DESIGN OF OPTIMUM DUCTS USING AN EFFICIENT 3-D VISCOUS COMPUTATIONAL FLOW ANALYSIS

Ravi K. Madabhushi
Ralph Levy
Steven M. Pincus*

Scientific Research Associates
P.O. Box 1058
Glastonbury, CT 06033

ABSTRACT

Design of fluid dynamically efficient ducts is addressed through the combination of an optimization analysis with a three-dimensional viscous fluid dynamic analysis code. For efficiency, a parabolic fluid dynamic analysis was used. Since each function evaluation in an optimization analysis is a full three-dimensional viscous flow analysis requiring 200,000 grid points, it is important to use both an efficient fluid dynamic analysis and an efficient optimization technique. Three optimization techniques are evaluated on a series of test functions. The Quasi-Newton (BFGS, $\eta = .9$) technique was selected as the preferred technique. A series of basic duct design problems are performed. On a two-parameter optimization problem, the BFGS technique is demonstrated to require half as many function evaluations as a steepest descent technique.

INTRODUCTION

The subject study combines an existing, well-proven, three-dimensional, viscous duct flow analysis with a formal optimization procedure to design a duct that is optimum for a given application. The fluid dynamics code used in this effort is a three-dimensional, viscous flow, forward marching calculation developed by Scientific Research Associates under ONR and NASA Lewis Research Center Contract support. This analysis solves a set of fluid flow equations

---

*Present address: AVCO, Stratford, CT

that are parabolic in the predominant flow direction. Physical approximations are made to the Navier-Stokes equations resulting in an analysis that is much faster to use by a factor of 10 to 100 than the full Navier-Stokes equations for flows which have an a priori specifiable predominant flow direction, such as duct flows. The capability of this analysis has been clearly demonstrated by Levy, Briley and McDonald (Ref. 1) and by Towne (Ref. 2). These references document application of this analysis to a wide variety of viscous flow duct problems with configurations such as 90° bends, 180° bends, 'S' bends, circular to square cross-section, etc. In all cases, comparison with existing high quality experimental data was shown to be very good.

In its original form, this parabolic flow analysis provides a powerful tool for analyzing three-dimensional, viscous flow in ducts. However, the engineering design problem is the design of ducts for a specific application subject to specified constraints. The subject study initiated a program to combine the viscous flow analysis with an optimization procedure to design such ducts.

Development of this design procedure will give the aerospace and hydrodynamic design and development community a powerful new tool to aid in the design of fluid dynamic components. Such a tool would automate the duct design procedure. Currently, ducts are designed through a series of calculations and tests which are performed until a duct is obtained which meets certain design criteria. The current process does not ensure an orderly progression toward a best design and indeed does not even ensure the final design is optimum in any sense. The subject study could provide a tool of validated accuracy that would quickly and accurately design a duct within engineering (user-specified) constraints that is the optimum for the specific application.

Since each function evaluation in an optimization process is a three-dimensional viscous flow clculation requiring 200,000 grid points, it is important to use efficient optimization technqiues. Three optimization techniques are evaluated on a series of ten test functions to determine an optimization technique judged best for the type of probelms encountered in duct flow analysis. These are presented in the following section.

OPTIMIZATION

Modelling the performance of a physical system by an objective function means associating a real number with each set of system independent variables. This objective function gives a single measure of "goodness" of the system for the given values of the variables. The task of optimizing is finding the value and location of the minimum of the objective function. The discussion of optimization techniques which follows is practical for problems with a small number of variables (less than 200).

For the subject application, the objective function is highly non-linear and requires solving a coupled system of partial differential equations to evaluate the function value at a given point (set of independent variable values). Objective function evaluations can be very expensive compared to any other calculations needed for each of the optimization methods considered. One optimization method will be judged better than another if it consistently needs fewer objective function evaluations to achieve a desired level of accuracy, since it would cost less to use the better optimization to achieve that accuracy. The general form of an optimization problem is

$$\begin{aligned} &\underset{x \varepsilon R^n}{\text{minimize}} \quad F(x) \\ &\text{subject to } c_i(x)=0, \quad i = 1,2,\ldots,s; \\ &\qquad\qquad\quad c_j(x)>0, \quad j = s+1,\ldots,m. \end{aligned}$$

The objective function is $F(x)$, and the constraints are given by the c's where x is a vector of independent variables of the system.

Search Direction

Several types of iterative search methods were investigated. They all have the following form in common. At the start of the k-th iterations, x_k is the current estimate of the minimum. The k-th iteration then consists of the computation of a search vector, p_k, from which the new estimate, x_{k+1}, is found from:

$$x_{k+1} = x_k + \alpha_k p_k \qquad (1)$$

where α_k is obtained by one-dimensional search. The methods studied ultimately stem from the central notion from calculus of the descent direction. The calculus theorem states that at a given point the local direction of steepest descent is in the direction that is opposite to the gradient. In Eq. (1) one could compute the gradient at x_k, denoted as g_k, and let $p_k = -g_k$; then for suitably small α_k, some function reduction would be guaranteed; that is, $F_{k+1} < F_k$, where $F_k = F(x_k)$. This choice of p_k defines GRAD, the gradient method, also known as the steepest descent method.

