A Dynamic Parametrization Scheme for Shape Optimization Using Quasi-Newton Methods

John T. Hwang * and Joaquim R. R. A. Martins †

University of Michigan, Ann Arbor, Michigan, 48109, United States

A variable parametrization scheme is developed and demonstrated for shape optimization using quasi-Newton methods. The scheme performs adaptive parametrization refinement while preserving the approximate Hessian of the shape optimization problem and enables free-form shape design using quasi-Newton optimization methods. Using a B-spline parametrization, the scheme is validated using a 1-D shape approximation problem and is shown to improve efficiency and optimal solution quality compared to the traditional quasi-Newton method. The scheme is also applied to a 3-D test problem, demonstrating the feasibility of free-form shape optimization using parametrization refinement and a method for partially constraining the degrees of freedom.

Nomenclature

A	Jacobian of the constraints
B	matrix whose column vectors are B-spline basis vectors
c	vector of constraints
C	control point
D	design space reduction tensor — relates the design variables to the control points
f	objective function; function defining the parametric representation of a manifold
g	gradient vector
$\mathcal{H}$	Hessian matrix
$\mathcal{L}$	Lagrangian of the optimization problem
m	dimension of the manifold
$\mathcal{M}$	manifold
n	dimension of the Euclidean space that is a superset of the manifold
N	interpolating shape function
P	discretized point
T	parametrization transformation matrix
V	point on a manifold
x	vector of design variables
λ	Lagrange multiplier
ξ	parametric coordinates of a point on a manifold
Superscripts	
$^+$	left Moore–Penrose pseudoinverse
*	optimal value
Subscripts	
$_0$	initial value

*PhD Candidate, Department of Aerospace Engineering.

†Associate Professor, Department of Aerospace Engineering.

I. Introduction

Shape optimization is used in a wide range of engineering disciplines. Its roots lie in structural design where load carrying efficiency can be very sensitive to the shape of the structure, motivating the use of optimization. Examples include the design of fillets, structures with holes,[1] general 2-D bodies,[2] and curvilinear-stiffened panels.[3] Shape optimization in fluid mechanics is another large area of research with numerous applications in aerospace engineering. Specific examples range from the design of aeroshells to shape design of aircraft outer mold lines (OML).[4] Beyond structural and fluid mechanics, other applications to which shape optimization has been applied include hull design,[5] automotive aerodynamics,[6] electromagnetics,[7] and trajectory optimization.[8]

In design settings, the importance of shape optimization is likely to increase as computational tools mature. With greater fidelity, physics-based models become more complex and unintuitive, increasing the value of optimization for circumventing the need to explore design spaces manually. Additionally, more accurate models could enable shape optimization to produce physically realizable designs rather than simply offering an aid for understanding the physics. However, numerical optimizers must evolve with the computational tools in order to achieve these benefits. This presents a challenge because analysis tools such as those in computational fluid dynamics (CFD) and finite element analysis (FEA) achieve high accuracy by computing millions of state variables at high computational cost. In addition to minimizing this cost, optimizers must consider a larger number of design variables that provides the design flexibility to take advantage of the increased level of fidelity. Thus, a high-fidelity analysis tool necessitates an efficient, high-fidelity shape optimization algorithm to make full use of its capabilities.

In recent literature, a commonly explored avenue for improving the efficiency of shape optimization algorithms is the use of an adaptive or refinement-based parametrization. Kohli and Carey[9] proposed a shape optimization algorithm with adaptive refinement, citing the increased likelihood of finding global optima and the reduced cost of evaluating gradients with finite-difference schemes in early iterations. Desideri et al.[10] applied a nested, adaptive Bezier parametrization to a 2-D aerodynamic shape optimization problem and found improved convergence rates and solution quality with respect to the objective function. Nagy et al.[11] extended the isogeometric analysis concept with a refined parametrization for sizing variables, which are optimized simultaneously with the shape variables, while Han and Zingg[12] applied an evolutionary geometry parametrization to airfoil and wing shape optimization problems, achieving improved efficiency.

In each of these works, the concept of varying the parametrization during the optimization process succeeded in reducing computation time, improving the accuracy of the optimal solution, or both. What remains to be addressed is the implementation of a dynamically varying parametrization within a quasi-Newton optimizer. Previous studies use gradient-free optimizers or restart the optimization algorithm after each refinement, losing the approximate Hessian information.

Integrating parametrization refinement with quasi-Newton methods would significantly enhance what are already the most efficient optimization algorithms for high-cost solvers such as CFD. It has been shown that gradient-based optimizers are one or two orders of magnitude more efficient than gradient-free optimizers,[13] a gap that only grows with the number of design variables. Among gradient-based options, quasi-Newton methods add the benefit of nearly second-order convergence close to the optimum, which makes it possible to satisfy stricter convergence tolerances. In conjunction with the adjoint method, they achieve even more favorable scaling with the number of design variables. The efficiency resulting from the synergy of the quasi-Newton and adjoint methods is evident in a 3-D aerodynamic shape optimization algorithm developed by Hicken and Zingg,[14] which can optimize 251 shape variables on a 1.2×10^6 node CFD mesh in roughly 100 CPU hours. This synergy is also an enabling component in multidisciplinary design optimization (MDO) as demonstrated by Kennedy and Martins,[15] who solved a multi-point aerostructural design problem with 97236 structural degrees of freedom, 9440 aerodynamic panels, and 446 design variables in less than 300 CPU hours using the coupled adjoint method and a quasi-Newton optimizer.

An important part of shape optimimization is the choice of basis in the parametrization. A widely used approach is to represent the shape using B-splines with uniform weights (hereafter referred to as simply B-splines) and to manipulate the shape through the B-spline control points. B-splines have many desirable mathematical properties; in particular, they represent the unique splines with minimum support for a given order, yielding local control. Furthermore, they are constructed from polynomials, meaning their evaluation and differentiation are both exact and fast to compute. B-splines are especially convenient because they are linear with respect to the control points.

Given that high-fidelity shape optimization is the objective, the approach advocated here is adjoint-

based quasi-Newton optimization with a B-spline parametrization. The primary contribution of this paper is the development of a dynamic parametrization scheme that is compatible with quasi-Newton optimization algorithms. This scheme offers three advantages. First, it addresses the likely presence of multiple local minima due to the large design space, by removing design flexibility in early iterations and gradually restoring it as the optimum is approached. Second, it enables free-form shape optimization by allowing for constraints on the degrees of freedom of the B-spline control points, which are necessary to define a well-posed problem but is not compatible with quasi-Newton methods using the traditional approach. Finally, it allows for automatic positioning of control points where they are most required.

