A STRUCTURED REDUCED SEQUENTIAL QUADRATIC PROGRAMMING AND ITS APPLICATION TO A SHAPE DESIGN PROBLEM

by

Kyehong Kang

Dissertation submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

APPROVED:

Terry L. Herdman

Terry L. Herdman, Chairman

John A. Burns

John A. Burns

Eugene M Cliff

Eugene M Cliff

Max D. Gunzburger

Max D. Gunzburger

Tao Lin

Tao Lin

September, 1994

Blacksburg, Virginia

A STRUCTURED REDUCED SEQUENTIAL QUADRATIC PROGRAMMING AND ITS APPLICATION TO A SHAPE DESIGN PROBLEM

by

Kyehong Kang

Committee Chairman: Terry L. Herdman

Mathematics

(ABSTRACT)

The objective of this work is to solve a model one dimensional duct design problem using a particular optimization method. The design problem is formulated as an equality constrained optimization, called *All at once* method, so that the analysis problem is not solved until the optimal design is reached. Furthermore, the block structure in the Jacobian of the linearized constraints is exploited by decomposing the variables into the design and flow parts. To achieve this, Sequential quadratic programming with BFGS update for the reduced Hessian of the Lagrangian function is used with *Variable reduction method* which preserves the structure of the Jacobian in representing the null space basis matrix. By updating the reduced Hessians only of which the dimension is the number of design variables, the storage requirement for Hessians is reduced by a large amount. In addition, the flow part of the Jacobian can be computed analytically.

The algorithm with a line search globalization is described. A global and local analysis is provided with a modification of the paper by Byrd and Nocedal [Mathematical Programming 49(1991) pp 285-323] in which they analyzed the similar algorithm with the

Orthogonal factorization method which assumes the orthogonality of the null space basis matrix. Numerical results are obtained and compared favorably with results from the *Black box method*- unconstrained optimization formulation.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Dr. Herdman, for his encouragement and financial support and I also thank all the committee members including Dr. Burns, Dr. Cliff, Dr. Gunzburger, and Dr. Lin for their help and support.

This research was supported in part by the Air Force Office of Scientific Research under Grant AFOSR F49620-93-1-0280. The author was also supported by the mathematics department at Virginia Tech in the form of a teaching assistant position during the research. I would like to thank Dr. Burns for reading a draft of the thesis and making valuable comments and suggestions.

TABLE OF CONTENTS

1 INTRODUCTION **1**

1.1 A Description of the Problem 4

1.1.1 Analysis Problem 4

1.1.2 Numerical Schemes for the Analysis Problem 6

1.2 The Design Problem 8

2 A STRUCTURED REDUCED SEQUENTIAL QUADRATIC PROGRAMMING **9**

2.1 Local Algorithm 23

2.2 Global Algorithm 27

3 A GLOBAL AND LOCAL CONVERGENCE OF THE ALGORITHM **32**

3.1 Global Behavior of the Algorithm 32

3.2 Local Behavior of the Algorithm 44

4 NUMERICAL IMPLEMENTATION AND RESULTS **57**

4.1 Numerical Implementation 57

4.2 Numerical Results 70

4.3 Conclusion 81

LIST OF FIGURES

4.1.1 Exact solution versus Numerical solution using Artificial Viscosity scheme. 59
4.1.2 Exact solution versus Numerical solution using Enquist-Osher scheme. . . 59
4.1.3 Exact solution versus Numerical solution using Godunov scheme. 60
4.2.1 3D objective function plot using exact solution. 74
4.2.2 3D objective function plot using Godunov scheme. 74
4.2.3 3D objective function plot using Artificial Viscosity scheme. 75
4.2.4 Contour plot of objective function using exact solution. 75
4.2.5 Contour plot of objective function using Godunov scheme. 76
4.2.6 Contour plot of objective function using Artificial Viscosity scheme. . . . 76
4.2.7 Cross Section of Objective function. 77
4.2.8 Duct area matching-Artificial Viscosity used. 77
4.2.9 Velocity matching-Artificial Viscosity used. 78
4.2.10 Duct area matching-Artificial Viscosity used. 78
4.2.11 Velocity matching-Artificial Viscosity used. 79
4.2.12 Duct area matching-Enquist-Osher used. 79
4.2.13 Velocity matching-Enquist-Osher used. 80
4.2.14 Duct area matching-Godunov used. 80

4.2.15 Velocity matching-Godunov used. 81

LIST OF TABLES

4.1 The CPU time comparison of All At Once versus Black Box 73

Chapter 1

INTRODUCTION

Our model design problem is a one dimensional duct problem which was recently investigated by Frank and Shubin in [1]. In [1] Frank and Shubin presented several numerical approaches to this problem and compared the resulting designs. We are particularly interested in the "all at once" method because the numerical results based on this method appears to be cost effective when compared with other methods. This is the primary motivation for further investigation on this method.

The design problem under consideration is to find the shape of the duct which minimizes the distance from the given goal velocity. The design parameters are the coefficients of the B-splines that generates the duct shape.

The Black box approach involves the following steps:

1. Input design variables
2. B-spline generates the duct shape
3. Calculate flow velocities by solving the flow equation
4. Measure the distance from the goal velocities.

Although this approach seems to be a natural optimization formulation, it has a drawback in step 3 if solving the flow equation is time consuming, because we need to solve the flow equation whenever we evaluate the objective function.

In the all at once approach we use a discretized flow equation as a constraint and the objective function only measures the difference between the computed velocity and the goal velocity. Since most constrained optimization methods, including sequential quadratic programming allows infeasible points during the iteration we do not solve the analysis problem but compute the residual of the discretized equation.

There are two particular issues associated with this design problem; one is how to handle non-smoothness in the optimization incurred by the shock of the flow equation under consideration, the other is to develop a time-efficient optimization method. Our approach to address these issues is to design a particular problem-oriented optimization algorithm. That is, for the first issue, we employ a simple backtracking line search algorithm in [7] with non-differentiable l_1 merit function. For the second issue, the all at once approach with variable reduction method reduces the amount of work significantly.

As an optimization method, sequential quadratic programming with BFGS update for the reduced Hessian of the Lagrangian function is applied to the design problem. Natural decomposition of the design and the flow variables, along with the variable reduction method in representing the null space bases, enables us to exploit the sparsity pattern of the Jacobian matrix of the discretized equations. In fact, the flow variable part of the Jacobian turns out to be tridiagonal and it can be computed analytically. We observe that this reduces the amount of work dramatically. Also the dimension of the Hessian is significantly reduced by updating the design variable part of the Hessian only.

The Sequential Quadratic Programming with reduced Hessian update algorithm without globalization along with the variable reduction method appears in [11]. Local convergence proofs were given in [8] which used an orthogonal factorization method. Also a similar algorithm is provided and analyzed in [22] in infinite dimensional optimization. A great deal of work was done for the SQP with reduced Hessian update along with orthogonal factorization method, for example, [2],[4], [5], and [8]. Also, the super-linear convergence characterization theorem in [21] by Dennis and Moré is extended in [2] by Powell using the orthogonal factorization method. However, there are few results for the the variable reduction method. This method requires a certain structure in the constraint gradient. We need information that allows us to determine which submatrix of the constraint gradient at each iteration is nonsingular. Hence, it may not be considered as a general nonlinear programming method. Therefore, we give the convergence proofs of the algorithm based on [5].

The design problem which follows the introduction of the ananlysis problem is described in section 1.2. In chapter 2 we present the algorithm based on reduced sequential quadratic programming with BFGS update and an analysis is given in chapter 3. We implement this method to the design problem and numerical results will be presented in chapter 4.

1.1 A Description of the Problem

1.1.1 Analysis Problem

The analysis problem is the one that Frank and Shubin considered in [1]. The steady flow of an invicid fluid in a duct of variable cross-sectional area $A(\xi)$ is governed by the Euler equations.

$$F_\xi + G = 0, \quad 0 \leq \xi \leq 1, \tag{1.1.1}$$

where

$$F = \begin{pmatrix} \rho u A \\ (\rho u^2 + P)A \\ (\rho E + P)uA \end{pmatrix}, \quad G = \begin{pmatrix} 0 \\ -PA_\xi \\ 0 \end{pmatrix},$$

and ξ is the distance along the duct, ρ is the density, u is velocity, $E = e + u^2/2$, and e is specific internal energy, and P is pressure. $A(\xi)$ is assumed to be smooth and A_ξ denotes the derivative with respect to ξ. The pressure P is given by the equation of state for a perfect gas, $P = (\gamma - 1)\rho e$, $\gamma = C_p/C_v$ is the gas constant, where C_v, C_p denote specific heats at constant volume and at constant pressure respectively. We used $\gamma = 1.4$ for air. The derivation of the Euler equations can be found in chapter 5 in [23] by Anderson.

We can reduce (1.1.1) to a single ordinary differential equation as explained in [13]. In brief, we combine the continuity equation, $\rho u A = C$, C is a constant, and conservation of energy in the form of $h + \frac{u^2}{2} = H$, where $h = \gamma e$ and H is total enthalpy constant and substitute the result in the momentum equation to remove ρ, e, then we obtain

$$f_\xi + g = 0, \tag{1.1.2}$$

where

$$f(u) \equiv u + \bar{H}/u, \quad g(u,\xi) \equiv \frac{A_\xi}{A}(\bar{\gamma}u - \bar{H}/u),$$

$\bar{\gamma} = (\gamma-1)/(\gamma+1)$, and $\bar{H} = 2H\bar{\gamma}$ are given constants, $\gamma = 1.4$, $\bar{H} = 1.2$.

Again, as noted in [1], applying standard results for hyperbolic conservation laws, we know that the flow is sonic (where f' vanishes) at $u = u_* \equiv \sqrt{\bar{H}}$, supersonic for $u > u_*$, and subsonic for $u < u_*$.

Solutions with shocks (discontinuity at ξ_s) satisfy the Rankine-Hugoniot jump relation (see [24]), $f(u_l) - f(u_r) = 0$, or, equivalently,

$$u_l u_r = \bar{H}, \tag{1.1.3}$$

where we assume that u is a smooth solution before and after the shock, and u_l and u_r are the limits of $u(\xi)$ as ξ approaches ξ_s from the left and from the right respectively. The entropy condition states that such jumps are always from supersonic to subsonic which implies

$$u_l > u_* > u_r, \tag{1.1.4}$$

we note the equation (3.60) in [23].

We specify two boundary conditions, $u = u_{in}, u = u_{out}$, at inlet and at outlet respectively and we assume that the duct is always diverging, i.e., $A_\xi > 0$. These conditions are required to ensure the well-posedness of the analysis problem.

We pose the analysis or forward problem, specified so that the solution contains a single shock at ξ_s, is supersonic for $0 < \xi < \xi_s$, and subsonic for $\xi_s < \xi < 1$. Hence, the analysis problem is, given $A(\xi)$ with $A_\xi > 0$, to find a solution u to the following boundary value problem:

$$\begin{pmatrix} f_\xi + g = 0, \text{ away from shocks;} \\ u_l u_r = \bar{H} \text{ and } u_l > u_* > u_r, \text{ at shocks;} \\ u(\xi = 0) = u_{in} > u_* \\ u(\xi = 1) = u_{out} < u_* \end{pmatrix}. \tag{1.1.5}$$

The analysis problem given above is well-posed and existence and uniqueness and continuous dependence of the solution on the data is known. A detailed description of the analytic solution is given in [1].

1.1.2 Numerical Schemes for the Analysis Problem

We discretize [0 1] into cell centered equally spaced uniform mesh and let $\xi_j = (j - 1/2)h, h = 1/J$, where J is the number of unknown grid values. Let u_j denote an approximation to the value of the exact solution at the jth grid cell. A finite difference scheme for (1.1.2) to solve (1.1.5) numerically gives

$$c_j \equiv \frac{f_{j+1/2} - f_{j-1/2}}{h} + g_j = 0. \tag{1.1.6}$$

We employ three different numerical schemes by choosing different numerical flux functions, $f_{J+1/2}$ and $f_{j-1/2}$ in terms of u_j, u_{j+1} and u_{j-1}, u_j respectively; Godunov scheme, Enquist-Osher scheme, and Artificial Viscosity scheme which make $f_{j+1/2}$ a $C^0, C^1,$ *and* C^∞ function of its arguments respectively. In particular, consider the following:

Godunov scheme:

$$f_{j+1/2} = \left(\begin{array}{ll} f_{j+1} & if\ u_j, u_{j+1} < u_*; \\ f_j & if\ u_j, u_{j+1} \geq u_*; \\ f_* & if\ u_j \leq u_* \leq u_{j+1}; \\ max(f_j, f_{j+1}) & if\ u_{j+1} \leq u_* < u_j \end{array} \right). \tag{1.1.7}$$

Here $f_j = f(u_j)$, etc., and u_* is the speed of sound.

Enquist-Osher scheme:

$$f_{j+1/2} = \left(\begin{array}{ll} f_{j+1} & if\ u_j, u_{j+1} < u_*; \\ f_j & if\ u_j, u_{j+1} \geq u_*; \\ f_* & if\ u_j, u_* \leq u_{j+1}; \\ f_j + f_{j+1} - f_* & if\ u_{j+1} \leq u_* < u_j \end{array} \right). \tag{1.1.8}$$

Artificial Viscosity scheme:

$$f_{j+1/2} = 1/2(f_{j+1} + f_j - \alpha(u_{j+1} - u_j)) \tag{1.1.9}$$

where we took $\alpha = 1$.

Later we will show how the schemes smear out the shock by comparing the exact solution with numerical solutions using different numerical schemes.

1.2 The Design Problem

The design problem is formulated as follows:

given $\hat{u}_j$ for $j = 1, ..., J$, seek u_j which

$$\begin{aligned} \text{minimizes} \quad & \tfrac{1}{2}\sum_{j=1}^{J} |u_j - \hat{u}_j|^2 \\ \text{subject to} \quad & c_j \equiv \tfrac{f_{j+1/2} - f_{j-1/2}}{h} + g_j = 0 \end{aligned} \tag{1.2.1}$$

where, $f(u) \equiv u + \bar{H}/u, \quad g(u, \xi) \equiv \frac{A_\xi}{A}(\bar{\gamma} u - \bar{H}/u)$.

As we see, the discretized ordinary differential equation is an equality constraint in our optimization formulation.

Chapter 2

A STRUCTURED REDUCED SEQUENTIAL QUADRATIC PROGRAMMING

A Structured reduced Sequential Quadratic Programming indicates that we update the reduced Hessian only using the BFGS formula and exploit the "structure" of the Jacobian of the linearized constraint function that comes from the design problem.

We consider the general nonlinear programming problem.

$$\begin{aligned} &\text{Minimize } F(x) \\ &\text{subject to } c(x) = 0,\ x \in R^n,\ c(x) \in R^t \end{aligned} \tag{2.0.2}$$

where we assume that $F(x)$ and $c(x)$ are twice continuously differentiable in an open convex region that contains a strong local minimizer x_*.

The first order Karush-Kuhn-Tucker (or Kuhn-Tucker) necessary condition in [3] states that if x_* is a solution to (2.0.2) and the Jacobian at x_* has full rank, then there exists a vector (called Lagrange multiplier) λ_* such that

$$\begin{aligned} \nabla_x F(x_*) &= \textstyle\sum_{i=1}^{t} \lambda_* \nabla_x c_i(x_*), \\ c(x_*) &= 0. \end{aligned} \tag{2.0.3}$$

If we define a function, called a Lagrangian function,

$$\mathcal{L}(x,\lambda) = F(x) - \sum_{i=1}^{t} \lambda^i c_i(x) \tag{2.0.4}$$

then the necessary condition can be given as

$$\begin{aligned}
\nabla\mathcal{L}(x_*,\lambda_*) &= \begin{pmatrix} \nabla_x \mathcal{L}(x_*,\lambda_*) \\ \\ \nabla_\lambda \mathcal{L}(x_*,\lambda_*) \end{pmatrix} \\
&= \begin{pmatrix} \nabla_x F(x_*) - \sum_{i=1}^{t} \lambda_*^i \nabla_x c_i(x_*) \\ \\ -c(x_*) \end{pmatrix} \\
&= 0
\end{aligned} \tag{2.0.5}$$

for some λ_*. Note that ∇, ∇_x denotes the gradients of the function involved with respect to (x,λ) and x respectively.

Hence, the pair (x_*,λ_*) is a stationary point of the Lagrangian function $\mathcal{L}(x,\lambda)$ in R^{n+t}. Therefore an algorithm can be sought in finding a stationary point of $\mathcal{L}(x,\lambda)$, i.e., solve

$$\nabla\mathcal{L}(x,\lambda) = 0 \tag{2.0.6}$$

where $\mathcal{L}(x,\lambda)$ is given in (2.0.4).

The obvious choice is to apply Newton's method to the function $\nabla\mathcal{L}$, which leads us to solve the following equation for p_k, a newton step, given x_k and λ_k at the kth iteration.

$$\nabla\nabla\mathcal{L}(x_k,\lambda_k)p_k = -\nabla\mathcal{L}(x_k,\lambda_k), \quad p_k \in R^{n+t} \tag{2.0.7}$$

and then x_{k+1} will be the first t components of $x_k + p_k$.

