PDE-constrained optimization with error estimation and control

Jason E. Hicken* and Juan J. Alonso †

Stanford University, Stanford, California 94305-3030

This paper describes an algorithm for PDE-constrained optimization that controls numerical errors using error estimates and grid adaptation. A key aspect of the algorithm is the use of adjoint variables to estimate errors in the first-order optimality conditions. Multilevel optimization is used to drive the optimality conditions and their estimated errors below a specified tolerance. The error estimate requires two additional adjoint variables, but only at the beginning and end of each optimization cycle. Moreover, the adjoint systems can be formed and solved with limited additional infrastructure. The approach is general and can accommodate both reduced-space and full-space (i.e. one-shot) formulations of the optimization problem. The algorithm is illustrated using the inverse design of a nozzle using the quasi-one-dimensional Euler equations.

I. Introduction

Many aerospace design problems involve complex phenomena that cannot be predicted accurately with simple models. Examples of such phenomena include the sonic-boom from supersonic aircraft, the noise generated from an open-rotor engine, and the unstart of a scramjet engine. In order to perform optimal design in these contexts, engineers must rely on the numerical solution of partial-differential equations (PDEs) to model the complex physics. This is one of the motivations behind the field of PDE-constrained optimization in aerospace engineering.

The value of PDE-constrained optimization has been demonstrated for many aerospace problems. An example is aerodynamic shape optimization, which has been used for airfoil[1–5] and both wing and aircraft design.[6–10] In aerodynamic shape optimization, a parameterized shape is manipulated by an optimization algorithm to improve some objective, e.g. minimize drag for a fixed lift; this objective is predicted using the solution of the *discretized* Euler, Navier-Stokes, or RANS equations.

The accuracy of the discretization plays a significant role in PDE-constrained optimization, since numerical errors in the discrete solution can pollute quantitative measure of the objective. Of course, this is also an issue for individual numerical analyses. However, unlike a single numerical analysis, there is the danger that the optimization algorithm will manipulate the numerical errors in order to "improve" the objective. Consequently, the optimized design may be a reflection of the errors in the discretization rather than the physics of the problem.

The performance of the design can be verified, in some cases, by performing a mesh refinement study; see, for example, Ref. 11. At best, this can only tell us that the objective was accurately predicted by the coarse discretization. On the refined grid the optimality conditions may not be satisfied, and the design may not be a local optimum for the continuous problem.

Strategies exist to estimate and control numerical errors in a systematic way. For example, a posteriori output error estimates and adaptation based on the adjoint variables have been shown to be particularly effective for engineering problems that involve functionals.[12–17]

In contrast to single analyses, incorporating adjoint-based error estimation and control into PDE-constrained optimization has received much less attention. Becker and Rannacher[14] propose using the Lagrangian as the output of interest during optimization; this is also the approach taken by Lu in his thesis.[18] The Lagrangian

*Postdoctoral fellow, Department of Aeronautics and Astronautics, Member AIAA

†Associate Professor, Department of Aeronautics and Astronautics, Senior Member AIAA

is attractive in this context, because the primal and adjoint variables have symmetric relationships in the error estimate and no additional adjoint systems need to be solved.

In this paper we consider an alternative to the Lagrangian for the output of interest during optimization: the norm of the objective gradient. This choice is motivated by following thought experiment. Suppose we have estimated and controlled the error in the Lagrangian to some specified tolerance. Nevertheless, the gradient of the objective/Lagrangian can remain inaccurate. In this scenario, the optimization algorithm may find a spurious local optimum or the location of the optimum may be incorrect. Another possibility is that the gradient-norm error may be small, but the Lagrangian error may be large. In this scenario, we have located the local optimum, but then perform unnecessary refinement to improve the accuracy of the Lagrangian[a].

We show that the gradient-norm error estimates require the solution of two additional adjoint systems; however, these adjoint problems are straightforward to solve if the infrastructure for the adjoint solver is already available. Moreover, we propose a multilevel optimization framework in which these additional adjoint problems are only needed at the beginning and end of each optimization cycle, so their added computational cost is limited.

The paper is organized as follows. We begin by reviewing the generic PDE-constrained optimization problem and describe the reduced-space quasi-Newton algorithm adopted for this work. Section III covers the error estimation process: first, we briefly review the adjoint-weighted residual method for general functionals, and, subsequently, we specialize the method to estimating errors in the gradient-norm and verify these error estimates. In Section IV, we propose an optimization framework that can use gradient-norm error estimates and illustrate this framework using an inverse-design problem on a quasi-one-dimensional nozzle flow. Conclusions can be found in Section V

II. PDE-constrained Optimization

A. Problem Formulation

Ideally, we would like to solve the following nonlinear PDE-constrained problem that involves an infinite-dimensional state variable:

$$\begin{array}{lll} \text{minimize} & \mathcal{J}(x,\mathcal{U}), & x \in R^m,\ \mathcal{U} \in \mathsf{V}, \\ \text{subject to} & \mathcal{R}(\mathcal{U},\mathcal{Z};x) = 0, & \forall\ \mathcal{Z} \in \mathsf{V}, \end{array} \tag{1}$$

where V denotes a function space appropriate for the given PDE. The PDE is represented by the semilinear form $\mathcal{R} : \mathbb{R}^m \times \mathsf{V} \times \mathsf{V} \to \mathbb{R}$, which corresponds to the weak formulation of the PDE. We assume that the objective function, $\mathcal{J} : \mathbb{R}^m \times \mathsf{V} \to \mathbb{R}$, depends explicitly on the state variable $\mathcal{U} \in \mathsf{V}$ and a finite-dimensional control, or design, variable $x \in \mathbb{R}^m$.

Of course, we cannot solve the PDE constraint in (1) analytically (in general), so we must discretize the constraint and seek a finite-dimensional solution. Consequently, the optimization problem (1) is reformulated as follows:

PDEOpt

$$\begin{array}{lll} \text{minimize} & J_h(x,u_h), & x \in \mathbb{R}^m,\ u_h \in \mathsf{V}_h, \\ \text{subject to} & z_h^T \mathcal{R}_h(u_h;x) = 0, & \forall\ z_h \in \mathsf{V}_h. \end{array} \tag{2}$$

Here, the function space V has been replaced by the finite-dimensional vector space V_h.

B. Solution Strategy Overview

There are several approaches that are suitable for solving the problem PDEOpt. When the problem is sufficiently smooth, gradient-based approaches are attractive because they are highly efficient at finding local optima. In this section, we briefly describe the reduced-space gradient-based algorithm used in this work; however, the proposed gradient-norm error estimates are compatible with other gradient-based algorithms, such as the the full-space, i.e. one-shot, approach.

[a]This scenario may not be an issue in practice, since we may want an accurate objective anyway.

The reduced-space approach to PDE-constrained optimization is popular in aerodynamic shape optimization.[8,9,19,20] Reduced-space algorithms recast PDEOpt as a minimization problem that depends only on the design variables x:

$$\text{minimize} \quad J_h\left(x, u_h(x)\right), \qquad x \in \mathbb{R}^m, \tag{3}$$

where u_h is defined as an implicit function of x through the discretized PDE constraint (2). In other words, the PDE constraint is enforced at each optimization iteration.

The reduced-space problem (3) is solved using the limited-memory BFGS[21] quasi-Newton method with line searches based on the strong-Wolfe conditions.[22] BFGS constructs an approximation to the Hessian of J_h using changes in the gradient and the design variables from one iteration to the next. The necessary objective gradients can be computed using finite-differences, algorithmic differentiation, or adjoint variables. We use the adjoint variables,[19,23] because they produce accurate gradients in a computationally efficient manner (the gradient cost is essentially independent of the number of design variables).

