rSQP Optimization of Large-Scale Reacting Flow Applications with MPSalsa

A.G. Salinger[1], R.P. Pawlowski[1], J.N. Shadid[1], B. van Bloemen Waanders[2], R. Bartlett[3], G.C. Itle[3], and L. Biegler[3]

[1] Parallel Computational Sciences Department
[2] Optimization and Uncertainty Quantification Department
Sandia National Laboratories, Albuquerque, NM, USA
[3] Department of Chemical Engineering
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. An rSQP optimization algorithm has been linked to the MPSalsa parallel reacting flows code. This goal is to develop SAND (Simultaneous Analysis aNd Design) methods for use with large-scale PDE simulations. MPSalsa is a unstructured grid finite element code that uses a fully coupled Newton method to solve the PDEs governing fluid flow, heat transfer, and non-dilute mass transfer. In this paper, we present results for optimization of a Chemical Vapor Deposition reactor for growing thin films of Gallium Nitride. In particular, we address issues of inexactness in the Jacobian matrix and of solution multiplicity.

1 Introduction

In this paper we present our current work in applying rSQP (reduced Sequential Quadratic Programming) optimization algorithms to steady state reacting flow applications. This is a SAND (Simultaneous Analysis aNd Design) approach that has the potential to be more efficient than traditional black box (NAND) approaches. The advantage of the SAND approach is that the nonlinear PDE problem (or constraints) is converged simultaneously to the optimization problem, while in the NAND approach, the PDE problem is converged every time the design parameters are changed. The drawbacks of the SAND approach are that the interface of the application code and the optimizer is more intrusive, requiring the ability to request linear solves of the Jacobian matrix for the PDE problem, the residuals of the discretized PDEs for a given state vector, and gradients of the objective function. Another outstanding issue, which we will touch on in this paper, is the level of accuracy in the Jacobian matrix needed for the optimization method to converge to the right solution.

Reacting flow applications are modeled by coupled sets of PDEs, and can have nonlinearities due to convection, reaction rates, and dependency of physical properties on the local state (e.g. calculating density from the ideal gas law). Some examples of reacting flow systems of engineering interest are combustion systems, catalytic reactors, and Chemical Vapor Deposition (CVD) reactors. In this paper we will study a CVD reactor design problem

of interest to researchers in the Chemical Processing Science department at Sandia.

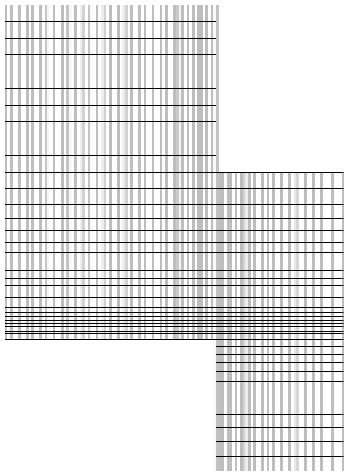
In Section 2 we present the CVD reactor optimization problem. In Section 3.1 the governing PDEs, the solution methods, and the stability analysis algorithms implemented in MPSalsa are presented. In Section 3.2 we briefly present the optimization methods and interface to MPSalsa. In Section 4 the outcome of our preliminary runs are shown, including some interesting results concerning the effects of inexactness in the Jacobian and the effects of solution stability and multiplicity.

2 CVD Reactor Optimization Problem

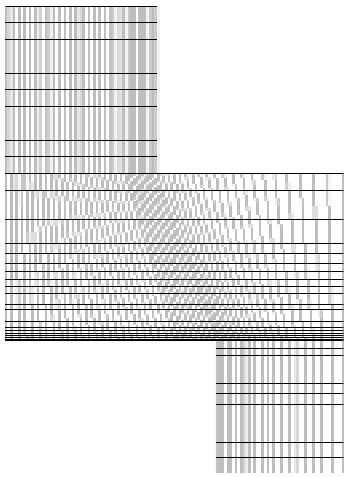
The rotating disk reactor is a common configuration for performing Chemical Vapor Deposition (CVD) of thin films, including numerous semiconducting materials. The optimization problem formulated in this paper is generated from the work of Sandia researchers to improve the design of the inlet of a rotating disk CVD reactor for use in growing thin films of Gallium Nitride (GaN). GaN is used in blue light emitting diodes and other photonic devices. The quality of the electronic device is highly dependent on the uniformity of the growth rate at different positions in the reactor. We are attempting to use simulations and optimization algorithms to determine if a new reactor, designed with a restricted inlet for reducing the costs of reactant gases, can achieve highly uniform GaN film growth.

