A Sparse Superlinearly Convergent SQP with Applications to Two-dimensional Shape Optimization

Mihai Anitescu*†
Mathematics and Computer Science Division, Argonne National Laboratory
9700 South Cass Avenue, Argonne, Illinois 60439
E-mail: anitescu@mcs.anl.gov

Radu Serban
Department of Mechanical Engineering, The University of Iowa
ERF 229, Iowa City, Iowa 52242
E-mail: rserban@ccad.uiowa.edu

February 12, 1998

Abstract

Discretization of optimal shape design problems leads to very large nonlinear optimization problems. For attaining maximum computational efficiency, a sequential quadratic programming (SQP) algorithm should achieve superlinear convergence while preserving sparsity and convexity of the resulting quadratic programs. Most classical SQP approaches violate at least one of the requirements. We show that, for a very large class of optimization problems, one can design SQP algorithms that satisfy all these three requirements. The improvements in computational efficiency are demonstrated for a cam design problem.

*Address all correspondence to this author.

†The work of this author was supported by the Mathematical, Information and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

1 Introduction

Within the class of potentially very large scale problems, shape optimization occupies an important place, being an essential part of the design of structures and mechanisms. The reduction of the continuous problem to a finite one via discretization or spline function approximations (Braibant and Fleury 1984) can lead to a very large nonlinear constrained optimization problem (NLP). Because of its origin, however, this NLP can be extremely sparse, as measured by the fill-in that appears in the rows of the Jacobian of the resulting constraints. In fact, since most relations that need to be satisfied by the continuous problem are local (involving an unknown function at a point and its derivatives), the finite problem constraints will have nonzero entries only at a very small set of neighboring points. It is therefore desirable to design an algorithm for the NLP that takes advantage of this structure and still has good convergence properties.

This is the requirement that motivates our work. One of the traditional ways of solving the NLP is sequential quadratic programming (SQP) (Bertsekas 1982). At each step of this iterative procedure, a constrained quadratic program is solved. The desirable features of such an algorithm would be to achieve fast convergence, to generate easy-to-solve quadratic programs, and to preserve sparsity for fast linear algebra resolution. Fast convergence is usually associated with superlinear convergence (Bertsekas 1982). A "not difficult" quadratic program is one that has a positive semidefinite matrix in the objective function because the resulting problem is convex. Results from the past decade show that such quadratic programs have only polynomial complexity (Wright 1997), and are easy to solve from the viewpoint of complexity theory.

Unfortunately, traditional SQP approaches do not achieve, at the same time, superlinear convergence and convexity and sparsity of the resulting quadratic program (QP). Using the Hessian of the Lagrangian of the NLP at the current point results in a QP that is sparse but not generally convex. In Betts and Frank (1994) sparsity and convexity of the QP are preserved by using the exact Hessian of the Lagrangian with a diagonal modification, but the rate of convergence is reduced to a linear one. Although an acceleration can be observed by adaptively reducing the diagonal perturbation to zero, it is not clear whether superlinear convergence and convexity of the QP can be guaranteed under the usual assumptions. Finally, Powell's method (Powell 1978), based on the BFGS formula, will result in a convex QP and will exhibit superlinear convergence (with some line search modifications; see Bertsekas 1982) but it will destroy the sparse pattern of the problem.

In this paper we show that a very large class of NLPs, many of which originate in the discretization of optimal shape design optimization problems, can lead to superlinearly convergent SQPs with convex and sparse QPs. The key observation is that these NLPs typically have far more total constraints (equality and inequality) than unknowns, and it is therefore likely that there will be as many active constraints as variables. Therefore, near convergence, the sequence approaches the behavior of the Newton step for the nonlinear equation corresponding to the satisfaction of the active constraints, regardless of the matrix used in the QP. A constant, sparse positive definite matrix will be enough to ensure the convexity and sparsity of the resulting QP.

An SQP algorithm, implemented in Matlab, that solves the QP based on an interior-point technique (Vanderbei 1994) with sparsity support is used for the NLP that results from the discretization of a cam design problem. Comparisons between this approach and Powell's algorithm are provided, as well as between the interior-point and Matlab QP resolutions.

