View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

# Practical Aspects of Variable Reduction Formulations and Reduced Basis Algorithms in Multidisciplinary Design Optimization

Robert Michael Lewis *
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23681-0001

## Abstract

This paper discusses certain connections between nonlinear programming algorithms and the formulation of optimization problems for systems governed by state constraints. I work through the calculation of the sensitivities associated with the different formulations and present some useful relationships between them. These relationships have practical consequences; if one uses a reduced basis nonlinear programming algorithm, then the implementations for the different formulations need only differ in a single step.

* This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001 and by the State of Texas contract 1059.

**1. Introduction.** This paper discusses certain useful relationships between nonlinear programming algorithms and the formulation of optimization problems for systems governed by state constraints. The simplest instance of such a problem is

$$\text{minimize} \quad \tilde{f}(x) = f(x, u(x)), \tag{1}$$

where $f$ is a real-valued function and $u(x)$ is computed via the solution of a system of equations

$$h(x, u(x)) = 0.$$

Think of $x$ as the design variables of the problem, and of $u$ as the state variables describing the physical state of the system to be optimized. The equations $h(x, u(x)) = 0$ represent typically a system of ODE or PDE that govern the system. Given a value of the design variables $x$, we can solve these equations for the state $u(x)$. In the computational setting, the design variables $x$ and state variables $u$ will each lie in some finite-dimensional vector space.

More generally, of course, the problem could have additional constraints. However, to present the points I wish to make I will consider only the original, simpler problem (1).

I will assume that the state equations are a block system; two blocks will suffice for my purposes:

$$h(x, u_1, u_2) = \begin{pmatrix} h_1(x, u_1, u_2) \\ h_2(x, u_1, u_2) \end{pmatrix}.$$

In the context of Multidisciplinary Design Optimization, the blocks $h_i$ might describe the state equations for different disciplines, such as structures and aerodynamics, together with interdisciplinary consistency conditions, and the aggregate $h$ describes the coupled multidisciplinary analysis system. See [5] for a further discussion of this point.

I will further assume that given $x$ and $u_1$, we can solve

$$h_1(x, u_1, u_2(x, u_1)) = 0$$

for $u_2(x, u_1)$, and, vice versa, given $x$ and $u_2$, we can solve

$$h_2(x, u_1(x, u_2), u_2) = 0$$

for $u_1(x, u_2)$. This hypothesis simply reflects the independent solubility of the disciplinary analyses. Note that this hypothesis means that the partition of the state variables corresponds to that of the state equations; i.e., if there are $n_i$ equations in $h_i$, then there would be $n_i$ state variables in $u_i$, and the system solved for $u_i$ would be square.

I will consider three possible formulations of the optimization problem (1). The first is the problem in its ostensibly unconstrained form; following [5], I will refer to this as the *Multidisciplinary Feasible* (MDF) formulation:

$$\begin{array}{ll} \text{minimize} & f_{MDF}(x) \equiv f(x, u_1(x), u_2(x)) \\ \text{where} & h_1(x, u_1(x), u_2(x)) = 0 \\ & h_2(x, u_1(x), u_2(x)) = 0. \end{array} \tag{2}$$

At each iteration, we feed $x$ into the multidisciplinary analysis system and solve for the state $u$. Only the design variables $x$ are treated as independent variables. This is an example of a *variable reduction* approach, in which we use the equality constraints to define some of the variables as functions of the others, and eliminate them as independent variables.

At the other extreme, we could view the state equations purely as equality constraints, yielding the *All-at-Once* (AAO) formulation:

$$\begin{array}{ll} \text{minimize} & f_{AAO}(x, u_1, u_2) \equiv f(x, u_1, u_2) \\ \text{subject to} & h_1(x, u_1, u_2) = 0 \\ & h_2(x, u_1, u_2) = 0. \end{array} \tag{3}$$

This approach has many different names in many different fields. For instance, it is known as Simultaneous Analysis and Design (SAD or SAND) in structural optimization [11]. It is with this view of the state equations as equality constraints that "feasible" appears in the name "Multidisciplinary Feasible"; at each iteration we are feasible with respect to the state constraints. The rationale for such a formulation is that rather than spend a lot of time attaining multidisciplinary feasibility far from an optimal solution, this constrained formulation allows us the freedom to attain feasibility only at the same time as optimality.

Intermediate to these two formulations is one that, using John Dennis's apt terminology, I will call the *In-Between* formulation. In this approach we eliminate one of the blocks of state variables via the state equations (I choose to eliminate $u_2$) to arrive at the following problem:

$$\begin{array}{ll} \text{minimize} & f_{IB}(x, u_1) \equiv f(x, u_1, u_2(x, u_1)) \\ \text{subject to} & h_{IB}(x, u_1) \equiv h_1(x, u_1, u_2(x, u_1)) = 0 \\ \text{where} & h_2(x, u_1, u_2(x, u_1)) = 0. \end{array} \tag{4}$$

In the taxonomy presented in [5], the In-Between formulation corresponds to the Individual Discipline Feasible formulation. The consistency constraints of [5] correspond to $h_1$ in (4), while the individually feasible disciplines are subsumed in $h_2$. This formulation shares with the AAO formulation the advantage of additional degrees of freedom afforded by the constraints, while having perhaps substantially fewer explicit variables in the optimization problem, since we have performed a partial variable reduction by eliminating $u_2$ as an independent variable. For a further discussion of the relative advantages and disadvantages of these formulations, see [5].

The goal of this paper is to show how these three formulations, the associated sensitivities, and reduced basis methods for their solution are related. I will show how one may, in a fairly simple manner, pass between the different formulations together with algorithms for their solution. It will turn out that to apply a reduced basis algorithm to the three formulations requires, in principle, only a change to one step of the optimization algorithm. This is due to the structure of the problem introduced by the state constraints. This structure has important consequences for the design and implementation of disciplinary analysis codes if they are to be used in optimization.

The work in [7] compares these three formulations for a model large-scale parameter estimation problem and shows the potential promise of the In-Between formulation. The implementation used in that work realizes the idea presented here, and the three formulations were solved with substantially the same code, as will be discussed in §5.

In §2 I will review the notions of the reduced gradient and the reduced Hessian. In §3, I will examine the sensitivity calculations associated with the MDF and In-Between formulations. There I will show how they can be obtained from those of the AAO formulation. In §4 I will outline how a reduced basis nonlinear programming (NLP) algorithm works, and in §5 show how one may pass between reduced basis algorithms for the solution of the different formulations.