The finite element mesh for the base shape of the reactor is shown in Figure 1(a). This is an axisymmetric (2D) model, where the left side is the axis of symmetry. A mixture of trimethylgallium, ammonia, and hydrogen gases ($Ga(CH_3)_3$, NH_3, and H_2) enter the top of the reactor, flow over the disk, which is heated, and then flow down the annular region out the bottom of the mesh. At the heated disk, the $Ga(CH_3)_3$ and NH_3 react to deposit a GaN film and release three molecules of methane (CH_4). This simplified mechanism has been shown to work well in approximating film uniformities since the growth rate of GaN is predominantly transport limited Pawlowski et al., 2000. This mesh depicts a restricted inlet design, where the top of the reactor has a smaller radius than the lower part of the reactor.

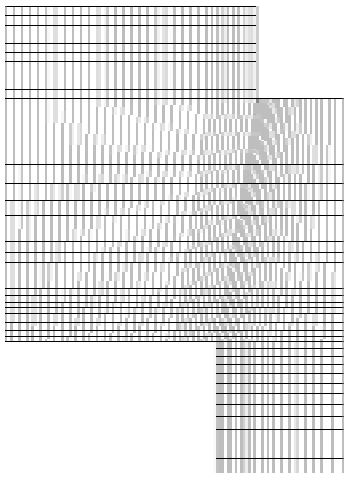
The main parameter used in this paper is the inlet velocity of the gases, V. Two additional parameters in this model define the shape of the inlet, namely the Shoulder Radius and Shoulder Height, which define the position where the mesh transitions from the inlet radius to the larger reactor radius. The mesh is moved algebraically and continuously as a function of these geometric design parameters. Figure 1(b) shows how the mesh changes for a decreased shoulder radius, and Figure 1(c) shows how the mesh deforms continuously for larger values of the shoulder radius and shoulder height. If the optimum occurs too far away from where the initial mesh is generated, it would be appropriate to remesh the new geometry from scratch.



(a)



(b)



(c)

Fig. 1. Three different meshes for the restricted inlet design of the rotating disk reactor are shown: (a) the base case mesh where the shoulder radius is above the edge of the disk and the height is half of the inlet height; (b) a mesh when the shoulder radius parameter is decreased; (c) a mesh where the shoulder radius and height are both increased above the base case.

The objective function measures the uniformity of the growth rate of GaN over the disk. We chose an L^2 norm over an $L^{\inf}$ norm so that it is continuous and with a continuous derivative. Since the L^2 norm had very small values over a range of parameters, the log was taken. The final form of the objective function is

$$\text{Objective Function} = F = log(SD + 10^{-10}) \tag{1}$$

where the standard deviation squared SD is defined as

$$SD = \frac{1}{N_n} \sum_{i=1}^{N_n} (\frac{g_i - g_{ave}}{g_{ave}})^2. \tag{2}$$

Here N_n is the number of nodes on the surface, g_i is the growth rate of GaN at node i, and g_{ave} is the average growth rate.

3 Numerical Methods

3.1 Reacting Flow Simulation

The governing equations and numerical methods summarized in this section have been implemented in the MPSalsa computer code, developed at Sandia National Labs. More complete descriptions of the code and capabilities can be found in the following references [Shadid et al., 1996, Salinger et al., 1996, Salinger et al., 1999b, Pawlowski et al., 2000, Eldred et al., 1996]. The fundamental conservation equations for momentum, heat, and mass transfer are presented for a reacting flow application. The equations for fluid flow consist of a momentum balance (the Navier-Stokes equations) and the total mass balance (continuity equation). The steady-state momentum equation takes the form:

$$\rho(\mathbf{u} \bullet \nabla)\mathbf{u} - \nabla \bullet \mathbf{T} - \rho\mathbf{g} = 0, \tag{3}$$

where $\mathbf{u}$ is the velocity vector, ρ is the mixture density, and $\mathbf{g}$ is gravity vector. $\mathbf{T}$ is the stress tensor for a Newtonian fluid:

$$\mathbf{T} = -P\mathbf{I} - \frac{2}{3}\mu(\nabla \bullet \mathbf{u})\mathbf{I} + \mu[\nabla\mathbf{u} + \nabla\mathbf{u}^T] \tag{4}$$

Here P is the isotropic hydrodynamic pressure, μ is the mixture viscosity, and $\mathbf{I}$ is the unity tensor. The total mass balance is given by:

$$\nabla \bullet (\rho\mathbf{u}) = 0 \tag{5}$$

The density depends on the local temperature and composition via the ideal gas law. For nondilute systems, the multicomponent formulation is used:

$$\rho = \frac{P_o \sum_{j=1}^{N_g} W_j X_j}{RT}, \tag{6}$$

where P_o is the thermodynamic pressure, R is the gas constant, T is the temperature, X_j is the mole fraction of the j^{th} species, W_j is the molecular weight of the j^{th} species, and N_g is the number of gas-phase species (which is 4 for the model in this paper.

