A Computational Design Method for High-Velocity Channels

by

C. O. E. Burg[1]
D. H. Huddleston[2]
R. C. Berger[3]

March 1, 1998

Abstract

With the growth of urban centers in recent years and the resulting changes in infrastructure, the capacity of many existing storm water management systems has been stretched to its limit. Altered surface hydrology increases surface runoff which reaches drainage channels more quickly, resulting in greater system loads. To handle this increased flow, many of these channels need to be redesigned. Complicating this process are the restrictions on channel designs due to other structures, such as roads, bridges and multi-story buildings, because alterations to these structures can be quite expensive. Designing high-velocity channels to handle the increased water flow while minimizing the alterations to existing structures is a formidable task, with the current design technique being a trial and error approach.

In this paper, we present an efficient, deterministic design method, which applies the adjoint variable formulation of direct differentiation to a computational, open-channel flow model in order to obtain the derivative of an objective function with respect to the design variables. From these derivatives, we modify the design variables with the goal of minimizing the objective function. The particular CFD code (HIVEL2D) uses an unstructured, Petrov-Galerkin, finite element method to solve the unsteady, two-dimensional, depth-averaged, shallow water equations. The test cases involve channel contraction problems with one, two and three design variables, where uniform downstream flow is the goal. For these cases, the iterative design process produces channels that yield about a 90% improvement over straight wall contractions.

[1]Graduate Student, Computational Engineering, Mississippi State University.
[2]Associate Professor, Civil Engineering, Mississippi State University, Member ASCE, Member AIAA.
[3]Research Hydraulic Engineer, USAE Waterways Experiment Station, Coastal and Hydraulics Laboratory, Vicksburg, Mississippi.

1 Introduction

Flows through high-velocity channels produce hydraulic phenomona, such as hydraulic jumps and standing waves, in conjunction with changes in the channel shape, such as a contraction or the existence of a bridge pier or other obstructions in the flow field. These standing waves propagate down the channel far beyond the obstruction. Occasionally, a change in the shape of the channel causes the flow to transition from high velocity to low velocity flow, where the depth of flow is much greater and may result in flooding. By modifying the channel design, the damaging effects of these hydraulic phenomena can be reduced or eliminated.

The shape of the channel is determined parametrically by one or more design variables. For instance, the design variables for a channel consisting of straight wall segments would be the locations of the intersections of these segments, or for a channel with curved walls, the design variables could be the control points for a bezier curve or for a polynomial interpolating curve.

Once the channel's design variables $\vec{\beta}$ are chosen, an appropriate objective function F is selected. The objective function could measure the non-uniformity of the flow, the maximum height over a region or along the walls, or the change in the average energy or the head loss through a channel transition. Since the flow is dependent on the channel design, the objective function will indirectly be a function of the design variables. By adjusting the design variables, the value of the objective function can be minimized. The derivative of the objective function with respect to each of the design variables is approximated by using the adjoint variable formulation of direct differentiation[1], and the design variables are adjusted accordingly, which modifies the channel design. In the literature, these derivatives are referred to as the design space derivatives.

To simulate the flow through the channel, HIVEL2D[2] is used. This code solves the unsteady, two-dimensional, depth-averaged, viscous, shallow water equations[3], which ignore Coriolis forces and wind effects. It uses a Newton-Raphson iteration method to advance the solution in time and a Petrov-Galerkin finite element formulation for the spatial domain. This approach is presented in Section 3.

A conceptually simple approach to estimate each term $\frac{dF}{d\beta_i}$ is to use the finite difference approximation for each design variable, $\frac{dF}{d\beta_i}(\vec{\beta}) \approx \frac{F(\vec{\beta}+\Delta\beta_i)-F(\vec{\beta})}{\beta_i}$. For a problem with N design variables, one steady-state simulation is needed to calculate $F(\vec{\beta})$ and a steady-state simulation is needed for each perturbed grid $\vec{\beta} + \Delta\beta_i$, in order to calculate the one-sided finite difference. Hence, $N + 1$ steady-state simulations are needed, if a finite difference approximation for the gradient is used. Rather, by using the adjoint variable formulation for direct differentiation, the design space gradients can be estimated without the need for further steady-state simulations. This method and the corresponding modifications to HIVEL2D are presented in Section 4.

Once the gradient of the objective function is known, the design variables are adjusted using the method of steepest descent[4], or $\vec{\beta}^{n+1} = \vec{\beta}^{n} - \lambda\alpha\nabla F$, where α is a scaling term to ensure that the magnitude of the gradient and of the design variables are similar and λ is determined from one linear search step. The particular implementation of the method of steepest descent is presented in Section 5.

The material presented in this paper demonstrates that the adjoint variable formulation of direct differentiation can be applied to open-channel design optimization. Hence, channel design optimization can be accomplished in a deterministic fashion and with a much more reasonable amount of computation. To illustrate this technique, application has been made to four channel contraction design problems.

2 Background

Many different fluid dynamics codes simulate the flow of water through man-made open-channels and through other structures. Designers use these codes, in conjunction with scale models, to analyze the merits of a particular channel design. By using their expertise, they adjust the channel design in order to obtain the desired effects. Unfortunately, this trial and error process relies heavily on the designer, is not deterministic and provides no information concerning the nearness to an optimal solution.

Researchers in fields such as structural engineering and aerospace engineering have studied a variety of design problems and have successfully extended the capabilities of their simulation codes to include deterministic design optimization routines. The adjoint variable formulation of direct differentiation, which has been successfully used to calculate design space gradients for various aerospace applications, can be applied to the incompressible fluid dynamics codes that are used in modelling flow through man-made channels. The equations governing flow through man-made channels have some differences from those governing flow around airfoils. In particular, the friction models, turbulence models and dissipation models are different, and the hydraulic jumps in supercritical flow are inherently unsteady. Further, the simulation code HIVEL2D uses a finite-element approach, which is different from the algorithms used in the design optimization codes in aerospace engineering. In this paper, the effects of these differences on the optimization process are not analyzed, but rather, it is shown that the technique of the adjoint formulation of direct differentiation can be applied to the simulation code, HIVEL2D, despite these differences.