**2. The reduced gradient and Hessian.** In this section I will review the notions of the reduced gradient and the reduced Hessian. First recall the classical implicit function theorem [4].

THEOREM 2.1 (THE IMPLICIT FUNCTION THEOREM). *Suppose $\Phi$ is a mapping from an open subset of $\mathbf{R^m} \times \mathbf{R^n}$ into $\mathbf{R^n}$. Suppose $(y_0, v_0)$ is a point in $\mathbf{R^m} \times \mathbf{R^n}$ such that*

*1. $\Phi(y_0, v_0) = 0$;*
*2. $\Phi$ is continuously differentiable at $(y_0, v_0)$;*
*3. the matrix $\frac{\partial \Phi}{\partial v}(y_0, v_0)$ is invertible.*

*Then there exists a neighborhood $\Gamma$ of $y_0$ such that for each $y \in \Gamma$, the equation $\Phi(y, v) = 0$ is soluble for $v(y) \in \mathbf{R^n}$. Moreover, the derivative of this solution $v(y)$ with respect to $y$ is given by*

$$\frac{dv}{dy} = -\left(\frac{\partial \Phi}{\partial v}\right)^{-1} \frac{\partial \Phi}{\partial y}. \tag{5}$$

Formally, the latter formula is the result of implicit differentiation of $\Phi(y, v) = 0$ to obtain

$$\frac{\partial \Phi}{\partial y} + \frac{\partial \Phi}{\partial v}\frac{dv}{dy} = 0$$

and then (5).

Now, suppose that $v(y)$ is the unique solution to a state relation

$$\Phi(y, v(y)) = 0 \tag{6}$$

where $\Phi$ is twice continuously differentiable and $\partial\Phi/\partial v$ is invertible. In practice, the validity of these hypotheses typically follows from the existence and uniqueness theory for the solution of the equation represented by (6). With these hypotheses, I may apply the Implicit Function Theorem to compute sensitivities. Before doing so, let me first define the matrix

$$W = W(y, v) = \begin{pmatrix} I \\ -\frac{\partial \Phi}{\partial v}^{-1}(y, v)\frac{\partial \Phi}{\partial y}(y, v) \end{pmatrix}.$$

I will call $W$ the *injection operator* associated with $\Phi$ since it is a one-to-one mapping from $\mathbf{R^m}$ into $\mathbf{R^m} \times \mathbf{R^n}$ and is invertible on its range (i.e., $W$ has full column rank). Its transpose $W^T$ I will call the *reduction operator.*

Let

$$N = \frac{\partial \Phi}{\partial y} \qquad \text{and} \qquad B = \frac{\partial \Phi}{\partial v}.$$

We call $B$ the *basic block* since, by assumption, $B = \partial\Phi/\partial v$ is invertible and hence is a basis of $\mathbf{R^n}$. The associated state variables $v$ are the *basic variables.* The remaining design variables, $y$, are the *non-basic variables*, and $N$ the non-basic block. In terms of the basic and non-basic blocks,

$$W = \begin{pmatrix} I \\ -B^{-1}N \end{pmatrix}.$$

Note that $W$ is a basis for the nullspace of the linearized state constraint operator:

$$\nabla \Phi^T W = \begin{pmatrix} N & B \end{pmatrix} \begin{pmatrix} I \\ -B^{-1}N \end{pmatrix} = 0. \tag{7}$$

I will return to this point in §4 and §5.

Finally, given $F : \mathbf{R^m} \times \mathbf{R^n} \to \mathbf{R}$, define $\lambda$ by

$$\lambda = \lambda(y, v) = -\left(\frac{\partial \Phi}{\partial v}\right)^{-T} (y, v) \nabla_v F(y, v) \tag{8}$$

and the Lagrangian $\ell(y, v; \lambda)$ by

$$\ell(y, v; \lambda) = F(y, v) + \lambda^T \Phi(y, v).$$

The quantity $\lambda$ is a Lagrange multiplier estimate arising from variable reduction [10]. For those familiar with the language of "the adjoint method" in control, $\lambda$ is the *costate* or *adjoint state.*

PROPOSITION 2.2. *Let $\tilde{F}(y) = F(y, v(y))$, where $v(y)$ is computed via (6), and $F : \mathbf{R^m} \times \mathbf{R^n} \to \mathbf{R^q}$. Then*

$$\nabla_y \tilde{F}^T(y) = \nabla_{(y,v)} F^T(y, v(y)) W(y, v(y)). \tag{9}$$

*If $F$ is a real-valued function, then*

$$\nabla_y^2 \tilde{F}(y) = W^T(y, v(y)) \left( \nabla^2_{(y,v)} F(y, v(y)) + \nabla^2_{(y,v)} \Phi(y, v(y)) \lambda(y, v(y)) \right) W(y, v(y)). \tag{10}$$

By $\nabla^2_{(y,v)} \Phi \lambda$ I mean

$$\nabla^2_{(y,v)} \Phi \lambda = \sum_{i=1}^{n} \lambda_i \nabla^2_{(y,v)} \Phi_i.$$

In the case of a real-valued $F$, transposition of (9) tells us that $\nabla \tilde{F}$ is given by

$$\nabla_y \tilde{F}(y) = W^T(y, v(y)) \nabla_{(y,v)} F(y, v(y)). \tag{11}$$

The quantity appearing on the right-hand side of (11) is known as the *reduced gradient* [10], since it is the product of the reduction operator $W^T$ with $\nabla F$. Similarly, the quantity on the right-hand side of (10) is called the *reduced Hessian* of the Lagrangian.

*Proof.* Computing the Jacobian of $\tilde{F}$, we see that

$$\frac{d\tilde{F}}{dy}(y) = \frac{\partial F}{\partial y}(y, v(y)) + \frac{\partial F}{\partial v}(y, v(y)) \frac{dv}{dy}(y).$$

This and the Implicit Function Theorem then yield

$$\begin{aligned}
\frac{d\tilde{F}}{dy}(y) &= \frac{\partial F}{\partial y}(y,v(y)) - \frac{\partial F}{\partial v}(y,v(y))\left(\frac{\partial \Phi}{\partial v}\right)^{-1}(y,v(y))\frac{\partial \Phi}{\partial y}(y,v(y)) \\
&= \begin{pmatrix} \frac{\partial F}{\partial y}(y,v(y)) & \frac{\partial F}{\partial v}(y,v(y)) \end{pmatrix} \begin{pmatrix} I \\ -\left(\frac{\partial \Phi}{\partial v}\right)^{-1}(y,v(y))\frac{\partial \Phi}{\partial y}(y,v(y)) \end{pmatrix}.
\end{aligned}$$

which is (9).

