Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences
(ICIDES-III). Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

OPTIMAL DESIGN OF SOLIDIFICATION PROCESSES

p. 13

Jonathan A. Dantzig and Daniel A. Tortorelli

Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, Illinois 61801 USA

1: INTRODUCTION

An optimal design algorithm is presented for the analysis of general solidification processes, and is demonstrated for the growth of GaAs crystals in a Bridgman furnace. The system is optimal in the sense that the prespecified temperature distribution in the solidifying materials is obtained to maximize product quality. The optimization uses traditional numerical programming techniques which require the evaluation of cost and constraint functions and their sensitivities. The finite element method is incorporated to analyze the crystal solidification problem, evaluate the cost and constraint functions, and compute the sensitivities. These techniques are demonstrated in the crystal growth application by determining an optimal furnace wall temperature distribution to obtain the desired temperature profile in the crystal, and hence to maximize the crystal's quality. A similar problem is investigated by Dantzig and Chao [1], however their approach does not utilize numerical optimization techniques.

Several numerical optimization algorithms are studied to determine the proper convergence criteria, effective one-dimensional search strategies, appropriate forms of the cost and constraint functions, etc. In particular, we incorporate the conjugate gradient and Quasi-Newton methods for unconstrained problems[2]. The efficiency and effectiveness of each algorithm is presented in the example problem.

We have chosen to adapt an existing commercially available finite element program, FIDAP [3], to compute of the sensitivities, rather than develop a new code. Thus, we are in position to investigate larger and more complicated problems in the future without significant code development. The explicit sensitivities are computed analytically by the adjoint technique[4], which has been applied to nonlinear transient conduction problems by Tortorelli *et. al.* [5]. Large computational savings and accurate calculations are realized by utilizing an explicit approach as opposed to the costly and sometimes unreliable finite difference method [5, 6].

In the following section, a brief outline of the conjugate gradient and quasi-Newton methods for unconstrained optimization are presented. In section 3, the adjoint sensitivity method is reviewed and presented in a specialized form appropriate to the processing problem. An example problem is presented in the last section.

2: METHODS OF ANALYSIS

2.1 Optimization Algorithms

Vanderplaats [2] presents an excellent exposition of the algorithms which have been developed to resolve design optimization problems. The search method algorithms for unconstrained problems are characterized by sequential searches in the design space to reduce the value of the objective function, G. Beginning at a specified initial point in design space **b**, a line search is performed to find the minimum value of the objective in a search direction, S. Once the minimum is found in this direction, the present design is updated and a new search direction is chosen. This process is repeated until the design converges to its minimum objective function value. In this section we will briefly outline three search methods for unconstrained optimization. The methods are distinguished by the manner in which the sequence of search directions is determined.

Search methods which utilize derivatives of the objective function tend to be more efficient (*i. e.* will require fewer iterations) than zero-order methods. This is true because the gradients suggest the direction one should move in design space to reduce the value of the objective function. The sensitivity analyses, described in the next section, provide this gradient information at relatively little additional cost beyond that which is required to analyze the process and evaluate the objective function. Accordingly, the discussion here is limited to these first-order gradient-based methods, specifically, the methods of steepest decent, Fletcher- Reeves conjugate gradient, and the Quasi-Newton are described. The methods differ in the way that the search directions are determined.

Line searches are performed for all of the above-described algorithms. We are using a variant of Brent's Method for this purpose[7]. In this technique, the objective function is assumed to vary quadratically with the scalar α along the vector in design space given by **b** + αS. Thus, the problem becomes one of finding the value of α corresponding to the minimum G. If G were truly quadratic in α, then a combination of three function evaluations or derivatives with respect to α would suffice to obtain the minimum. In practice, G is generally not quadratic in α, hence this technique requires repeated evaluations of G and its derivatives to determine the minimum. In some cases, the parabolic interpolation can diverge. To circumvent this problem, Brent's Method uses interval sectioning when divergence of the parabolic interpolation is detected.

Once the minimum for the given search direction is found, a new direction must be chosen. The most simplistic algorithm uses the gradient to determine the new search direction, *i. e.*

$$\mathrm{S} = -\nabla G \tag{1}$$

This "Method of Steepest Descent" has been shown to be inefficient [2]. Better algorithms utilize information about previously searched directions to construct the next search vector. In the Fletcher-Reeves conjugate gradient method, the new search direction is given by

$$\mathrm{S}^j = -\nabla G(\mathbf{b}^j) + \frac{|\nabla G(\mathbf{b}^j)|^2}{|\nabla G(\mathbf{b}^{j-1})|^2}\mathrm{S}^{j-1} \tag{2}$$

where $\mathbf{b}^j$ is the design vector at the beginning of the j^{th} line search. Such a selection of the S ensures that the search directions are Q-orthogonal. After several steps, it is possible that searching in direction S^j will not improve the objective, and the process is then re- initialized with Equation (1).