In these methods, one must estimate the gradient at each starting point of a one-dimensional search. Using the first order approximation in a Taylor expansion about the starting point, this involves 'n' function evaluations per linear search, where n is the number of independent variables under consideration. To include second order behavior, one can choose either to directly compute the Hessian matrix of second derivatives by finite differences, or to somehow approximate it from previous information. Direct evaluation of the Hessian matrix at each one-dimensional starting point would cost many additional function evaluations, and has long since been rejected by practitioners. A conjugate gradient method (CG) is designed as a crude but easily applied approximate quadratic search update. The Quasi-Newton methods approximate the Hessian matrix by a positive definite matrix for each one-dimensional search.

In the CG method, the gradient vector is computed for each one-dimensional search, k, and the search direction, p_k, is computed by:

$$p_0 = -g_0$$

$$p_k = -g_k + \beta_k p_{k-1}$$

A common choice for β_k is:

$$\beta_k = (g_k^T g_k) / (g_{k-1}^T g_{k-1}) \quad (2a)$$

or equivalently

$$\beta_k = (||g_k|| / ||g_{k-1}||)^2 \quad (2b)$$

where g^T refers to the transpose of the vector g. This algorithm performs quite a bit better than does GRAD, but it can still be substantially improved upon by the Quasi-Newton methods. There are two reasons for this. First, functions with strong coupling between independent variables typically require many CG direction updates to successfully improve the successive approximations to the optimum. Second, each of the two gradient methods presented above require accurate one-dimensional searches to give good performance.

Quasi-Newton optimization methods are based on Newton's method, which is designed to achieve termination (minimum) in a single iteration for a quadratic function, given the first and second derivatives of the function. Consider an arbitrary point, x_k, and a vector, p_k, for a quadratic function. The Taylor series gives the expansion of the gradient at $x_{k+1} = x_k + p_k$ as $g_{k+1} = g(x_k+p_k) = g_k + Gp_k$, since higher-order derivative terms are 0. If x_{k+1} is to be the minimum of the function, then g_{k+1} must be 0 and then p_k is given by

$$p_k = -G^{-1}g_k$$

For non-quadratic functions, one iteration termination will not be achieved, but a method which accurately estimates G^{-1} should provide an effective scheme for successive search directions. This logic formed the intuition behind the Quasi-Newton methods, which were developed in the 1960's and early 1970's. Until then, the CG method was the most commonly used search update procedure. In the early 1960's, Davidon, and then Fletcher and Powell independently came up with a Quasi-Newton method in which, at the k-th iteration,

$$p_k = -H_k g_k$$

where H_k is approximated from gradient information. This updating gave substantial improvement over the gradient methods in that it required many fewer function evaluations to achieve a given degree of accuracy. In 1970, several additional authors independently came out out with an improved updating formula for H_k. The authors were Broyden, Fletcher, Goldfarb and Shanno, and their formula is known as the BFGS formula.

$$H_{k+1} = \left(I - \frac{\Delta x_k \Delta g_k^T}{\Delta x_k^T \Delta g_k}\right) H_k \left(I - \frac{\Delta x_k \Delta g_k^T}{\Delta x_k^T \Delta g_k}\right)^T + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} \qquad (3)$$

One-Dimensional Search

The objective of the one-dimensional search algorithm is to minimize the objective function evaluated along the search direction. Assuming that the function evaluated along this direction is unimodal, three different methods of performing one-dimensional search were studied: Golden Section, Safeguarded Quadratic, and Brent's hybrid method. A detailed description of the algorithms is not presented here because it turned out that the choice of one-dimensional algorithm made very little difference in the determination of a best optimization method.

One-Dimensional Termination Criteria

The termination requirement for stopping a given one-dimensional search is that two reduction criteria are met: a reduction of the objective function and a specified reduction in the directed gradient

$$F(x_k+\alpha p_k) - F(x_k) < \mu\alpha p_k^T g(x_k) \text{ , for } 0<\mu<1/2 \qquad (4)$$

$$\left|p_k^T g(x_k+\alpha p_k)\right| < -\eta p_k^T g(x_k) \text{ , where } 0<\eta<1 \qquad (5)$$

The choice of η in Eq. (5) is the parameter that determines the crudeness of the one-dimensional search; η near 1 gives a crude search, η near 0 gives a very accurate one. The tradeoffs are that a crude search requires fewer function evaluations per linear search, but more directional searches, with the more accurate searches having the opposite behavior.