Next, the Hessian for real-valued $F$. We have

$$\frac{d^2\tilde{F}}{dy^2} = \frac{d}{dy}\left[F_y(y,v(y)) + F_v(y,v(y))\frac{dv}{dy}\right]$$

or

$$\frac{d^2\tilde{F}}{dy^2} = \frac{\partial F_y}{\partial y} + \frac{dv}{dy}^T\frac{\partial F_y}{\partial v} + \frac{\partial F_v}{\partial y}\frac{dv}{dy} + \frac{dv}{dy}^T\frac{\partial F_v}{\partial v}\frac{dv}{dy} + F_v\frac{d^2v}{dy^2}.$$

We can rewrite this in terms of the reduction and injection matrices as

$$\begin{aligned}
\frac{d^2\tilde{F}}{dx^2} &= \begin{pmatrix} I & \frac{dv}{dy}^T \end{pmatrix} \begin{pmatrix} \frac{\partial F_y}{\partial y} & \frac{\partial F_v}{\partial y} \\ \frac{\partial F_y}{\partial v} & \frac{\partial F_v}{\partial v} \end{pmatrix} \begin{pmatrix} I \\ \frac{dv}{dy} \end{pmatrix} + F_v\frac{d^2v}{dy^2} \\
&= W^T\left(\nabla^2_{(y,v)}F\right)W + F_v\frac{d^2v}{dy^2}.
\end{aligned}$$

Repeated implicit differentiation of $\Phi(y,v(y)) = 0$ yields

$$\frac{d}{dy}\left(\Phi_y(y,v(y)) + \Phi_v(y,v(y))\frac{dv}{dy}\right) = 0,$$

whence

$$\frac{d^2v}{dy^2} = -\left(\frac{\partial \Phi}{\partial v}\right)^{-1}\left(\frac{\partial \Phi_y}{\partial y} + \frac{dv}{dy}^T\frac{\partial \Phi_y}{\partial v} + \frac{\partial \Phi_v}{\partial y}\frac{dv}{dy} + \frac{dv}{dy}^T\frac{\partial \Phi_v}{\partial v}\frac{dv}{dy}\right).$$

Thus,

$$\begin{aligned}
F_v\frac{d^2v}{dy^2} &= -\frac{\partial F}{\partial v}\left(\frac{\partial h}{\partial u}\right)^{-1}\left(\frac{\partial \Phi_y}{\partial y} + \frac{dv}{dy}^T\frac{\partial \Phi_y}{\partial v} + \frac{\partial \Phi_v}{\partial y}\frac{dv}{dy} + \frac{dv}{dy}^T\frac{\partial \Phi_v}{\partial v}\frac{dv}{dy}\right) \\
&= \lambda^T\left(\frac{\partial \Phi_y}{\partial y} + \frac{dv}{dy}^T\frac{\partial \Phi_y}{\partial v} + \frac{\partial \Phi_v}{\partial y}\frac{dv}{dy} + \frac{dv}{dy}^T\frac{\partial \Phi_v}{\partial v}\frac{dv}{dy}\right) \\
&= W^T\left(\nabla^2_{(y,v)}\Phi\lambda\right)W.
\end{aligned}$$

Thus we finally arrive at

$$\frac{d^2\tilde{F}}{dy^2}(y) = W^T\left(\nabla^2_{(y,v)}F + \nabla^2_{(y,v)}h\lambda\right)W = W^T\left(\nabla^2_{(y,v)}\ell(y,v(y);\lambda(y,v(y))\right)W,$$

which is (10). $\square$

**3. Sensitivities for the three formulations.** In this section I will derive some formulae relating the sensitivities of the MDF, In-Between, and All-at-Once formulations. You will see that the sensitivities are all derived from those of the AAO formulation in simple and interesting ways.

Proposition 2.2 says that the gradient and Hessian associated with the MDF formulation are the reduced gradient and reduced Hessian of the Lagrangian associated with the AAO formulation, as we will now discuss. Let $B_{AAO}$ and $N_{AAO}$ denote the basic and non-basic blocks associated with the AAO formulation,

$$B_{AAO} = \begin{pmatrix} \dfrac{\partial h_1}{\partial u_1} & \dfrac{\partial h_1}{\partial u_2} \\ \dfrac{\partial h_2}{\partial u_1} & \dfrac{\partial h_2}{\partial u_2} \end{pmatrix} \qquad N_{AAO} = \begin{pmatrix} \dfrac{\partial h_1}{\partial x} \\ \dfrac{\partial h_2}{\partial x} \end{pmatrix},$$

let $W_{AAO}$ denote the reduction matrix associated with the AAO formulation,

$$W_{AAO} = \begin{pmatrix} I_x \\ -B_{AAO}^{-1} N_{AAO} \end{pmatrix},$$

let $\lambda_{AAO}$ denote the Lagrange multiplier associated with the AAO formulation,

$$\lambda_{AAO} = -B_{AAO}^{-T} \nabla_u f_{AAO},$$

and let $\ell_{AAO}$ denote the Lagrangian associated with the AAO formulation,

$$\ell_{AAO}(x, u; \lambda) = f_{AAO}(x, u) + \lambda^T h(x, u).$$

Then we have this corollary of Proposition 2.2.

Corollary 3.1. *Suppose that $(x, u)$ is feasible with respect to the multidisciplinary analysis constraints: $h(x, u) = 0$. Then the gradients and Hessians at $(x, u)$ of the MDF and AAO formulations are related by*

$$\nabla_x f_{MDF}(x) = W_{AAO}^T(x, u) \nabla_{(x,u)} f_{AAO}(x, u) \tag{12}$$

$$= W_{AAO}^T(x, u) \nabla_{(x,u)} \ell_{AAO}(x, u; \lambda_{AAO}(x, u)) \tag{13}$$

*and*

$$\nabla_x^2 f_{MDF}(x) = W^T(x, u) \left( \nabla_{(x,u)}^2 \ell_{AAO}(x, u; \lambda_{AAO}(x, u)) \right) W_{AAO}(x, u). \tag{14}$$

*Proof.* The relations (12) and (14) follow immediately from Proposition 2.2. The equation (13) is a consequence of the fact that $W_{AAO}^T \nabla h = 0$. □

