Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Crystal Growth 275 (2005) e349–e353

JOURNAL OF CRYSTAL GROWTH

www.elsevier.com/locate/jcrysgro

A framework for optimization of crystal growth processes applied to VGF growth of fluorides

Rainer Backofen[a,*], Axel Voigt[a], Dirk Wulff-Molder[b]

[a] Caesar Research Center, Ludwig-Erhard-Allee 2, 53175 Bonn, Germany
[b] Korth Kristalle GmbH, Am Jägersberg 6, 24161 Altenholz (Kiel), Germany

Available online 10 December 2004

Abstract

We develop a flexible framework for optimization of crystal growth processes. This framework is based on an accurate and robust process model and combines two optimization loops. Adapting of model parameters and optimizing of process parameters.

PACS: 81.10.Fq; 44.05.+e

Keywords: A1. Computer simulation; A2. Gradient freeze technique; B1. Halide

1. Introduction

In numerical modeling of industrial crystal growth the ultimate goal is to optimize running processes [1]. This objective sets two strong requirement on the numerical description of the growth process. It must be accurate within a given tolerance to describe the process appropriately and it must be robust to variations in parameters to allow for optimization. Such models can be set up in commercial software packages, e.g. FIDAP or CrysVUn, [2,3]. These packages usually fulfill the requirement on robustness. But to achieve the needed accuracy adjustments in parameters have to be made. Due to poor specifications of the involved materials or aging of these materials the uncertainties in the used material data have to be considered. To overcome this problem the uncertain material parameters have to be identified from experimental data for each individual process or furnace. This already sets the first stage of an optimization loop: find appropriate model parameters in a process model which allow to numerically describes the industrial process within a given tolerance. After this loop and a validation of the derived process model a second optimization loop follows which optimizes the process parameters to improve the industrial process. So two steps are needed to quantitative optimize

*Corresponding author.
E-mail address: rainer.backofen@caesar.de (R. Backofen).

0022-0248/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.jcrysgro.2004.10.132

crystal growth processes: identify material parameters and optimize process parameter, see Fig. 1.

Using commercial codes to set up such an optimization loop has a big drawback. The included optimization possibilities are often to restrictive, e.g. restricted to heater power only. In order to overcome this restrictions we developed a framework to interface a simulation model (CrysVUn) with more flexible optimization tools. The optimization is done by Quasi-Newton methods, which are adapted to the special needs and restriction of our optimization problem.

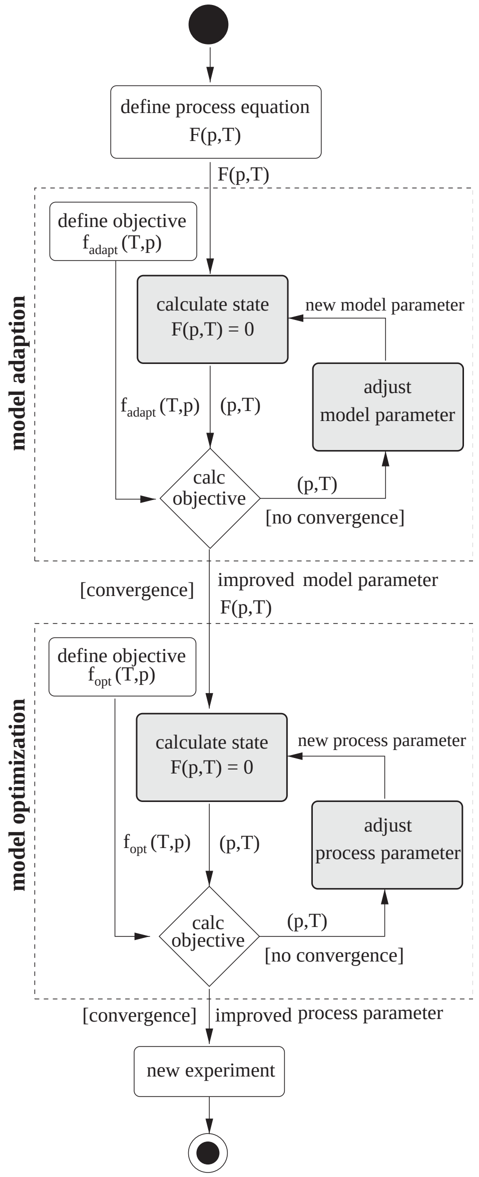

Fig. 1. Optimization framework.