In the Quasi-Newton methods for unconstrained minimization, we consider a Taylor series expansion of G about the present design, $\mathbf{b}_0$.

$$G(\mathbf{b}) \approx G(\mathbf{b}_0) + \nabla G \cdot \delta\mathbf{b} + \frac{1}{2}\delta\mathbf{b}^{\mathrm{T}}\mathrm{H} \cdot \delta\mathbf{b} \tag{3}$$

where $\mathbf{H}$ is the Hessian matrix, and $\delta\mathbf{b} = \mathbf{b} - \mathbf{b}_0$ is the new search direction. Differentiating this equation with respect to the design variation and setting the result to zero (for the minimum) yields

$$\delta\mathbf{b} = -\mathbf{H}(\mathbf{b}_0)^{-1}\nabla G(\mathbf{b}_0) \tag{4}$$

Rather than compute the Hessian inverse, which is usually difficult because it contains second-order sensitivity information, we construct a series of approximations to $\mathbf{H}^{-1}$ from

$$\begin{aligned} (\mathbf{H}^{-1})^1 &= \mathbf{I} \\ (\mathbf{H}^{-1})^{j+1} &= (\mathbf{H}^{-1})^j + \mathbf{D}^j \end{aligned} \tag{5}$$

where

$$\mathbf{D}^j = \frac{\delta\mathbf{b}\cdot\delta\mathbf{b}}{\delta\mathbf{b}\cdot\delta(\nabla G)} - \frac{(\mathbf{H}^{-1})^j(\mathbf{H}^{-1})^{jT}}{\delta(\nabla G)^T(\mathbf{H}^{-1})^j\delta(\nabla G)} \tag{6}$$

This approximation to the Hessian inverse (Equation (5)) is then used with Equation (4) to determine the appropriate design increment.

A more detailed discussion of these algorithms is given in Reference [2]. Clearly, the use of these algorithms requires that the sensitivities be computed accurately, and because they are computed many times (once per design iteration), they must also be computed efficiently. In the next section, an efficient algorithm is described for obtaining the sensitivities after analyzing the original problem.

2.2 Explicit Design Sensitivity Analysis using an Adjoint Method

Tortorelli, *et al.*[5] described a Lagrange multiplier method for formulating the adjoint design sensitivities for nonlinear transient thermal systems. The variation of a general design functional may be expressed in explicit form with respect to variations in the prescribed boundary conditions. However, the design functional depends on these explicit quantities *and* implicitly on the temperature field. To obtain the explicit sensitivities, the implicit dependency on the temperature field must be resolved.

The design functional is expressed as

$$G(\mathbf{b}) = \int_B f(T)dV + \int_{\partial B} g(T,\mathbf{b})dA \tag{7}$$

where the temperature $T(\mathbf{x},\mathbf{b})$ represents the implicit response fields in G, $\mathbf{b}$ is the vector of design parameters, and the position vector is denoted by $\mathbf{x}$. The design vector will be used to define the boundary conditions, which ultimately control the values of all the response quantities and G. All quantities are defined in the region B or on the bounding surface ∂B (with outward unit normal vector $\mathbf{n}$), and are assumed to be smooth enough to justify the operations performed. Furthermore, differentiability of G with respect to the design is assumed.

The response quantities are implicitly defined by the design and the following mixed boundary value problem

$$\nabla\cdot\mathbf{q} + r = 0 \quad in\ B \tag{8}$$

with boundary conditions

$$\begin{aligned} T &= T^P \quad on\ A_T \\ q^s &= q^p(T,\mathbf{b}) \quad on\ A_q \\ q^s &= h(T,\mathbf{b})(T - T_\infty(\mathbf{b})) \quad on\ A_h \end{aligned} \tag{9}$$

where $\mathbf{q}(\mathbf{x},\mathbf{b})$ is the heat flux vector, $r(T,\mathbf{g},\mathbf{x},\mathbf{b})$ represents the temperature– and temperature gradient–dependent internal heat generation, $\mathbf{g}(\mathbf{x},\mathbf{b}) \equiv \nabla T(\mathbf{x},\mathbf{b})$ is the temperature gradient, $q^s \equiv \mathbf{q}\cdot\mathbf{n}$ is the surface heat flux, and $h(T,\mathbf{b})$ is a convective transport coefficient between the surface of the domain and the ambient temperature, $T_\infty(\mathbf{b})$. A_T, A_q and A_h are complementary subsurfaces of ∂B and correspond to surfaces with prescribed temperature T^p, prescribed flux q^p, and prescribed convective boundary conditions, respectively. Note that the prescribed flux, heat transfer coefficient and ambient temperature distribution are all functions of the design vector, $\mathbf{b}$, and the temperature to allow modeling of nonlinear heat flux and convective loads, as well as radiation. Note also that the internal heat generation term may be used to model convective transport terms when fluid flow is present. To complete the representation, a constitutive relation is introduced for the heat flux where $\hat{\mathbf{q}}$ is a general function of the position, temperature and temperature gradient,

$$\mathbf{q} = \hat{\mathbf{q}}(\mathbf{x},T,\mathbf{g}) \tag{10}$$

We will follow a finite element formulation, where Equation (8) is written in weak form and the boundary conditions in Equation (9) enter after integrating by parts and applying the divergence theorem.[8] First define the weighted residual, R, as

$$R(T,\mathbf{b},\lambda) \equiv -\int_B \{\nabla\lambda\cdot\mathbf{q} - \lambda r\}dV + \int_{A_q} \lambda q^p dA + \int_{A_h} \lambda h(T - T_\infty)dA \tag{11}$$

where λ is a weighting function which will described in more detail below. We use a *displacement* approach, in which the only dependent field is the temperature. Thus, Equations (9) and (10) are strictly enforced, and $\lambda = 0$ on A_T. When R is equal to zero (for all admissable λ) then Equation (8) is satisfied.

