

Modeling and Control of Cadmium Zinc Telluride Grown via an Electro-dynamic Gradient Freeze Furnace

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Lisa San Lun

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jeffrey J. Derby, Co-Adviser
Prodromos Daoutidis, Co-Adviser

December, 2007

© Lisa San Lun 2007

Acknowledgments

There are so many people that have helped and supported me during the past (nearly) five years of my graduate studies. First and foremost I thank my advisors Jeffrey J. Derby and Prodromos Daoutidis of the Chemical Engineering and Materials Science department at the University of Minnesota. From them I have learned, not only technical things presented in this thesis, but more intangibles on how to conduct research, how to think about a problem, and how to present my work clearly. I am indebted to Andrew Yeckel without whom none of this work would be possible. His friendship and mentoring has been invaluable, and something I will always treasure. I also thank my thesis committee members Professors Satish Kumar, Eray Aydil, and Victor Barocas for your support and encouragement.

There are many staff members of the department that made my graduate school experience so rich. I would like to thank Julie Prince for keeping all of the graduate students in line and doing it so graciously. You can always count on me to help setup the AIChE reception. I am also grateful to Teresa Bredahl who does all the little things that you don't know are important until she's on vacation. Teresa always brightens up my day with her smile. And there's Mary Nissen, who without none of us would get paid. Mary probably knows more rules and regulations than any other person in the university. Without these people, our department would grind to halt. We are very lucky to have the best staff in the entire university.

I've worked with many talented and exceptional students while at Minnesota. I'd like to thank residents of the Popcorn Lounge (or Control Group office) Paul Sonda, Marie Contou-Carrere, Michael Baldea, Ziomara Gerdtzen, and Sujit Jogwar. Mike has also been a great friend, and I can't imagine my time here without you. Ziomara was not only my groupmate but also my roommate for two years. She is a great role model on how to be a smart, confident woman in engineering. Love you! With the addition of Sujit to the group, I lost the largest single-person office in the department. He helped me not go crazy from my own company, and soon he'll be the one in charge. I did not know my Derby Group

members as well, but they are great bunch of people: Arun Pandey, Bing Dai, Hadrian Djohari, Yong-Il Kwon, David Gasperino, and Wei Wang. It's great seeing old members at AIChE Annual Meetings. I'd also like to acknowledge the Kumar Group students who lost their office and moved in with me: Rick Lenz, Scott Roberts, Shawn Dawd, Nazish Hoda, and Balram Suman. They've made my last year here a lot of fun and active. Also, I'd like to thank our visitor from Norway Håkon Dahl-Olson, his wife Maren, and son Magnus. I'm glad they liked the exotic American Halloween, and I hope they keep it up at home.

I also made many friends during my time here. I'd like to thank Dave and Emily Umulis and their daughter Madeline. Dave is the future hot-shot professor. I hope his students will appreciate his sense of humor as much as I do. Emily is my scrapbooking buddy, even though our projects never seem to end. A huge shout out to Mike and Judy Dolgovskij. The hours Mike and I spend throwing things across the room and watching videos of people jumping off stuff kept me sane. Judy is one of the most generous and caring people I know once you get passed her sharp tongue. You go, girlfriend! Thanks to Andy Warta, my dancing partner and companion to all things geeky. Slowly we will convince the world that Bruce Campbell is the greatest actor of all time. In the past couple of years I've also gotten a lot closer to Ling Zhang, Katie Wlaschin and her husband Aaron. I'm going to miss the summer alfresco lunches with Katie. Ling is the sweetest thing ever, and she can cook up a storm! I'd like to acknowledge my SCPO companions Spyros Vicatos and Marlene Castro. Spyros is such a sweetheart and my favorite TA. Marlene and I will always share our love for the "Beach Boy". She knows who I mean. I also would like to thank Professor L. E. "Skip" Scriven and his wife Dorene for their sharing their love of classical music and their company. We all miss Skip profoundly.

I did manage to have a small social life outside of school. My roommate of the past two years and going, Leanna Stowman, has been a great break from all things technical. My first roommate in the Twin Cities, Sherri Schummer, was my partner in battling loud neighbors. Sherri studied Chinese and had the misfortune of having a non-speaking, ethnically Chinese roommate. I'd like to specially mention my friend Jocelyn Wong who I've known entirely my whole life, and her partner Michelle Cheuk. They added to my cultural well-roundedness and I miss them everyday. I've also been very lucky to have some of the best friends in the world from my undergraduate days at Northwestern University. My core group, who'll I'll retire with à la *Golden Girls*, are Julia Lin, Lissette Lau, Yee Man Wong Bergstrom, Kaushal Desai, Ralph DiLallo, and Nina Wang. Every time I see them it's as if we've never been apart. I can't put into words how much they all mean to me. I'd also like to acknowledge some other NU alums and staff I've kept in touch with:

Lars Bergstrom, Phil Tang, Tani Sethy Sailer, Lisa Lu Wang, Frank Ferko, Ruth Young, and Professor Randy Snurr. In addition, I'd like to thank my IBM co-workers who really taught me how to be an engineer: Roger Dandurand, Jim Nakos, Beth Ward, Rob Miller, Sean Dion, Tammie Carrier, Jim Quinlivan, and Frank Ammirato. Thank you for all your support.

Profound thanks to my alter ego Cheryl Cheng who is probably my best friend. She probably knows me better and more about me than any other person in the world. I can always count on her when I'm feeling down or just want to chat about 24. She's never afraid to try new things, and I am not afraid to be dragged along when she's there.

Finally, I am indebted to my family especially my parents Albert and Donna Lun. They have always believed that I am capable to great things, and have pushed me to realize my goals. My smarter older sister Nicole Lun and more talented younger brother Michael Lun have been my constant companions all my life. I am also blessed by and extended family that never tires of spending time with me. My aunts Ivy Lee and Emily Chau have been a great source of love for me. My aunt Lynda Eng is also a great friend. She'll probably say that I owe her a t-shirt from Minnesota. My grandmother Grace Eng is an amazing woman, and has accomplished so much in her life by raising two daughters alone in a foreign country. She has generously supported my siblings and I with her strength and love. My Yeah-Yeah and Ma-Ma, Lik-Sang and Chor-Shing Yeung, could not be here today, but I know how proud they would be. They always wanted a doctor in the family. Thank you all for helping me the person I am today.

to my parents K. Albert Lun and Donna K. Lun

Abstract

In this thesis, numerical models are used to study the effect of novel processing methods to grow bulk, single crystal cadmium zinc telluride (CZT) in a vertical Bridgman (VB) furnace. Additionally, we investigate new mathematical algorithms for improved solving capability of equations that describe such crystal growth systems.

A two-dimensional crystal growth model for the simulation of bulk crystal growth in a VB system is presented. This model consists of conservation equations for coupled continuum level transport of heat, mass, and momentum. Thermodynamic relations associated with phase change are also included. The Galerkin finite element method is used to discretize the spatial portion of the governing equations. The resulting sets of nonlinear algebraic equations are solved using Newton's method.

Novel processing methods that are not practical to attempt in experiments are investigated using numerical modeling. A two-dimensional, planar, crystal growth model is used to explore the effect of ampoule tilting on zinc segregation in a CZT crystal. Tilting is shown to improve lateral segregation. We also analyze the use of closed-loop control to improve the macroscopic melt-crystal interface shape during growth by changing the furnace temperature gradient. Targeted closed-loop control on the temperature gradient adjacent to the solid only gave the best results and unexpectedly produced a favorable convex shape.

A multi-scale crystal growth model is developed by coupling pre-existing codes, one which specializes in modeling the complex crystal growth process and the other which specializes in modeling the heat transfer effects in a furnace. Previously, a coupling algorithm based on the block Gauss-Seidel method was used but it converged unreliably [136, 196]. Here, we use an Approximate Block Newton approach where we approximate Newton's method used to solve the two separate models as if they were one monolithic model.

A Schur complement formulation is employed to solve the free-boundary problem associated with melt crystal growth systems. With this form, the difficult interface location part of the problem is mapped away from the equations governing transport. We assess

the behavior of this method using two-dimensional simulations, but the goal is to improve solvability of three-dimensional problems.

