter with fully coupled fluid–structural interaction," *Computers & Fluids*, Vol. 36, No. 5, 2007, pp. 856–867.

[4] Stanford, B. K., and Beran, P. S., "Analytical sensitivity analysis of an unsteady vortex-lattice method for flapping-wing optimization," *Journal of Aircraft*, Vol. 47, No. 2, 2010, pp. 647–662.

[5] Day, M. S., and Bell, J. B., "Numerical simulation of laminar reacting flows with complex chemistry," *Combustion Theory and Modelling*, Vol. 4, No. 4, 2000, pp. 535–556.

[6] Jeong, S., Minemura, Y., and Obayashi, S., "Optimization of combustion chamber for diesel engine using kriging model," *Journal of fluid Science and Technology*, Vol. 1, No. 2, 2006, pp. 138–146.

[7] Dreyfus, R., Baudry, J., Roper, M. L., Fermigier, M., Stone, H. A., and Bibette, J., "Microscopic artificial swimmers," *Nature*, Vol. 437, No. 7060, 2005, p. 862.

[8] Mosetti, G., Poloni, C., and Diviacco, B., "Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 51, No. 1, 1994, pp. 105–116.

[9] Xudong, W., Shen, W. Z., Zhu, W. J., Sørensen, J. N., and Jin, C., "Shape optimization of wind turbine blades," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, Vol. 12, No. 8, 2009, pp. 781–803.

[10] Wächter, A., and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, Vol. 106, No. 1, 2006, pp. 25–57.

[11] Zhong, X., "Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows," *Journal of Computational Physics*, Vol. 128, No. 1, 1996, pp. 19–31.

[12] Christopher A, K., and Mark H, C., "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," 2001.

[13] Giles, M. B., and Pierce, N. A., "An introduction to the adjoint approach to design," *Flow, turbulence and combustion*, Vol. 65, No. 3-4, 2000, pp. 393–415.

[14] Zahr, M. J., and Persson, P.-O., "An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems," *Journal of Computational Physics*, Vol. 326, 2016, pp. 516–543.

[15] Persson, P.-O., Bonet, J., and Peraire, J., "Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 17-20, 2009, pp. 1585–1595.

[16] Farhat, C., Lesoinne, M., and Maman, N., "Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution," *International Journal for Numerical Methods in Fluids*, Vol. 21, No. 10, 1995, pp. 807–835.

[17] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "Torsional springs for two-dimensional dynamic unstructured fluid meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 163, No. 1-4, 1998, pp. 231–245.

[18] Van Zuijlen, A., de Boer, A., and Bijl, H., "Higher-order time integration through smooth mesh deformation for 3D fluid–structure interaction simulations," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 414–430.

[19] Rendall, T., and Allen, C., "Unified fluid–structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559.

[20] Froehle, B., and Persson, P.-O., "A high-order discontinuous Galerkin method for fluid–structure interaction with efficient implicit–explicit time stepping," *Journal of Computational Physics*, Vol. 272, 2014, pp. 455–470.

[21] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[22] Peng, Z., and Zhu, Q., "Energy harvesting through flow-induced oscillations of a foil," *Physics of Fluids*, Vol. 21, No. 12, 2009, p. 123602.

[23] Peraire, J., and Persson, P.-O., "The compact discontinuous Galerkin (CDG) method for elliptic problems," *SIAM Journal on Scientific Computing*, Vol. 30, No. 4, 2008, pp. 1806–1824.