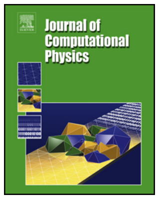
Journal of Computational Physics 252 (2013) 468–494

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Output-based mesh adaptation for high order Navier–Stokes simulations on deformable domains

Steven M. Kast *, Krzysztof J. Fidkowski

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:
Available online 28 June 2013
Received 28 March 2013
Received in revised form 20 May 2013
Accepted 2 June 2013

Keywords:
Output error estimation
Space–time adaptation
Unsteady adjoint
Arbitrary Lagrangian–Eulerian
Discontinuous Galerkin
Geometric conservation law

ABSTRACT

We present an output-based mesh adaptation strategy for Navier–Stokes simulations on deforming domains. The equations are solved with an arbitrary Lagrangian–Eulerian (ALE) approach, using a discontinuous Galerkin finite-element discretization in both space and time. Discrete unsteady adjoint solutions, derived for both the state and the geometric conservation law, provide output error estimates and drive adaptation of the space–time mesh. Spatial adaptation consists of dynamic order increment or decrement on a fixed tessellation of the domain, while a combination of coarsening and refinement is used to provide an efficient time step distribution. Results from compressible Navier–Stokes simulations in both two and three dimensions demonstrate the accuracy and efficiency of the proposed approach. In particular, the method is shown to outperform other common adaptation strategies, which, while sometimes adequate for static problems, struggle in the presence of mesh motion.

1. Introduction

In a typical CFD simulation, most of the data goes to waste. The user is often interested in computing a certain output such as lift or drag, and has little use for auxiliary information. But if this is the case—if the sole objective is to obtain an accurate output—resolving all regions of the flow with equal precision is both unnecessary and inefficient. Instead, a better strategy is to (i) estimate the amount of error in the output, (ii) determine *where* this error originates from, and (iii) drive that error down by targeting the regions of the mesh responsible for it. This strategy of output error estimation and mesh adaptation is especially important for unsteady problems. Here, we present these output-based techniques for Navier–Stokes simulations with moving meshes, such as a wing in flapping flight, and show the efficiency gains possible for these cases.

In large part, the success of an output-based method hinges on the quality of the output error estimate. Accurate *a posteriori* error estimates can be obtained by solving an adjoint problem for the outputs of interest, and this is the strategy we adopt here. An adjoint provides the sensitivity of an output to perturbations in the residuals of the governing equations, which in the context of error estimation can identify regions of the domain contributing most to the output error. These regions can then be adapted to reduce the error and to obtain a more accurate output. This concept is not new, and has received considerable attention for steady problems [1–7]. However, focus has only recently shifted toward use of these methods for unsteady problems.

Computing an unsteady adjoint can be expensive, due in part to significant storage requirements for nonlinear problems, and techniques for reducing these costs are the subject of ongoing research [8–12]. Fortunately, however, the large costs often come with an even larger payoff. These payoffs have been observed in optimization problems [13–15] where a premium

* Corresponding author.
E-mail address: kastsm@umich.edu (S.M. Kast).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.06.007

is placed on obtaining accurate sensitivities with respect to a large number of inputs. This same reasoning extends to unsteady CFD simulations, where small errors made in remote regions of the space–time domain can coalesce into large errors in the output of interest. Identifying these errors with an unsteady adjoint and eliminating them through mesh adaptation can lead to significant computational savings.

Some work in this area has already been done. In a finite-element context, output error estimation for scalar parabolic problems was studied in [16] and [17], with a high order reconstructed adjoint used to drive dynamic space–time mesh adaptation. Recently, spatial-only [18] and combined space–time [19] adaptation have been performed for two-dimensional Navier–Stokes simulations on static domains. Within a finite volume framework, temporal-only adaptation has been shown for the Euler equations on deforming domains [20,21], while on static domains a spatial-only [22] and preliminary space–time adaptation [23] have been demonstrated. Finally, in recent work [24–26], Fidkowski et al. presented combined space–time adaptation strategies for Euler and Navier–Stokes simulations on static domains. In each of the above works, improvements in output convergence were obtained through the use of unsteady adjoint-based adaptation.