Since

$$\nabla\mathcal{L}(x,\lambda) = \begin{pmatrix} \nabla_x F(x) \\ \\ -c(x) \end{pmatrix} - \sum_{i=1}^{t} \lambda^i \begin{pmatrix} \nabla_x c_i(x) \\ \\ 0 \end{pmatrix}, \tag{2.0.8}$$

we have

$$\nabla\nabla\mathcal{L}(x,\lambda) = \nabla \begin{pmatrix} \nabla_x F(x) \\ \\ -c(x) \end{pmatrix} - \sum_{i=1}^{t} \nabla \left(\lambda^i \begin{pmatrix} \nabla_x c_i(x) \\ \\ 0 \end{pmatrix} \right),$$

$$\nabla \begin{pmatrix} \nabla_x F(x) \\ \\ c(x) \end{pmatrix} = \begin{pmatrix} \nabla \frac{\partial F(x,\lambda)}{\partial x_1}^T \\ \nabla \frac{\partial F(x,\lambda)}{\partial x_2}^T \\ \vdots \\ \nabla \frac{\partial F(x,\lambda)}{\partial x_n}^T \\ \nabla c_1^T \\ \vdots \\ \nabla c_t^T \end{pmatrix} \tag{2.0.9}$$

$$= \begin{pmatrix} \nabla_{xx}^2 F(x) & 0 \\ \\ -\nabla_x c(x) & 0 \end{pmatrix}.$$

And it is easy to see that

$$\sum_{i=1}^{t} \nabla \begin{pmatrix} \nabla_x c_i(x) \\ 0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{t} \lambda_i \nabla_x^2 c_i & \nabla_x c^T \\ 0 & 0 \end{pmatrix}. \tag{2.0.10}$$

By putting (2.0.9) and (2.0.10) together we obtain the expression for the Jacobian of $\nabla\mathcal{L}(x,\lambda)$, we have

$$\begin{aligned} \nabla\nabla\mathcal{L}(x,\lambda) &= \nabla \begin{pmatrix} \nabla_x F(x) \\ -c(x) \end{pmatrix} - \textstyle\sum_{i=1}^{t} \nabla \begin{pmatrix} \nabla_x c_i(x) \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \nabla_{xx}^2 F(x) - \sum_{i=1}^{t} \lambda^i \nabla_{xx}^2 c_i(x) & -\nabla_x c(x)^T \\ -\nabla_x c(x) & 0 \end{pmatrix}. \end{aligned} \tag{2.0.11}$$

We note that

$$\nabla_{xx}^2 \mathcal{L}(x,\lambda) = \nabla_{xx}^2 F(x) - \sum_{i=1}^{t} \lambda^i \nabla_{xx}^2 c_i(x),$$

and $\nabla_x c(x)$ is the transpose of the Jacobian of $c(x)$.

We define A_k and g_k by

$$\begin{aligned} A_k &\equiv \nabla_x c(x_k)^T = (\nabla_x c_1(x_k)\ \nabla_x c_2(x_k)\ \ldots\ \nabla c_t(x_k)) \\ g_k &\equiv \nabla_x F(x_k), \end{aligned}$$

respectively, and let $F(x_k) \equiv F_k$, $c(x_k) \equiv c_k$. If we let $p_k = (\delta x, \delta\lambda)^T$ and replace $\nabla\nabla\mathcal{L}(x_k,\lambda_k)$ in (2.0.7) with expressions that we obtained, then the equation for newton step becomes

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L}(x_k,\lambda_k) & -A_k \\ -A_k^T & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta\lambda \end{pmatrix} = \begin{pmatrix} -g_k + \sum_{i=1}^t \lambda_k^i \nabla_x c_i(x_k) \\ c_k \end{pmatrix}.$$

Here λ_k^i is the ith component of λ_k.

If we let $\lambda \equiv \lambda_k + \delta\lambda$ and $\delta \equiv \delta x$, we can further simplify to obtain,

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L}(x_k,\lambda_k) & -A_k \\ -A_k^T & 0 \end{pmatrix} \begin{pmatrix} \delta \\ \lambda \end{pmatrix} = \begin{pmatrix} -g_k \\ c_k \end{pmatrix}. \tag{2.0.12}$$

By solving (2.0.12) we get the next iterates

$$x_{k+1} = x_k + \delta$$
$$\lambda_{k+1} = \lambda.$$

Equation (2.0.12) is equivalent to the system

$$\begin{aligned} \nabla^2_{xx}\mathcal{L}(x_k,\lambda_k)\delta &= -(g_k - A_k\lambda) \\ A_k^T\delta &= -c_k, \end{aligned} \tag{2.0.13}$$

and (2.0.13) is equivalent to solving the following quadratic programming problem in the sense that the first order necessary condition of the Lagrangian function of the following quadratic programming problem leads to (2.0.13),

$$\begin{aligned} &\min\ g_k^T\delta + \tfrac{1}{2}\delta^T\nabla^2_{xx}\mathcal{L}(x_k,\lambda_k)\delta \\ &\text{subject to } A_k^T\delta = -c_k. \end{aligned} \tag{2.0.14}$$

More precisely, the solution δ and the associated Lagrange multiplier λ from (2.0.14) solve (2.0.13). First we consider the constraint $A_k^T\delta = -c_k$. We decompose R^n into the null space and the range space, i.e., $R^n = \mathcal{N}(A_k^T) \oplus \mathcal{R}(A_k^T)$ where $\mathcal{N}(A_k^T)$ denotes the null space of A_k^T and $\mathcal{R}(A_k^T)$ denotes the range space. Let Z_k and Y_k be matrices whose columns form a basis of each space respectively and we pick Y_k such that $A_k^TY_k = I_t$.

In view of this decomposition,

$$\delta = Z_kw_k + Y_kr_k,$$

the constraint becomes

$$\begin{aligned} A_k^T\delta &= A_k^T(Z_kw_k + Y_kr_k) \\ &= A_kY_kr_k \\ &= r_k \\ &= -c_k. \end{aligned}$$

Hence the constraint completely determines the range component of δ and we should have

$$r_k = -c_k.$$

If we denote $\nabla^2_{xx}\mathcal{L}(x_k,\lambda_k)$ by W_k and substitute $\delta = Z_kw_k - Y_kc_k$ back into (2.0.14), we obtain an unconstrained minimization with respect to w_k as follows:

$$Min \ (Z_k^T g_k - Z_k^T W_k Y_k c_k)^T w_k + \frac{1}{2} w_k^T Z_k^T W_k Z_k w_k. \tag{2.0.15}$$

The solvability of this unconstrained problem is guaranteed by the fact that $Z_k^T W_k Z_k$ is approximated by a positive definite matrix using BFGS updating formula. Hence, formally we can write the following equation for w_k,

$$(Z_k^T W_k Z_k) w_k = -(Z_k^T g_k - Z_k^T W_k Y_k c_k). \tag{2.0.16}$$

Then the δ of the Newton step p_k in (2.0.7) is given by

$$\delta = -Z_k (Z_k^T W_k Z_k)^{-1} (Z_k^T g_k - Z_k^T W_k Y_k c_k) - Y_k c_k. \tag{2.0.17}$$

If we ignore the term $Z_k^T W_k Y_k c_k$ in (2.0.17), then the formula for δ simplifies to

$$\delta = -Z_k (Z_k^T W_k Z_k)^{-1} (Z_k^T g_k) - Y_k c_k. \tag{2.0.18}$$

Thus, we need to update the projected Hessian -$Z_k^T W_k Z_k$ only. Therefore, we may take advantage of the small dimension of the projected Hessian which is $n - t$. In fact, this is an important aspect of the structure of this problem. Also, while the full Hessian might not always be positive definite at an optimal point, the projected Hessian is always positive definite by the second order sufficiency condition. One major disadvantage is that we lose superlinear convergence. We may have a 2-step superlinear convergence rate with some assumptions. Fletcher explains briefly in [3] why we expect 2-step superlinear convergence. If we update the projected Hessian while ignoring the term $Z_k^T W_k Y_k c_k$ and also, if the

Hessian approximating positive definite matrix B_k is available at kth iteration, then the formula for w_k becomes

$$B_k w_k = -Z_k g_k \tag{2.0.19}$$

and so the formula for δ changes to

$$\delta = -Z_k B_k^{-1} Z_k^T g_k - Y_k c_k. \tag{2.0.20}$$

Even though we do not need λ for iterations, an estimate for λ is necessary for globalization of the algorithm and the convergence analysis. An appropriate formula will be presented after we consider the null space basis and range space basis matrices, i.e., Z_k and Y_k.

Fletcher explains a general scheme for computing suitable Y_k and Z_k in [3] as follows: choose any $n \times (n-t)$ matrix E_k such that $(A_k \; E_k)$ is nonsingular and choose Y_k and Z_k such that

$$(A_k \; E_k)^{-1} = \begin{pmatrix} Y_k^T \\ Z_k^T \end{pmatrix},$$

then it follows that $A_k Z_k = 0$ and $A_k^T Y_k = I_t$ which are the requirements that we imposed. It is easy to see that the columns of such Z_k and Y_k represent bases for the respective spaces. The two methods emerge if we take different matrices for E_k. If A_k is assumed to have full rank, QR factorization of A_k is available so A_k can be written as

$$A_k = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = (Q_1 \; Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix} \tag{2.0.21}$$

where Q is an orthogonal matrix and Q_1 and Q_2 are its block components whose dimensions are $n \times t$ and $n \times (n-t)$ respectively. Here R is an $t \times t$ upper triangular matrix. If we choose

$$E_k = Q_2$$

then

$$(A_k\ E_k)^{-1} = (Q_1 R\ Q_2)^{-1} = \begin{pmatrix} R^{-1}Q_1^T \\ Q_2^T \end{pmatrix}$$

and we obtain

$$Y_k = Q_1 R^{-1^T}$$

$$Z_k = Q_2.$$

These choices of Y_k and Z_k is termed the orthogonal factorization method.

However, our interest lies in the second choice where

$$E_k = \begin{pmatrix} 0 \\ I \end{pmatrix}.$$

In this case, the identity

$$\begin{pmatrix} V_k^T & 0 \\ U_k^T & I \end{pmatrix}^{-1} = \begin{pmatrix} V_k^{-1^T} & 0 \\ -U_k^T V_k^{-1^T} & I \end{pmatrix} = \begin{pmatrix} Y_k^T \\ Z_k^T \end{pmatrix}$$

gives expressions for Y_k and Z_k, that is,

$$Y_k = \begin{pmatrix} V_k^{-1} \\ \\ 0 \end{pmatrix}$$

$$Z_k^T = \begin{pmatrix} -V_k^{-1}U_k \\ \\ I \end{pmatrix} \tag{2.0.22}$$

where,

$$A_k^T = (V_k \; U_k), \tag{2.0.23}$$

and V_k is $t \times t$ and U_k is $t \times (n - t)$. This representation of Z_k and Y_k is called the **Variable Reduction Method**. It is easy to see that the sets of columns of Y_k and Z_k form bases of the range space and the null space of A_k^T respectively. The advantage of the latter formulation of Y_k and Z_k is that it allows us to exploit the structure of the submatrix V_k, in our case, it is a tridiagonal matrix. On the other hand, the disadvantage is that we lose orthogonality of matrix Z_k which is often assumed in the literature to prove convergence results.

Hence our focus is given to develop an algorithm in which we can exploit this structure but convergence properties are sustained. Since the decomposition of variables into the design part and flow part in our case naturally leads to a structured block matrix V_k in A_k, the latter choices fits well into our problem.

We note that Y_k and Z_k exist only if V_k is nonsingular so we should ensure nonsingularity of V_k during the iterations. In computation, the inverse of V_k is not formed, we only need to solve the matrix equations with tridiagonal coefficient matrix V_k.

For the Lagrange multiplier estimate λ_k we use the block matrices in (2.0.23). What we need to require of λ_k is the estimate

$$||\lambda_k - \lambda_*|| \leq \gamma_L ||x_k - x_*|| \tag{2.0.24}$$

for some constant γ_L and for large enough k.

The common choice of the first order multiplier estimate is a solution of the problem,

$$Min_{\lambda \in R^t} ||A_k \lambda - g_k||.$$

With a QR factorization of A_k in (2.0.21) λ can be obtained by solving

$$R\lambda = Q_1 g_k.$$

Hence this formula agrees with orthogonal factorization method in representing Z_k.

The block matrices in (2.0.23) can be employed to estimate λ_k using V_k only. We solve the first t equations only in

$$A_k \lambda = \begin{pmatrix} V_k^T \\ U_k^T \end{pmatrix} \lambda = g_k.$$

If we denote the first t components of g_k by $\tilde{g}_k$, this implies that we solve

$$V_k^T \lambda = \tilde{g}_k. \tag{2.0.25}$$

This formula will be shown to satisfy the **consistency** relationship (2.0.24) near x_*.

The following formulas are those that we need when updating the reduced Hessian B_k. We note the similarity to the BFGS updating formula in the unconstrained case with objective function $f(x)$

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} \tag{2.0.26}$$

where

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$s_k = x_{k+1} - x_k.$$

We give a motivation for the resulting formulas that we will use. The starting point for the derivation of the BFGS formula would be the Quasi-Newton condition which comes from the Taylor expansion of $\nabla f(x_{k+1})$ about x_k after truncation:

$$\nabla f(x_k + s_k) - \nabla f(x_k) = \nabla^2 f(x_k) s_k. \tag{2.0.27}$$

If we let

$$y_k = g_{k+1} - g_k = \nabla f(x_k + s_k) - \nabla f(x_k)$$

$$s_k = x_{k+1} - x_k$$

$$G_k = \nabla^2 f(x_k),$$

then we have the following

$$y_k = G_k s_k.$$

If we impose this condition on the next Hessian approximation B_{k+1} then we obtain the Quasi-Newton condition

$$B_{k+1}s_k = y_k. \tag{2.0.28}$$

These s_k and y_k will be needed in the BFGS updating formula (2.0.26).

In case of a linearly constraint optimization problem

$$Min\ f(x)$$

$$A^T x = b,$$

it follows that if the columns of Z forms a basis for the null space of A^T and x_k is a current feasible point, then the next iterate x_{k+1} is given by

$$x_{k+1} = x_k + Zw_k, \quad \text{for some } w_k.$$

Consequently we have an unconstrained optimization over w_k,

$$Min\ \varphi(w_k) \equiv f(x_k + Zw_k).$$

Applying Taylor's theorem to $\nabla\varphi(w_k)$ about $w_k = 0$, and truncating after the quadratic term produces the identity,

$$\nabla_{w_k}\varphi(w_k) - \nabla_{w_k}\varphi(0) = \nabla^2_{w_k}\varphi(0)w_k, \tag{2.0.29}$$

and by the chain rule we have

$$\begin{aligned} \nabla_{w_k}\varphi(w) &= Z^T\nabla_x f(x_k + Zw_k)Z^T\nabla_x g_{k+1} \\ \nabla_{w_k}\varphi(0) &= Z^T\nabla_x f(x_k)Z^T\nabla_x g_k \\ \nabla^2_{w_k}\varphi(0) &= Z^T\nabla^2 f(x_k)Z. \end{aligned}$$

Therefore, (2.0.29) becomes

$$Z^T g_{k+1} - Z^T g_k = Z^T \nabla^2 f(x_k) Z w_k.$$

Here $Z^T g_k$ and $Z^T \nabla^2 f(x_k) Z$ are called the reduced gradient vector and the reduced Hessian matrix respectively. If we denote these reduced Hessian matrix and reduced gradient vector by M_k and $\tilde{y_k}$, then the equation above reads

$$\tilde{y_k} = M_k w_k.$$

Hence, the corresponding Quasi-Newton condition in a linearly constrained optimization can be obtained from forcing the next reduced Hessian update, B_{k+1}, to satisfy the equation above,

$$B_{k+1} w_k = \tilde{y_k}.$$

It follows that the BFGS formula in the linearly constraint case remains the same with changes on s_k and y_k as

$$\begin{aligned} s_k &= w_k \\ y_k &= Z^T g_{k+1} - Z^T g_k. \end{aligned}$$

If we extend the above formula to a problem with nonlinear constraints, one choice would be

$$\begin{aligned} y_k &= Z_{k+1}^T g_{k+1} - Z_k^T g_k \\ s_k &= w_k. \end{aligned}$$

Other variants of this formula are given by Nocedal and Overton in [4]. We use a slightly different formula,

$$y_k = Z_{k+}^T g_{k+} - Z_k^T g_k \tag{2.0.30}$$

$$s_k = w_k, \tag{2.0.31}$$

where $g_{k+} = \nabla F(x_k + Z_k w_k)$, Z_{k+}^T is the null space basis matrix of A_{k+}^T and $k+$ means "at $x_k + Z_k w_k$". Hence our BFGS updating formula is the same as (2.0.26) with the above formula for y_k and s_k.

2.1 Local Algorithm

We are now in a position to describe the algorithm without global consideration. In order to exploit the structure of the Jacobian in the linearized constraint using the variable reduction method we decompose the variables as follows:

$$n = n_f + n_d$$

$$x_k = (u_k, d_k)$$

where n is the number of total variables, n_f is the number of flow variables, and n_d is the number of design variables. Here u_k is the flow variable, and d_k is the design variable. Accordingly, we decompose $c'(x_k)$ $(= A_k^T$ in earlier notation) as

$$c'(x_k) = \begin{pmatrix} \nabla c_1(x_k)^T \\ \nabla c_2(x_k)^T \\ \vdots \\ \nabla c_{n_f}(x_k)^T \end{pmatrix} = (c'_f(x_k), c'_d(x_k)).$$

The previous discussion implies

$$Z_k = \begin{pmatrix} -c'_f(x_k)^{-1} c'_d(x_k) \\ I_{n_d} \end{pmatrix}, \quad Y_k = \begin{pmatrix} c'_f(x_k)^{-1} \\ 0 \end{pmatrix}.$$

With these matrices, we are given $x_k = (u_k, d_k)$, and B_k, we obtain w_k by solving

$$B_k w_k = -Z_k^T \nabla F(x_k).$$

It follows that

$$\begin{aligned} \delta_k &= (\delta_{k_f}, \delta_{k_d})^T \\ &= Z_k w_k - Y_k c(x_k) \\ &= \begin{pmatrix} -c'_f(x_k)^{-1} c'_d(x_k) \\ I_{n_d} \end{pmatrix} w_k - \begin{pmatrix} c'_f(x_k)^{-1} \\ 0 \end{pmatrix} c(x_k) \\ &= \begin{pmatrix} -c'_f(x_k)^{-1} (c'_d(x_k) w_k + c(x_k)) \\ w_k \end{pmatrix}. \end{aligned}$$

We set

$$\begin{aligned}
\delta_{k_f} &= -c'_f(x_k)^{-1}(c'_d(x_k)w_k + c(x_k)) \\
\delta_{k_d} &= w_k
\end{aligned}$$

and we get the new iterate $x_{k+1} = (u_{k+1}, d_{k+1})$, where

$$\begin{aligned}
u_{k+1} &= u_k + \delta_{k_f} \\
&= u_k - c'_f(x_k)^{-1}(c'_d(x_k)w_k + c(x_k)) \\
d_{k+1} &= d_k + \delta_{k_d} \\
&= d_k + w_k.
\end{aligned}$$

We also need to compute an intermediate point $x_k + Z_k w_k$

$$\begin{aligned}
x_k + Z_k w_k &= \begin{pmatrix} u_k \\ d_k \end{pmatrix} + \begin{pmatrix} -c'_f(x_k)^{-1}c'_d(x_k)w_k \\ w_k \end{pmatrix} \\
&= \begin{pmatrix} u_k - c'_f(x_k)^{-1}c'_d(x_k)w_k \\ d_k + w_k \end{pmatrix}.
\end{aligned}$$

Combining these results yields the following algorithm.

