Sandia National Laboratories

Operated for the U.S. Department of Energy by
Sandia Corporation

Albuquerque, New Mexico 87185- 0557

date: March 16, 1999
(Original February 18, 1999; revised March 16, 1999, TWS)

to: Distribution

from: T. Simmermacher, 9119, MS0557

subject: Slot Coating Optimization (GT-010.2)

keywords: DAKOTA, cost function, slot coating, optimization, dynamic contact angle sensitivity

input records: None

Introduction

This memo describes in detail the optimization process applied to a slot coater. Initially, many parameter studies are performed to insure that the cost function and its gradient are being accurately calculated. This is critical when using a gradient-based optimizer. If the gradient is difficult or impossible to calculate accurately, either analytically or through a finite difference scheme, then a non-gradient based method should be used such as a genetic algorithm or a coordinate pattern search technique.

Cost Function and Constraint Design

The goal of an optimization is to find the design parameters such as material properties, geometric dimensions, inflow velocities, etc., that minimize a cost (performance) function that is dependent on those parameters, subject to constraints. The careful design of the function to be minimized and the constraints is critical to insure that the optimized design is in fact the desired design.

The design of a cost function and constraints is largely an art form. A design issue such as minimizing the distance between two curves, $x(t)$ and $y(t)$, can be formulated as

$$\min\ (x(t) - y(t))^2 \qquad \text{(EQ 1)}$$

or, equivalently

$$\min\ |x(t) - y(t)| \qquad \text{(EQ 2)}$$

or any multitudes of formulations. Each one has its advantages and each may result with a slightly different solution or a different convergence rate. In the case of the distance between two lines, EQ (2) has an undefined derivative when $x=y$. This discontinuity could cause problems with a gradient based optimization routine. The numerical condition of two similar cost functions may affect the results of an optimization. If a very accurate optimum is required, subtracting then squaring two very similar quantities could result in round -off errors near the optimum, as would be done with EQ (1). So if one representation of a cost function is not performing as one would like, a different representation or scaling may perform better.

A slot coater design will be described in this memo. The parameters available for optimization are shown in Figure 1. The initial configuration is based upon a design found in:

> Sartor (1990), *Slot Coating*, Ph.D. Thesis University of Minnesota, Available on University Microfilms, Ann Arbor, Michigan.

Slot Coater Optimization Example

The optimization posed here attempts to optimize the shape of the lip. For this memo only bound constraints will be necessary although a constraint that is never active will be included to enable a comparison between constrained and unconstrained optimizers. The cost function will be based upon the motion of the wetting line (left end of die land marked L2 in Figure 1). The objective will be to determine the shape of the lip that makes the position of the wetting line as insensitive as possible to changes in web speed and back pressure. This would insure that the wetting line position would be fairly constant despite the expected, yet unpredictable changes in the web motion and in the vacuum.

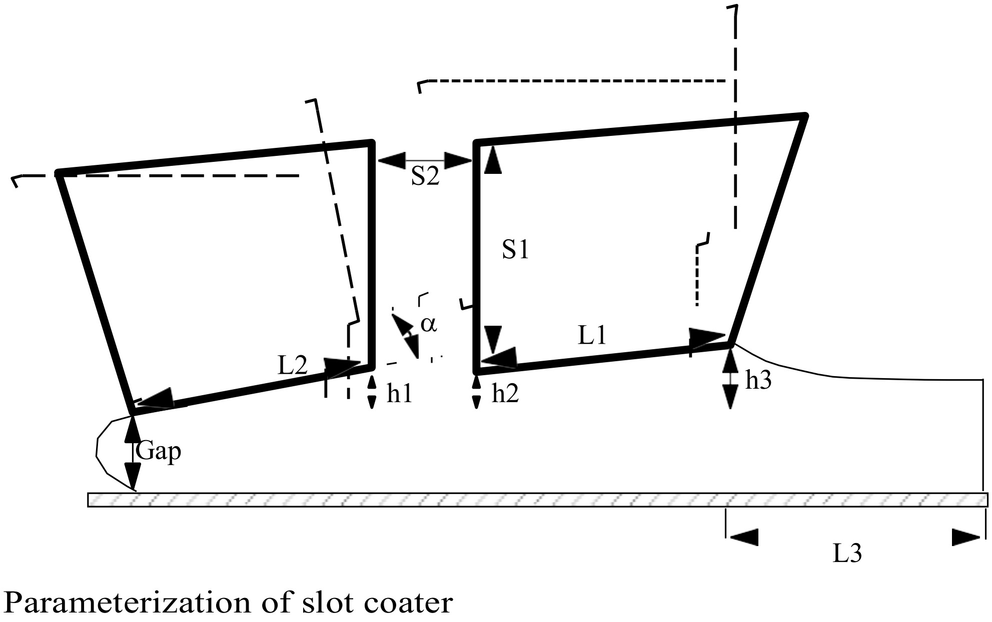

Figure 1. Parameterization of slot coater

A simple approach to formulating the above described cost function is to use partial derivatives. Although this a conceptually easy approach, care will have to be taken because effectively, second derivatives will be necessary for a gradient-based optimization approach. While finite differencing for first order derivatives can be difficult, the problems are even more severe for second order derivatives.