The adjoint-based gradient is derived by introducing the Lagrangian

$$L_h(x, u_h, \psi_h) \equiv J_h(x, u_h) + \psi_h^T R_h(u_h; x).$$

Setting the derivatives of L_h to zero, we arrive at the first-order optimality conditions

$$\partial_{\psi_h} L_h = 0 \quad \Rightarrow \quad v_h^T R_h(x, u_h) = 0, \qquad \forall\, v_h \in \mathsf{V}_h. \tag{2}$$

$$\partial_{u_h} L_h = 0 \quad \Rightarrow \quad v_h^T \left[\left(\partial_{u_h} J_h\right)^T + \left(\partial_{u_h} R_h\right)^T \psi_h \right] = 0, \qquad \forall\, v_h \in \mathsf{V}_h, \tag{4}$$

$$\partial_x L_h = 0 \quad \Rightarrow \quad \partial_x J_h + \psi_h^T \partial_x R_h = 0. \tag{5}$$

The first optimality condition recovers the discretized PDE constraint (2). The optimality condition (4) is the discrete adjoint equation. Once u_h and ψ_h are determined using the first two conditions, the gradient of objective can be computed from the expression on the left-hand-side of (5). Note that the gradient of objective is non-zero in general, and satisfying condition (5) is the task of the unconstrained optimization algorithm.

Convergence of the reduced-space problem is determined using a norm of the gradient. Specifically, the problem (3) is considered converged if

$$\|\partial_x L_h\| = \|\partial_x J_h + \psi_h^T \partial_x R_h\| \leq \epsilon, \tag{6}$$

where ϵ is a user-defined tolerance. We can be confident that the variable x is close to a local optimal design of PDEOpt if it satisfies the inequality (6), for sufficiently small ϵ. Unfortunately, satisfying this convergence test does not guarantee that x is a solution of the continuous problem (1), because discretization errors are present in (6). Estimating and controlling the errors in the convergence criterion will provide confidence that x is optimal not only for the discrete problem but also for the continuous problem of interest. This is the topic of the next section.

III. Estimating the Error in First-order Optimality

This section begins with a brief review of the adjoint-weighted residual method of estimating output errors; for additional details on this method see Refs. 12–15, 17, 24. Subsequently, we show how this method can be used to estimate errors in the norm of the objective gradient.

A. Adjoint-weighted Residual Method

Suppose we want to evaluate the functional $\mathcal{J}(\mathcal{U})$, where the function $\mathcal{U}$ is the solution of a PDE. As discussed earlier, in general we cannot solve the PDE explicitly, so we discretize the problem and compute the approximations $u_h \approx \mathcal{U}$ and $J_h(u_h) \approx \mathcal{J}(\mathcal{U})$. The adjoint-weighted residual method provides an estimate of the error between the discrete and continuous functionals:

$$\delta\mathcal{J}_h \equiv J_h(u_h) - \mathcal{J}(\mathcal{U}).$$

We will review this error estimation method in the context of a summation-by-parts (SBP) finite-difference discretization.[25] Although the details differ from one discretization to another, the general error estimation procedure is similar.

Consider a uniform computational mesh with spacing $h = 1/n$, and let $u_h \in \mathbb{R}^{n+1}$ be the solution of the discretized PDE on this mesh. The discretized PDE is represented by the Galerkin statement

$$z_h^T R_{h,p}(u_h) = 0, \qquad \forall\ z_h \in \mathbb{R}^{n+1}, \tag{7}$$

where $R_{h,p}$ denotes the residual. For now, we have excluded the dependence of the residual on the design variable x. The subscript p indicates that the discretization uses a p-order accurate SBP operator; to be precise, the operator is $2p$-order accurate in the interior and p-order accurate at a finite number of boundary nodes, resulting in a $p+1$ order accurate solution u_h. Geometric terms in the functional are also discretized with the same SBP operator. The discrete functional is denoted by $J_{h,p}(u_h)$ and is $2p$-order accurate[b].

The adjoint-weighted residual estimate for the error in $J_{h,p}(u_h)$ is given by[25]

$$\begin{aligned} \delta J_{h,p} &\equiv J_{h,p}(u_h) - J_{h,q}(u_h) - \psi_h^T R_{h,q}(u_h) \\ &= \delta \mathcal{J}_{h,p} + \mathrm{O}(h^{2p-r+3}), \end{aligned} \tag{8}$$

where ψ_h is the adjoint variable, i.e. the solution of (4). The integer $r \leq 2$ is the order of the derivative operator present in the continuous PDE. In particular, the estimate is $\mathrm{O}(h^{2p+2})$ accurate for hyperbolic problems and $\mathrm{O}(h^{2p+1})$-order accurate for second-order elliptic and parabolic problems.

The error estimate (8) consists of two parts. The first part, $J_{h,p}(u_h) - J_{h,q}(u_h)$, is the difference between the functional evaluated with a p-order accurate SBP operator and a q-order accurate SBP operator, $q > p$. This first part accounts for errors in geometric quantities (e.g. surface normals, metric Jacobians, etc).

The second part of the error estimate is the adjoint-weighted residual $\psi_h^T R_{h,q}(u_h)$. Here, ψ_h is the discrete adjoint variable satisfying the dual (adjoint) linear system

$$z_h^T S_{h,p}(\psi_h; u_h) \equiv \psi_h^T R'_{h,p}[u_h](z_h) + J'_{h,p}[u_h](z_h) = 0, \qquad \forall\ z_h \in \mathbb{R}^{n+1}. \tag{9}$$

The prime indicates Fréchét linearization with respect to the variable in the square brackets; this operation is equivalent to partial differentiation for the present SBP finite-difference discretizations. For example,

$$\psi_h^T R'_{h,p}[u_h](z_h) \equiv \psi_h^T \left[\frac{\partial R_{h,p}}{\partial u_h} \right] z_h.$$

The dual problem (9) has the same p-order accurate boundary closure as the primal problem (7). While ψ_h and u_h are evaluated using p-order accurate discretizations, they are substituted into $R_{h,q}$ in the error estimate (8), which uses a q-order accurate SBP operator.

B. Gradient-norm Error Estimation

The output of interest during optimization is the norm of the objective gradient. This norm appears in the first-order optimality condition (6), which provides a measure of how close the design is to a stationary point and is used to decide when to terminate the optimization. However, in PDE-constrained optimization, discretization errors affect both the objective and its gradient. To what extent can we trust the norm of the (discretized) gradient as a criterion for convergence?

If we had some estimate for the error in the gradient norm, we could construct a more reliable criterion for convergence. Fortunately, the adjoint-weighted residual method can be readily applied to gradient-norm error estimation. To use the method, we need only recognize that the "residual" for the gradient-norm is a compound residual consisting of the primal and dual problems.

Let $G_{h,p} \in \mathbb{R}^m$ be the gradient of some output functional $J_{h,p}$ with respect to the design variables $x \in \mathbb{R}^m$. Using the reduced-space approach, recall that the gradient can be expressed as (see (5))

$$G_{h,p}(x, u_h, \psi_h) = \partial_x J_{h,p}(x, u_h) + \psi_h^T \partial_x R_{h,p}(u_h; x),$$

where the primal (u_h) and adjoint (ψ_h) variables satisfy the equations (7) and (9), respectively. We will eventually consider error estimates for both the L2- and infinity-norm of the gradient. For now, we will use $N_{h,p}$ to denote a generic norm of the gradient $G_{h,p}$.

$$\|G_{h,p}\| \equiv N_{h,p}\left(G_{h,p}^{(1)}, G_{h,p}^{(2)}, \ldots, G_{h,p}^{(m)}\right) = N_{h,p}(x, u_h, \psi_h),$$

[b]This superconvergence assumes that the problem is discretized in a dual-consistent manner.[18,26,27]

where $G_{h,p}^{(i)}$ is the i^{th} component of $G_{h,p}$.