Contents

Contents	x
List of Tables	xi
List of Figures	xv
Notation	xxiii
I Introductory Material	1
1 Introduction	2
1.1 Melt crystal growth methods	3
1.2 Physics of vertical Bridgman growth	5
1.2.1 Heat transfer effects	7
1.2.2 Mass transfer effects and segregation	8
1.2.3 Momentum transfer effects	9
1.3 Modeling vertical Bridgman systems	10
1.4 Thesis format	14
2 Model and Numerical Implementation	17
2.1 Chapter Summary	17
2.2 Model and Governing Equations	18
2.2.1 Navier-Stokes Equations	19
2.2.2 Conservation of Energy	24
2.2.3 Conservation of Mass	26
2.3 Numerical Implementation	27

2.3.1	Discretization of Computational Domain	27
2.3.2	Discretization of Unknowns	28
2.3.3	Weighted Residual Equations	31
2.3.4	Deforming Boundary	34
2.3.5	Time Integration	36
2.3.6	Continuation	39
2.4	Conclusion	40
II	Crystal Growth Modeling	42
3	Tilting of Ampoule to Decrease Lateral Zinc Segregation	43
3.1	Chapter Summary	43
3.2	Introduction	44
3.3	Modeling Approach	46
3.4	Simulation Results	48
3.4.1	Flow and Mass Transfer in the Melt	49
3.4.2	Axial and Lateral Segregation of Zinc in the Crystal	53
3.4.3	Overall Segregation of Zinc in the Crystal	55
3.5	Discussion and Conclusion	58
4	Control of Interface Shape Via Furnace Gradient	61
4.1	Chapter Summary	61
4.2	Introduction	62
4.3	Base Case Model	65
4.4	Open-Loop Study	66
4.4.1	Model Formulation	68
4.4.2	Results and Discussion	70
4.5	Closed-Loop Control Study	78
4.5.1	Model Formulation	78
4.5.2	Results and Discussion	80
4.6	Concluding Remarks	85
III	Algorithm Developments	88
5	Multi-Scale Model via an Approximate Block Newton Method	89

5.1	Chapter Summary	89
5.2	Introduction	90
5.3	Approximate Block Newton Method	96
5.4	Example Problems	107
5.4.1	One-Dimensional Model	107
5.4.2	Two-Dimensional Model	114
5.4.3	Electrodynamic Gradient Freeze Furnace Model	127
5.5	Concluding Remarks	134
6	Schur complement formulation for solving a free-boundary crystal growth problem	136
6.1	Chapter Summary	136
6.2	Introduction	137
6.3	Model Formulation	138
6.3.1	Governing Equations	139
6.3.2	Schur Complement Formulation	141
6.3.3	Approximations and Preconditioning	146
6.4	Results and Discussion	148
6.4.1	Comparison of Convergence Rate	149
6.4.2	Scaling of Computational Effort	152
6.4.3	Effectiveness of Approximations and Preconditioning	163
6.5	Concluding Remarks	166
IV	Concluding Remarks and Appendix	168
7	Concluding remarks	169
7.1	Thesis Summary	169
7.2	Directions for Future Research	172
7.2.1	Investigating the Approximate Block Newton Method	173
7.2.2	Development of time-dependent, multi-scale model for control applications	175
7.2.3	Implement Schur complement formulation on 3D code	178

A Use of Under-Relaxation in Coupling Global and Local Crystal Growth	
Models	180
A.1 Chapter Summary	180
A.2 Introduction	181
A.3 Formulation of Iterative Algorithm	182
A.3.1 Convergence of Iterative Algorithm	183
A.4 Problem Formulation and Results	184
A.4.1 Coupling Technique	185
A.4.2 Results	185
A.5 Conclusions	189
B Alternative Derivation of Approximate Block Newton Method	190
Bibliography	197

List of Tables

2.1	Physical properties of materials.	21
2.2	Characteristic values	23
2.3	Dimensionless groups of CZT for $L_o = 2.54$ cm and $\Delta T_f = 2.33$ K/cm . . .	23
2.4	Types of basis functions	30
3.1	System operating parameters	48
3.2	Mesh comparison	49
4.1	Physical properties of CdTe [92], system dimensions, and operating parameters.	70
4.2	Mesh size comparison of percent deviation of velocity magnitude at various cross-sections within the melt.	71
6.1	Comparison of L_2 norm of residual vector as a function of Newton iteration of the direct frontal solver and the general Schur complement formulation method.	148
6.2	The minimum number of (inner) Krylov vectors needed for Newton's method to converge to the solution and to converge quadratically.	152
6.3	Description of computational meshes used to determine the scaling dependencies of methods M1, M2, and M3 on problem size.	153
6.4	Minimum number of Krylov vectors required by methods M2 and M3 for Newton's method to converge.	160
6.5	Effect of preconditioner $\tilde{S}$ on number of Newton iterations required to reach convergence, and on number of iterations of outer solver during first Newton iteration.	165

List of Figures

1.1	Schematic diagram of a Czochralski melt growth furnace.	4
1.2	Schematic diagram of a vertical Bridgman melt growth furnace.	6
1.3	Modeling approaches for various length and time scales, reproduced from Derby [46].	11
2.1	Schematic of the vertical Bridgman crystal growth system	22
2.2	Schematic of the computational domain	22
2.3	Example of a discretized domain in one dimension	28
2.4	A sample element and its nodes	29
2.5	Parametric mapping to a unit square parent element	33
2.6	Sample mesh with deformed boundary at melt-crystal interface.	36
3.1	Streamline contours of various tilt angles	50
3.2	Mole fraction contour of zinc at different angles of tilt	51
3.3	Axial zinc concentration is plotted along the centerline ($y = 0.0$ in) for different angles of tilt.	54
3.4	Lateral zinc concentration is plotted across the width of the crystal at $x = 1.5$ in. for different tilt angles.	56
3.5	Lateral zinc concentration profiles for tilt angles of 0° and 10° at an axial height in the crystal of $x = 1.5$ in.	56
3.6	Location of $0.6 \text{ in} \times 0.6 \text{ in}$ square with optimal c_{RMS} , defines in § 3.4.3, for each case.	57
3.7	Optimal c_{RMS} plots as a function of ampoule tilt angle; c_{RMS} is defined in § 3.4.3 and is computer over the optimal square area indicated in Figure 3.6 for each grown crystal.	58

4.1	Schematic diagram of the computational domains and meshes of the (a) global model and (b) local model.	67
4.2	Axial furnace temperature gradient for different values of α	69
4.3	Perturbation of furnace temperature gradient for different values of β	69
4.4	Streamline and temperature contours as a function of the gradient magnification factor, α	73
4.5	Maximum value of streamfunction versus α	74
4.6	Radial temperature in the melt plotted slightly above the height of the solid interface for various values of α	74
4.7	Streamline and temperature contours comparing effect of increased heat convection in the melt versus increased heat conduction in the solid	75
4.8	Axial height of interface versus radius for (a) $\alpha = 1$ and (b) $\alpha = 10$ at different values of β	76
4.9	Diagram depicts interface deflection	79
4.10	Proportional control simulation results for control	82
4.11	Representative system results at $t = 16$ hrs for closed-loop control solely along the solid portion of the furnace boundary.	84
5.1	(a) Schematic representation of the one-dimensional model problem. (b) Steady-state temperature profiles of the one-dimensional model.	108
5.2	Comparison between convergence maps of the one-dimensional model as a function of α and β of (a) block Gauss-Seidel and (b) the Approximate Block Newton methods for $R = 0$ and $R = 5.67$	113
5.3	Convergence maps of the one-dimensional model using the approximate Block Newton method for different values of radiation number R	115
5.4	Comparison of the one-dimensional model between the approximate Block Newton and block Gauss-Seidel methods for (a) percentage of converged cases and (b) average number of iterations as a function of radiation number R	116
5.5	Comparison of the residual norm versus iteration number one-dimensional model between the exact Newton and the Approximate Block Newton methods for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-8}$ for (a) $\alpha, \beta = (1,0)$ and the (b) $\alpha, \beta = (0.3,0.32)$ at $R = 5.67$	117
5.6	Schematic representation of the two-dimensional model problem.	119