Three values of η were chosen in making comparisons, $\eta = .9$, .05, and .001. These values for η represent coarse, fairly accurate, and very accurate one-dimensional searches. As noted in the literature, and as reinforced in Table 1, the gradient methods require a very small η, typically .001 or smaller, to be effective. A value of $\mu = .0001$ was maintained for Eq. (4). An exact linear search results when it is required that $\left|p_k^T g(x_k+\alpha p_k)\right| = 0$, i.e., that $\eta = 0$. Virtually all of the theoretical investigations of performance of search direction algorithms (BFGS, CG, etc.) require that exact linear searches be performed. However, as shown in the next section, a very small value of η does not give excellent algorithm efficiency in terms of function evaluations.

Initial Point in One-Dimensional Search

In analyzing the performance of alternative algorithms for various test functions, the choice of starting point for each one-dimensional search was crucial. It was found that the consensus in the literature for an initial choice of α_k in Eq. (1) was $\alpha_k = 1$. This choice works exactly for quadratic functions when exact gradient and Hessian information is known for BFGS. In fact, for BFGS with a course search ($\eta = .9$). this initial point is usually good that for virtually all of the test functions, after the first few direction searches, most one-dimensional searches are terminated after either one or two function evaluations. $\alpha_k = 1$ for all k was chosen as the starting point for all algorithms that are compared below.

Test Functions

Each of the candidate algorithms was tested for a variety of functions. Five of the functions were taken from the standard literature (Refs. 3-4). These were functions that many authors used to draw conclusions about the effectiveness of various algorithms. Each of the functions is constructed with certain properties that might be difficult for some optimization methods to handle, such as a steep valley, or a near stationary point. The other five test functions were chosen by us and include a representative range of multivariate polynomials which might be more typical of the type of behavior that would be encountered in practice. For each function, the formula, a brief description, the starting point, and the location and value of the minimum are given.

Rosenbrock

$$F(x_1,x_2) = 100(x_2-x_1^2)^2 + (1-x_1)^2 ,$$

starting at (-1.2,1.0).

This is the most commonly referenced test function; it was first posed in 1960. It has a steep, banana-shaped valley, with minimum value of 0 at (1,1).

Helix

$$F(x_1,x_2,x_3) = 100((x_3-10\theta)^2 + (r-1)^2) + x_3^2 ,$$

where

$$r = (x_1^2+x_2^2)^{1/2} ,$$

and

$$2\pi\theta = \begin{cases} \arctan(x_2/x_1) , & x_1 > 0, \\ \pi + \arctan(x_2/x_1) , & x_1 < 0, \end{cases}$$

This function has a helical valley with 0 minimum at (1,0,0). It was first posed by Fletcher and Powell, two of the leading researchers in the field, in 1963.

Woods

$$F(x_1,x_2,x_3,x_4) = 100(x_2-x_1^2)^2 + (1-x_1)^2 + 90(x_4-x_3^2)^2 + (1-x_3)^2$$

$$+ 10.1\{(x_2 -1)^2 + (x_4 -1)^2\} + 19.8(x_2 -1)\ (x_2-1)$$

starting at (-3,-1,-3,-1).

Again, this function has a minimum value of 0 at (1,1,1,1). The function has a near stationary point when $F(x) \approx 7.82$. Performance of algorithms on this function can vary widely due to some x_k becoming close to the near-stationary point. Furthermore, the non-unimodality safeguard is extremely important here as in Rosenbrock, since without this, many methods take alternative paths to the solution because of picking the "wrong valley". This function was posed by Colville in 1968.

Singular

$$F(x_1,x_2,x_3,x_4) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 +10(x_1-x_4)^4 ,$$

starting at (3, -1,0,1).

The Hessian matrix at the solution $x^* = (0,0,0,0)$ has two zero eigenvalues. The minimum value is again 0, but the function is slowly varying in the neighborhood of the solution. This function was posed by Powell in 1962.

Biggs

$$F(x_1,x_2,x_3,x_4,x_5) = \sum_{i=1}^{11} \left((x_3 e^{-x_1 z_i} - x_3 e^{-x_2 z_i} + 3e^{-x_5 z_i}) - y_i\right)^2 ,$$

where $y_i = e^{-z_i} - 5e^{-10 z_i} + 3e^{-4 z_i}$,

and $z_i = (0.1)i$, $i=1,2,\ldots 11$,

starting at (1,2,1,1,1), with initial function value 13.39.

This function has a global minimum with value 0 at (1,10,1,5,4), together with a local minimum near (1.78,16.12,-0.59,4.71,1.78), with function value ≈.00265. It was posed by Biggs in 1972.

New Polynomial #1

$$F(x_1,x_2,x_3) = x_1^4 + x_2^4 + x_3^4,$$

starting at (0.4,0.6,0.8).

This function has minimum value of 0 at (0,0,0).

New Polynomial #2

$$F(x_1,x_2,x_3) = x_1^4 + x_2^4 + x_3^4 + 3x_1x_2^2 + x_2^3 + x_1x_3 ,$$

starting at (-1,-2,-3).

This function has a global minimum of -5.8979 at (-1.4134,-1.8786,.7070).