We want an error estimate for the norm $N_{h,p}$. To use the adjoint-weighted residual method, we must identify the dual problems corresponding to this functional and the governing equations (7) and (9). To simplify the process, recognize that the test functions for the primal and dual problems are independent, so we can write a single residual equations for both problems[c]:

$$v_h^T R_{h,p}(u_h; x) + z_h^T S_{h,p}(\psi_h; x, u_h) = 0, \qquad \forall\, v_h, z_h \in \mathbb{R}^{n+1}.$$

Consider infinitesimal perturbations δu_h, $\delta\psi_h$, $\delta R_{h,p}$ and $\delta S_{h,p}$ applied to the state, adjoint, and their respective residuals. Using the compound residual, these perturbations are related by equation (see, for example, Ref. 17 for a review)

$$\delta R_{h,p}(v_h) + \delta S_{h,p}(z_h) + v_h^T R'_{h,p}[u_h](\delta u_h; x) + z_h^T S'_{h,p}[u_h](\psi_h; x, \delta u_h) + z_h^T S'_{h,p}[\psi_h](\delta\psi_h; x, u_h) = 0, \qquad \forall\, v_h, z_h \in \mathbb{R}^{n+1}. \quad (10)$$

The adjoint variables for the gradient-norm functional are defined as the sensitivity of $N_{h,p}$ to the compound residual perturbation. Specifically, if w_h and λ_h denote these adjoint variables, then

$$\delta N_{h,p} \equiv N_{h,p}(x, u_h + \delta u_h, \psi_h + \delta\psi_h) - N_{h,p}(x, u_h, \psi_h) = \delta R_{h,p}(\lambda_h) + \delta S_{h,p}(w_h).$$

Linearizing the norm about (u_h, ψ_h) we find

$$N'_{h,p}[u_h](x, \delta u_h, \psi_h) + N'_{h,p}[\psi_h](x, u_h, \delta\psi_h) = \delta R_{h,p}(\lambda_h) + \delta S_{h,p}(w_h). \quad (11)$$

Substituting w_h for z_h and λ_h for v_h in the expression for the compound residual perturbation (10) and using the result to replace $\delta R_{h,p}(\lambda_h) + \delta S_{h,p}(w_h)$ in (11), we have

$$\begin{aligned} &N'_{h,p}[u_h](x, \delta u_h, \psi_h) + N'_{h,p}[\psi_h](x, u_h, \delta\psi_h) \\ &\qquad = -\lambda_h^T R'_{h,p}[u_h](\delta u_h; x) - w_h^T S'_{h,p}[u_h](\psi_h; x, \delta u_h) - w_h^T S'_{h,p}[\psi_h](\delta\psi_h; x, u_h) \end{aligned}$$

Finally, the above equation should be true for arbitrary perturbations δu_h and $\delta\psi_h$; thus, we arrive at the following linear systems for the adjoint variables of the gradient-norm.

$$w_h^T S'_{h,p}[\psi_h](z_h; x, u_h) = -N'_{h,p}[\psi_h](x, u_h, z_h), \qquad \forall\, z_h \in \mathbb{R}^{n+1} \quad (12)$$

$$\lambda_h^T R'_{h,p}[u_h](z_h; x) = -N'_{h,p}[u_h](x, z_h, \psi_h) - w_h^T S'_{h,p}[u_h](\psi_h; x, z_h), \qquad \forall\, z_h \in \mathbb{R}^{n+1} \quad (13)$$

The linear system (12) depends only on w_h while (13) depends on both w_h and λ_h. Consequently, (12) is solved first and, subsequently, w_h is substituted into (13) to solve for λ_h. Once these adjoint variables are available, they can be substituted into the gradient-norm error estimate, which follows from (8).

$$\delta N_{h,p} \equiv N_{h,p}(x, u_h, \psi_h) - N_{h,q}(x, u_h, \psi_h) - \lambda_h^T R_{h,q}(u_h; x) - w_h^T S_{h,q}(\psi_h; x, u_h)$$

As with the functional error estimate, $N_{h,q}$ denotes the gradient (5) evaluated with the q-order accurate SBP operator. Similarly for the residuals $R_{h,q}$ and $S_{h,q}$. The gradient-norm error estimate will have the same asymptotic convergence rate as the functional error estimate, i.e. h^{2p+2} for hyperbolic problems and h^{2p+1} for second-order elliptic and parabolic problems.

C. A Closer Look at the Gradient-norm Adjoint Equations

The adjoint linear systems (12) and (13) seem onerous to evaluate and solve, but closer inspection reveals that these equations are less daunting than they first appear. Consider their respective system matrices. The i^{th} row of the matrix in (12) can be determined by making the choice $z_h = e_i$, where e_i is the i^{th} column of the identity matrix as follows:

$$\begin{aligned} w_h^T S'_{h,p}[\psi_h](e_i; x, u_h) &= \frac{\partial}{\partial\alpha}\left[w_h^T S_{h,p}(\psi_h + \alpha e_i; x, u_h)\right]\Big|_{\alpha=0} \\ &= \frac{\partial}{\partial\alpha}\left[(\psi_h + \alpha e_i)^T R'_{h,p}[u_h](w_h; x) + J'_{h,p}[u_h](w_h; x)\right]\Big|_{\alpha=0} \\ &= e_i^T \left[\frac{\partial R_{h,p}}{\partial u_h}\right] w_h. \end{aligned}$$

[c] We are creating a block vector of residuals.

We recognize the system matrix multiplying w_h in (12) as simply the Jacobian of $R_{h,p}$ evaluated at u_h. Similarly, setting $z_h = e_i$ in (13) we find

$$\begin{aligned}\lambda_h^T R'_{h,p}[u_h](e_i;x) &= \frac{\partial}{\partial\alpha}\left[\lambda_h^T R_{h,p}(u_h+\alpha e_i;x)\right]\Big|_{\alpha=0}\\ &= \lambda_h^T R'_{h,p}[u_h](e_i;x)\\ &= e_i^T\left[\frac{\partial R_{h,p}}{\partial u_h}\right]^T\lambda_h.\end{aligned}$$

The matrix multiplying λ_h is the transposed Jacobian of $R_{h,p}$.

The infrastructure necessary to solve a linear system involving the transposed Jacobian is readily available in optimization algorithms that use the adjoint-approach to compute the gradient. The infrastructure for solving a system involving the Jacobian matrix itself, e.g. (12), is less common — notwithstanding Newton-based solvers for the primal problem. However, much of the machinery used to solve the transposed-Jacobian system can be adapted. Indeed, building and solving the Jacobian system is straightforward in comparison. For example, Jacobian-vector products needed in Krylov iterative methods can be constructed using finite-differences, the complex-step method, or the forward-mode of algorithmic differentiation.

Next we consider the right-hand sides of the linear systems (12) and (13). For the i^{th} row of (12) we have (recalling the gradient definition (5))

$$\begin{aligned}N'_{h,p}[\psi_h](x,u_h,e_i) &= \frac{\partial}{\partial\alpha}\left[N_{h,p}(x,u_h,\psi_h+\alpha e_i)\right]\Big|_{\alpha=0}\\ &= \sum_{j=1}^{m}\left(\frac{\partial N_{h,p}}{\partial G_{h,p}^{(j)}}\right)\frac{\partial}{\partial\alpha}\left[G_{h,p}^{(j)}(x,u_h,\psi_h+\alpha e_i)\right]\Big|_{\alpha=0}\\ &= \sum_{j=1}^{m}\left(\frac{\partial N_{h,p}}{\partial G_{h,p}^{(j)}}\right)\left(e_i^T\frac{\partial R_{h,p}}{\partial x_j}\right).\end{aligned}$$

This is simply the product between the sensitivity of the norm to the gradient components and the sensitivity of the primal residual to the design variables. The former is easy to compute while the latter is available in any adjoint-based optimization framework.

