CHAPTER IV

PRELIMINARY EXAMPLE

4.1. Problem Definition

The goals of the preliminary example are to compare the various methods for approximating the design space gradient and to discuss the various optimization algorithms in the context of an open-channel flow design optimization problem. An inverse design problem has been chosen where the target design and its depth are already known and the objective function is a least squares function that measures the variation from the target depths over a specified region. As a result, a successful optimization algorithm should be able to identify the target design variables to high accuracy and should be able to drive the objective function to machine zero.

In this example, a channel contraction is modeled where the channel contracts from 10 m to 8 m, as shown in Figure 4.1.1. The length of the contraction is determined from design variable 2. The shape of the contraction is defined by a B-spline curve with seven control points. The middle three control points are design variables 4, 5 and 6, but the first two control points and last two control points are chosen so that the contraction curve connects with the channel walls before and after the contraction. Design variable 1 controls the inflow depth, and design variable 3 determines the bed slope within the contraction. Furthermore, the entire length of the computational domain is constant, so that design variable 2 indirectly controls the length of the channel downstream of the contraction. The average slope is also fixed; thus design variable 3 indirectly controls the slope of the channel upstream of the contraction. Since the constant inflow velocity is calculated via Manning's equation, which involves bed slope, the inflow properties are controlled by more than one design variable. The complicated interactions of the design variables are shown in Figure 4.1.1.

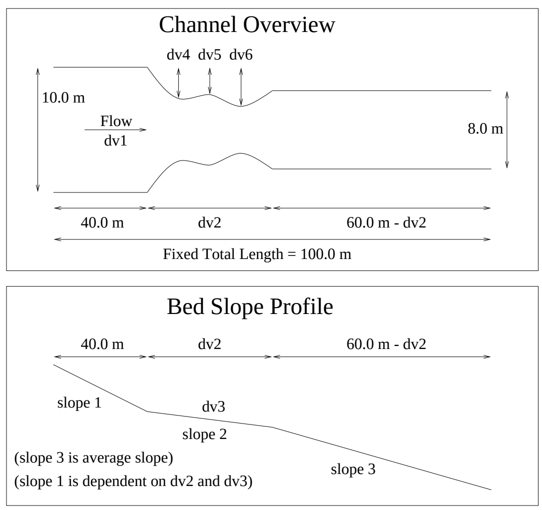

Figure 4.1.1 Channel Description for Preliminary Example.

To set up the inverse design problem, the steady-state solution for the target set of design variables is calculated. The target set of scaled design variables is (1.8, 0.9, 0.8, 1.0, 0.2, 0.4). The depths for three columns of nodes across the channel and downstream of the contraction are stored in an array h_k^{target}. The objective function measures the distance that the depths for the current set of design variables are from the targets depths via

$$F(\vec{\beta}) = \sum_{k=1}^{45} \left(h_k^{target} - h_k(\vec{\beta}) \right)^2 \tag{4.1}$$

The specifics of the preliminary problem are given in Table 4.1.1.

Table 4.1.1. Flow and Geometry Characteristics for Preliminary Example.

Channel Characteristic	Value
Width Before Contraction	10.0 m
Width After Contraction	8.0 m
Total Length of Contraction	100.0 m
Length Before Contraction	40.0 m
Length of Contraction	dv2 (β_2)
Length After Contraction	60.0 m - dv2
Discretization	15 by 276
Overall Bed Slope	0.0125
Total Drop in Bed Elevation	1.25 m
Bed Slope Within Contraction	dv3 (β_3)
Bed Slope After Contraction $s1$	$\frac{1.25 - dv2*dv3 - (60.0 - dv2)*0.0125}{40.0}$
Manning's n	0.013
Inflow Depth h	dv1 (β_1)
Inflow Discharge Rate Q	$\frac{(10h)^{5/3}\sqrt{s1}}{n(10+2h)^{2/3}}$
NROWS	15
NCOLUMNS	Varied