In general, the nonlinear nature of the problem will require that Newton-Raphson iteration be performed to find the zero of the residual. We introduce a truncated Taylor series expansion to update the temperature field from T^I at iteration I to T^{I+1} at iteration $I+1$:

$$R^{I+1} \approx R(T^I,\mathbf{b},\lambda) + \left\{\frac{\partial R(T^I,\mathbf{b},\lambda)}{\partial T}\right\}\Delta T = 0 \tag{12}$$

where $\Delta T \equiv T^{I+1} - T^I$ and

$$\begin{aligned}\frac{\partial R(T^I,\mathbf{b},\lambda)}{\partial T}\Delta T = &-\int_B \left\{\nabla\lambda\cdot\frac{\partial\hat{\mathbf{q}}}{\partial T}(\Delta T) + \nabla\lambda\cdot\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{g}}\nabla(\Delta T) - \lambda\frac{\partial r}{\partial T}\Delta T - \lambda\frac{\partial r}{\partial\mathbf{g}}\nabla(\Delta T)\right\}dV + \\ &\int_{A_q}\lambda\frac{\partial q^p}{\partial T}\Delta T dA + \int_{A_h}\lambda\left[\frac{\partial h}{\partial T}(T - T_\infty) + h\right]\Delta T dA\end{aligned} \tag{13}$$

In finite element analyses, R and $\frac{\partial R}{\partial T}$ form the residual vector and tangent stiffness matrix, respectively. The incremental problem given in Equation (12) is solved iteratively until the solution converges.

As we described earlier, changes in $\mathbf{b}$ affect the boundary conditions, which in turn affect the response quantities, which ultimately alter the value of the response functional G. The objective of sensitivity analysis then, is to derive an explicit expression for ∇G in which only variations of the design parameters $\delta\mathbf{b}$, are present.

In the Lagrange multiplier method for the adjoint sensitivity analysis, the residual is adjoined to G to define an augmented functional G^*,

$$G^* = \int_B f dV + \int_{\partial B} g dA - \int_B (\nabla\lambda\cdot\hat{\mathbf{q}} - r)\lambda dV + \int_{A_q}\lambda q^p dA + \int_{A_h}\lambda h(T - T_\infty)dA \tag{14}$$

This ensures that the governing equations are satisfied. In this equation, λ can be interpreted as a Lagrange multiplier, which will eventually be identified as the temperature field of a second, fictitious *adjoint* problem defined over B. Note that since the augmented term and its derivative are both identically zero, $G^* = G$ and $\nabla G^* = \nabla G$.

Formal differentiation of Equation (14) with respect to the design vector gives

$$\begin{aligned}\frac{dG}{d\mathbf{b}} =& \int_B \frac{\partial f}{\partial T}\frac{\partial T}{\partial \mathbf{b}} dV + \int_{\partial B} \left(\frac{\partial g}{\partial \mathbf{b}} + \frac{\partial g}{\partial T}\frac{\partial T}{\partial \mathbf{b}}\right) dA - \\ & \int_B \left(\nabla\lambda \cdot \frac{\partial \mathbf{q}}{\partial T}\frac{\partial T}{\partial \mathbf{b}} + \nabla\lambda \cdot \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{g}}\frac{\partial \nabla T}{\partial \mathbf{b}} - \lambda\frac{\partial r}{\partial T}\frac{\partial T}{\partial \mathbf{b}} - \lambda\frac{\partial r}{\partial \mathbf{g}}\frac{\partial \nabla T}{\partial b}\right) dV + \int_{A_q} \lambda\left(\frac{\partial q^p}{\partial T}\frac{\partial T}{\partial \mathbf{b}} + \frac{\partial q^p}{\partial \mathbf{b}}\right) dA + \\ & \int_{A_h} \lambda\left(\left[\frac{\partial h}{\partial T}\frac{\partial T}{\partial \mathbf{b}} + \frac{\partial h}{\partial \mathbf{b}}\right](T - T_\infty) + h\left[\frac{\partial T}{\partial \mathbf{b}} - \frac{\partial T_\infty}{\partial \mathbf{b}}\right]\right) dA\end{aligned} \qquad (15)$$

With the exception of λ and the implicit terms involving the derivatives of T and ∇T with respect to b, all of the terms in Equation (15) are known once the original analysis problem is solved. In the sensitivity analysis, we will eliminate the implicit terms by a particular choice of the Lagrange multiplier λ.

To this end, we separate ∇G^* into terms which explicit quantities, ∇G^*_E, and those which are implicit quantities∇G^*_I, where

$$\nabla G^*_E = \int_{\partial B} \frac{\partial g}{\partial \mathbf{b}} dA + \int_{A_q} \lambda\frac{\partial q^p}{\partial \mathbf{b}} dA + \int_{A_h} \lambda\left(\frac{\partial h}{\partial \mathbf{b}}(T - T_\infty) - h\frac{\partial T_\infty}{\partial \mathbf{b}}\right) dA \qquad (16)$$

and

$$\begin{aligned}\nabla G^*_I =& \overbrace{\left\{\int_B \frac{\partial f}{\partial T}\frac{\partial T}{\partial \mathbf{b}} dV + \int_{\partial B} \frac{\partial g}{\partial T}\frac{\partial T}{\partial \mathbf{b}} dA\right\}}^{\frac{\partial G}{\partial T}\cdot\frac{\partial T}{\partial \mathbf{b}}} - \\ & \int_B \left(\nabla\lambda \cdot \frac{\partial \hat{\mathbf{q}}}{\partial T}\frac{\partial T}{\partial \mathbf{b}} + \nabla\lambda \cdot \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{g}}\frac{\partial \nabla T}{\partial \mathbf{b}} - \lambda\frac{\partial r}{\partial T}\frac{\partial T}{\partial \mathbf{b}} - \lambda\frac{\partial r}{\partial \mathbf{g}}\frac{\partial \nabla T}{\partial \mathbf{b}}\right) dV + \\ & \int_{A_q} \lambda\frac{\partial q^p}{\partial T}\frac{\partial T}{\partial \mathbf{b}} dA + \int_{A_h} \lambda\left(\frac{\partial h}{\partial T}\frac{\partial T}{\partial \mathbf{b}}(T - T_\infty) + h\frac{\partial T}{\partial \mathbf{b}}\right) dA\end{aligned} \qquad (17)$$