II. The Shape Optimization Problem

The goal in a shape optimization problem is to find the manifold of a certain type that minimizes the objective function and satisfies the constraints. A manifold is a topological space locally homeomorphic to an open subset of Euclidean space of dimension m and is a subset of Euclidean space of dimension n. Thus, a point in the manifold is a vector of size n uniquely located by m coordinates, giving the following definition:

$$\mathcal{M} = \{(V_1, V_2, \ldots, V_n) \in \mathbb{R}^n \mid V_l = f_l(\xi_1, \xi_2, \ldots, \xi_m)\}, \tag{1}$$

where the vector V is a point in the manifold, the vector ξ represents the parametric coordinates of a point in the manifold, and f_l is a continuous surjective function that maps the set of the parametric coordinates ξ to the set of l^{th} entries of $V \in \mathcal{M}$.

The description of the manifold must be simplified to a finite number of points so that it can be represented numerically. When partial differential equations (PDEs) are involved, this involves discretizing the domain into a finite number of cells (finite volume method), elements (finite element method), or nodes (finite difference method). The natural tendency is to interpret this discretization step as the selection of a finite subset of the manifold by evaluating its parametric representation, f_k, at a discrete set of locations, changing the object being optimized from a manifold into a discrete set of points. In truth, this discretization step is not an evaluation of the manifold at a finite set of points — rather, its representation by a set of basis functions, denoted N_i, that spans a subset of the original set of manifolds. The discrete points, P_{il}, are simply the coefficients of these basis functions. The basis functions are interpolating functions that are often implied by the context; in the isoparametric finite element method the basis functions are the shape functions in the global coordinate frame, in finite-difference formulae the basis functions are polynomials of a specific order, and with numerical integration the basis functions are the polynomials used to derive the quadrature rule.

The next step is the selection of a parametrization, which is assumed here to be linear for simplicity. The discrete points themselves may be used as the degrees of freedom, but typically this results in a larger number than desired. Thus, one can use a much smaller number of basis vectors, B_{ij}, whose coefficents are the control points, C_{jl}.

Thus, the continuous, infinite degree of freedom manifold is reduced to one described by a discrete set of points, then reduced again to the control points (Einstein notation is used):

$$\begin{aligned}
\mathcal{M} &= \{(V_1, V_2, \ldots, V_n) \in \mathbb{R}^n \mid V_l = f_l(\xi_1, \xi_2, \ldots, \xi_m)\} \\
\mathcal{M} &= \{(V_1, V_2, \ldots, V_n) \in \mathbb{R}^n \mid V_l = N_i(\xi_1, \xi_2, \ldots, \xi_m) P_{il}\} \\
\mathcal{M} &= \{(V_1, V_2, \ldots, V_n) \in \mathbb{R}^n \mid V_l = N_i(\xi_1, \xi_2, \ldots, \xi_m) B_{ij} C_{jl}\}.
\end{aligned} \tag{2}$$

What this shows is that discretizing the manifold does not mean that the discrete object is what we are now optimizing — we are still optimizing a manifold, albeit in a subset of the original design space. That is, the set of all parametrized manifolds is a subset of the set of all discretized manifolds, and the set of all discretized manifolds is a subset of the set of all manifolds. With both discretization and parametrization, we lose some accuracy in the optimal shape, though this loss is mitigated by an appropriate choice of interpolation functions and basis vectors, respectively. The choice of basis vectors is more relevant to the current discussion since shape optimization only deals with the parametrization step.

It is apparent that m and n are two important characteristics of a shape optimization problem that are associated with the manifold being optimized. The dimension of the manifold, m, gives topological information regarding the structure of the points in the manifold as m is equal to the number of coordinates required to uniquely identify a given point in the manifold. The second one, n, is the dimension of the

Euclidean space that is a superset of the manifold. Because the parametric representation of a manifold — that is, the functions f_l — is not unique, each point in the manifold must have only $n-m$ degrees of freedom in a shape optimization problem. Among the space of possible directions in which a point can move, a subspace of dimension m represents movement within the manifold, which does not actually change the shape. The degrees of freedom that do change the shape are denoted, x.

A manifold is then reduced from a continuous object with infinite degrees of freedom to one represented by a finite number of variables through three stages:

1. $V_l = N_i(\xi_1, \xi_2, \ldots, \xi_m)P_{il}$ — Discretization: representation by points which are then interpolated
2. $V_l = N_i(\xi_1, \xi_2, \ldots, \xi_m)B_{ij}C_{jl}$ — Parametrization: representation by control points and their basis
3. $V_l = N_i(\xi_1, \xi_2, \ldots, \xi_m)B_{ij}D_{jkl}x_k$ — Design space reduction: representation by design variables

where x is the vector of design variables and D_{jkl} relates the design variable x_k to the control point C_{jl}. D_{jkl} represents the $n-m$-dimensional subspace in which control points are permitted to move, so in general it is a function of the control points.

Using this description, we can characterize some common classes of shape optimization problems:

- **Free-form shape optimization**: In free-form shape design problems, the design space reduction tensor evolves as the gradient-based optimizer progresses from iteration to iteration. In this case m constraints must be enforced for each control point, preventing movement tangent to the manifold.
- **Dimension reduction**: In contrast to free-form shape optimization, the design space reduction tensor simply eliminates m of the n dimensions in this case, and as a consequence the tensor D_{jkl} is constant and has a very simple form. An example is in airfoil design, where the control points are often permitted to move only normal to the chord line and are fixed in the chord-wise direction.
- **Problems with high-level shape parameters**: In wing design, design variables with physical meaning, such as sweep and span, are often used. In these types of problems, the Jacobian of the points with respect to the design variables is the analogue of the basis matrix of the B-spline parametrization, for example. The column vectors of the Jacobian are the shape functions corresponding to the design variables. Here, the set of design variables is mapped directly to the set of discretized points.
- **Trajectory optimization**: Trajectory optimization problems are special cases in which $m = 1$ and $n = 2$ or $n = 3$.
- **Partial differential equations**: Within the given formulation, partial differential equations (PDEs) can be treated as shape optimization problems. For example, the 2-D Laplace equation, $\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$, is a problem with $m = 2$ and $n = 3$. The set of all ϕ, x, and y form a 3-D Euclidean space, and the manifold is a surface that seeks to minimize the residual in the discretized PDE. This view of PDEs suggests the extension of shape optimization concepts such as B-spline and adaptive parametrizations for the solution of PDEs.