4.2. Analysis of Design Space Gradient

The design space gradient $\nabla_{\vec{\beta}} F$ is the gradient of the objective function with respect to the design variables. As discussed in Section 3.5., there are a variety of techniques available to calculate these derivatives including central finite difference, automatic differentiation such as ADIFOR, the complex Taylor's series expansion (CTSE) method and discrete sensitivity analysis. Furthermore, within discrete sensitivity analysis, the sparse Jacobian matrix $\frac{\partial W}{\partial Q}$ can be calculated in at least four ways - hand-differentiating the analytic expressions for $W(Q)$, using central finite differences, using the complex Taylor's series expansion method and using automatic differentiation tools such as ADIFOR. The analytic Jacobian matrix used in this research is not exact because it does not include the analytic derivatives of each term in the residual vector $W(Q)$. ADIFOR might be used to calculate this matrix, but since the matrix is sparse, a straight-forward implementation would be extremely computationally expensive because it would calculate every entry in the matrix. In this section, the design space gradient is calculated via finite differences, CTSE, ADIFOR and discrete sensitivity analysis for the initial set of design variables, which are (1.2, 0.6, 0.6, 0.8, 0.5, 0.6). The accuracy, computational efficiency and ease of use are analyzed with the conclusion that discrete sensitivity analysis with complex Taylor's series expansion of the Jacobian matrix is the best choice for this research.

Within discrete sensitivity analysis, the vector $\frac{dW}{d\beta}\big|_{\text{Q fixed}}$ is calculated via central differences, which requires the choice of a perturbation size $\Delta\beta$. Furthermore, the finite difference approximation of $\nabla_{\beta}F$ and of $\frac{\partial W}{\partial Q}$ require perturbation sizes. These perturbation sizes are chosen from within the range where the results are fairly consistent regardless of perturbation size. Perturbation analyses have been conducted to ensure that the perturbation sizes are in the proper range, but these analyses are omitted because they are not the focus of the current research.

The design space gradient generated by the various methods are presented in Tables 4.2.1 and 4.2.2. The derivatives estimated via discrete sensitivity analysis are calculated using the direct formulation rather than the adjoint variable formulation.

The results for the two formulations are in agreement to at least 12 significant digits, so their difference is quite small in comparison with the differences between other methods.

Table 4.2.1. Design Space Gradient Estimated Directly.

Variable	Finite Differences	Complex Expansion	ADIFOR
1	-78.789303538329	-78.789303543062	-78.789303543062
2	-4.4849021598736	-4.4849021394650	-4.4849021394650
3	0.48257301443755	0.48257304568685	0.48257304568697
4	-3.0353533482241	-3.0353532605749	-3.0353532605749
5	0.97931921239081	0.97931919595815	0.97931919595815
6	1.8955220237871	1.8955220265249	1.8955220265249

Table 4.2.2. Design Space Gradient Estimated via Discrete Sensitivity Analysis.

Variable	Approx. Analytic	Finite Diff.	Complex Expan.
1	-75.788778027029	-78.789314791974	-78.789303667435
2	-4.4848934416146	-4.4848787759972	-4.4849019423175
3	0.48233525668586	0.48257366816760	0.48257390861006
4	-3.0356656094503	-3.0353386511355	-3.0353530337111
5	0.97934854039888	0.97932104015682	0.97931859591698
6	1.8958424996999	1.8955272169893	1.8955211112235

The derivatives produced by ADIFOR and the complex Taylor's series expansion (CTSE) method are assumed to be highly accurate due to their theoretical developments. For this example, these derivatives are the same, except for design variable #3. The finite difference approximation to $\frac{dF}{d\beta}$ is correct to at least 7 significant digits. In regards to discrete sensitivity analysis, the analytic approach produced the poorest results with accuracies ranging from 3 to 5 significant digits. This inaccuracy can be attributed to the difficulty of obtaining the analytic expression for $\frac{\partial W}{\partial Q}$. The accuracy of the other two techniques involving discrete sensitivity analysis resemble the accuracy of the finite difference and CTSE methods in general. The finite difference approach is not as accurate as the complex Taylor's series expansion, which is accurate to between 6 and 8 significant digits. Thus, concerning accuracy, the best method for discrete sensitivity analysis is the complex Taylor's series expansion method.