Algorithm 1: Given $x_k = (u_k, d_k)$, B_k;

1. **Get w_k:**

   Compute $Z_k = \begin{pmatrix} -c'_f(x_k)^{-1} c'_d(x_k) \\ \\ I_{n_d} \end{pmatrix}$

   Solve for w_k: $B_k w_k = -Z_k^T \nabla F(x_k)$

2. **Update $x_k = (u_k, d_k)$:**

   Solve for v_k: $c'_f(x_k) v_k = -(c'_d(x_k) w_k + c(x_k))$

   Set

   $u_{k+1} = u_k + v_k$

   $d_{k+1} = d_k + w_k$

3. **Get y_k:**

   Solve for η_k: $c'_f(x_k)\eta_k = -c'_d(x_k) w_k$

   Compute

   $y_k = Z(u_k + \eta_k \ d_k + w_k)^T)\nabla F(u_k + \eta_k \ d_k + w_k)^T) - Z_k^T \nabla F(x_k)$

4. **Update B_k:**

   If $y_k^T w_k$ is not too small,

   $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T w_k} - \frac{(B_k w_k)(B_k w_k)^T}{w_k^T B_k w_k}$

5. **Update variables:**

   Set

   $x_k \Leftarrow (u_{k+1} \ d_{k+1})^T$

   $B_k \Leftarrow B_{k+1}$

   Go to 1

2.2 Global Algorithm

We will globalize the algorithm 1 using an l_1 merit function:

$$\phi_{\mu_k}(x) = F(x) + \mu_k \|c(x)\|_1 \qquad (2.2.1)$$

where, μ_k is a positive constant changing with iterations.

The essential feature of the merit function is that the direction δ_k obtained from our local algorithm should be a descent direction. In order to ensure the descent direction, we use the following rule in [5] in choosing μ_k,

$$\mu_k = \begin{cases} \mu_{k-1} & if\ \mu_{k-1} \geq \|\lambda_k\|_\infty + \rho, \\ \|\lambda_k\|_\infty + 2\rho & otherwise, \end{cases} \qquad (2.2.2)$$

where ρ is some positive constant.

This formula also guarantees that for some k_0, $\mu_k = \mu$ for all $k \geq k_0$ if $\|\lambda_k\|_\infty$ is bounded.

We employ the backtracking line search strategy in Dennis & Schnabel's book-[7]. We set $\eta \in (0, 1/2)$ and when we obtain δ_k from step 2 in the algorithm 1, with $\alpha_k = 1$ we check

$$\phi(x_k + \alpha_k \delta_k) \leq \phi(x_k) + \eta \alpha_k D\phi(x_k; \delta_k). \qquad (2.2.3)$$

If (2.2.3) is not satisfied, reduce α_k and check the condition again and keep continuing this process until the condition is satisfied or the algorithm is terminated because the step size is too small. When (2.2.3) holds for some α_k we set

$$x_{k+1} = x_k + \alpha_k \delta_k.$$

In step 2 in Algorithm 1, we set

$$\begin{aligned} u_{k+1} &= u_k + \alpha_k v_k \\ d_{k+1} &= d_k + \alpha_k w_k. \end{aligned}$$

The nondifferentiability of the l_1 merit function may cause slow convergence. The algorithm may not be able to take a full step even near x_* which is necessary to obtain 2 step super linear convergence locally. To avoid this problem we add a second order correction step which is given by $so_k = -Y_k c(x_k + \delta_k)$. The effect is to decrease the quantity $\|c(x)\|$ so that it is of the order $\|e_k\|^3$ where $e_k = x_k - x_*$.

The second order correction step is implemented when we check the condition (2.2.3) with $\alpha_k = 1$. If it is not satisfied then before we reduce α_k we add a second order correction step so_k to the step $x_k + \delta_k$ then check the condition again, i.e.

$$\phi_{\mu_k}(x_k + \delta_k + so_k) \leq \phi_{\mu_k}(x_k) + \eta D\phi_{\mu_k}(x_k; \delta_k).$$

If this holds then we take this step and leave the line search routine otherwise we reduce α_k until the condition (2.2.3) is satisfied or the line search routine is terminated. In short, we try two different search direction against condition (2.2.3), i.e. δ_k and $\delta_k + so_k$ before we start reducing α_k.

Now we obtain Algorithm 2 with global consideration:

Algorithm 2: Given $x_k = (u_k, d_k)$, B_k:

1. **Get w_k:**

   Compute $Z_k = \begin{pmatrix} -c_f'(x_k)^{-1}c_d'(x_k) \\ \\ I_{n_d} \end{pmatrix}$

   Solve for w_k: $B_k w_k = -Z_k^T \nabla F(x_k)$

2. **Get δ_k,λ_k, and μ_k:**

   Solve for v_k: $c_f'(x_k)v_k = -(c_d'(x_k)w_k + c(x_k))$

   $\delta_k = (v_k, w_k)^T$

   Solve for λ_k: $c_f'(x_k)\lambda_k = \tilde{g}_k$

   Fix positive constant ρ

   $$\mu_k = \begin{cases} \mu_{k-1} & if\ \mu_{k-1} \geq \|\lambda_k\|_\infty + \rho, \\ \|\lambda_k\|_\infty + 2\rho & otherwise \end{cases}$$

3. **line search with second order correction:**

   Set $\alpha_k = 1$

   3.1:

   $u_{k+1} = u_k + \alpha_k v_k$

   $d_{k+1} = d_k + \alpha_k w_k$

   $x_{k+1} = (u_{k+1}, d_{k+1})^T$

   $fix\ \eta = 10^{-4}$

   if $\phi_{\mu_k}(x_{k+1}) \leq \phi_{\mu_k}(x_k) + \eta\alpha_k D\phi_{\mu_k}(x_k; \delta_k)$ then

   $w_k = \alpha_k w_k$

go to step 4,

otherwise

if $\alpha_k = 1$ then

$so_k = (-c_f'(x_k)^{-1} c(x_k + \delta_k)\ 0)^T$

$u_{k+1} = u_k + v_k + \tilde{so}_k,\ \tilde{so}_k = -c_f'(x_k) c(x_k + \delta_k)$

$d_{k+1} = d_k + w_k$

$x_{k+1} = (u_{k+1}, d_{k+1})$

if $\phi_{\mu_k}(x_{k+1}) \leq \phi_{\mu_k}(x_k) + \eta D\phi_{\mu_k}(x_k; \delta_k)$ then

go to step 4,

reduce α_k and go to 3.1

4. **Get** y_k:

   Solve for η_k: $c_f'(x_k)\eta_k = -c_d'(x_k) w_k$

   Compute

   $y_k = Z(u_k + \eta_k, d_k + w_k)^T \nabla F(u_k + \eta_k, d_k + w_k)^T - Z_k^T \nabla F(x_k)$

5. **Update** B_k:

   If $y_k^T w_k$ is not too small,

   $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T w_k} - \frac{(B_k w_k)(B_k w_k)^T}{w_k^T B_k w_k}$

6. **Update variables**:

   Set

   $x_k \Leftarrow (u_{k+1}, d_{k+1})^T$

   $B_k \Leftarrow B_{k+1}$

Go to 1.

Chapter 3

A GLOBAL AND LOCAL CONVERGENCE OF THE ALGORITHM

3.1 Global Behavior of the Algorithm

Our global analysis is a simple modification of the analysis by Byrd and Nocedal in [5]. In that paper they analyzed the global behavior of the same algorithm with the slightly different formulas in updating Hessians and estimating Lagrange multipliers, but the analysis was done with *the orthogonal factorization method* in which $Z_k^T Z_k = I$ is assumed. Now that this assumption no longer holds in our algorithm, the proofs will be presented without the orthogonality assumption but we will follow their framework in [5] with the l_1 merit function;

$$\phi_{\mu_k}(x) = F(x) + \mu_k \|c(x)\|_1 \tag{3.1.1}$$

We make the following assumptions.

Assumption 1 *The sequence x_k from Algorithm 1 is contained in a convex set D with the following properties,*

1. *For some positive constants m, M,*

$$
\begin{aligned}
y_k^T s_k / \|s_k\|^2 &\geq m \\
\|y_k\|^2 / y_k^T s_k &\leq M
\end{aligned}
\tag{3.1.2}
$$

hold where,

$$
B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k}. \tag{3.1.3}
$$

2. The functions $F : R^n \to R, c : R^n \to R^t$ and their first and second derivatives are uniformly bounded by a constant M_0 in D.

3. For some positive constants γ_0, δ_0,

$$
\|V_k^{-1}\| \leq \gamma_0 \tag{3.1.4}
$$

$$
\|U_k\| \leq \delta_0 \tag{3.1.5}
$$

hold, where $A_k^T = (V_k, U_k)$, and

$$
Z_k^T = \begin{pmatrix} -V_k^{-1} U_k \\ I \end{pmatrix}, Y_k = \begin{pmatrix} V_k^{-1} \\ 0 \end{pmatrix}. \tag{3.1.6}
$$

We note that Assumption 1-3 implies

$$
\|Z_k\| \leq (\gamma_0 \delta_0 + 1) \tag{3.1.7}
$$

$$
\|Y_k\| \leq \gamma_0. \tag{3.1.8}
$$

The following lemma in [5] gives the explicit formula for the directional derivative of the l_1 merit function at x_k in the direction of δ_k where δ_k is the solution of the quadratic subproblem in Algorithm 1. We provide the proof in order to use some of the equations in the following results.

Lemma 1

$$D\phi_{\mu_k}(x_k;\delta_k) = g_k^T\delta_k - \mu_k\|c_k\|_1 \tag{3.1.9}$$

if $\phi_{\mu_k}(x) = F(x) + \mu_k\|c(x)\|_1$.

Proof. From Taylor's theorem and $A_k^T\delta_k = -c_k, \alpha \leq 1,$ *and* $\mu_k > 0$, we have

$$\begin{aligned} \phi_{\mu_k}(x_k+\alpha\delta_k) - \phi_{\mu_k}(x_k) &\leq \alpha g_k^T\delta_k + \mu_k\|c_k + \alpha A_k^T\delta_k\|_1 + b_1\alpha^2\|\delta_k\|^2 - \mu_k\|c_k\|_1 \\ &\leq \alpha(g_k^T\delta_k - \mu_k\|c_k\|_1) + b_1\alpha^2\|\delta_k\|^2 \end{aligned} \tag{3.1.10}$$

for some positive b_1 that depends on μ_k.

But on the other hand, we also have

$$\begin{aligned} \phi_{\mu_k}(x_k+\alpha\delta_k) - \phi_{\mu_k}(x_k) &\geq \alpha g_k^T\delta_k + \mathcal{O}(\|\alpha\delta_k\|^2) + \mu_k\|c_k + \alpha A_k^T\delta_k\|_1 \\ &\quad -\mu_k\mathcal{O}(\|\alpha\delta_k\|^2) - \mu_k\|c_k\|_1 \\ &= \alpha(g_k^T\delta_k - \mu_k\|c_k\|_1) + \mathcal{O}(\|\alpha\delta_k\|^2) - \mu_k\mathcal{O}(\|\alpha\delta_k\|^2). \end{aligned}$$

Hence we can take a positive constant b_2 such that

$$\phi_{\mu_k}(x_k+\alpha\delta_k) - \phi_{\mu_k}(x_k) \geq \alpha(g_k^T\delta_k - \mu_k\|c_k\|_1) - b_2\alpha^2\|\delta_k\|^2 \tag{3.1.11}$$

for sufficiently small α. Now we set $b_3 = \max(b_1, b_2)$ and replace both b_1 and b_2 by b_3 in (3.1.10) and (3.1.11). If we divide (3.1.10) and (3.1.11) by α and take limits as $\alpha \to 0$, we obtain the result.

□

In the lemma below it is proved that the search direction δ_k from Algorithm 1 is a descent direction and μ_k eventually remains constant for all sufficiently large k with appropriately chosen μ_k.

Lemma 2 *If the rule below for choosing μ_k is taken, then the δ_k from Algorithm 1 is a descent direction and μ_k remains constant for all $k > k_0$ for some index k_0 when the merit function ϕ_{μ_k} in (3.1.1) is used. We have*

$$\mu_k = \begin{cases} \mu_{k-1} & \text{if } \mu_{k-1} > \|\lambda_k\|_\infty + \rho \\ \|\lambda_k\|_\infty + 2\rho & \text{otherwise} \end{cases} \tag{3.1.12}$$

where ρ is a fixed positive constant.

Proof. If we substitute $\delta_k = -Z_k B_k^{-1} Z_k^T g_k - Y_k c_k$ into the formula for $D\phi_{\mu_k}(x_k; \delta_k)$ in Lemma 1, we obtain

$$D\phi_{\mu_k}(x_k; \delta_k) = g_k^T \delta_k - \mu_k \|c_k\|_1 \tag{3.1.13}$$

$$= -g_k^T Z_k B_k^{-1} Z_k^T g_k - g_k^T Y_k c_k - \mu_k \|c_k\|_1. \tag{3.1.14}$$

Since the definition of Y_k in (3.1.6) and the Lagrange multiplier estimate in (2.0.25) imply

$$\begin{aligned} g_k^T Y_k c_k &= \tilde{g}^T V_k^{-1} c_k \\ &= \lambda_k^T c_k, \end{aligned}$$

we have

$$D\phi_{\mu_k}(x_k;\delta_k) = -g_k^T Z_k B_k^{-1} Z_k^T g_k - \lambda_k^T c_k - \mu_k \|c_k\|_1. \tag{3.1.15}$$

Also B_k^{-1} is positive definite since B_k is positive definite so that

$$-(Z_k^T g_k)^T B_k^{-1} (Z_k^T g_k) < 0.$$

Therefore,

$$\|\lambda_k^T\|_\infty < \mu_k \tag{3.1.16}$$

would be sufficient to guarantee $D\phi_{\mu_k}(x_k;\delta_k) < 0$ because

$$-\lambda_k^T c_k \leq |\lambda_k^T c_k| \leq \|\lambda_k^T\|_\infty \|c_k\|_1 < \mu_k \|c_k\|_1.$$

It is obvious that the sufficient condition (3.1.16) is satisfied if we choose μ_k according to the rule (3.1.12).

Moreover, we have from Assumption 1-2 and 1-3 that

$\|\lambda_k\|_\infty \leq \sqrt{t}\|\lambda_k\|_2$ and $\|\lambda_k\|_2 = \|V_k^{-1^T} \tilde{g}_k\| \leq \|V_k^{-1}\| \|\tilde{g}_k\| \leq \gamma_0 M_0$,

and this implies

$$\|\lambda_k\|_\infty \leq \sqrt{t}\gamma_0 M_0.$$

Hence $\|\lambda_k\|_\infty$ is bounded. Therefore, μ_k will stop increasing when $\mu_k \geq \mu_{k_0} > \sqrt{t}\gamma_0 M_0$ and $\mu_k = \mu_{k_0}$ for all $k > k_0$ for some index k_0. This completes the proof.

□

We now state the theorem by Byrd and Nocedal in [6] which gives bounds for important quantities. One of which can be used as a measure of decrease in the merit function. We will discuss this quantity more after we state the theorem.

Theorem 1 *Let $\{B_k\}$ be generated by the BFGS formula (3.1.3) where, for all $k \geq 1, s_k \neq 0$ and*

$$y_k^T s_k / \|s_k\|^2 \geq m > 0$$

$$\|y_k\|^2 / y_k^T s_k \leq M.$$

Then, for any $p \in (0,1)$, there exist constants $\beta_1, \beta_2, \beta_3 > 0$ such that, for any $k \geq 1$, the relations,

$$\cos\theta_j \geq \beta_1 \tag{3.1.17}$$

$$\beta_2 \leq \|B_j s_j\| / \|s_j\| \leq \beta_3, \tag{3.1.18}$$

hold for at least $[pk]$ values of $j \in [1,k]$ where

$$\cos\theta_j = \frac{s_j^T B_j s_j}{\|s_j\| \|B_j s_j\|}.$$

□

We now take a close look at $\cos\theta_k$. If we decompose δ_k as

$$\delta_k = h_k + v_k = Z_k w_k - Y_k c_k,$$

then (3.1.9) can be written as

$$D\phi_{\mu_k}(x_k;\delta_k) = g_k^T h_k - \lambda_k^T c_k - \mu_k \|c_k\|_1. \tag{3.1.19}$$

Since $g_k^T h_k = g_k^T Z_k Z_k^T h_k$ if $Z_k^T Z_k = I$, Byrd and Nocedal considered the cosine angle between h_k and $-Z_k Z_k^T g_k$ as a measure of decrease in the merit function. Again, if we assume $Z_k^T Z_k$ then this quantity can be computed as

$$\begin{aligned} \cos\tilde{\theta}_k &= \frac{-(Z_k Z_k^T g_k)^T h_k}{\|Z_k Z_k^T g_k\| \|h_k\|} \\ &= \frac{-g_k^T h_k}{\|Z_k^T g_k\| \|h_k\|}. \end{aligned} \tag{3.1.20}$$

Then, we also have

$$D\phi_{\mu_k}(x_k;\delta_k) = -\|Z_k^T g_k\| \|h_k\| \cos\tilde{\theta}_k - \lambda_k^T c_k - \mu_k \|c_k\|_1. \tag{3.1.21}$$

This means if $\|Z_k^T g_k\|, \|h_k\|$ are not too small, then $\cos\tilde{\theta}_k$ is required to be large for sufficient decrease. The cosine angle satisfies the relation,

$$\cos\tilde{\theta}_j = \frac{s_j^T B_j s_j}{\|s_j\| \|B_j s_j\|}.$$

However it seems that it is more appropriate to consider the projected null space component and the projected gradient, i.e., $w_k, Z_k^T g_k$, directly. If we do so, then we can avoid the use of the orthogonality of Z_k but we can still follow their analysis with a slight modification. Moreover, since $B_k w_k = -Z_k g_k$, it is likely that w_k and $-Z_k g_k$ play the roles of δ_k and $-g_k$ in a unconstrained minimization in which the above theorem is stated and proved.