New Polynomial #3

$$F(x_1,x_2,x_3) = (x_1-x_2)^4 + (x_2-x_3)^4 + x_1^2 ,$$

starting at (-1,-2,-3).

This function has a global minimum of 0 at (0,0,0).

New Polynomial #4

$$F(x_1,x_2,x_3,x_4) = (x_1-x_2-x_3-x_4)^4 + (2x_1+4x_3)^4 + (x_2-x_3)^4 + (x_4-x_1)^2 ,$$

starting at (-1,-2,-3,-4).

This function has a global minimum of 0 at (0,0,0,0).

New Polynomial #5

$$F(x_1,x_2,x_3,x_4) = (x_1-x_2-x_3-x_4)^4 + (2x_1+4x_3)^4 + 20\ (x_2-x_3)^4 + (x_4-x_1)^2 ,$$

starting at (-1,-2,-3,-4).

This function has a global minimum of 0 at (0,0,0,0). It differs from the previous function in that the third term has a coefficient much larger than unity; in engineering applications, it is often the case that the objective function is much more sensitive to one variable than to the others.

Results of Comparisons

Each of the test functions was optimized with a gradient (GRAD), conjugate gradient (CG) and a Quasi-Newton technique (BFGS). Comparisons were made for each test function and choice of algorithm based on the number of function evaluations needed to approximate the true function minimum to within .0001. Table 1 presents the measure of algorithm effectiveness for each of the ten functions and for the nine algorithms that use Brent's one-dimensional search technique. Three possible search direction updates are considered, BFGS, CG, and GRAD, and for each of those, three values for the gradient reduction parameter, $\eta = 0.9$, 0.05, and 0.001, as the termination criterion for the one-dimensional searches.

Table 1

ALGORITHM COMPARISON - NUMBER OF FUNCTION EVALUATIONS

FUNCTION	BFGS			CG			GRAD		
	η=0.9	η=0.05	η=0.001	η=0.9	η=0.05	η=0.001	η=0.9	η=0.05	η=0.001
ROSENBROCK	85	153	179	>>600	>600	>>700	>>550	>>700	>>800
HELIX	105	161	200	>>650	>800	425	>>600	>>800	>>800
WOODS	110	152	>300	>>800	>>800	>>800	>>750	>>850	>>950
SINGULAR	121	131	152	460	690	610	>>600	>>800	>>800
BIGGS	86	105	132	>>600	>>1000	>>800	>>500	>>600	>>700
POLYNOMIAL #1	22	17	23	28	20	22	28	14	21
POLYNOMIAL #2	71	100	102	200	220	125	300	310	135
POLYNOMIAL #3	41	70	92	62	90	77	>250	420	525
POLYNOMIAL #4	198	210	132	220	153	250	>>500	>>650	>>800
POLYNOMIAL #5	175	212	210	238	227	275	>>550	>>650	>>750

For some of the table entries, numbers are preceded by either > or >>. In both those cases, after 60 directional searches, the objective function was not yet within .0001 of the minimum. If it were within .001 of the minimum, the notation '>' is used with the nearest rounded-off number of total function evaluations after the 60 searches. If the function remained greater than .001 away from the minimum after the 60 searches, the result in exactly the same manner, the symbol '>>' was used to indicate that it is expected to take many more evaluations to reach the stopping criterion.

The BFGS algorithm with $\eta = 0.9$ clearly gives the best algorithm performance, sometimes by an overwhelming amount when compared to gradient methods. First, note that for eight of the ten test functions, BFGS with $\eta = 0.9$ gives the best performance, and in the other two cases (Polynomials #1 and #4), it is not too far from the optimum. Second, note that for each of the first five test functions, only one of the gradient algorithms requires fewer than six times the number of function evaluations required by BFGS with $\eta = 0.9$ to achieve the desired measure of effectiveness. Even in that case (Singular function), nearly four times as many evaluations are needed by the best gradient technique to achieve the stated accuracy as are needed for the BFGS algorithm with $\eta = 0.9$. Furthermore, in general, the BFGS algorithm with $\eta = 0.9$ performs better than the BFGS algorithm with $\eta = 0.05$, which in turn performs better than the BFGS algorithm with $\eta = 0.001$.

These conclusions are quantified in several different ways. First, the total number of function evaluations needed to reach the .0001 accuracy over all ten functions for each algorithm are averaged. For the ten functions, the BFGS, $\eta = 0.9$ case averages 101.4 evaluations, the BFGS, $\eta = 0.05$ case averages 131.1 evaluations, and the BFGS, $\eta = 0.001$ case averages 152.2 evaluations, with none of the gradient method algorithms averaging less than 380 evaluations. These sums are conservatively estimated for those methods with > or >> as the table entries by assuming that in those cases, the number of evaluations needed was * in the entry >>* or >*.

The first five test functions are the dominant effect in this disparity, since none of the gradient methods perform adequately for any of these functions. Since these five functions were constructed by researchers in the field, they may represent function behavior that demonstrated the shortcomings of a particular method and that occurred infrequently. With that in mind, the same comparisons were performed for the five polynomial functions that were constructed. The idea here was that a typical function that occurs in practice lies somewhere "between" the first five test functions and the second five functions, in terms of difficulty for the gradient methods to handle efficiently.