5.7	Temperature contours of the (a) global and (b) local domains of the linear two-dimensional model, $Ra = 0$, where $\Delta T = 1$ K.	123
5.8	(a) Temperature contours of the global domain, and (b) temperature and streamline contours of the local domain of the nonlinear two-dimensional model, $Ra = 6 \times 10^3$	124
5.9	Comparison of (a) temperature [K] and (b) heat flux [W/m^2] as a function of arc length (with respect to the ampoule radius) along the shared boundary for $Ra = 0$ and $Ra = 6 \times 10^3$ for the two-dimensional model.	125
5.10	Residual norm of boundary data versus block Newton iteration number at different number of Krylov vectors for (a) $Ra = 0$ and (b) $Ra = 6 \times 10^3$ of the two dimensional model.	126
5.11	Results of the self-consistent model of an EDG furnace: (a) temperature image plot of global furnace calculated by CrysVUn, scale indicated, and (b) temperature and streamline contours of the local crystal calculated by Cats2D.	129
5.12	Comparison of (a) temperature [K] and (b) heat flux [$\text{W}/\text{m}^2/\text{s}$] as a function of arc length (with respect to ampoule radius) along the shared boundary of the EDG furnace model.	131
5.13	The dependence of rate of convergence on Krylov space size is determined by plotting the residual L_2 norm of the difference in Cats2D and CrysVUn boundary data versus block Newton iteration number for $\epsilon = 1 \times 10^{-4}$	132
5.14	The dependence of rate of convergence on ϵ for $KV = 18$	133
5.15	The dependence of rate of convergence on ϵ for $KV = 22$	133
6.1	(a) Schematic of two-dimensional, two-phase sample problem discretized by a representative 40×40 element mesh, and (b) solution of the temperature field.	140
6.2	Comparison of number of Newton iterations required to reach convergence as a function of number of (inner) Krylov vectors.	151
6.3	Plot of $\log(\text{operation time})$ versus $\log(N_{nodes})$ for methods M1, M2, and M3.	157
6.4	Scaling dependency of (a) method M2 on k_G , (b) method M3 on k_i , and (c) method M3 on k_o	158
6.5	Scaling relationship between minimum number of (a) Krylov vectors used by M2 k_G , (b) inner number of Krylov vectors k_i with respect to N_{total} and (c) outer number of Krylov vectors k_o with respect to $N_{interface}$	161

6.6	Scaling dependency of operation time with respect to N_{nodes} for M2 and M3 solving two-dimensional problems.	162
A.1	Comparison between convergence of the one-dimensional model for $R = 0$ and 5.67 as a function of the coupling parameters α and β , for: (a) unrelaxed iterations, $\omega = 1$; and (b) under-relaxed iterations $\omega = 0.1$. Light areas represent algorithms that will converge to the solution for a suitable initial guess.	187
A.2	Performance measures of the block Gauss-Seidel iterations as a function of under-relaxation parameter, ω	188

Notation

Dimensions are given in terms of mass, length, time, degree (temperature), energy, and dimensionless (—). For unitless quantities, dimensions are left blank. Boldface terms are vectors or tensors. Where multiple definitions are given, first chapter appearance is referenced.

Symbol	Description	Units
A	area, § Chapter 2	length ²
A, B, C	subblocks of Jacobian matrix, Chapter 6	—
A_Z, B_Z, C_Z	number of nonzero entries in subblock matrices A , B , and C , respectively	—
$\tilde{A}$	approximation of block matrix A	—
$\mathbf{C}$	placeholder matrix for ABN method	—
C_p	heat capacity	energy/(mass×time)
c	species concentration	mass/length ³
$\hat{c}$	approximate concentration	mass/length ³
c_i	initial species concentration	mass/length ³
c_o	characteristic species concentration	mass/length ³
D	diffusion coefficient	length ² /time
D_ξ, D_ζ	nodal diffusivities	length ² /time
$\mathbf{d}$	nodal displacement vector	length

continued on next page

continued from previous page

Symbol	Description	Units
$\mathbf{e}$	unit vector	—
$\mathbf{F}$	nonlinear vector function	—
$\mathcal{F}$	fixed-point solver	—
$\mathbf{f}$	solver residual	—
f	frontwidth of frontal solver	—
$\mathcal{G}$	fixed-point solver	—
g	gravitational constant	length/time t^2
$\mathbf{g}$	residual of fixed-point solver	—
$\mathbf{g}$	gravitational acceleration vector	length/time t^2
H	height	length
H_{in}	inner ampoule height	length
H_{out}	outer ampoule height	length
$\mathcal{H}$	Sobolev space	—
H_f	enthalpy of fusion	energy/mass
h	heat transfer coefficient	energy/(length $^2 \times$ time $\times$ degree)
$\mathbf{I}$	identity tensor	—
J	Jacobian of mapping, Chapter 2	—
	Jacobian matrix, Chapter 6,	
$\mathbf{J}$	Jacobian matrix, Chapter 5	—
K	segregation coefficient	—
k	thermal conductivity, Chapter 2	energy/(length $\times$ time $\times$ degree)
	number of Krylov vectors, Chapter 6	—
k_a	thermal conductivity of ampoule	energy/(length $\times$ time $\times$ degree)
k_G	number of Krylov vectors used by GMRES	—
k_i, k_o	number of inner and outer Krylov vectors	—

continued on next page

continued from previous page

Symbol	Description	Units
k_m	thermal conductivity of melt	energy/(length $\times$ time $\times$ degree)
k_s	thermal conductivity of crystal	energy/(length $\times$ time $\times$ degree)
L	length	length
L_o	characteristic length	length
M	mass	mass
M1	direct frontal solver	—
M2	iterative solver, GMRES	—
M3	Schur complement formulation with direct solver	—
M4	Schur complement formulation with iterative solver	—
$\mathbf{M}$	mass matrix	—
N_c	number of concentration nodes	—
N_{el}	number of geometry nodes, Chapter 2	—
	number of elements in z -direction, Chapter 6	—
N_{nodes}	number of nodes in z -direction	—
N_{total}	total number of unknowns	—
N_p	number of pressure nodes	—
N_T	number of temperature nodes	—
N_v	number of velocity nodes	—
N_Z	number of nonzero entries in a Jacobian matrix	—
$\mathbf{n}$	outward normal unit vector	—
n	number of field and mesh unknowns	—
m	number of interface unknowns	—
p	pressure	mass/(length $\times$ time ²)
$\hat{p}$	approximate pressure	mass/(length $\times$ time ²)
q	heat flux	energy/(length ² $\times$ time)

continued on next page

continued from previous page

Symbol	Description	Units
$\mathbf{q}$	vector of unknowns, Chapter 2	—
	placeholder vector, Chapter 5	—
R	radius	length
R_{in}	inner ampoule radius	length
R_{out}	outer ampoule radius	length
$\mathbf{R}$	residual vector, Chapter 2	—
$\mathbf{r}$	residual vector, Chapter 5	—
$\mathcal{R}$	weighted residual	—
r, z	cylindrical coordinates	length
r_f	vector of field unknowns	—
r_i	vector of interface unknowns	—
$\mathbf{r}_1, \mathbf{r}_2$	residuals of coupling problem	—
$\mathbf{r}_S$	Schur complement	—
	right-hand-side *	—
S	Schur complement matrix, Chapter 6	—
$\mathbf{S}$	Schur complement matrix, Chapter 5	—
$\tilde{S}$	approximate Schur complement matrix	—
S_ξ, S_η	one-dimensional Jacobians of mapping	—
$\mathbf{T}$	stress tensor	mass/(length $\times$ time ²)
T	temperature	degree
$\hat{T}$	approximate temperature	degree
T_f	furnace temperature	degree
T_H	high temperature	degree
T_i	interface temperature	degree
T_L	low temperature	degree
T_{mp}	melting-point temperature	degree
T_o	characteristic temperature, Chapter 2	degree

continued on next page

continued from previous page

Symbol	Description	Units
	base case temperature, Chapter 4	degree
dT	temperature perturbation	degree
∇T_f	furnace temperature gradient	degree/length
t	time, Chapter 2	time
	placeholder vector, Chapter 6	—
Δt	time step	time
$\mathbf{t}$	unit tangent vector	—
u	placeholder vector, Chapter 6	—
V	volume	length ³
V_o	characteristic velocity	length/time
$\mathbf{V}_a$	ampoule translation rate	length/time
$\mathbf{V}_{ms}$	velocity of melt-crystal interface	length/time
$\mathbf{v}$	velocity	length/time
$\hat{\mathbf{v}}$	approximate velocity	length/time
W_f	total work of field update step in method M4	—
W_{M1}, W_{M3}, W_{M4}	total work of methods M1, M3, and M4, respectively	—
W_r	total work of right-hand-side step in method M4	—
W_S	total work os Schur complement step in method M4	—
w	weight function, Chapter 2	—
	placeholder vector, Chapter 6	—
$\mathbf{w}$	Krylov vector, Chapter 5	—
X_H	axial height of interface at external boundary	length
x, y	Cartesian coordiantes	length
x	placeholder vector	—
$\mathbf{x}$	position vector, Chapter 2	length
	unknowns associated with	—

