Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

Efficient second order optimization technique for structural optimization

M. El-Sayed, S. Anderson and K. Zumwalt
Department of Mechanical Engineering
GMI Engineering and Management Institute, Flint, MI 48504

Abstract

Constrained optimization methods classically fall into the categories of zero- or first-order algorithms. In this paper, a second order constrained optimization method, sequential quadratic programming with quadratic constraints (SQPQC), is developed. This method uses first- and second-order derivative information to build a second-order Taylor's series expansion to the constrained optimization problem. By using duality, the resulting subproblem, to find the search direction, is transformed into an unconstrained optimization problem with respect to the dual variables. Comparisons with robust feasible directions (RFD), for a curved plate shape optimization test case, is presented.

1. Introduction

Many numerical methods exist to solve all the classes of the optimization problem. For nonlinear constrained optimization, the prevalent methods are PLBA, RFD, GRG, and penalty methods. PLBA [1-3], one of the better methods, uses first-order information to approximate the hessian of the Lagrange function and solves a quadratic subproblem for the search direction. While performing better than its predecessors, PLBA still requires too many function evaluations to solve large design problems. Robust Feasible Directions (RFD) [4], finds and then tries to follow constraints to the optimum. This algorithm has the same fault as PLBA.

To overcome this fault, the first- and second-order derivative of the functions are incorporated into a second-order Taylor's series expansion of the constrained optimization problem. Using this information, this new method creates an unconstrained subproblem which yields the change for the design variables. After updating the design, new function and derivative information form another subproblem are obtained. This subproblem is solved repeatedly until convergence.

Only one other second-order method is presented in literature [5]. This method uses the Augmented Lagrangian penalty method to solve the resulting subproblem. To solve this second-order subproblem, Newton's method is used. The method performed well, but had problems dealing with points close to the optimum. Even with this drawback, the number of function evaluation were reduced.

This paper presents an efficient method for sequential quadratic programming with quadratic constraints (SQPQC). The derivation of the method is presented followed by the algorithm. Specific details on the implementation of important steps of the algorithm are presented. Finally, a test cases to evaluate SQPQC against RFD is discussed.

2. Derivation

Sequential Quadratic Programming with Quadratic Constraints is based on solving a second-order Taylor's series approximation to the objective function and second-order approximations to the constraints. A second-order Taylor's series expansion is written as:

$$f(x_i) = f(x_i^0) + \frac{\partial f(x_i^0)}{\partial x_j} S_j + \frac{1}{2} \frac{\partial^2 f(x_i^0)}{\partial x_j \partial x_k} S_j S_k \qquad (1)$$

where $S_i = (x_i - x_i^0)$, x_i are the independent variables, and i, j, k = 1 to the number of independent variables. From equation (1), the second-order approximation to the constrained optimization problem is stated as:

minimize:

$$f^0 + \frac{\partial f^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 f^0}{\partial x_i \partial x_j} S_i S_j$$

subject to:

$$g_m^0 + \frac{\partial g_m^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 g_m^0}{\partial x_i \partial x_j} S_i S_j \le 0 \qquad (2)$$

$$h_l^0 + \frac{\partial h_l^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 h_l^0}{\partial x_i \partial x_j} S_i S_j = 0$$

$$S_i^L \le S_i \le S_i^U$$

where m = 1 to the number of inequality constraints M, and l = 1 to the number of equality constraints, L. Equation (2) is transformed into a single equation using the Lagrange function.

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

$$L(S_i, u_m, v_l) = f^0 + \frac{\partial f^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 f^0}{\partial x_i \partial x_j} S_i S_j +$$
$$u_m \left(g_m^0 + \frac{\partial g_m^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 g_m^0}{\partial x_i \partial x_j} S_i S_j \right) +$$
$$v_l \left(h_l^0 + \frac{\partial h_l^0}{\partial x_i} S_i + \frac{1}{2} \frac{\partial^2 h_l^0}{\partial x_i \partial x_j} S_i S_j \right) \tag{3}$$

with

$$u_m \geq 0.$$

Applying the theorems of Kuhn-Tucker Optimality Criteria and Saddle Point Conditions for the Lagrange Function, equation (1) is transformed into an unconstrained optimization problem with bound of nonnegativity using the Lagrange multipliers as the independent variables and the change in primal space as dependent variables. The Kuhn-Tucker conditions are stated as

$$\frac{\partial L}{\partial x_i} = 0, \frac{\partial L}{\partial u_m} \leq 0, \frac{\partial L}{\partial v_l} = 0$$
$$u_m g_m \leq 0 \quad \text{not summed on m.} \tag{4}$$

and

$$u_m \geq 0$$

The Saddle Point Conditions for the Lagrange Function is that the minimum in primal space is a maximum is dual space. Primal space has x as the independent variables while dual space uses u and v as the independent variables. In this section, S is used to distinguish that it is dependent on u and v. Using equation (4) in equation (3), yields the set of equations to solve for S given below.