The optimization framework is applied to an industrial growth problem for fluoride crystals.

2. Method

The numerical algorithms behind the used software package can formally be defined by a process equation:

$$\vec{F}(\vec{T},\vec{p}) = (F_0(\vec{T}_0,\vec{p}_0), F_1(\vec{T}_1,\vec{p}_1), \dots F_N(\vec{T}_N,\vec{p}_N)) = (0,0\dots 0), \quad (1)$$

where $F_i(\vec{T}_i,\vec{p}_i) = 0$ symbolize the solution of the global heat transfer problem at growth state i (time instant) and fixed parameters $\vec{p}_i$ (material parameter, geometry, heater power). In this setting the process equation defines the temperature distribution $\vec{T}_i$ at different growth states to the specified parameters $\vec{p}_i$. The optimization problem is formulated as

$$\min_{\vec{p}} f(\vec{p}, \vec{T}(\vec{p})) \quad (2)$$

$$\text{subject to } \vec{F}(\vec{T},\vec{p}) = 0, \quad (3)$$

$$p_{i,\min} \leqslant p_i \leqslant p_{i,\max}, \quad (4)$$

where f is the objective function which measures the deviation of the simulated and wanted thermal field $\vec{T}$. The parameter space for a single parameter p_i may be restricted by maximal and minimal values, e.g., tolerance on the material parameters, maximal heating power.

In the stated optimization situation we assume to be near to an optimum or looking for improvements near a working process. So local optimization is used to get an improved set of parameters. To get global optimization other algorithms can be used, e.g. genetic algorithm [4].

Local optimization algorithms, based on the ideas of the Newton method, minimize in every iteration step an approximation to the objective function at current parameter set $\vec{p}^k$

$$f(\vec{p}) = f(\vec{p}^k) + (\vec{p} - \vec{p}^k) \cdot \nabla f(\vec{p}^k) + \tfrac{1}{2}(\vec{p} - \vec{p}^k) \cdot H \cdot (\vec{p} - \vec{p}^k), \qquad (5)$$

where H is the Hessian, $H_{i,j} = \frac{\partial^2 f}{\partial p_i \partial p_j}$. A direction $\vec{d}$ is than defined by minimizing Eq. (5).

$$\vec{d} = \vec{p} - \vec{p}^k = -H^{-1} \cdot \nabla f(\vec{p}^k). \qquad (6)$$

Along this direction a line search based on a backtracking scheme is applied [5].

Since we use a closed software package for the simulation, the partial derivatives has to be calculated numerically by finite differences

$$\frac{\partial f}{\partial p_i} = \frac{f(\vec{p}) - \vec{f}(\vec{p} + h\vec{e}_i)}{h} + o(h). \qquad (7)$$

If n is the number of parameters, then $n+1$ calculations of the process equation are needed to calculate these derivatives. The Hessian takes $n(n+1)/2$ calculations. The far most time consuming part in our process model is the evaluation of the process equation. So especially in higher dimensional parameter spaces, the calculation of the Hessian by finite differences is not practical. Quasi-Newton methods bypass the problem of calculating the Hessian by evaluating the partial derivatives. The Hessian is approximated by an iterative procedure during the optimization loop. The most popular method is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [5]. The BFGS method additionally assures that $\vec{d}$ is always a descent direction.

Constraints in the parameter space are easily included by a logbarrier function Lb [5]

$$Lb(p_i) = -\mu(\log(p_i - p_{i,\min}) + \log(p_{i,\max} - p_i)) \qquad (8)$$

and added to the objective function. The logbarrier functions introduce a penalty term in the objective function when the parameter reaches the bounds. The numerical parameter μ adjusts the steepness of the logbarrier function near the bounds. The equality constraint due to the process equation is implicitly taken to calculate the objective function. So the process equation is always exactly fulfilled in every optimization step.

2.1. Implementation

In order to use the optimization algorithm the process equation and the objective function has to be evaluated automatically. CrysVUn is able to work in a batch version without graphical user interface and may be interfaced by other programs. Parameter variations can be done by modifying the material and geometry description in the file describing the setup and calculation results. Information needed to evaluate the objective function is extracted in the same way. Modification of parameters, starting and evaluating of simulations with CrysVUn are managed by an interface programmed in PERL. The interface is quite flexible and allows easily for change in material parameters, heater powers and numerical parameters. Simple geometrical changes are also possible.