Hence we focus on the cosine angle between w_k and $-Z_k g_k$ in the projected space, i.e.,

$$\cos\theta_k = \frac{-w_k^T Z_k^T w_k}{||w_k|| ||Z_k^T g_k||}. \tag{3.1.22}$$

Then (3.1.15) becomes

$$D\phi_{\mu_k}(x_k;\delta_k) = -||w_k|| ||Z_k^T g_k|| \cos\theta_k - \lambda_k^T c_k - \mu_k ||c_k||_1, \tag{3.1.23}$$

and we also have

$$\cos\theta_k = \frac{s_j^T B_j s_j}{||s_j|| ||B_j s_j||}. \tag{3.1.24}$$

The equation above implies

$$\cos\tilde{\theta_k} = \cos\theta_k,$$

but this is not surprising because if we assume $Z_k^T Z_k = I$, then since $||h_k|| = ||Z_k w_k|| = ||w_k||$, their formulas coincide with ours.

We also have the following relation which will be used in later.

$$\frac{||s_k||}{||B_k s_k||} = \frac{||w_k||}{||Z_k^T g_k||}. \tag{3.1.25}$$

The theorem claims that there are p times k number of "good" iterates. We follow Byrd and Nocedal's definition of "good" iterates.

Definition 1 *Let p in Theorem 1 have the value $p = 5/6$. We define J to be the set of iterates for which (3.1.17) and (3.1.18) hold. We will call J the set of "good" iterates.*

The following lemma gives the relation between $||w_k||$ and $||Z_k^T g_k||$ for the good iterates.

Lemma 3 *Suppose that the iterates x_k generated by Algorithm 1 satisfy Assumption 1. Then for any $j \in J$,*

$$\tfrac{1}{\beta_3}\|Z_j^T g_j\| \leq \|w_j\| \leq \tfrac{1}{\beta_2}\|Z_j^T g_j\|, \tag{3.1.26}$$

$$\|\delta_j\|^2 \leq \tfrac{\gamma_0 \delta 0}{\beta^2}\|Z_j^T g_j\|^2 + M\gamma^2\sqrt{t}\|c_j\|_1. \tag{3.1.27}$$

Proof. The first inequality follows from the relation (3.1.25). For the second inequality, by (3.1.7), (3.1.8), Assumption 1-2, the above inequality, and $\|c_j\| \leq \sqrt{t}\|c_j\|_1$, we have

$$\begin{aligned}
\|\delta_j\|^2 &= \|Z_j w_j\|^2 + \|Y_j c_j\|^2 \\
&\leq (\gamma_0\delta_0 + 1)^2\|w_j\|^2 + M\gamma_0^2\sqrt{t}\|c_j\|_1 \\
&\leq \frac{(\gamma_0\delta_0 + 1)^2}{\beta_2^2}\|Z_j^T g_j\|^2 + M\gamma_0^2\sqrt{t}\|c_j\|_1.
\end{aligned}$$

□

We have proven in Lemma 2 that the search direction δ_k from our algorithm is a descent direction, but we provide here a stronger result that gives a lower bound in reduction for the set of "good" iterates.

Lemma 4 *Let the iterates x_k be generated by Algorithm 1 with the weight chosen according to the rule (3.1.12) for all $k > 0$ so that*

$$\mu_k \geq \|\lambda_k\|_\infty + \rho. \tag{3.1.28}$$

Suppose that Assumption 1-2 are satisfied. Then for all $k \geq 1$,

$$D\phi_{\mu_k}(x_k;\delta_k) \leq -\|Z_k^T g_k\|\|w_k\|\cos\theta_k - \rho\|c_k\|_1, \tag{3.1.29}$$

and there is a positive constant b_4 such that for all $j \in J$,

$$D\phi_{\mu_j}(x_j;\delta_j) \leq -b_4[\|Z_j^T g_j\|^2 + \|c_j\|_1]. \tag{3.1.30}$$

Moreover for any value μ there is a positive constant γ'_μ such that if $j \in J$ and $\mu_j = \mu$ then

$$\phi_\mu(x_j) - \phi_\mu(x_{j+1}) \geq \gamma'_\mu[\|Z_j^T g_j\|^2 + \|c_j\|_1]. \tag{3.1.31}$$

Proof. The inequality (3.1.29) follows immediately from (3.1.23) and (3.1.28). For (3.1.30), it follows from Theorem 1 and Lemma 3 that

$$\cos\theta_j \geq \beta_1$$

$$\tfrac{1}{\beta_3}\|Z_j^T g_j\| \leq \|w_j\| \leq \tfrac{1}{\beta_2}\|Z_j^T g_j\|$$

for all $j \in J$. If we combine these two inequalities with (3.1.29) then we have

$$D\phi_{\mu_j}(x_j;\delta_j) \leq -\frac{\beta_1}{\beta_3}\|Z_j^T g_j\|^2 - \rho\|c_j\|_1,$$

and if we set $b_4 = \min(\beta_1/\beta_3, \rho)$, we obtain (3.1.30).

To consider the decrease in ϕ_μ, for $j \in J$, the line search condition (2.2.3) enforces

$$\phi_{\mu_j}(x_j) - \phi_{\mu_j}(x_{j+1}) \geq -\eta\alpha_j D\phi_{\mu_j}(x_j;\delta_j). \tag{3.1.32}$$

Combining this with (3.1.30), we obtain

$$\phi_{\mu_j}(x_j) - \phi_{\mu_j}(x_{j+1}) \geq \eta b_4\alpha_j[\|Z_j^T g_j\|^2 + \|c_j\|_1]. \tag{3.1.33}$$

Now it remains to show that α_j is bounded below to achieve (3.1.31).

Suppose $\alpha_j < 1$, this means that the line search condition failed for some step length $\tilde{\alpha}$, where $\tau\tilde{\alpha} \leq \alpha_j$ (in practice $\tau = 0.1$). The failure of the line search condition implies

$$\phi_{\mu_j}(x_j + \tilde{\alpha}\delta_j) - \phi_{\mu_j}(x_j) > \eta\tilde{\alpha}D\phi_{\mu_j}(x_j;\delta_j), \tag{3.1.34}$$

and (3.1.10) with α replaced by $\tilde{\alpha}$ yields

$$\phi_{\mu_j}(x_j + \tilde{\alpha}\delta_j) - \phi_{\mu_j}(x_j) \leq \tilde{\alpha}D\phi_{\mu_j}(x_j;\delta_j) + \tilde{\alpha}^2 b_1\|\delta_j\|^2. \tag{3.1.35}$$

Equations (3.1.34) and (3.1.35) are combined to yield

$$(\gamma - 1)D\phi_{\mu_j}(x_j;\delta_j) < \tilde{\alpha}b_1\|\delta_j\|^2. \tag{3.1.36}$$

We also have an upper bound of $\|\delta_j\|^2$ from Lemma 3,

$$(\eta - 1)D\phi_{\mu_j}(x_j;\delta_j) < \tilde{\alpha}b_1\frac{(\gamma_0\delta_0 + 1)^2}{\beta_2^2}\|Z_j^T g_j\|^2 + \tilde{\alpha}b_1 M\gamma_0^2\sqrt{t}\|c_j\|_1, \tag{3.1.37}$$

and if we set $b_5 = \max(\frac{(\gamma_0\delta_0+1)^2}{\beta_2^2}, M\gamma_0^2\sqrt{t})$ and define $K_j \equiv \|Z_j^T g_j\|^2 + \|c_j\|_1$, we have

$$-\tilde{\alpha}b_1 b_5 K_j < (1 - \gamma)D\phi_{\mu_j}(x_j;\delta_j). \tag{3.1.38}$$

Now (3.1.30) and $(1 - \eta) > 0$ imply

$$(1 - \eta)D\phi_{\mu_j}(x_j;\delta_j) \leq -b_4(1 - \eta)K_j. \tag{3.1.39}$$

By combining the last two inequalities, we obtain

$$\tilde{\alpha} > \frac{b_4(1 - \eta)}{b_1 b_5} > 0,$$

hence by taking $\gamma'_\mu = \frac{b_4(1-\eta)}{b_1 b_5}$ we establish (3.1.31) and the proof is complete.

□

The proof of the following theorem is in [5], it is provided for completeness.

Theorem 2 *Let the sequence x_k be generated by Algorithm 1 using the l_1 merit function with μ_k chosen by the rule (3.1.12). Suppose that Assumption 1 is satisfied. Then*

$$\liminf_{k\to\infty} (\|Z_k^T g_k\| + \|c_k\|) = 0. \tag{3.1.40}$$

Proof. By Lemma 2 and Theorem 1 there exists an index k_0 such that $\mu = \mu_k$ for all $k > k_0$ and there exists a set J of good iterates . And also Lemma 2 guarantees that $\phi_\mu(x_k)$ decreases at each iterate, hence we have for $k > k_0$,

$$\begin{aligned}
\phi_\mu(x_{k_0}) - \phi_\mu(x_{k+1}) &= \sum_{j=k_0}^{k} (\phi_\mu(x_j) - \phi_\mu(x_{j+1})) \\
&\geq \sum_{j\in J\cap[k_0,k]} (\phi_\mu(x_j) - \phi_\mu(x_{j+1})) \\
&\geq \gamma'_\mu \sum_{j\in J\cap[k_0,k]} [\|Z_j^T g_j\|^2 + \|c_j\|_1].
\end{aligned}$$

By assumption 1-2 $\phi_\mu(x)$ is bounded below for all $x \in D$, so the sum is finite, therefore

$$\lim_{j\in J, j\to\infty} (\|Z_j^T g_j\| + \|c_j\|_1) = 0, \tag{3.1.41}$$

and by Theorem 1, J is infinite as $k \to \infty$ the theorem follows.

□

3.2 Local Behavior of the Algorithm

In this section we now consider a local minimizer x_* which satisfies the second order sufficiency conditions and we will include these conditions in our assumptions. Even though the Algorithm is expected to obtain super linear convergence rate, only local convergence will be shown and we leave super linear convergence analysis for future work. Most results from [5] hold with our algorithm but we must convince ourselves of a few assertions and change the proofs since we do not have the orthogonality condition on $Z(x)$. The noticeable difference can be observed in showing Lipschitz type continuity of $Z(x)$ because we do not rely on QR decomposition to update Z_k.

We consider a local minimizer x_*. Our goal here is to show that the algorithm is locally R-linearly convergent if x_* satisfies the second order sufficiency conditions.

We make the following assumptions:

Assumption 2 *The point x_* is a local minimizer for problem (2.0.2) at which the following conditions hold:*

1. The functions $F(x) : R^n \to R$ and $c_i(x) : R^n \to R$ for all $i = 1, ..., t$ are twice continuously differentiable and their Hessians are Lipschitz continuous in a neighborhood of x_*, where $c(x) = (c_1(x), ..., c_t(x))^T$.

2. We decompose $A(x)^T$ as

$$A(x)^T = (V(x), U(x))$$

where $V(x)$ is a $t \times t$ block submatrix and $U(x)$ is a $t \times (n-t)$ block submatrix of $A(x)^T$.

Then $V(x)$ is invertible at x_*, and x_* satisfies the second order sufficiency conditions. Hence, there exists a Lagrange multiplier λ_* such that

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = g(x_*) - A(x_*)\lambda_* = 0. \tag{3.2.1}$$

3. For all $w \in R^{n-t}$, $w \neq 0$, we have

$$w^T G_* w > 0 \tag{3.2.2}$$

where

$$G(x, \lambda) = Z(x)^T (\nabla^2_{xx} \mathcal{L}(x, \lambda) - \sum_{i=1}^{t} \lambda^i \nabla^2_{xx} c_i(x)) Z(x), \tag{3.2.3}$$

and $G_* = G(x_*, \lambda_*)$.

4. The line search has the property that, for x_k sufficiently close to x_* and for large enough μ,

$$\phi_\mu((1-\theta)x_k + \theta x_{k+1}) \leq \phi_\mu(x_k) \tag{3.2.4}$$

for all $\theta \in [0, 1]$.

Before we state some consequences of these assumptions we provide a lemma necessary for our analysis, it can be found in [16].

We define $L(R^p, R^q)$ to be the space of linear mapping from R^p to R^q.

Lemma 5 *Let $A, C \in L(R^n, R^n)$ and assume that A is invertible, with $\|A\| \leq \alpha$. If $\|A - C\| \leq \beta$ and $\alpha\beta < 1$, then C is also invertible, and*

$$\|C^{-1}\| \leq \alpha/(1 - \alpha\beta). \tag{3.2.5}$$

The subsequent lemmas are the consequences of Assumption 2. $B(x, a)$ denotes $\{y : \|y - x\| < a\}$ and $\bar{B}(x, a)$ denotes $\{y : \|y - x\| \leq a\}$ respectively.

Lemma 6 *There exists a $\delta > 0$ such that $V(x)$ is invertible and $\|V(x)^{-1}\| \leq \gamma_3$ in $\bar{B}(x_*, \delta)$ for some constant γ_3, moreover $V(x)^{-1}$ is Lipschitz continuous in $\bar{B}(x_*, \delta)$,i.e.,*

$$\|V(x)^{-1} - V(y)^{-1}\| \leq \gamma_2 \|x - y\| \quad \forall x, y \in \bar{B}(x_*, \delta). \tag{3.2.6}$$

Proof. Let $\|V(x_*)\| = \alpha$. Then by Lemma 5 and the continuity of $V(x)$ at x_*, there exist β and δ_1 such that $\alpha\beta < 1$ and

$$\|V(x) - V(x_*)\| < \beta \quad \forall x \in B(x_*, \delta_1)$$

moreover, $V(x)$ is invertible and

$$\|V(x)^{-1}\| \leq \alpha/(1 - \alpha\beta) \quad \forall x \in B(x_*, \delta_1).$$

By taking γ_3 as $\alpha/(1 - \alpha\beta)$, we obtain

$$\|V(x)^{-1}\| \leq \gamma_3 \quad \forall x \in \bar{B}(x_*, r), \tag{3.2.7}$$

when $r < \delta_1$.

Now by Assumption 2-1 there exist δ_2, γ_1 such that

$$\|V(x) - V(y)\| \leq \gamma_1 \|x - y\| \quad \forall x, y \in \bar{B}(x_*, \delta_2). \tag{3.2.8}$$

If we take a $\delta < \min(\delta_1, \delta_2)$, then we have for all x, y in $\bar{B}(x_*, \delta)$,

$$\begin{aligned}
\|V(x)^{-1} - V(y)^{-1}\| &= \|V(x)^{-1}(V(x) - V(y))V(y)^{-1}\| \\
&\leq \alpha\gamma_3\|V(x) - V(y)\| \\
&\leq \alpha\gamma_3\gamma_1\|x - y\|,
\end{aligned}$$

hence $V(x)^{-1}$ is Lipschitz continuous in $\bar{B}(x_*, \delta)$ with $\gamma_2 = \alpha\gamma_3\gamma_1$.

□

The next lemma will show that the Lagrange multiplier estimate formula in (2.0.25) as a function of x is Lipschitz continuous in a neighborhood of x_*.

Lemma 7 *There exists a $\delta > 0$ such that the Lagrange multiplier estimate $\lambda(x) = V(x)^{T^{-1}}\tilde{g}(x)$ is Lipschitz continuous in $\bar{B}(x_*, \delta)$, that is, there exists a γ_L such that*

$$\|\lambda(x) - \lambda(y)\| \leq \gamma_L\|x - y\| \quad \forall x, y \in \bar{B}(x_*, \delta). \tag{3.2.9}$$

Proof. By (3.2.6),(3.2.7), and the fact that $g(x)$ is also Lipschitz continuous in a neighborhood of x_*, there is a δ such that for all $x, y \in \bar{B}(x_*, \delta)$,

$$\begin{aligned}
\|V(x)^{T^{-1}}\| &\leq \gamma_3 \\
\|\tilde{g}(x) - \tilde{g}(y)\| &\leq \gamma_4\|x - y\| \\
\|V(x)^{T^{-1}} - V(y)^{T^{-1}}\| &\leq \gamma_2\|x - y\|.
\end{aligned}$$

We have

$$
\begin{aligned}
\|\lambda(x) - \lambda(y)\| &= \|V(x)^{T^{-1}}\tilde{g}(x) - V(y)^{T^{-1}}\tilde{g}(y)\| \\
&\leq \|V(x)^{T^{-1}}\tilde{g}(x) - V(x)^{T^{-1}}\tilde{g}(y)\| \\
&\quad + \|V(x)^{T^{-1}}\tilde{g}(y) - V(y)^{T^{-1}}\tilde{g}(y)\| \\
&\leq \|V(x)^{T^{-1}}\|\|\tilde{g}(x) - \tilde{g}(y)\| + \|V(x)^{T^{-1}} - V(y)^{T^{-1}}\|\|\tilde{g}(y)\| \\
&\leq \gamma_3\gamma_4\|x - y\| + max\{\|\tilde{g}(z)\| : z \in \bar{B}(x_*, \delta)\}\gamma_2\|x - y\| \\
&= (\gamma_3\gamma_4 + \gamma_2 max\{\|\tilde{g}(z)\| : z \in \bar{B}(x_*, \delta)\})\|x - y\|.
\end{aligned}
$$

Therefore, we obtain (3.2.9) with

$$\gamma_L = \gamma_3\gamma_4 + \gamma_2 max\{\|\tilde{g}(z)\| : z \in \bar{B}(x_*, \delta)\}.$$

□

In the Orthogonal Factorization Method, much effort was made to ensure the property of the null space basis in this lemma, for example see [17], [18], and [19]. But in our case (the Variable Reduction Method), this property can be obtained from the definition of the null space basis that we have chosen.