III. Theory and Algorithms

III.A. Methods for Design Space Reduction

As was previously shown, m constraints must always be enforced on the control points, where m is the dimension of the manifold. Often, the simplest approach is to prevent movement in m of the coordinate axes; however, this often does not provide sufficient design flexibility. In these cases, a more general method must be used to compute the $n-m$ directions in which movement is permitted, where n is the total number of coordinates. There are multiple options for selecting the basis for this space, and two are proposed below.

Weak Orthogonality Method One proposed method is to compute the basis for the space of allowed movement by enforcing the orthogonality of Δf_l with the m tangent vectors $\frac{\partial f_l}{\partial \xi_k}$ for $k = 1, \ldots, m$, albeit in a weak sense. This can be achieved by minimizing

$$\int_{\xi_m} \cdots \int_{\xi_1} \left[\left(\Delta f_l \frac{\partial f_l}{\partial \xi_1} \right)^2 + \left(\Delta f_l \frac{\partial f_l}{\partial \xi_2} \right)^2 + \cdots + \left(\Delta f_l \frac{\partial f_l}{\partial \xi_m} \right)^2 \right] d\xi_1 \cdots d\xi_m. \tag{3}$$

Assuming each point carries uniform weight, this is equivalent to minimizing

$$\sum_{i,j}\left(\Delta P_{ik}\frac{\partial P_{ik}}{\partial \xi_j}\right)\left(\Delta P_{il}\frac{\partial P_{il}}{\partial \xi_j}\right) = \Delta C_k\left(\sum_{i,j} B_i B_i \frac{\partial P_{ik}}{\partial \xi_j}\frac{\partial P_{il}}{\partial \xi_j}\right)\Delta C_l \tag{4}$$

with respect to a single control point, ΔC. This expression is of the form $X^T AX$, where $A = \sum_{i,j} B_i B_i \frac{\partial P_{ik}}{\partial \xi_j}\frac{\partial P_{il}}{\partial \xi_j}$ and $X = \Delta C$, and it is minimized by the solution to the linear system, $AX = 0$.

The trivial solution, $X = 0$, is not of interest, nor is any single vector X since the goal here is to find $n - m$ directions in which control point movement is to be permitted. It turns out that this information is contained in the singular vectors corresponding to the $n - m$ lowest singular values of A. Because of the physical meaning of the above linear system, the singular values of A drop significantly after the first m as the remaining $n - m$ singular vectors are not strongly represented in A. These $n - m$ orthonormal singular vectors resemble the basis of the null space of a rank deficient square matrix as they span a subspace whose vectors approximately solve $AX = 0$. In special cases such as when the curve is a straight line or the surface is a flat plane, the matrix A is mathematically singular and the $n - m$ least dominant singular vectors actually span the null space of A. In all cases, inserting one of these $n - m$ singular vectors into the original quantity, $X^T AX$, yields the corresponding singular value, which will be relatively small, confirming that we are indeed minimizing this metric and maximizing orthogonality.

Surface Normals Method The special case of $m = 2, n = 3$ corresponds to shape optimization of a surface in $\mathbb{R}^3$, which is common since it represents the physical problem of boundary shape design of an object in three-dimensional space. In this case, we can use the vector normal to the surface as the direction of allowed movement, avoiding the computational cost of the method outlined above. This normal vector can be evaluated at the location on the surface closest to the B-spline control point or at a general fixed location on the surface mapped to the control point.

The design space reduction tensor in this case is

$$D_{jkl} = \frac{\partial C_{jl}}{\partial x_k} = \begin{bmatrix} \hat{n}_1 & \underline{0} & \dots & \underline{0} \\ \underline{0} & \hat{n}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \underline{0} \\ \underline{0} & \dots & \underline{0} & \hat{n}_r \end{bmatrix}^T, \tag{5}$$

where j and k are the row and column indices, respectively, of the matrix above and r is the number of control points.

The weak orthogonality and the surface normals methods are related in that the former maximizes orthogonality over the support of the basis vector corresponding to the control point while the latter maximizes orthogonality at the point at which the maximum of the basis vector occurs. Figure 1 compares the weak orthogonality and surface normals methods for a cosine curve.

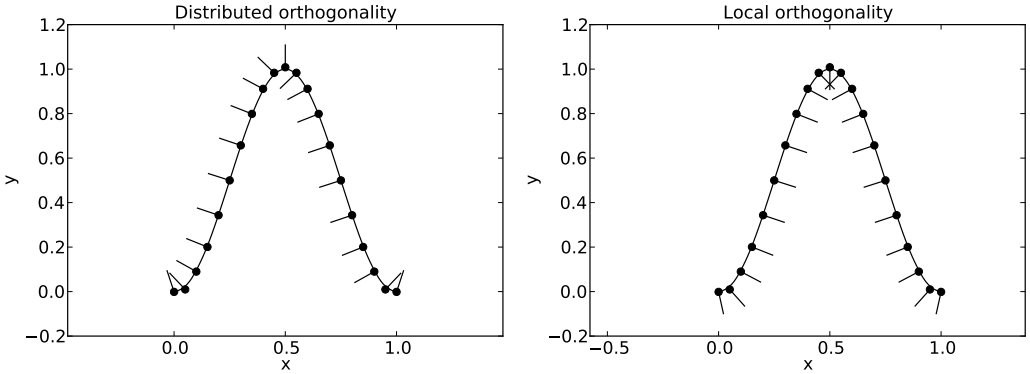

Figure 1: Control point normal vectors plotted for a simple trigonometric curve. On the left is the weak orthogonality method and on the right is the surface normals method, in the simpler case of a curve.

III.B. Parametrization Transformations

The parametrization transformations described in this section are a central part of the dynamic parametrization algorithm. Henceforth, the l index used in previous sections is collapsed into the corresponding i and j indices, using Voigt form. For a B-spline curve, a vector of points evaluated at a set of parameter values w is given by the matrix product of the basis function matrix and the vector of control points C. The basis function is nonlinear in w, giving the B-spline its order; however, the B-spline itself is linear in the control points. This is a useful feature as the Jacobian of the parametrization is simply the basis vector matrix:

$$\frac{\partial P_i}{\partial C_j} = B_{ij}. \tag{6}$$

Different parametrizations of the same curve are related through the common set of points that they represent. Thus, the Jacobian of the transformation between two parametrizations, C and $\tilde{C}$, is

$$T_{ij} = \frac{\partial C_i}{\partial \tilde{C}_j} = \frac{\partial C_i}{\partial P_k}\frac{\partial P_k}{\partial \tilde{C}_j}, \tag{7}$$

where $\frac{\partial C_i}{\partial P_k}$ is the left Moore–Penrose pseudoinverse of $\frac{\partial P_k}{\partial C_i}$. The matrix expression for the Jacobian of the transformation is given by

$$T = \frac{\partial C}{\partial \tilde{C}} = B^\dagger \tilde{B}. \tag{8}$$

Using this transformation matrix, it is possible to relate gradients and Hessians in one parametrization to another. The transformation of the gradient of the Lagrangian can be used to derive that of the Hessian of the Lagrangian, yielding