The right-hand side of (13) consists of two terms. The first term involves the gradient-norm $N_{h,p}$, and its i^{th} component is given by

$$\begin{aligned}N'_{h,p}[u_h](x,e_i,\psi_h) &= \frac{\partial}{\partial\alpha}\left[N_{h,p}(x,u_h+\alpha e_i,\psi_h)\right]\Big|_{\alpha=0}\\ &= \sum_{j=1}^{m}\left(\frac{\partial N_{h,p}}{\partial G_{h,p}^{(j)}}\right)\frac{\partial}{\partial\alpha}\left[G_{h,p}^{(j)}(x,u_h+\alpha e_i,\psi_h)\right]\Big|_{\alpha=0}\\ &= \sum_{j=1}^{m}\left(\frac{\partial N_{h,p}}{\partial G_{h,p}^{(j)}}\right)\left(\frac{\partial^2 J_{h,p}}{\partial u_h\partial x_j}+\psi_h^T\frac{\partial^2 R_{h,p}}{\partial u_h\partial x_j}\right)e_i.\end{aligned}\tag{14}$$

The second term of the right-hand side of (13) involves the dual residual $S_{h,p}$. The i^{th} component of this term is

$$\begin{aligned}w_h^T S'_{h,p}[u_h](\psi_h;x,e_i) &= \frac{\partial}{\partial\alpha}\left[w_h^T S_{h,p}(\psi_h;x,u_h+\alpha e_i)\right]\Big|_{\alpha=0}\\ &= \frac{\partial}{\partial\alpha}\left\{\psi_h^T R'_{h,p}[u_h+\alpha e_i](w_h;x)+J'_{h,p}[u_h+\alpha e_i](w_h;x)\right\}\Big|_{\alpha=0}\\ &= \frac{\partial}{\partial u_h}\left(\psi_h^T\frac{\partial R_{h,p}}{\partial u_h}w_h+\frac{\partial J_{h,p}}{\partial u_h}w_h\right)e_i.\end{aligned}\tag{15}$$

We see that (14) and (15) require second-order derivatives of the objective and residual $R_{h,p}$; however, these Hessian components appear in products with vectors, so they do not need to be explicitly formed and

stored. The necessary Hessian-vector products can be computed by applying finite-difference approximations, the complex-step method,[28,29] or the forward-mode of algorithmic differentiation[30] to the adjoint residual $S_{h,p}$.

To illustrate this, we consider the use of finite-difference approximations to compute the right-hand-side of (13). An approximation to (14) is given by

$$\begin{aligned} N'_{h,p}[u_h](x, e_i, \psi_h) &= \frac{S_{h,p}(\psi_h; x + \epsilon y, u_h) - S_{h,p}(\psi_h; x, u_h)}{\epsilon} + \mathrm{O}(\epsilon) \\ &= y_j \left(\psi_h^T \frac{\partial^2 R_{h,p}}{\partial u_h \partial x_j} + \frac{\partial^2 J_{h,p}}{\partial u_h \partial x_j} \right) + \mathrm{O}(\epsilon) \end{aligned} \tag{16}$$

where the components of y are defined by

$$y_j \equiv \frac{\partial N_{h,p}}{\partial G_{h,p}^{(j)}}.$$

For an approximation to (15) we perturb the primal variable u_h appearing in $S_{h,p}$:

$$\begin{aligned} w_h^T S'_{h,p}[u_h](\psi_h; x, e_i) &= \left[\frac{S_{h,p}(\psi; x, u_h + \epsilon w_h) - S_{h,p}(\psi; x, u_h)}{\epsilon} \right]_i + \mathrm{O}(\epsilon) \\ &= \left[\psi_h^T \frac{\partial^2 R_{h,p}}{\partial u_h^2} w_h + \frac{\partial^2 J_{h,p}}{\partial u_h^2} w_h \right]_i + \mathrm{O}(\epsilon). \end{aligned} \tag{17}$$

Thus, the right-hand side of (13) can easily be computed using the above forward-difference approximations, provided the dual residual $S_{h,p}$ can be computed for a given u_h, ψ_h and x. Granted, choosing ϵ such that the approximations are accurate often requires trial-and-error, and the finite-difference approach may fail entirely for poorly scaled problems. In such cases, complex-step or forward-mode algorithmic differentiation are reliable alternatives.

D. Error Estimate Verification

We verify the proposed gradient-norm error estimates using the inviscid nozzle problem. The flow is modelled using the quasi-one-dimensional Euler equations. If $A(x)$ denotes the nozzle cross-sectional area, then the governing equations are

$$\frac{\partial \mathcal{F}}{\partial x} - \mathcal{G} = 0, \qquad \forall\, x \in [0, 1], \tag{18}$$

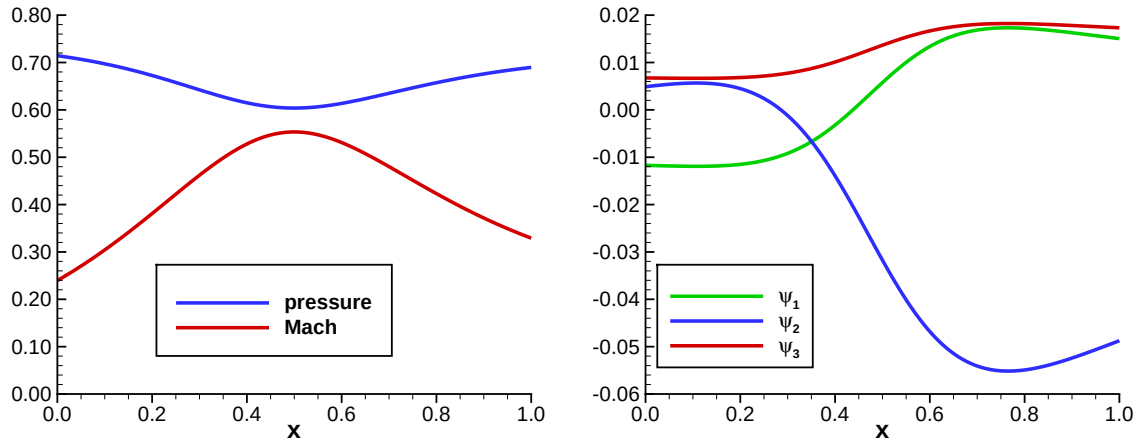
where the flux and source are given by

$$\mathcal{F} = \begin{pmatrix} \rho u A \\ (\rho u^2 + p) A \\ u(e + p) A \end{pmatrix}, \qquad \text{and} \quad \mathcal{G} = \begin{pmatrix} 0 \\ p \frac{dA}{dx} \\ 0 \end{pmatrix},$$

respectively. The unknown variables are the density, ρ, momentum per unit volume, ρu, energy per unit volume, e, and pressure, p. The underdetermined system is closed using the ideal gas law equation of state for the pressure: $p = (\gamma - 1)(e - \frac{1}{2}\rho u^2)$.

Boundary conditions at the inlet and outlet are provided by the exact solution, which is determined using the Mach relations. The stagnation temperature and pressure are 300K and 100,000 kPa, respectively. The specific gas constant is taken to be 287 J/(kg K) and the critical nozzle area is $A^* = 0.8$. For the remaining discussion, the equations and variables are nondimensionalized using the density and sound speed at the inlet.

The nozzle area is parameterized using a cubic b-spline consisting of 22 control points and an open uniform knot vector. The control points at the ends of the nozzle are fixed such that the area satisfies $A(0) = 2$ and $A(1) = 1.5$. Consequently, there are 20 design variables. For the purposes of verifying the error estimate, these remaining control points are set such that the nozzle area is a cubic function whose



(a) pressure and Mach number

(b) adjoint ψ for $\mathcal{J}$

Figure 1. Pressure and Mach number (left) for the error estimate verification, and the adjoint variables (right) corresponding to the inverse-design functional

minimum area of 1 is located at $x = 0.5$. Figure 1(a) shows the pressure and Mach number corresponding to this nozzle area and the given stagnation values.

The quasi-one-dimensional Euler equations are discretized using a second-order accurate (i.e. $p = 1$) SBP scheme, with boundary conditions imposed weakly using penalty terms.[31,32] Scalar fourth-difference artificial dissipation is added to prevent oscillatory solutions. The discretized Euler equations are solved using a Newton-GMRES algorithm.[33,34] GMRES[35] is preconditioned using an LU factorization of a first-order accurate discretization based on nearest-neighbours. The linear adjoint problems are also solved using GMRES with the appropriate LU or $\mathrm{U}^T\mathrm{L}^T$ factorized first-order preconditioner.