Lemma 8 *There exists a constant $\hat{a}$ and a neighborhood of x_* such that for all x in the neighborhood,*

$$\|Z(x) - Z(x_*)\| \leq \hat{a}\|x - x_*\|, \tag{3.2.10}$$

where, Z(x) is the null space basis matrix defined in (2.0.22).

Proof. From the formula for $Z(x)$ we have

$$\|Z(x) - Z(x_*)\| = \|V(x)^{-1}U(x) - V(x_*)^{-1}U(x_*)\|,$$

and

$$\begin{aligned} \|V(x)^{-1}U(x) - V(x_*)^{-1}U(x_*)\| &\leq \|V(x)^{-1}U(x) - V(x)^{-1}U(x_*)\| \\ &\quad + \|V(x)^{-1}U(x_*) - V(x_*)^{-1}U(x_*)\| \\ &\leq \|V(x)^{-1}\|\|U(x) - U(x_*)\| \\ &\quad + \|V(x)^{-1} - V(x_*)^{-1}\|\|U(x_*)\|. \end{aligned}$$

By Assumption 2-2 and Lemma 6 we have γ_2, γ_3, and γ_5, all positive such that

$$\begin{aligned} \|V(x)^{-1}U(x) - V(x_*)^{-1}U(x_*)\| &\leq \gamma_3\gamma_5\|x - x_*\| + \gamma_2\|U(x_*)\|\|x - x_*\| \\ &= (\gamma_3\gamma_5 + \gamma_2\|U(x_*)\|)\|x - x_*\|. \end{aligned}$$

The result follows if we set $\hat{a} = \gamma_3\gamma_5 + \gamma_2\|U(x_*)\|$.

$\square$

Lemma 9 *Let $G(x)$ be a Lipschitz continuous function from R^n to $L(R^p, R^p)$ in a neighborhood of c. Assume $G(x)$ is symmetric for all $x \in R^n$. If $G(c)$ is positive definite then $G(x)$ is also positive definite in a neighborhood of c, and there exist m, M such that for all $w \in R^p$ and for all x in that neighborhood of c,*

$$m\|w\|^2 \leq w^T G(x) w \leq M\|w\|^2. \tag{3.2.11}$$

Proof. Since $G(c)$ is symmetric positive definite there exist α, β, for example, α:the smallest eigenvalue, β:the largest eigenvalue of $G(c)$ such that

$$\alpha\|w\|^2 \le w^T G(c) w \le \beta \|w\|^2.$$

Then,

$$\begin{aligned} |w^T G(x) w - w^T G(c) w| &\le \|w\|^2 \|G(x) - G(c)\| \\ &\le \|w\|^2 K \|x - c\| \end{aligned}$$

for some constant K and for all x in a neighborhood of c by the hypothesis.

Hence, for $w \neq 0$ in R^p, we have

$$\left|\frac{w^T G(x) w}{w^T w} - \frac{w^T G(c) w}{w^T w}\right| \le K\|x - c\|.$$

If $\|x - c\| < \alpha/2K$, then

$$\alpha/2 < \frac{w^T G(x) w}{w^T w} < \beta + \alpha/2.$$

By taking $m = \alpha/2, M = \beta + \alpha/2$, we obtain (3.2.11), and the positive definiteness of $G(x)$ in a neighborhood of c follows from (3.2.11).

□

The following lemma originated in the paper [2] by M.J.D. Powell. It indicates that $c(x)$ and $Z(x)^T g(x)$ may be regarded as a measure of the error at x. The proof of this lemma is identical to that in [5] except for one place where the orthogonality of $Z(x)$ is needed.

Lemma 10 *If Assumption 2 hold, then for all x sufficiently near x_*,*

$$\gamma_6||x - x_*|| \leq ||c(x)|| + ||Z(x)^T g(x)|| \leq \gamma_7||x - x_*|| \tag{3.2.12}$$

for some positive constants γ_6, γ_7.

Proof. Define the function $H : R^{n+t} \rightarrow R^{n+t}$ by

$$H(x, \lambda) = \begin{pmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ \\ c(x) \end{pmatrix}.$$

Note that $H(x_*, \lambda_*) = 0$, and $H(x, \lambda)$ is also differentiable since each component function is differentiable in a neighborhood of x_*.

We define a norm $|| \cdot ||_e$ on R^{n+t} by

$$||(u^T, v^T)^T||_e = ||u|| + ||v|| \quad \forall u \in R^n, \forall v \in R^t,$$

by the corresponding induced matrix norm , for $(n + t) \times (n + t)$ matrices.

We will show that

$$\gamma_6||x - x_*|| \leq ||H(x, \lambda(x))||_e \leq \gamma_7||x - x_*|| \tag{3.2.13}$$

for x sufficiently near x_* where $\lambda(x)$ is given by the formula in (2.0.25).

Also, we establish the equality

$$||H(x, \lambda(x))||_e = ||Z(x)^T g(x)|| + ||c(x)||, \tag{3.2.14}$$

which will yield the theorem.

The Jacobian of $H(x, \lambda)$ at (x_*, λ_*) is given by

$$H'(x_*, \lambda_*) = \begin{pmatrix} \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*) & -A(x_*) \\ -A(x_*)^T & 0 \end{pmatrix}. \tag{3.2.15}$$

We need to show that $H'(x_*, \lambda_*)$ is nonsingular. Suppose that

$$H'(x_*, \lambda_*)(u^T \; v^T) = 0$$

for some $u \in R^n$ and $v \in R^t$, then

$$\begin{aligned} \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)u - A(x_*)v &= 0 && (3.2.16) \\ A(x_*)^T u &= 0. && (3.2.17) \end{aligned}$$

This implies that u belongs to the null space of $A(x_*)^T$ hence there exists a $w \in R^{n-t}$ such that $u = Z(x_*)w$. On the other hand, if we premultiply (3.2.17) by u^T we have $u^T\nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)u = 0$, which implies

$$w^T Z(x_*)^T \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)Z(x_*)w = 0. \tag{3.2.18}$$

Now Assumption 2-3 gives $w = 0, u = 0$ and also $v = 0$ because $A(x_*)$ has full rank. Hence $H'(x_*, \lambda_*)$ is nonsingular.

The differentiability of H at (x_*, λ_*) implies that for any $\epsilon > 0$,

$$\|H(x, \lambda) - H'(x_*, \lambda_*)\begin{pmatrix} x - x_* \\ \lambda - \lambda_* \end{pmatrix}\|_e \leq \epsilon(\|x - x_*\| + \|\lambda - \lambda_*\|),$$

for all (x, λ) sufficiently close to (x_*, λ_*).

Hence we obtain

$$\|H(x, \lambda)\|_e \leq (\|H'(x_*, \lambda_*)\|_e + \epsilon)(\|x - x_*\| + \|\lambda - \lambda_*\|). \tag{3.2.19}$$

Now if we substitute $\lambda(x)$ for λ in the Lagrange estimate formula (2.0.25), then by Lemma 7 we have for all x sufficiently near x_*

$$\|H(x, \lambda(x))\|_e \leq \gamma_7 \|x - x_*\|, \tag{3.2.20}$$

where $\gamma_7 = (\|H'(x_*, \lambda_*)\|_e + \epsilon)(1 + \gamma_L)$.

Since for any $z \in R^{n+t}$ and $A, A^{-1} \in L(R^{n+t}, R^{n+t})$

$$\|x\|_e = \|Ix\|_e = \|A^{-1}Ax\|_e \leq \|A^{-1}\|_e \|Ax\|_e,$$

we have

$$\frac{\|x\|_e}{\|A^{-1}\|_e} \leq \|Ax\|_e. \tag{3.2.21}$$

We obtain for all x sufficiently near x_*

$$\gamma_6 \|x - x_*\| \leq \|H(x, \lambda(x))\|_e, \tag{3.2.22}$$

where $\gamma_6 = 1/\|H'(x_*, \lambda_*)^{-1}\|_e - \epsilon$ and assume that ϵ is small enough so that $1/\|H'(x_*, \lambda_*)^{-1}\|_e - \epsilon$ is positive.

We obtain (3.2.13) by combining the inequalities (3.2.20) and (3.2.22).

It remains to show the equality (3.2.14). It will be established by showing that

$$\|\nabla_x \mathcal{L}(x, \lambda(x))\| = \|Z(x)^T g(x)\|. \tag{3.2.23}$$

We note that $\lambda(x)$ solves

$$V(x)^T\lambda(x) = \tilde{g}(x), \tag{3.2.24}$$

where,

$$A(x)^T = (V(x)\ U(x)),\ \mathit{and}$$

$$g(x) = \begin{pmatrix} \tilde{g}(x) \\ \\ g_2(x) \end{pmatrix},$$

$\tilde{g}(x)$ contains the first t components of the gradient vector $g(x)$.

Then, we have

$$\begin{aligned}
\nabla_x \mathcal{L}(x, \lambda(x)) &= g(x) - A(x)\lambda(x) \\
&= \begin{pmatrix} \tilde{g}(x) - V(x)^T\lambda(x) \\ \\ g_2(x) - U(x)^T\lambda(x) \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ \\ g_2(x) - U(x)^T\lambda(x) \end{pmatrix}.
\end{aligned}$$

On the other hand, our formula for $Z(x)$ in (2.0.22) yields

$$\begin{aligned}
Z(x)^T g(x) &= \begin{pmatrix} -V(x)^{-1}U(x) \\ \\ I \end{pmatrix} \begin{pmatrix} \tilde{g}(x) \\ \\ g_2(x) \end{pmatrix} \\
&= -U(x)^T V(x)^{T^{-1}} \tilde{g}(x) + g_2(x) \\
&= -U(x)^T \lambda(x) + g_2(x)
\end{aligned}$$

Hence we obtained (3.2.23) and this completes the proof.

□

The following lemma and theorem will be stated without proofs, because the same proofs in [5] work for our case without modification. But we note that all the lemmas that were used in the proofs are established here without using an orthogonality assumption on $Z(x)$. The following lemma shows that the merit function may be regarded as a measure of error.

Lemma 11 *Suppose that Assumption 2 hold at x_*. Then for any $\mu > \|\lambda_*\|_\infty$, there exist constants γ_8 and γ_9, such that for all $\|x - x_*\|$ sufficiently small,*

$$\gamma_8 \|x - x_*\|^2 \leq \phi_\mu(x) - \phi_\mu(x_*) \leq \gamma_9(\|Z(x)^T g(x)\|^2 + \|c(x)\|_1) \tag{3.2.25}$$

□

Now the following theorem shows that once x_k falls in some neighborhood of x_*, then x_k converges to x_* R-linearly, and this section is completed.

Theorem 3 *Let $\{x_k\}$ be generated by Algorithm 1 using the l_1 merit function, with μ_k chosen by (2.2.2). Suppose that x_* satisfies Assumption 2, and that $\{\|\lambda(x_k)\|\}$ is bounded. Then μ is a fixed value for all sufficiently large k, and there is a neighborhood of x_* such that if any iterate x_{k_0} falls in that neighborhood, with $\mu_{k_0} = \mu$, then $\{x_k\}$ converges to x_*. Furthermore*

$$\phi_\mu(x_{k+1}) - \phi_\mu(x_*) \leq r^{k-k_0}(\phi_\mu(x_{k_0}) - \phi_\mu(x_*)), \;\; k \geq k_0 \tag{3.2.26}$$

for some constant $r < 1$, and

$$\sum_{k=1}^{\infty} \|x_k - x_*\| < \infty.$$

$\square$

Chapter 4

NUMERICAL IMPLEMENTATION AND RESULTS

4.1 Numerical Implementation

We recall the discretized design problem:

given $\hat{u}_j$ for $j = 1, ..., J$, seek u_j which

$$\begin{aligned} &\text{minimizes} \quad \tfrac{1}{2}\textstyle\sum_{j=1}^{J} |u_j - \hat{u}_j|^2 \\ &\text{subject to} \quad c_j \equiv \tfrac{f_{j+1/2}-f_{j-1/2}}{h} + g_j = 0 \end{aligned} \tag{4.1.1}$$

where, $f(u) \equiv u + \bar{H}/u$, $g(u,\xi) \equiv \frac{A_\xi}{A}(\bar{\gamma}u - \bar{H}/u)$. The domain of both the velocity function $u(\xi)$ and the duct area function $A(\xi)$ is [0 1]. We divided [0 1] into equally spaced J number of cell centered grid with a center in the jth grid having coordinate $\xi_j = (j - 1/2)h$ where $h = 1/J$ and those functions will be evaluated at the center. $u(\xi_j)$ will be denoted by u_j. We took $J = 40$ for the experiment.

By choosing appropriate schemes for computing fluxes $f_{j+1/2}$, the solution of the following subproblem (nonlinear system of equations) approximates the solution of the continuous analysis problem (1.1.5).

$$c_j \equiv \frac{f_{j+1/2} - f_{j-1/2}}{h} + g_j = 0 \tag{4.1.2}$$

where, $f(u) \equiv u + \bar{H}/u, \quad g(u,\xi) \equiv \frac{A_\xi(\xi)}{A(\xi)}(\bar{\gamma}u - \bar{H}/u)$, $\bar{\gamma} = 1/6$, $\bar{H} = 1.2$.

The functions $A(\xi)$, $A_\xi(\xi)$ are the evaluations of the cubic spline defined by B-splines and its derivative at ξ. We also note that $A(\xi)$ and $A_\xi(\xi)$ depends on the design parameters. The boundary conditions, $u_0 = 1.299$ and $u_J = 0.506$ are imposed at the inlet and outlet.

Three different constraint functions can be formulated depending on how we compute $f_{j+1/2}$ and $f_{j-1/2}$, i.e., the Artificial Viscosity, Enquist-Osher, and Godunov schemes making the constraint function C^∞,C^1, and C^0 respectively.

In order to see how the numerical solutions different from the analytic solution we compared an analytic solution with different numerical solutions in figures 4.1.1 through 4.1.3.

As we mentioned earlier in section 1, if we treat this subproblem-(4.1.2) as a constraint while we consider both velocities and design parameters as independent variables then we obtain an Equality constrained formulation called *All at once method.* On the other hand, if the only variables are design parameters and computing objective function involves solving the flow equation with given design variables, then we are led to *Black box method.*

Design variables are the coefficients of B-splines which generates duct shapes. We have six parameters to fix to determine a unique spline in the interval, two of them are design variables which give heights at two mid points, another two of them are heights at two both ends in our case we fixed them as $A(0) = 1.05$ and $A(1) = 1.745$, and the last two

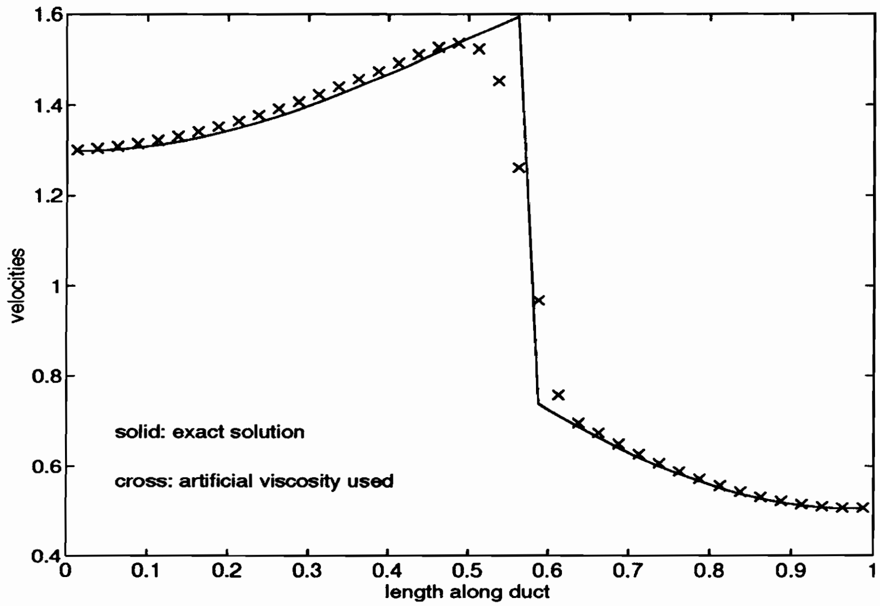

Figure 4.1.1: Exact solution versus Numerical solution using Artificial Viscosity scheme.

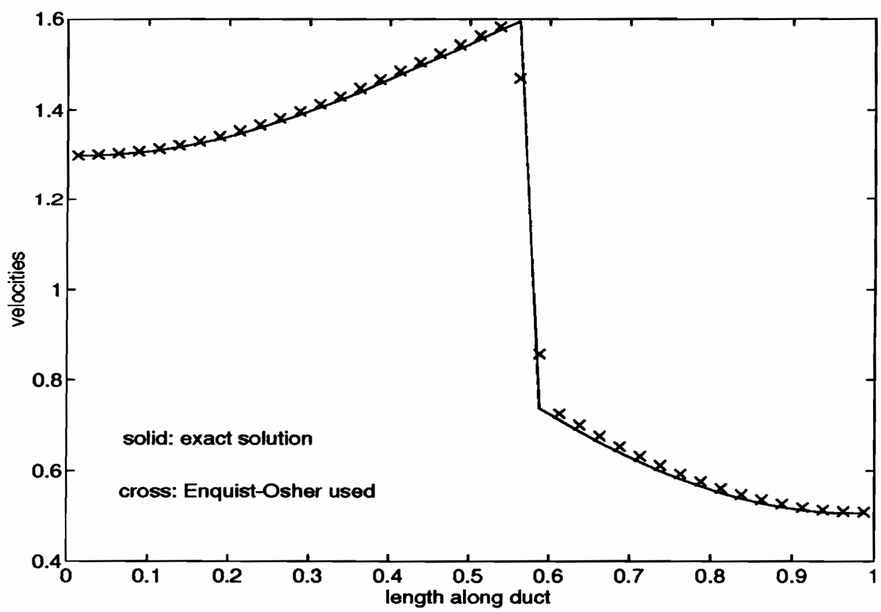

Figure 4.1.2: Exact solution versus Numerical solution using Enquist-Osher scheme.