The energy conservation equation is given as:

$$\rho \hat{C}_p(\mathbf{u} \bullet \nabla)T = \nabla \bullet (\lambda \nabla T) - S, \tag{7}$$

where $\hat{C}_p$ is the mixture heat capacity and λ is the mixture thermal conductivity. The last term on the right hand side S is the source term due to the heat of reaction, which is negligible under the process conditions in this example problem.

The species mass balance equation is solved for N_g-1 species:

$$\rho(\mathbf{u} \bullet \nabla)Y_k) = \nabla \bullet \mathbf{j}_k + W_k \dot{\omega}_k \quad \text{for } k = 1, \ldots, N_g\text{-}1, \tag{8}$$

where Y_j is the mass fraction of the j^{th} species, $\mathbf{j}_k$ is the flux of species k relative to the mass averaged velocity $\mathbf{u}$ and $\dot{\omega}_k$ is the molar rate of production of species k from gas-phase reactions. A special species equation, which enforces the sum of the mass fractions to equal one, replaces one of the species balances (usually the species with the largest mass fraction):

$$\sum_{k=1}^{N_g} Y_k = 1 \quad \text{for } k = N_g \tag{9}$$

The diffusive flux term (Multicomponent Dixon-Lewis Formulation) includes transport due to both concentration gradients and thermal diffusion (Soret effect):

$$\mathbf{j}_k = \rho Y_k \left(\frac{1}{X_k \overline{W}} \sum_{j \neq k}^{N_g} W_j D_{kj} \nabla X_j - \frac{D_k^T}{\rho Y_k} \frac{\nabla T}{T} \right) \tag{10}$$

Where X_j is the mole fraction of species j, D_{kj} is the ordinary multicomponent diffusion coefficient, and D_k^T is the thermal diffusion coefficient. $\overline{W}$ is the mean molecular weight of the mixture given by:

$$\overline{W} = \sum_{k=1}^{N_g} X_k W_k = \frac{1}{\sum_{k=1}^{N_g} \frac{Y_k}{W_k}} \tag{11}$$

The conversion between mass (Y_k)and mole (X_k) fractions is:

$$Y_k = \frac{W_k}{\overline{W}} X_k \tag{12}$$

At the disk surface, surface chemical reactions take place. In general these can be very complicated, but for this model problem the reaction has been

shown to be approximated very well by a transport limited model. In this case, the growth rate of GaN on the surface (as well as the consumption of $Ga(CH_3)_3$ and NH_3, and the production of CH_4) is proportional to the concentration of trimethylgallium ($Ga(CH_3)_3$) at the surface.

In general, the numerous physical properties in the above equations are dependent on the local temperature and composition. In the MPSalsa code, we use the Chemkin library and database format to obtain these physical properties. These terms add considerable nonlinearity to the problem.

The above system of 9 coupled PDEs (for unknowns $\mathbf{u}_r$, $\mathbf{u}_z$, $\mathbf{u}_\theta$, P, T, $Y_{Ga(CH_3)_3}$, Y_{CH_4}, Y_{NH_3} and Y_{H_2}) are solved with the MPSalsa code. MPSalsa uses a Galerkin/least-squares finite element method to discretize these equations over the spatial domain. While this code is designed for general unstructured meshes in 2D and 3D, and runs on massively parallel computers, this application is 2D, uses the mesh shown in Figure 1(a), and was run on a single processor workstation. The discretized system contains 22000 unknowns.

A fully coupled Newton's method is used to robustly calculate steady-state solutions. While analytic Jacobian entries are supplied for derivatives with respect to the solution variables and the density, derivatives of the other physical properties are only calculated with the numerical Jacobian option. The resulting linear system at each iteration is solved using the Aztec package of parallel, preconditioned iterative solvers. In this paper, we exclusively used an ILU preconditioner and the GMRES solver with no restarts. On a single processor SGI workstation, a typical matrix formulation required 9 seconds for the inexact analytic Jacobian and 96 seconds to calculate the (nearly) exact finite difference numerical Jacobian. A typical linear solve required 40 seconds.

Parameter continuation methods have been implemented in MPSalsa via the LOCA library. These algorithms include an arclength continuation algorithm for tracking solution branches even when they go around turning points. As will be seen in Section 4, this is a powerful tool for tracking out solution multiplicity. In addition, a turning point tracking algorithm has been implemented to directly track out the region of multiplicity as a function of a second parameter. A complementary tool of performing linearized stability analysis by approximating the few rightmost eigenvalues of the linearized time dependent problem has also been successfully implemented [Lehoucq and Salinger, 2001, Salinger et al., 1999a, Burroughs et al., 2001].

