View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Apollo

Structural and Multidisciplinary Optimization manuscript No.
(will be inserted by the editor)

Differential Geometry Tools for Multidisciplinary Design Optimization, Part I: Theory

Craig Bakker · Geoffrey T. Parks

Received: date / Accepted: date

Abstract Analysis within the field of Multidisciplinary Design Optimization (MDO) generally falls under the headings of architecture proofs and sensitivity information manipulation. We propose a differential geometry (DG) framework for further analyzing MDO systems, and here, we outline the theory undergirding that framework: general DG, Riemannian geometry for use in MDO, and the translation of MDO into the language of DG. Calculating the necessary quantities requires only basic sensitivity information (typically from the state equations) and the use of the implicit function theorem. The presence of extra or non-differentiable constraints may limit the use of the framework, however. Ultimately, the language and concepts of DG give us new tools for understanding, evaluating, and developing MDO methods; in Part I, we discuss the use of these tools and in Part II, we provide a specific application.

Keywords Multidisciplinary Design Optimization · Differential Geometry · Design Analysis

1 Introduction

Multidisciplinary Design Optimization (MDO) is concerned with the optimization of systems that have coupled subsystems or disciplines. These systems typically have several disciplines, each with design variables specific to that discipline; state variables that represent the outputs of their respective disciplines; and design variables common to more than one discipline – local design, state, and global design variables, respectively. MDO applies various decomposition strategies, known as architectures, to these systems to make them more amenable to optimization. These strategies include Analytical Target Cascading (ATC) (Kim 2001), Simultaneous Analysis and Design (SAND) (Haftka 1985), Quasiseparable Decomposition (QSD) (Haftka and Watson 2005), Multidisciplinary Feasible (MDF) (Cramer et al. 1994), Collaborative Optimization (CO) (Braun and Kroo 1997), Bi-Level Integrated System Synthesis (BLISS) (Sobieszczanski-Sobieski et al. 1998), Concurrent Subspace Optimization (CSSO) (Bloebaum et al. 1992), and Individual Discipline Feasible (IDF) (Cramer et al. 1994); naming variations exist within the literature, but these are the most commonly used appellations. Theoretical MDO work has focused on architecture equivalence or convergence proofs and the determination of design sensitivities for MDO, but few architectures have these proofs; most decomposition methods that have such proofs require special problem structures not typically found in MDO (Haftka and Watson 2006). In this paper, we develop a differential geometry (DG) framework for facilitating the analysis and evaluation of MDO methods: Part I details the underlying theory, and Part II uses that theory to investigate QSD.

Previously presented as *On the Application of Differential Geometry to MDO*. In: 12th Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA, Indianapolis, Indiana

C. Bakker · G.T. Parks
Department of Engineering
University of Cambridge
Cambridge, United Kingdom
CB2 1PZ E-mail: ckrb2@cam.ac.uk

2 Review of MDO analysis

In light of the extensive analytical work we are about to undertake, beginning with this paper, it is appropriate to review the analysis which has previously been done

within the field. This work falls broadly under two categories: architecture proofs and the manipulation and analysis of sensitivity information.

2.1 Architecture convergence and equivalence

There are not many convergence or equivalence proofs for the various architectures in the MDO literature. Convergence is a fairly self-explanatory concept, but it should be noted that, in the context of MDO, equivalence refers to the correspondence between (optimal) solutions of the decomposed problem and (optimal) solutions of the original problem.

Most of the extant MDO analysis has been focused around ATC. ATC was developed by Kim (2001), and he conjectured that ATC was convergent based on its similarity to Hierarchical Overlapping Coordination, a decomposition-based optimization method with proven convergence. Michelena et al. (2003) later proved this conjecture correct given a hierarchical problem structure and certain coordination schemes.

Considerable effort has also been put into the investigation of penalty functions and their weightings within an ATC context, as these can strongly affect convergence performance and efficiency. Han and Papalambros (2010) proved the existence of convergent weights with L_∞ norms (though not how to find such weights); however, it seems that most of the literature in this area has focused on Augmented Lagrangian coordination (Kim et al. 2006; Tosserams et al. 2006, 2009).

Work in solution equivalence has been done with QSD. With convexity assumptions and the quasiseparable problem structure, Haftka and Watson (2005) proved the equivalence of optimal solutions (both global and local) between the original and decomposed problems while also showing that feasible points in one problem corresponded to feasible points in the other.

Conversely, convergence issues for CO in both theory and practice have been highlighted; the problem arises because the penalty functions used lead to either nondifferentiability or vanishing Lagrange multipliers at optimal points (Alexandrov and Lewis 2002). Some solutions have been proposed to keep the concept underlying the CO architecture while overcoming these issues: Modified Collaborative Optimization (MCO) from DeMiguel and Murray (2000) and Enhanced Collaborative Optimization (ECO) from Roth and Kroo (2008). DeMiguel and Murray (2001; 2006) provided equivalence and convergence proofs for their proposal, but it is interesting to note how similar their formulation ends up being to that of ATC. This is meant not as a disparagement of their work but rather as an observation that certain solution methods (i.e. a combination of objective and penalty functions at each level of a multi-level hierarchical optimization problem) seem to lend themselves more easily to convergence proofs than others.

2.2 Sensitivity calculations

Some work has been done to translate sensitivity information between architectures (Alexandrov and Lewis 2000, 2003), but so far, this has only been done between the monolithic architectures (MDF, IDF, and SAND). Decomposition in the distributed architectures seems to be a significant obstacle to such translation.

Using the implicit function theorem, Sobieszczanski-Sobieski (1990a) developed the Global Sensitivity Equations (GSE) to calculate cross-disciplinary sensitivities without needing costly multidisciplinary analyses; he then extended the method for the calculation of higher-order sensitivities (Sobieszczanski-Sobieski 1990b). Although this did not completely eliminate the costs involved in sensitivity analysis, it did make substantial reductions possible: for architectures that require this information for their decomposition procedures (e.g. BLISS and CSSO), a mechanism like the GSE is necessary for good performance on coupled systems.

Optimum Sensitivity Analysis (OSA) is used to calculate the derivatives of an optimal solution with respect to problem parameters (i.e. quantities that were held constant during the optimization) (Barthelemy and Sobieszczanski-Sobieski 1983a,b). These calculations enable perturbation analyses to be done on the solution without needing costly re-optimizations each time. This is particularly useful for distributed or multi-level MDO architectures because OSA can then be used to obtain derivatives cheaply from the lower level problem for use in the upper level problem (or vice versa).