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \tilde{C}_i} &= \frac{\partial \mathcal{L}}{\partial C_k}\frac{\partial C_k}{\partial \tilde{C}_i} \\ \tilde{g} &= g B^\dagger \tilde{B} \end{aligned} \tag{9}$$

and

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial \tilde{C}_i \partial \tilde{C}_j} &= \frac{\partial C_p}{\partial \tilde{C}_i}\frac{\partial^2 \mathcal{L}}{\partial C_p \partial C_q}\frac{\partial C_q}{\partial \tilde{C}_j} \\ \tilde{H} &= \tilde{B}^T B^{\dagger T} H B^\dagger \tilde{B}. \end{aligned} \tag{10}$$

The transformation must only be made from a finer parametrization to a coarser one. If H is $m \times m$ and $\tilde{H}$ is $\tilde{m} \times \tilde{m}$, the highest rank that $\tilde{H}$ can have is m, making it a singular matrix if $\tilde{m} > m$. Transformation to a larger Hessian requires creating information, whereas transformation to a smaller Hessian simply involves project onto a lower dimensional space. Moreover, transformation to a larger Hessian would involve computing the Jacobian, $\frac{\partial C}{\partial \tilde{C}}$, of a coarse parametrization with respect to a finer one, requiring that a fine representation of a shape be fitted by a coarser parametrization, which will produce large errors.

The method of computing the transformation matrix, T, merits further discussion as it can have a large computational cost. The naive approach would be to explicitly form the pseudoinverse, $B^\dagger$, by computing the full Singular Value Decomposition or the QR factorization of B. The most efficient approach is to recognize that T is given by an overdetermined system with multiple right hand sides, described by

$$\frac{\partial P}{\partial C}\frac{\partial C}{\partial \tilde{C}} = \frac{\partial P}{\partial \tilde{C}}, \tag{11}$$

which is of the form $AX = B$, where the least-squares solution is equivalent to that of the square system, $A^T AX = A^T B$. Since the B-spline basis yields sparse matrices when discretized and the number of right hand sides is large, the best approach is to compute and use a sparse LU factorization of $A^T A$ as opposed to an iterative solver.

III.C. Quasi-Newton Algorithm with a Dynamic Parametrization

The method that follows outlines the application of the dynamic parametrization scheme to quasi-Newton SQP optimization for nonlinear, continuous, constrained problems.[16] For simplicity, 1-D shape optimization is assumed — design space reduction is not addressed here.

First, the SQP method is reviewed. An SQP optimizer applies a Newton or quasi-Newton algorithm to the KKT conditions. Sub-problems are repeatedly generated and solved in which a quadratic approximation of the objective function is used along with a linearization of the constraints.

Consider an optimization problem with objective function f, constraint functions c, and design variables x. We define the Lagrangian as $\mathcal{L} = f + c^T\lambda$, the constraint Jacobian as $A = \frac{\partial c}{\partial x}$, the objective function gradient as $g = \frac{\partial f}{\partial x}$, and the Hessian of the Lagrangian with respect to the design variables as $H = \frac{\partial^2 \mathcal{L}}{\partial x^2}$. Since the vectors of design variables, x, and of Lagrange multipliers, λ, are concatenated into a combined set of unknowns, the Newton or quasi-Newton step is computed from

$$\begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x^2} & \frac{\partial^2 \mathcal{L}}{\partial x \partial \lambda} \\ \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial x} & \frac{\partial^2 \mathcal{L}}{\partial \lambda^2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{bmatrix}. \tag{12}$$

The key to the quasi-Newton SQP method that distinguishes it from the application of an unconstrained quasi-Newton optimizer directly to the KKT conditions is the fact that we only approximate the matrix block $\frac{d^2 \mathcal{L}}{dx^2}$, since we know the other matrix blocks. Inserting the known quantities for those blocks yields

$$\begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x^2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} g + A^T\lambda \\ c \end{bmatrix}. \tag{13}$$

The order in which the Jacobian evaluation, Hessian update, and step computation are performed is illustrated in Figure 2 using the eXtended Design Structure Matrix (XDSM).[17] The thick, grey lines indicate data flow, the thin black lines indicate process flow, the numbers indicate order of execution, variables on the upper triangular indicate feed-forward, and those on the lower indicate feed-back.

For shape optimization, the only problem-dependent component is the analysis module; all other components are part of the shape optimizer. At the start of an SQP major iteration, the parametrization module is called to evaluate the discrete points describing the shape based on the current set of design variables. This vector, P, is passed to the user-supplied analysis function, which provides the objective and constraint values as well as their respective adjoint vectors. Combined with geometric sensitivities, this information is used to compute the objective function gradient and constraint Jacobian. After updating the Hessian with these new first-order sensitivities, the step direction is computed by solving the resulting linear system.

We now incorporate dynamic parametrization refinement into this algorithm. Since Hessian transformations must always reduce the size of the Hessian, we must always store and update the Hessian with respect to a finer parametrization than that at which we compute the step. Thus, the parametrization we initially define for approximating the Hessian places an upper limit on the degree to which we can refine the current parametrization. In the algorithm presented here, we keep track of two parametrizations — a fine one that is fixed at which to update and store the Hessian, and a working one which is coarser but is dynamically refined, and is the level at which we compute each step.

Practically, this approach can be implemented with only minor additions to the existing algorithm, as the bulk of the regular SQP method is executed as normal, but at the fine parametrization level. The major difference is that we transform the Newton system to the coarse parametrization immediately prior to computing the step, and we transform the computed step back to the fine parametrization. This view of dynamic parametrization refinement suggests a simple interpretation of the method; we are effectively optimizing at the fine level, though each design step is computed in a subspace that gradually increases in dimensionality after each refinement.

The other difference between the current method and the traditional SQP method is that there is an additional reparametrization stage between the Hessian update and the computation of the step. There are many options for the reparametrization scheme; intuitively, it seems appropriate to use the current adjoint information and the current set of points and place the control points where they are needed to best capture both vectors — e.g., concentrate the control points in regions of high curvature through some quantitative scheme. The SQP method with dynamic parametrization refinement is illustrated in Figure 3.