Next we turn attention to the sensitivities associated with the In-Between formulation (4). We will denote by $S$ the Schur complement

$$S = \frac{\partial h_1}{\partial u_1} - \frac{\partial h_1}{\partial u_2} \left( \frac{\partial h_2}{\partial u_2} \right)^{-1} \frac{\partial h_2}{\partial u_1}. \tag{15}$$

This is the Schur complement of the basic block $B_{AAO}$ upon elimination of the variable $u_2$. You can easily check that

$$B_{AAO}^{-1} =$$

$$\begin{pmatrix} S^{-1} & -S^{-1} \dfrac{\partial h_1}{\partial u_2} \left( \dfrac{\partial h_2}{\partial u_2} \right)^{-1} \\ -\left( \dfrac{\partial h_2}{\partial u_2} \right)^{-1} \dfrac{\partial h_2}{\partial u_1} S^{-1} & \left( \dfrac{\partial h_2}{\partial u_2} \right)^{-1} + \left( \dfrac{\partial h_2}{\partial u_2} \right)^{-1} \dfrac{\partial h_2}{\partial u_1} S^{-1} \dfrac{\partial h_1}{\partial u_2} \left( \dfrac{\partial h_2}{\partial u_2} \right)^{-1} \end{pmatrix} \tag{16}$$

I will also denote by $W_2$ the reduction matrix associated with the elimination of $u_2$:

$$W_2 = \begin{pmatrix} I_x & 0 \\ 0 & I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial x} & -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial u_2} \end{pmatrix} \tag{17}$$

The next result concerns the basic and non-basic blocks for the In-Between formulation. Equations (15) and (19) mean that the basic block associated with the In-Between formulation (4) is the Schur complement that results if $u_2$ is eliminated from the basic block of the All-at-Once formulation.

PROPOSITION 3.2. *The nonbasic block associated with the In-Between formulation (4) is*

$$\frac{\partial h_{IB}}{\partial x} = \frac{\partial h_1}{\partial x} - \frac{\partial h_1}{\partial u_2}\left(\frac{\partial h_2}{\partial u_2}\right)^{-1}\frac{\partial h_2}{\partial x}. \tag{18}$$

*The basic block associated with the In-Between formulation (4) is*

$$\frac{\partial h_{IB}}{\partial u_1} = \frac{\partial h_1}{\partial u_1} - \frac{\partial h_1}{\partial u_2}\left(\frac{\partial h_2}{\partial u_2}\right)^{-1}\frac{\partial h_2}{\partial u_1}. \tag{19}$$

*Proof.* Applying Proposition 2.2 with $y = (x, u_1)$, $v = u_2$, and $\tilde{F}(x, u_1) = h_{IB}(x, u_1, u_2(x, u_1))$ yields

$$\begin{aligned} \begin{pmatrix} \dfrac{\partial h_{IB}}{\partial x} & \dfrac{\partial h_{IB}}{\partial u_1} \end{pmatrix} &= \begin{pmatrix} \dfrac{\partial h_1}{\partial x} & \dfrac{\partial h_1}{\partial u_1} & \dfrac{\partial h_1}{\partial u_2} \end{pmatrix} W_2 \\ &= \begin{pmatrix} \dfrac{\partial h_1}{\partial x} & \dfrac{\partial h_1}{\partial u_1} & \dfrac{\partial h_1}{\partial u_2} \end{pmatrix} \begin{pmatrix} I_x & 0 \\ 0 & I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial x} & -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial u_1} \end{pmatrix}, \end{aligned}$$

from which we obtain (18) and (19). ☐

Let

$$W_{IB} = \begin{pmatrix} I_x \\ -\left(\dfrac{\partial h_{IB}}{\partial u_1}\right)^{-1} \dfrac{\partial h_{IB}}{\partial x} \end{pmatrix} \tag{20}$$

denote the reduction matrix associated with eliminating $u_1$ from the In-Between formulation. Let

$$\lambda_{IB} = -\left(\frac{\partial h_{IB}}{\partial u_1}\right)^{-T} \nabla_{u_1} f_{IB} \tag{21}$$

denote the Lagrange multiplier associated with the In-Between formulation. In the In-Between formulation, we eliminate $u_2$ from the AAO formulation as an explicit variable; the following proposition says that the the multiplier $\lambda_{IB}$ can be obtained from the AAO multiplier by dropping the component corresponding to the constraints that define $u_2$.

PROPOSITION 3.3. *At a point $(x, u_1)$ feasible with respect to $h_{IB}$, i.e., satisfying $h_{IB}(x, u_1) = 0$ (and* a fortiori*, $h_2(x, u_1, u_2(x, u_1) = 0)$, we have*

$$\lambda_{IB}(x, u_1) = (I_{u_1} \quad 0) \quad \lambda_{AAO}(x, u_1, u_2). \tag{22}$$

*Proof.* Applying Proposition 2.2 to compute $\nabla_{(x,u_1)} f_{IB}(x, u_1)$ we have

$$\begin{aligned}
\lambda_{IB} &= -S^{-T} \nabla_{u_1} f_{IB}(x, u_1) \\
&= -S^{-T} \left(0 \quad I_{u_1}\right) \nabla_{(x,u_1)} f_{IB}(x, u_1) \\
&= -S^{-T} \left(0 \quad I_{u_1}\right) W_2^T \nabla_{(x,u_1,u_2)} f_{AAO}(x, u_1, u_2).
\end{aligned}$$

Writing this out leads to

$$\begin{aligned}
\lambda_{IB} &= -S^{-T} \left(0 \quad I_{u_1}\right) \begin{pmatrix} I_x & 0 & -\dfrac{\partial h_2}{\partial x}^T \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-T} \\ 0 & I_{u_1} & -\dfrac{\partial h_2}{\partial u_1}^T \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-T} \end{pmatrix} \nabla_{(x,u_1,u_2)} f_{AAO}(x, u_1, u_2) \\
&= -\left(0 \quad S^{-T} \quad -S^{-T} \dfrac{\partial h_2}{\partial u_1}^T \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-T}\right) \nabla_{(x,u_1,u_2)} f_{AAO}(x, u_1, u_2) \\
&= -\left(I_{u_1} \quad 0\right) B_{AAO}^{-T} \begin{pmatrix} 0 & I_{u_1} & 0 \\ 0 & 0 & I_{u_2} \end{pmatrix} \nabla_{(x,u_1,u_2)} f_{AAO}(x, u_1, u_2) \\
&= -\left(I_{u_1} \quad 0\right) B_{AAO}^{-T} \nabla_u f_{AAO}(x, u_1, u_2) \\
&= -\left(I_{u_1} \quad 0\right) \quad \lambda_{AAO}.
\end{aligned}$$

□

We have the following factorization of the All-at-Once reduction operator in terms of the In-Between reduction operator and the reduction operator associated with the elimination of $u_2$ as an explicit variable.