continued on next page

continued from previous page

Symbol	Description	Units
	$\mathbf{r}_1$, Chapter 5	
Y	placeholder matrix	—
y	placeholder vector	—
$\mathbf{y}$	unknowns associated with $\mathbf{r}_2$, Chapter 5	—
z_H	axial position corresponding to T_H	length
z_i	axial position of interface	length
z_L	axial position corresponding to T_L	length
z_{mp}	axial position of T_{mp}	length
z	placeholder vector	—
Greek symbols		
α	thermal diffusivity, Chapter 2, § 2.2.2	length ² /times
	continuation parameter, Chapter 2, § 2.3.6	—
	gradient magnification factor, Chapter 4	—
	mixed boundary condition parameter, Chapter 5	—
β	thermal compressibility of melt, Chapter 2	degree ⁻¹
	perturbation maxima, Chapter 4	—
	mixed boundary condition parameter, Chapter 5	—
δ	boundary, Chapter 2	length
	unknown update vector, Chapter 6	—
δ_f	field unknowns update vector	—
δ_i	interface unknowns update	—

continued on next page

continued from previous page

Symbol	Description	Units
	vector	
δ_z	displacement in z -direction	—
ϵ	emissivity of ampoule/crucible	—
ϵ_i	GMRES tolerance of inner iteration	—
ϵ_o	GMRES tolerance of outer iteration	—
κ	ratio of thermal conductivity	—
Λ	density ratio	—
μ	dynamic viscosity, Chapter 2	mass/(length $\times$ time)
	Poisson's ratio, Chapter 6	—
μ_g	chemical potential	energy
ν	kinematic viscosity	length ² /time
ξ, ζ	unit cell coordinates	—
ρ	density	mass/length ³
σ	linear-elastic stress tensor	mass/(length $\times$ times ²)
ϕ	basis function	—
ψ	basis function	—
Ω	domain	length ²
$\delta\Omega$	boundary	length
∇	gradient operator	length ⁻¹
∇^2	Laplacian operator	length ⁻²
Dimensionless groups		
Gr	Grashof number	
Ma	Marangoni number	
Pe _M	Peclet number, mass	
Pe _T	Peclet number, thermal	
Pr	Prandtl number	
Ra	Rayleigh number	
Sc	Schmidt number	
Sf	Stefan number	
Subscripts		

continued on next page

continued from previous page

Symbol	Description	Units
a	ampoule	
B	bottom	
f	field	
G	GMRES	
$global$	variables belonging to global model	
i	interface, Chapter 6, § 6.3.2 inner, Chapter 6, § 6.4.1	
$local$	variables belonging to local model	
m	melt	
o	reference, Chapter 2 outer, Chapter 6	
S	side, Chapter 5 Schur complement, Chapter 6	
s	solid crystal	
$shared$	shared boundary	
T	top	
z	z -direction	
η	η direction	
ξ	ξ direction	
Superscripts		
j	nodal point index	
k	residual index, Chapter 2 iteration index, Chapter 5	
Overbar		
$\hat{a}$	approximation of variable	
$\dot{a}$	time derivative of variable	
$\tilde{a}$	variable is fixed	

Part I

Introductory Material

Chapter 1

Introduction

Contents

1.1	Melt crystal growth methods	3
1.2	Physics of vertical Bridgman growth	5
1.2.1	Heat transfer effects	7
1.2.2	Mass transfer effects and segregation	8
1.2.3	Momentum transfer effects	9
1.3	Modeling vertical Bridgman systems	10
1.4	Thesis format	14

Cadmium zinc telluride (CZT), a II-VI semiconductor, is a material of great technological interest that has received intense research attention of late related to production of large, high-quality crystals. There are many industrial applications for CZT. It is used as a substrate in the fabrication of high performance infrared focal plane arrays made of mercury cadmium telluride (HgCdTe) [86, 178], because of its less brittle nature and high lattice matching with HgCdTe. More recently, and arguably more importantly, CdZnTe is a powerful gamma-ray and x-ray detector that operates at room-temperature conditions. Electronic devices based on CZT are able to detect and identify isotopes emitting gamma-radiation [26, 51, 79, 142].

Relatively large, single crystalline CZT is needed for radiation detector devices. An example size is a (0.6 in $\times$ 0.6 in $\times$ 0.6 in) cube. To grow such sizes, CZT is typically grown via the vertical Bridgman (VB) process. Ideally, CZT is grown as a single crystal with uniform composition and minimal crystalline defects, however, this rarely occurs in practice. Achieving these goals is also incompatible with producing crystals at high

growth rates. Rudolph and Mühlberg [147] detail the challenges of growing cadmium telluride, most of which applies to CZT growth. Presently, only a few companies worldwide manufacture CZT, and none produce large quantities leading to low availability of material at high cost. This in turn creates a bottleneck to mass production of devices based on CZT at a reasonable price.

Melt crystal growth of CZT is a slow and expensive process. Cadmium also has a relatively high partial pressure and will out-gas from the melt path. For stoichiometry control CZT is grown in a confined container. The materials used for the container, or ampoule, is opaque in CZT growth so no visible data is available during growth. Furthermore, the unusual thermophysical properties of CZT result in atypical growth for semiconductor crystals. For example, the melt phase of cadmium telluride is semiconducting due to its high melt viscosity and low melt thermal conductivity, while more common semiconductors such as silicon and gallium arsenide have metallic melt phases [69]. In addition, cadmium telluride has a relatively large latent heat of fusion and a fairly large equilibrium segregation coefficient which, when combined, cause considerable difficulty maintaining uniform zinc composition in the grown crystal. Due to these different factors, well understood growth models and processes, such as for silicon or gallium arsenide, cannot be applied to CZT. Therefore, to understand the driving forces that influence CZT growth and improve the process, computational modeling becomes necessary.

1.1 Melt crystal growth methods

The majority of semiconductor single crystals are grown by directional solidification from the melt phase by many melt crystal growth methods. The premise of such methods is that the material is solidified from its liquid phase, where the solid phase consists, preferably, of a single crystal. These methods are broadly classified as meniscus-defined techniques and confined growth techniques [21, 23, 78]. The classification of the method is based on what determines the shape of the growing crystal. In meniscus-defined techniques, the meniscus between the melt and a gaseous environment shapes the crystal near the solidification interface. In confined growth techniques, the crystal takes the shape of the surrounding container known as an ampoule or crucible.

The most widely used meniscus-defined growth technique is the Czochralski method (CZ) shown in Figure 1.1. A seed crystal is dipped into a pool of the molten material, and the crystal grows as the seed is slowly pulled out of the pool. Cylindrical boules of single crystalline semiconductors such as silicon, germanium, gallium arsenide, and indium

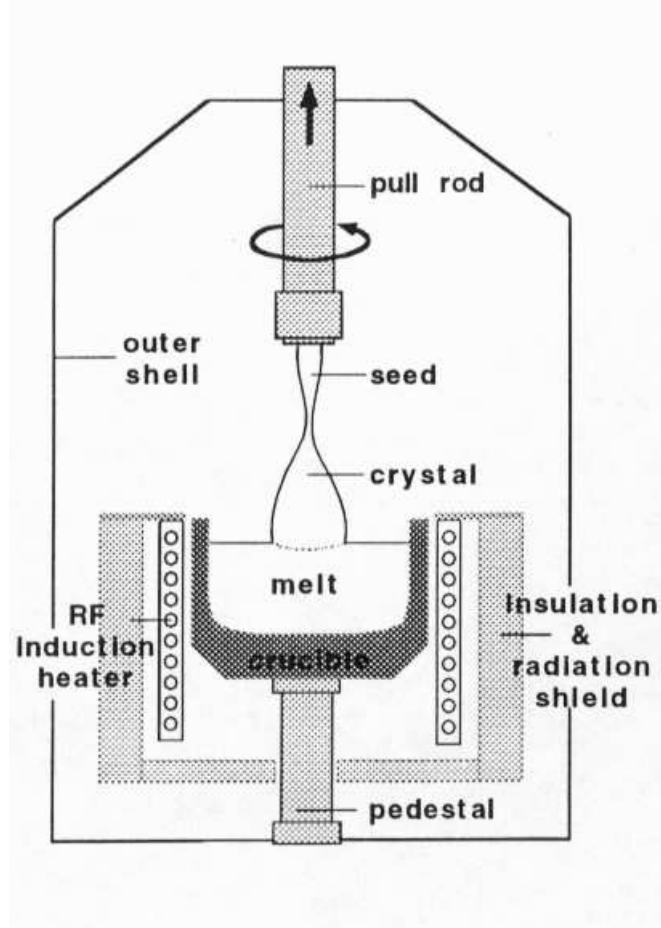


Figure 1.1: Schematic diagram of a Czochralski melt growth furnace.

phosphide are grown via variations of the CZ method [78, 171, 172] .