Results may be temperature at points, position of interfaces or temperatures along lines. This values may be directly passed to the optimization algorithm or used to calculate the objective function externally.

A simple parallelization scheme is realized. Independent calculations to a set of parameters are calculated simultaneously on a network or multiprocessor PC.

3. Calculations

We consider VGF growth of BaF_2 (200 mm diameter) with a moving crucible, Fig. 2. The process is modeled in CrysVUn as a set of pseudo stationary simulations at different growth stages [6]. Four snapshopts which are evenly distributed in time are used. The heating powers and growth velocities are taken according to the experiment and used as input parameters for CrysVUn which takes global heat transfer by conduction and radiation into account. The semitransparent properties of BaF_2 are modeled by a multiband model. The goal in the optimization is to achieve a constant growth velocity.

In Fig. 3 the position of the phase boundary at the crystal axes experimentally obtained and

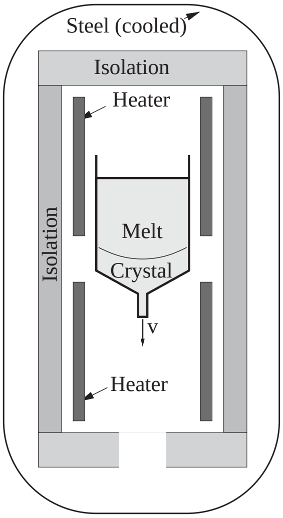

Fig. 2. VGF (Bridgmann) process.

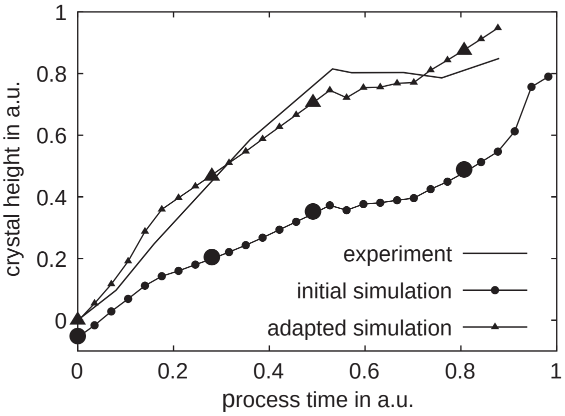

Fig. 3. Position of phase boundary in experiment and simulation. Circles indicate calculation with the inititial parameter set. Triangles are phase boundary obtained after adaption. Big symbols indicate the four snap shots used for adaption of the model.

predicted by simulation are compared. Three parameters are chosen to adapt the process model.

The thermal conductivity of the outer isolation, a grafite felt, is modified by adding a parameter (p_1). This is motivated by the quite poor specification of grafite felts and aging properties. Additionally this parameter is meant to mask the uncertainties changing the heat flow from the heater to the autoclave. This parameter is often used to get the right temperature level in the furnace. This parameter is already adapted in advance for initial simulation in order to get a phase boundary near seeding position ($p_1 = 0.09$).

The position of the phase boundary is defined by the interplay of the heat flow in the crucible and the BaF_2. This is taken into account by a factor for the thermal conductivity of the crucible support (p_2), and the solid phase of BaF_2 (p_3).

The variation of the conductivity of BaF_2 crystal is also meant to adjust the simplified approximation of the optical properties of BaF_2 in the calculation. The conductivity of the isolations may vary with different processes due to aging and with different furnaces due to accuracy of material specification. But the adapted conductivity of the BaF_2 should be constant for all treated processes.

The objective function is defined as the weighted mean deviation of the calculated $\mathrm{x}_{i,\mathrm{num}}$ and measured $\mathrm{x}_{i,\mathrm{exp}}$ phase boundary.

$$f_{\mathrm{adapt}} = \sum_i a_i (x_{i,\mathrm{num}} - x_{i,\mathrm{exp}})^2. \tag{9}$$

In five BFGS iterations a improved set of parameters are achieved. During this first optimization loop the process equation was solved approximately 40 times to adapt the model parameter. So 120 pseudostationary calculations were done.[1] The improved parameter set is $p = (0.089, 1.5, 2.6)$. The quite big changes in the thermal conductivities of BaF_2 and cone isolation indicate that this masks some other effects that are not taken into account in the simulation. Never the less, pragmatically, this can be taken as a model to predict the phase boundary for this process.