where $\frac{\partial T}{\partial \mathbf{b}} = 0$ on A_T. On examination of Equation (17) and Equations (8) and (9) we note that the implicit term can be annihilated by solving the following adjoint problem: Find that value of λ for which

$$-\frac{\partial G}{\partial T} = \frac{\partial R(T, \mathbf{b}, \lambda)}{\partial T}\frac{\partial T}{\partial \mathbf{b}} \qquad (18)$$

for all admissible $\frac{\partial T}{\partial \mathbf{b}}$. Note that $\frac{\partial G}{\partial T}$ is the indicated quantity in Equation (17). This equation is linear in λ, and is the adjoint operator for the incremental problem (Equation (12)). This allows us to solve the adjoint problem efficiently when the finite element method is used.

Indeed, after solving the original problem with Newton-Raphson iteration, we next store the final decomposed tangent stiffness matrix. Then the adjoint load vector ($\frac{\partial G}{\partial T}$) is formed which corresponds to the following adjoint loads:

$$
\begin{aligned}
r_\lambda &= \frac{\partial f}{\partial T} \quad in\ B \\
\lambda &= 0 \quad on\ A_T \\
q_\lambda^s &= \frac{\partial g}{\partial T} \quad on\ A_q \\
q_\lambda^s &= \left(\frac{\partial h}{\partial T}(T - T_\infty) + h\right)\lambda + \frac{\partial g}{\partial T} \quad on\ A_h
\end{aligned}
\qquad (19)
$$

Finally, we perform a back substitution on the transpose (adjoint) of the decomposed stiffness matrix to evaluate λ. Once λ is determined, then $\nabla G_I^* \equiv 0$ and the sensitivities are obtained directly from Equation (16). The efficiency of this method lies in the fact that a single back-substitution into the already decomposed stiffness matrix, followed by substitution in Equation (16), yields all of the components of the sensitivity vector. In general, the solution of the primal problem requires several Newton-Raphson iterations. Hence, the added cost of evaluating the sensitivities is relatively small.

In the finite element evaluation of the adjoint load vector and ∇G_E^*, the same numerical quadrature is used as that used to evaluate G, the tangent stiffness matrix and residual. This ensures that consistent results are obtained. In the next section, these methods will be used in an example problem concerning Bridgman crystal growth.

3: APPLICATION TO A CRYSTAL GROWTH PROCESS

3.1 Bridgman Crystal Growth

When crystals for electronics applications are grown using the Bridgman process, the finished bulk crystals are sliced into thin wafers perpendicular to the growth direction. Electronic devices are then fabricated on these wafers. The properties of the devices are highly dependent on the degree of perfection and compositions of the wafer. Since these attributes are set during growth of the crystal, control of the growth process is vital. In particular, fluid flow in the melt during solidification can interact with the solute field near the crystal-melt interface to adversely affect the chemical composition of the crystal.[9] The primary means for controlling the convective flow is to control the shape of the crystal-melt interface, which may be accomplished by defining appropriate process parameters.

The latest generation of Bridgman furnaces are divided into several independent heating zones along their length, so that complex temperature distributions can be applied. While this gives these furnaces great flexibility, it also necessitates that detailed analyses be performed to relate the temperatures imposed on the furnace wall to the temperature distribution produced in the crystal.

Using the techniques described in the preceding sections, a model is presented for determining the optimal temperature distribution to impose on the furnace wall to produce the desired temperature distribution in the crystal. In particular, the desired temperature distribution in the crystal becomes the objective, and the temperatures on the furnace wall comprise the design parameters. In our example problem, the furnace to be examined is one that will be used in low gravity space processing.

A configuration proposed by researchers at GTE for growing GaAs crystals in space is illustrated in Figure 1.[10] In the proposed experiment, a round pyrolytic boron nitride crucible with graphite end plugs and a quartz bottom is used to contain a GaAs charge. The entire container is to be filled on earth, then sent into space, where it will be placed in a programmable gradient furnace, melted and resolidified in a controlled manner. The geometry

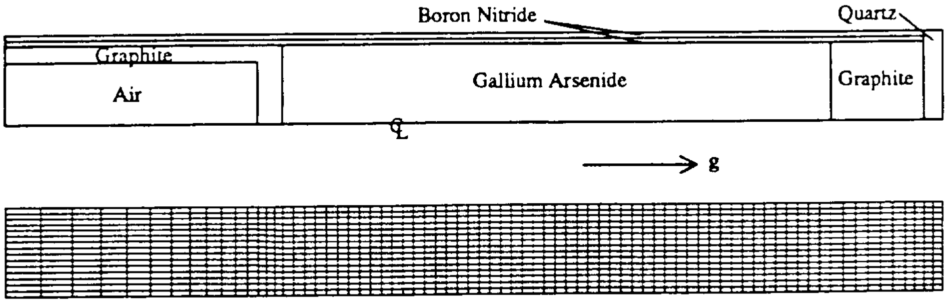

Fig. 1: Schematic view of the model for the proposed experiment to grow GaAs crystals, and corresponding finite element mesh containing 1216 nodes and 1230 elements.

of the experimental apparatus was exploited to describe the process using a two-dimensional axisymmetric finite element model. It will be assumed that the ampoule is maintained with its axis parallel to the gravity vector.