2 The Optimization Problem

This section describes the class of problems under consideration and sufficient conditions to obtain superlinearly convergent SQPs with convex and sparse QPs.

2.1 The Continuous Formulation

The target problem in our case has a finite number of equality constraints, and an infinite number of inequality constraints, indexed by a real variable whose domain is a finite, closed interval. Since we wish to incorporate two-dimensional shape optimization as part of our model, we also include a shape function, $y(t)$, as a variable in the inequalities to be satisfied. Extension to multidimensional shape optimization is similar, and we therefore restrict our attention to just one shape parameter. Let $t \in [a, b]$ be the parameter of the shape. Let F be a finite set of points in $[a, b]$. Let D be a domain in R^m. The problem that we consider is as follows:

$$\min \int f(y(t), t, x) + \sum_{t_i \in F} g(t_i, x) \quad x \in \mathcal{D} \tag{1}$$

$$\sum_{j=1}^{N_i} a_{ij}(x,t) y^{(j)}(t) + f_i(x, t, y(t)) \leq 0 \quad t \in [a,\, b], \quad i = 1..p \tag{2}$$

$$g_i(x, t_i, y(t_i)) = 0 \quad t \in F, \quad i = 1..p. \tag{3}$$

We wish to include derivatives of the shape function, y, in the formulation, because several meaningful constraints are based on them. A good example is the convexity of the shape constraint, which could involve the shape function and its first two derivatives (depending on the formulation).

2.2 The Discretized Optimization Problem

The main emphasis of this work is on the treatment of the nonlinear optimization problem that results from discretizing (1). The resulting problem has the familiar form

$$\min f(x) \tag{4}$$

$$g_i(x) \leq 0, \quad 1 \leq i \leq n \tag{5}$$

$$h_i(x) = 0, \quad 1 \leq i \leq p. \tag{6}$$

For the targeted class of problems, the constraints have a particular structure. There are few equality constraints corresponding to the set F, compared with the constraints originating from the discretization of the inequality constraints (2). Since the original constraints (2) are local constraints (for a given point t_i. only the values of $y(t)$ and its derivatives at t_i enter the constraint equations), the resulting inequality constraints at a discretization node t_i, $g_i(x)$. depend only on the value of y at t_i and a small, fixed length set of neighboring points. The Jacobian matrix of the constraints is therefore very sparse. having an almost banded structure.

Since the inequality constraints in (4) originates in (2), their number is at least the number of discretizing points. We assume that (4) is such that the number of constraints *exceeds* the number of variables. This assumption is reasonable, especially if there are at least two inequality constraints in (2).

We also invoke the standard assumption that x^* is *regular*, in other words, that the Jacobians of the equality constraints and the active constraints are linearly independent at x^*.

Assumption E Our major assumption is that the total number of active constraints (equality constraints plus active inequality constraints) is exactly equal to the total number of variables. This condition is difficult to check on (1), but can be observed to hold in most applications of interest. One explanation is that, in most cases, the objective function in (1) is a functional that depends on the values of $y(t)$ at all points t (one way to rigorously define that is with respect to the Frechet derivative). If there were fewer constraints active than the number of variables, a direction of descent could be found for this case.

2.3 The Sequential Quadratic Programming Algorithm

The method used here to solve (4) is the sequential quadratic programming (SQP). At each point x_k the following problem is solved:

$$\min \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \tag{7}$$

$$\nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad i = 1..n \tag{8}$$

$$\nabla h_i(x_k)^T d + h_i(x_k) = 0 \quad i = 1..p. \tag{9}$$

An Armijo-rule-based line search is used to determine $x^{(k+1)}$. As a penalty function to measure the extent of infeasibility, we use the usual exact penalty function

$$P(x) = \max\{g_i(x), |h_j(x)|, i = 1..n, j = 1..p\}. \tag{10}$$

The line search evaluates decreases of the function $f(x) + cP(x)$, where c is a sufficiently large constant that is nondecreasing at each iteration, although it becomes constant after a sufficiently big k. As a variant of this method one can solve an additional quadratic program of the same form as (7) and do an arc search instead of a line search, in order to guarantee superlinear convergence. For details, see (Bertsekas 1982).