While large radius crystals can be grown using the CZ method, this method is unsuitable for the growth of certain materials. Some materials have high vapor pressures where stoichiometry control in the melt is difficult in a CZ system. With confinement, the melt can be subjected to a high over-pressure to prevent out gassing. Confined growth is also used for materials where low axial temperature gradients are required to produce low dislocation densities in the crystal [23]. The thermal environment that the growing crystal is subjected to, both melt and solid phases, can be carefully controlled in a confined system.

The Bridgman-Stockbarger method is the most commonly used confined growth technique where growth occurs in a container known as an ampoule or crucible. Bridgman furnaces are either aligned with gravity, vertical Bridgman (VB) systems, or perpendicular

to gravity, horizontal Bridgman (HB) systems. Figure 1.2 shows a schematic of a vertical Bridgman furnace where growth occurs upwards from the bottom, or vertical direction. The use of the end product influences the choice between VB and HB growth methods. VB furnaces are cylindrical making it more practical for most semiconductor uses. HB furnaces are more likely to be used when sheets of material are needed. Single crystalline gallium arsenide, indium phosphide, and cadmium zinc telluride are grown via variants of the Bridgman-Stockbarger system [64, 78, 119, 158].

A VB system is made up of a cylindrical furnace that surrounds a cylindrical ampoule. Raw materials are loaded into the ampoule, completely melted and homogenized through convective mixing. The ampoule is often sealed at the top with a gaseous overhead on top of the melt. The furnace temperatures are then adjusted to vary along the furnace length with temperatures hotter at the top and cooler at the bottom, temperatures relative to the crystal's melting point. Traditionally, solidification was induced by translating the ampoule through furnace from the hot end toward the cool end. A variation on the Bridgman-Stockbarger method is the gradient freeze method where heater powers of multi-zone heaters are varied with respect to time while the ampoule is held stationary. The crystal is removed from the ampoule after solidification. Crystallization is initialized by either placing a seed crystal at the cool end of melt or by heterogeneous nucleation by undercooling a portion of the melt. In general, HB furnaces work by the same theory as VB furnaces, but in the horizontal direction. This thesis focuses on crystal growth performed using the VB method and its gradient freeze analog as it is the method used to grow the types of CZT crystals we are interested in.

1.2 Physics of vertical Bridgman growth

All melt crystal growth systems are subject to the same physical governing forces, but we will analyze them in the context of a VB system. A VB furnace is a multi-domain system in which a wide range of physical phenomena occur over a large extent of length and time scales (atoms to cubic centimeters) [46]. These physical phenomena include coupled bulk transport of heat, mass, and momentum transfer along with thermodynamic phenomena such as phase change and equilibrium partitioning of solutes along the solidification front [43].

Heat transport plays an especially important role in that melt crystal growth furnaces are essentially heat transfer unit operations where a high temperature melt is solidified into a crystal at a lower temperature. Heat conduction, heat convection, and radiation

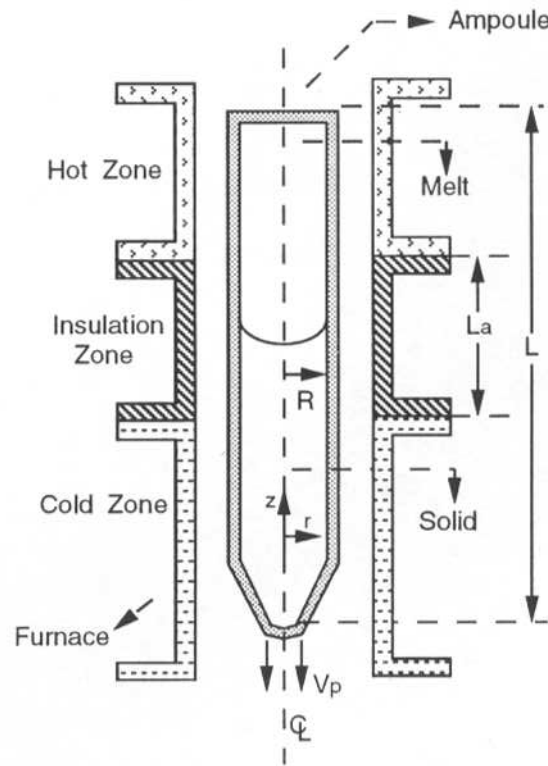


Figure 1.2: Schematic diagram of a vertical Bridgman melt growth furnace. Furnace imposed temperature profile is shown. The melting point temperature of the crystal lies between the temperature T_{hot} and T_{cold} . The ampoule is pulled downwards. Solidification occurs from the bottom to the top of the ampoule. The axis of the VB system is aligned with the gravity vector.

all play a role in VB systems. In the furnace domain heat transfer is governed by heat conduction and radiation. Both heat conduction and convection are important in the melt phase. In addition, momentum convection in the melt is driven by temperature gradients, buoyancy driven flow, which in turn affects the solute concentration in both the melt and solid phases. The solid phase is governed by heat conduction. Depending on the material, radiation effects may be present in the growing crystal. The macroscopic shape between the melt and crystal is influenced by the interaction of the material with the thermal environment. A thorough understanding of heat transfer and its interactions with mass transport, momentum transport, and phase change is needed for effective design of VB systems for the material of interest.

1.2.1 Heat transfer effects

Melt temperatures should be high enough to maintain the material in a molten state, but lower than temperatures where the melt either decomposes or vaporizes. Also, the axial temperature gradient that drives solidification should be large enough to prevent the occurrence of constitutional supercooling where the equilibrium melting temperature changes with variations in concentration at the melt-crystal interface. A wavy interface can develop that can become successively more unstable as the underlying driving forces increase. Cells, and even dendrites, can form [127]. The phenomenon is common in melt growth of multiple component materials [65]. On the other hand, large axial temperature gradients can result in undesirably high thermal stresses in the crystal which can cause in high dislocation density in the crystal [173]. Intense time-dependent buoyant flows driven by high thermal gradients can cause fluctuations in the temperature and dopant concentration which in turn can cause the melt-crystal interface to melt and resolidify. This leads to striations, or local compositional non-uniformities, in the crystal [89, 90].

The interplay between the heat transfer environment and physical properties affects the shape of the melt-crystal interface known as the *interface effect* [80]. The physical properties include heat conductivities of the melt, solid, and ampoule and the rate latent heat is released at the interface during solidification.

Three general macroscopic interface shapes can occur: flat, concave, and convex. Crystals grown with no radial temperature gradient and constant axial temperature gradient have flat interfaces, and no thermal stresses are produced [138]. Curved interfaces are more likely to have adverse interactions at the melt-crystal-ampoule tri-point that can result in twinning and higher dislocation density [173]. With an interface that is concave with re-

spect to the melt, these interactions tend to propagate toward the bulk crystal [21, 138]. Conversely, an interface that is convex with respect to the melt favors grain selection [31] and minimizes the potential for defects by propagating spurious crystal nucleated at the tri-junction toward the ampoule wall. Thus these new grains grow away from the bulk crystal, and self eliminate themselves against the ampoule wall. Much work has been conducted to understand what causes interface shape and how to achieve a desired shape by changing ampoule material, ampoule translation rate, furnace gradient, or adding a submerged heater [1, 20, 31, 33, 56, 57, 66, 80, 110, 130, 159, 180, 183, 192].

Under typical growth conditions in a VB system, CZT has a concave interface shape due to the smaller heat conductivity in the solid than the melt, and the relatively large amount of latent heat generated at the interface. The typical method of minimizing a concave curvature is to increase the applied temperature gradient. However, CZT is also brittle and prone to cracking caused by thermal stresses which puts an upper limit on the furnace temperature gradient.

1.2.2 Mass transfer effects and segregation

The spatial distribution of composition is important to crystal quality especially in multi-component semiconductor alloys such gallium-doped germanium and cadmium zinc telluride. In the case of CZT, zinc can be considered a dopant that is present in high concentrations. Mass transfer is driven by both mass convection and diffusion in the melt phase and diffusion in the solid crystal phase. During growth, equilibrium partitioning along the melt-crystal interface results in a depletion or accumulation in the melt of the dopant species depending on which phase the dopant is preferentially incorporated [65]. This partitioning effect coupled with diffusion and convection in the melt results in compositional non-uniformities, also known as segregation.