3.2 rSQP optimization

The optimization problem was solved by interfacing MPSalsa with the rSQP code from Carnegie Mellon university. Future work will use the updated rSQP++ code. The optimization problem is formulated as follows:

$$\begin{array}{ll} \min & f(y,z) \\ \text{s.t.} & c(y,z) = 0 \\ & x^L \leq x \leq x^U \\ & \mathbf{x} = \begin{bmatrix} y \\ z \end{bmatrix} \\ & x \in R^n, y \in R^m, m \leq O(10^6) \\ & z \in R^{n-m}, n-m = O(1-50) \end{array} \tag{13}$$

where f is the objective function, c are the constraint equations (i.e., the residual vector for the discretized PDE problem), y is the vector of length m of state variables corresponding to the velocity, temperature, pressure, and mass fraction unknowns, and z is a vector of length $n-m$ of decision or optimization variables.

We use the reduced SQP optimization algorithm, as it generally requires the fewest number of function and gradient evaluations and is considered more robust than other optimization methods. The SQP method can be derived from the application of a Newton method to the KKT conditions of (13). At each iteration k, SQP forms and solves a quadratic programming subproblem written as:

$$\begin{array}{ll} \text{Min} & g_k^T d + \frac{1}{2} d^T W_k d \\ & d \in R^n \\ \text{s.t.} & A_k^T d + c_k = 0 \\ & d^L \leq x_k + d \leq d^U \end{array} \tag{14}$$

where g_k is the gradient of the objective function, W_k is the Hessian of the Lagrangian or its approximation, A_k is the Jacobian of the constraints, and d is the step size calculated from the optimization problem. The subscript k indicates that these quantities are calculated at x_k. (For convenience we will suppress this subscript for the remainder of this section.)

If second derivative information is not available then the Hessian matrix is often approximated with quasi-Newton updates. However, under these conditions (14) can be prohitive to solve because either a dense matrix W_k or its quasi-Newton update factors of dimension n must be stored. Instead, we modify the formulation of (14) through a reduced space decomposition. Here the the search direction d is represented by a range space step (Yp_Y) and a null space step (Zp_Z). This decomposition is summarized as follows:

$$\begin{array}{l} A^T = [N|C] \\ d = Yp_Y + Zp_Z \\ \mathbf{Y} = \begin{bmatrix} N^T C^{-T} \\ I \end{bmatrix} \\ \mathbf{Z} = \begin{bmatrix} I \\ -C^{-1} N \end{bmatrix} \end{array} \tag{15}$$

where C is the Jacobian of constraints with respect to the state variable (which is the Jacobian matrix from the PDE model) and N is the matrix of constraint gradients with respect to the design variables. Note that $A^T Z = 0$ and the $Y^T Z = 0$. Also, an alternate choice for the range space basis, $Y^T = [I \mid 0]$ leads to a cheaper (but sometimes less robust) computation if $n - m$ is large.

The vectors p_Y and p_Z are obtained by substituting (15) into (14) to yield a reduced space quadratic programming problem Biegler et al., 1995 that we solve here using a variation of the Goldfarb-Idnani algorithm (called QPKWIK).

$$\begin{array}{ll} \text{Min} & (Z^T g + w)^T p_Z + \frac{1}{2} p_Z^T B p_Z \\ \text{s.t.} & a - z_k \leq p_Z \leq b - z_k \\ & a - z_k - Y p_Y \leq C^{-1} N p_Z \leq b - z_k - Y p_Y \\ & p_Y = -(A^T Y)^{-1} c \\ & B \approx Z^T W Z \qquad (BFGS \text{ approximation}) \\ & x^{k+1} = x^k + d \end{array} \tag{16}$$

The following steps are performed for each iteration of the optimizer.

1. An initial guess of the variables must be supplied to start the algorithm.
2. Calculate f,∇f,c,C,N
3. Calculate $p_Y = (A^T Y)^{-1} c$ using the linear solver
4. Calculate $C^{-1} N$ using a linear solver n-m times
5. Solve the reduced QP using QPKWIK to generate p_Z
6. Calculate d=$Y p_Y + Z p_Z$
7. Are the Karush Kuhn Tucker conditions satisfied? If Yes, then stop
8. Apply a line search to find a stepsize α that satisfies the Armijo conditions.
9. Set $x_{k+1} = x_k + \alpha d$ and $k = k + 1$. Goto step 2

One addition to this algorithm is the option to take second order correction steps. This allows additional iterations of the model equations at each rSQP iteration in order to keep the constraint residuals small. As seen below, in some cases this option can lead to significant improvement in the performance of the optimizer.