The cost function used can be formally stated as

$$\min J = \left(\gamma \frac{\partial X_{dcl}}{\partial U_{web}} + \beta \frac{\partial X_{dcl}}{\partial P_v} \right)^2 , \qquad \text{(EQ 3)}$$

where γ and β are scaling factors, X_{dcl} is the position of the wetting line, U_{web} is the web velocity, and P_v is the back pressure. The parameters to be optimized have yet to be defined. The scaling factors allow the designer to weight the two sensitivities based upon engineering judgement. Later, some ad-hoc methods of initially choosing the weightings will be given and a method to show the trade-off between the two sensitivities will be presented.

The sensitivities in the cost function will be calculated using forward differences defined as

$$\frac{\partial f}{\partial y} \approx \frac{f((1+\varepsilon)y) - f(}{\varepsilon y} , \qquad \text{(EQ 4)}$$

where $f(y)$ and ε is a chosen finite difference increment. With the finite difference defined as in EQ 4, the step size is defined as a percentage and is therefore independent of the magnitude of the independent variable. The increment must be carefully chosen and the smoothness of the function must be considered. The effect of a poorly converged simulation on the estimated derivative will be shown in the section describing parameter studies.

Constraints could be easily added to prevent some undesirable characteristics of the flow to occur. As will be seen, the unconstrained problem will generate one solution which has a large recirculation region under the lip. This region could be substantially reduced by placing a residence time constraint on the particles in the flow. Another constraint might be the stability of the flow. This type of constraint might specify that the largest pole must be smaller than a certain value, insuring that the largest pole never goes unstable, despite uncertainty or disturbances in the flow regime.

The initial values of all the parameters are shown in Figure 2 along with the initial mesh. Hopefully, the initial values places the slot coater in a practical point in design space. If the design space is multi-modal (e.g. having many optimum), the results can be dependent on the starting point. For example, when using a gradient-based approach, the optimizer will always want to go downhill. If there are many hills in the design space, the optimum will depend on which hill and what side of the hill the initial point is chosen. A global optimizer, such as a genetic algorithm (GA) based optimizer will tend to be less sensitive to the initial points than will a gradient based optimizer.