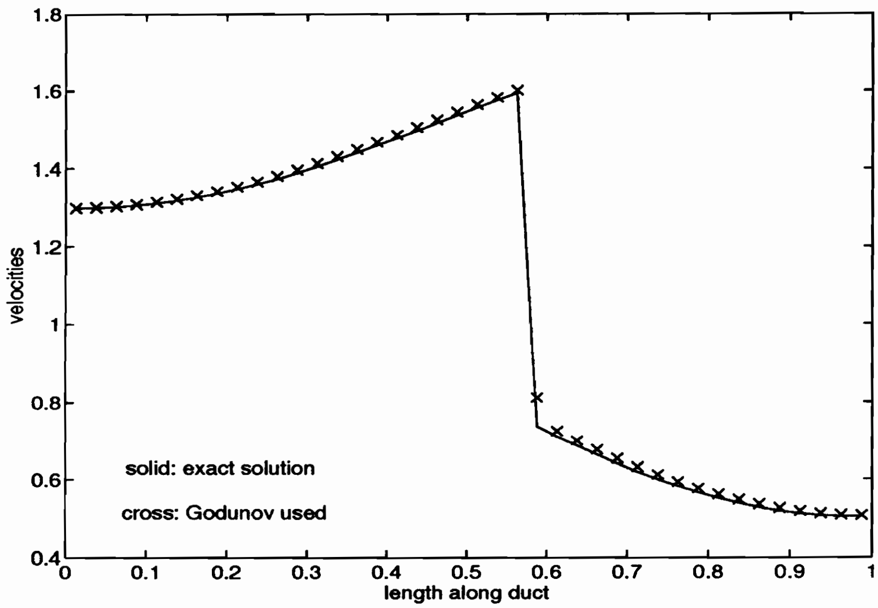

Figure 4.1.3: Exact solution versus Numerical solution using Godunov scheme.

parameters fix the slopes at both ends we gave both zero slopes.

In order to generate a target duct shape and a goal velocity field we fixed the design parameter. A spline generating routine gives a duct shape and for the goal velocities we used the scheme in [1] to solve the flow equation analytically or computationally in some cases.

Both optimization approaches were considered, unconstrained formulation (Black box method) and equality constrained formulation (All at once method), but we focus on the latter. Some routines are based on the pseudo codes in [7] and LAPACK routines are used for linear algebra manipulations.

We consider the unconstrained formulation first. We only need to define an objective function:

$$F(d) = 1/2 \sum_{j=1}^{J} |u_j - \hat{u}_j|^2$$

where $d = (d_1 \dots d_{n_d})$, design parameters, and $\hat{u}_j$ is a goal velocities, u_j is computed by solving the following nonlinear system of equations

$$c(u, d) = \frac{f_{j+1/2} - f_{j-1/2}}{h} + g_j = 0 \tag{4.1.3}$$

where, $f(u) \equiv u + \bar{H}/u$, $g(u, \xi) \equiv \frac{A_\xi(\xi)}{A(\xi)}(\bar{\gamma} u - \bar{H}/u)$, $\bar{\gamma} = 1/6$, $\bar{H} = 1.2$. and $x = (u\ d)$, u is the flow variable of dimension $n_f (= J)$ and d is design variable of dimension n_d. We take $n_f = 40$ and $n_d = 2$ for the experiment. We used a forward difference approximation for the first derivative of the objective function and the Hessian is updated using the BFGS formula so there is no need for the second derivative.

For the equality constrained formulation, described by Algorithm 2, we need to define an objective function $F(x)$ and a constraint function $c(x)$,

$$F(x) = 1/2 \sum_{j=1}^{J} |u_j - \hat{u}_j|^2$$

where $x = (u, d)$, $u = (u_1, \dots, u_J)$, and $d = (d_1, \dots, d_{n_d})$. The equation (4.1.3) becomes a constraint in the Equality constrained formulation.

For the equality constrained optimization experiment, in order to implement Algorithm 2 successfully to the design problem it is crucial that $c'_f(x_k)$ is nonsingular. Also to see the advantage that we claimed, we should show that it is a tridiagonal matrix.

Theorem 4 *In step 1 in Algorithm 2, the $n_f \times n_f$ matrix $c'_f(x_k)$, $n_f = J$ for our case, is tridiagonal and nonsingular with the following assumptions:*

1. *there exists only one index $k(1 < k < J)$ such that*

$$u_k > u_* > u_{k+1}.$$

2. *For some positive ρ and σ*

$$u_j \geq u_* + \rho \quad \forall j < k$$

and

$$u_j \leq u_* - \sigma \quad \forall j > k$$

where u_* denotes the sound of speed.

3. *For some positive β*

$$A_\xi(d, \xi_j) \geq \beta > 0,$$

and u_j is bounded above by $M > 0$ and $1 \leq A(d, u_j) \leq 2$.

Proof. We provide detailed proofs for the Godunov Scheme and the Enquist-Osher scheme. Since very similar argument applies to the proof for the Enquist-Osher scheme, we only provide the structure of $c'_j(x)$ for the Enquist-Osher scheme and made the conclusion from there. First it is easy to see that the matrix is tridiagonal because if we let $x = (u, d)$,

$$c_j(x) = \frac{(f_{j+1/2} - f_{j-1/2})}{h} + \frac{A_\xi(d,\xi_j)}{A(d,\xi)}(\bar{\gamma} u_j - \bar{H}/u_j),$$

$f_{j+1/2}$ in (1.1.7) is a function of u_j, u_{j+1}, and u_* and $f_{j-1/2}$ is also a function of u_{j-1}, u_j, and u_*. It follows that

$$\frac{\partial c_j(x)}{\partial u_i} = 0$$

if $|i - j| > 1$. And this shows that $c'_f(x)$ is a tridiagonal matrix.

The nonsingularity of the matrix $c'_f(x)$ will be shown by establishing that $\det c'_f(x) > 0$. Because of Assumption 1 and the definition of $f_{j+1/2}$ and $f_{j-1/2}$ in Godunov scheme, four different $c_j(x)$'s can be formed depending on the index j, i.e., $j < k$, $j = k$, $j = k+1$, and $j > k+1$. For each case we will show that

$$\frac{\partial c_j(x)}{\partial u_j} > 0, \quad \det c'_f = \prod_{i=1}^{J} \frac{\partial c_j(x)}{\partial u_j}$$

which yields $\det c'_f(x) > 0$.

Case 1: $j < k$. If $u_{j-1}, u_j, and\ u_{j+1} \geq u_*$ then

$$\begin{aligned} c_j(x) &= \frac{f_j - f_{j-1}}{h} + g(x,\xi_j) \\ &= \frac{u_j + \bar{H}/u_j - (u_{j-1} + \bar{H}/u_{j-1})}{h} + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} u_j - \bar{H}/u_j). \end{aligned}$$

Hence

$$\frac{\partial c_j(x)}{\partial u_i} = 0 \quad \text{unless} i = j-1\ or\ j$$

$$\begin{aligned}\frac{\partial c_j(x)}{\partial u_j} &= (1-\bar{H}/u_j^2)/h + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} + \bar{H}/u_j^2) \\ \frac{\partial c_j(x)}{\partial u_{j-1}} &= (-1+\bar{H}/u_{j-1}^2)/h.\end{aligned}$$

We note from $u_j \geq u_* + \rho$ that

$$\begin{aligned}1-\bar{H}/u_j^2 &\geq \frac{2\rho u_* + \rho^2}{u_j^2} \\ &\geq \frac{2\rho u_* + \rho^2}{M^2} \\ &> 0.\end{aligned}$$

And $A_\xi \geq \beta$, $A \leq 2$ imply

$$\frac{\partial c_j(x)}{\partial u_j} \geq (\frac{2\rho u_* + \rho^2}{M^2})/h + \frac{\beta}{2}(\bar{\gamma} + \bar{H}/M^2) > 0.$$

Case 2: $j > k+1$. If $u_{j-1}, u_j,$ *and* $u_{j+1} \leq u_*$ then we obtain

$$\begin{aligned}c_j(x) &= \frac{f_{j+1} - f_j}{h} + g(x,\xi_j) \\ &= \frac{u_{j+1} + \bar{H}/u_{j+1} - (u_j + \bar{H}/u_j)}{h} + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma}u_j - \bar{H}/u_j).\end{aligned}$$

Hence,

$$\frac{\partial c_j(x)}{\partial u_i} = 0 \quad \text{unless } i = j+1 \textit{ or } i = j.$$

In a similar fashion as above we can show that

$$\frac{\partial c_j(x)}{\partial u_j} \geq (\sigma u_*/M^2 + \sigma/M)/h + \beta/2(\bar{\gamma} + \bar{H}/M^2) > 0.$$

Case 3: $j = k$. If $u_j > u_*, u_{j+1} < u_*$, then

$$c_j(x) = \frac{max(f_j, f_{j+1}) - f_{j-1}}{h} + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma}u_j - \bar{H}/u_j).$$

We have two different $\frac{\partial c_j(x)}{\partial u_j}$s depending on the max function above,

$$\frac{\partial c_j(x)}{\partial u_j} = \begin{cases} (1 - \bar{H}/u_j^2)/h + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} + \bar{H}/u_j^2) & \text{if } f_j \geq f_{j+1} \\ \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} + \bar{H}/u_j^2) & \text{otherwise} \end{cases}$$

but we always get positive $\frac{\partial c_j(x)}{\partial u_j}$. We can easily show positiveness using a similar argument in case 1. We note that if $f_j < f_{j+1}$ we obtain three non-zeros, $\frac{\partial c_j(x)}{\partial u_{j-1}}$, $\frac{\partial c_j(x)}{\partial u_j}$, and $\frac{\partial c_j(x)}{\partial u_{j+1}}$.

Case 4: if $j = k + 1$. If $u_{j-1} > u_*, u_j < u_*$, then

$$c_j(x) = \frac{(f_{j+1} - f_j) - max(f_{j-1}, f_j)}{h} + \frac{A_\xi(d, \xi_j)}{A(d, \xi_j)}(\bar{\gamma} u_j - \bar{H}/u_j)$$

so we again obtain two different $\frac{\partial c_j(x)}{\partial u_j}$s,

$$\frac{\partial c_j(x)}{\partial u_j} = \begin{cases} \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} + \bar{H}/u_j^2) & \text{if } f_{j-1} \geq f_j \\ (-1 + \bar{H}/u_j^2)/h + \frac{A_\xi(d,\xi_j)}{A(d,\xi_j)}(\bar{\gamma} + \bar{H}/u_j^2) & \text{otherwise.} \end{cases}$$

In this case also we have positive $\frac{\partial c_j(x)}{\partial u_j}$, and if $f_j \geq f_{j+1}$ we obtain three non-zeros, $\frac{\partial c_j(x)}{\partial u_{j-1}}$, $\frac{\partial c_j(x)}{\partial u_j}$, and $\frac{\partial c_j(x)}{\partial u_{j+1}}$.

Therefore, the matrix has a very nice structure, there are only diagonal and subdiagonal non-zero elements when $j < k$ and there are three non-zero elements when $j = k$ or $j = k+1$ (only once) and there are only diagonal and super diagonal non-zero elements after the row with three non-zero elements. The following picture illustrates the structure of the matrix $c_j'(x)$,

if $f_j < f_{j+1}$ then

$$
c'_j(x) = \left[\begin{array}{ccccccccccc}
d_1 & & & & & & & & & & \\
X & d_2 & & & & & & & & & \\
 & X & d_3 & & & & & & \text{\Large 0} & & \\
 & & & \ddots & & & & & & & \\
 & & & & \ddots & & & & & & \\
 & & & & X & d_k & X & & & & \\
 & & & & & & d_{k+1} & X & & & \\
 & & \text{\Large 0} & & & & & \ddots & & & \\
 & & & & & & & & \ddots & & \\
 & & & & & & & & & d_{J-1} & X \\
 & & & & & & & & & & d_J
\end{array} \right]
$$

otherwise

$$
c_j'(x) = \begin{bmatrix}
d_1 & & & & & & & & & & \\
X & d_2 & & & & & & & & & \\
 & X & d_3 & & & & & \mathbf{0} & & & \\
 & & & \ddots & & & & & & & \\
 & & & & \ddots & & & & & & \\
 & & & & X & d_k & & & & & \\
 & & & & & X & d_{k+1} & X & & & \\
 & & \mathbf{0} & & & & & \ddots & & & \\
 & & & & & & & & \ddots & & \\
 & & & & & & & & & d_{J-1} & X \\
 & & & & & & & & & & d_J
\end{bmatrix}
$$

From this we can conclude that $\det c_f'(x) = \prod_{i=1}^{J} \frac{\partial c_j(x)}{\partial u_j}$ is greater than zero in either case so that $c_f'(x)$ is nonsingular.

Now for the Enquist-Osher scheme, similar argument leads to the following structure of the matrix $c_j'(x)$,

$$
c_j'(x) = \begin{bmatrix}
d_1 & & & & & & & & & \\
\mathrm{X} & d_2 & & & & & & & & \\
& \mathrm{X} & d_3 & & & & \mathbf{0} & & & \\
& & & \ddots & & & & & & \\
& & & & \ddots & & & & & \\
& & & & \mathrm{X} & d_k & \mathrm{p} & & & \\
& & & & & \mathrm{q} & d_{k+1} & \mathrm{X} & & \\
& & \mathbf{0} & & & & & \ddots & & \\
& & & & & & & & \ddots & \\
& & & & & & & & d_{J-1} & \mathrm{X} \\
& & & & & & & & & d_J
\end{bmatrix}
$$

If we denote the 2 by 2 matrix in the middle block by $K_{2,2}$, i.e.,

$$
K_{2,2} = \begin{pmatrix} d_k & p \\ q & d_{k+1} \end{pmatrix}
$$

then, since

$$
\det c_j'(x) = \Big(\prod_{i \neq k, i \neq k+1}^{J} d_i \Big) \det K_{2,2},
$$

a similar argument for Godunov scheme shows

$$\prod_{i \neq k, i \neq k+1}^{J} d_i > 0,$$

it remains to show that $\det K_{2,2} \neq 0$.

The definition of the Enquist-Osher scheme gives us

$$\begin{aligned}
K_{2,2} &= \begin{pmatrix} \frac{\partial c_k}{\partial u_k} & \frac{\partial c_k}{\partial u_{k+1}} \\ \frac{\partial c_{k+1}}{\partial u_k} & \frac{\partial c_{k+1}}{\partial u_{k+1}} \end{pmatrix} \\
&= \begin{pmatrix} d_k & p \\ q & d_{k+1} \end{pmatrix}
\end{aligned}$$

where

$$\begin{aligned}
d_k &= (1 - \bar{H}/u_k^2)/h + \frac{A_\xi(d, \xi_k)}{A(d, \xi_k)}(\bar{\gamma} + \bar{H}/u_k^2) \\
p &= (1 - \bar{H}/u_{k+1}^2)/h \\
q &= (-1 + \bar{H}/u_k^2)/h \\
d_{k+1} &= (-1 + \bar{H}/u_{k+1}^2)/h + \frac{A_\xi(d, \xi_{k+1})}{A(d, \xi_{k+1})}(\bar{\gamma} + \bar{H}/u_{k+1}^2).
\end{aligned}$$

Hence we obtain

$$\begin{aligned}
\det K_{2,2} &= 1/h(1 - \bar{H}/u_k^2)\frac{A_\xi(d, \xi_{k+1})}{A(d, \xi_{k+1})}(\bar{\gamma} + \bar{H}/u_{k+1}^2) \\
&\quad +1/h(-1 + \bar{H}/u_{k+1}^2)\frac{A_\xi(d, \xi_k)}{A(d, \xi_k)}(\bar{\gamma} + \bar{H}/u_k^2) \\
&\quad +\frac{A_\xi(d, \xi_k)}{A(d, \xi_k)}\frac{A_\xi(d, \xi_{k+1})}{A(d, \xi_{k+1})}(\bar{\gamma} + \bar{H}/u_{k+1}^2) \\
&\quad > 0.
\end{aligned}$$

We can compute an actual lower bound with a similar argument in case 1 for the proof of the Godunov scheme. This completes the proof.

□

4.2 Numerical Results

First we make some observation on the design problem with the black box approach because this experiment gives us insight into how the presence of the shock affects the optimization problem. Some difficulties that arise in this approach leads us to consider the equality constrained formulation of the problem.

In figures 4.2.1 through 4.2.3, we plot the objective function $F(d)$ on the (d_1, d_2) plane with velocities computed analytically, using Godunov scheme, and using Artificial Viscosity.

The objective function plot in figure 4.2.1 shows the difficulty of finding a minimum point. There is a big deep valley and the whole picture looks like a stairs. The phenomenon can be explained as follows: we took 40 nodes (centers of the cell) in [0 1], if the shock of the goal velocities is located in between, for example, the 18th and 19th node, then the difference between any solution velocities with the shock lying in between the 18th and 19th node and goal velocities is small. Hence all the design pairs in the valley generate the same location of shocks as goal velocities but if a shock is one node away from the shock of the goal velocities we take one step up from the valley. This explains the presence of the stairs and the long valley. Although the bottom looks flat, if we zoom in we can see the existence of the minimum point, in fact, we plotted contour lines to see this.

When we use the Godunov scheme we observed smeared out edges. The Artificial Viscosity scheme almost removed all the edges which means less trouble in running the optimization but we can not expect good matching. Also the ellipsoids in contours with the artificial viscosity scheme are thicker than those with the Godunov scheme (see Figure 4.2.6).