All of this is mainly concerned with the use of basic sensitivity information to gain further knowledge about the system – it does not consider how sensitivity information is calculated in the first place. The calculation of that basic information is an important area of research in its own right. See van Keulen et al. (2005) for a review thereof. Martins and Hwang (2012) also provide a review of the different methods available for calculating derivatives in multidisciplinary systems; they include methods for obtaining basic sensitivity information and for obtaining cross-disciplinary sensitivities.

3 Differential geometry theory

The basic idea behind our framework is that MDO consists of optimization done on manifolds – which are es-

sentially higher-dimensional versions of surfaces – and that the language of DG can thus be used to describe the relevant processes and quantities; the constrained optimizations in Euclidean spaces are reconceptualized as unconstrained optimizations on Riemannian manifolds embedded in those Euclidean spaces. We do this translation in Section 4, but first, we give the theory underlying that translation. The explanation given in this section is not meant to be exhaustive – it merely gives the context and background necessary to make later derivations comprehensible. More detailed treatments are available elsewhere. The material for this introduction is mainly drawn from Boothby (1986) and others are cited where pertinent.

3.1 Basic concepts in differential geometry

Differential geometry is concerned with doing mathematics (such as calculus) on generally non-Euclidean spaces. In Euclidean geometry, the basic structures of space are linear: lines, planes, and their analogues in higher dimensions; DG, on the other hand, deals with manifolds (which may be Euclidean but often are not). Manifolds are abstract mathematical spaces which locally resemble the spaces described by Euclidean geometry but may have a more complicated global structure. They can also be considered extrinsically or intrinsically: an extrinsic view of a manifold considers it as being embedded in a higher-dimensional Euclidean space whereas the intrinsic view considers the manifold more abstractly without any need for a surrounding space (Ivancevic and Ivancevic 2007).

In the context of our reconceptualization of MDO as discussed earlier, this means that we could consider the constraint manifolds extrinsically, as being embedded in the total design space (which would be a higher-dimensional Euclidean space), or intrinsically, without reference to the total design space in which it is embedded. The extrinsic view would see the manifold as a set of points in $\Re^n$, but, as we will show in this section, manifolds have additional mathematical structures which do not apply to sets of points in general.

Consider a manifold M of dimension n (denoted as an n-manifold). About any point $p \in M$, there exists a coordinate neighbourhood, also known as a chart, consisting of a neighbourhood U of p and a mapping $\varphi : U \to \tilde{U}$, where $\tilde{U}$ is an open subset of $\Re^n$. If M is a manifold with a boundary ∂M, and $p \in \partial M$, then

$$\tilde{U} \subset H^n, H^n = \left\{ \left(x^1, x^2, \ldots, x^n \right) \in \Re^n | x^n \geq 0 \right\} \qquad (1)$$

Note that ∂M is itself a manifold of dimension $n-1$. The mapping φ need not be unique, however; Fig. 1 shows an example of these charts. This mapping is what allows us to take the familiar analyses done in (flat) Euclidean space and apply them to (curved) manifolds. Ultimately, the chart is what makes it possible to define things like coordinate reference frames and derivatives.

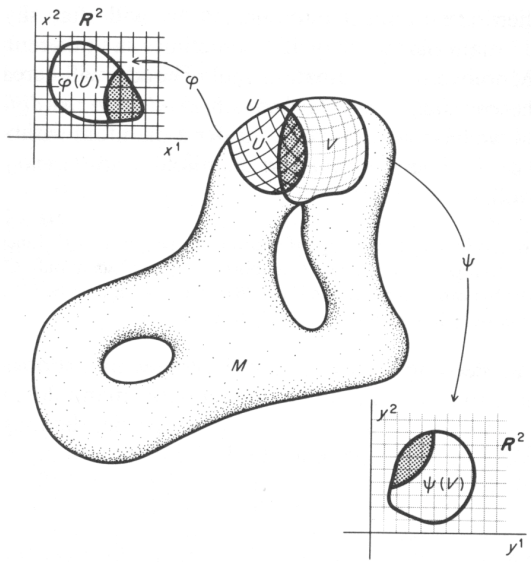

Fig. 1 Overlapping charts on a 2-manifold in 3-space (Boothby 1986, used with permission)

Differentiable manifolds are simply manifolds such that φ is differentiable at every $p \in M$. For the purposes of this discussion, we assume that all the manifolds we encounter are sufficiently differentiable that questions of discontinuities or singularities do not arise.

Given any two manifolds M and N (and $\Re^n$ is a manifold, albeit a globally Euclidean one), it is possible to construct a product manifold $M \times N$ with structure inherited from M and N. Conversely, a submanifold N of a manifold M is a subset of points in M such that N itself is also a manifold, and submanifolds inherit certain properties from the manifolds in which they are embedded. Identifying product manifold or submanifold structures is valuable because it can simplify calculations done on such manifolds and help to show the relationships between different manifolds. Figure 2 gives an example of some submanifolds: the black curve is a submanifold of the surface in which it is embedded, which is in turn a submanifold of the surrounding Euclidean space $\Re^3$. For more on product manifolds, submanifolds, and the properties of both, see Boothby (1986) or Yano and Kon (1984).

The tangent space at p is the vector space consisting of all vectors tangent to M at p, and it is denoted by $T_p(M)$. It is itself an n-manifold. The tangent bundle $T(M)$ consists of the union of the tangent spaces at

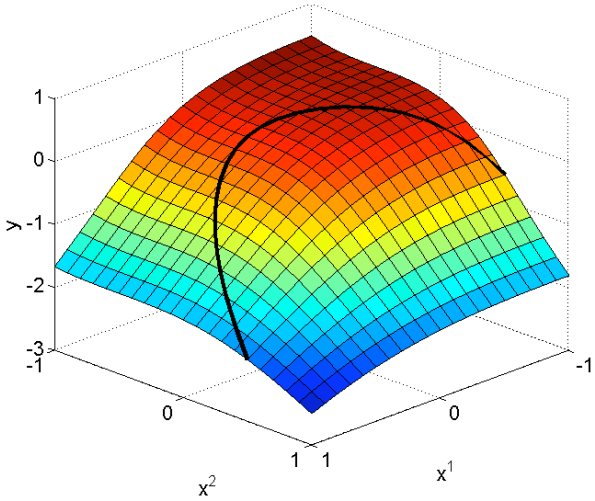

Fig. 2 A 1-D submanifold of a 2-D manifold in $\Re^3$

each point $p \in M$, and it is a $2n$-manifold. Vector fields on the manifold "live" in the tangent bundle. The tangent spaces and the tangent bundle also have duals – the cotangent spaces $T_p^*(M)$ and cotangent bundle $T^*(M)$, respectively, where covector fields "live" (Ivancevic and Ivancevic 2007); the dual of a vector space is the space of linear functionals such that the product of a member of a vector space with a member from the dual space returns a scalar. These tangent and cotangent spaces are important for defining higher-order tensors, too, and they play a role in derivative calculations as well as the determination of angles and lengths on the manifold.