$$\left(\frac{\partial^2 f^0}{\partial x_i \partial x_j} + u_m \frac{\partial^2 g_m^0}{\partial x_i \partial x_j} + v_l \frac{\partial^2 h_l^0}{\partial x_i \partial x_j} \right) S_j = -\left(\frac{\partial f^0}{\partial x_i} + u_m \frac{\partial g_m^0}{\partial x_i} + v_l \frac{\partial h_l^0}{\partial x_i} \right) \tag{5}$$

Now, the dual optimization problem can be stated as:

Minimize: $\quad H(u_m, v_l) = -L(S_i(u_m, v_l), u_m, v_l) \qquad (6)$

with

$$u_m \geq 0$$

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

This dual problem may be solved by two approaches. The first is an unconstrained optimization method with a modification on the search direction to impose nonnegativity on the Lagrange multipliers corresponding to the inequality constraints. The second is to apply optimality criteria with slack variables in the Lagrange function. This paper uses the first approach.
Once the quadratic approximation to the problem is solved, the primal variables are updated by

$$x_i = x_i^0 + S_i. \tag{7}$$

By iterations of equations (6) and (7), the algorithm for Sequential Quadratic Programming with Quadratic Constraints results. Given x_i^0, max_itt, ε_1, and ε_2. proceeded with the following sequence of steps.

Algorithm : Sequential Quadratic Programming with Quadratic Constraints (SQPQC)

1. Set p=0.
2. Determine the objective function and constriant values.
3. Determine the potential constraints.
4. Get the the objective function and potential constraint gradients, and if the design point is close to a constraint or there is a constraint violation, calculate the hessians .
5. Agian, depending on the constraint values, either solve the quadratic approximation to the optimization problem, equation (6), resulting with S, u, and v, or perform an unconstrained search with an inexact line search to move closer to the constraint boundaries.
6. If the descent condition then stop.
7. Update the primal variables using equation (7).
8. Get the objective function and constriant values.
9. if Abs $\left(L\left(x_i^p, u_m^p, v_l^p\right) - L\left(x_i^{p-1}, u_m^{p-1}, v_l^{p-1}\right)\right) \le \varepsilon_1$ then stop.
10. Determine the potential constraints.
11. Get the objective function and potential constraint gradients.
12. if $\left(\left\|\dfrac{\partial L(x_i, u_m, v_l)}{\partial x_i}\right\| \le \varepsilon_2\right)$ then stop.
13. Get the objective function and potential constraint hessians.
14. p = p + 1.
15. if (p>max_itt) then stop with an iteration error, else goto 5.

There are many details to implement the algorithm above. Since the real details of the algorithm above are in step 5, the following sections give the algorithms for its implementation.

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

3. Unconstrained Optimization Subproblem

The first-order unconstrained optimization method of conjugate directions is used to solve the approximation to the optimization problem. This method requires the first derivative of the dual objective function, H, which is given by the chain rule as:

$$\frac{\partial H(u_s, v_t)}{\partial u_m} = -\frac{\partial L(x_r, u_s, v_t)}{\partial S_i}\frac{dS_i}{du_m} - \frac{\partial L(x_r, u_s, v_t)}{\partial u_m} = -\frac{\partial L(x_r, u_s, v_t)}{\partial u_m}$$
$$= -\left(g_m^0 + \frac{\partial g_m^0}{\partial x_i}S_i + \frac{1}{2}\frac{\partial^2 g_m^0}{\partial x_i \partial x_j}S_i S_j\right) \qquad (8)$$

and

$$\frac{\partial H(u_s, v_t)}{\partial v_l} = -\frac{\partial L(x_r, u_s, v_t)}{\partial S_i}\frac{dS_i}{dv_l} - \frac{\partial L(x_r, u_s, v_t)}{\partial v_l} = -\frac{\partial L(x_r, u_s, v_t)}{\partial v_l}$$
$$= -\left(h_l^0 + \frac{\partial h_l^0}{\partial x_i}S_i + \frac{1}{2}\frac{\partial^2 h_l^0}{\partial x_i \partial x_j}S_i S_j\right) \qquad (9)$$

These two equations are calculated as part of solving for the dual objective function. While there is no overhead in determining the first derivative of the dual function, the second derivative is numerically expensive.