PROPOSITION 3.4. *Suppose that $(x, u_1)$ is feasible with respect to $h_{IB}$, as in Proposition 3.3. Then*

$$W_{AAO} = W_2 W_{IB}. \tag{23}$$

*Proof.* The proof is another icky calculation. We have

$$W_2 = \begin{pmatrix} I_x \\ -S^{-1} \left( \dfrac{\partial h_1}{\partial x} - \dfrac{\partial h_1}{\partial u_2} \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial x} \right) \end{pmatrix}$$

and

$$W_{IB} = \begin{pmatrix} I_x & 0 \\ 0 & I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial x} & -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \dfrac{\partial h_2}{\partial u_2} \end{pmatrix},$$

so

$$W_2 W_{IB} = \begin{pmatrix} I_x \\ -S^{-1}\left(\dfrac{\partial h_1}{\partial x} - \dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x}\right) \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x} + \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x}S^{-1}\left(\dfrac{\partial h_1}{\partial x} - \dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x}\right) \end{pmatrix}$$

But from (16) we see that the last two rows of this product are none other than

$$-B_{AAO}^{-1}N_{AAO} =$$

$$-\begin{pmatrix} S^{-1} & -S^{-1}\dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1}S^{-1} & \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} + \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1}S^{-1}\dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \end{pmatrix}\begin{pmatrix} \dfrac{\partial h_1}{\partial x} \\ \dfrac{\partial h_2}{\partial x} \end{pmatrix}$$

Thus $W_{AAO} = W_2 W_{IB}$. □

The next proposition says that the reduced gradient and reduced Hessian for the In-Between and All-at-Once formulations are the same at appropriately feasible points.

Proposition 3.5. *Suppose that $(x, u_1)$ is feasible with respect to $h_{IB}$, as in Proposition 3.3. Then*

$$W_{IB}^T \nabla_{(x,u_1)} f_{IB}(x, u_1) = W_{AAO}^T \nabla_{(x,u_1,u_2)} f_{AAO}(x, u_1, u_2), \tag{24}$$

$$W_{IB}^T \nabla_{(x,u_1)} \ell_{IB}(x, u_1; \lambda_{IB}) = W_{AAO}^T \nabla_{(x,u_1,u_2)} \ell_{AAO}(x, u_1, u_2; \lambda_{AAO}), \tag{25}$$

$$\nabla^2_{(x,u_1)} \ell_{IB}(x, u_1; \lambda_{IB}) = W_2^T \left(\nabla^2_{(x,u_1,u_2)} \ell_{AAO}(x, u_1, u_2; \lambda_{AAO})\right) W_2, \tag{26}$$

*and*

$$W_{IB}^T \nabla^2_{(x,u_1)} \ell_{IB}(x, u_1; \lambda_{IB}) W_{IB} = W_{AAO}^T \nabla^2_{(x,u_1,u_2)} \ell_{AAO}(x, u_1, u_2; \lambda_{AAO}) W_{AAO}, \tag{27}$$

*where $\lambda_{IB} = \lambda_{IB}(x, u_1)$ and $\lambda_{AAO} = \lambda_{AAO}(x, u_1, u_2)$.*

*Proof.* From Proposition 2.2 we have

$$\nabla_{(x,u_1)} f_{IB} = W_2^T \nabla_{(x,u_1,u_2)} f_{AAO};$$

this combined with Proposition 3.4 yields (24). Equation (25) then follows since $W_{IB}^T \nabla h_{IB} = 0$ and $W_{AAO}^T \nabla h = 0$.

Applying Proposition 2.2 to $\tilde{F} = \ell_{IB}$ we obtain

$$\nabla^2_{(x,u_1)} \ell_{IB} = W_2^T \left(\nabla^2_{(x,u_1,u_2)} f_{AAO} + \nabla^2_{(x,u_1,u_2)} h_1 \lambda_{IB} + \nabla^2_{(x,u_1,u_2)} h_2 \lambda_2\right) W_2$$

where $\lambda_2$ is given by

$$\lambda_2 = -\left(\frac{\partial h_2}{\partial u_2}\right)^{-T} \begin{pmatrix} 0 & I_{u_2} \end{pmatrix} \left(\nabla_{(x,u_1,u_2)} f_{AAO} + \nabla_{(x,u_1,u_2)} h_{IB} \lambda_{IB}\right).$$

Using (16) and (21), we obtain

$$\begin{aligned}\lambda_2 &= -\left(\frac{\partial h_2}{\partial u_2}\right)^{-T}\left(\frac{\partial f}{\partial u_2}^T + \frac{\partial h_1}{\partial u_2}^T\left(-S^{-T}\left(\frac{\partial f}{\partial u_1}^T - \frac{\partial h_2}{\partial u_1}^T\left(\frac{\partial h_2}{\partial u_2}\right)^{-T}\frac{\partial f}{\partial u_2}^T\right)\right)\right)\\ &= (0 \quad I_{u_2})\left(-B_{AAO}^{-T}\nabla_u f_{AAO}\right)\\ &= (0 \quad I_{u_2})\,\lambda_{AAO}.\end{aligned}$$

At this point the component of $\lambda_{AAO}$ corresponding to the block $h_2$ reappears.

Moreover, (22) tells us that $\lambda_{IB} = (I_{u_1} \quad 0)\,\lambda_{AAO}$. Thus,

$$\nabla^2_{(x,u_1)}\ell_{IB} = W_2^T\left(\nabla^2_{(x,u_1,u_2)} f_{AAO} + \nabla^2_{(x,u_1,u_2)} h\lambda_{AAO}\right)W_2,$$

which is (26). Moreover, by Proposition 3.4,

$$\begin{aligned}W_{IB}^T\nabla^2_{(x,u_1)}\ell_{IB}W_{IB} &= W_{IB}^T W_2^T\left(\nabla^2_{(x,u_1,u_2)} f_{AAO} + \nabla^2_{(x,u_1,u_2)} h\lambda_{AAO}\right)W_2 W_{IB}\\ &= W_{AAO}^T\nabla^2_{(x,u_1,u_2)}\ell_{AAO}W_{AAO}.\end{aligned}$$

□

**4. Reduced basis algorithms.** In this section I will discuss a model Sequential Quadratic Programming (SQP) subproblem and *reduced basis* methods for its solution. SQP subproblems similar to this one are at the heart of many equality constrained NLP algorithms.