Let $F : M \to N$ be a smooth mapping between manifolds. Given F, which maps between points of the two manifolds, we can induce transformations of other mathematical objects between the two manifolds, denoting such inductions with a correctly placed $*$. For example, $F_* : T_p(M) \to T_{F(p)}(N)$. A subscript $*$ indicates that the induced mapping goes in the same "direction" as the original mapping, whereas a superscript indicates that the induced mapping goes in the opposite "direction" as the original mapping; see Chapter Four of Boothby (1986) for further explanation.

This idea of induced mappings allows us to develop coordinate bases for the tangent and cotangent spaces, using the mapping φ, at p. Consider $\varphi_* : T_p(M) \to T_{\varphi(p)}(\Re^n)$. In keeping with typical DG notation, we define the basis vectors of $T_{\varphi(p)}(\Re^n)$ as

$$\frac{\partial}{\partial x^i}, \; i = 1, 2, \ldots, n \tag{2}$$

We can then define the basis vectors of $T_p(M)$ as

$$\mathbf{E}_i = \varphi_*^{-1}\left(\frac{\partial}{\partial x^i}\right), \; i = 1, 2, \ldots, n \tag{3}$$

The $\mathbf{E}_i$'s thus form a basis for $T_p(M)$, and an analogous procedure can be performed for the cotangent basis vectors (represented as dx^i in $T^*_{\varphi(p)}(\Re^n)$ and $\mathbf{E}^i$ in $T_p^*(M)$). Figure 3 shows how the grid in Euclidean space (with its associated basis vectors in the directions of the coordinate axes) are mapped back to the manifold with the basis vectors correspondingly transformed. This shows the connection, mentioned earlier, between the manifold's chart and its coordinates.

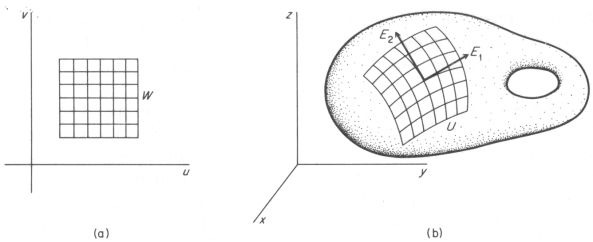

Fig. 3 Grid with basis vectors on a manifold (Boothby 1986, used with permission)

Moreover, although there are an infinite number of possible bases that could be chosen for each space, for each set of basis vectors $\mathbf{E}_i$, there exists a unique set of dual basis vectors $\mathbf{E}^i$ such that $\mathbf{E}^i(\mathbf{E}_j) = \delta^i_j$, where δ^i_j is the Kronecker delta, and the basis vectors as just defined form such a pairing. Keep in mind, however, that since φ, in general, varies across the manifold, the tangent and cotangent basis vectors will also vary from point to point. Figure 4 shows a set of orthonormal basis vectors varying over the manifold.

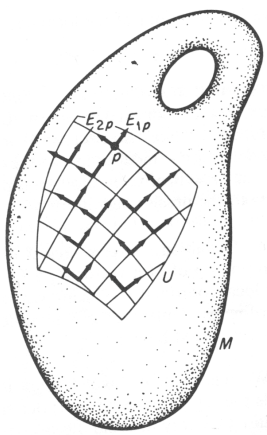

Fig. 4 A field of basis vectors on a manifold (Boothby 1986, used with permission)

We will use the notation

$$\mathbf{E}_i = \frac{\partial}{\partial w^i},\ \mathbf{E}^i = dw^i \tag{4}$$

The significance of the subscript and superscript indices is explained in Section 3.2. Given this notation, a vector X on the manifold can be expressed as

$$X = X^1 \frac{\partial}{\partial w^1} + X^2 \frac{\partial}{\partial w^2} + \ldots + X^n \frac{\partial}{\partial w^n} \tag{5}$$

and a covector ω would be expressed as

$$\omega = \omega_1 dw^1 + \omega_2 dw^2 + \ldots + \omega_n dw^n \tag{6}$$

Consider now a scalar function f. Its differential df and directional derivative in the direction X are given, respectively, by

$$df = \sum_i \frac{\partial f}{\partial w^i} dw^i \tag{7}$$

$$Xf = \sum_i X^i \frac{\partial f}{\partial w^i} \tag{8}$$

3.2 Tensors and tensor notation

A tensor is a geometrical object that behaves a certain way under coordinate transformations; a tensor representation independent of the chosen basis is unnecessarily abstract for our purposes, so we will use index notation. Tensors come in three basic types: contravariant, covariant, and mixed. These types are denoted by indices: contravariant indices are superscripted, covariant indices are subscripted, and the order of the tensor is determined by the number of indices. For example, a first-order contravariant tensor (i.e. a vector) would be denoted by v^i, and a first-order covariant tensor (i.e. a covector) would be v_i. It is understood that the tensor has a distinct value for each $i = 1, 2, \ldots, n$, where n is the dimension of the space under consideration. Continuing on, t^{ij}, t_{ij}, and t^i_j would be second-order contravariant, covariant, and mixed tensors, respectively. The tensor R^i_{jkl} would be a fourth-order mixed tensor with one contravariant index and three covariant indices (Ivancevic and Ivancevic 2007). To link back to the discussion of tangent and cotangent spaces, the covariant indices of tensors "live" in the tangent space and products thereof whereas the contravariant indices live in the cotangent space and products thereof. This matters because covariant and contravariant indices behave differently under covariant differentiation (as shown in Section 3.3) and coordinate transformations. We also mention the Einstein summation convention. The convention is that repeated indices are summed over:

$$a^i b_i = \sum_i a^i b_i \tag{9}$$

Index notation with the summation convention will be used throughout this paper unless otherwise noted.