However, the most important factor in securing superlinear convergence is the choice of the H_k matrix. Let $L(x^*, \lambda^*)$ be the Lagrangian function associated with the program (4). To obtain superlinear convergence, the matrix H_k has to be "close" in some sense to $\nabla^2_{xx}L(x_k, \lambda_k)$. In addition, there are two desirable properties that H_k should satisfy. It should be positive definite to ensure that (7) can be reasonably easy to solve (since a quadratic program with a matrix that is not positive definite is NP hard). Also it H_k should be sparse for computational efficiency.

A choice is $H_k = \nabla^2_{xx}L(x_k, \lambda_k)$. The sparsity of this problem is preserved, but there is no guarantee that H_k will be positive definite. Another classical choice is based on the BFGS rank one update (Powell's algorithm) Bertsekas (1982). Although this will generate a positive definite matrix H_k, the rank-one update will destroy the sparsity. Therefore, neither of these choices will lead to a program (7) that is both sparse and convex.

Our goal is to determine whether it is possible to obtain SQP with convex and sparse QPs that achieve superlinear convergence, under the assumptions set forth at the end of the preceding subsection.

Theorem 2.1 *Assume that the solution x^* is a regular point of the constraints of (4) and that the number of equality constraints plus the number of active inequality constraints equals the number of unknowns in (4). Let H_k A. $\forall k$, any constant matrix. Then, if $x_k \longrightarrow x^*$, and the step length allowed by the penalty function is at least unity for all k sufficiently large, the convergence is superlinear.*

Proof A sufficient condition to obtain superlinear convergence is to be able to take unit steps along d_k that decrease the penalty function and to ensure that the sequence H_k be uniformly bounded, positive definite on the column span of Z^* and satisfy

$$\lim_{k \to \infty} [H_k - \nabla^2_{xx}L(x_k, \lambda_k)]Z^*. \tag{11}$$

Here Z^* is matrix whose columns are a base for the nullspace of the Jacobian matrix of the active constraints (Bertsekas, Prop 4.32, 1982).

Since x^* is regular, the Jacobian of the active and equality constraints has full row rank. By Assumption **E**, it follows that this Jacobian is square and invertible. Therefore, its nullspace is 0, or $Z^* = 0$. It is immediate that any constant sequence A satisfies all the requirements on H_k for superlinear convergence of x_k. $\nabla\nabla\nabla$

One way to ensure that the step length is unity for all k sufficiently big is to solve an additional QP and to do an arc search (Bertsekas 1982). Since this is not the focus of our investigation, we simply assume the stepsize to be unity for all sufficiently big k.

An interesting conclusion is that sequential linear programming (A=0) will actually achieve superlinear convergence under these conditions. The cases of interest are, of course, those for which A is positive semidefinite, resulting in a convex and sparse QP (7). In our experiments we choose $H_k = 0$ and $H_k = I$.

The fact that the superlinear convergence does not depend on the choice of the matrix A might appear surprising. In reality, the fact that the Jacobian of the constraints is invertible (it is square and full row rank) constrains the problem to such an extent that the direction found by (7) is actually determined almost in completely by the constraints. Thus, for sufficiently big k, the method behaves like Newton's method: it solves the nonlinear system that requires the equality constraints and active equality constraints to be equal to zero. However, far from the solution, (7) ensures that the descent of the penalty function and guarantees good global behavior.

In all fairness, we must emphasize that Assumption **E** is almost impossible to check on the initial problem (1) or its discretization (4). This assumption, however, is expected to hold in most cases of interest, especially when the objective function in (1) presents some uniformity with respect to the values of the shape function $y(t)$.

3 Numerical Experiments

As an example, consider the problem of designing the shape of a cam. Although simple, this example offers the possibility to test different theoretical issues related to the optimization procedure. The objective of this example is to maximize the area of the valve opening for one rotation of the cam. The variables of the optimization problem are the m values $r_k, k = 1, ..., n$ defined in Fig. 1. The shape of the cam is assumed to be circular over an angle of $\frac{6}{5}\pi$ of its circumference, with radius R, and the m radii r_k representing the design parameters are equally distributed over an angle of $\frac{4}{5}\pi$.