4 Results

4.1 One Parameter Model

The first results are shown in Figure 2 for the one parameter system. Here the inlet velocity V is the design parameter while the Shoulder Radius and Shoulder Height parameters are held fixed at 6.35 and 5.08 as in Figure 1(a). Starting at a velocity of $V = 20$ (cm/sec), a simple continuation run

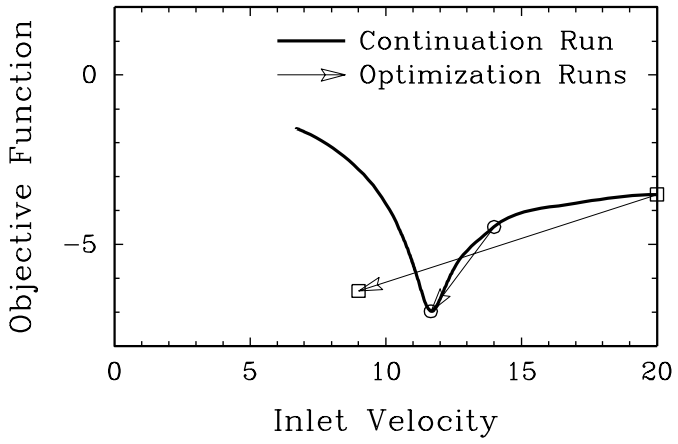


Fig. 2. Results for a 1 parameter continuation run (bold line), showing the Objective Function as a function of the inlet velocity of the reactant gases. Two results for the rSQP optimizer are shown, where the run starting at $V = 14$ (circle symbols with connecting arrow) converged to the expected local minimum while the run starting at $V = 20$ (square symbols with connecting arrow) converged to a point not seen on the continuation run.

down to a velocity of $V = 7$ showed a clear minimum near $V = 11.7$ and Objective Function $F = -6.9$.

Two runs of this problem using the rSQP optimizer were performed. For this run, the exact numerical Jacobian was used, and up to 5 second order correction steps per iteration were allowed. The linear solver tolerance was set at a relative residual reduction of 10^{-8}. When starting at $V = 20$ and converged PDE constraints, the optimizer converged in 15 iterations to a point at $V = 9.00$ and $F = -6.36$ (in about 3 hours compute time). However, when starting at $V = 14$ and with a converged steady-state solution, the optimizer reached the minimum at $V = 11.67$ and $F = -6.967$ in 14 iterations. As can be seen in Figure 2, the first run does not appear to even be on the solution branch of converged PDE constraints.

Three deposition profiles as a function of radial position are shown in Figure 3. The profile at the initial conditions of $V = 20$ has a minimum growth rate at the center and has a 8.5% nonuniformity. The solution found by the optimizer and (appears to be the minimum from the continuation runs) at $V = 11.67$ shows a much flatter profile with an internal maximum, and an overall nonuniformity of 1.2%. The other solution found by the optimizer at $V = 9.00$ has a very similar shape, a smaller overall growth rate, and a 1.8% nonuniformity. Growth rate nonuniformities in the neighborhood of 1.0% are desirable.

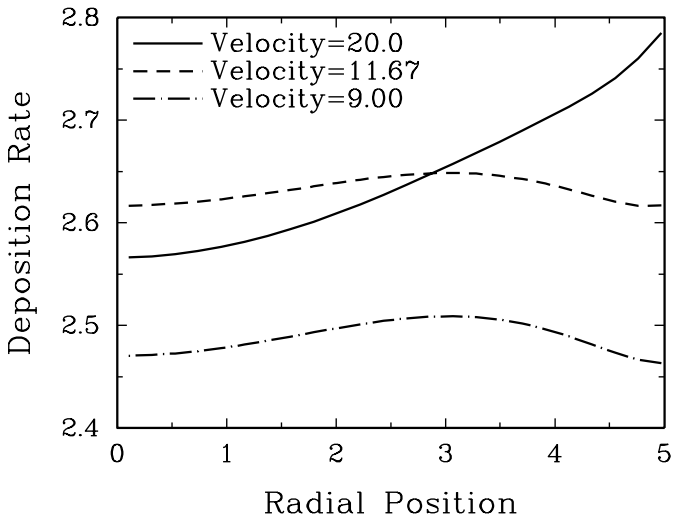


Fig. 3. Radial profiles of the surface deposition rate at three different solutions: the inital guess at $V = 20$, and the final solutions from the two optimization runs at $V = 11.67$ and $V = 9.00$.