The dual objective function is highly nonlinear and presents a burden on the unconstrained optimization process. The function may spike to negative infinite when no bounds on S are imposed. This fact is especially evident when the objective function's hessian is indefinite or close to singular. To overcome this difficulty, a sufficiently large value is added to the diagonals of the objective function's hessian when the hessian values are small compared to the values found in the constraints or the hessian is close to singular. The value should be larger than the largest magnitude found in the hessians of the constraints and positive. The effect of this value is that the dual objective function is made convex and has a value close to the values of the constraints; thus, the constraint values cannot dominate the objective function value and produce a spike to negative infinite.

Some modifications to conjugate directions were made to account for the nonnegativity imposed on the dual variables corresponding to the inequality constraints. When the gradient value corresponding to a inequality constraint is positive and the Lagrange multiplier for that constraint is zero, the gradient value is zeroed. Also, the update to the search direction is restarted since the conjugate direction is invalid.

The line search methods of golden section and cutback are both implemented. The golden section algorithm assumes a unimodal piece of the objective function is bounded. Due to the highly nonlinear characteristics of the dual objective function, this assumption is violated. The problem is to find a point close to the start where there exists a decrease in the dual objective function. To accomplish this task, a loop to step back toward the lower point is performed if the function value at point

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

one is greater than the starting value and the value at point two. Also, the function value at point two must be greater than the value at the starting point. The function value is also augmented by the addition of a penalty factor for constraint violation which ensure the method proceeds to an answer.

4. Handling of Side Constraints

The formulation of SQPQC did not included side constraints because there are two ways to deal with them. The first is to directly implement them as inequality constraints. This method was tried first and was found to impose a step function on the Lagrange multipliers corresponding to the side constraint. The unconstrained algorithm would the diverge. The second approach was to enforce the side constraints after the unconstrained optimization problem returned with a search direction. To enforce a side constraint, the search direction value corresponding to a design variable which steps beyond its bound is truncated to move only to the bound.

5. Test Case

Plate with Curved Edges

In figure (1), a plate is subjected to a distributed load of 625 lb/in and is constrained along the bottom edge. The two curved edges are β-splines with A and B being the center control point for each respectively. The design element is meshed into a three by three element surface.
These points are allowed to move horizontally along the **V** direction. Along with the control points, β_1 and β_2 for each spline are added to the set of design variables for a total of six design variables defining the model. The initial values are the location of the control given by the drawing of the model, β_1 equal to 1, and β_2 equal to 0 for each spline. The objective function is to minimize the area (the density is set to 1) with the constraints being the element stress may not exceed 10,000 psi. The starting area is 43.2 in^2 with no constraint violation.