3 Discretization and Simulation Algorithms

Analysis of open-channel flow in hydraulically steep channels can be made by solving the unsteady, two-dimensional, depth-averaged, viscous, shallow water equations[3]. The particular formulation of these equations is given below with h representing depth of flow, p representing the flow discharge in the x direction and q representing the discharge in the y direction. This formulation neglects free-surface stresses and Coriolis forces.

$$\begin{aligned}
\frac{\partial h}{\partial t} + \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} &= 0 \\
\frac{\partial p}{\partial t} + \frac{\partial}{\partial x}\left(\frac{p^2}{h} + \frac{1}{2}gh^2 - h\sigma_{xx}\right) + \frac{\partial}{\partial y}\left(\frac{pq}{h} - h\sigma_{xy}\right) &= -gh\frac{\partial z}{\partial x} - g\frac{n^2 p\sqrt{p^2+q^2}}{C_0 h^{7/3}} \\
\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{pq}{h} - h\sigma_{yx}\right) + \frac{\partial}{\partial y}\left(\frac{q^2}{h} + \frac{1}{2}gh^2 - h\sigma_{yy}\right) &= -gh\frac{\partial z}{\partial y} - g\frac{n^2 q\sqrt{p^2+q^2}}{C_0 h^{7/3}}
\end{aligned} \tag{1}$$

For these equations, g is the acceleration of gravity, the $\sigma's$ are the various Reynold's stresses, z is the elevation of the channel bed, n is Manning's roughness coefficient and C_0 is a dimensional constant, which is 1.0 when the depth and velocities are measured in meters and meters/second and 2.208 when the depth and velocities are measured in feet and feet/second

The simulation code HIVEL2D[2] is used to solve the shallow water equations. HIVEL2D is an unstructured, finite element solver, which is used to simulate flow in both the supercritical and subcritical regimes and allows the specification of the appropriate boundary information for both cases. The approximation of actual water flow by the two-dimensional, shallow water equations is limited to the cases where the channel bed slope is small and where the depth-averaged assumptions are valid (i.e., the velocity in the vertical direction is negligible). Hence, HIVEL2D is not applicable to river flow simulation or to simulation of flow over a spillway.

HIVEL2D uses a Petrov-Galerkin finite element formulation for the spatial derivatives and the Newton-Raphson iterative method for the resulting system of nonlinear equations, $W^*(Q^n, X)$, where the vector X represents the coordinates of a particular grid and Q^n is the current set of flow variables. $W^*(Q^n, X)$ contains terms that represent the temporal derivative. At each time step, the Newton-Raphson iterative method solves the system by updating the set of flow variables Q^n, after solving the following system of equations for ΔQ

$$\left[\frac{\partial W^*(Q^n)}{\partial Q}\right][\Delta Q] = -[W(Q^n)] \tag{2}$$

4 Sensitivity Equations

For open-channel flow design problems, the objective function is directly a function of the flow variables. These flow variables depend on the channel design, which are determined from the design variables $\vec{\beta}$. Thus, the objective function may be written as $F = F[Q_{ss}(\vec{\beta})]$ where Q_{ss} denotes a vector of flow variables and $\vec{\beta}$ denotes a vector of design variables. The flow variables represent the steady-state values for a particular set of design variables. In a more general setting, the objective function can be a function of the flow variables Q, the grid discretization X, which is a vector of spatial coordinates dependent on $\vec{\beta}$, and the design variables $\vec{\beta}$. For the results in this paper, the objective function is only a function of Q_{ss}. In order to minimize the objective function, the gradient is needed and is calculated via a discrete implementation of the adjoint variable formulation of direct differentiation[1], which is presented below.

As discussed in Section 3, after discretization, an implicit algorithm yields a system of nonlinear algebraic equations, $W^*(Q(\vec{B}), X(\vec{B}), \vec{B})$. For the steady-state solution $Q_{ss}(\vec{B})$, the system of equation is equal to zero. Furthermore, the terms containing the temporal derivative can be ignored, resulting in a simplified system of equations, or

$$\left[W\left(Q_{ss}(\vec{\beta}), X(\vec{\beta}), \vec{\beta}\right)\right] = [0] \tag{3}$$

For this paper, the system of nonlinear equations is not directly dependent on $\vec{\beta}$. The code HIVEL2D solves this set of equations as described in the previous section.

In order to apply derivative based optimization algorithms to the objective function F, it is necessary to estimate the variation of F with respect to each design variable β_i. Hence,

$$\frac{dF}{d\beta_i} = \left[\frac{\partial F}{\partial Q}\right]^T \left[\frac{\partial Q}{\partial \beta_i}\right] \tag{4}$$

Once a steady-state solution has been obtained, differentiation of the system of equations arising from the solution algorithm yields

$$\left[\frac{dW}{d\beta_i}\right] = [0] = \left[\frac{\partial W}{\partial Q}\right]\left[\frac{\partial Q}{\partial \beta_i}\right] + \left[\frac{\partial W}{\partial X}\right]\left[\frac{\partial X}{\partial \beta_i}\right] + \left[\frac{\partial W}{\partial \beta_i}\right] \tag{5}$$

Multiplying eq. (5) by an arbitrary adjoint vector $[\lambda]$ and adding to eq. (4) yields

$$\frac{dF}{d\beta_i} = \left(\left[\frac{\partial F}{\partial Q}\right]^T + [\lambda]^T\left[\frac{\partial W}{\partial Q}\right]\right)\left[\frac{\partial Q}{\partial \beta_i}\right] + [\lambda]^T\left[\frac{\partial W}{\partial X}\right]\left[\frac{\partial X}{\partial \beta_i}\right] + [\lambda]^T\left[\frac{\partial W}{\partial \beta_i}\right] \tag{6}$$

By choosing $[\lambda]$ such that

$$\left[\frac{\partial F}{\partial Q}\right]^T + [\lambda]^T\left[\frac{\partial W}{\partial Q}\right] = [0]^T \tag{7}$$

eq. (6) simplifies to

$$\frac{dF}{d\beta_i} = [\lambda]^T \left[\frac{\partial W}{\partial X}\right] \left[\frac{\partial X}{\partial \beta_i}\right] + [\lambda]^T \left[\frac{\partial W}{\partial \beta_i}\right] = [\lambda]^T \left[\frac{dW}{d\beta_i}\right] \Bigg|_{Q fixed} \tag{8}$$

The vector $[\lambda]$ is calculated by taking the transpose of eq. (7) and solving the resulting equation

$$\left[\frac{\partial W(Q_{ss})}{\partial Q}\right]^T [\lambda] = - \left[\frac{\partial F(Q_{ss})}{\partial Q}\right]^T \tag{9}$$

Since HIVEL2D solves the similar equation,

$$\left[\frac{\partial W(Q^n)}{\partial Q}\right] [\Delta Q] = - [W(Q^n)] \tag{10}$$

HIVEL2D can be used to solve for $[\lambda]$ by changing the right-hand side and using the transpose of $\left[\frac{\partial W}{\partial Q}\right]$. From eq. (9), it can be seen that $[\lambda]$ is the same for each set of design variables and needs to be calculated just once for each design iteration. If there are more than one objective function to be optimized, then eq. (9) must be solved once for each objective function.