For the five polynomials, the GRAD directional search technique still performed poorly, especially for polynomials 4 and 5. For these polynomials, the BFGS, $\eta = 0.9$ case averages 101.4 evaluations, the BFGS, $\eta = 0.05$ case averages 121.8 evaluations, and the BFGS, $\eta = 0.001$ case averages 111.4 evaluations, while the CG, $\eta = 0.9$ case averages 149.6 evaluations, the CG, $\eta = 0.05$ case averages 142.0 evaluations, and the CG, $\eta = 0.001$ case averages 149.8 evaluations. Note that the BFGS, $\eta = 0.05$ case had a higher average than the BFGS, $\eta = 0.001$ case for these polynomials was the result of the calculation for polynomial #4. This is atypical, for among the ten test functions, the BFGS, $\eta = 0.05$ case gave superior performance to the BFGS, $\eta = 0.001$ case for the first eight functions, with the last function giving virtually identical performance for the two algorithms.

The BFGS, $\eta = 0.9$ case is clearly the most efficient of the algorithms. In turn, the BFGS, $\eta = 0.05$ algorithm is the next most efficient, followed by the BFGS, $\eta = 0.001$ case. The BFGS algorithms perform qualitatively better than do the CG algorithms, which in turn perform qualitatively better than do the GRAD algorithms. Stated differently, in all cases exhibited among the test functions, the BFGS algorithms reached the .0001 approximation of the exact minimum with reasonable efficiency. For some of those functions, both gradient methods (CG and GRAD) were very inefficient (costly) in approximating the minimum, while for another subset of the test functions, CG was reasonably efficient at approximating the minimum while GRAD was not. It was never the case that GRAD was efficient when CG was not, or that either gradient algorithm was efficient when BFGS was not. The BFGS, $\eta = 0.9$ case always worked with at least a modest amount of improvement in efficiency over the alternative methods, and sometimes the BFGS search methods gave dramatic reductions in the number of evaluations needed to achieve the desired accuracy over the gradient techniques. This method requires only function values at any point and estimates approximate derivatives and Hessian matrices from function evaluations.

PEPSIG OPTIMIZATIONS

In the current investigation, use of the PEPSIG analysis in conjunction with formal optimization procedures to find optimum duct shapes is demonstrated for realistic problems. The PEPSIG code is used to solve for the flow field in the duct. The drop in mass average total (stagnation) pressure is used as the performance index (objective function) for optimization. Each case was run using approximately 200,000 grid points to provide the required accuracy. The test case considered in this investigation involves optimizing a duct having an offset bend for minimum pressure drop. The viscous flow calculations were performed at a Reynolds number of 50,000 based on entrance mean velocity and at a Mach number of 0.3. The mass flow through the duct is held constant and the pressure drop is non-dimensionalized with ρu^2.

Geometry of the Duct

The geometry of the duct to be optimized is characterized by the shape of the centerline and the variation of the tube radius along the length of the duct. The shape of the centerline depends on the offset, H, and the bend length, L, over which the offset takes place. The centerline, as shown in Fig. 1, has an upstream straight section (1)-(2), an S-shaped bend section

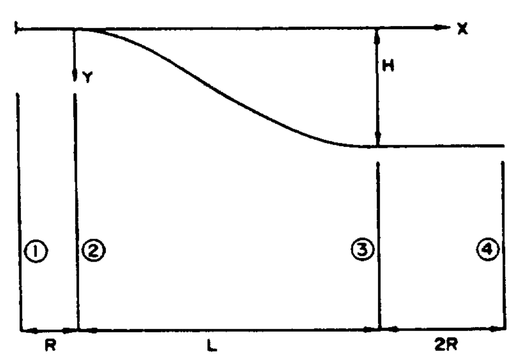

Fig. 1 - Shape of the Centerline

(2)-(3) and a downstream straight section (3)-(4). In the figure, R is the uniform radius of the tube upstream and downstream of the bend and the computational domain extends over all three sections. The portion of the centerline in the bend, i.e., section (2)-(3) is represented by a fifth order polynomial with vanishing first and second derivatives at the end points and is given as:

$$\frac{Y}{H} = 10\left(\frac{X}{L}\right)^3 - 15\left(\frac{X}{L}\right)^4 + 6\left(\frac{X}{L}\right)^5 \qquad (6)$$

One-parameter and two-parameter optimizations were performed in this study.

One-Parameter Optimizations:

A series of one-parameter optimizations were performed to optimize independently the parameters characterizing the duct geometry. First the bend length, L, was optimized for a uniform duct of offset, H = 1 and radius, R = 1. If the bend length, L, is too large, there will be a large pressure drop due to frictional losses. If the length is too small, there will be stronger curvatures producing large pressure losses due to flow turning and separation. Thus, there should exist an optimum bend length, L, that would strike a balance between the two kinds of losses. This optimum value was found to be at $L_{opt} = 3.875 \pm 0.015$ and the corresponding pressure drop was $\Delta(\bar{P}_T) = 0.0336$. This will be used as a reference duct for comparison in the remaining optimizations.