Subsequent parameter continuation and linearized stability analysis calculations revealed that this solution is indeed a solution to the PDE constraints, yet a solution that is linearly unstable. The results of an arclength parameter continuation run with linear stability determinations are shown in Figure 4. The dashed line indicated physically unstable solutions while the solid lines are locally stable. One can see that there are three local minima in the objective function, only one of which is linearly stable. Over a large range of inlet velocities, $6.11 < V < 15.86$, there are three solutions that exist at the same parameter values. The rSQP optimizer, when started at $V = 20$, jumped into the basin of attraction for a local minimum at $V = 9.00$. The physical basis for the multiplicity is well understood. Recirculation flow cells can develop as a result of the buoyancy force of the heated reactor surface.

4.2 Three Parameter Model

The one parameter model showed that it is imperative to be aware of solution multiplicity and unstable solution branches. Continuation runs on the turning points defining the boundaries of multiplicity were performed to see how the region of multiplicity changes as a function of the additional geometric parameters. The effect of Shoulder Radius on the multiplicity region is shown in Figure 5, and the effect of Shoulder Height on the region of multiplicity is shown in Figure 6. The results show that the maximum velocity where multiplicity occurs has a direct dependence on the Shoulder Radius and is

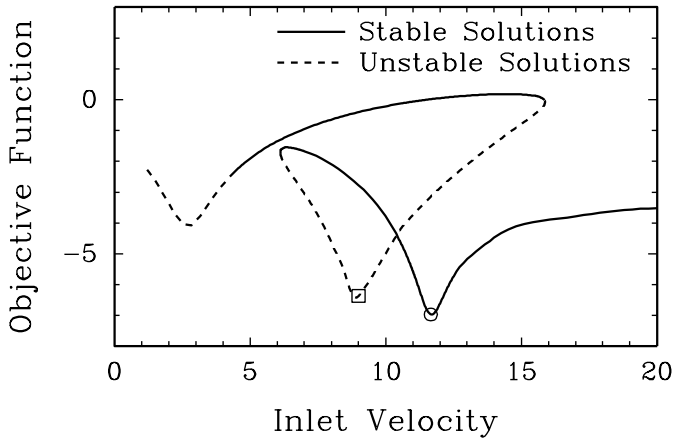


Fig. 4. Results for a 1 parameter continuation run with arclength continuation and linearized stability analysis are shown. The dashed lines represent unstable solution branches. The symbols show the results of the two optimization runs from Figure 2.

relatively insensitive to the Shoulder Height. The minimum velocity where multiplicity occurs is insensitive to the Shoulder Radius but has an inverse dependence on the Shoulder Height.

A single three-parameter optimization run was performed, starting at the same conditions where the one-parameter run (with parameter Velocity and fixed Shoulder Radius and Shoulder Height) converged to the stable minimum: Velocity = 14.0, Shoulder Radius = 6.35, and the Shoulder Height = 5.08. The run was performed with up to 5 second order correction steps per optimization iteration. After 60 iterations, the objective function had been driven down to $F = -6.32$, which is not as low as the $F = -6.967$ achieved in the 1 parameter optimization. Possible reasons for this are that the three-parameter model is converging to a local minimum or that the singularities in the region are causing convergence problems. Future runs will need to be made to fully understand this preliminary result. The result of the three-parameter run is compared to the one-parameter run in Figure 7.

4.3 Effects of Jacobian Inexactness and Second Order Corrections

To test the effects of inexactness in the Jacobian and Second Order Correction Steps on the convergence of the optimization algorithm, three more runs of the 1-parameter model were performed. These all started at $V = 14$ for comparison with the successful optimization run, which was computed with a full numerical Jacobian and up to 5 second order correction steps per iteration.

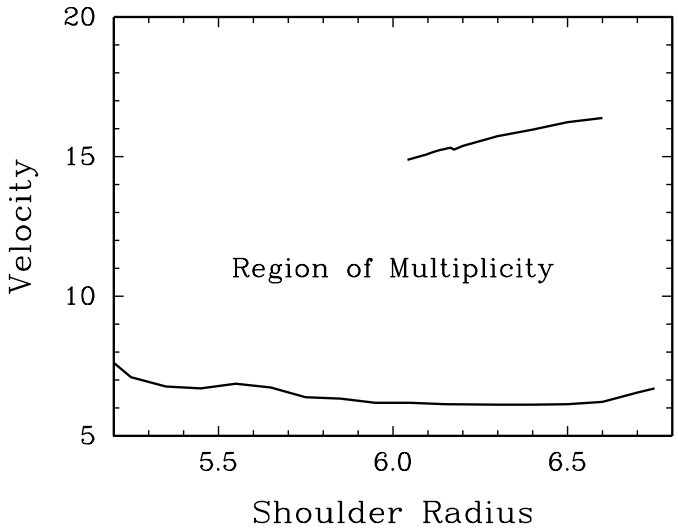


Fig. 5. Results of turning point continuation runs showing how the region of multiplicity identified in Figure 4 changes as a function the geometric Shoulder Radius parameter.