Once $[\lambda]$ is determined, $\frac{dF}{d\beta_i}$ can be approximated using the difference quotient for $\frac{dW}{d\beta_i}$. Thus,

$$\frac{dF}{d\beta_i}(\vec{\beta}) \approx [\lambda]^T \left[\frac{W(\vec{\beta} + \Delta\beta_i) - W(\vec{\beta})}{\Delta\beta_i}\right] \tag{11}$$

The primary advantage of the discrete adjoint formulation of direct differentiation is that for each iteration in the design optimization algorithm, the method of direct differentiation requires only one solution of eq. (9) per design iteration, instead of repeated solutions of eq. (10) in order to obtain a steady-state solution for each perturbed design variable.

5 Optimization Routine

For a particular set of design variables $\vec{\beta}_o$, only the function value $F(\vec{\beta}_o)$ and the gradient $\nabla F(\vec{\beta}_o)$ are known. Thus, the method of steepest descent is used to update the design variables. In general, the method of steepest descent updates the design variables by moving a certain distance in the direction of the gradient (if a maximum value is desired) or in the opposite direction of the gradient (if a minimum value is desired).

A scaling factor α is used to insure that the function value and the gradient are of the same magnitude as the design variables. In particular, $\alpha = F(\vec{\beta}_o)/|\nabla F(\vec{\beta}_o)|^2$. In one dimension, this value of α reduces the technique to Newton's Method for solving $F(\beta) = 0$. For general design optimization problems, the necessary condition for $\vec{\beta}_*$ to be the optimal solution is $\nabla F(\vec{\beta}_*) = 0$, but $F(\vec{\beta}_*)$ is not necessarily zero. Thus, as $\vec{\beta}$ approaches a minimum, the value of α grows without bound, unless $F(\vec{\beta})$ also approaches zero. To avoid this problem, the scaling factor α is limited in magnitude to some value, dependent on the problem.

To determine the best distance to travel in the gradient direction, an additional function evaluation is performed to obtain $F(\vec{\beta}_1) = F(\vec{\beta}_o - \alpha\nabla F(\vec{\beta}_o))$, and the curve in the gradient direction is approximated by a quadratic function, $\tilde{f}(\lambda) = F(\vec{\beta}_o - \lambda\alpha\nabla F(\vec{\beta}_o))$. Using $\tilde{f}(0) = F(\vec{\beta}_o)$, $\tilde{f}'(0) = -\alpha|\nabla F(\vec{\beta}_o)|^2$ and $\tilde{f}(1) = F(\vec{\beta}_o - \alpha\nabla F(\vec{\beta}_o))$, the minimum of the quadratic function occurs at

$$\lambda_{min} = \frac{F(\vec{\beta}_o)}{2F(\vec{\beta}_1)} \tag{12}$$

unless α has been set to its maximum value, in which case

$$\lambda_{min} = \frac{\alpha|\nabla F(\vec{\beta}_o)|^2}{2(F(\vec{\beta}_1) - F(\vec{\beta}_o) + \alpha|\nabla F(\vec{\beta}_o)|^2)} \tag{13}$$

If $\lambda > 1.0$, then λ is set to 1.0. This precaution is used when the function value $F(\vec{\beta}_1)$ is much less than $F(\vec{\beta}_o)$, in which case the quadratic function does not accurately reflect the problem to be minimized and its results are ignored.

The method of steepest descent has some disadvantages. Primarily, it yields only linear convergence which requires a large number of design iterations to converge to a solution. Also, this method often demonstrates a "zig-zag" behavior as it slowly progresses towards a solution. This behavior is demonstrated in Test Problem #3. If the Hessian matrix (second derivative matrix) was known or could be directly calculated, then the optimization process could yield superlinear, or even quadratic, convergence. The Hessian matrix can be approximated in various ways, but these approximations require many function evaluations, possibly negating the advantages of its use.

6 Test Problem #1 - Channel Contraction

The first test problem involved a channel contraction, with three design variables. The grid was structured with a width of 9 grid points and a length of 49 grid points, uniformly spaced in both directions. The channel contraction covered a length of 100 ft and contracted from 40 ft to 20 ft wide, as shown in Figure 6.1. The shape of the contraction was a curve determined by a sixth degree polynomial equation. This polynomial curve was chosen such that the curve matched the walls of the channel before and after the contraction, matched the slope of these walls before and after the contraction and passed through three specified points, which were the design variables. These three points were equally spaced in the direction of the flow and spaced normal to the general flow direction based on the value of the three design variables.

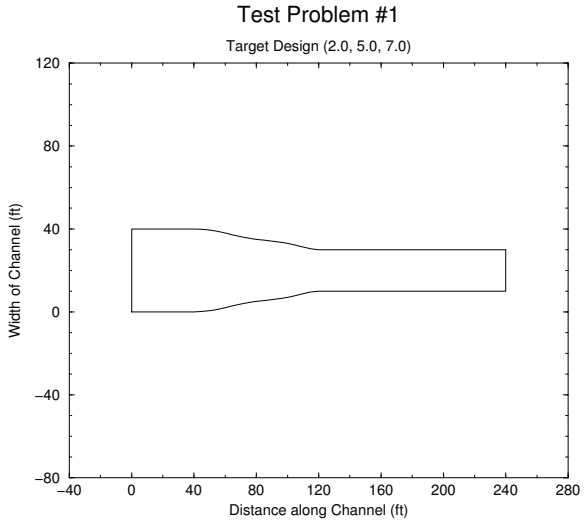


Figure 6.1. Target Channel Contraction.

The flow conditions upstream of the channel contraction were the depth = 1.0 ft, the tangential velocity = 28.375 ft/sec and the normal velocity = 0.0 ft/sec, resulting in a Froude number of 5.0, which was in the supercritical regime.

The goal of this test was to iterate from an initial set of design variables to a target set for which the flow was already determined. The target set was (2.0, 5.0, 7.0) and is shown in Figure 6.2.