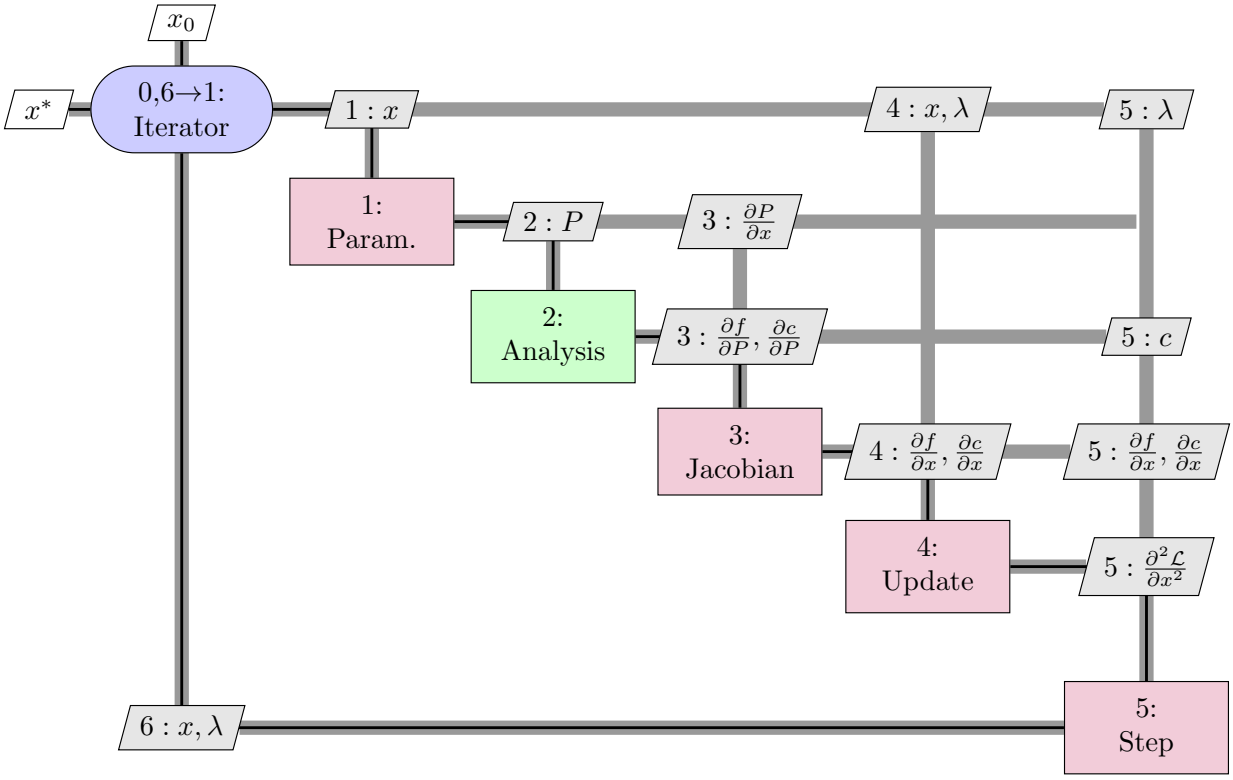

Figure 2: eXtended Design Structure Matrix[17] for quasi-Newton SQP-based shape optimization.

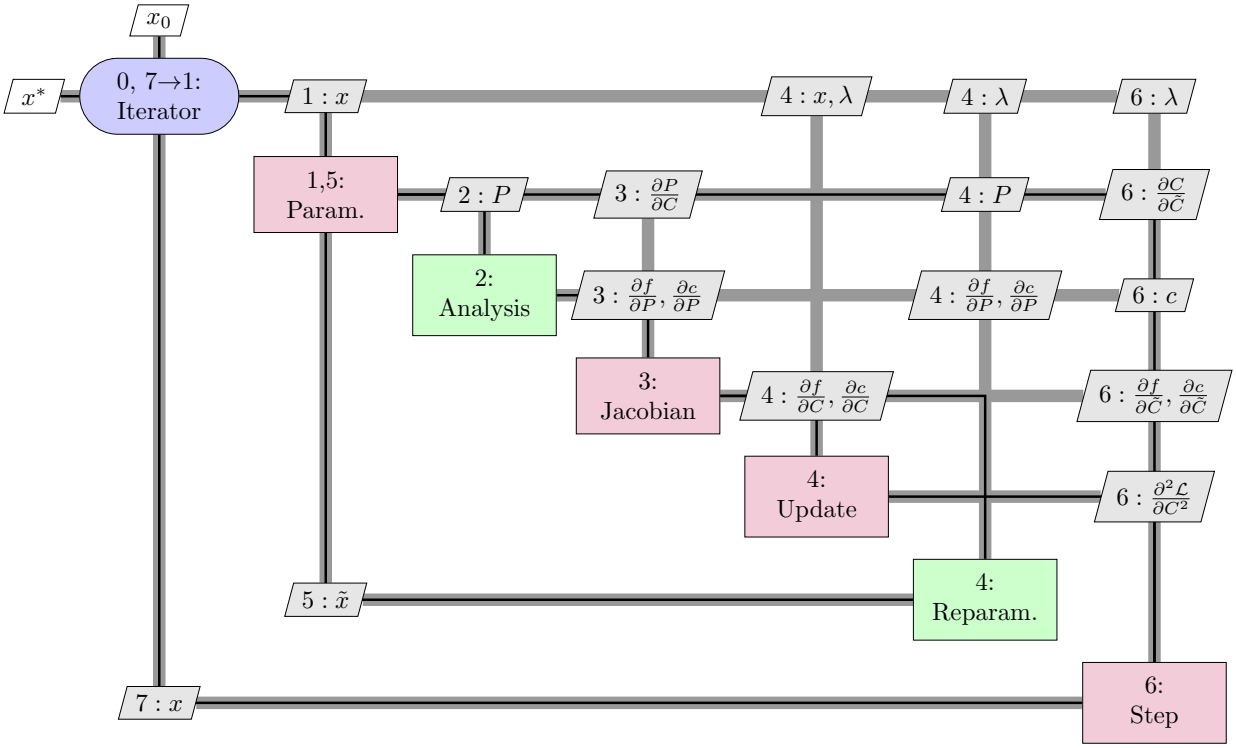

Figure 3: eXtended Design Structure Matrix[17] for the incorporation of the dynamic parametrization scheme.

IV. Model Problems

The variable parametrization algorithm presented is tested on two model problems. They are shape approximation problems, so chosen to create the simplest possible environment in which to implement the algorithms developed here and avoid the influence of the physics of the model. In other words, the idea is to isolate the parametrization scheme in the shape design problem by assuming the optimal shape is known and evaluating the ability of the shape optimization algorithm to find it.