It was seen for this optimum duct that there are significant regions of flow separation due to the adverse pressure gradient resulting from the curvature change about the inflexion point of the bend. If the separation region is either suppressed or at least reduced to a smaller region, the losses associated with separation may be reduced and an improvement in duct performance may be obtained. One way to alleviate the adverse pressure gradient is to contract the cross-sectional area of the duct in the region of flow separation. A contraction will accelerate the flow, thus helping to reduce separation. On the other hand, too much contraction may worsen the performance because more reduction in area would mean higher velocities near the minimum area, which would lead to higher friction losses. Also, since the duct exit radius is held equal to the duct inlet radius, the divergence in the duct following the contraction may lead to more flow separation. Thus, an optimum contraction may exist and a one-parameter optimization was performed to determine the optimum contraction for a fixed bend length, L.

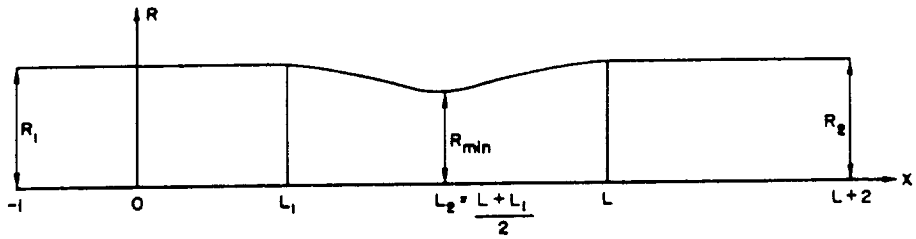

Fig. 2 - Variation of Duct Radius Along the Length of the Duct

The changing radius of the duct is shown in Fig. 2 and is represented by a sixth order polynomial as follows. R_{min} is the minimum radius of the duct.

$$-1 \leq X \leq L_1; \quad R = R_1 \tag{7}$$

$$L_1 \leq X \leq L; \quad R = A + BX + CX^2 + DX^3 + EX^4 + FX^5 + GX^6 \tag{8}$$

$$L \leq X \leq L + 2, \quad R = R_2 \tag{9}$$

The boundary conditions are:

$$@ \ X = L_1: \quad R = R_1, \ \frac{dR}{dX} = \frac{d^2R}{dX^2} = 0 \tag{10}$$

$$@ \ X = L_2 = \frac{L + L_1}{2} \ ; \ R = R_{min} \tag{11}$$

$$@ \ X = L: \quad R = R_2, \ \frac{dR}{dX} = \frac{d^2R}{dX^2} = 0 \tag{12}$$

The seven coefficients A to G in Eq. (8) are evaluated using the boundary conditions as given by Eqs. (10) through (12). Fig. 3 shows the geometry of the duct for this and subsequent optimizations. For $L = 3.875$, $R_1 = R_2 = 1.0$, $L_1 = 0.32L$, $L_2 = 0.66L$ and $H = 1.0$, Fig. 4 shows the variation of the pressure drop with R_{min} and the optimum R_{min} was found to be 0.97 ± 0.01 with a pressure drop, $\Delta(\bar{P}_T)$ of 0.0322. For a reduction in the area of about 6% corresponding to $(R_{min})_{opt} = 0.97$, the pressure drop is reduced by about