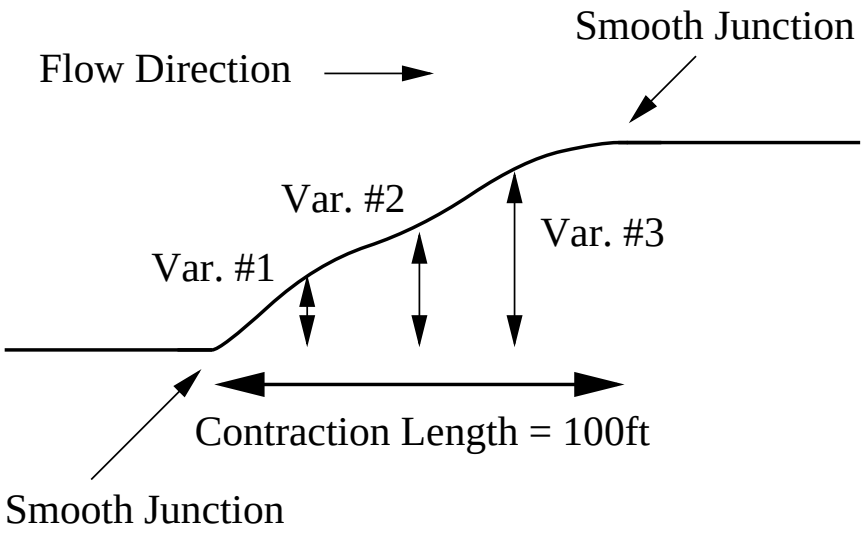


Figure 6.2. Design Variables for Contraction.

The objective function was chosen as the sum of the square of the differences between the target depth and the current iteration's depth for a set of grid points downstream of the contraction, in particular the grid points across the flow along one column of the grid, or

$$F(\beta_1, \beta_2, \beta_3) = \sum_{i=262}^{270} (depth(node_i, target) - depth(node_i, current))^2 \qquad (14)$$

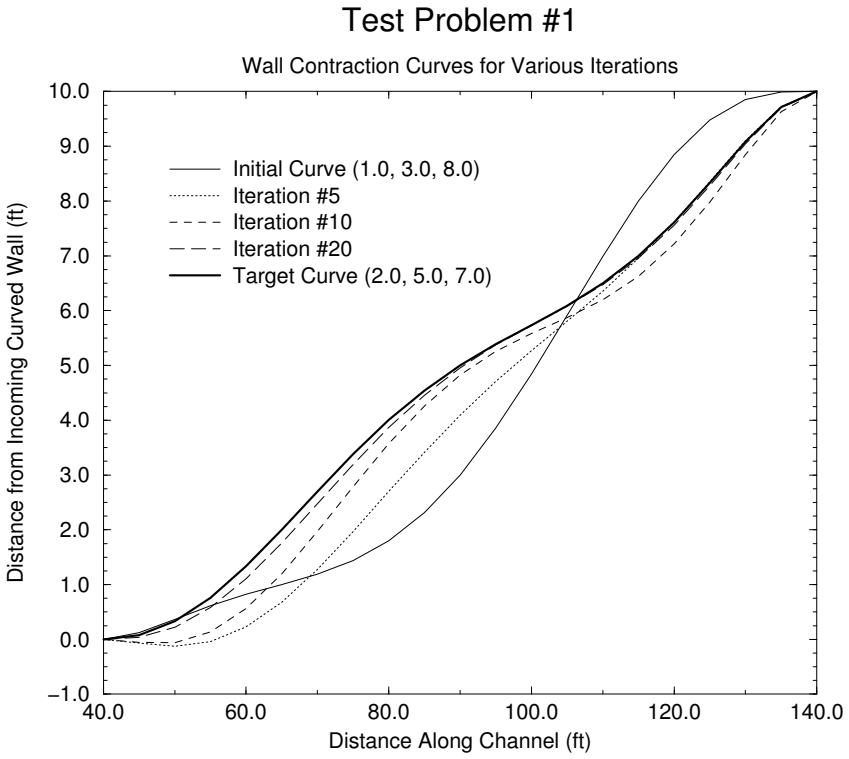


Figure 6.3. Contraction Curves for Various Iterations.

The initial set of design variables was (1.0, 3.0, 8.0), which produced an objective function value of 5.2522. Using the adjoint variable formulation of direct differentiation to estimate the value of $\frac{\partial F}{\partial \beta_i}$ for each design variable, the design variables were updated for successive iterations, using the method of steepest descent with one linear search as described in Section 5. Some of the curves for various iterations are shown in Figure 6.3. After 29 iterations, the design variables were (1.956, 4.999, 6.995), which produced an objective function value of 0.0001005. The stopping criterion was when each design space derivative was less than 0.01.

In order to observe the optimization process, the value of the function for each iteration was plotted versus the iteration number. Figure 6.4 shows that the design process updated the design variables so that the objective function continued to approach zero.

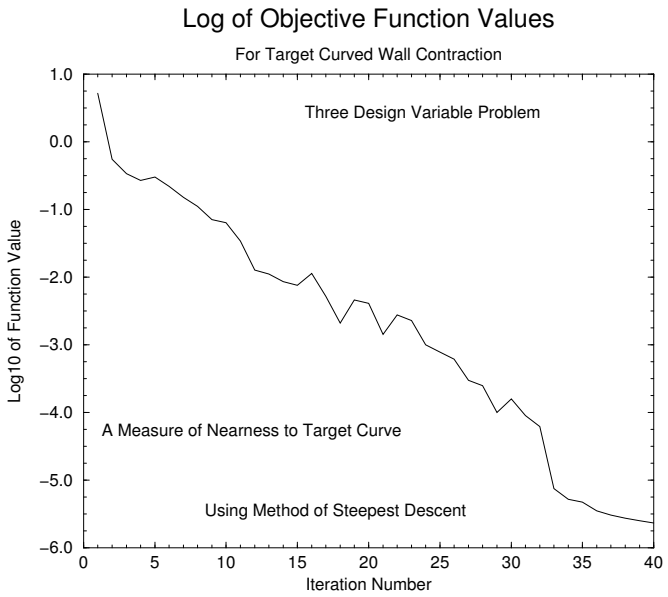


Figure 6.4. Log of Objective Function versus Iteration.

To demonstrate the iteration process, the values of the design variables, the design space derivatives and the scaling terms are listed for some of the iterations.