IV.A. Shape Approximation of a Curve

Here, we are optimizing a curve in $\mathbb{R}^2$ with one dimension eliminated, to compute the optimal y coordinates of uniformly spaced B-spline control points. A least squares minimization problem is solved using a BFGS quasi-Newton optimizer and a backtracking line search with a sufficient decrease condition.[16] The objective function is the square of the l^2-norm of the differences between the y coordinates yielded by the B-spline and those of the analytical function to be approximated, $y = \tanh(3\pi x)$, in the interval, $[0, 1]$.

Five shape optimization algorithms are compared:

1. 'Coarse' — Fixed parametrization with 4 control points
2. 'Fine' — Fixed parametrization with 20 control points
3. 'Restarted refinement' — Adaptive parametrization with the quasi-Newton algorithm restarted after each refinement
4. 'Uniform refinement' — Uniform-spacing parametrization refinement with Hessian transformations
5. 'Dynamic refinement' — Adaptive parametrization with Hessian transformations

The curve is discretized with 500 points and each of the variable parametrization cases begins with 4 control points and adds 4 control points when a gradient norm of 10^{-6} is reached until a maximum of 20 control points and a gradient norm of 10^{-12} is reached. For the Hessian transformation cases, the approximate Hessian is stored with respect to a parametrization with 100 control points. The adaptive parametrization schemes redistribute control points such that the arc lengths of segments of the curve between control points are equal. The effect is that control points are more concentrated in x wherever the current curve has higher slope. This is shown in Figure 4.

Figure 5 illustrates the effectiveness of the dynamic parametrization scheme with Hessian transformations. The objective function is used as the metric for convergence since all of the schemes parametrize the same discrete representation of the curve, independent of the number of design variables. This is plotted against the sum of function and gradient evaluations since this algorithm is designed for adjoint-based optimization with large-scale nonlinear systems of equations. With the adjoint method, the cost of a gradient evaluation is expected to be lower than a function evaluation, since the latter involves solving a nonlinear system as opposed to a linear system in CFD, for example. However, the cost of a function evaluation is reduced in practice, since the previous set of solved state variables can be used as the initial guess for the nonlinear solver. Thus, it is assumed that function and gradient evaluations are roughly equivalent in cost for the problems for which this algorithm is intended, making their sum a good metric for computational cost.

As expected, Figure 5 shows that the coarse parametrization case converges in the fewest number of function and gradient evaluations but achieves a much higher objective function value at the optimum than the other cases. The fine parametrization requires more than double the number of evaluations, but significantly improves the quality of the optimal curve. The adaptive, restarted algorithm is able to improve the quality of the optimal curve further while finishing with the same number of control points as the fine parametrization case. However, it expends double the computational effort because of the iterations required to rebuild the Hessian approximation after each restart. The uniform refinement with Hessian transformations case does not suffer from this problem. Since the Hessian approximation is continuously built up in this case, it is no surprise that this case and the fine parametrization case reach the quadratric neighborhood of the optimum in roughly the same number of evaluations — they both require 20 gradient updates to obtain a full approximation of the 20×20 Hessian. The adaptive parametrization with Hessian transformations yields the most accurate optimal shape, in fewer evaluations than all but the coarse parametrization. Figure 6 shows the objective function and gradient norm histories for this last case.

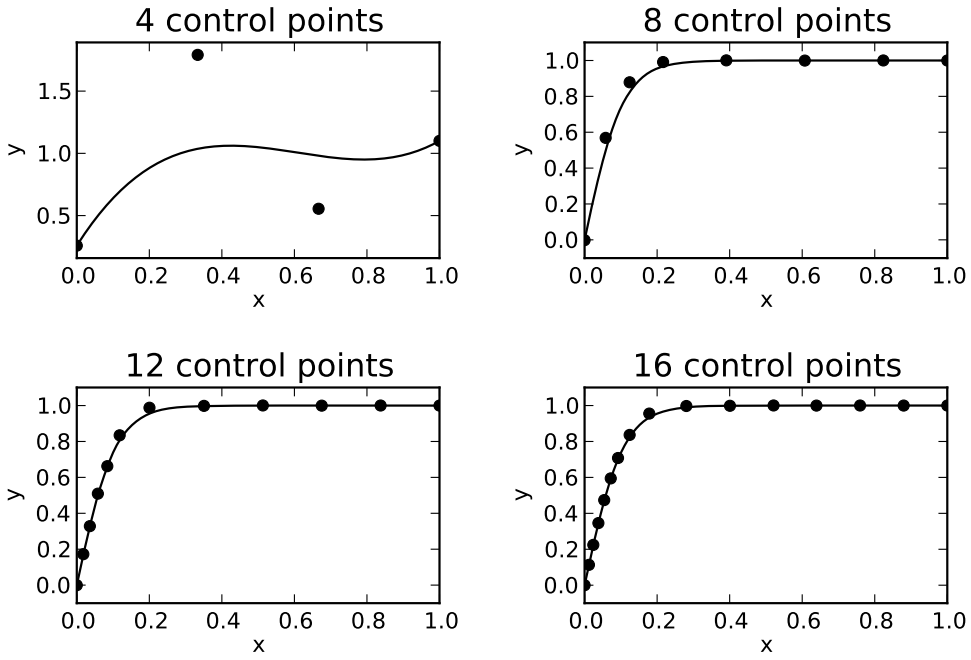

Figure 4: Snapshots of the shape during an adaptive parametrization shape optimization. The curve being approximated is the hyperbolic tangent function. Each snapshot corresponds to the partially converged shape just prior to the next parametrization refinement. Note the x-positions of the control points — a constant arc length condition is used to determine the new locations of the control points for each refinement (the plot is not to scale).