Assuming a simple, linear relationship between the shape of the cam and the valve opening area yields the following objective function:

$$f = -\pi R_v^2 \sum_{i=1}^{m} r_i, \tag{12}$$

where R_v is a constant related to the geometry of the valve. Note that the expression of f involves all the r_i's (all portions of the shape intervene in the objective function). Assumption **E** therefore is expected to hold. If there are too few inequalities active, there might be r_i's that do not appear in any active constraint. But then, f would be unbounded.

A number of p constraints written as

$$r_{\frac{m-p}{2}+k} = R_t, \qquad k = 1, ..., p, \tag{13}$$

require the tip of the cam to be on a circle of radius R_t. Additional constraints enforce convexity of the optimal shape and limit its curvature. With the notations of Fig. 1, the convexity constraints are equivalent to the requirement that the sum of the areas of triangles $OA_{k-1}A_k$ and OA_kA_{k+1} is larger than the area of the triangle $OA_{k-1}A_{k+1}$. In terms of the design parameters r_k, these constraints become

$$-r_k r_{k+1} - r_k r_{k-1} + 2r_{k-1}r_{k+1}\cos(\Delta\theta) \leq 0, \qquad k = 2, ..., m-1 \tag{14}$$

where $\Delta\theta = 0.8\pi/(m-1)$ is the angle between two consecutive radii. Additional convexity constraints are imposed at r_1 and r_m, as well as at two fictitious points r_{-1} and r_{m+1}:

$$\begin{aligned} -r_1r_2 - Rr_1 + 2Rr_2\cos(\Delta\theta) &\leq 0 \\ -Rr_m - r_{m-1}r_m + 2Rr_{m-1}\cos(\Delta\theta) &\leq 0 \\ -Rr_1 - R^2 + 2Rr_1\cos(\Delta\theta) &\leq 0 \\ -R^2 - Rr_m + 2Rr_m\cos(\Delta\theta) &\leq 0 \end{aligned} \tag{15}$$

Curvature is controlled through the maximum allowed variation in consecutive radii, that is,

$$\left(r_{k+1} - r_k\right)^2 - \left(\alpha(\Delta\theta)\right)^2 \leq 0, \qquad k = 1, ..., m-1, \tag{16}$$

with α a given constant.

The following default values were used for the model constants:

$$m = 101 \text{ and } 401, p = 3, R = 1.0, R_v = 1.0, R_t = 2.0, \alpha = 1.5. \tag{17}$$

Since a feasible initial estimate of the design parameters is difficult to obtain, the cam is initially considered to be a circle of radius R. Optimal solutions obtained for different numbers of design parameters are presented in Fig. 2. The solid line solution is obtained for $m = 101$ design parameters, while the dashed line solution corresponds to $m = 401$.

Clearly, an increased number of design parameters lead to more stringent curvature constraints (by decreasing the value of $\Delta\theta$) resulting in a solution with lower optimal cost function. Because of the discrete nature of the problem solved, the cam will still exhibit corners, which become less prominent as $\alpha \to 0$. If needed, the shape can be smoothed by spline interpolation.

Figure 3 shows the influence of the coefficient α in (16) on the shape of the optimal cam. Results are presented for $\alpha = 1.5$ (solid line) and $\alpha = 2.0$ (dashed line). Larger values of the cost function can be obtained by increasing the value of α, which corresponds to milder curvature constraints.

In order to prove the theoretical observations of Section 2 related to the advantages of interior-point methods and sparse solvers, two different quadratic programming algorithms were used. The first one is provided within Matlab and uses an active set strategy, similar to the one described by Gill et. al. (1981). The second one, Loqo (Vanderbei 1994 and 1997), is an interior-point algorithm that uses a one-phase primal-dual path-following method. Table 1 presents the evolution of the norm of the Newton direction, using the interior-point algorithm, with $m = 101$. The following three cases are considered:

- $H = 0$ - equivalent to a sequentially linear; programming method;
- $H = I$;
- Rank-one updated QP matrix (Powell's algorithm, without the correction for the arc search).