Iter.	β_1	β_2	β_3	$F(\vec{\beta}_o)$	$\frac{\partial F}{\partial \beta_1}$	$\frac{\partial F}{\partial \beta_2}$	$\frac{\partial F}{\partial \beta_3}$	$F(\vec{\beta}_1)$	α	λ
1	1.0	3.0	8.0	5.252	0.826	-3.135	4.025	0.552	0.197	1.000
2	0.838	3.616	7.209	0.552	0.214	-0.913	1.312	0.357	0.212	0.774
3	0.802	3.766	6.993	0.338	0.330	-0.680	-0.272	0.448	0.525	0.377
4	0.737	3.901	7.047	0.269	0.197	-0.627	0.303	0.231	0.513	0.583
5	0.678	4.089	6.956	0.303	-0.041	-0.341	0.617	0.194	0.609	0.784
...	...	...	...	...	...	...	...	...	...	...
25	1.896	4.983	6.984	.00079	-0.021	0.001	0.005	.00054	1.712	0.73
26	1.921	4.982	6.991	.00062	-0.013	-0.029	-0.0002	.0003	0.632	1.00
27	1.930	4.999	6.991	.00030	-0.014	-0.012	-0.0010	.0003	0.850	0.58
28	1.937	4.993	6.991	.00025	-0.012	-0.003	-0.002	.00010	1.669	1.00
29	1.956	4.999	6.995	.00010	-0.008	-0.007	-0.0008	.00018	0.847	0.28

7 Test Problem #2 - Optimal Channel Contraction Length

For the second test problem, the channel contraction had straight walls with the only design variable being the length of the contraction. From the equations governing an oblique hydraulic jump caused by a straight wall channel contraction, it can be demonstrated that the flow immediately downstream of the contraction is uniform if the contraction length is chosen properly. This phenomenon occurs because the waves created at the beginning of the contraction strike the walls at the end of the contraction and are negated by the waves created at the end of the contraction, as shown in Figure 7.1.

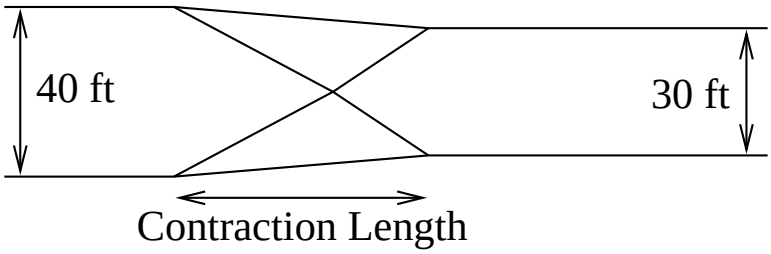


Figure 7.1. Overview of Channel Showing the Oblique Waves.

The correct contraction length is dependent on the properties of the incoming flow and on the amount of the channel contraction. For this problem, the Froude number was 5.0, the flow depth was 1.0 ft, and the channel contracted from 40.0 ft to 30.0 ft. The grid was a structured grid with 9 grid points across the channel. The number of grid points in the contraction was dependent on the length of the contraction so that the distance between grid points was approximately 5.0 ft. The objective function was chosen as a measure of the non-uniformity along 10 lines of 9 nodes across the channel, or the sum of the squares of the difference between the depths at adjacent grid points in 10 lines across the channel and is given by

$$F(\beta) = \sum_{\text{col=k}}^{\text{k+9}} \left(\sum_{row=2}^{9} \Big(depth(col, row) - depth(col, row-1) \Big)^2 \right) \tag{15}$$

Starting with a contraction length of 80.0 ft, the function value was 3.700. After 10 iterations, the value of the objective function was 0.062 for a contraction length of 157.87 ft. Using the equations governing oblique hydraulic jumps, the length of the contraction was determined to be approximately 155.65 ft. Since the equations governing oblique hydraulic jumps are derived using different assumptions from those used in the shallow water equations, one would expect the observed discrepancy in the two answers.

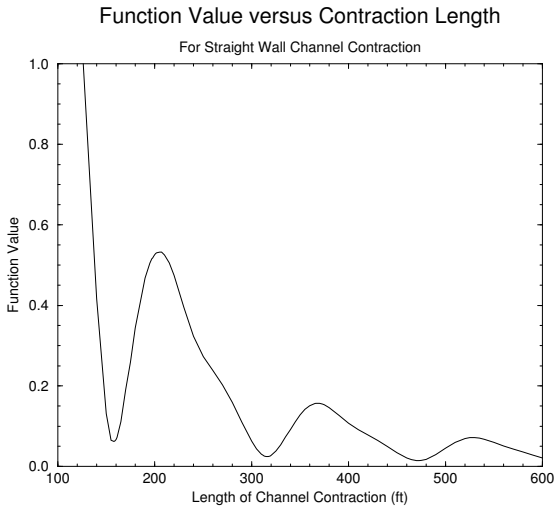


Figure 7.2. Graph of the Objective Function.

After finding the minimum value of the objective function, the function value was computed for various contraction lengths and plotted in Figure 7.2. From this plot, three minimums are seen, near lengths of 160, 315 and 480. These minimums correspond to the initial oblique wave hitting the end of the contraction with no reflections, with one reflection and with two reflections, respectively. By plotting the objective function value versus the iteration number (Figure 7.3), it is clear that the method of steepest descent found an excellent value of the design variable at iteration number 3. Further significant improvement could not be made because the minimum value of the objective function was not zero.