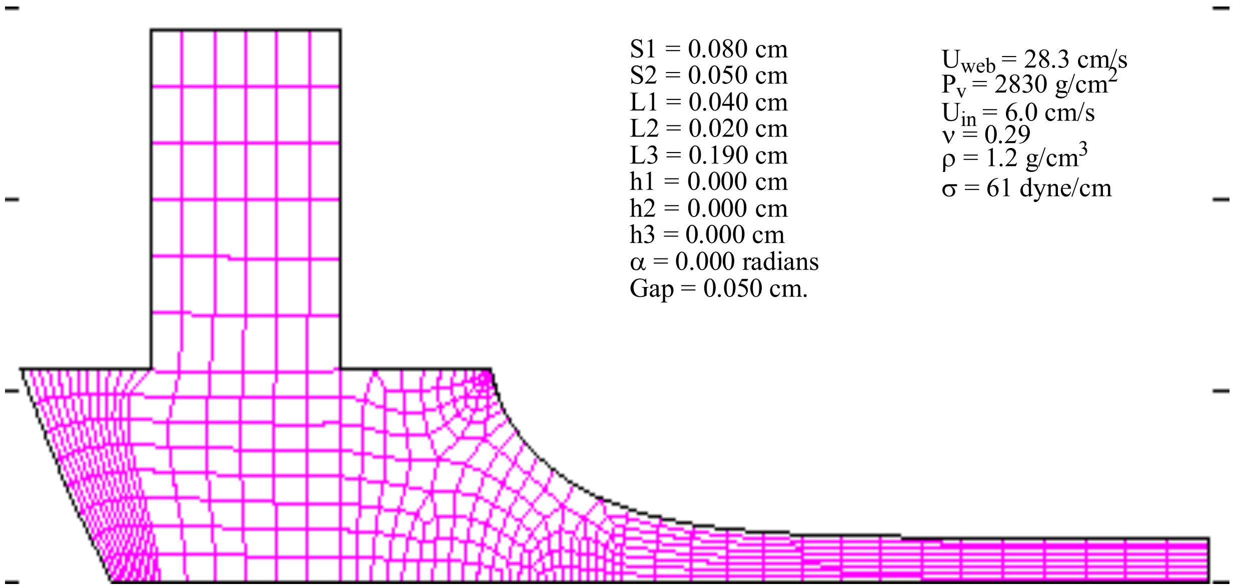

Figure 2. Initial design of slot coater

Parameter Space Analysis

Before an optimization is performed, it is a good idea to perform some parameter studies to map out the design space to determine if the simulation is adequately converged, to determine the shape of the design space, and to determine the sensitivity of the cost functions to each of the candidate parameters.

Determining if the simulation is adequately converged is critical when finite differencing is used. A poorly converged simulation, while producing a plausible solution at a single point, may produce a non-smooth surface when the solutions at many points are compared. Figures 3 and 4 show the results of a well converged simulation and a poorly converged simulation. The function plotted is the position of the wetting line with respect to the web speed and the back pressure. Note that this parameter study was performed on a very small scale, on the order of the half the finite difference step size (finite difference step size is 0.1%). This function will be used in the calculation of the sensitivities used in the cost function (EQ 3).

A poorly converged solution typically is very non-smooth. Note that if the poorly converged simulation (Figure 4) is used in a finite difference calculation, the result might not represent the true value of the derivative accurately. In fact, the estimate of the derivative could conceivably have the wrong sign. Many times, a strangely behaving optimization can be tamed by checking the quality of the function evaluation by a local parameter study in the neighborhood where the optimization is flaking out.