In figures 4.2.4 through 4.2.6, we plot three contours of the objective functions computed analytically, using the Godunov scheme, and the artificial viscosity scheme respectively. Thick black lines indicate steep jumps in values.

We plot the objective function with one design variable fixed while the other varies in figure 4.2.7. We computed velocities using the Godunov scheme. The picture shows that even though the use of the Godunov scheme makes sharp edges a lot smoother compared with those in figure 4.2.1 there are still some quite sharp edge and spots with almost zero slope which gives us trouble because this could mean nondifferentiability or gradient zero.

In summary we observed in the Black box approach that, as we expected, the optimization runs are very costly because we need to solve the nonlinear system of equations to evaluate the objective function, especially when we compute gradients using forward finite difference scheme. Another difficulty is, as the plot of objective function shows, the presence of many possible local minima.

Now we turn to the equality constrained formulation of the problem. Although algorithm 2 is expected to obtain 2-step super linear convergence, with assumptions that both objective function and constraint function are twice continuously differentiable, we found one example in which we can see this convergence rate, but in general we could not observe this convergence rate in the 1-D model duct problem.

Because methods other than the Artificial Viscosity scheme do not generate a C^2 constraint function, we used the A.V. scheme. We set up the goal velocity in the objective function using the A.V. scheme so that we can expect perfect matching if the optimization succeeds. The design variables that generate a goal velocity are $d_1 = 1.12$, $d_2 = 1.43$. Initial designs are $d_1 = 1.26$, $d_2 = 1.67$ and for the initial velocity we used A.V. to compute those with design variables, $d_1 = 1.20$ and $d_2 = 1.60$. In figure 4.2.8 we show the initial, goal, and final (computed) duct shape and we also show initial, goal, and final velocities evaluated at cell centered nodes in figure 4.2.9.

In figures 4.2.10 through 4.2.15, we used analytically computed goal velocities and we tried Artificial Viscosity, Enquist-Osher, and Godunov schemes to match goal velocities.

For the comparison of our approach with black box approach, since we do not solve the underlying ordinary differential equation exactly, our approach is time efficient. We also note that we could save a lot of time because we were able to compute the Jacobian of the $n_f \times n_f$ block matrix of $c'_f(x)$ analytically which is more than 95% of whole Jacobian, due to the decomposition of variables. We observed that the black-box code took much more time in solving the discretized ordinary differential equation as we expected and sometimes the Newton solver had trouble solving with the given design data. In that case that the code looped over until it reaches the iteration limit.

In terms of robustness our all at once approach with decomposition of variables is quite

Table 4.1: The CPU time comparison of All At Once versus Black Box

	Initial design	Objective function	Scheme for initial vel.	Scheme for goal vel.	No. of iterations	CPU time
B.B.	1.10,1.47	4.36-04	Godunov	Analytic	10	2m 30s
A.O.	1.10,1.47	4.09-04	Godunov	Analytic	58	17.6s
B.B.	1.10,1.47	4.36-04	Artificial	Analytic	10	2m 22s
A.O.	1.10,1.47	4.86-04	Artificial	Analytic	27	9.2s
B.B.	1.20,1.60	4.36-04	Godunov	Analytic	13	4m 49s
A.O.	1.20,1.60	4.71-04	Godunov	Analytic	37	12.0s
B.B.	1.20,1.60	4.36-04	Artificial	Analytic	13	4m 42s
A.O.	1.20,1.60	4.45-04	Artificial	Analytic	62	17.4s
B.B.	1.14,1.48	4.36-04	Artificial	Analytic	10	4m 38s
A.O.	1.14,1.48	1.50-03	Artificial	Analytic	9	5.5s
B.B.	1.20,1.60	6.06-08	Artificial	Artificial	10	5m 23s
A.O.	1.20,1.60	1.19-07	Artificial	Artificial	92	32.5s

robust. But we can not claim that it is as robust as black box approach. The location of the shock of the initial velocity is almost always far from the goal velocity.

The performance of our approach versus the black-box approach is compared in table 4.1. We compared the performance by measuring CPU times. The goal velocity is computed analytically with the design variables $d_1 = 1.12, d_2 = 1.43$ except in the last experiment. We computed the initial guesses for the velocities using both the Artificial Viscosity scheme and the Godunov scheme. The computation was done on Sun Sparc-station server 1000. B.B. and A.O. means Black box and All at once respectively. The Godunov scheme was used for the first four runs and the Artificial Viscosity was used for the rest.

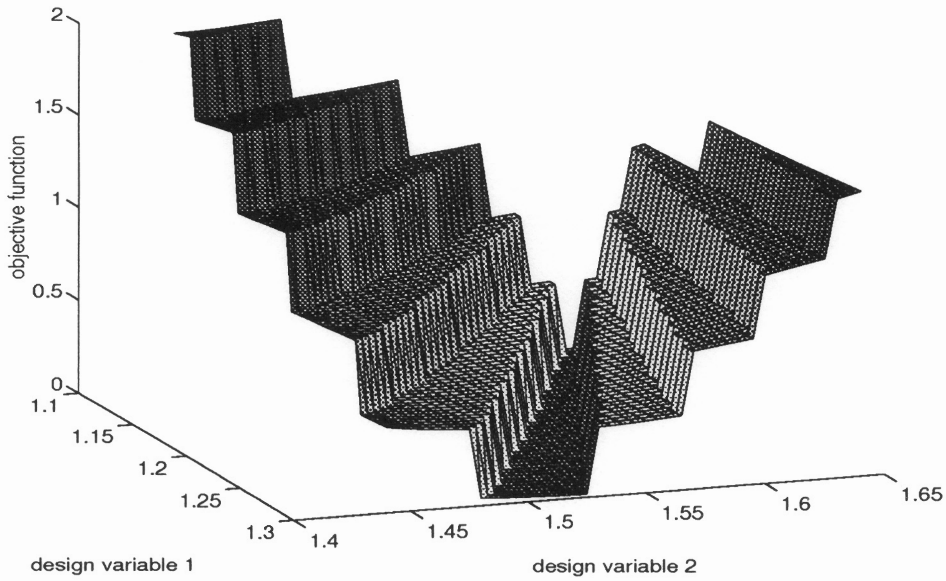

Figure 4.2.1: 3D objective function plot using exact solution.

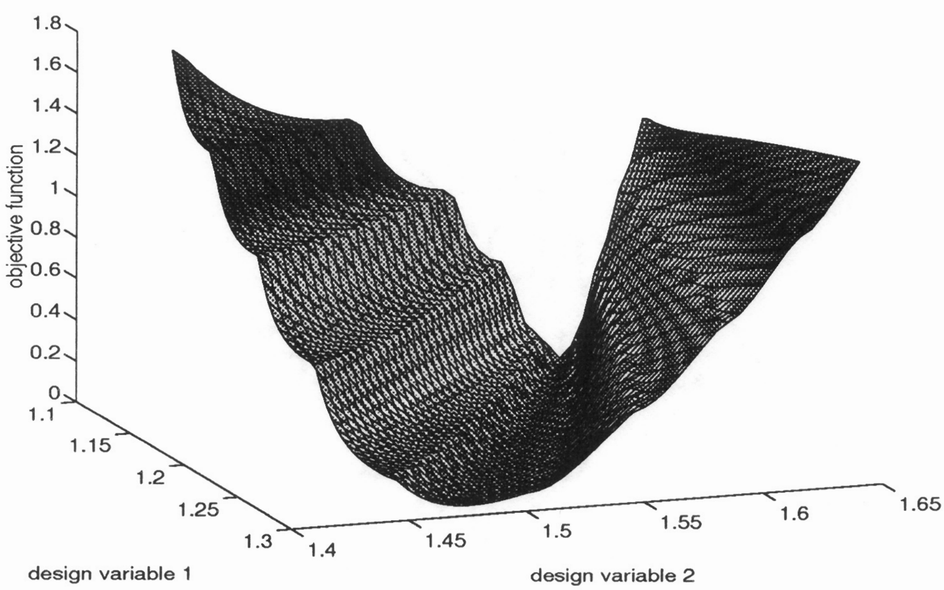

Figure 4.2.2: 3D objective function plot using Godunov scheme.

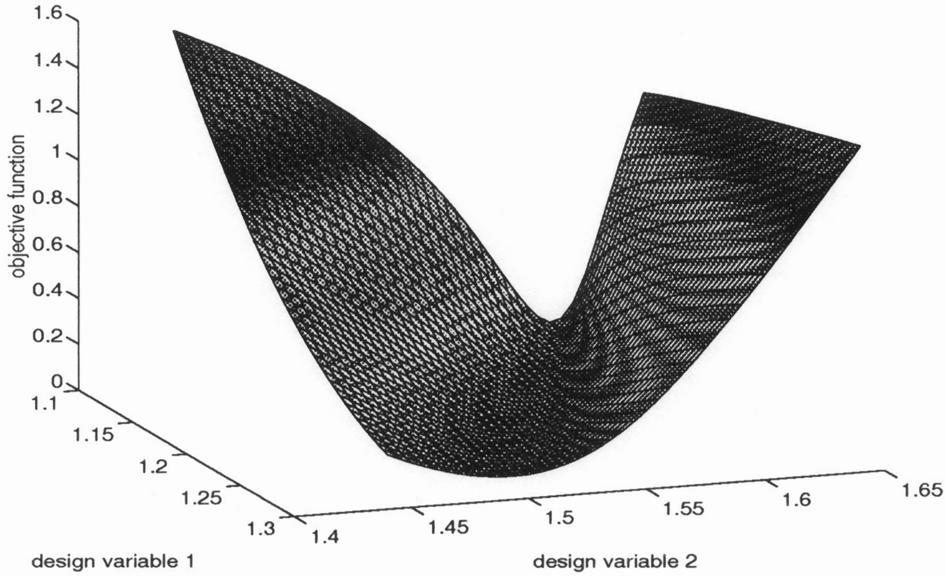

Figure 4.2.3: 3D objective function plot using Artificial Viscosity scheme.

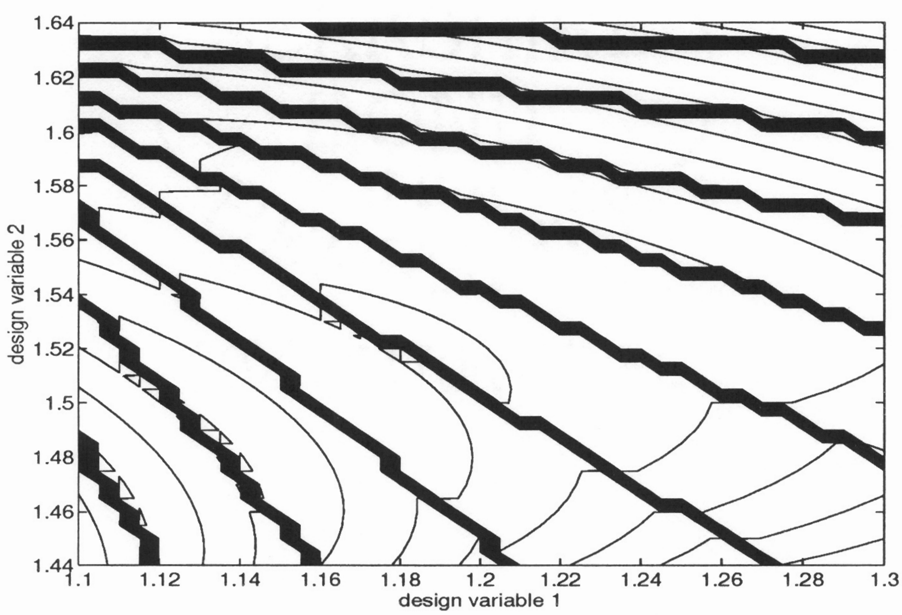

Figure 4.2.4: Contour plot of objective function using exact solution.

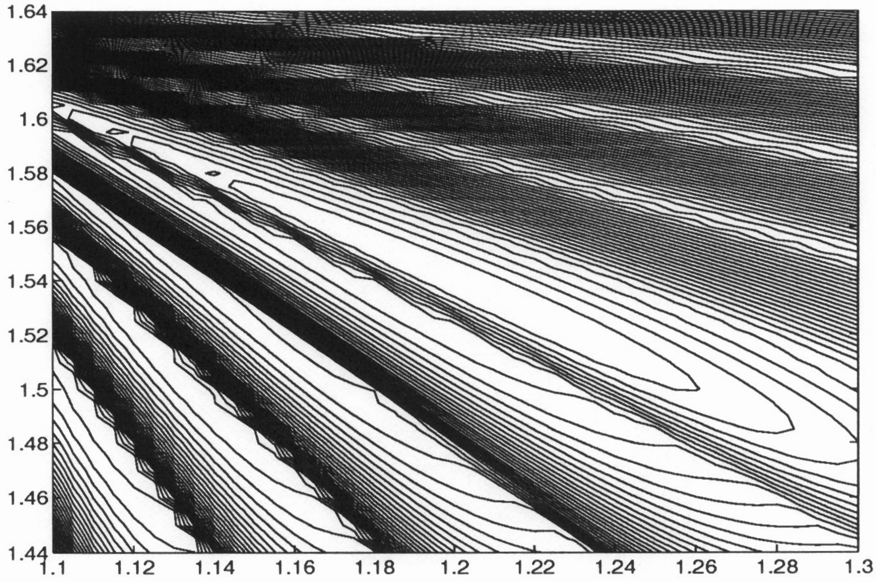

Figure 4.2.5: Contour plot of objective function using Godunov scheme.

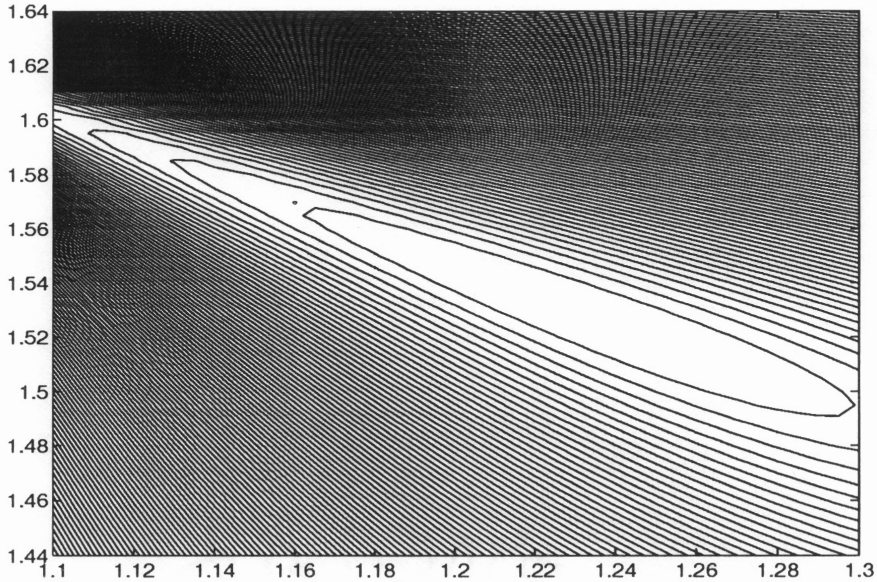

Figure 4.2.6: Contour plot of objective function using Artificial Viscosity scheme.

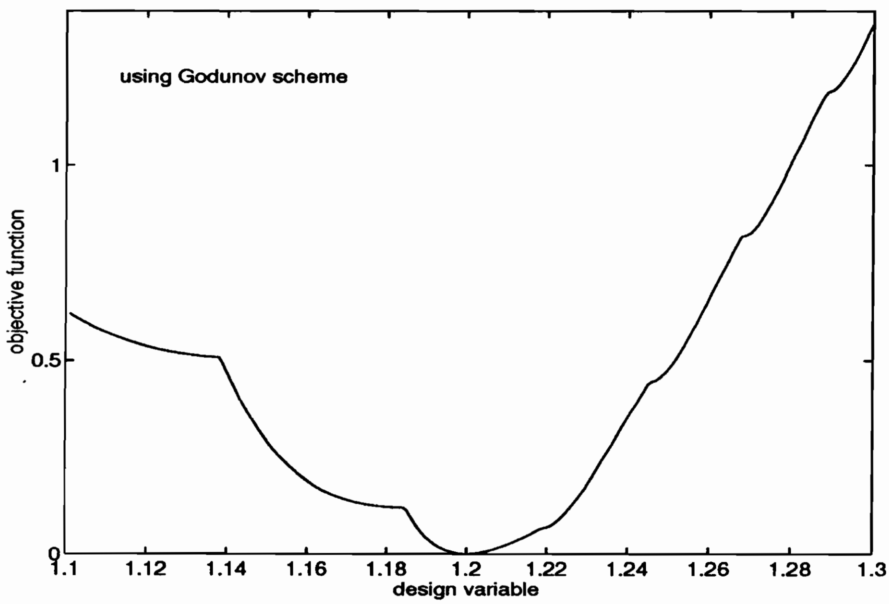

Figure 4.2.7: Cross Section of Objective function.

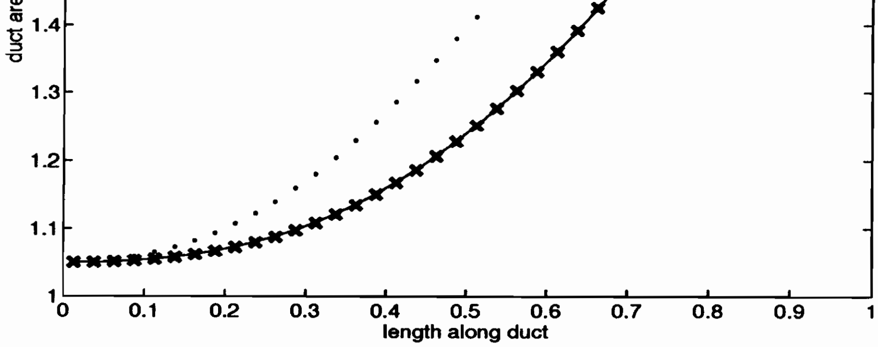

Figure 4.2.8: Duct area matching-Artificial Viscosity used.