3.3 Riemannian geometry

A Riemannian manifold is a differentiable manifold with a symmetric, positive definite bilinear form (known as the metric tensor). The metric tensor is an important tool for doing calculations on the manifold. Given a Riemannian manifold M, the metric tensor g_{ij} defines an inner product on $T_p(M)$ and this makes it possible to perform a number of different mathematical operations on the manifold. In general, g_{ij} is defined by

$$g_{ij} = \mathbf{E}_i \cdot \mathbf{E}_j \tag{10}$$

where $\mathbf{E}_i$ and $\mathbf{E}_j$ are as previously defined. For example, for Cartesian coordinates in $\Re^n$, this is just

$$g_{ij} = \mathbf{E}_i \cdot \mathbf{E}_j = \delta_{ij} \tag{11}$$

where δ_{ij} is the Kronecker delta. The inner product for a pair of vectors is then defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = g_{ij} u^i v^j \tag{12}$$

Lengths on the manifold are calculated with g_{ij}:

$$ds^2 = g_{ij} dw^i dw^j \tag{13}$$

The metric tensor is one of the properties inherited by submanifolds and product manifolds – the submanifold and product manifold structures are reflected in the forms of their respective metric tensors. Christoffel symbols and curvature tensors can also be derived from g_{ij}. If g_{ij} is known at a point on M, it is possible to calculate the Christoffel symbols and the curvature of the manifold at that point:

$$\Gamma^i_{kl} = \frac{1}{2} g^{im} \left(\frac{\partial g_{mk}}{\partial w^l} + \frac{\partial g_{ml}}{\partial w^k} - \frac{\partial g_{kl}}{\partial w^m} \right) \tag{14}$$

$$R^i_{jkl} = \frac{\partial \Gamma^i_{jl}}{\partial w^k} - \frac{\partial \Gamma^i_{jk}}{\partial w^l} + \Gamma^m_{jl} \Gamma^i_{mk} - \Gamma^m_{jk} \Gamma^i_{ml} \tag{15}$$

$$R_{ij} = R^l_{ilj},\ R = g^{ij} R_{ij} \tag{16}$$

where Γ^i_{kl} is a Christoffel symbol, R^i_{jkl} is the Riemann curvature tensor, R_{ij} is the Ricci curvature tensor, R is the scalar curvature, g^{ij} is the inverse of g_{ij}, n is the manifold dimension, and $\mathbf{w} = \left\{w^1, \ldots, w^n\right\}^T$ are the domain variables (i.e. the manifold coordinates). We mention also the Weyl curvature tensor, C_{ijkl}; though a relevant quantity, its formula is too long to show conveniently here. If g_{ij} exists, then it is positive definite, and thus its inverse also exists (and is positive definite). The Christoffel symbols measure how the basis vectors change along the manifold

$$\nabla_{\mathbf{E}_i}\mathbf{E}_j = \Gamma^k_{ij}\mathbf{E}_k \tag{17}$$

and they show up in two important places: the covariant derivative and the geodesic equation; Γ^i_{jk} are essentially intermediate quantities – they are typically used for calculating other pieces of information. The geodesic equation is

$$\ddot{w}^i + \Gamma^i_{jk}\dot{w}^j\dot{w}^k = 0 \tag{18}$$

Solutions of the geodesic equation are the paths that particles moving along the manifold would take if they were not subjected to any external forces (Ivancevic and Ivancevic 2007). They are to curved spaces what straight lines are to flat spaces: great circles on a sphere are a familiar example. Minimal geodesics are also used to define distances between points on a manifold.

The covariant derivative, denoted with a subscript semi-colon, does two things: it projects the derivative onto the tangent space (Boothby 1986), and it maintains the tensorial character of whatever it derivates (Ivancevic and Ivancevic 2007). The first few formulae for the covariant derivative are

$$v^i_{;k} = v^i_{,k} + \Gamma^i_{jk}v^j \tag{19}$$

$$v_{i;k} = v_{i,k} - \Gamma^j_{ik}v_j \tag{20}$$

$$t^i_{kl;q} = t^i_{kl,q} + \Gamma^i_{qs}t^s_{kl} - \Gamma^s_{kq}t^i_{sl} - \Gamma^s_{lq}t^i_{ks} \tag{21}$$

where a subscript comma denotes an ordinary partial derivative (i.e. $v_{i,j} = \frac{\partial v_i}{\partial w^j}$). Notice how covariant and contravariant indices are differentiated differently; also, $g_{ij;k} = 0$ (Szekeres 2004). In the rest of this paper, where total derivatives need to be distinguished from regular partial derivatives, d will be used instead of ∂, but ∂ will generally be used to indicate the multivariate nature of the derivatives being taken.

The Riemann curvature tensor is somewhat more difficult to interpret: it measures the "acceleration" between geodesics (Szekeres 2004); R_{ij} describes how a neighborhood, as the points in the neighborhood move along geodesics, changes in volume, and C_{ijkl} describes how that neighborhood changes in shape. These two tensors capture the information contained in R^i_{jkl} (Penrose 1989). Each of these curvature tensors shows, in its own way, the nature and extent to which the manifold deviates from flat space.

4 Translating MDO into differential geometry

With the requisite background in DG, we can now translate the mathematical form of an MDO problem into the relevant geometric terminology; we will use a general MDO formulation so as to make our translation, and thus work done with the framework, similarly general and widely applicable within MDO.

4.1 Mathematical definition of MDO

Consider a generic MDO problem as

$$\min f(\mathbf{x}, \mathbf{y}, \mathbf{z}) \tag{22}$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \leq \mathbf{0} \tag{23}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0} \tag{24}$$

where $\mathbf{x}$ is the vector of local design variables, $\mathbf{y}$ is the vector of state variables, $\mathbf{z}$ is the vector of global design variables, $\mathbf{g}$ is the vector of inequality constraints, and $\mathbf{h}$ is the vector equation that defines the state variables; (24) is just a rearrangement of the state equations

$$y^i = \psi^i(\mathbf{x}_{(i)}, \tilde{\mathbf{y}}_{(i)}, \mathbf{z}),\ i = 1, 2, \ldots \tag{25}$$

where $\mathbf{x}_{(i)}$ is the set of local variables for discipline i and $\tilde{\mathbf{y}}_{(i)}$ is the set of all state variables excluding y^i. For the purposes of this paper, we will only focus on the equality constraints. We recognize, though, that the inequality constraints will eventually have to be considered (see Section 6.1). The variables can be further simplified by lumping them together: $\mathbf{w} = \{\mathbf{x}\ \mathbf{z}\}$ and $\mathbf{v} = \{\mathbf{w}\ \mathbf{y}\}$

Defining the MDO problem this way raises some questions. Firstly, there is the question of notation: there are several different notations in the MDO literature (Cramer et al. 1994; Martins and Lambe 2013). This particular notation was used for the clarity and simplicity which it would lend to our later derivations. Secondly, there is the question of terminology: what we have labeled as state variables have been variously described as state, coupling, or linking variables, each with slightly different roles in the overall optimization.