In this work, we extend output-based adaptation techniques to high order Navier–Stokes simulations on deforming domains, in both two and three dimensions. Such simulations have far-reaching applications, from bio-inspired flight to aircraft maneuver and flutter analysis. The runs are generally computationally intensive and the resulting solutions are often rich in features. Building on previous work [26,27], we employ combined space–time adaptation of the computational domain, with dynamic-in-time order refinement used to resolve the critical flow features identified by the adjoint. On deformable domains, satisfaction of the so-called geometric conservation law (GCL) requires the adjoint system to change. In this work, we present the required modifications to the adjoint system for an ALE discontinuous Galerkin (DG) method [28], and derive a new discrete adjoint for the GCL itself. We then incorporate this adjoint into the error estimation and adaptation strategy, and evaluate its performance on a series of 2D and 3D test cases. These cases verify the validity of the output-based strategy and show the computational savings that can be achieved.

The remainder of the paper is organized as follows: in Section 2 we introduce the mapping for deformable domains; in Section 3 we discuss the GCL; in Section 4 we describe the primal discretization; in Section 5 we present the error estimation and adjoint discretization; and in Section 6 we discuss the mesh adaptation. Finally, results for several compressible Navier–Stokes simulations are presented in Section 7.

2. Arbitrary Lagrangian–Eulerian mapping

The Navier–Stokes equations can be written in conservation form as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \qquad \vec{\mathbf{F}} = \vec{\mathbf{F}}^i(\mathbf{u}) - \vec{\mathbf{F}}^v(\mathbf{u}, \nabla \mathbf{u}), \tag{1}$$

where $\mathbf{u}(\vec{x}, t) \in \mathbb{R}^s$ is the state vector, $\vec{x} \in \mathbb{R}^d$ is the spatial coordinate, $t \in \mathbb{R}$ is time, and $\vec{\mathbf{F}}^i$ and $\vec{\mathbf{F}}^v$ are the inviscid and viscous fluxes, respectively [29]. For the cases considered in this work, the physical domain in which Eq. (1) holds is deforming in time, and a direct solution would be difficult to obtain. Instead, we can map the problem to a fixed reference domain and solve using an arbitrary Lagrangian–Eulerian (ALE) approach. Since the reference domain remains fixed for all time, standard numerical methods for static problems can then be employed to obtain the solution. A simple and effective ALE method for DG was recently introduced by Persson et al. [28], and we follow their approach here. In this method, the physical equations are transformed to equivalent reference-domain equations, and the deformation of the mesh is encapsulated within the mapping between these domains.

2.1. ALE mapping

The transformation between reference and physical domains is summarized graphically in Fig. 1, and definitions of relevant variables are given in Table 1. Each point $\vec{X}$ in the static reference domain is mapped to a corresponding point $\vec{x}(\vec{X}, t)$ in the physical domain, based on some desired deformation of the mesh. With this mapping, we can then relate each point in the deformed physical domain *back* to a fixed point in the reference domain. The strategy is then to solve for new state variables $\mathbf{u}_X = g\mathbf{u}$ in the reference domain, and later apply the inverse transformation to obtain the physical states we are actually interested in. The variable g is the Jacobian of the mapping, and its deviation from unity indicates whether a region in the reference domain is contracted or dilated in the physical domain.

If we define the reference-domain flux, $\vec{\mathbf{F}}_X$, to be

$$\vec{\mathbf{F}}_X = \vec{\mathbf{F}}_X^i - \vec{\mathbf{F}}_X^v, \qquad \vec{\mathbf{F}}_X^i = g\mathcal{G}^{-1}\left(\vec{\mathbf{F}}^i(\mathbf{u}) - \mathbf{u}\vec{v}_G\right), \qquad \vec{\mathbf{F}}_X^v = g\mathcal{G}^{-1}\vec{\mathbf{F}}^v(\mathbf{u}, \nabla \mathbf{u}),$$

where $\vec{\mathbf{F}}_X^i$ and $\vec{\mathbf{F}}_X^v$ are the inviscid and viscous components, respectively, then the new equations to be satisfied are simply

$$\left.\frac{\partial \mathbf{u}_X}{\partial t}\right|_X + \nabla_X \cdot \vec{\mathbf{F}}_X(\mathbf{u}_X, \nabla_X \mathbf{u}_X) = \mathbf{0}. \tag{2}$$

Implementation of these equations involves the introduction of grid velocity terms into the boundary conditions and the inviscid flux function, as well as modifications to the viscous discretization. The viscous modifications entail the inclusion