The commercial code FIDAP[3], with modifications to enable the design sensitivities to be calculated, was used for the analysis. The container and melt were modeled using four-noded linear isoparametric elements, whereas the presence of the furnace wall was represented by a specified temperature distribution exchanging heat by radiation with the exterior surface of the ampoule. Further details of the radiation calculation are given below.

The governing equations and boundary conditions for these types of problems are well established[11], and are reproduced here only to the extent necessary for the present discussion. In addition to the energy balance equation, we must consider the momentum balance equation to model buoyancy-driven convection in the crystal. The density was assumed to be constant, except for thermal expansion in the liquid phase, which is included by the Boussinesq approximation. With this assumption, the steady form of the momentum balance equation is

$$\rho_0(\mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \mu\nabla^2\mathbf{u} + \rho_0\mathbf{g}_o(1-\beta(T-T_{ref})) \tag{20}$$

where $\mathbf{u}$ is the velocity, p is the pressure, μ is the dynamic viscosity, $\mathbf{g}_o$ is the gravity vector, β is the volumetric thermal expansion coefficient, and T_{ref} is the temperature at which the density is ρ_0 (in this case, the melting temperature). Note that the presence of the buoyancy term couples the momentum balance equations to the energy balance equation. There is no slip of the liquid past the solid, so that the velocity of the fluid is zero at all of the boundaries of the melt.

The steady form of the energy balance equation, adopting Fourier's Law ($\hat{\mathbf{q}} = -k(\mathbf{x},T)\mathbf{g}$) for the constitutive relation for heat flux, is given by

$$\rho_0 c_p(\mathbf{u}\cdot\nabla T) = \nabla\cdot(k\nabla T) \tag{21}$$

where c_p is the specific heat and k is the temperature dependent thermal conductivity. The advection term on the left-hand side of this equation defines the internal heat generation term, r, noted above. Heat is conserved at the crystal-melt interface, requiring that

$$k_s\nabla T_s\cdot\mathbf{n} - k_l\nabla T_l\cdot\mathbf{n} = 0 \tag{22}$$

where $\mathbf{n}$ is a unit vector normal to the interface and the subscripts l and s refer to the liquid and solid phases, respectively. For very dilute alloys, the interface temperature can be assumed to be the melting temperature of the parent phase, denoted T_m.

Even though the problem is steady, latent heat can be convected by the fluid flow in the melt. Latent heat evolution was included in the model using an enthalpy-specific heat method.[12] This method requires that the enthalpy of solidification be spread over a range of temperatures, and in all of the simulations which follow, this interval was chosen to be 1K. The specific heat was computed from the gradients of enthalpy and temperature at each element integration point

$$c_p = \sqrt{\frac{\nabla H \cdot \nabla H}{\nabla T \cdot \nabla T}} \tag{23}$$

and assembled using a lumped mass matrix formulation. These formulations ensure that the entire heat content of the material is accounted for in a computationally efficient way.[12]

The ends of the ampoule were considered to be insulated. Heat was transferred between the ampoule and the furnace wall by radiation only. The ampoule was assumed to fit closely in the furnace, so that radiation exchange was limited to opposing faces in the furnace, *i.e.* no view factor calculations were required. The Stefan-Boltzmann law was factored, so that a nonlinear convection coefficient, h_{eff}, was defined for each integration point

$$\begin{aligned} q_{rad} &= \sigma\varepsilon\left(T^4 - T_{furnace}^4\right) \\ &= \underbrace{\sigma\varepsilon\left(T^2 + T_{furnace}^2\right)\left(T + T_{furnace}\right)}_{h_{eff}}\left(T - T_{furnace}\right) \end{aligned} \tag{24}$$

For all cases, the emissivity was taken to be constant at 0.7. The material properties used in the simulations are given in the Appendix.

Pressure was eliminated as a degree of freedom using a penalty method.[13] In this formulation, the continuity equation for an incompressible fluid is modified to allow an artificial compressibility, so that

$$\nabla \cdot \mathbf{u} = -\varepsilon_p p \tag{25}$$

where ε_p is a penalty parameter, taken to be 1×10^{-8} in all cases. The resulting coupled nonlinear equations for the velocities and temperatures were resolved at each time step by Newton-Raphson iteration or by successive substitution. Convergence was declared when both the rms change in each field variable and the residual errors in the finite element equations fell below 1×10^{-3}.

The temperature distribution for a constant temperature gradient of 5 K/mm along the furnace wall was known to produce significant undesirable curvature of the crystal-melt interface.[1] The primary reason for is the variation in thermal conductivity between the liquid and solid. Thus, a constant temperature gradient results in unequal heat fluxes at the interface. (See Equation (22).) To alleviate this problem, the temperatures applied along the furnace wall will be adjusted to produce a specified temperature distribution within the crystal. The procedures developed in the previous sections were used for this purpose.