The objective function is based on an inverse problem for the pressure:

$$\mathcal{J} = \int_0^1 \frac{1}{2}(p - p_{\text{targ}})^2 \, dx, \tag{19}$$

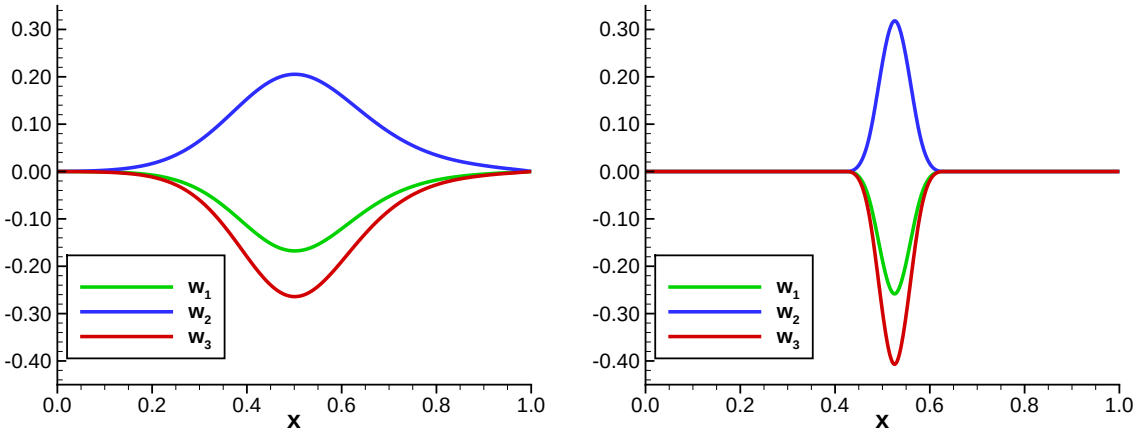
where the target pressure is the constant $p_{\text{targ}} = 1/\gamma$. The objective function is discretized using a quadrature based on the SBP norm. To compute the gradient-norm error estimate, a third-order accurate (i.e. $q = 2$) SBP scheme is adopted. The adjoint variables for this objective function are plotted in Figure 1(b).

Before verifying the accuracy of the gradient-norm error estimates, let us briefly investigate the adjoint variables w_h and λ_h associated with the L2- and infinity-norm. Figure 2 plots these variables for the present problem. The most salient feature is the localized nature of w_h corresponding to the infinity-norm. However, this compact support derives from the compact support of the b-spline basis functions, and it is not a general property of w_h.

Error estimates for $\|G_{h,p}\|_2$ and $\|G_{h,p}\|_\infty$ are shown in Figures 3(a) and 3(b), respectively, for a sequence of 32 uniform grids with nodes $n = 21, 41, 61, \ldots, 641$. The figures compare the error estimates with the actual error; the actual error is based on a solution from a fourth-order accurate discretization on a grid with $n = 2561$ nodes. The figures also plot the difference between the actual error and the estimated error, which reveals the rate of convergence of the error estimate.

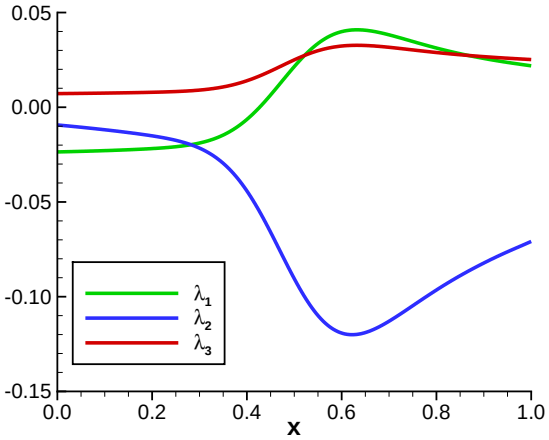
The results show that the error estimates for both the L2- and infinity-norms of the gradient are accurate for this problem, with the exception of the coarsest grid ($n = 21$ nodes). The error estimate for the L2-norm has the predicted fourth-order convergence rate. The estimate for the infinity-norm shows some noise on the finer grid levels. The index of the gradient component with largest absolute value is likely changing with refinement. Nevertheless, the error estimate remains effective for this problem.

Finally, we repeated the error-estimate verification using the finite-difference approximations (16) and (17) to compute the right-hand-side of the adjoint system (13). We found that the finite-difference approximations had minimal impact on the accuracy of the error estimates. For completeness, results from these tests are provided in Appendix A.

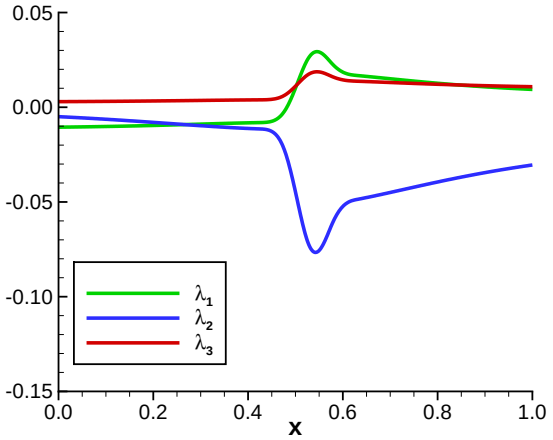


(a) adjoint w_h for $\|G_{h,p}\|_2$

(b) adjoint w_h for $\|G_{h,p}\|_\infty$

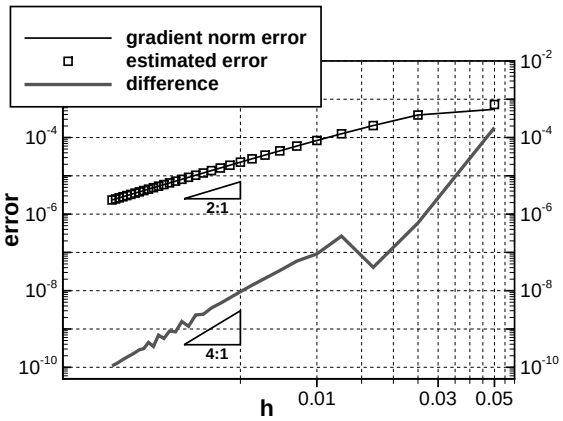


(c) adjoint λ_h for $\|G_{h,p}\|_2$

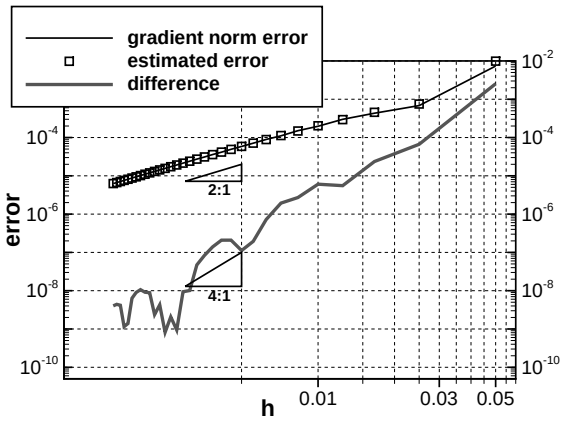


(d) adjoint λ_h for $\|G_{h,p}\|_\infty$

Figure 2. Adjoint variables corresponding to the L2-norm of the gradient (left) and the infinity-norm of the gradient (right)



(a) Error and estimate for $\|G_{h,p}\|_2$



(b) Error and estimate for $\|G_{h,p}\|_\infty$

Figure 3. Error and error estimate in $\|G_{h,p}\|_2$ (left) and $\|G_{h,p}\|_\infty$ on a sequence of grids. Also shown is the difference between the error and error estimate.

IV. Optimization With Error Estimation and Control

A. Algorithm Overview

There are several possible ways to incorporate gradient-norm error estimates into an optimization framework. In Algorithm 1, we propose a multilevel optimization that uses the error estimates at the beginning and end of each cycle.