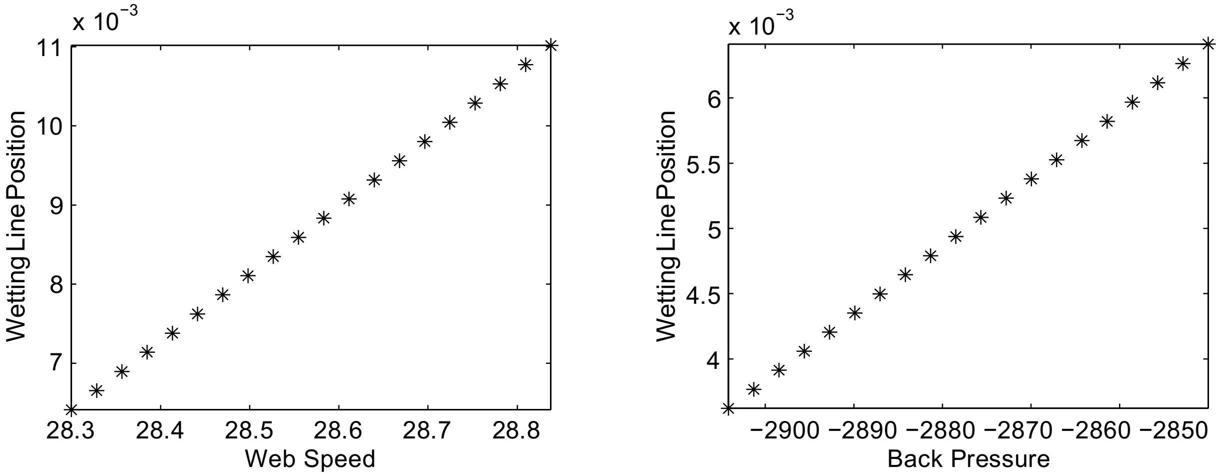

Figure 3. Converged simulation for a finite difference parameter study

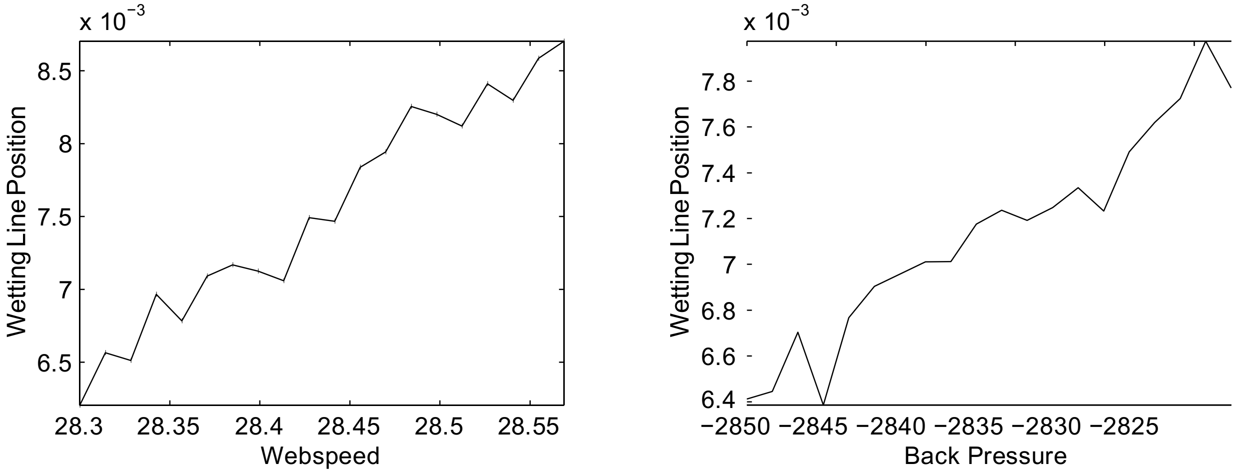

Figure 4. Poorly converged simulation for a finite difference parameter study

Once the sensitivities for the cost function are insured to be accurately calculated, the gradient of the cost function needs to be studied. The gradient in the cost function presented here is basically a second derivative. The individual sensitivities used in the cost function are plotted in Figure 5 at half the finite difference step size (finite difference step size is 0.1%). The two parameters chosen for the study are the gap height and the inlet angle, α. These parameters are the parameters used for the optimization study.

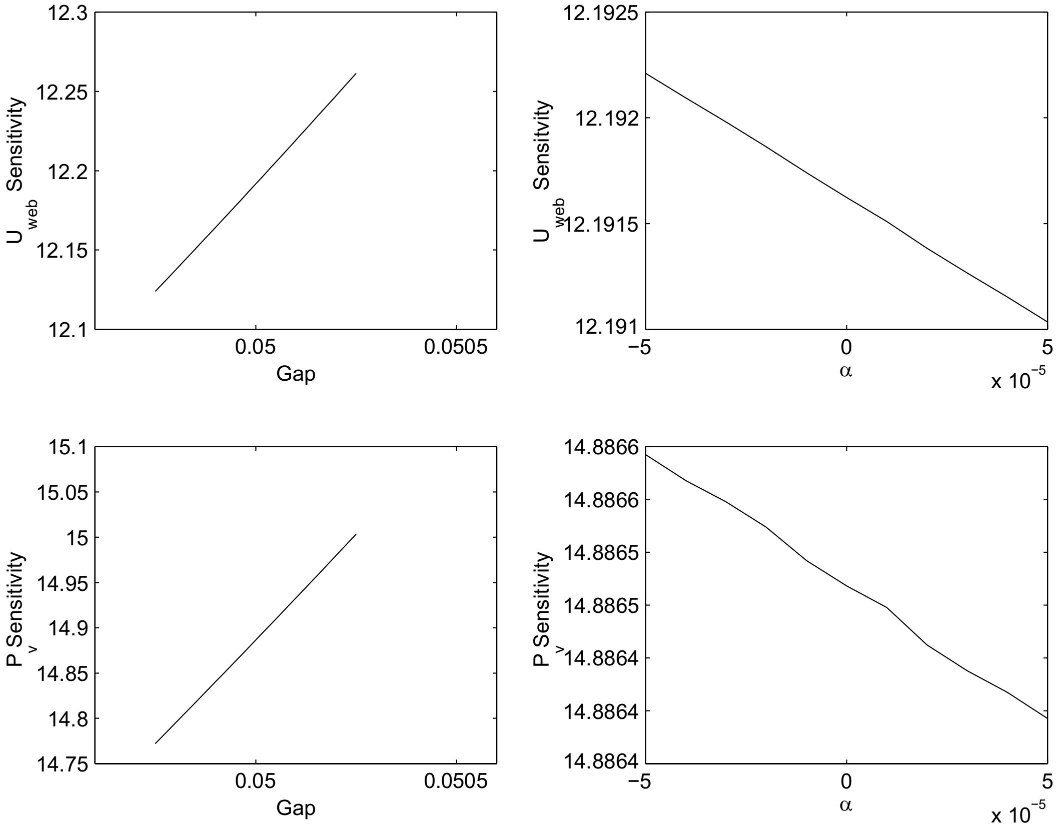

Figure 5. Cost function smoothness

Once the function evaluation has been determined to be correctly calculated, the weightings γ and β need to be determined. For this study, each scaling will be broken into two parts as

$$\sqrt{J} = (\gamma_2\gamma_1)\frac{\partial X_{dcl}}{\partial U_{web}} + (\beta_2\beta_1)\frac{\partial X_{dcl}}{\partial P_v} \quad . \qquad \text{(EQ 5)}$$