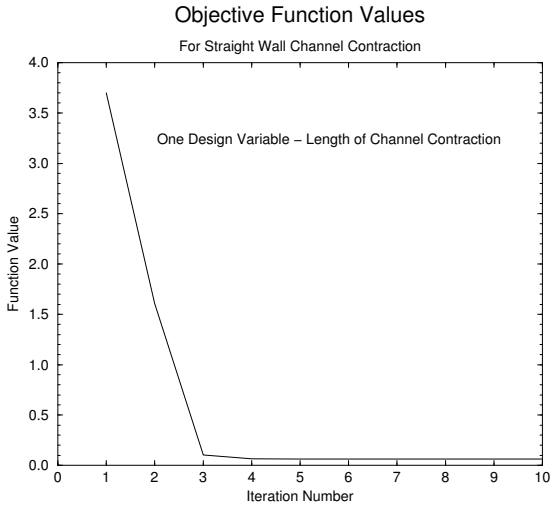
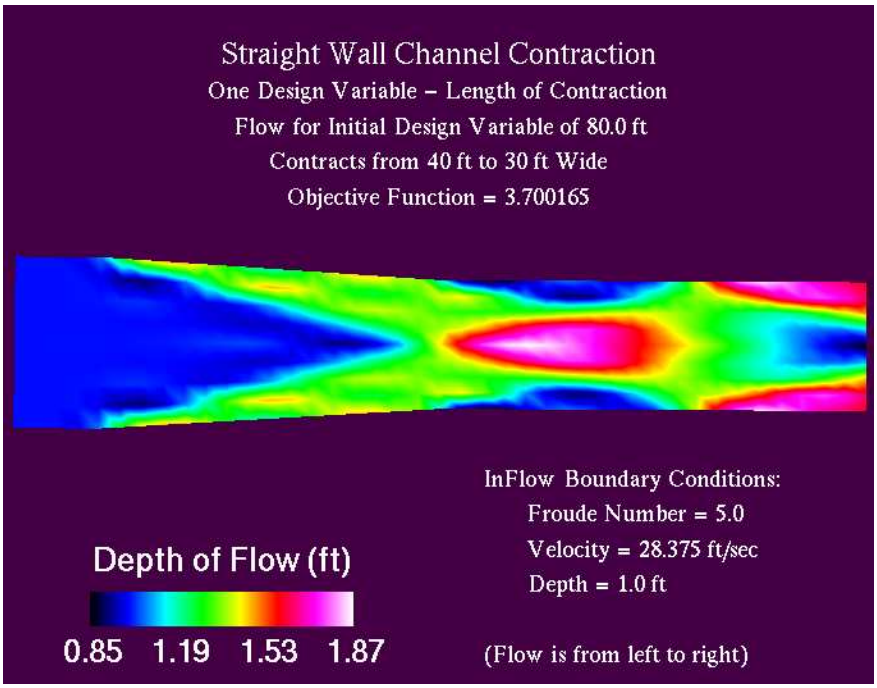
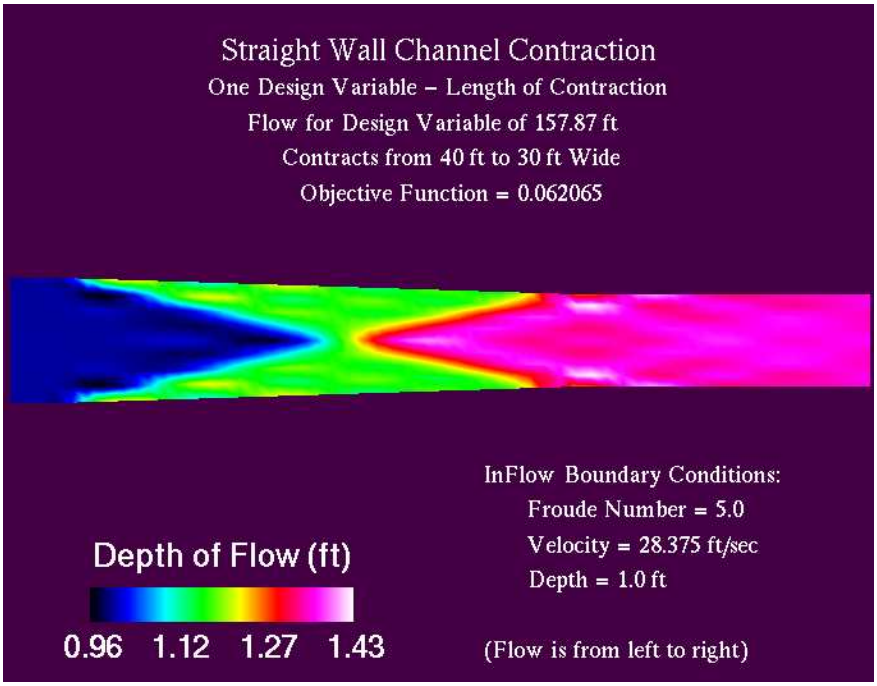


Figure 7.3. Graph of the Objective Function versus Iteration.

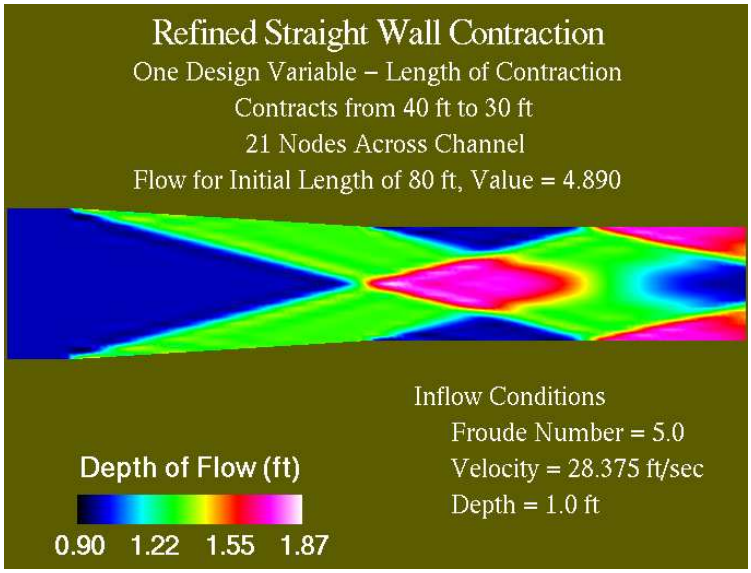
The plots of the depths for the steady state solutions generated by the initial design and the final design are given below.



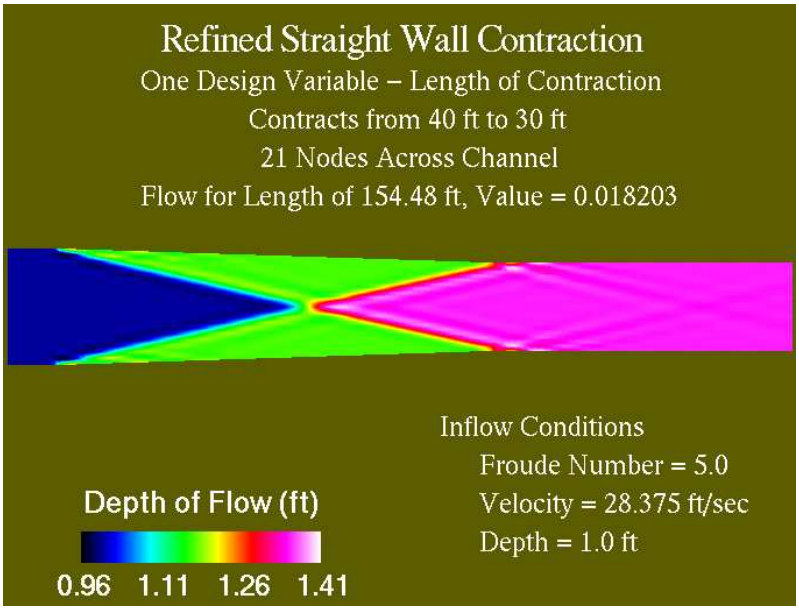


8 Test Problem #2a - Optimal Contraction Length on a Refined Grid

The goal of any optimization routine is to locate the region of an optimal value of a function for a particular problem. In the first two test problems, the derivatives produced by the adjoint variable formulation of direct differentiation coupled with the method of steepest descent guided the optimization process to the proper region in design space. In performing good computational work, one of the goals is to solve the discretized set of equations in such a way that the grid spacing does not affect the solution. For the first two test cases, the grid size was not refined enough to produce this grid independence. In order to test the grid independence of the design algorithm, the optimization process was applied to the previous problem with a more refined grid, using 21 nodes across the channel instead of 9 nodes and a corresponding number of nodes along the length of the channel.