Compositional non-uniformities in the axial and radial are termed axial segregation and radial segregation, respectively. Axial segregation is controlled primarily by the dominate form of mass transfer in the melt: diffusion or convection. In diffusion-limited growth, the liquid concentration of the solute at the interface is constant leading a constant concentration in the solid along its entire length except for initial and final transients [175]. On the other extreme is perfect convective mixing in the melt. At this limit, the crystal growth rate is slow enough for mixing to remove all concentration gradients in the melt, but still fast enough to make solid diffusion negligible [173]. In this scenario, the solid concentration is no longer constant, but instead a function of solid fraction such that axial segregation

occurs. The majority of crystal growth situations lie between these two limits. Radial segregation is due to the curvature of the melt-crystal interface [32,37,38]. If the interface is concave toward the melt, the material in the center of the crystal solidifies before the material along the ampoule wall, and vice versa for a convex interface. In diffusion-limited growth curvature would not be a problem as the concentration at the interface would still be constant. However, mass convection plays a role in most crystal growth so that solute concentration is a function of solid fraction leading to radial segregation. If the interface is flat, radial segregation does not occur. Another consideration is time-varying melt flow that can result in striations in the crystal [89,90].

Constitutional supercooling also affects the solute distribution in the crystal [65,173]. In a situation where both diffusion and mass convection are driving forces in the melt, the composition in the melt changes with position. The equilibrium melting temperature is dependent on the local composition. Supercooling of the melt in front of the interface is caused by lowering of the melting temperature due to accumulation or depletion of a solute in melt because of partitioning at the interface. If a protuberance grows on the interface, it finds itself in an undercooled environment and can rapidly grow [65] and causes morphological instabilities at the interface. This in turn can lead to cellular or dendritic patterns [175]. Constitutional supercooling is prevented by either increasing the axial temperature gradient ahead of the interface or by lowering the growth rate.

1.2.3 Momentum transfer effects

Fluid flow in a melt growth system is mainly driven by buoyancy effects caused by temperature variations in the bulk of the melt. If the melt has a free-boundary with a gaseous phase, the surface tension gradient caused by temperature variations along this boundary can cause Marangoni flow in the melt [113]. The melt can exhibit a wide range of flow behavior from time-independent laminar flow to intense turbulent flow depending on the driving force, system geometry, and boundary conditions [13,125,126]. In general the flow is two-dimensional due to the inherent axisymmetry of the VB system. Three-dimensional flows are encountered in these systems due to symmetry breaking flow instabilities [123] or system imperfections such as tilting or asymmetric heating [100,105,187,189].

As previously mentioned, the interaction between fluid flow and species transport plays an important role in setting the composition profile of the crystal. A magnetic field can be imposed to damp intense time-dependent flow if the melt is a conducting material [87,103]. With a large enough magnetic field, the flows may be suppressed to the point where

diffusion-limited growth is reached. Growing crystals in a microgravity environment can also promote diffusion-limited growth since buoyant flows are suppressed considerably [28]. Accelerated crucible rotation is used to promote mixing in the melt and thereby reduce the possibility of constitutional supercooling [27, 75, 190]. Steady crucible rotation can be used to suppress flow in the melt to reduce axial segregation [100, 105, 194].

1.3 Modeling vertical Bridgman systems

The goal of crystal growth processes is to produce large, defect-free, compositionally uniform, single crystals. CZT crystals with all of these characteristics are never grown, and yields of useable material are very low. The crystals are often contaminated with crystallographic defects such as grain boundaries, twins, inclusions, precipitates, and microcracks [147]. Compositional in-homogeneities from axial and radial segregation also reduce crystal quality [79]. A considerable amount of research has been done to develop crystal growth processes to reduce crystallographic defects and improve composition uniformity [23]. Mathematical modeling in conjunction with empirical knowledge gained from experiments has been successfully used to enhance our understanding of the connection between processing conditions during crystal growth and crystal quality [193]

A wide range of physical phenomena occur during crystal growth, and over a vast extent of length and times scales [46]. It is not possible to simulate all of the physics within one model, so compromises must be made. Figure 1.3 illustrates different modeling approaches used for various length and time scales. Atomic behavior on the tens of nanometers scale is studied using *ab initio* molecular dynamics methods. Simulations at this scale are limited to systems of a few hundred atoms on time scales of tens of picoseconds. Larger ensembles and longer time scales are studied by employing molecular dynamics models based on classical potentials. The kinetic Monte Carlo method is utilized on systems of even larger length scales and longer time scales. At the bulk crystal growth level, continuous governing equations best describe the process where processing conditions can be connected to the final crystal structure. Work has also been done to link the kinetic growth on the crystal surface to bulk transport at this meso-scale length [40, 43, 98, 174]. Phase field methods have been used to model meso-scale phenomena such as dendritic growth [60]. The work presented in this thesis is based on continuum level physics.

Modeling of crystal growth processes began with simple one-dimensional models that had analytical or semi-analytical solutions [65]. Tiller et al. [164, 175] studied equilibrium partitioning in an unidirectional solidification model assuming diffusion in both the melt

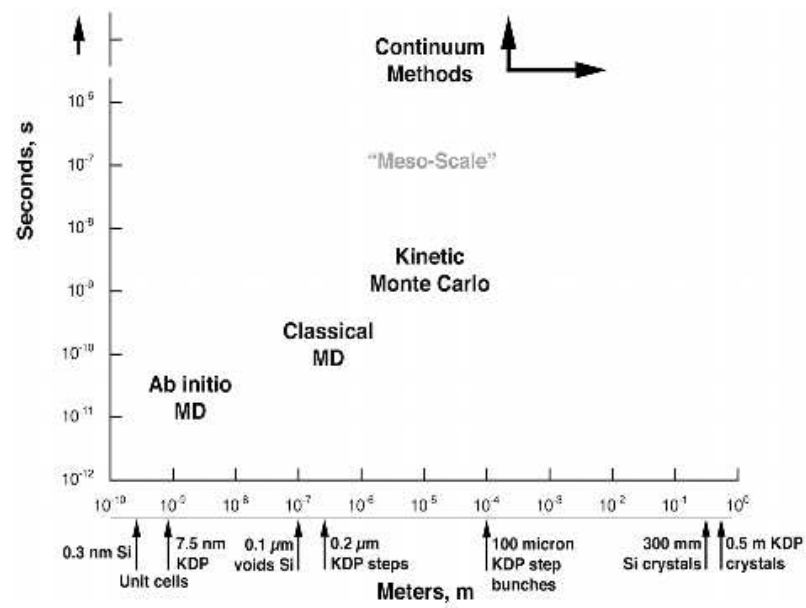


Figure 1.3: Modeling approaches for various length and time scales, reproduced from Derby [46].

and crystal, but ignoring convection in the melt which predicted an almost uniformly doped crystal. However, axial segregation is known to occur in the presence of melt convection. In the limit of complete mixing of the melt, the axial composition profile of the solute is given by the Scheil equation [155]. In reality convection is not always intense enough to justify the assumption of complete mixing in the melt. Therefore, to describe axial segregation caused by the interaction of diffusion and intermediate levels of convection, Burton, Prim, and Schlichter [25] formulated a model that invoked the concept of a stagnant-film as first proposed by Nernst [131]. The BPS model, as it's popularly known, proposed the existence of a stagnant-film of thickness δ ahead of the solidification interface in the melt, or a diffusion layer. This model assumed diffusion only within the stagnant-film and complete mixing outside this layer.

Though the BPS model is successful in fitting experimental axial segregation data in most cases, it has several drawbacks. In a system there is never a demarcation between well-mixed and quiescent parts of the melt. Convection and diffusion are important throughout the melt. Furthermore, use of a single aphysical parameter δ grossly over-simplifies the interactions between furnace design, heat transfer, buoyancy driven flow on solute partitioning in crystal growth systems. Coriell and Sekerka [37,38] showed that considerable radial segregation occurred in unidirectional solidification of binary alloys due to a curved melt-crystal interface, but BPS, being a one-dimensional model, cannot predict radial segregation. Wilcox [185] and Rosenberger and Müller [145] discussed the drawbacks of the BPS model. Brown and Kim [24] also highlighted the limitations of the BPS model, and showed that results from detailed numerical simulations of a two-dimensional model of Bridgman growth better matched solute segregation data from growth experiments on a gallium-doped germanium system.

By the 1980's, computing power had risen to a level where such detailed numerical simulations of two-dimensional models were possible. By exploiting the cylindrical geometry of the VB system, an assumption of azimuthal symmetry (or axisymmetry) reduced three-dimensional systems to two dimensions. These early efforts typically did not include momentum transport but only heat conduction and included only the melt, crystal and ampoule [1,31–33,66,80,130]. However, thermodynamic phenomena such as phase change, release of latent of heat of fusion, and equilibrium partitioning of solutes were also included as well as predicting the macroscopic shape of the melt-crystal interface. As the field matured, more sophisticated models were developed that coupled bulk transport phenomena of heat, mass, and momentum transfer [24,58,59,88,92–94,99,114,187,194]. Much of this work is summarized in a still-relevant review article by Brown [23], and more recent reviews

by Müller [124], Lan [102], and Yeckel and Derby [193].