The first scaling (subscript 1) is a simple scaling based upon the magnitude of the numerator and denominator used for the finite difference calculation. Basically this scaling uses the nominal order of magnitudes of the quantities used for the finite difference to normalize the sensitivities. For the first sensitivity, $\frac{\partial X_{dcl}}{\partial U_{web}}$, the web speed ($U_{web}$) is nominally 28.3 cm/s and the numerator is approximately 0.01 centimeters (see

Figure 3). An appropriate initial scaling would be $\gamma_1 = \frac{28.3}{0.01}$. Similarly, for the second sensitivity, $\frac{\partial X_{dcl}}{\partial P_v}$, the nominal value of the back pressure (P_v) is 2850 g/cm^2 making an appropriate initial scaling of $\beta_1 = \frac{2850}{-0.01}$. By setting $\gamma_2 = 0.5$, both sensitivities will be approximately the same at the initial point used in this optimization.

The second scaling (subscript 2) can be chosen based on engineering judgement as to the relative importance of each sensitivity.

Lastly, it is usually helpful to plot out the design space, if possible. In this case, there are only two parameters so a visualizable response surface can be generated (Figure 8). If generating a response surface is not possible or useful, at least a parameter study in each of the design variables will give one a feeling of how the surface will look (Figure 9), although the results could be deceiving.