For the purposes of this work, we consider state variables to be outputs of their respective disciplines; these variables would then, typically, be inputs to other disciplines, but they could be used only within other state equations belonging to their discipline. The key point is that state variables are not seen as directly controllable by the designer: lift and drag, for example, would be state variables from an aerodynamics discipline in aircraft design. This has been done to mirror real-world design problems and for ease of derivation later on.

Manifolds are useful for MDO because the state equations, which describe the interactions between the different disciplines and state variables, implicitly define a manifold. This is the feasible design manifold – the space of all feasible designs. Let M_{feas} be the manifold defined by (24).

4.2 Tangent vectors

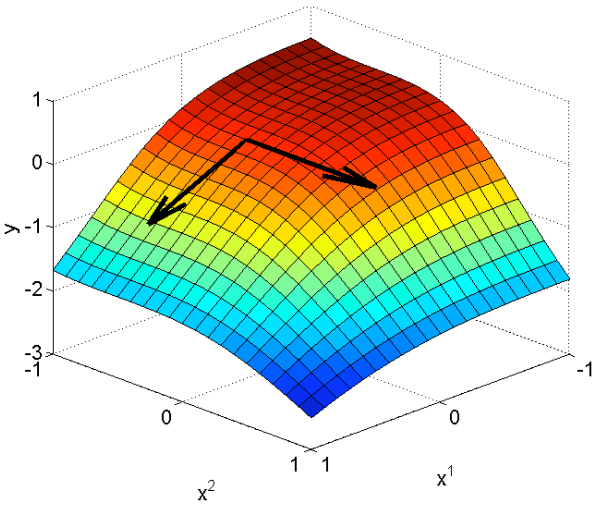

Fig. 5 A 2-manifold in 3-space with tangent vectors

Consider a 2-manifold embedded in 3-space as in Fig. 5. The tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ can be given as

$$\mathbf{t}_1 = \begin{Bmatrix} 1 \\ 0 \\ \frac{\partial y}{\partial x^1} \end{Bmatrix}, \ \mathbf{t}_2 = \begin{Bmatrix} 0 \\ 1 \\ \frac{\partial y}{\partial x^2} \end{Bmatrix} \tag{26}$$

It is possible to generalize these formulae to a manifold of arbitrary dimension and co-dimension. Consider the manifold defined by (24) and assume that the manifold is an n-dimensional manifold embedded in m-space (co-dimension $m-n$). The tangent vectors are then

$$\begin{aligned} \mathbf{t}_i = \frac{\partial \mathbf{v}}{\partial w^i} &= \left\{ \frac{\partial v^1}{\partial w^i}, \ldots, \frac{\partial v^m}{\partial w^i} \right\}^T \\ &= \left\{ 0, \ldots, 0, 1, 0, \ldots, 0, \frac{\partial y^1}{\partial w^i}, \ldots, \frac{\partial y^{m-n}}{\partial w^i} \right\}^T \end{aligned} \tag{27}$$

for $i = 1, 2, \ldots, n$ where the 1 is in the i-th slot and the necessary derivatives are calculated from (24) using the implicit function theorem (with $\mathbf{y}$ as an implicit function of $\mathbf{w}$).

It is straightforward to show that the tangent vectors are all linearly independent (though not all mutually orthogonal or of unit length). As such, the $\mathbf{t}_i$'s form a basis for the tangent space of the manifold, and so we can define them as the basis vectors that will be used throughout the rest of the paper: $\mathbf{E}_i = \mathbf{t}_i = \frac{\partial}{\partial w^i}$.

It is interesting to note some characteristics of this choice of tangent vectors. Each tangent vector corresponds directly to a design variable. The disciplinary subspaces of the tangent space are therefore just the linear combinations of the tangent vectors corresponding to the discipline's design variables. This also simplifies the derivation of later geometric quantities in terms of the pertinent partial derivatives. Defining the tangent vectors gives us our coordinate reference frame and thus our metric tensor (which is critical for future quantitative analysis). We do not use this information in Part II, but we highlight its use elsewhere in Section 5.

4.3 Derivation of geometric quantities

Assuming sufficient differentiability, and given the definition of (10), g_{ij} is defined by

$$g_{ij} = \frac{\partial v^k}{\partial w^i}\frac{\partial v^k}{\partial w^j} = \delta_{ij} + \frac{\partial y^k}{\partial w^i}\frac{\partial y^k}{\partial w^j} \tag{28}$$

The same results could be obtained by using submanifold theory and the metric induced from the embedding space ($\Re^m$), but this way is more geometrically intuitive. The Christoffel symbols are then

$$\Gamma^i_{kl} = g^{im}\frac{\partial y^s}{\partial w^m}\frac{\partial^2 y^s}{\partial w^k \partial w^l} \tag{29}$$

and the derivatives of Γ^i_{jk} are

$$\begin{aligned} \frac{\partial \Gamma^i_{kl}}{\partial w^j} &= \frac{\partial g^{im}}{\partial w^j}\frac{\partial y^s}{\partial w^m}\frac{\partial^2 y^s}{\partial w^k \partial w^l} \\ &+ g^{im}\frac{\partial^2 y^s}{\partial w^m \partial w^j}\frac{\partial^2 y^s}{\partial w^k \partial w^l} + g^{im}\frac{\partial y^s}{\partial w^m}\frac{\partial^3 y^s}{\partial w^k \partial w^l \partial w^j} \end{aligned} \tag{30}$$

where

$$\frac{\partial g^{im}}{\partial w^j} = -g^{in}\left(\frac{\partial^2 y^s}{\partial w^j \partial w^n}\frac{\partial y^s}{\partial w^p} + \frac{\partial y^s}{\partial w^n}\frac{\partial^2 y^s}{\partial w^j \partial w^p}\right) g^{pm} \tag{31}$$

Finally, R^i_{jkl}, R_{ij}, and R can be calculated from (15) and (16); the explicit formulae have been omitted due to their length. Fortunately for the sake of practical calculation, the third implicit derivatives in $\Gamma^i_{kl,j}$ cancel out in the determination of the Riemann curvature tensor, so it and the other curvature tensors only depend on the second implicit derivatives.

The particular structure of MDO problems thus simplifies the calculations so that the relevant quantities can be computed from the implicit derivatives of the state variables with respect to the design variables. Implicit derivatives can be calculated from the basic sensitivity information using the implicit function theorem.