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

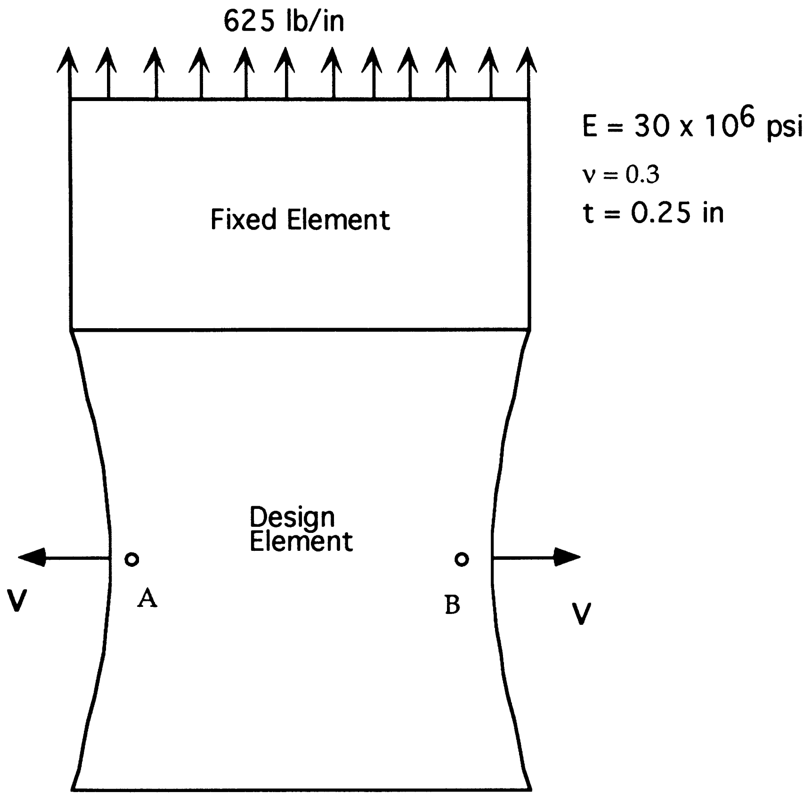

Figure 1 : Curved Plate Initial Shape

6. Results

Test case results are presented in tables (1) and (2) for SQPQC and RFD respectively. The number of constrained search iterations is listed first. This number is the number of cycles performed by the optimization method in order to obtain a solution. Next, the number of linear solutions is given. The entry is the number of times the column solver [6], is called and represents the amount of work done for constraint, gradient, and hessian evaluations. The number of model updates is a counter for the times the structural model is changed. Both the number of linear solutions and the number of model updates are used to determine the amount of work done in the analysis. The constraint violation is a measure of how accurate the solution is. Time is the total time to run the optimization process on DEC VAX 4500.

From table (1) and table (2), it is clear that SQPQC outperformed RFD in this test case. The reason the number of linear solutions are equal even though the number of iterations are different is because SQPQC requires hessian evaluation which increases this number. The significant advantage of SQPQC is the fewer number of model updates. RFD has a problem overcoming a slight constraint violation resulting in a lower area. The optimum shape in shown in figure (2), for both SQPQC and RFD.

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

Table 1: SQPQC Results for the Curved Plate

Iterations	2
Number of linear solutions	11
Number of model updates	3
Final area	42.12
Constraint violation	0.00
Time	1.66 sec.

Table 2: RFD Results for the Curved Plate

Iterations	7
Number of linear solutions	11
Number of model updates	9
Final area	42.11
Constraint violation	7.22e-1
Time	2.74 sec.

7. Conclusion

An efficient second-order optimization method was developed. This method was found to be superior to robust feasible directions, for the test case considered. While sequential quadratic programming with quadratic constraints requires more work to evaluate the hessians of the objective function and constraints, the method performs fewer iterations and thus, less function, gradient, and hessian evaluation are needed.

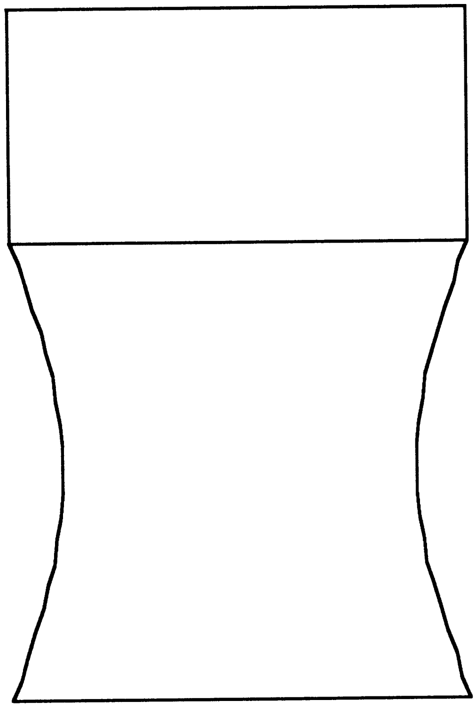

Figure 2 : Curved Plate Optimum Shape

8. References

1. Arora, J. S., *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.

2. Arora, J. S. and Tseng C. H. , "On Implementation of Computational Algorithms for Optimal Design 1: Preliminary Investigation", *International Journal for Numerical Methods in Engineering*, 26, 1988, 1365-1382.

3. Arora, J. S. and Tseng C. H., "On Implementation of Computational Algorithms for Optimal Design 2: Extensive Numerical Investigation", *International Journal for Numerical Methods in Engineering*, 26, 1988, 1383-1402.

4. Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, Inc., New York, 1984.

5. Cao, Q. and Krishnaswami, P., "In Support of Second-Order Optimization", Advances in Design Automation, ASME, 1, 1991, 201-207.

6. Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.