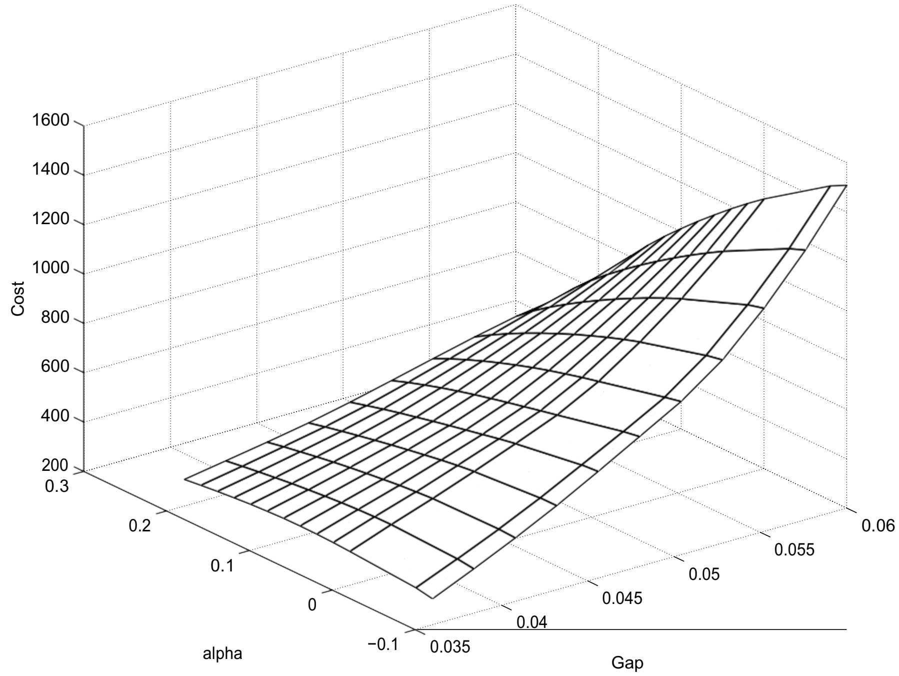

Figure 6. Response surface for slot coater

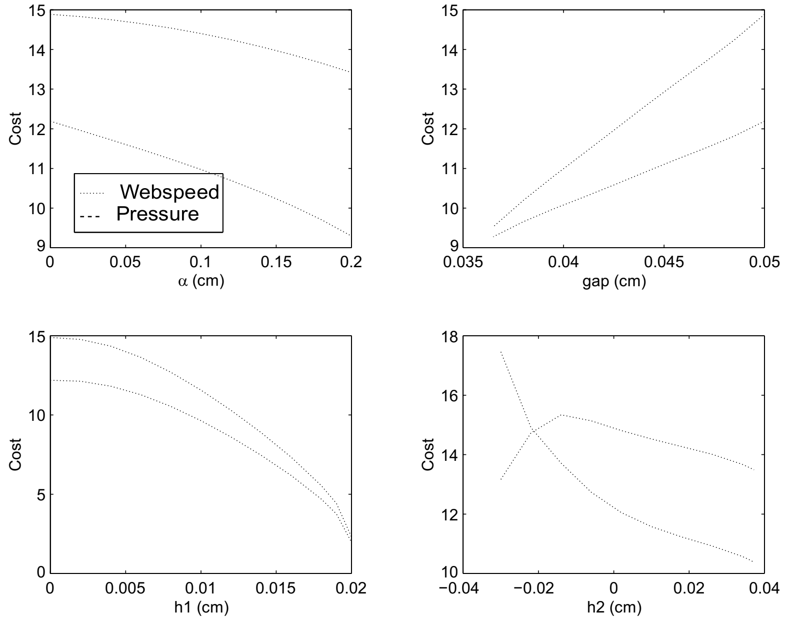

Figure 7. Parameter studies of four variables

Optimization Strategies

This section will describe the results of running a few different optimizers to compare the results that they calculate and the number of function evaluations necessary to for them to converge. Different situations call for different optimization strategies. There is no one optimization method that is good in all situations. It is good to try a few different methods starting at a few different starting points, especially if a good parameter map of the design space has not been generated, to insure that the optimum returned by the optimization is in fact the optimal point. In this example a combination of gradient-based (sequential quadratic programming, SQP) and non-gradient based (coordinate pattern search, CPS) will be used. A non-gradient method is good when gradient information is expensive to calculate, if the gradient information is inaccurate, or if the design space is discontinuous. Also, a global, non-gradient based method, such as a genetic algorithm (GA) can be used to globally search a design space for a promising region, then the optimizer could switch to a gradient based method to “fine-tune” the optimum.

In general, one of the quickest converging gradient based constrained method is the SQP method. It is a constrained version of the BFGS algorithm meaning that it generates an approximation to the Hessian (second derivative) matrix with a penalty method to enforce the nonlinear constraints. In the DOT package, running the SQP algorithm with no constraints reverts to the BFGS method. As will be seen, there are differences in the performance of a SQP run with no constraints (BFGS) and a run with only one constraint that is always inactive (the constraint is always negative). This is most likely due to the implementation of the constraint logic which although should never affect the optimization, does subtly affect the convergence.

Figure 10 compares the convergence of the NPSOL implementation of the SQP algorithm and the DOT implemented. The NPSOL algorithm requires three more function evaluations that did the DOT algorithm. Also note how the NPSOL algorithm searches around the true optimum in an attempt to better the value. This optimum is on a bound and NPSOL may have a more thorough constraint interrogation. The NPSOL algorithm also converges to a slightly higher value because it is not as tight to the constraint boundary. In both algorithms, DAKOTA's automated continuation algorithm was needed to traverse the long drop from the initial points (gap=0.05 cm and alpha = 0.0 radians, cost=729) to the optimal values (gap=0.035, alpha=0.2, cost=300).

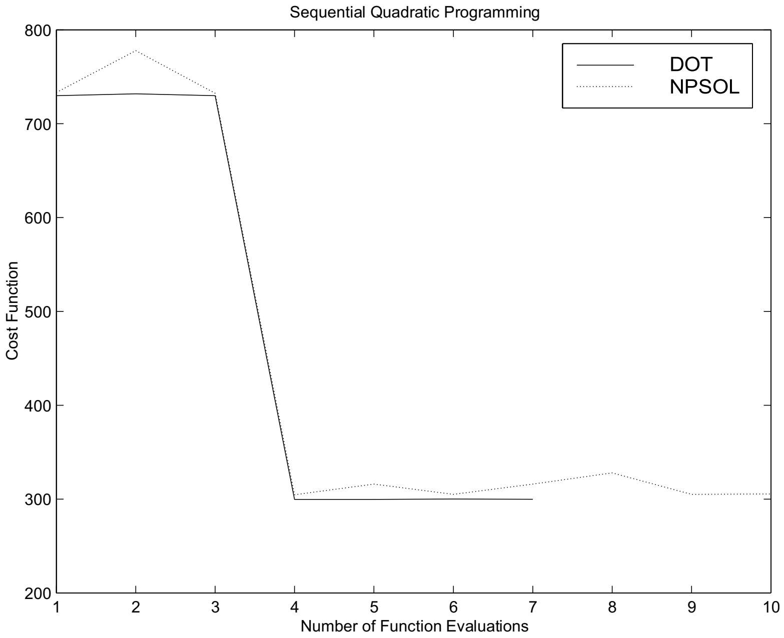

Figure 8. Convergence of SQP Algorithms

The next comparison is shown in Figure 11. This compares DOTs constrained optimization (SQP) with the unconstrained BFGS algorithm. The SQP with no constraints and the BFGS show identical convergence. The DOT package, when SQP is run without any constraints, reassigns the algorithm to the BFGS algorithm and therefore they should be identical. The more interesting comparison is between the SQP with a constraint that is always satisfied (inactive) and the BFGS. These runs would be expected to be identical because the SQP uses the BFGS algorithm with some modifications to handle the constraints. Since the constraint is never active, the convergence should be the same as the BFGS. According to this study the constraint logic subtly affects the convergence. In fact the BFGS algorithm finds the other optimum in this design space (alpha=-0.1, gap=0.35, cost=276). This optimum is slightly lower in cost and has the added advantage of not having a large recirculation region, as will be shown later.