The fundamental SQP subproblem for (3) is

$$\begin{array}{ll}\text{minimize} & \frac{1}{2}s^T H s + g^T s\\ \text{subject to} & \nabla h^T s + h = 0,\end{array}\tag{28}$$

where $H$ is an approximation to the Hessian of the Lagrangian and $g$ is the gradient of the Lagrangian. For a trust-region algorithm one would have, in addition, some manner of trust-region constraint, but for now I will ignore this detail. This subproblem may be stated in many equivalent ways; for a discussion, see [15].

Now, if one had a point $s^{LF}$ that was feasible with respect to the linearized constraints, and a basis $Z$ for the nullspace of the Jacobian of the constraints, then one could express $s$ as

$$s = s^{LF} + Z\sigma$$

and the SQP subproblem could be reduced to an unconstrained problem:

$$\text{minimize} \quad \frac{1}{2}\sigma^T Z^T H Z\sigma + (g + Hs^{LF})^T Z^T\sigma.\tag{29}$$

In a *reduced basis* approach to an equality constrained optimization problem we take advantage of a certain form of basis for the nullspace of the linearized constraints [10]. At each iteration of the optimization algorithm, we seek a partition of the Jacobian of the constraints of the form

$$\nabla h^T = (N \quad B)$$

where $B$ is invertible; then

$$Z = \begin{pmatrix} I \\ -B^{-1}N \end{pmatrix}$$

is a basis for the nullspace of $\nabla h^T$.

But we have already seen an example of such a nullspace basis at work. In the case of state constraints, we have an immediate and fixed choice for the basis matrix $B$, to wit, the linearized state operator. This is the import of (7). Moreover, the invertibility of $B$ allows us to obtain a linearly feasible point $s^{LF}$; we may simply take

$$s^{LF} = \begin{pmatrix} 0 \\ -B^{-1}h \end{pmatrix}. \tag{30}$$

This corresponds to solving the linearized state equation, or, if one prefers, taking one unglobalized step of Newton's method towards the solution of the state equation.

In terms of $W$, then, the model SQP subproblem (28) could be transformed to the unconstrained problem

$$\text{minimize} \quad \tfrac{1}{2}\sigma^T W^T H W \sigma + (g + H s^{LF})^T W^T \sigma.$$

One would solve this problem for the reduced step $\sigma$—a step in the design variables alone—and then take the step $s = W\sigma$ in all the degrees of freedom in the optimization problem, both the design variables and those state variables kept explicit.

Note that this is not only a convenient reduction, but in the context of MDO this is almost surely a necessary step. The SQP subproblem involves attaining feasibility with respect to the linearized state constraints. Therefore the SQP subproblem is necessarily more difficult than solving the linearized multidisciplinary analysis problem, which, for problems involving PDE, may be quite difficult in itself. In order to solve the optimization subproblem we *must* exploit the structure of the problem—in this case, its relation to the state constraints—and avail ourselves of the pieces available from the solution of the analysis problem.

**5. A simple way of implementing reduced basis algorithms for the different formulations.** In this section I will show how the relations between the different formulations in §3 mean that, in principle, it is possible to implement an optimization algorithm for the AAO formulation that with a single modification becomes an algorithm for either the MDF or In-Between formulations. I have actually done such an implementation for the work described in [7].

The key to such an implementation lies in the relations that were presented in §3. To make matters concrete, I will discuss the following trust-region algorithm for large-scale equality constrained optimization problems. This algorithm was used for the work in [7] and is a relative of the algorithms described in [3] and [6]. I use the subscript and superscript "$c$" to denote quantities associated with the current iterate, and the subscript and superscript "+" to denote quantities associated with the putative next iterate, following [8].

At each iteration, we approximately solve the subproblem

$$\begin{array}{ll} \text{minimize} & \tfrac{1}{2}s^T H s + g^T s \\ \text{subject to} & \| \nabla h^T s + h_c \| \le \theta \\ & \| s \| \le \delta \end{array} \tag{31}$$

where $H$ is an approximation to the Hessian of the Lagrangian, $g$ is the gradient of the Lagrangian, $\delta$ is the current trust radius, and $\theta$ is chosen to enforce Fraction of Cauchy Decrease on $\| \nabla h^T s + h_c \|^2$ over a trust-region of radius $r\delta$, where $0 < r < 1$ is fixed.

The approximate solution of the subproblem is effected in two stages. The first is to take a step in the basic variables to improve linear feasibility, rather as in (30):

$$s^{LF} = \alpha d, \tag{32}$$

where

$$d = \begin{pmatrix} 0 \\ -\left(\dfrac{\partial h}{\partial u}\right)^{-1} h_c \end{pmatrix}.$$

The quantity $\alpha$ is chosen to solve

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \parallel \alpha \nabla h^T d + h_c \parallel^2 \\ \text{subject to} & \parallel \alpha d \parallel \leq r\delta. \end{array}$$

The second stage is then to improve optimality subject to the constraint of not degrading the improved linear feasibility achieved by $s^{LF}$. Let

$$\Theta = \nabla h^T s^{LF} + h_c.$$

The subproblem we solve to improve optimality subject to improved linear feasibility is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} s^T H s + g^T s \\ \text{subject to} & \nabla h^T s + h_c = \Theta \\ & \parallel s \parallel \leq \delta. \end{array} \tag{33}$$

This we do by using the reduced basis and writing the desired step as

$$s = s^{LF} + W\sigma$$

and solving the trust-region problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T W^T H W \sigma + (g + H s^{LF})^T W^T \sigma \\ \text{subject to} & \parallel s^{LF} + W\sigma \parallel \leq \delta. \end{array} \tag{34}$$

We then test the aggregate step $s$ using the augmented Lagrangian

$$L(y; \lambda, \rho) = f(y) + \lambda^T h(y) + \frac{\rho}{2} \parallel h(y) \parallel^2,$$

as the merit function with the penalty weight $\rho$ updated as described in [9]. Figure 1 contains a rough outline of the algorithm for the AAO formulation.

Now suppose that I modify this algorithm by adding an analysis step in which given design variables $x_c$, I solve the multidisciplinary analysis problem $h(x_c, u_c) = 0$. This single modification, illustrated in Figure 2, yields a trust region algorithm for the MDF formulation.