Temperature fields predicted by the above two-dimensional models have also been used to solve for von Mises stresses in the crystal by assuming a continuum thermoelasticity model [18, 59, 82] for the crystal. Defects are known to be generated when the stresses in the system are greater than the critically resolved shear stress (CRSS) of the crystal [23]. Based on the thermoelastic stress analysis, qualitative predictions can be made on the density of defects in the crystal. Models capable of making quantitative predictions of defect density have also been developed [83].

Though the VB system has an azimuthal symmetry, three-dimensional effects are known to occur in these systems due to symmetry-breaking system imperfections caused by an ampoule tilt, ampoule misalignment with the furnace, and defective heating elements. Three-dimensional flows can also develop when the driving forces of buoyant convection pass some critical point [23]. Three-dimensional, quasi-steady-state, models [100, 105, 187, 189] have been used to predict three-dimensional flow patterns and solute segregation. Recent developments in parallel computing algorithms and hardware have helped in the implementation of these three-dimensional models.

These models, however, neglect modeling the furnace environment, and the effect of the surrounding furnace is included as an idealized heat transfer boundary condition along the ampoule walls. Other researchers, such as Müller and coworkers and Dupret et al. [55, 96], took a separate route and concentrated on developing highly detailed, two-dimensional models of the crystal growth furnaces that included radiative transport over complicated geometries. These efforts, however, have only limited capability of modeling the complex fluid dynamics that can occur in the melt phase. Instead, these models have been used to supply realistic heat transfer boundary conditions along the ampoule boundaries to models focusing on the coupled transport phenomena occurring in the growing crystal. This has allowed the analysis of the effect of the furnace domain on the crystal growth domain. Many of these approaches stopped short of seeking a self-consistent multi-scale solution [133, 188]. Recent efforts by Derby and coworkers have been made to rectify this situation by coupling pre-existing computer codes to exploit the expertise of the standalone models into one, self-consistent, multi-scale model [45, 136, 196].

Development of process control for the VB/VGF systems has been extremely slow because very few system variables can be measured in real time and *in-situ*. Chin and Carlson [33] and Seidensticker et al. [157] used open-loop simulations to investigate the effects of temperature boundary conditions on interface shape and their suitability to a control algorithm. Azuma et al. [6] developed an automatic feedback control system to keep

the temperature at the interface of silicon germanium, which is dependent on composition, during growth to control bulk composition. The ampoule translation rate was updated based on the position of the interface, which was detected automatically by images captured by an *in-situ* monitoring system via a charge-coupled-device camera. Sonda et al. [165,167] studied the feasibility of using feedback control to suppress oscillatory flows via crucible rotation with numerical modeling on indium antimonide. Batur and co-workers [11,12,168] fashioned a transparent, multi-zone furnace for the production of lead bromide. A video camera and real-time imaging software was able to observe the melt-crystal interface shape through the transparent furnace. The temperature profile was adjusted by comparing the observed interface shape to the desired shape. A good overview of recent work of active growth control is provided by Lan [102].

1.4 Thesis format

The purpose of this thesis can be divided into two parts. The first is to study novel VB processing applications for the improvement on CZT growth. The second is to develop new mathematical methods to improve our modeling capability of VB growth of semiconductor alloys.

Chapter 2 continues the introductory material by presenting a two-dimensional mathematical model of a VB crystal growth system for semiconductor alloys. This model consists of conservation equations for heat, momentum, and mass transport with an accurate representation of the phase change boundary. Details of the furnace are entered as idealized heat transfer boundary conditions. These equations form a set of time-dependent partial differential equations. The Galerkin finite element method is used to discretize the equations in the spatial coordinates, and an implicit trapezoidal time marching scheme is used to solve the temporal dependency. Newton's method is used to solve the resulting set of nonlinear algebraic equations. The governing equations and their implementation are discussed in detail.

Chapters 3 and 4 explore two novel processing methods for VB growth of CZT. In Chapter 3, we investigate the use of purposefully tilting the ampoule in order to improve zinc segregation in the crystal using a two-dimensional, planar model. Tilting the axis of the ampoule away from the direction of gravity strongly affects the structure and strength of melt convection, which, in turn, modifies the overall heat transfer and the shape of the melt-crystal interface. While these effects have relatively little influence on axial segregation, the pattern of lateral segregation is significantly changed by ampoule tilting. Optimal tilt

angles are found that flatten locally lateral compositional profiles, and that decrease a measure of overall segregation compared to that grown in a perfectly vertical ampoule.

In Chapter 4, numerical simulations are performed to assess the dependency of CZT interface shape on thermal gradients, and the feasibility of closed-loop control to control the interface shape in electrodynamic gradient (EDG) furnaces. Open-loop results explain how larger furnace gradients in these systems tend to flatten the shape of the melt-crystal interface. Convection dominates heat transfer through the melt, and mixing acts to radially homogenize the melt temperature field. These features do not significantly change with gradient conditions. In contrast, changing the rate of heat conduction through the solid, notable via changes to the furnace profile adjacent to the crystal, has an overriding influence on the interface shape. Increased gradients increase the ability for latent heat to be transported axially through the crystal and flatten the interface; however, the sensitivity of the shape of the interface to details of the furnace heating profile decreases as overall gradients increase.

Based on the results of the open-loop study, a simple proportional control algorithm is applied to a quasi-steady-state model to control the interface shape by adjusting the thermal gradient of the furnace temperatures. Three scenarios are enacted: actuation along the entire external boundary, along the solid portion only, and along the melt portion only. Actuation along the solid only proved to be the most successful of the three scenarios, and produced the unexpected benefit of a favorable convex interface shape.

Original mathematical modeling techniques not used before to study VB crystal growth systems are presented in Chapters 5 and 6. In Chapter 5, we explore a modular coupling approach of a melt crystal growth system, where the code CrysVUn is used to compute furnace-scale heat transfer and the code Cats2D is used to calculate melt fluid dynamics and phase-change phenomena. Modular coupling of numerical codes will enable the development of increasingly more sophisticated numerical models that maximize effectiveness, capability, and flexibility while keeping development costs down compared to building a monolithic multi-scale model from the bottom up. Previously, a coupling scheme known as the block Gauss-Seidel method was used to couple these two codes together which proved to be unreliable with regards to convergence [135, 136, 196]. Here, a strategy called the Approximate Block Newton method is applied to overcome the shortcomings of the block Gauss-Seidel method. A direct comparison of the two methods is demonstrated via a one-dimensional heat transfer problem that is representative of melt-crystal growth. In addition, results of a two-dimensional melt crystal growth model are presented, where the two codes, CrysVUn and Cats2D, are coupled using an Approximate Block Newton

method. Finally, initial results of a complex, multi-scale model of an industrial gradient freeze furnace constructed using the Approximate Block Newton method are presented. The Approximate Block Newton method proves to be far superior to the block Gauss-Seidel algorithm in terms of convergence and efficiency.

In Chapter 6, we use a Schur complement formulation to solve a two-dimensional, free-boundary, melt-crystal growth problem by exploiting the underlying structure of the discretized equation set. Instead of applying a direct or iterative technique to solve the full matrix equation resulting from Newton's method, the equation set is segregated into block form separating the field unknowns and the variables describing the location of the melt-crystal interface. The Schur complement of the Jacobian matrix is formed using block Gauss elimination. The resulting Schur complement problem can be solve using either direct or iterative methods, here we use full LU decomposition and GMRES respectively. We achieve quadratic convergence with the direct solver, and quadratic convergence is possible with GMRES depending on its convergence criterion. The Schur complement formulation is computationally slower than solving Newton's method with either a direct or iterative method, however, but scales with problem size in a manner similar to iterative techniques. Estimates on how the methods scale in three-dimensional problems are also made. Reductions in computation time can be made through approximations and preconditioning of the Schur complement problem.

Chapter 7 concludes the thesis by providing a summary followed by a brief overview of possible research directions for the future.