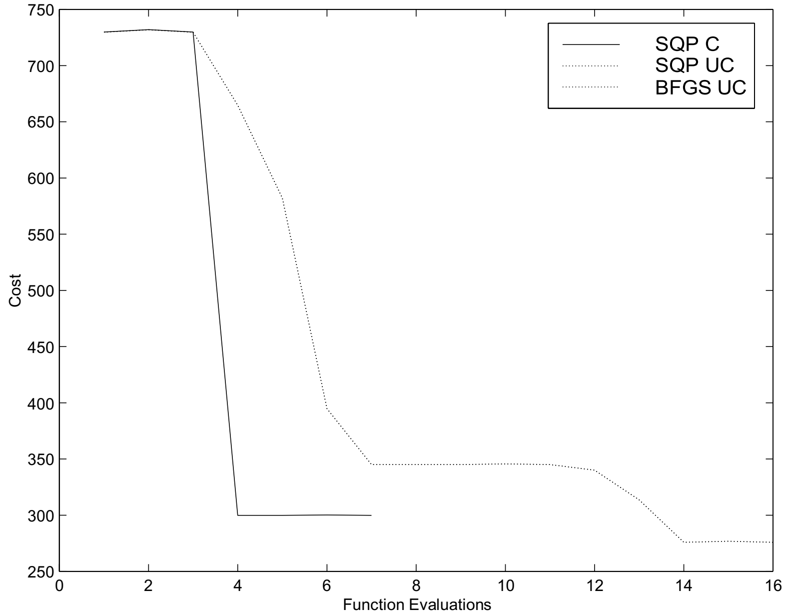

Figure 9. Comparison between Constrained (C), Unconstrained (UC) SQP, and BFGS

The final comparison utilizes a the coordinate pattern search algorithm which is a non-gradient based optimization strategy. This technique is good if a gradient calculation is infeasible or inaccurate. Figure 12 shows the convergence of two different runs. The difference between the runs is the specified initial step size. As can be seen, a too small of step size slows down convergence. The oscillation at the end of the optimization is due to the manner in which the constraints were handled. The CPS method, as implemented in DAKOTA, does not utilize bound constraints which are important in this problem. To enforce these constraints, a penalty function formulation is used. This basically adds a large penalty to the cost function when the bound is violated. When the CPS algorithm gets on the bound, every other step is into the infeasible region and the cost is very large. This discontinuity causes the CPS algorithm to oscillate.