THEOREM 5.1. *The modified algorithm in Figure 2 is equivalent to a trust region algorithm for the MDF formulation.*

*Proof.* I will check the following:

1. At every step, we are feasible with respect to the constraint $h(x_c, u_c) = 0$. This is, of course, true by design.

Initialization: Choose an initial $(x_c, u_c)$.
Until convergence, do {

1. Compute the multiplier

$$\lambda_{AAO} = -\left(\frac{\partial h}{\partial u}\right)^{-1} \nabla_u f_{AAO}.$$

2. Test for convergence.
3. Construct a local model of $\ell_{AAO}$ about $(x_c, u_c)$.
4. Take a step $s^{LF}$ to improve linear feasibility:

$$s^{LF} = \alpha \begin{pmatrix} 0 \\ -\left(\frac{\partial h}{\partial u}\right)^{-1} h_c \end{pmatrix}.$$

5. Subject to the improved linear feasibility, take a step to improve optimality:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T W_{AAO}^T H_{AAO} W_{AAO}\sigma + (g_{AAO} + H_{AAO}s^{LF})^T W_{AAO}^T \sigma \\ \text{subject to} & \| \, s^{LF} + W_{AAO}\sigma \, \| \le \delta. \end{array}$$

6. Set $s = (s_x, s_u) = s^{LF} + W_{AAO}\sigma$.
7. Evaluate $(x_+, u_+) = (x_c, u_c) + (s_x, s_u)$.
8. Update $(x_c, u_c)$, $\delta$, &c.

}

FIG. 1. *Reduced basis algorithm for the AAO formulation.*

2. The subproblems we solve in the context of the AAO formulation correspond to a trust region subproblem for the MDF formulation.
3. The merit function we use to evaluate the step taken for the AAO formulation corresponds to the correct merit function for the MDF formulation.

Because we solve the analysis problem $h(x_c, u_c) = 0$ at each iteration, $h_c = 0$ and so $s^{LF} = 0$. The subproblem we solve to improve optimality then becomes

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T W_{AAO}^T H_{AAO} W_{AAO}\sigma + g_{AAO}^T W_{AAO}^T \sigma \\ \text{subject to} & \| \, W_{AAO}\sigma \, \| \le \delta. \end{array}$$

However, in light of Corollary 3.1, this is none other than

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T \nabla_x^2 f_{MDF}\sigma + \nabla_x f_{MDF}^T \sigma \\ \text{subject to} & \| \, W_{AAO}\sigma \, \| \le \delta. \end{array}$$

This we recognize as a trust-region problem for the MDF formulation, which establishes Point 2. Note, however, the non-standard scaling of the trust region constraint. As discussed in [14], this scaling is quite natural since it measures the effect of changes in the design variables on the state variables.

Moreover, Point 3 holds. Because $h(x, u) = 0$ at each step, the augmented Lagrangian merit function reduces to

$$f_{AAO}(x, u(x)) + \lambda_{AAO}^T h(x, u(x)) + \frac{\rho}{2} \| \, h(x, u(x)) \, \|^2 = f_{AAO}(x, u(x)) = f_{MDF}(x),$$

Initialization: Choose an initial $x_c$.
Until convergence, do {
0. **Analysis: Solve** $h(x, u(x)) = 0$ **for the state** $u(x)$.
1. Test for convergence.

$\vdots$

7. **Analysis: Solve** $h(x_+, u_+) = 0$ **for** $u_+(x_+)$ **and evaluate** $(x_+, u_+)$.

$\vdots$

}

FIG. 2. *Reduced basis algorithm for the AAO formulation with a multidisciplinary analysis step.*

Initialization: Choose an initial $(x^c, u_1^c)$.
Until convergence, do {
0. **Analysis: Solve** $h_2(x, u_1, u_2(x, u_1)) = 0$ **for** $u_2(x, u_1)$.
1. Test for convergence.

$\vdots$

7. **Analysis: Solve** $h_1(x^+, u_1^+, u_2^+) = 0$ **for** $u_2^+(x_+, u_1^+)$ **and evaluate** $(x_+, u_+)$.

$\vdots$

}

FIG. 3. *Reduced basis algorithm for the AAO formulation with an analysis step that eliminates* $u_2$.

which is the merit function for the unconstrained MDF formulation. Thus we have a trust region algorithm for the MDF formulation. □

Thus, simply by the addition of an analysis step the reduced basis algorithm for the AAO formulation becomes a reduced basis algorithm for the MDF formulation, without any re-implementation of sensitivities or other procedures. In fact, this reduced basis algorithm is an instance of a class of algorithms known as Generalized Reduced Gradient methods, which date back to Abadie and Carpentier in the early 1960s. For further discussion and illustration of these methods applied to problems with state constraints, see [1, 2, 12, 13, 14].

Now, what happens if I add a partial analysis step to the algorithm for the AAO formulation, an analysis that enforces feasibility with respect to the block of constraints $h_2$, as in Figure 3? As it happens—as it was meant to happen—we obtain a reduced basis algorithm for the In-Between formulation.

THEOREM 5.2. *The modified algorithm in Figure 3 is equivalent to a reduced basis trust region algorithm for the In-Between formulation.*

*Proof.* I will check the following:

1. At every step, we are feasible with respect to the In-Between state constraint $h_2(x, u_1, u_2(x, u_1)) = 0$. This is true by design.
2. The subproblem we solve in the context of the AAO formulation to improve model feasibility yields a similar step for the In-Between formulation.
3. The subproblem we solve in the context of the AAO formulation to improve optimality subject to improved model feasibility does the same for the In-Between formulation.
4. The merit function we use to evaluate the AAO formulation corresponds to the correct merit function for the In-Between formulation.