3.2 Implementation of Design Sensitivity Analysis

The commercial finite element code FIDAP[3] was modified to perform the adjoint load and sensitivity calculations described in the previous section. The sensitivity calculations neglected the fluid velocities and the coupling to the momentum equations. However, the optimization still converged in an acceptable number of iterations because the problem is dominated by the thermal aspects. A shell program was then written to coordinate

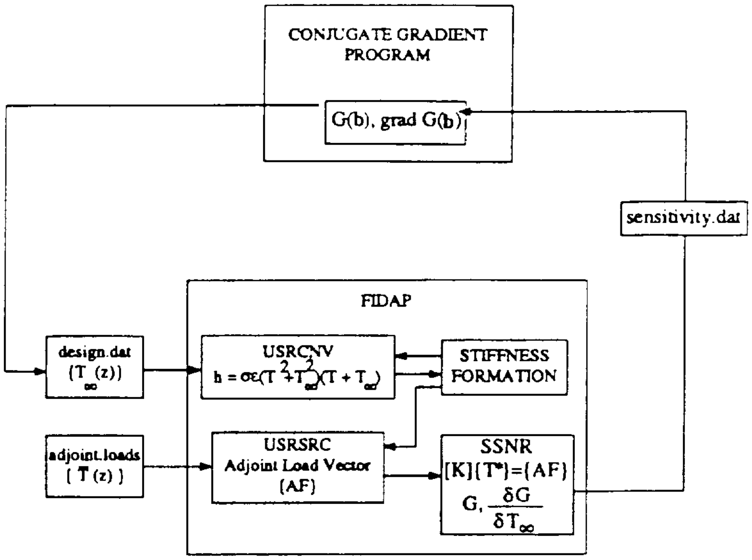

Fig. 2: Schematic of link between conjugate gradient search algorithm and FIDAP.

the nonlinear heat transfer analysis, the sensitivity analysis, and the numerical optimization. A schematic diagram showing the details of the interface between the shell program and FIDAP is shown in Figure 2.

The file *adjoint.loads* in Figure 2 contained the information describing the desired temperature profile, $\overline{T}(z)$, to be attained in the ampoule. This distribution was specified on both the center-line and the outer radius of the crystal (inner radius of the ampoule). The objective function was then defined as the error between the desired and computed temperatures at N discrete points

$$G = \sum_{i=1}^{N} (T_i - \overline{T}_i)^2 \tag{26}$$

Thus, G represents the function to be minimized.

The only design variables allowed in the problem were the furnace wall temperatures, $T_\infty(z)$. Note, however, that in view of Equation (24) there is an implicit dependence of the heat transfer coefficients on $T_\infty(z)$ which must be accounted for.

The progress of the optimization is illustrated in Figures 3 – 5. It is easy to see that the search through the design space converges quickly to the optimal solution. For this case, the ambient temperature at each position on the furnace wall opposite each surface node on the ampoule comprised the 76 design variables. The fact that there are so many design degrees of freedom leads to the unrealistic fluctuations seen in the furnace wall temperature profile.

This same case was then modeled using nine zones to span the entire length of the furnace. The ten specified wall temperatures represent the design parameters, and the intermediate wall temperatures were determined via linear interpolation. The results for this case are shown in Figure 6. It can be seen that equivalent results are obtained for the internal temperature. It is interesting to note, however, that the results for the latter case are not simply an average of the results from the former.

Notice that in all of these cases, the sudden changes in slope in the objective function led to sharp changes in the furnace wall temperature profile and that the ampoule temperature was unable to capture the sudden change. Accordingly, a new objective function was defined which maintained the discontinuity in slopes at the crystal-melt

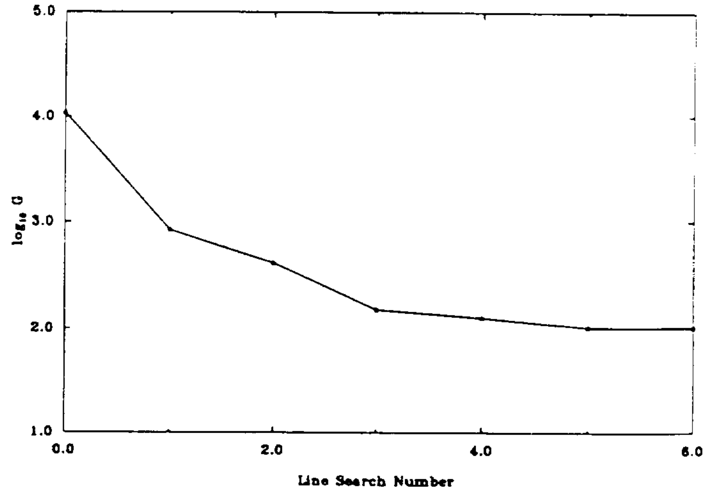

Fig. 3: Progress of the value of the objective function during the optimization

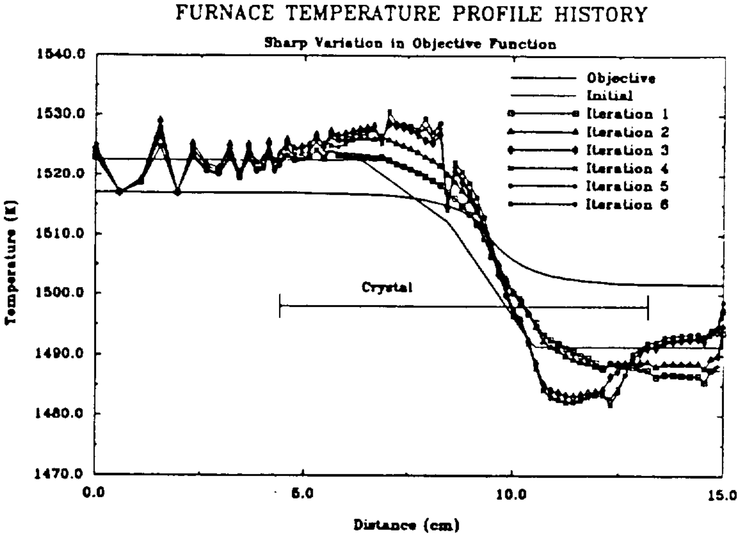

Fig. 4: Progress of the furnace temperature profiles during the optimization

interface, and rolled off exponentially with distance from the interface. The results, shown in Figure 7, illustrate that one may attain the final objective, if the physics of the problem allows it. This is the nature of optimization, where existence and uniqueness of solutions is not always guaranteed.