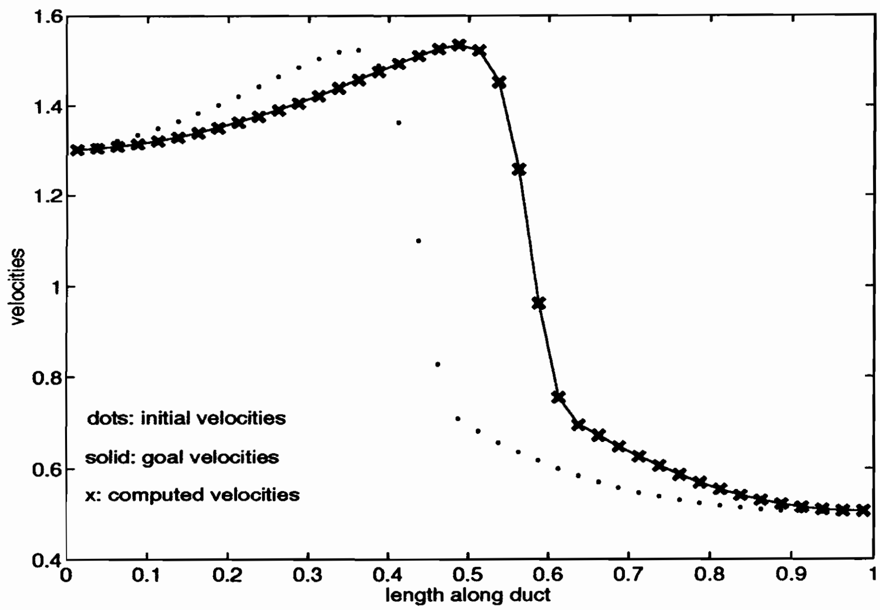

Figure 4.2.9: Velocity matching-Artificial Viscosity used.

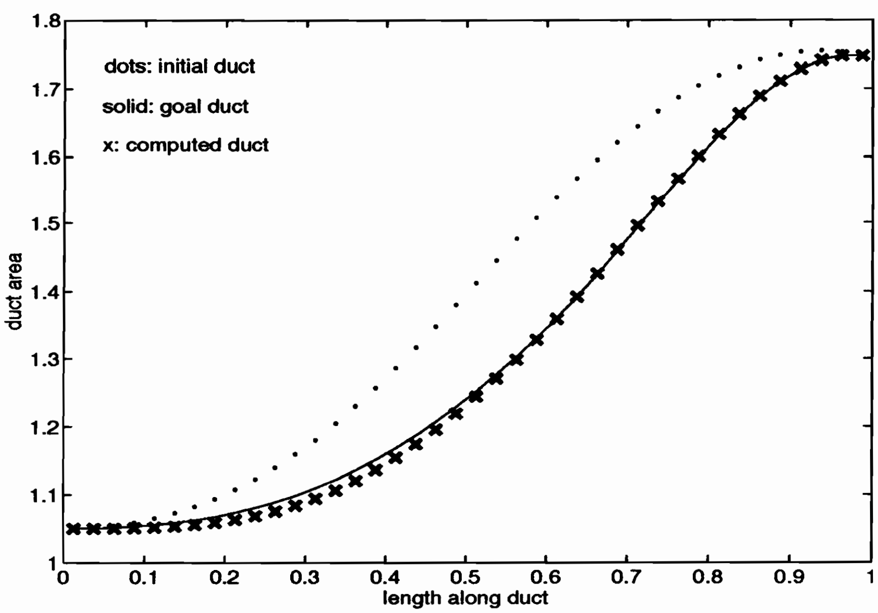

Figure 4.2.10: Duct area matching-Artificial Viscosity used.

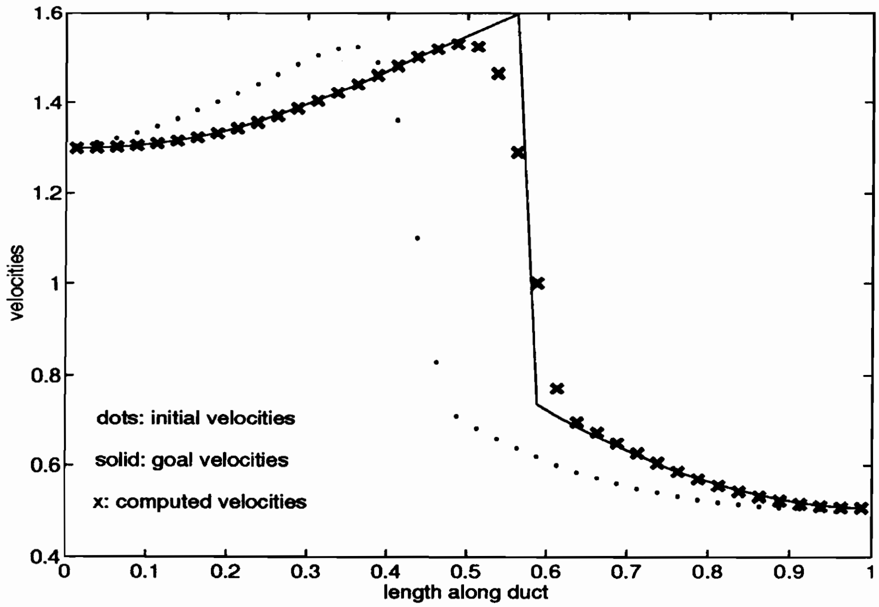

Figure 4.2.11: Velocity matching-Artificial Viscosity used.

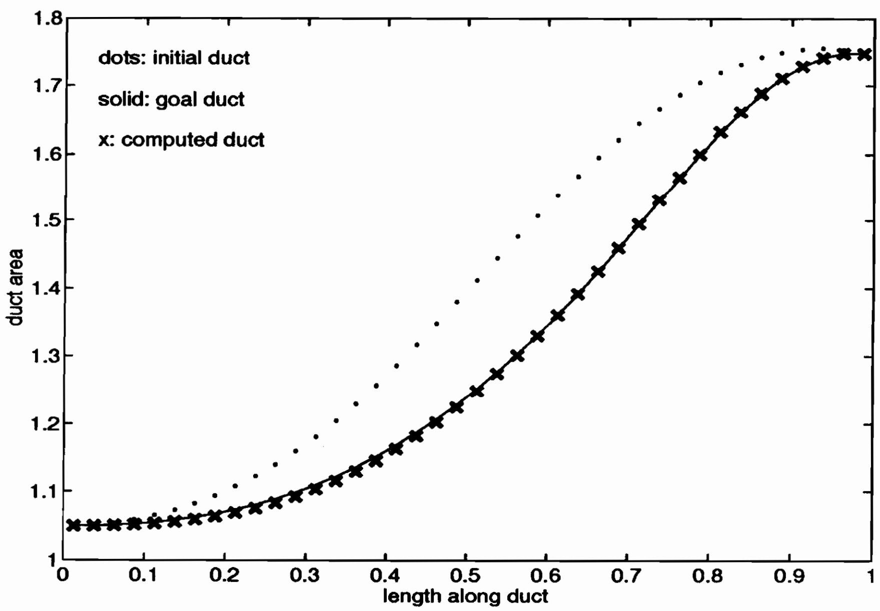

Figure 4.2.12: Duct area matching-Enquist-Osher used.

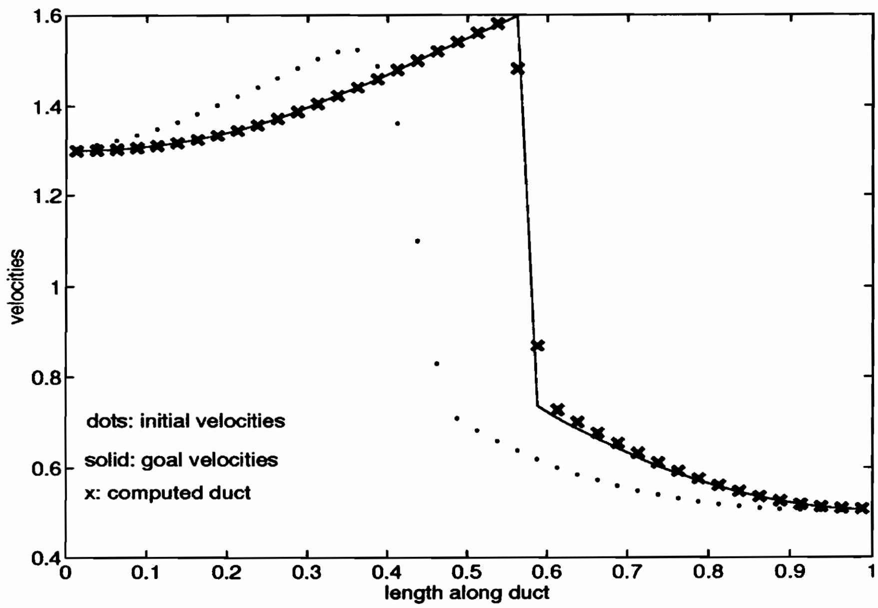

Figure 4.2.13: Velocity matching-Enquist-Osher used.

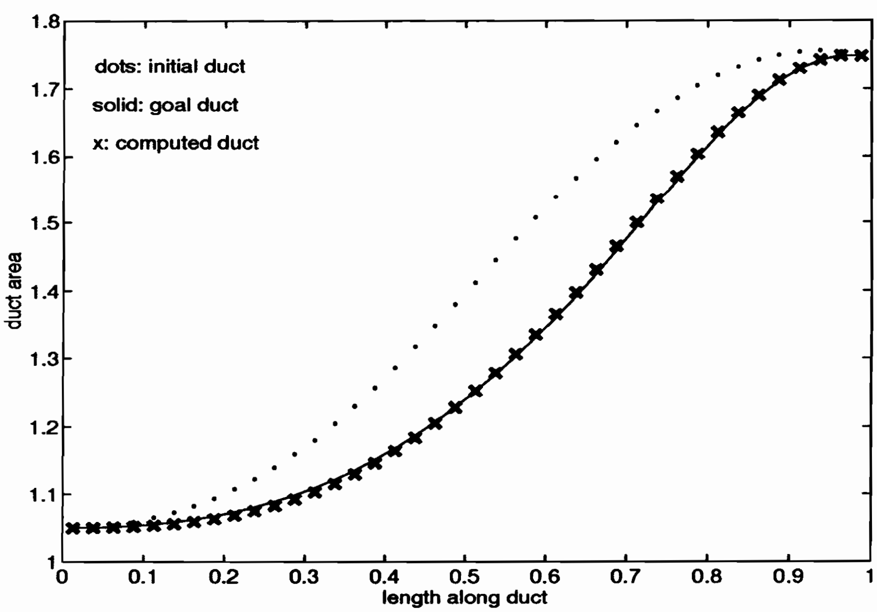

Figure 4.2.14: Duct area matching-Godunov used.

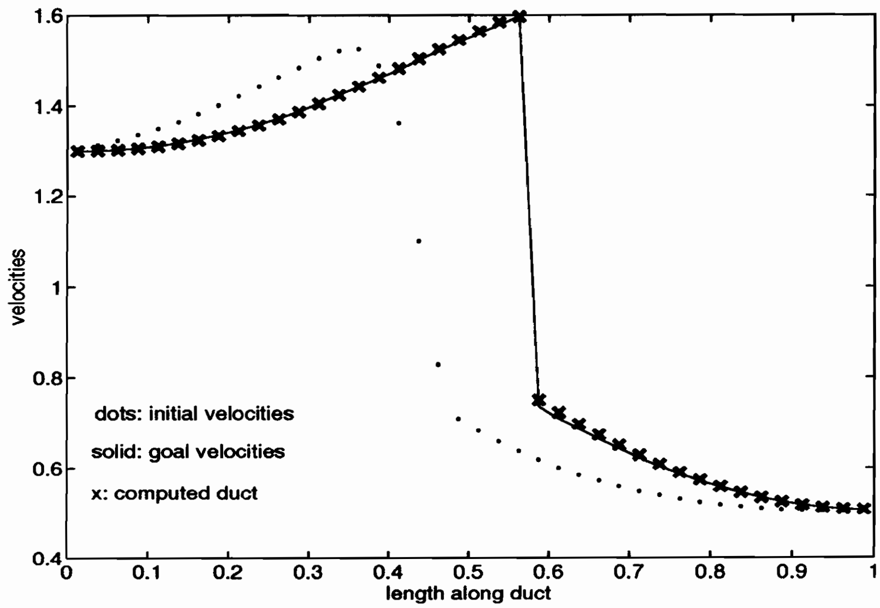

Figure 4.2.15: Velocity matching-Godunov used.

4.3 Conclusion

We applied a particular optimization method to solve a model 1-D design problem. An equality constrained optimization formulation with rather simple line search globalization is expected to reduce the adverse effect of a shock and the variable reduction method in representing the null space basis matrices is efficient in reducing the amount of work. Numerical experiment shows that the all at once approach should be emphasized as a time efficient method. In addition, the variable reduction method allows us to compute most of the Jacobian matrix analytically, which saves a great deal of time. If we take more grid points in the space dimension then the savings could be even more. Moreover, since the dimension of the Hessian is the number of design variables, the reduced Sequential

Quadratic Programming reduces the storage requirement.

During the computation it is found that the optimization runs are quite sensitive to the initial μ (parameter in the l_1 merit function) and the maximum allowable step length. The initial μ should be small enough; a large maximum allowable step length induced iterates out of the bound.

Since the benefit from not solving the underlying flow equation could be huge, the algorithm can be applied to other design or control problem in which solving a underlying equation is very time consuming.

The super linear convergence rate is expected to be observed in smooth problems and its convergence proof would be future work. Also improved globalization (for example the trust region approach) should help increase the robustness of the algorithm.

REFERENCES

[1] Paul D. Frank and Gregory R. Shubin, *"A Comparison of optimization based Approaches for a Model Computational Aerodynamics Design Problem,"* Journal of Computational Physics, vol 98, 1992, pp. 74-89.

[2] M.J.D. Powell, (1978) The convergence of variabe metric methods for nonlinearly constrained optimization calculations, in *Nonlinear Programming 3*(Eds O.L.Mangasarian, R.R. Meyer and S.M.Robinson), Academic Press, New York.

[3] R. Fletcher 1987 *Practical Methods of Optimization (2nd Ed.)* by John Wiley & Sons.

[4] J. Nocedal and M.L. Overton, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM Journal on Numerical Analysis 22 (1985) pp. 821-850.

[5] Richard H. Byrd and Jorge Nocedal,*An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49, pp. 285-323, 1991.

[6] Richard H. Byrd and Jorge Nocedal, *A Tool for the Analysis of Quasi-Newton Methods with Application to Unconstrained Minimization*, SIAM Journal on Numerical Analysis Vol. 26, No. 3, pp. 727-739, 1989.

[7] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[8] T.F. Coleman and A.R. Conn, *On the local convergence of a quasi-Newton method for the nonlinear programming problem*, SIAM Journal on Numerical Analysis 21 (1984) pp. 755-769

[9] Gary A. Sod, *Numerical Methods in Fluid Dynamics*, Cambridge University Press 1985.

[10] Philip E. Gill, Walter Murray, and Margaret H. Wright, *Practical Optimization*, Academic Press 1981.

[11] F.-S. Kupfer and E.W. Sachs, *Numerical Solution of a Nonlinear Parabolic Control Problem by a Reduced SQP Method*, Computational Optimization and Applications, 1,(1992), pp. 113-135.

[12] Jens Lorenz, *Convergence of Upwind Schemes for a Stationary Shock*, Mathematics of Computation v. 46, No 173, 1986, pp. 45-57.

[13] A.B. Stephens and G.R. Shubin, *Existence and Uniqueness For an Exponentially Derived Switching Scheme*, SIAM J. Numer. Anal. vol 20, No 5, 1983.

[14] Pedro Embid, Jonathan Goodman,and Andrew Majda, *Multiple Steady States for 1-D Transonic flow*, SIAM J.Sci. Stat. Comput. Vol 5, No.1, 1984, pp. 21-41.

[15] Günther Hämmerlin and Karl-Heinz Hoffmann, *Numerical Mathematics*, Springer-Verlag 1991.

[16] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press 1972.

[17] Thomas F. Coleman and Danny C. Sorensen, *A Note on The Computation of an Orthonormal Basis for The Null Space fo a Matrix*, Mathematical Programming 29 (1984) pp. 234-242.

[18] G.W. Stewart and Margaret H. Wright, *Properties of a Representation of Basis for The Null Space*, Mathematical Programming 33 (1985) pp. 172-186.

[19] Richard H. Byrd and Robert B. Schnabel, *Continuity of The Null Space Basis and Constrained Optimization*, Mathematical Programming 35 (1986) pp. 32-41.

[20] R. Narducci, B. Grossman, Raphael T. Haftka, *Design Sensitivity Algorithms for an Inverse Design Problem Involving a Shock Wave* Virginia Polytechnic Institute and State University Dept. Aerospace and Ocean Engineering.

[21] J.E. Dennis, Jr. and Jorge J. Moré, *A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods*, Mathematics of Computation, Vol 28, No 126, 1974, pp. 549-560.

[22] K. Kunisch and E. W. Sachs, *Reduced SQP Methods for Parameter Identification Problems*, SIAM J. Numer. Anal. Vol. 29, No 6, pp. 1793-1820, December 1992.

[23] John D. Anderson, Jr., *Modern Compressible Flow with Historical Perspective*, McGraw-Hill, 1982.

[24] Michael Renardy and Robert C. Rogers, *An introduction to Partial Differential Equation*, Springer-Verlag 1992.

VITA

Kyehong Kang was born in Seoul, Korea on September 2, 1961. He received the B.S. degree in Mathematics from Seoul National University in 1985, the M.S. degree in Mathematics from Virginia Polytechnic Institute and State University in 1988. He held teaching assistant positions and enjoyed working with students. He is a member of American Mathematical Society and Society for Industrial and Applied Mathematics.

Kyehong Kang