As noted in Section 2, superlinear convergence is obtained in all three cases. However, significant efficiency improvements are obtained when the QP matrix is constant (either zero or identity) when compared with the case in which the matrix is updated. This is due to the loss of sparsity generated by the rank-one update. Table 2 compares the CPU time (in seconds), as reported by Matlab, spent in the QP solver in each of the above three cases. The advantage of using interior-point methods is highlighted by the time required to solve the same three cases by using the quadratic programming method available in Matlab (see Table 2).

Similar results are obtained when the number of design variables is increased to $m = 401$. In this case, however, the active-set algorithm from

Matlab failed to converge. This is due to the very large number of active constraints, which in this case becomes $2m + p + 1 = 806$. Results obtained by using the Loqo algorithm are presented in Table 3.

4 Conclusions

We have proposed a simple SQP algorithm that achieves superlinear convergence for a class of problems while generating convex and sparse quadratic programs for improved computational performance. As an additional advantage, only first derivative information is used. The assumption here is that the number of active constraints will equal the total number of variables at the optimal point. The SQP simply uses a constant positive semidefinite matrix for H_k at each step. The cases tested were $H_k = 0$ and $H_k = I$. As expected, there were no major differences between the sequences of iterates in the two, since, near the solution, both behave like a Newton method for the nonlinear system made of the equality and active inequality constraints.

For all methods used, superlinear convergence, as well as a number of active constraints equal to the number of variables, were observed. Therefore, Assumption **E** did hold, as assumed. However, the computing time needed for the solutions of the QP has been almost an order of magnitude less for our approach compared with the case involving the rank-one updates from BFGS. The interior-point algorithm used presents several orders of magnitude performance improvement over the algorithm provided by Matlab.

Nevertheless, it is difficult to check whether a given problem satisfies Assumption **E**, although some guidelines can be followed. Future work will include investigating the possibility of relaxing some of the assumptions used in this paper and the use of similar algorithms for dynamics-based cam design.

References

Betts, J. T., and Frank P. D., 1994, A Sparse Nonlinear Optimization Algorithm, Journal of Nonlinear Optimization Theory and Applications, Vol. 82, No. 3, pp. 519–541.

Bertsekas. D. P., 1982, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, N.Y.

Braibant, V. and Fleury, C., 1984, Shape Optimal Design Using B-splines, Computer Methods in Applied Mechanics and Engineering, Vol. 44, pp. 247–267.

Fletcher, R., 1987, Practical Methods of Optimization, John Willey & Sons, N.Y.

Gill, P. E., Murray, W., and Wright, M. H., 1981, Practical Optimization, Academic Press, N.Y.

Powell, M. J. D., 1978, The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations, Nonlinear Programming 3 (O. L. Mangasarian, R. Meyer and S. Robinson, eds.), pp. 27–63, Academic Press, N.Y.

Vanderbei, R. J., 1994, LOQO: An Interior-Point Code for Quadratic Programming, Technical Report SOR-94-21, Statistics and Operations Research, Princeton University, Princeton.

- - - , 1997, LOQO User's Manual - Version 3.10, Technical Report SOR-97-08, Statistics and Operations Research, Princeton University, Princeton.

Wright, S. J., 1997, Primal-Dual Interior Point Methods, SIAM, Philadelphia.

Zhou, J. L. and Tits, A. L., 1996, An SQP Algorithm for Finely Discretized Continous Minimax Problems and Other Minimax Problems with Many Objective Functions, Siam Journal of Optimization, Vol. 6, No. 2, pp. 461–487.