For each case, the progress through the numerical optimization was very similar. Five to ten line searches were required, with six to eight function evaluations along each line. This latter number was found to be very sensitive to the convergence tolerance for the parabolic interpolation. Setting the tolerance below 0.01 resulted in many more function evaluations with no improvement in the overall results. The problems ran to completion in about one hour on a Sun SPARCstation 1+.

A quasi-Newton method was also used, but for this problem the results were almost identical. The quasi-Newton procedure typically required one more line search than the conjugate gradient method, but there were not enough tests done to draw any definitive conclusions.

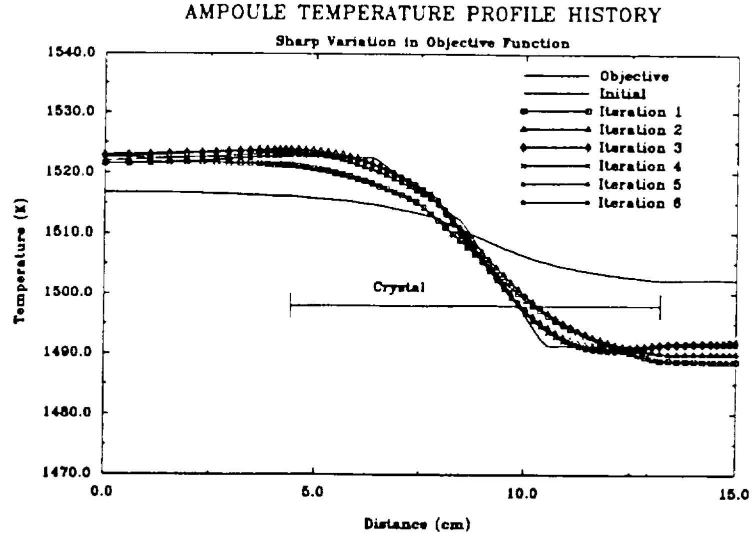

Fig. 5: Progress of the ampoule temperature profiles during the optimization

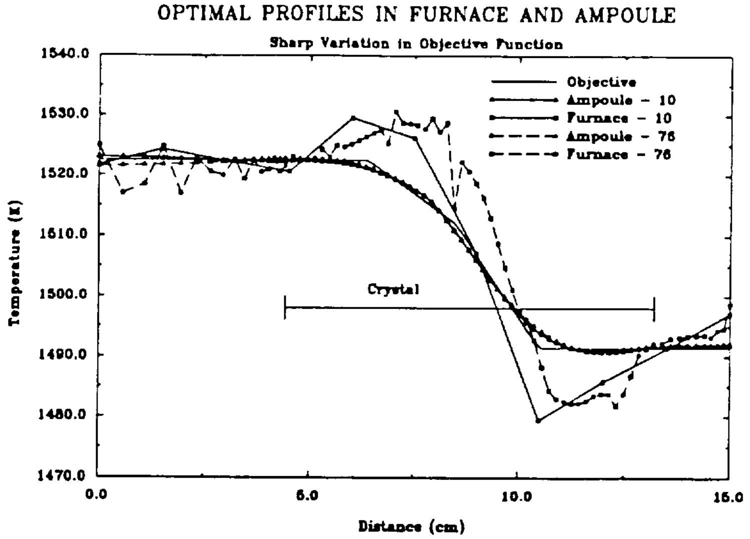

Fig. 6: Comparison of optimal solutions using 10 and 76 heating zones, respectively.

4: CONCLUSIONS

The results of the preceding section indicate the practicality of optimal process design and the utility of the sensitivity analysis for this class of problems. The optimal solution can be found with little user intervention. Indeed, the only work required beyond that for the normal analysis is the definition of the design variables and objective function.

In the future, we would like to extend this work to consider transient problems. However, the analysis becomes more complicated because the transient problem requires a convolution integral to be evaluated in the adjoint method. Other methods, such as direct differentiation, may prove to be more efficient for this class of problems.

The sensitivity formulation used for this work did not include the advective terms in the governing equations, and the fluid velocities were also not considered in evaluating the implicit variations of G. This will be important for advection-dominated flows, and this work is in progress.