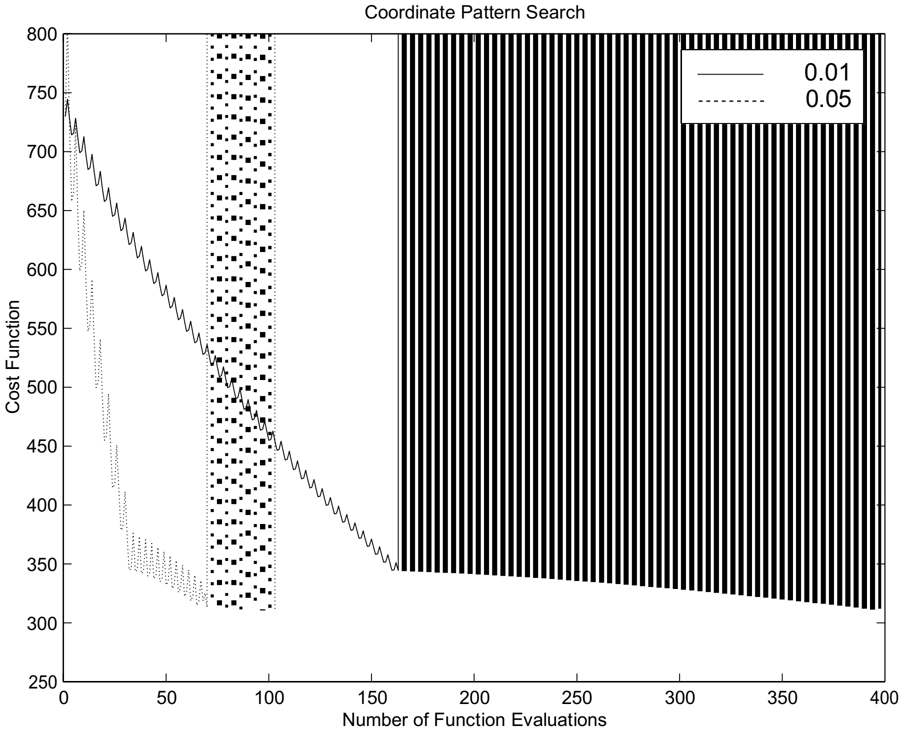

Figure 10. Comparison between CPS with an initial step of 0.01 and 0.05

Conclusions

An optimization on a slot coater was performed using GOMA as the simulation code and DAKOTA as the optimizer. For all these studies a single starting point was used (Gap=0.05 cm, alpha = 0.0 rad, cost = 729). The design space is convex in alpha when the gap becomes close to the bounded optimums of 0.035 cm (see Figure 7). This produces two potential optimums. The first is at (Gap = 0.035 cm, alpha=0.2, cost = 300) which is at a bound for both design variables. The resulting flow is shown in Fig. 13. Note the large recirculation upstream of the coater. A constraint would have to be added in order to eliminate that undesirable feature. From Fig. 7 it can be seen that the gradients over most of the design space are sloping towards this optimum.

The second optimum occurs at (Gap=0.035 cm, alpha=-0.1 rad, cost=276). This optimum is slightly lower in cost that the previous optimum but is only found by one optimizer at the given starting point. The flow from this design is shown in Fig. 14. There is no recirculation in this design.

.

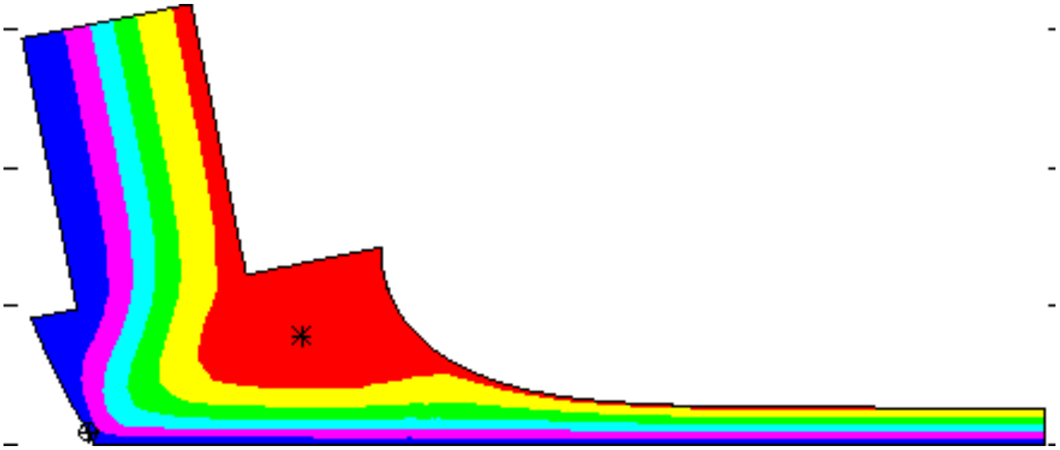

Figure 11. Stream Function for optimal design at Gap=0.035 cm, alpha=0.2 rad (cost=300)

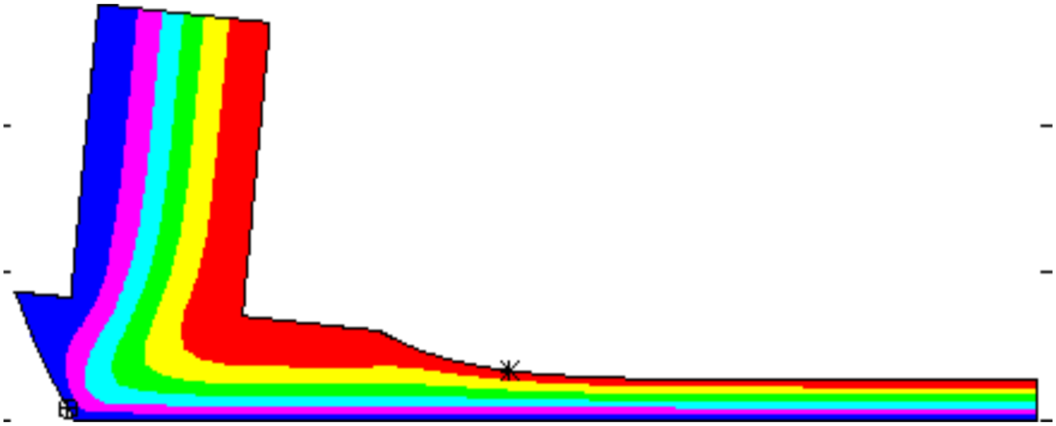

Figure 12. Stream function for optimal design at Gap=0.035 cm, alpha = -0.1 rad (cost =276)