At the beginning of level l, the gradient-norm error is estimated and used to define the tolerance that measures convergence of the optimization on that level (line 5 of the algorithm). In particular, the optimization at level l is considered converged if

$$N_{h_l,p}(x) \leq \tau \delta N_{h_l,p}(x_l),$$

where $\delta N_{h_l,p}(x_l)$ is the gradient-norm error estimate for the initial design x_l at level l. The parameter $\tau > 1$ is used to help prevent the gradient-norm from becoming smaller than its error: if the norm becomes smaller than its error, the optimization will target numerical errors rather than locating the desired local optimum.

A new error estimate is computed for the gradient-norm of the design x produced by the optimization on level l (see line 6). The sum of this error estimate and the gradient-norm is used to determine if the analytical first-order optimality conditions are sufficiently satisfied. Let $N_{h_0,p}(x_0)$ denote the gradient norm of the initial design on the initial grid. If the inequality

$$N_{h_l,p}(x) + \delta N_{h_l,p}(x) \leq \epsilon N_{h_0,p}(x_0), \tag{20}$$

is satisfied, and $\epsilon \in (0,1)$ is sufficiently small, we can be confident that the design x accurately approximates a local optimum for the analytical problem.

Algorithm 1: Optimization with Gradient-Norm Error Estimation and Control

```
1  choose τ ≥ 1 and ϵ ∈ (0, 1)
2  choose initial value for design variables, x_0
3  for l = 0, 1, 2, ... do
4      compute gradient-norm error estimate: δN_{h_l,p}(x_l)
5      find x such that N_{h_l,p}(x) ≤ τ δN_{h_l,p}(x_l)          call optimization algorithm
6      recompute gradient-norm error estimate: δN_{h_l,p}(x)
7      if N_{h_l,p}(x) + δN_{h_l,p}(x) ≤ ϵ N_{h_0,p}(x_0) then   check for multilevel convergence
8          exit loop
9      else
10         adapt grid
11     end
12     interpolate solution (and adjoints, if necessary)
13     set x_{l+1} = x
14 end
```

B. Error Control

If the criterion (20) is not satisfied by the optimized design produced in iteration l, then we must adapt the grid to reduce the gradient-norm error. For this work we use a uniform refinement based on a simple model for the error. More sophisticated approaches based on the remaining error in the adjoint-weighted residual method[16,36] should be considered for more complex two- and three-dimensional domains.

For an SBP finite-difference discretization with a p-order accurate boundary closure, the functional will be $2p$ order accurate for the dual-consistent discretizations considered here.[27] Thus, a simple model for the error in a functional is

$$\begin{aligned} \delta \mathcal{J}_{h,p} &\equiv J_{h,p}(u_h) - \mathcal{J}(\mathcal{U}) \\ &\approx \alpha h^{2p} \end{aligned}$$

Approximating the true error, $\delta\mathcal{J}_{h,p}$, by the error estimate, $\delta J_{h,p}$, we can solve for α in the above: $\alpha = \delta J_{h,p}/h^{2p}$. Next, we compute a target mesh spacing h^*, such that the error model is below the convergence tolerance $\epsilon N_{h_0,p}(x_0)$ divided by τ:

$$h^* \equiv \left(\frac{\epsilon N_{h_0,p}(x_0)}{\alpha\tau}\right)^{\frac{1}{2p}}.$$

Dividing the desired convergence tolerance by τ helps ensure that the error estimates on the new grid remain a factor of τ smaller than the gradient norm.

Finally, h^* is used to compute the number of nodes on the subsequent optimization level:

$$n_{l+1} = \min\left(\left\lceil\frac{1}{h^*}\right\rceil + 1, 4(n_l - 1) + 1\right).$$

The above formula for n_{l+1} contains a safeguard to prevent excessive refinement caused by inaccurate error estimates on coarse grids. In the following example, this safeguard was used only at the end of level 1.

C. Results

The multilevel optimization framework is illustrated using an inverse design of a quasi-one-dimensional nozzle flow. The governing equations and discretization are the same as those used in Section III.D. The nozzle area is parameterized using 7 cubic b-spline control points. As before, the two control points at the inlet and outlet of the domain are fixed such that $A(0) = 2$ and $A(1) = 1.5$, respectively. Thus, there are 5 design variables parameterizing the interior shape of the nozzle.

The target nozzle area is the cubic function of x that passes through the inlet and outlet areas and has a local minimum at $x = 0.5$ given by $A(0.5) = 1$. This same nozzle was used in the error-estimate verification. The initial design x_0 is a nozzle with linearly varying area between the inlet and outlet.

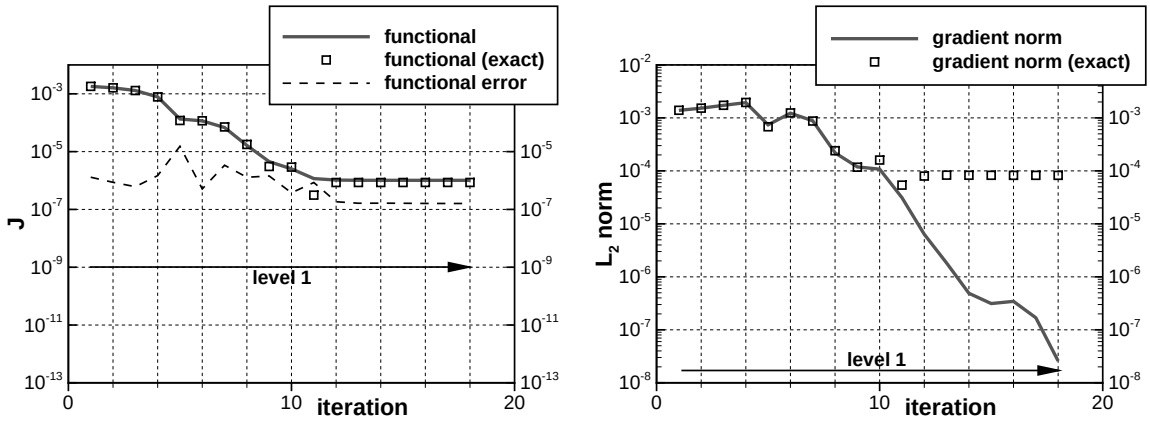
The target area is used, together with the Mach relations for the nozzle, to define the target pressure p_{targ} appearing in the inverse-design objective (19). The resulting target pressure corresponds to the analytical solution and is not achievable with a finite number of nodes. This makes the inverse design problem well-suited to verifying the optimization framework, because the objective $J_{h,p}$ should tend to zero with refinement.

For a baseline result, we perform the inverse design on a fixed grid with 9 nodes. Figures 4(a) and 4(b) show the functional and L2-norm of the gradient at each iteration of the optimization. The figures include the approximate values $J_{h,p}$ and $N_{h,p}$, computed using the $p = 1$ (second-order) discretization, as well as "exact" values, estimated using a uniformly refined grid with 4 times the number of nodes and a $p = 3$ (fourth-order) discretization. Examining the gradient-norm history, we see that the optimization is manipulating numerical errors beyond iteration 11; in other words, the computational effort expended from iteration 12 onward is wasted.

Next, we present the results from the multilevel optimization. The tolerance used to determine convergence in Algorithm 1 was set to $\epsilon = 10^{-3}$, and we set $\tau = 10$ to help keep the gradient-norm error smaller than the gradient-norm. Figures 5(a) and 5(b) plot the history of the functional and gradient norm during the multilevel process. Recall that after the level 1 optimization, subsequent optimization cycles are initiated with the previous level's optimal design; consequently, we observe that each optimization cycle requires fewer iterations than the previous. As with the fixed-grid results, the figures compare the approximate values of $J_{h,p}$ and $N_{h,p}$ with their "exact" values. In contrast with the fixed-grid results, we see that the exact values are driven toward zero, as desired.