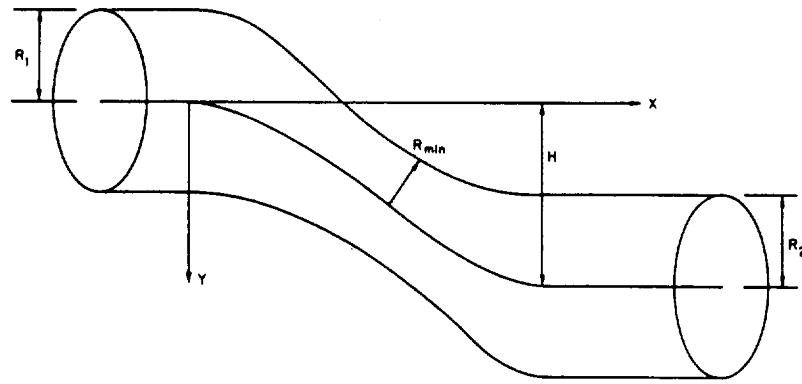

Fig. 3 - Geometry of the Duct for One and Two-Parameter Optimizations

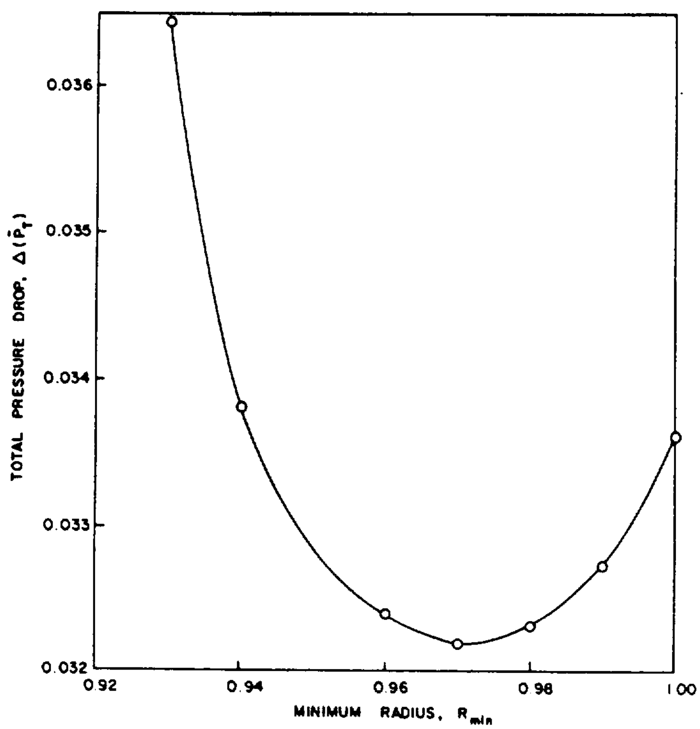

Fig. 4 - Efect of Contraction on Total Pressure Drop for a Fixed Bend Length.

4.2% with respect to the reference duct of uniform cross-section and the flow separation is reduced to a smaller region, thus improving the fluid dynamic performance of the duct.

The location in the duct where the contraction starts and where the maximum contraction lies may each be critical to the performance. If the contraction starts too early, then most of the contraction would be in the favorable pressure gradient region and may not help much with flow separation.

If the contraction starts too far in the duct, it won't be effective in improving the performance. In the next optimization, the location of the start of the contraction, L_1, is optimized for $R_{min} = 0.97$, $L = 3.875$ and $L_2 = \frac{L + L_1}{2}$. ($L_2 = \frac{L + L_1}{2}$ implies that contraction is symmetric about L_2 and ends at the end of the bend.) Fig. 5 shows the effect of L_1 on the duct performance and optimum L_1 was found to be 0.24L ± 0.04 L with a pressure drop, $\Delta(\bar{P}_T)$, of 0.0320. This $\Delta(\bar{P}_T)$ is about 0.62% less than the $\Delta(\bar{P}_T)$ of the previous optimization (corresponding to $L_1 = 0.32L$) and the overall reduction in the pressure drop with respect to the reference case is about 4.8%.

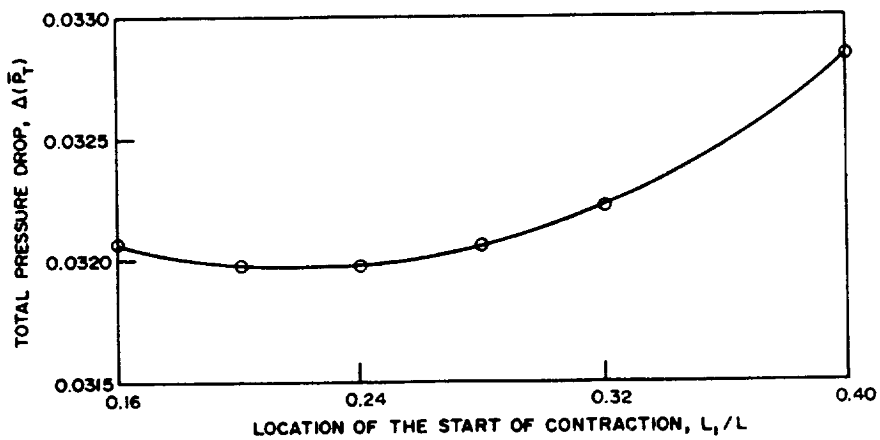

Fig. 5 - Effect of Location of the Start of Contraction, L_1, on Total Pressure Drop.

Two-Parameter Optimizations:

In the above one-parameter optimizations, the bend length, L, the contraction, R_{min}, and the start of the contraction, L_1, were each optimized independently for minimum pressure drop for a duct of fixed offset. However, these parameters interact with each other and the optimization should be performed in a way to account for this interaction. In the following examples, two-parameter optimizations are performed where the two parameters are varied simultaneously for an optimum combination. L and R_{min} are the two parameters to be optimized for a fixed L_1. It was stated in the optimization section of this report that the BFGS method is generally more efficient than the gradient method for the examples considered. We will now demonstrate that the same is true for realistic design problems such as a duct having an offset bend.