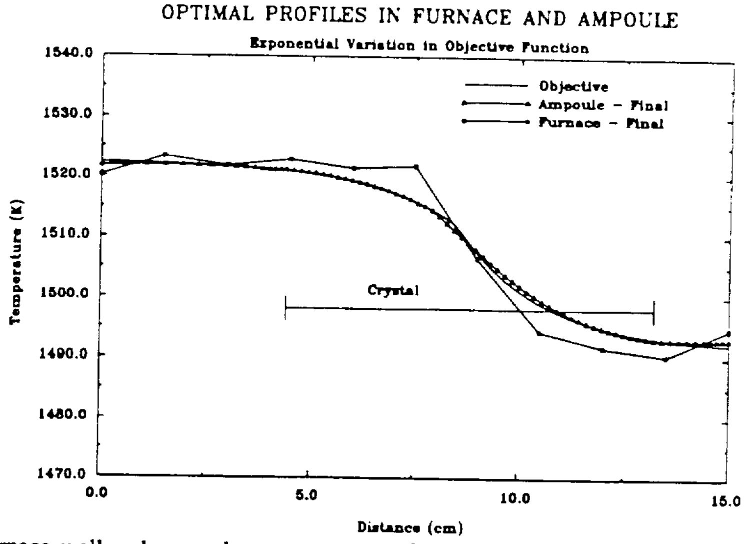

Fig. 7: Optimal furnace wall and ampoule temperature profiles for exponential variation in the objective function

5: ACKNOWLEDGMENT

The authors wish to thank NASA, which has supported this work under grant NASA NAGW-1683 and NAG3-1286, and the National Center for Supercomputing Applications for partial support of one of the investigators.

6: REFERENCES

[1] J. A. Dantzig and L. S. Chao. Interface shape control in bridgman crystal growth. In M. Rappaz and M. Ozgu, editors, *Modeling of Casting, Welding and Advanced Solidification Processes*, page in press, Warrendale, PA, 1991. TMS-AIME.

[2] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: with Applications*. McGraw-Hill, New York, 1984.

[3] M. S. Engelman. *FIDAP Theoretical Manual*. Fluid Dynamics International, Evanston, IL, 1987.

[4] E. J. Haug, K. Choi, and V. Komkov. *Design Sensitivity Analysis of Elastic Mechanical Systems*. Academic Press, New York, 1986.

[5] D. A. Tortorelli, R. B. Haber, and S. C.-Y. Lu. Design Sensitivity Analysis for Nonlinear Transient Thermal Systems. *Computer Methods in Applied Mechanics and Engineering*, 75:61–78, 1990.

[6] D. Tortorelli. Sensitivity Analysis for Nonlinear Constrained Elastostatic Systems. In S. Saigal and S. Mukherjee, editors, *Sensitivity Analysis and Optimization with Numerical Methods, AMD Vol. 115*, pages 115–126, 1990.

[7] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes*. Cambridge Universiy Press, Cambridge, 1986.

[8] R. D. Cook. *Concepts and Applications of Finite Element Analysis*. Wiley, New York, 1981.

[9] C. J. Chang and R. A. Brown. Radial segregation induced by natural convection and the melt/solid interface shape in vertical Bridgman growth. *J. Crystal Growth*, 63:343, 1983.

[10] J. A. Kafalas and A. H. Bellows. A comparative study of the influence of buoyancy driven fluid flow on gaas crystal growth. *Proc. AIAA Conference*, 1988.

Appendix A: Material Properties Used in the Simulations

Table I: Material properties used in the simulations

Material	Quartz	Air	Graphite	PBN - 1	PBN - 2	GaAs
Density (g/mm^3)						
	2.2×10^{-3}	2.35×10^{-7}	1.83×10^{-3}	1.9×10^{-3}	1.9×10^{-3}	5.71×10^{-3}
Thermal Conductivity (W/mmK)						
273 K	3.4×10^{-4}	1.15×10^{-5}	3.0×10^{-2}	2.50×10^{-2}	4.0×10^{-4}	1.7×10^{-3}
750 K	4.7×10^{-4}	1.43×10^{-5}	2.27×10^{-2}	1.70×10^{-2}	5.0×10^{-4}	1.7×10^{-3}
1060 K	6.6×10^{-4}	1.88×10^{-5}	1.78×10^{-2}	1.58×10^{-2}	5.6×10^{-4}	1.7×10^{-3}
1220 K	7.5×10^{-4}	2.02×10^{-5}	1.52×10^{-2}	1.51×10^{-2}	6.0×10^{-4}	1.7×10^{-3}
1511 K	7.5×10^{-4}	2.24×10^{-5}	1.29×10^{-2}	1.50×10^{-2}	6.0×10^{-4}	1.7×10^{-3}
1512 K	7.5×10^{-4}	2.24×10^{-5}	1.29×10^{-2}	1.50×10^{-2}	6.0×10^{-4}	3.5×10^{-3}
1600 K	7.5×10^{-4}	2.30×10^{-5}	1.24×10^{-2}	1.50×10^{-2}	6.0×10^{-4}	3.5×10^{-3}
Specific Heat (J/gK)						Enthalpy (J/g)
273 K	0.123	0.294	0.19	0.20	0.20	27.4
750 K	0.244	0.294	0.19	0.40	0.40	75.4
1060 K	0.278	0.294	0.19	0.44	0.44	106.5
1220 K	0.284	0.294	0.19	0.47	0.47	122.6
1511 K	0.299	0.294	0.19	0.47	0.47	151.8
1512 K	0.299	0.294	0.19	0.47	0.47	325.5
1600 K	0.299	0.294	0.19	0.47	0.47	575.5
Viscosity (g/mm s)						
<1511 K						1.0×10^{20}
>1512 K						1.7×10^{-3}

[11] M. E. Glicksman, S. R. Coriell, and G. B. McFadden. Interaction of Flows with the Crystal-Melt Interface. *Ann. Rev. Fluid Mech.*, 18:307–335, 1986.

[12] J. A. Dantzig. Modeling Liquid-Solid Phase Changes with Melt Convection. *International Journal of Numerical Methods in Engineering*, 28:1769–1785, 1989.

[13] M. Bercovier and M. S. Engelman. A Finite Element for Incompressible Fluid Flows. *J. Comp. Physics*, 30:181, 1979.