4.4 The objective function and its derivatives

It is now necessary to see how the original MDO objective function is affected by this translation. The simplest way forward is to define the objective function $\eta(\mathbf{w}) = f(\mathbf{w}, \mathbf{y}(\mathbf{w}))$ and proceed from there. For a scalar function, the covariant derivative is identical to the regular derivative:

$$\eta_{;i} = \eta_{,i} = \frac{\partial \eta}{\partial w^i} = \frac{\partial f}{\partial w^i} + \frac{\partial f}{\partial y^k}\frac{\partial y^k}{\partial w^i} \tag{32}$$

This is just the reduced gradient of f. However, taking the covariant derivative again does not result in the reduced Hessian. The reduced Hessian and the second covariant derivative of η are, respectively,

$$\eta_{,ij} = \frac{\partial^2 f}{\partial w^i \partial w^j} + \frac{\partial^2 f}{\partial w^i \partial y^k}\frac{\partial y^k}{\partial w^j} + \frac{\partial^2 f}{\partial w^j \partial y^k}\frac{\partial y^k}{\partial w^i} + \frac{\partial^2 f}{\partial y^m \partial y^k}\frac{\partial y^k}{\partial w^i}\frac{\partial y^m}{\partial w^j} + \frac{\partial f}{\partial y^k}\frac{\partial^2 y^k}{\partial w^i \partial w^j} \tag{33}$$

$$\eta_{;ij} = \eta_{,ij} - \Gamma^l_{ij}\eta_{,l} \tag{34}$$

This should not affect the optimality conditions, as $\eta_{,l} = 0$ at an optimum point. Calculating the objective function derivatives like this demonstrates the use of the covariant derivative, which we introduced in Section 3.3, and we use these objective function covariant derivatives in research described in Section 5.

5 Use of the differential geometry framework elsewhere

We give here an overview of how the framework has been, is being, and may be utilized. In Sections 3 and 4, we provided more theory than is necessary for our application in Part II, and we did this with the intention that the theory outlined here be the foundation for the future evaluation and development of MDO methods. Our goal is to develop a framework for a wide range of MDO analysis. As such, we wish to outline the other work related to this theory, mention which parts of our framework that research is drawing from, and show how that work interacts with identified needs within MDO. The results detailed in Part II provide a concrete example, valuable in its own right, of the kind of work this framework can enable. Part II, with the rest of Section 5, serves as incentive for the effort of putting together Sections 3 and 4.

5.1 Foundational work

Certain pieces of work laid the foundation and provided the motivation for our framework. A notable item was the development of a coupling metric for MDO problems (Bakker et al. 2013a). This coupling metric addresses the desire expressed by Agte et al. (2010) for a unified coupling measure; our metric measures both coupling bandwidth and coupling strength but can be reduced to a single scalar value. We have also shown that our coupling measure extends to Multi-Objective Optimization (MOO) problems. This is a step towards connecting MDO and MOO as desired by Tolson and Sobieszczanski-Sobieski (1985) and Lewis and Mistree (1998). Our coupling metric was developed using g_{ij}.

Secondly, we used our framework to give a geometric interpretation of several common MDO architectures (Bakker et al. 2012). From this, we made some qualitative observations about architecture behaviour – a step towards the analysis desired by Sobieszczanski-Sobieski and Haftka (1997) while being valuable for our own work. Having defined the architecture manifolds, we need only to substitute them into our future results. In other words, the analysis we do with our framework can be generalized to those architectures. Our work here used some submanifold concepts as well as g_{ij}.

Thirdly, we have analytically linked optimization algorithm behaviour in MDO to manifold properties by combining theory from Ordinary Differential Equations with differential geometry (Bakker et al. 2013b). We connected algorithm stability with manifold curvature which in turn relates to our coupling metric. This can

show how different architectures (and thus the architectures' manifolds) affect algorithm stability on a given problem. Our work here addresses both the interest in convergence expressed by Sobieszczanski-Sobieski and Haftka (1997) and the desire of Agte et al. (2010) to use coupling information to guide decomposition decisions. These results drew on covariant derivatives (including but not limited to the objective function), the geodesic equation, and the curvature tensors.

5.2 Current applications and research directions

Following up on work already accomplished, we are currently working on using our coupling metric to develop coupling suspension techniques for MDO problems and to further investigate architecture characteristics and performance. We are also continuing to pursue the link we have established between MDO and MOO: we can combine our coupling metric with current MOO techniques and numerical solution methods for Partial Differential Equations to generate smart Pareto fronts (Bakker and Parks 2014); a smart Pareto front has solution points concentrated in areas of high interest (Mattson et al. 2004). This further serves our intention of making connections between MOO and MDO. Similar methods may also be used improve data sampling techniques in metamodelling. All of this uses g_{ij}.

In a new area, we are looking at applying Riemannian optimization algorithms to MDO problems; the mathematical structure provided by our framework now makes it possible to implement these algorithms computationally. Given that they have been derived for optimization on curved spaces, they may be more effective than their Euclidean counterparts on MDO manifolds. These Riemannian algorithms require using covariant differentiation (and thus Γ^i_{jk}).

5.3 Areas for future research

Researchers have identified three other related areas of optimization which they would also like to incorporate into MDO: optimal control (Allison and Herber 2013; Giesing and Barthelemy 1998), uncertainty quantification and propagation (Collopy et al. 2012; Lewis and Mistree 1998; Simpson and Martins 2011; Tolson and Sobieszczanski-Sobieski 1985), and topology optimization (Allison and Herber 2013; Simpson and Martins 2011). We believe that DG has the capability to provide a solid theoretical foundation for making these incorporations. The relevant mathematical theory already exists; it is simply a question of translating and applying that theory. Such work would likely require more DG than has been described here, but those additions would *build upon* our work here, not replace it.

6 Present framework limitations

Having touched on some of our framework's capabilities, we now consider its limitations under two main categories: constraint management and the calculation and differentiation of the relevant quantities.

6.1 Constraints

Constraint-related concerns can, at this point, come under two headings: inequality constraints and additional (non-state) equality constraints.

Inequality constraints define the boundary of M_{feas}, not M_{feas} itself, but they tend to play an important role in constrained optimization because optimal solutions often lie on the boundary of the feasible set. We have not yet explicitly taken them into account, however. One option for handling them would be to turn each inequality into an equality through the use of a slack variable and then consider the slack variables as artificial state variables. This seems less than ideal, however, for two reasons. Firstly, it is not effective at handling explicit bounds on variables: for example, if the slack variables $\mathbf{s}$ were implemented as $\mathbf{g}(\mathbf{w},\mathbf{y})+\mathbf{s}=\mathbf{0}$, all of the slack variables would still be required to satisfy $\mathbf{s}\geq\mathbf{0}$, and in any case, explicit bounds on the existing state and design variables would be left untouched. These variables' bounds define a manifold boundary just as much as the inequality constraints, albeit in a simpler way, so it seems that little is gained by this approach.