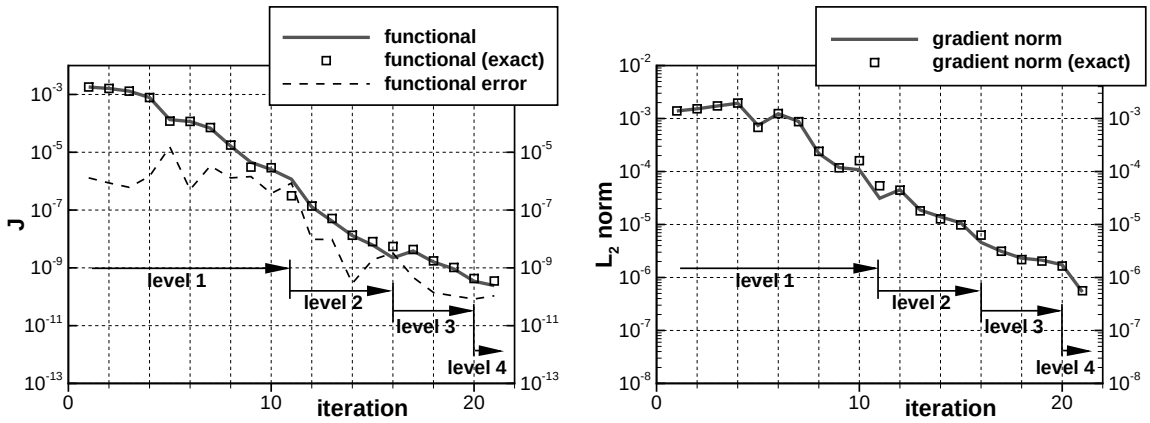
In general, an analytical solution will not be available to verify the accuracy of the gradient-norm. This brings us to the gradient-norm error estimates, plotted in Figure 6. To produce this plot, we have computed the gradient-norm error and error estimate at each optimization iteration; this was done for illustrative purposes only, and we do not advise computing the error estimates at every iteration. As the figure demonstrates, the error estimates track the true gradient-norm error well. The error estimate is least effective on the coarse grid used in level 1, which is not surprising given only 9 nodes are used with a second-order discretization.

Comparison of Figures 6 and Figure 5(b) shows that the gradient-norm error remains approximately one order of magnitude smaller than the gradient-norm, which was the target ratio defined by τ. Thus, we can be confident that the optimization process is not expending unnecessary effort manipulating numerical errors.



(a) functional history (b) gradient-norm history

Figure 4. Computed and exact functional values (left) and L2 norm of the gradient (right) at each iteration of the fixed-grid inverse design. The functional error is also included in Figure 4(a).



(a) functional history (b) gradient-norm history

Figure 5. Computed and exact functional values (left) and L2 norm of the gradient (right) at each iteration of the optimization process. The functional error is also included in Figure 5(a).

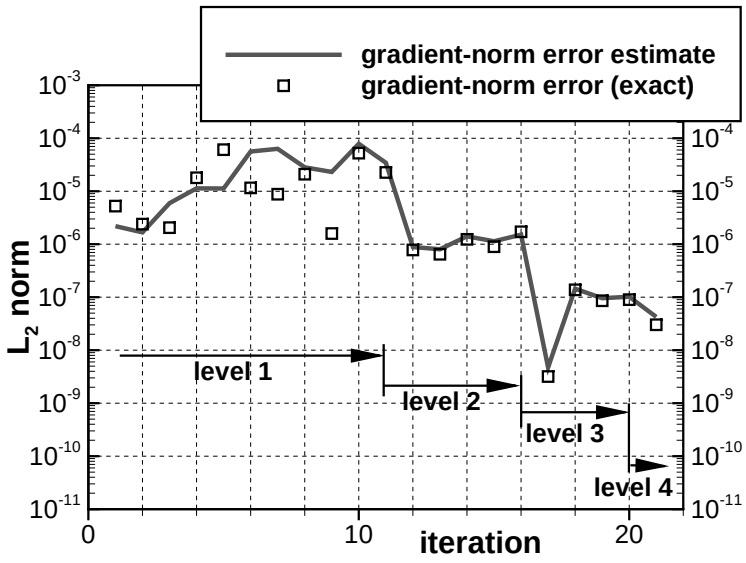


Figure 6. Gradient-norm error and error estimate at each iteration of the optimization.

At the conclusion of the optimization, the sum of the gradient-norm error and the gradient norm is below the desired tolerance of $10^{-3}N_{h_0,p}(x_0) \approx 10^{-6}$.

Finally, we examine the effectiveness of the multilevel optimization in recovering the target pressure and nozzle area. Figure 7(a) plots the pointwise error between the target and computed pressure at the end of each optimization cycle. Figure 7(b) is an analogous plot for the pointwise error between the target and computed nozzle area. Both figures demonstrate that the error in the pressure and area are reduced as the optimization progresses, albeit in an averaged rather than pointwise sense. Recall that the nozzle area at the ends of the domain nozzle is fixed, so there is no error in the area at the corresponding nodes.

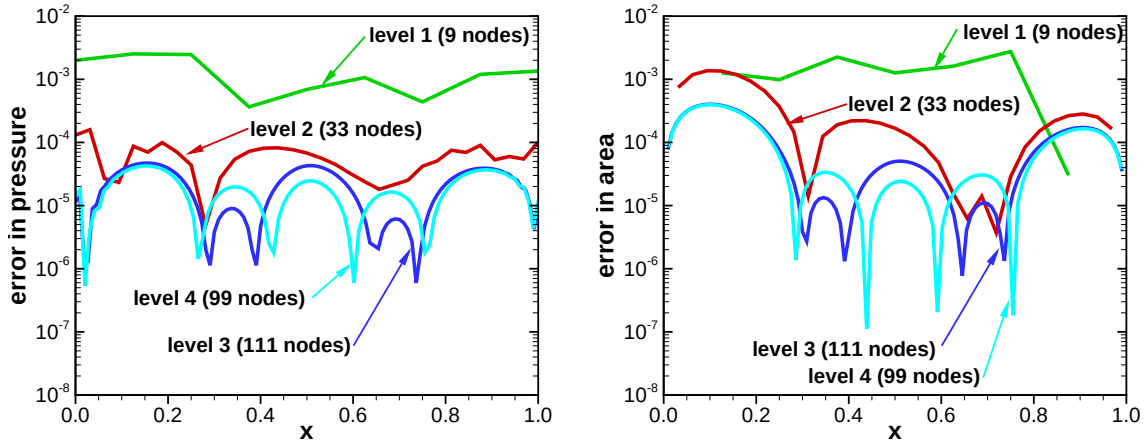
Figures 7(a) and 7(b) also indicate the number of nodes at each optimization level. We begin the optimization using a relatively coarse grid with 9 nodes (level 1). The subsequent levels contain 33 nodes (level 2), 111 nodes (level 3), and 99 nodes (level 4). Level 4 has fewer nodes than level 3, because the error estimate on level 3 (correctly) predicts that fewer nodes can achieve the desired relative tolerance of 10^{-3}.

V. Conclusions

Numerical errors must be carefully controlled in PDE-constrained optimization. For example, if numerical errors dominate the objective gradient, then the optimizer will target these errors rather than improving the performance of the design.

We have proposed error estimates for the norm of the objective gradient, which can be used to control accuracy during PDE-constrained optimization. The gradient-norm error estimates are constructed using the adjoint-weighted residual method. The objective gradient is a function of both the primal and the adjoint variables, so the error estimates require the solution of two additional dual problems. However, the system matrices for these dual problems are the Jacobian and transposed Jacobian of the primal residual. Moreover, the right-hand sides of the dual problems can be approximated using finite-differences. Consequently, most adjoint-based optimization algorithms have the infrastructure needed to solve such systems.

We verified the error estimates for both the L2- and infinity-norms of the gradient using an inverse-design objective for the quasi-one-dimensional nozzle. In addition, we proposed a multilevel optimization framework that incorporates the estimates. In this framework, the computational cost of the error estimates is limited by computing the estimates at the beginning and end of each optimization cycle only. The framework was illustrated by recovering a target area using inverse design of a quasi-one-dimensional nozzle. The estimates were shown to track the analytical gradient-norm error well; thus, the estimates helped minimize the computational cost of the optimization while controlling numerical errors.