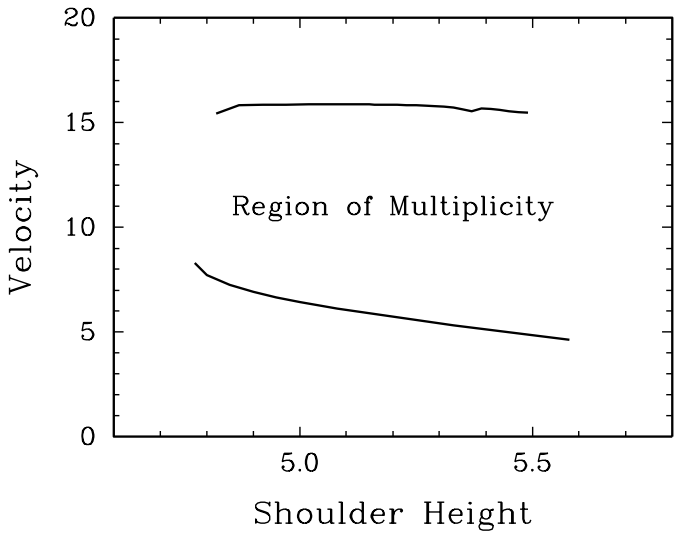


Fig. 6. Results of turning point continuation runs showing how the region of multiplicity identified in Figure 4 changes as a function the geometric Shoulder Height parameter.

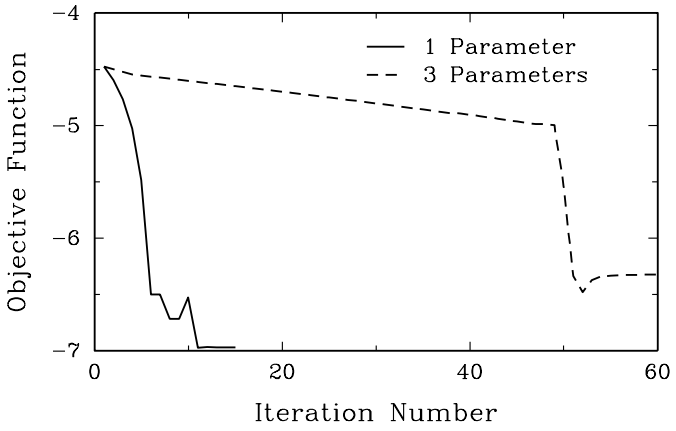


Fig. 7. A comparison of the 3-parameter optimization run after 60 iterations and the 1-parameter run, started at the same conditions, which converged after 14 iterations.

The results are shown in Figure 8. In the first additional run, the analytic (inexact) Jacobian was used, and the second order corrections were retained. This Jacobian leaves out the derivatives of all the physical properties with respect to the local state (temperature and composition), only including the correct density dependence. The Figure shows that this run converges visibly to the same optimum as the original case, both in iteration 11, though the original case reached the optimum in 14 iterations and the inexact case failed to meet the convergence criterion after 40 iterations. Two more runs were performed where no second order correction steps were allowed. The run with the inexact Jacobian converged visibly to the optimum after 86 iterations though had not converged within the tolerance after 100 iterations. The run with the exact numerical Jacobian without second order corrections had not yet converged to the optimum and was prematurely stopped after 120 iterations, surprisingly performing worse than the run with the inexact Jacobian.

For this problem, MPSalsa required 96 seconds to fill the full numerical Jacobian as comnpared to only 9 seconds for the analytic Jacobian, while an iterative linear solve required approximately 40 seconds. The runs with second order corrections required, on average, 5 linear solves per iteration, while the runs without second order corrections required exactly 2 linear solves per iteration. Therefore for this problem, the quickest numerical approach for visibly reaching the optimum was using the inexact analytic Jacobian and with the second order correction steps. The runs with the inexact Jacobian did not trigger the convergence tolerance set in the algorithm, and therefore performed many wasted iterations after visibly reaching the optimum.