Secondly, although there may be relatively few state equations in a given design problem, there are often many more inequality constraints, so handling those constraints in this way would turn a low-dimensional feasible design manifold into a very high-dimensional manifold composed mainly of slack variables. Increasing the dimension of the design space in this way is counterproductive to the whole endeavor of seeking a simplifying analysis tool. Another option would be to just consider the active set as equalities (thereby defining a pertinent submanifold of ∂M_{feas}). However, the active set would be changing throughout the process of optimization, and changes in the active set would change the nature and properties of the submanifold of ∂M_{feas} in question, thus changing the analysis tools like g_{ij}, so this may not be helpful.

We have also not considered equality constraints beyond the state equations, but it is possible to have additional equality constraints which do not correspond to

a disciplinary output. The solutions to these additional constraints, assuming that the state equations were satisfied as well, would define a submanifold of the manifold defined by the state equations. It would be possible to define a metric (with the requisite derivatives) for this submanifold using implicit function and submanifold theorems, but then which design variables should be considered as dependent variables for the additional equations? The state equations naturally have the state variables as dependent variables, but now a decision has to be made. A designer could take a set of design variables (presumably but not necessarily global design variables) and redefine them to be dependent variables, but then those variables would effectively be state variables, and the situation would just revert to the originally defined design problem with the only equality constraints being state equations; this would be effectively the same as re-labeling some of the w^i's as y^i's. In this case, there is no loss in generality to simply assume that there are no equality constraints other than the state equations. It is difficult to see how else the equality constraints might be profitably dealt with.

The application of Value-Driven Design (VDD) to MDO may be worth considering here (Collopy et al. 2012). VDD removes design emphasis from constraints and places it on the objective function: "While it is recognized that in some industries a complete elimination of constraints is not possible, VDD's goal is to eliminate as many as possible, incorporating those preferences communicated through the requirements into the value function." (Mesmer et al. 2013, p.10) If VDD is as valuable an addition to MDO as its proponents claim, and if VDD were to gain traction in MDO circles, then the additional constraints discussed above would become less important or even non-existent, and thus the issues identified in this section would become correspondingly less important or non-existent.

There is a connection here to the optimization problem's Lagrangian as well. Given the reconceptualization of the constrained optimization problem as an unconstrained problem on a Riemannian manifold (see the beginning of Section 3), the resulting Lagrangian (on the Riemannian manifold) could have *inequality* constraint terms, but it would not have equality terms corresponding to the state equations because those constraints would have already been "absorbed", for lack of a better term, into the Riemannian manifold and its structure; the additional non-state equality constraints mentioned earlier could show up in the Lagrangian, though. We reiterate, though, that part of the point of this framework is to facilitate seeing MDO as unconstrained optimization (or at least optimization without equality constraints) on Riemannian manifolds rather than simply as constrained optimization.

6.2 Calculability and differentiability

Throughout this analysis, we have tacitly assumed that for a given $\mathbf{w}$ there is a unique solution $\mathbf{y}$ to $\mathbf{h}(\mathbf{w},\mathbf{y}) = \mathbf{0}$. That is to say, there exists a point-to-point mapping

$$\mathbf{h_y}(\mathbf{w}) = \{\mathbf{y}|\mathbf{h}(\mathbf{w},\mathbf{y}) = \mathbf{0}\} \tag{35}$$

This need not be the case, however, and it is an issue that has been addressed in the similar context of Bilevel Programming Problems (Bard 1991). If (35) is in fact a point-to-set mapping, it may have differentiability problems; if each $\mathbf{y}$ is *locally* unique, that should not be a problem – all of the tensor quantities calculated so far are strictly local – but if not, it is uncertain what would happen. It effectively hinges on $\frac{\partial h}{\partial y}$ being invertible: we require it to be invertible in order to calculate the quantities described in Section 4.

The previous analysis has also assumed that the manifolds are sufficiently smooth to allow all of the operations that have been performed on them, and, moreover, that the Riemannian structure can be imposed on such a sufficiently smooth manifold. Every smooth manifold admits a Riemannian metric (Boothby 1986), but there are real-world design problems that are not smooth. The use of metamodelling in MDO becomes pertinent at this point: the metamodels often used to approximate those problems during optimization tend to smooth out the design space in the process of approximation (Sobieszczanski-Sobieski and Haftka 1997).

Furthermore, according to the Weierstrass theorem, any continuous function can be approximated to arbitrary precision by a smooth function (Boothby 1986). For our purposes, this means that even though we are only considering smooth design manifolds, those manifolds can be considered as arbitrarily good approximations to all continuous design manifolds. Thus the restriction is less severe than might be otherwise thought. Since, as previously noted, the tensor quantities being calculated are all local, the smoothness required of any point on the manifold would only have to be local, not global, too. If there are discontinuities in the manifold, however, those discontinuities could cause problems in ways similar to local non-uniqueness as discussed above.

7 Conclusions

The mathematics behind DG can initially appear somewhat intimidating, but once given concrete form in their

application to MDO, they actually prove to be quite tractable. The index notation, in particular, simplifies the bookkeeping involved, and the resulting calculations end up being less intimidating combinations of calculus and algebra. Visualizing high-dimensional manifolds is also notoriously difficult, but the mathematics here make it possible to work without needing direct geometric intuition into a given problem. We have elsewhere already used some of the tools provided by our framework; in Part II we will provide an example of how some of those tools can be used to investigate and improve the performance of QSD.

Acknowledgments

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Cambridge Commonwealth Trust and Cambridge Overseas Trust (CCT/COT).