In regards to computational efficiency, the derivatives produced by discrete sensitivity analysis are less costly to generate. Table 4.2.3 shows the computational cost of evaluating the objective function for the initial set of design variables and determining the design space gradient at this design point. In all cases, the initial conditions, the number of time steps and the step size are the same. By using the steady-state solution, once it has been calculated, as the initial conditions for the finite difference and complex Taylor's series expansion approximations of $\frac{dF}{d\beta}$, some computational expense can be saved, but this savings will not be substantial enough to affect the conclusions of this section. Furthermore, some computational savings can also result from a better use of time steps so that early termination of the flow solver occurs once the steady-state solution is determined. Execution time is used to estimate the computational cost for the various methods - memory usage has not been analyzed.

Table 4.2.3. Execution Times for a Design Iteration.

Technique	Execution Time
Central Finite Differences	2271.95 seconds
Complex Taylor's Series Expansion Method	4053.03 seconds
ADIFOR	4780.38 seconds
Discrete Sensitivity Analysis using Approx. Analytic $\frac{\partial W}{\partial Q}$	239.78 seconds
Discrete Sensitivity Analysis using Finite Differences $\frac{\partial W}{\partial Q}$	239.43 seconds
Discrete Sensitivity Analysis using Complex Expansion $\frac{\partial W}{\partial Q}$	247.69 seconds

The discrete sensitivity analysis codes are substantially faster than the other methods. Since the complex Taylor's series expansion and finite difference methods involve multiple steady-state solutions, their execution times are larger by a factor that scales with the number of design variables. For ADIFOR, the execution time also scales with the number of design variables since it calculates the derivatives by propagating the derivative information through the flow solver for each design variable. The computational cost is the least for the combination of finite differences and discrete sensitivity analysis, although the cost for each version of the discrete sensitivity analysis code is of the same order of magnitude.

Concerning the ease of implementation, the simplest method to implement is probably the finite difference method, since no modifications to the flow solver are needed. For the complex Taylor's series expansion method, the only changes to the flow solver deal with changing from real to complex arithmetic. These changes only affect input/output formats and conditional expressions, so implementation of this method is straight-forward. Using ADIFOR requires slightly more effort. The designer has to become familiar enough with ADIFOR so as to generate the proper derivative code. Then, this derivative code must be integrated within the existing code. Fortunately, the designers of ADIFOR have made it quite easy to learn, so that its application to the calculation of design space derivatives is straight-forward.

The use of discrete sensitivity analysis requires the rearrangement of subroutines and slight modifications to these subroutines. The grid generation subroutines or subroutines that produce the sensitivities of the grid with respect to the design variables are needed, which is also true for the other methods. Once the general structure of the discrete sensitivity analysis code has been written, the only difference in the three variations presented in this section deals with the calculation of the Jacobian matrix $\frac{\partial W}{\partial Q}$. Perhaps the simplest is the finite difference method, followed closely by the complex Taylor's series expansion method. If the exact analytic Jacobian is used in the flow solver, then there are no modifications to the Jacobian, and the analytic method is easiest to use, but if the Jacobian in the flow solver is only approximate, which is typically the case in most high-fidelity codes, much effort may be needed to derive the exact analytic Jacobian.

Due to the computational efficiency, the discrete sensitivity analysis methods are a much better choice for computational design optimization, even though their implementation requires more effort than the other methods. In regards to accuracy, the use of ADIFOR and the complex Taylor's series expansion method produce extremely accurate derivatives. The complex Taylor's series expansion version of discrete sensitivity analysis produces the most accurate derivatives, among the discrete sensitivity analysis methods. Thus, for the present research, the combination of discrete sensitivity analysis and complex Taylor's series expansion has been chosen as the best method to generate highly accurate design space derivatives efficiently.

4.3. Comparisons of Optimization Algorithms

Of the six algorithms for unconstrained optimization that are described in Appendix C, the method of steepest descent, the BFGS Hessian update method and the modified Gauss-Newton method have been applied to the preliminary example. Since the exact Hessian matrix is not available, Newton's method can not be used. The conjugate gradient method is not used because its convergence rate depends on the number of design variables - thus this method does not scale well for design problems with large numbers of design variables. The trust region model is not used because the convergence rate of the modified Gauss-Newton method is quite sufficient, especially considering the requirement of the trust region model for additional function evaluations. For the method of steepest descent and the BFGS update method, one additional function evaluation is performed in the search direction for each design iteration. Thus, these optimization algorithms are more computationally expensive per design iteration than the modified Gauss-Newton method, for this problem. (Since the computational cost of estimating the design space gradients scales with the number of design variables for the direct formulation of discrete sensitivity analysis, optimization problems where the number of design variables is far larger than the number of objective and constraint functions may benefit from the use of another optimization algorithm such as the BFGS method. The BFGS method also offers the potential for superlinear convergence and does not require the additional derivative information, needed for the Gauss-Newton method. Since there are several choices for estimating the design space gradients as well as several choices of optimization algorithm, these choices can greatly affect the performance of the overall design optimization algorithm.)