We have

$$\begin{aligned}
&B_{AAO}^{-1} h_c \\
&= \begin{pmatrix} S^{-1} & -S^{-1}\dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1}S^{-1} & \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} + \left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1}S^{-1}\dfrac{\partial h_1}{\partial u_2}\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1} \end{pmatrix} \begin{pmatrix} h_1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1} \end{pmatrix} S^{-1} h_1 \\
&= \begin{pmatrix} I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial u_1} \end{pmatrix} B_{IB}^{-1} h_{IB}
\end{aligned}$$

and hence

$$\begin{aligned}
s_{AAO}^{LF} &= \begin{pmatrix} 0 \\ -\alpha B_{AAO}^{-1} h \end{pmatrix} \\
&= \begin{pmatrix} I_x & 0 \\ 0 & I_{u_1} \\ -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x} & -\left(\dfrac{\partial h_2}{\partial u_2}\right)^{-1}\dfrac{\partial h_2}{\partial x} \end{pmatrix} \begin{pmatrix} 0 \\ s_{IB}^{LF} \end{pmatrix} \\
&= W_2 s_{IB}^{LF},
\end{aligned}$$

where $\alpha$ is chosen so that

$$\| s_{IB}^{LF} \|_{W_2} \equiv \| W_2 s_{IB}^{LF} \| = r\delta.$$

This settles Point 2. Again, note the non-standard scaling of the trust-region; this tracks the effect of the step on the implicit variable $u_2$.

Meanwhile, by Proposition 3.5,

$$\begin{aligned}
W_{AAO} &= W_2 W_{IB}, \\
W_{AAO}^T H_{AAO} W_{AAO} &= W_{IB}^T H_{IB} W_{IB}, \\
W_{AAO}^T g_{IB} &= W_{IB}^T g_{IB},
\end{aligned}$$

and

$$\begin{aligned}
W_{AAO}^T H_{AAO} s_{AAO}^{LF} &= W_{IB}^T W_2^T H_{AAO} W_2 s_{IB}^{LF} \\
&= W_{IB}^T H_{IB} s_{IB}^{LF}.
\end{aligned}$$

Thus we have the equivalence between the subproblem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T W_{AAO}^T H_{AAO} W_{AAO}\sigma + H_{AAO} s_{AAO}^{LF} + g_{AAO}^T \sigma \\ \text{subject to} & \| \, s_{AAO}^{LF} + W_{AAO}\sigma \, \| \leq \delta \end{array}$$

for the AAO formulation and the subproblem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sigma^T W_{IB}^T H_{IB} W_{IB}\sigma + H_{IB} s_{AAO}^{LF} + g_{IB}^T \sigma \\ \text{subject to} & \| \, s_{IB}^{LF} + W_{IB}\sigma \, \|_{W_2} \leq \delta \end{array}$$

for the In-Between formulation, which establishes Point 3.

Moreover, because $h_2(x, u_1, u_2(x, u_1)) = 0$ at each step, Proposition 3.3 tells us that the augmented Lagrangian merit function reduces to

$$\begin{aligned} f_{AAO}(x, u(x)) + \lambda_{AAO}^T h(x, u(x)) + \frac{\rho}{2} \| \, h(x, u(x)) \, \|^2 = \\ f_{IB}(x, u_1) + \lambda_{IB}^T h_{IB}(x, u_1) + \frac{\rho}{2} \| \, h_{IB}(x, u_1) \, \|^2 , \end{aligned}$$

which is the augmented Lagrangian for the In-Between formulation, confirming Point 4. Thus we have a reduced basis trust region algorithm for the In-Between formulation. □

I have shown that the addition of an appropriate analysis step to the reduced basis algorithm in Figure 1 for the AAO formulation yields reduced basis algorithms for the MDF and In-Between formulations. Similar results hold for other algorithms one might consider, such as an SQP approach using a line-search.

**6. Conclusion.** The results of §3 and §5 follow from structural features of optimization problems governed by state constraints. A number of other practical consequences of these structural features are discussed in [14]. The results presented here illustrate, I believe, some of the rich and useful structure of optimization problems with state constraints.

I have shown that a properly structured and implemented reduced basis algorithm can be modified in a conceptually and practically simple manner to produce algorithms for alternative problem formulations, demonstrating a truly gratifying connection between the formulation of a problem and algorithms for its solution. One can easily build a telescoping family of algorithms that correspond to different formulations of the problem, and have different computational behavior.

The key to this approach is the availability of certain components inside the state constraints; in particular, at the very least the actions on vectors of the operators

$$\frac{\partial h_i}{\partial x}, \frac{\partial h_i}{\partial u}, \left(\frac{\partial h_i}{\partial u}\right)^{-1}, \left(\frac{\partial h_i}{\partial u}\right)^{-T}.$$

If simulation codes were written to make these components accessible to those of us doing optimization, then our task would be greatly facilitated, and alternative formulations and algorithms could be easily explored.

## REFERENCES


[1] J. Abadie, *Application of the GRG algorithm to optimal control problems*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland Elsevier, 1970.

[2] J. Abadie and J. Carpentier, *Généralisation de la méthode du gradient réduit de Wolfe au cas des contraintes non-linéaires*, Tech. Rep. HR 6678, Électricité de France, 1965.

[3] N. Alexandrov, *Multilevel algorithms for nonlinear equations and equality constrained optimization*, Tech. Rep. TR93–20, Department of Computational and Applied Mathematics, Rice University, May 1993. Ph.D. thesis.

[4] R. C. Buck, *Advanced Calculus*, McGraw-Hill, New York, 1956.

[5] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin, *Problem formulation for multidisciplinary optimization*, SIAM J. Optimization, 4 (1994), pp. 754–776.

[6] J. E. Dennis, Jr., M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, Tech. Rep. TR92–28, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1992. Revised July 1993.

[7] J. E. Dennis, Jr. and R. M. Lewis, *A comparison of nonlinear programming approaches to an elliptic inverse problem and a new domain decomposition approach*. Submitted to SIAM Journal on Scientific and Statistical Computing.

[8] J. E. Dennis, Jr. and R. E. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

[9] M. M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust region algorithm for constrained optimization*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 266–290.

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1981.

[11] R. T. Haftka, Z. Gurdal, and M. Kamat, *Elements of Structural Optimization*, Kluwer Academic Publishers, 1990.

[12] L. S. Lasdon, J. P. E. Coffman, R. MacDonald, J. W. McFarland, and K. Sepehrnoori, *Optimal hydrocarbon reservoir production policies*, Operations Research, 34 (1986), pp. 40–54.

[13] L. S. Lasdon, A. D. Waren, A. Jain, and M. Ratner, *Design and testing of a generalized reduced gradient code for nonlinear programming*, ACM Transactions on Mathematical Software, 4 (1978), pp. 34–50.

[14] R. M. Lewis, *A nonlinear programming perspective on the optimization of systems described by differential equations and other complex simulations*. In preparation.

[15] R. A. Tapia, *Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation*, in Nonlinear Programming 3: Proceedings of the Special Interest Group on Mathematical Programming Symposium, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., New York, 1978, Academic Press.