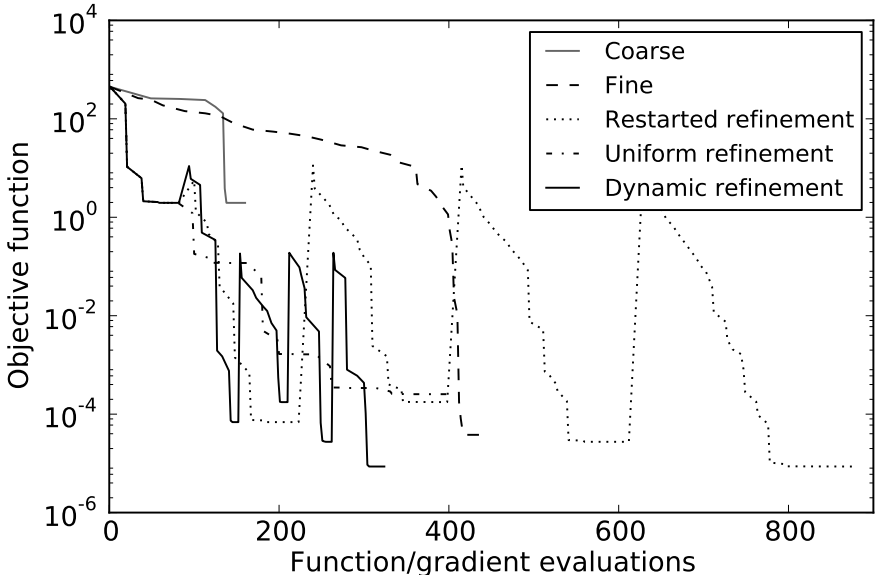

Figure 5: Comparison of the convergence histories of the various parametrization schemes. 'Coarse' and 'Fine' refer to fixed parametrizations with 4 and 20 control points, respectively. The 'Restarted refinement' case uses an adaptive parametrization scheme, but the quasi-Newton optimizer is restarted after each refinement. The 'Uniform refinement' refers to the use of Hessian transformations to maintain the Hessian approximation through the refinement, but the refinement redistributes control points uniformly. 'Dynamic refinement' refers to the same but with adaptive instead of uniform refinements.

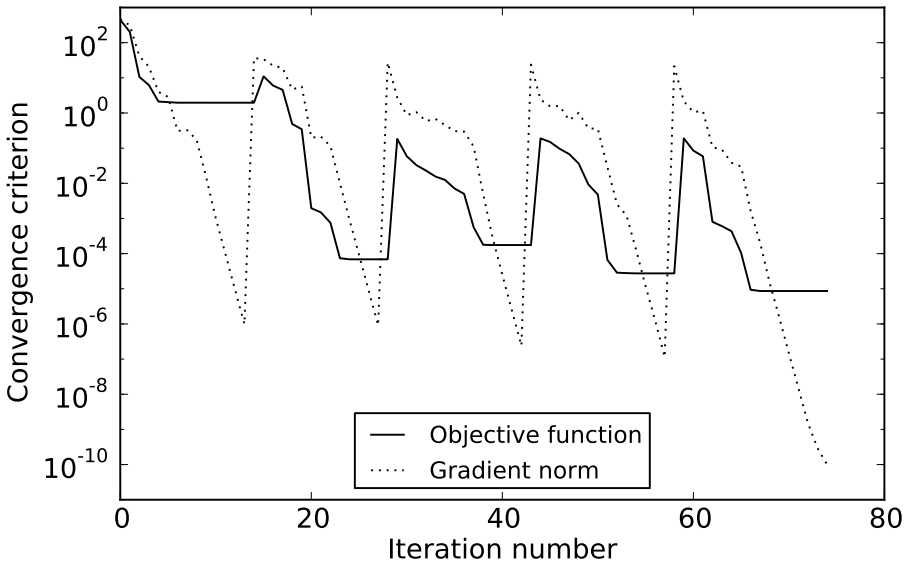

Figure 6: Convergence history of the adaptive refinement scheme with Hessian transformations.

IV.B. Shape Approximation of a Free-Form Surface

In the second model problem, a surface is optimized using a free-form parametrization. The goal is to morph a sphere into a cube or an ellipsoid using a shape approximation formulation. It is assumed that the target shape has an unknown, unstructured representation. The objective function is the sum of the squares of the differences between each point on the target shape and the nearest point on the surface being optimized. This creates a C^1 discontinuous function whenever the nearest point changes and possibilities for unphysical solutions because of a subset of points not being accounted for; however, it is sufficient as a model problem and provides a simple test problem for free-form shape design. Because of the discontinuities in the function gradient, the approximate Hessian is updated at every iteration using information from only the last ten iterations. The expectation is that close to the optimum, the nearest point for each point on the target shape will not change.

Applying symmetry, only half of the sphere is parametrized, using a patchwork of 5 B-spline surfaces discretized with a total of 4564 unique points. C^1 continuity is enforced at all boundaries between surfaces to facilitate the definition of surface normals. Control points are permitted to move only along the normal vector from the closest point on the surface. The parametrization is refined with a uniform distribution of points after convergence to a gradient norm of 10^{-1}.

In the first case, the sphere is morphed into an ellipsoid, which is a challenge for the algorithm because the surface normals allow for local shape changes, but are stiffer with respect to changes in the overall aspect ratio of the shape. An adaptive parametrization such as the constant arc length condition would address this difficulty, but it is not implemented here. For this problem, a relaxation factor on the Newton step size is found to be more effective than a backtracking line search in ensuring robust convergence.

In the second case, the sphere is morphed into a cube, which is difficult because of the sharp edges and corners. An adaptive parametrization would make a significant impact in this problem as well by concentrating more control points at the edges and corners and enabling higher curvatures. Since the control points are only permitted to move normal to the surface, large oscillations appear at the sharp edges, resembling Runge's phenomenon in interpolation. Convergence and final shape plots are shown in Figures 7 and 8, respectively.

Unlike the 1-D curve optimization problem, the objective here is not to compare different parametrization schemes; rather, it is to demonstrate that free-form shape optimization is feasible with adjoint-based quasi-Newton optimization using parametrization transformations and design space reduction. The enabling aspect of the variable parametrization approach is its ability to avoid local minima in early optimization iterations by only permitting smooth, low-order changes with an initially coarse parametrization. As the optimization

progresses, more degrees of freedom are added gradually, until the finest level of refinement. If a fixed parametrization with a large number of design variables is used, it is difficult to avoid an irreversible, catastrophic design step that results in an unphysical shape. The only alternative for fixed parametrizations is to rely on heavy relaxation, which results in a less accurate Hessian approximation and unreasonable computation times.