Starting with the initial set of design variables of (1.2, 0.6, 0.6, 0.8, 0.5, 0.6), the optimization process has been executed for 100 design iterations, using the three optimization algorithms. The convergence histories are given in Figure 4.3.1.

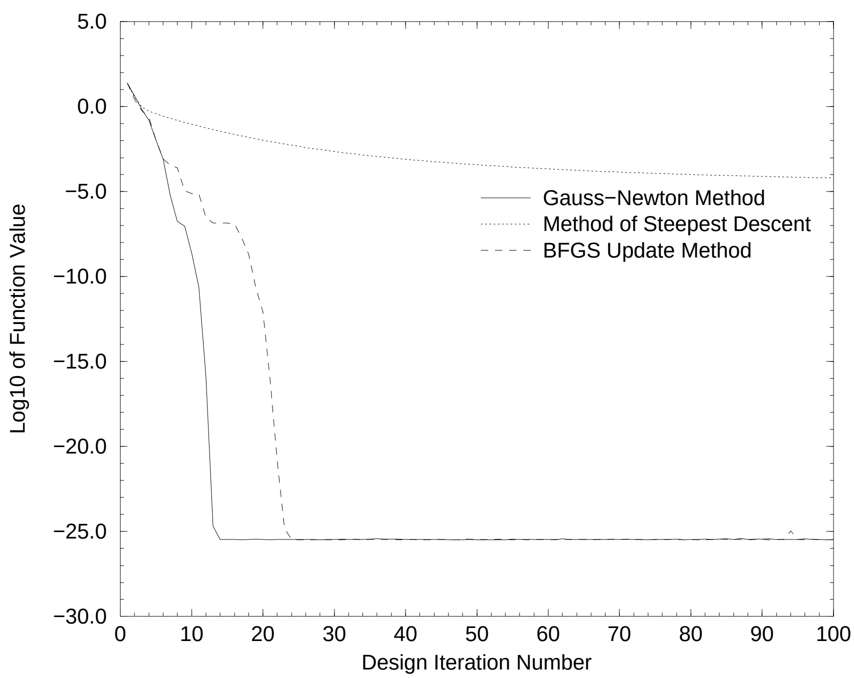

Figure 4.3.1. Optimization Histories for Three Algorithms

The Gauss-Newton method and BFGS update method are able to reduce the objective function to machine accuracy, but the method of steepest descent does not succeed even after 100 design iterations. For the method of steepest descent, the objective function value monotonically decreases, but the rate of convergence is not satisfactory. For the Gauss-Newton method, the objective function is on the order of

10^{-26} after only 14 design iterations; the objective function is on the same order for the BFGS update method after 24 iterations. Concerning the design variables, once the objective function is on the order of 10^{-26}, they are in agreement with the target set of design variables to at least 11 significant digits. Thus, both the Gauss-Newton method and BFGS update method appear to satisfy the criterion for excellent optimization algorithms in that they are able to drive the objective function to machine accuracy and are able to identify the optimal set of design variables to a high degree of accuracy. Since the Gauss-Newton method identifies the optimal solution in fewer design iterations and because the BFGS method requires an additional steady-state simulation, the Gauss-Newton method has been chosen for this research.

As demonstrated with this preliminary example, the complex Taylor's series expansion method applied to discrete sensitivity analysis is able to accurately approximate the design space derivatives with minimal computational expense. Furthermore, the modified Gauss-Newton method uses the derivative information quite effectively, reducing the number of design iterations that are necessary to converge to the optimal set of design variables. Thus, the overall computational expense for the design optimization process is dramatically reduced because the cost of approximating the design space derivatives is quite small, as is the number of design iterations.