The adapted model is now used to optimize the process. The objective is to have a constant growth velocity of the phase boundary. So the objective function is just the mean deviation to a optimal position of the phase boundary $x_{i,\mathrm{opt}}$:

$$f_{\mathrm{adapt}} = \sum_i a_i (x_{i,\mathrm{num}} - x_{i,\mathrm{opt}})^2. \tag{10}$$

[1] This is fastest achieved convergence. During research similar problems of this complexity were solved in typical 5-20 BFGS iterations.

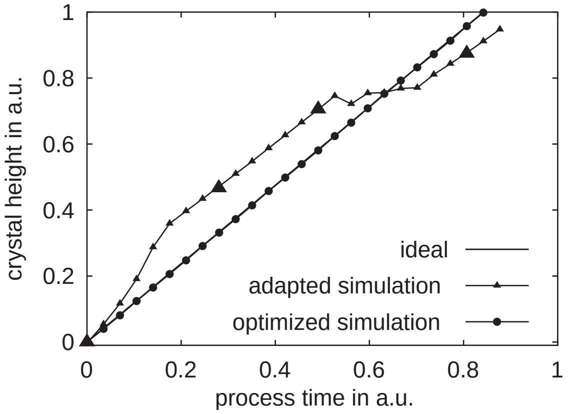

Fig. 4. Position of phase boundary in adapted and optimized model. Circles are phase boundary obtained after adaption, Fig. 3. Triangles are obtained after optimization. The optimized heating power are shown in Fig. 5.

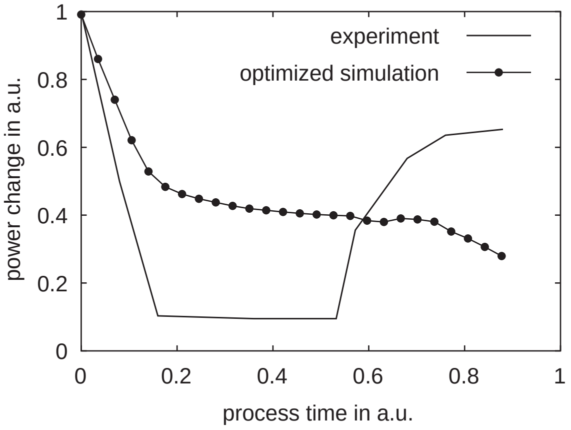

Fig. 5. Heating power in process and predicted after optimization of the adapted model.

The only available process parameter is the heating power of the lower heater. In Figs. 4 and 5 the results of the optimizations are shown. An optimized process leads to a smaller decrease of the heating power at the beginning and a smaller increase at the end. Surely, qualitatively this can be easily concluded without any simulation. But the simulation provides a quantitative prediction of the optimized process.

4. Conclusion

In order to solve various optimization problems in crystal growth an interface for CrysVUN was developed. The interface allows to change parameters of the calculation and extract results after the simulation in a highly automatic way. This is used to feed a optimization tool based on the BFGS method to do local optimization.[2] So problems like model adaption and process optimization may be done more effectively. Especially complex objective functions and parameter sets may be defined externally.

The framework was applied to the optimization of a BaF_2 VGF growth. Two steps, model adaption and process optimization, leads to an improved set of parameters. The adapted model and the optimized process has to be validated by further experiments.

Numerical parameters used in the optimization have to be adjusted with care to get fast and stable convergence, e.g., step width used to do numerical derivation or maximal and minimal allowed step width in the optimization.

[2] This interface may be also used in connection with other optimization tools.

Acknowledgements

This work was supported by BMBF through 03N1038.

References

[1] G. Müller, Experimental analysis and modelling of melt growth processes, J. Crystal Growth 237–239 (2002) 1628.

[2] http://www.fluent.com/software/fidap.

[3] Crystal Growth Laboratory, University Erlangen-Nürnberg, http:/ww6.ww.uni-erlangen.de/cgl/crysvun.

[4] T. Fühner, T. Jung, Use of genetic algorithms for the development and optimization of crystal growth processes, J. Crystal Growth 266 (2004) 229.

[5] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Series in Operations Research, Springer, New York, 1999.

[6] R. Backofen, M. Kurz, G. Müller, Process modeling of the industrial VGF growth process using the software package CrysVUN+ +, J. Crystal Growth 211 (1–4) (2000) 202.