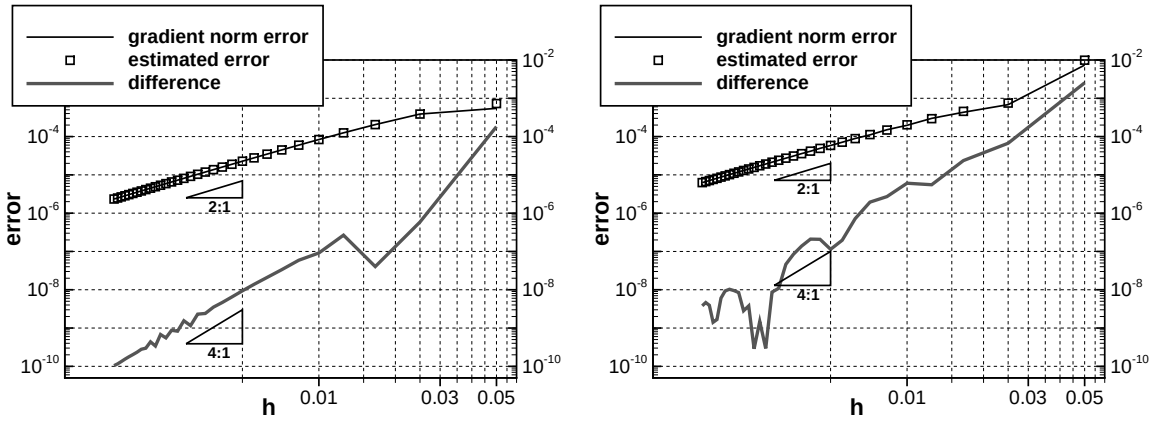
(a) error in pressure

(b) error in area

Figure 7. Pointwise error between the computed and target pressures (left) and nozzle areas (right) at the end of each optimization level.

A. Finite-difference-based Right-hand Side

Figures 8(a) and 8(b) verify that the error estimates for the L2-norm and infinity-norm (resp.) of the gradient remain reliable when the finite-difference approximations (16) and (17) are used in the adjoint system (13).



(a) Error and estimate for $\|G_{h,p}\|_2$

(b) Error and estimate for $\|G_{h,p}\|_\infty$

Figure 8. Error and error estimate in $\|G_{h,p}\|_2$ (left) and $\|G_{h,p}\|_\infty$ on a sequence of grids using the finite-difference approximations (16) and (17). Also shown is the difference between the error and error estimate.

References

[1]Jameson, A. and Reuther, J., "Control theory based airfoil design using Euler equations," *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City Beach, Sept. 1994.

[2]Vicini, A. and Quagliarella, D., "Airfoil and wing design through hybrid optimization strategies," *AIAA Journal*, Vol. 37, No. 5, May 1999, pp. 634–641.

[3]Takahashi, S., Obayashi, S., and Nakahashi, K., "Inverse design optimization of transonic wings based on multi-objective genetic algorithms," *AIAA Journal*, Vol. 37, No. 12, Dec. 1999, pp. 1656–1662.

[4]Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and multi-objective aerodynamic shape optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.

[5]Driver, J. and Zingg, D. W., "Numerical aerodynamic optimization incorporating laminar-turbulent transition prediction," *AIAA Journal*, Vol. 45, No. 8, Aug. 2007, pp. 1810–1818.

[6]Hicks, R. M. and Henne, P. A., "Wing design by numerical optimization," *Journal of Aircraft*, Vol. 15, No. 7, July 1978, pp. 407–412.

[7]Burgreen, G. W. and Baysal, O., "Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis," *AIAA Journal*, Vol. 34, No. 9, Sept. 1996, pp. 1761–1770.

[8]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1," *AIAA Journal*, Vol. 36, No. 1, Jan. 1999, pp. 51–60.

[9]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2," *AIAA Journal*, Vol. 36, No. 1, Jan. 1999, pp. 61–74.

[10]Morris, A. M., Allen, C. B., and Rendall, T. C. S., "Domain-element method for aerodynamic shape optimization applied to a modern transport wing," *AIAA Journal*, Vol. 47, No. 7, July 2009, pp. 1647–1659.

[11]Hicken, J. E. and Zingg, D. W., "Induced-drag minimization of nonplanar geometries based on the Euler equations," *AIAA Journal*, Vol. 48, No. 11, Nov. 2010, pp. 2564–2575.

[12]Venditti, D. A. and Darmofal, D. L., "Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow," *Journal of Computational Physics*, Vol. 164, No. 1, 2000, pp. 204–227.

[13]Pierce, N. A. and Giles, M. B., "Adjoint recovery of superconvergent functionals from PDE approximations," *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.

[14]Becker, R. and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, Vol. 10, 2001, pp. 1–102.

[15]Giles, M. B. and Süli, E., "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality," *Acta Numerica*, Vol. 11, 2002, pp. 145–236.

[16]Nemec, M. and Aftosmis, M. J., "Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes," *18th AIAA Computational Fluid Dynamics Conference*, No. AIAA–2007–4187, Miami, Florida, United States, June 2007.

[17]Fidkowski, K. J. and Darmofal, D. L., "Review of output-based error estimation and mesh adaptation in computational fluid dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[18]Lu, J. C.-C., *An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[19]Jameson, A., "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.

[20]Anderson, W. K. and Bonhaus, D. L., "Airfoil design on unstructured grids for turbulent flows," *AIAA Journal*, Vol. 37, No. 2, Feb. 1999, pp. 185–191.

[21]Liu, D. C. and Nocedal, J., "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, Vol. 45, 1989, pp. 503–528.

[22]Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer–Verlag, Berlin, Germany, 2nd ed., 2006.

[23]Pironneau, O., "On optimum design in fluid mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, 1974, pp. 97–110.

[24]Pierce, N. A. and Giles, M. B., "Adjoint and defect error bounding and correction for functional estimates," *Journal of Computational Physics*, Vol. 200, No. 2, 2004, pp. 769–794.

[25]Hicken, J. E., "Output error estimation for summation-by-parts finite-difference schemes," *Journal of Computational Physics*, in revision Nov 2011, pp. 25.

[26]Hartmann, R., "Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.

[27]Hicken, J. E. and Zingg, D. W., "Superconvergent functional estimates from summation-by-parts finite-difference discretizations," *SIAM Journal on Scientific Computing*, Vol. 33, No. 2, 2011, pp. 893–922.

[28]Squire, W. and Trapp, G., "Using complex variables to estimate derivatives of real functions," *SIAM Review*, Vol. 40, No. 1, 1998, pp. 110–112.

[29]Martins, J. R. R. A., Kroo, I. M., and Alonso, J. J., "An automated method for sensitivity analysis using complex variables," *The 38rd AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA–2000–0689, Reno, Nevada, 2000.

[30]Griewank, A., *Evaluating Derivatives*, SIAM, Philadelphia, PA, 2000.

[31]Funaro, D. and Gottlieb, D., "A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations," *Mathematics of Computation*, Vol. 51, No. 184, Oct. 1988, pp. 599–613.

[32]Carpenter, M. H., Gottlieb, D., and Abarbanel, S., "Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes," *Journal of Computational Physics*, Vol. 111, No. 2, 1994, pp. 220–236.

[33]Keyes, D. E., "Aerodynamic applications of Newton-Krylov-Schwarz solvers," *Proceedings of the 14th International Conference on Numerical Methods in Fluid Dynamics*, Springer, New York, 1995, pp. 1–20.

[34]Nielsen, E. J., Walters, R. W., Anderson, W. K., and Keyes, D. E., "Application of Newton-Krylov methodology to a three-dimensional unstructured Euler code," *12th AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 1995, AIAA Paper 95–1733.

[35]Saad, Y. and Schultz, M. H., "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, July 1986, pp. 856–869.

[36]Venditti, D. A. and Darmofal, D. L., "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.