Table 1: Convergence Analysis

	Quadratic Matrix		
Iteration	$H = 0$	$H = I$	Rank-1 update
1	4.596750	4.596745	4.596745
2	$8.025247 \cdot 10^{-1}$	$8.025299 \cdot 10^{-1}$	$8.025299 \cdot 10^{-1}$
3	$2.860296 \cdot 10^{-2}$	$2.860317 \cdot 10^{-2}$	$2.860336 \cdot 10^{-2}$
4	$8.256211 \cdot 10^{-4}$	$8.256270 \cdot 10^{-4}$	$8.255900 \cdot 10^{-4}$
5	$4.890005 \cdot 10^{-7}$	$4.898711 \cdot 10^{-7}$	$4.9219911 \cdot 10^{-7}$
6	$5.246231 \cdot 10^{-11}$	$2.973782 \cdot 10^{-11}$	$1.070026 \cdot 10^{-11}$

Table 2: Efficiency Analysis, $m = 101$

	Quadratic Matrix		
	$H = 0$	$H = I$	Rank-1 Update
Loqo	0.30	0.32	2.47
Matlab	124.84	145.22	151.27

Table 3: Efficiency Analysis, $m = 401$

	Quadratic Matrix		
	$H = 0$	$H = I$	Rank-1 update
Loqo	3.05	3.10	28.51

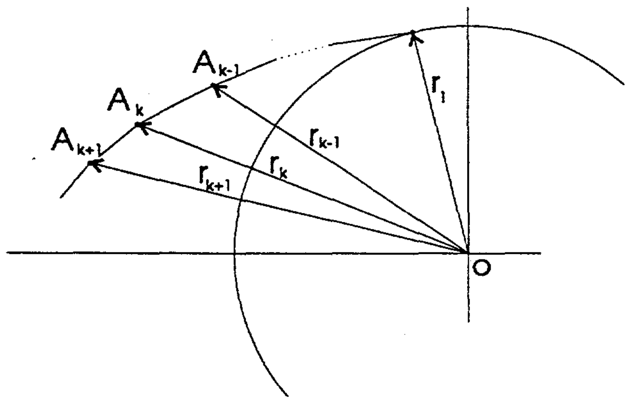

Figure 1: Definition of Design Parameters

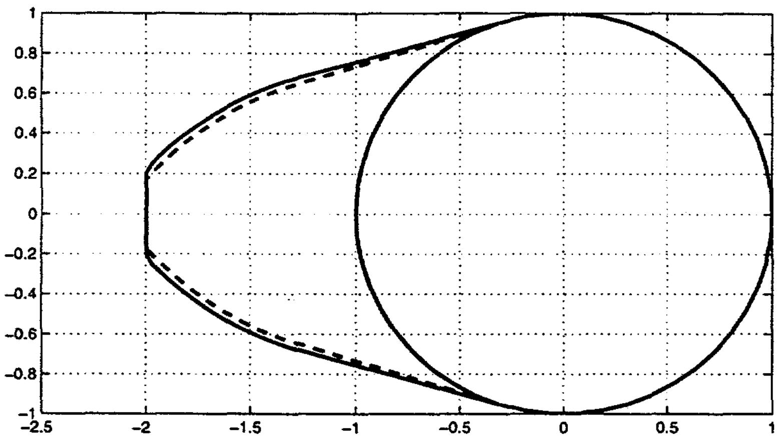

Figure 2: Influence of the Number of Design Parameters on the Optimal Shape

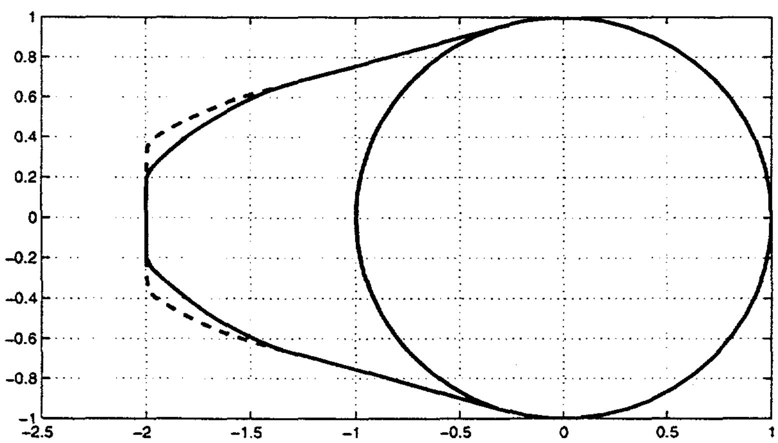

Figure 3: Influence of the Curvature Constraints on the Optimal Shape