After adjusting the objective function to measure the non-uniformity for the same region of the channel, the optimization process was applied with the initial contraction length of 80 ft, which yielded a value of 4.890. After 10 iterations, the contraction length was 154.48 ft producing an objective function of 0.018. The optimal contraction length for the original grid and the refined grid were off by over 3 ft. This difference is probably a result of the poor grid spacing, and the associated discretization error, of the previous problem. The results of this test problem indicate that the optimization process was able to locate the region of the optimal function value for this refined problem.



9 Test Problem #3 - Two Design Variable Channel Contraction Optimization

The third test problem was similar to the second test problem in that a uniform downstream flow was desired. In Test Problem #2, the design variable governed the length of the contraction, which consisted of one straight wall. In this test problem, the contraction length was fixed at 140.0 ft, which was shorter than the optimal length. In order to obtain uniform flow, the contraction walls were comprised of two adjustable, straight reaches instead of one, as shown in Figure 9.1. The two design variables controlled the location of the junction between the two reaches.

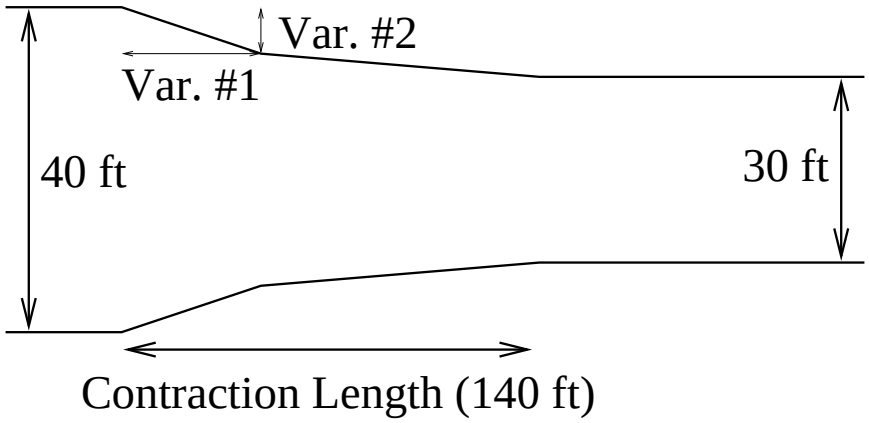


Figure 9.1. Overview of Channel Showing the Two Design Variables.

As in Test Problem #2, the grid was a regular grid with 9 grid points across the channel and 41 grid points along the length of the channel. The Froude number was 5.0, the inflow depth was 1.0 ft and the channel contracted from 40.0 ft to 30.0 ft. The number of grid points in each reach was adjusted so that the distance between the grid points in the two reaches were approximately the same. The objective function was similar to the one in the previous problem and was a measure of the non-uniformity across the channel. As a perfectly uniform flow was not expected, the minimum value of the objective function was not expected to be zero.

In this problem, the junction between the two reaches was initially located 70 feet downstream from the beginning of the contraction and 2.5 ft across the flow from the location of the initial channel width. Since these values were of different magnitudes, scaled design variables were used, so that both design variables were 2.5 initially. For these initial values, the objective function was 0.378. After 60 iterations, the objective function was 0.041 for design variables of 25.2868 ft and 0.3923 ft, a reduction of 89.3%.

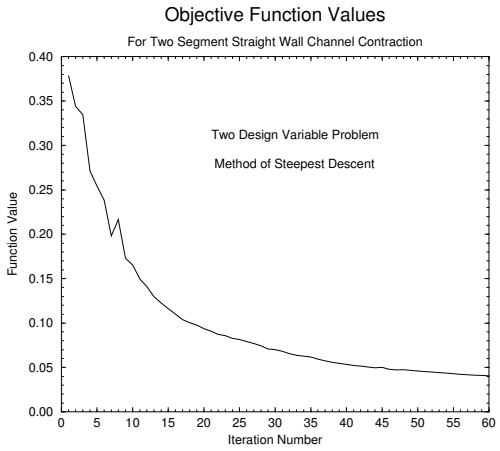


Figure 9.2. Graph of Objective Function Value versus Iteration.

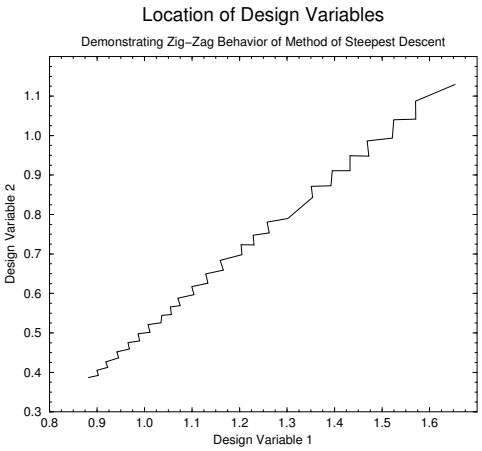
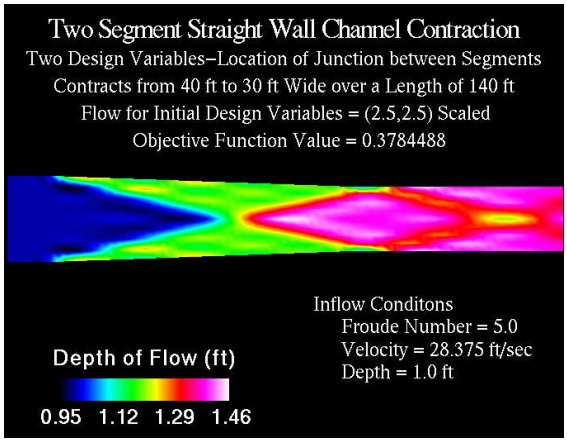
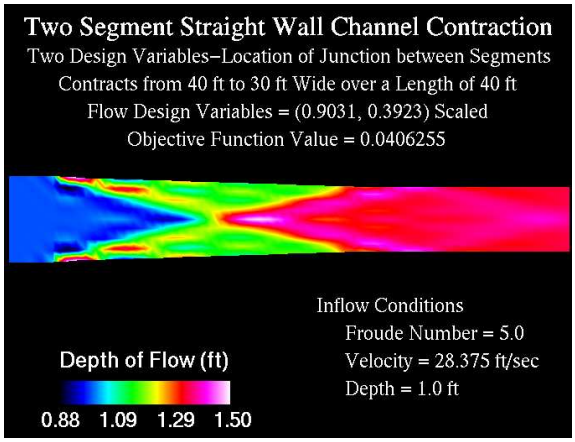


Figure 9.3. Graph of Location of Design Variables.