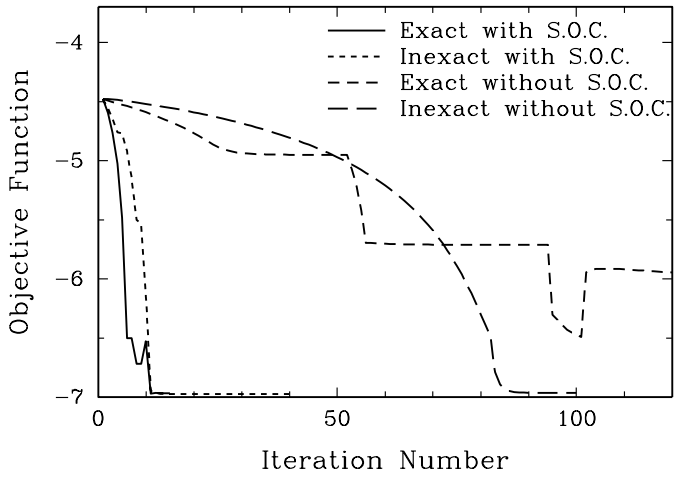


Fig. 8. A comparison of 4 runs for the 1-parameter model, comparing exact and inexact Jacobians, and with and without second order correction steps (S.O.C.).

Since there are numerous approximations in the model, particularly with the chemistry mechanisms, the optimum needs only be converged to two digits of accuracy.

5 Summary and Conclusions

We have successfully coupled an rSQP code with MPSalsa, a large-scale reacting flows code. We used the algorithms to study the restricted inlet design of the rotating disk chemical vapor deposition reactor for growth of thin GaN films. We have verified the optimization algorithms by comparison with a parameter continuation run for a 1-parameter model. We found that solution multiplicity can lead to problems, since the optimizer converged to a local minimum on an unstable solution branch on the very first run. Continuation of the turning points with respect to the additional geometric parameters, using algorithms in the LOCA library, delineated the region of solution multiplicity. The optimization of the full three-parameter model was run for 60 iterations, but had not yet reached convergence.

From this preliminary experience, we can draw an important conclusion: solution multiplicity of nonlinear steady-state problems must be recognized and can be diagnosed using stability analysis tools. The technique in this paper of tracking the region of multiplicity is not scalable to larger numbers of design parameters, and is more expensive than the optimization calculations. At a minimum, the stability of the candidate optimum must be checked with a linear stability analysis tool. Concerning inexactness in the Jacobian

matrix, and the effect of second order correction steps, we have gathered some evidence. For this run, it appears that inexactness in the Jacobian does not seriously hinder convergence, particularly if second order correction steps are used.

References

Biegler, L., Nocedal, J., and Schmid, C. (1995). A reduced hessian method for large-scale constrained optimization. *SIAM J. Opt.*, 5:314.

Burroughs, E. A., Romero, L. A., Lehoucq, R. B., and Salinger, A. G. (2001). Large scale eigenvalue calculations for computing the stability of buoyancy driven flows. *submitted to Journal of Computational Physics*. submitted.

Eldred, M., Hart, W., Bohnhoff, W., Romero, V., Hutchinson, S., and Salinger, A. (1996). Utilizing object-oriented design to build advanced optimization strategies with generic implementation. *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-96-4164-CP, Bellevue, WA*, pages 1568–1582.

Lehoucq, R. B. and Salinger, A. G. (2001). Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. *International Journal for Numerical Methods in Fluids*. in press.

Pawlowski, R. P., Theodoropoulos, C., Salinger, A. G., Mountziaris, T. J., Moffat, H. K., Shadid, J. N., and Thrush, E. J. (2000). Fundamental models of the metalorganic vapor-phase epitaxy of galluim nitride and their use in reactor design. *Journal of Crystal Growth*, 221:622–628.

Salinger, A. G., Devine, K. D., Hennigan, G. L., Moffat, H. K., Hutchinson, S. A., and Shadid, J. N. (1996). MPSalsa: A finite element computer program for reacting flow problems - part II user's guide. Technical report, Sandia National Laboratories, Albuquerque, New Mexico 87185. SAND96-2331.

Salinger, A. G., Lehoucq, R. B., and Romero, L. (1999a). Stability analysis of large-scale incompressible flow calculations on massively parallel computers. In *ISCFD Conference Proceedings*.

Salinger, A. G., Shadid, J. N., Hutchinson, S. A., Hennigan, G. L., Devine, K. D., and Moffat, H. K. (1999b). Analysis of gallium arsenide deposition in a horizontal chemical vapor deposition reactor using massively parallel computations. *Journal of Crystal Growth*, 203:516–533.

Shadid, J. N., Moffat, H. K., Hutchinson, S. A., Hennigan, G. L., Devine, K. D., and Salinger, A. G. (1996). MPSalsa: A finite element computer program for reacting flow problems - Part I theoretical development. Technical report, Sandia National Laboratories, Albuquerque, New Mexico 87185. SAND95-2752.