References

Agte J, de Weck O, Sobieszczanski-Sobieski J, Arendson P, Morris A, Spieck M (2010) MDO: Assessment and direction for advancement - an opinion of one international group. Structural and Multidisciplinary Optimization 40:17–33

Alexandrov NM, Lewis RM (2000) Algorithmic perspectives on problem formulations in MDO. In: 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis, AIAA, Long Beach, California

Alexandrov NM, Lewis RM (2002) Analytical and computational aspects of collaborative optimization for multidisciplinary design. AIAA Journal 40:301–309

Alexandrov NM, Lewis RM (2003) Dynamically reconfigurable approach to multidisciplinary problems. In: 16th AIAA Computational Fluid Dynamics Conference, AIAA, Orlando, Florida

Allison JT, Herber DR (2013) Multidisciplinary design optimization of dynamic engineering systems. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA, Boston, Massachusetts

Bakker C, Parks GT (2014) Multi-objective optimization, MDO, and differential geometry controlled pareto front solution spacing. In: 4th International Conference on Engineering Optimization, ISSMO, Lisbon, Portugal

Bakker C, Parks GT, Jarrett JP (2012) Geometric perspectives on MDO and MDO architectures. In: 12th Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA, Indianapolis, Indiana

Bakker C, Parks GT, Jarrett JP (2013a) Differential geometry and design coupling in MDO. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA, Boston, Massachusetts

Bakker C, Parks GT, Jarrett JP (2013b) Optimization algorithms and ODE's in MDO. In: ASME 2013 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, Portland, Oregon

Bard JF (1991) Some properties of the bilevel programming problem. Journal of Optimization Theory and Applications 68:371–378

Barthelemy JFM, Sobieszczanski-Sobieski J (1983a) Extrapolation on optimum design based on sensitivity derivatives. AIAA Journal 21:797–799

Barthelemy JFM, Sobieszczanski-Sobieski J (1983b) Optimum sensitivity derivatives of objective functions in nonlinear programming. AIAA Journal 21:913–915

Bloebaum CL, Hajela P, Sobieszczanski-Sobieski J (1992) Non-hierarchic system decomposition in structural optimization. Engineering Optimization 19:171–186

Boothby WM (1986) An Introduction to Differentiable Manifolds and Riemannian Geometry. Academic Press, Inc., Boston

Braun RD, Kroo IM (1997) Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In: Alexandrov N, Hussaini M (eds) Multidisciplinary Design Optimization: State of the Art, SIAM, Philadelphia, pp 98–116

Collopy PD, Bloebaum CL, Mesmer BL (2012) The distinct and interrelated roles of value-driven design, multidisciplinary design optimization, and decision analysis. In: 12th Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA, Indianapolis, Indiana

Cramer E, Dennis JE Jr, Frank PD, Lewis RM, Shubin GR (1994) Problem formulation for multidisciplinary optimization problems. SIAM Journal on Optimization 4:754–776

DeMiguel AV, Murray W (2000) An analysis of collaborative optimization methods. In: 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis, AIAA, Long Beach, California

DeMiguel V, Murray W (2006) A local convergence analysis of bilevel decomposition algorithms. Optimization and Engineering 7:99–133

Giesing JP, Barthelemy JFM (1998) A summary of industry MDO applications and needs. In: 7th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA, St. Louis, Missouri

Haftka RT (1985) Simultaneous analysis and design. AIAA Journal 23:1099–1103

Haftka RT, Watson LT (2005) Multidisciplinary design optimization with quasiseparable subsystems. Optimization and Engineering 6:9–20

Haftka RT, Watson LT (2006) Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems. Optimization and Engineering 7:135–149

Han J, Papalambros PY (2010) A note on the convergence of analytical target cascading with infinite norms. Journal of Mechanical Design – Transactions of the ASME 132:034,502

Ivancevic VG, Ivancevic TT (2007) Applied Differential Geometry: A Modern Introduction. World Scientific Publishing Co. Pte. Ltd., Singapore

van Keulen F, Haftka RT, Kim NH (2005) Review of options for structural design sensitivity analysis. part 1: Linear systems. Computer Methods in Applied Mechanics and Engineering 194:3213–3243

Kim HM (2001) Target cascading in optimal system design. PhD thesis, University of Michigan, Ann Arbor, Michigan

Kim HM, Chen W, Wiecek MM (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. AIAA Journal 44:2197–2207

Lewis K, Mistree F (1998) The other side of multidisciplinary design optimization: Accomodating a multiobjective, uncertain and non-deterministic world. Engineering Optimization 31:161–189

Martins JRRA, Hwang JT (2012) Review and unification of methods for computing derivatives of multidisciplinary systems. In: 53rd AIAA/ASME/ASCE/ASC Structures, Structural Dynamics, and Materials Conference, AIAA, Honolulu, Hawaii

Martins JRRA, Lambe AB (2013) Multidisciplinary design optimization: Survey of architectures. AIAA Journal 51:2049–2075

Mattson CA, Mullur AA, Messac A (2004) Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. Engineering Optimization 36:721–740

Mesmer BL, Bloebaum CL, Kannan H (2013) Incorporation of value-driven design in multidisciplinary design optimization. In: 10th World Congress on Structural and Multidisciplinary Optimization, ISSMO, Orlando, Florida

Michelena N, Park H, Papalambros PY (2003) Convergence properties of analytical target cascading. AIAA Journal 41:897–905

de Miguel AV (2001) Two decomposition algorithms for nonconvex optimization problems with global variables. PhD thesis, Stanford University, Stanford, California

Penrose R (1989) The Emperor's New Mind. Oxford University Press, Oxford, England

Roth BD, Kroo IM (2008) Enhanced collaborative optimization: A decomposition-based method for multidisciplinary design. In: ASME 2008 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, Brooklyn, New York

Simpson TW, Martins JRAA (2011) Multidisciplinary design optimization for complex engineered systems: Report from a National Science Foundation workshop. Journal of Mechanical Design – Transactions of the ASME 133:101,002

Sobieszczanski-Sobieski J (1990a) Sensitivity of complex, internally coupled systems. AIAA Journal 28:153–160

Sobieszczanski-Sobieski J (1990b) Higher order sensitivity analysis of complex coupled systems. AIAA Journal 28:756–758

Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: Survey of recent developments. Structural Optimization 14:1–23

Sobieszczanski-Sobieski J, Agte JS, Sandusky RR Jr (1998) Bi-level integrated system synthesis (BLISS). Tech. Rep. NASA/TM-1998-208715, NASA Langley Research Center, Hampton, Virginia

Szekeres P (2004) A Course in Modern Mathematical Physics. Cambridge University Press, Cambridge, England

Tolson RH, Sobieszczanski-Sobieski J (1985) Multidisciplinary analysis and synthesis: Needs and opportunities. In: 26th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference, AIAA, Orlando, Florida

Tosserams S, Etman LFP, Papalambros PY, Rooda JE (2006) An augmented Lagrangian relaxation for analytical target cascading. Structural and Multidisciplinary Optimization 31:176–189

Tosserams S, Etman LFP, Rooda JE (2009) Block-separable linking constraints in augmented Lagrangian coordination. Structural and Multidisciplinary Optimization 37:521–527

Yano K, Kon M (1984) Structures on Manifolds. World Scientific Publishing Co. Pte. Ltd., Singapore