The graph of the objective function value versus iteration (Figure 9.2) shows that the value decreased rapidly at first and much more slowly as the design variables got closer to the optimal solution. By observing succesive values of the design variables (Figure 9.3), it was clear that the method of steepest descent was displaying its well-known "zig-zag" behavior. The location of the design variables moved from the upper right to the lower left for successive iterations.





10 Test Problem #4 - Uniform Flow for a Curved Wall Contraction

Test Problem #4 used a channel contraction similar to the one used in Test Problem #1, where three design variables determined the polynomial that defined the channel contraction curve. Recall that the channel contracts from 40 ft to 20 ft over a distance of 100 ft. For this test case, the goal was uniform downstream flow, so the same type of objective function used in the previous test problem was employed. For comparison purposes, a straight wall channel contraction of this length and for the flow conditions produced an objective function value of 2.235.

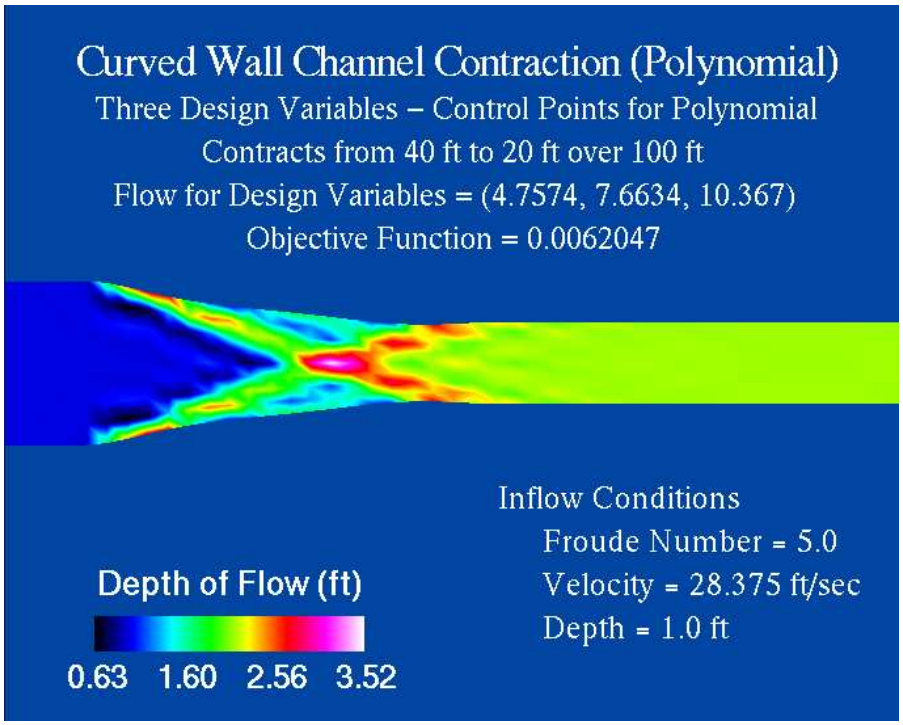
Because the goal was to obtain the global minimum of the objective function, the design iteration process was performed many times, starting from different initial sets of design variables. There were three possible outcomes for any design iteration process:

1.) Produce an invalid set of design variables, so that the simulation process failed for the resulting grid.

2.) Settle in a region of a local (sub-optimal) minimum.

3.) Settle in the region of the global minimum.

After running the design iteration process for 30 different initial sets of design variables, the design process blew up 3 times, found a local minimum 17 times and found the global minimum 10 times. For this problem, it appears that there is only one sub-optimal local minimum, near (3.49, 5.77, 8.30) yielding an function value of approximately 0.316. The global minimum lies near (4.76, 7.66, 10.37) and produces a function value of approximately 0.007, or a 99.6% improvement over the straight wall contraction.



11 Conclusion

From this preliminary investigation into using the adjoint variable formulation of direct differentiation to design high-velocity channels, the results are encouraging. The implementation using HIVEL2D demonstrates that it is possible to use this technique with a finite element solver and for the shallow water equations, neither of which had not been previously investigated.

From the viewpoint of an efficient design algorithm, the adjoint variable formulation is much better than a finite difference formulation of the gradient due to the need for only one steady-state solution calculation per design iteration, instead of $N + 1$ steady-state solutions where N is the number of design variables. Unfortunately, the method of steepest descent is not an efficient optimization algorithm as evidenced by the large number of design iterations, its linear convergence properties and "zig-zag" behavior. More research into efficient use of gradient information is recommended, including the study of a least-square formulation of the objective function, possible benefits of applying the conjugant gradient method, and the use of Hessian update methods.

The first two test cases presented in this paper were formulated such that an exact soluton for the design problem was known. In both cases, the design optimization technique correctly identified a close approximation to the solution and appeared to be converging to it. In the last two test problems, the optimal solution was not known. For the third test problem, the technique demonstrated an 89% improvement over the initial design. In the fourth test problem, a curved wall contraction was used whose shape was determined by 3 design variables. The global minimum was over 99% better than a straight wall contraction.

Hence, with simple geometries, the adjoint formulation of direct differentiation yields design space gradients that move the design variables towards an improved design, with much less computational expense. In all the test problems, this design process accurately identified a region in design space that contained an optimal solution for the particular problem. This deterministic technique is currently begin extended to more complex problems with friction and nonzero bed slopes. Also, various formulations of objective functions that measure non-uniformity are being analyzed, in order to find the most sensitive. The goal of this work is demonstrate that the adjoint variable formulation of direct differentiation can be coupled with a shallow water equation solver to yield a viable, deterministic, design optimization algorithm.

12 References

[1] Huddleston, D. and Soni, B. (1996), "Application of a Factored Newton-Relaxation Scheme to Calculation of Discrete Aerodynamic Sensitivity Derivatives," Inverse Problems in Engineering Vol. 3, pp.115-130.

[2] Stockstill, R., and Berger, R. (1994). "HIVEL2D: A Two-Dimensional Flow Model for High-Velocity Channels", Technical Report REMR-HY-12, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.

[3] Abbott, M. (1979). *Computation Hydraulics, Elements of the Theory of Free Surface Flows*, Pitman Advanced Publishing Limited, London.

[4] Scales, L. (1985). *Introduction to Nonlinear Optimization*, Springer-Verlag, New York.