For a fixed offset, $H = 1.0$, $R_1 = R_2 = 1.0$, $L_1 = 0.24L$, $L_2 = 0.62L$, optimum L and R_{min} are found using the gradient (steepest descent) and BFGS methods. With the gradient method, three one-dimensional searchs were performed and at the end of the third search the optimum values for L and R_{min} were found to be 3.719 and 0.954, respectively. The pressure drop, $\Delta(\bar{P}_T)$ is 0.0305 and the cumulative number of function evaluations for this method is 40. Using the BFGS method, two one-dimensional searches were performed for a comparable pressure drop as in the gradient method. The change in the search direction is shown in Fig. 6. At the end of the second search using the BFGS method, the pressure drop, $\Delta(\bar{P}_T)$, is 0.0306 and the cumulative number of function evaluations is seventeen. The optimum values for L and R_{min} are 3.559 and 0.958, respectively. Thus, for a comparable reduction in the pressure drop, the BFGS method needed only approximately half the number of function evaluations as that for the gradient method, reasserting that the BFGS method is a more efficient method.

At the end of the second search of the BFGS optimization, the overall reduction in the pressure drop compared to the reference case is about 9% and the length of the duct was reduced by about 8%. The reduction in pressure loss is an improvement in duct performance. Decreases in length are usually associated with reductions in weight and improved ease in packaging.

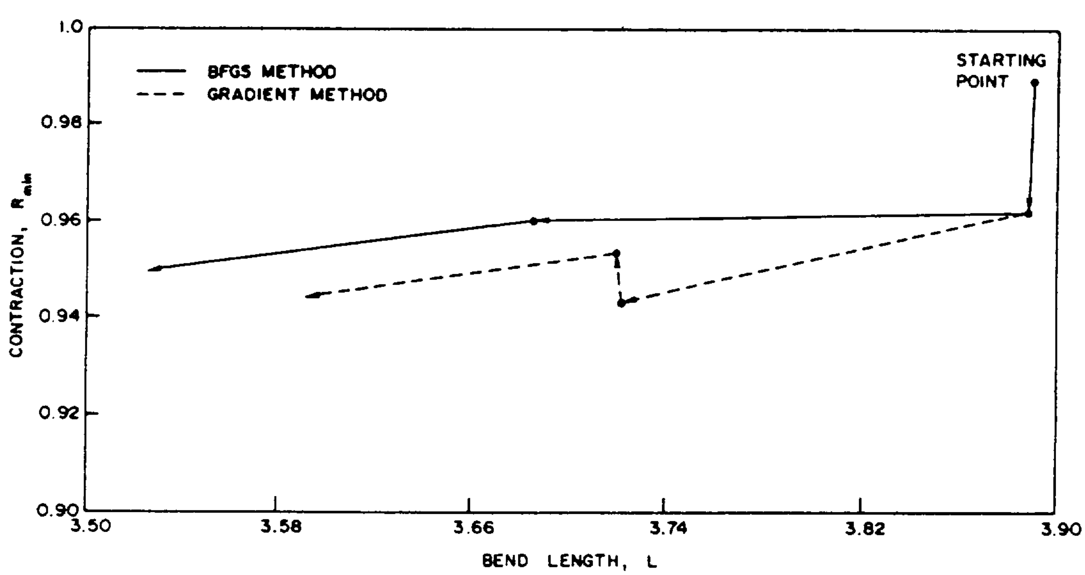

Fig. 6 - Search Directions for BFGS and Gradient Methods of Optimization

CONCLUSIONS

Design of fluid dynamically efficient ducts is addressed through the combination of optimization analysis with a three-dimensional viscous fluid dynamic analysis code. Since each function evaluation in the optimization analysis is a three-dimensional viscous flow analysis, which requires 200,000 grid points for accuracy, it is important to use both an efficient fluid dynamic analysis and an efficient optimization technique. Three optimization techniques were evaluated on a series of test functions. The Quasi-Newton (BFGS, $n = .9$) technique was selected as the preferred technique. A series of basic duct design problems was performed. On a two-parameter optimization problem the BFGS technique was demonstrated to require half as many function evaluations as a steepest descent technique. Use of this parabolic flow analysis rather than a full Navier-Stokes analysis in an optimization scheme can provide huge computer run time and cost savings. Optimization of realistic aerodynamic and hydrodynamic ducts can now be made a routine part of the design process for approximately the computer time required for a single function evaluation using an efficient three-dimensional Navier-Stokes analysis.

ACKNOWLEDGEMENT

This work was sponsored by NASA Lewis Research Center, Cleveland, Ohio.

REFERENCES

1. Levy, R., Briley, W.R. and McDonald, H.: Viscous Primary/Secondary Flow Analysis for Use with Non-Orthogonal Coordinate Systems, AIAA Paper 83-0556, 1983.

2. Towne, C.E.: Computation of Viscous Flow in Curved Ducts and Comparison with Experimental Data, AIAA Paper 84-0531, 1984.

3. Scales, L.E.: Introduction to Non-Linear Optimization. Springer-Verlag, New York, 1985.

4. Gill, P.E., Murray, W. and Wright, M.H.: Practical Optimization. Academic Press, London, 1981.