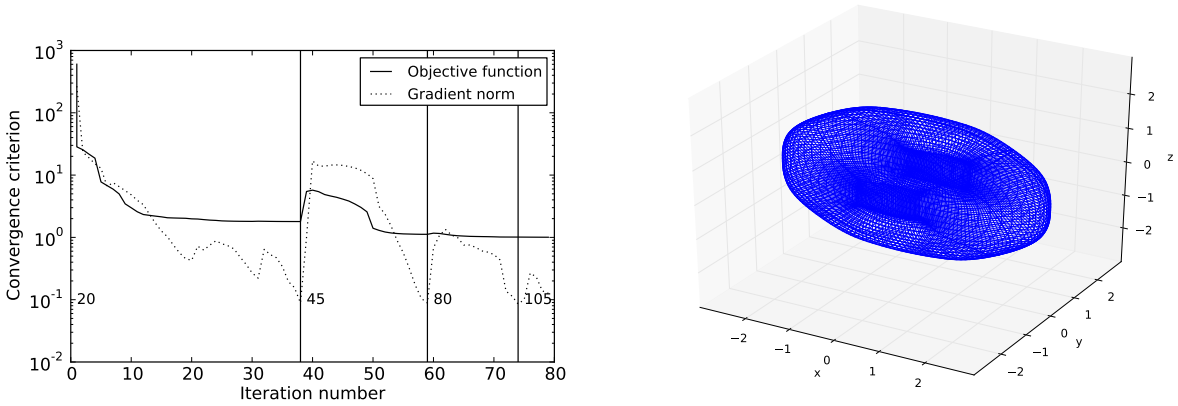

Figure 7: The free-form shape approximation problem in which a sphere morphs into an ellipsoid. On the left is the convergence history and on the right is the final shape. In the plot on the left, the small numbers indicate the total number of control points after each reparametrization.

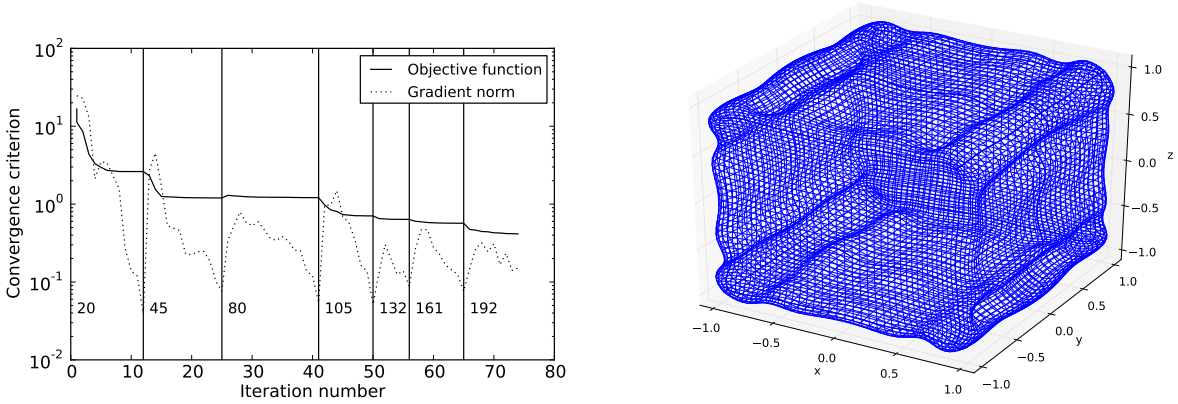

Figure 8: The free-form shape approximation problem in which a sphere morphs into an cube. On the left is the convergence history and on the right is the final shape. In the plot on the left, the small numbers indicate the total number of control points after each reparametrization.

V. Conclusion

The primary contribution of this work is the development and demonstration of a variable parametrization shape design algorithm that is compatible with quasi-Newton optimization. The premise is that adjoint-based quasi-Newton optimizers represent the only feasible approach for high fidelity shape optimization, but previous adaptive parametrization schemes are not compatible with this approach. A 1-D test problem showed that incorporating adaptive refinement reduces the number iterations and improves the quality of the optimum, while a 3-D test problem demonstrated the feasibility of free-form shape optimization using parametrization refinement. Beyond applying these algorithms to practical engineering problems, promising avenues for future work include in-depth study of refinement techniques, particularly ones incorporating sensitivity information, and applying adaptive refinement to the 3-D free-form shape optimization problem.

References

[1]Haftka, R. T. and Grandhi, R. V., "Structural shape optimization — A survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, No. 1, 1985, pp. 91–106.

[2]Espiga, F., Gracia, L., and Doblare, M., "Shape optimization of elastic homogeneous 2D bodies by the boundary element method," *Computers and Structures*, Vol. 33, No. 5, 1989, pp. 1233.

[3]Dang, T. D., Kapania, R. K., Slemp, W. C. H., Bhatia, M., and Gurav, S. P., "Optimization and Postbuckling Analysis of Curvilinear-Stiffened Panels Under Multiple-Load Cases," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1656–1671.

[4]Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet," *Journal of Aircraft*, Vol. 41, No. 3, 2004, pp. 523–530.

[5]Percival, S., Hendrix, D., and Noblesse, F., "Hydrodynamic optimization of ship hull forms in shallow water," *Applied Ocean Research*, Vol. 23, No. 2, 2004, pp. 51–62.

[6]Muyl, F., "Hybrid method for aerodynamic shape optimization in automotive industry," *Computers & Fluids*, Vol. 33, No. 5-6, 2004, pp. 849–858.

[7]Akcelik, V., Biros, G., Ghattas, O., Keyes, D., Ko, K., Lee, L.-Q., and Ng, E. G., "Adjoint methods for electromagnetic shape optimization of the low-loss cavity for the International Linear Collider," *Journal of Physics Conference Series*, Vol. 16, No. 1, 2005, pp. 435–445.

[8]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance Control and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[9]Kohli, H. S. and Carey, G. F., "Shape optimization using adaptive shape refinement," *nternational Journal for Numerical Methods in Engineering*, Vol. 36, No. 14, 1993, pp. 2435–2451.

[10]Desideri, J.-A., El Majd, B. A., and Janka, A., "Nested and self-adaptive Bezier parameterizations for shape optimization," *Journal of Computational Physics*, Vol. 224, 2007, pp. 117–131.

[11]Nagy, A. P., Abdalla, M. M., and Gurdal, Z., "Isogeometric sizing and shape optimisation of beam structures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, No. 17-20, 2010, pp. 1216–1230.

[12]Han, X. and Zingg, D. W., "An Evolutionary Geometry Parametrization for Aerodynamic Shape Optimization," June 2011, AIAA-2011-3536.

[13]Zingg, D., Nemec, M., and Pulliam, T. H., "A Comparative Evaluation of Genetic and Gradient-Based Algorithms Applied to Aerodynamic Optimization," *Shape design in aerodynamics REMN – 17/2008.*, 2008, pp. 103–126.

[14]Hicken, J. E. and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 400–413.

[15]Kennedy, G. J. and Martins, J. R. R. A., "Aerostructural design optimization of composite aircraft with stress and local buckling constraints using an implicit structural parametrization," April 2011.

[16]Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer, 2nd ed., 2006.

[17]Lambe, A. B. and Martins, J. R. R. A., "A Unified Description of MDO Architectures," June 2011.