Chapter 2

Model and Numerical Implementation

Contents

2.1	Chapter Summary	17
2.2	Model and Governing Equations	18
2.2.1	Navier-Stokes Equations	19
2.2.2	Conservation of Energy	24
2.2.3	Conservation of Mass	26
2.3	Numerical Implementation	27
2.3.1	Discretization of Computational Domain	27
2.3.2	Discretization of Unknowns	28
2.3.3	Weighted Residual Equations	31
2.3.4	Deforming Boundary	34
2.3.5	Time Integration	36
2.3.6	Continuation	39
2.4	Conclusion	40

2.1 Chapter Summary

We use computational modeling to study bulk crystal growth of cadmium zinc telluride grown in vertical Bridgman systems and a few variations. The model presented in this

chapter focuses on transport phenomena occurring within the ampoule portion of the furnace and the melt and solid phases of the growing crystal. While many physical phenomena occur over a large range of length and time scales, we focus on continuum level mechanics in our model. Coupled transport phenomena of momentum, heat, and mass transfer and their attendant boundary conditions are simulated by discretized using the finite element method on a structured, rectangular mesh. Newton's method is then used to solve the resulting set of nonlinear, algebraic equations. A second-order trapezoid scheme is used to time integrate the problem. The models presented here are all solved using the software code Crystallization and Transport Simulator, or Cats2D [195] developed by Yeckel and Goodwin.

2.2 Model and Governing Equations

The models presented in this thesis are of either vertical Bridgman systems or its variant the gradient freeze system. Both methods are used to grow bulk crystals by solidification from a molten phase. The solidification process occurs in a cylindrical container called an ampoule or crucible. A schematic diagram representing a basic vertical Bridgman system is provided in Figure 2.1. Raw material is placed within the ampoule which is then placed within a cylindrical furnace. The raw material is then melted by setting the furnace temperature greater than the melting point temperature. Next, an axial temperature gradient is applied in the furnace along the external ampoule wall, where temperatures (relative to the melting point) are hotter on the top and lower toward the bottom. Solidification occurs by either vertically translating the ampoule down through the furnace as in traditional vertical Bridgman growth, or by varying heater powers to, in effect, translate the temperature gradient up the furnace as in a gradient freeze system.

Simulating an entire crystal growth furnace is extremely challenging due to the disparate length scales involved, and the multi-physics nature of the problem. Such multi-scale problems, and solution methods are discussed in Chapter 5. In this chapter, we discuss our local model which focuses on continuum transport phenomena that occurs solely in the growing crystal and its containing ampoule. Furnace effects are applied as boundary conditions along the outer ampoule wall.

Additional assumptions are made of the system to further reduce the complexity of our model. The ampoule (and the furnace) is assumed to be axisymmetric, or uniform in the azimuthal direction, to reduce the model from three to two dimensions. This greatly reduces the computational effort needed to solve a crystal growth problem. Furthermore,

the frame of reference is fixed with respect to the center of the bottom of the ampoule. Zinc is considered a dilute species within the CdTe, allowing the system to be modeled as a binary mixture with binary diffusion coefficients, which agrees with industry design. Material properties are assumed to be constant with respect to zinc concentration, which allows the species concentration equation to be decoupled from the heat and momentum transfer equations. Measurements of IR absorption coefficient at the growth temperature indicate that internal radiation is not important, therefore the solid and liquid phases of CdTe are assumed to be opaque [92]. All material properties are assumed to be independent of temperature and pressure, except for fluid density. The physical properties are listed in Table 2.1.

Figure 2.2 shows a computational domain of a vertical Bridgman system. The computational domain consists of three regions: the melt (Ω_m), the solid crystal (Ω_s), and the quartz ampoule (Ω_a). In addition to the three regions, there are also four distinct boundaries in the system. These boundaries are between the ampoule and furnace ($\delta\Omega_{af}$); ampoule and melt ($\delta\Omega_{am}$); ampoule and crystal ($\delta\Omega_{as}$); and the melt and crystal ($\delta\Omega_{ms}$). The origin of the cylindrical coordinates is the intersection of the ampoule and the crystal along the centerline.

The governing transport equations are derived from first principles as discussed in detail in [19], and are non-dimensionalized to have order one variables. Spatial coordinates are scaled by the inner radius of the ampoule, $L_o = R_i$. Temperature is defined relative to the melting point temperature T_{mp} , with a characteristic value based on the furnace temperature gradient multiplied by the characteristic length $T_o = \nabla T_f * L_o$. Concentration is non-dimensionalized by the initial zinc concentration within the melt, $c_o = c_i$. Time is scaled by viscous dissipation of the melt, and velocity is scaled by buoyancy. Table 2.2 lists the characteristic values used to non-dimensionalize the field equations. The governing equations are scaled by several dimensionless quantities which are listed in Table 2.3 based on properties of CZT.

2.2.1 Navier-Stokes Equations

Momentum conservation in the melt domain Ω_m is described by the Navier-Stokes (NS) equations for incompressible flow with the continuity constraint. The Boussinesq approximation is applied to encompass buoyancy effects. Using the characteristic quantities out-

lined in the previous section, the NS equations become:

$$\frac{\partial \mathbf{v}}{\partial t} + \sqrt{\text{Gr}} \mathbf{v} \cdot \nabla \mathbf{v} = \nabla \cdot \mathbf{T} + \sqrt{\text{Gr}}(1 - T)\mathbf{g}, \quad (2.1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2.2)$$

where

$$\mathbf{T} \equiv -p\mathbf{l} + (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \quad (2.3)$$

$\mathbf{T}$ is the stress tensor, $\mathbf{l}$ is the identity tensor, p is the dynamic pressure, and $\mathbf{g}$ is a unit vector in the direction of gravity. The velocity vector contains the radial and axial velocities, $\mathbf{v} = (v_r, v_z)$. The Grashof number, $\text{Gr} \equiv \rho^2 g L^3 T_o / \mu^2$, is the ratio of buoyancy force to viscous force acting on a fluid. In a vertical Bridgman system momentum is convected by free convection driven by buoyancy effects.

In melt crystal growth systems, buoyant forces are extremely important in that free convection due to density variation is the main driving force. With such nonisothermal systems, fluid density depends on temperature. To account for buoyant force, we insert an equation of state into the NS equations. Here we use the Boussinesq approximation equation of state which is a Taylor expansion of density with respect to density, $\rho(T)$ [19]. The approximation for $\rho(T)$ is substituted for the density term multiplying the gravity vector $\mathbf{g}$ only. Under these conditions we assume fluid motion is slow, and the inertial term is small compared to gravity [46, 193]. Density can also change due to solute concentration differences. Since we assume we are dealing with a dilute solution, the buoyancy term depends solely on a constant multiple of the temperature gradient by β , the solution expansivity of the melt.

To complete the NS equations, the appropriate boundary conditions must be applied. The no-slip boundary condition is applied along the melt/ampoule boundary ($\delta\Omega_{am}$) and the crystal/melt interface ($\delta\Omega_{ms}$). :

$$\mathbf{v} = 0 \quad \text{on} \quad \delta\Omega_{am} \text{ and } \delta\Omega_{ms}. \quad (2.4)$$

Applying the no-slip boundary condition at the crystal/melt interface is not intuitive on first glance. Since the density of the melt and the crystal are not appreciably different, the overall volume of the system does not change upon solidification. Furthermore, the interface does not actually translate with a given velocity. In reality, the crystal melt

Table 2.1: Physical properties of materials.

Properties	Symbol	Value	Units	Description
Thermal conductivity	k_s	1.0×10^{-2}	W/m K	CdTe crystal [160]
	k_m	2.0×10^{-2}	W/m K	CdTe melt [160]
	k_a	2.8×10^{-2}	W/m K	Quartz ampoule [177]
Density	ρ_s	5.68	g/cm ³	CdTe crystal [68]
	ρ_m	5.68	g/cm ³	CdTe melt [68]
	ρ_a	2.20	g/cm ³	Quartz ampoule [177]
Heat capacity	$C_{p,s}$	1.595×10^{-1}	J/g K	CdTe crystal [158]
	$C_{p,m}$	1.870×10^{-1}	J/g K	CdTe melt [158]
	$C_{p,a}$	7.700×10^{-1}	J/g K	Quartz ampoule [177]
Heat of fusion	ΔH_f	209.2	J/g	CdTe [197]
Melting temperature	T_{mp}	1365	K	CdTe [197]
Thermal expansivity	β_T	5.0×10^{-4}	K ⁻¹	CdTe [147]
Kinematic viscosity	ν	8.0×10^{-3}	cm ² /s	CdTe melt
Diffusion coefficient	D_s	5.0×10^{-7}	cm ² /s	Zn in CdTe crystal [5]
	D_m	1.0×10^{-4}	cm ² /s	Zn in CdTe melt [156]
Segregation coefficient	K	1.35		Zn in CdTe [118]
Surface tension gradient	γ_T	-1.4×10^{-1}	g/K s ²	[184]

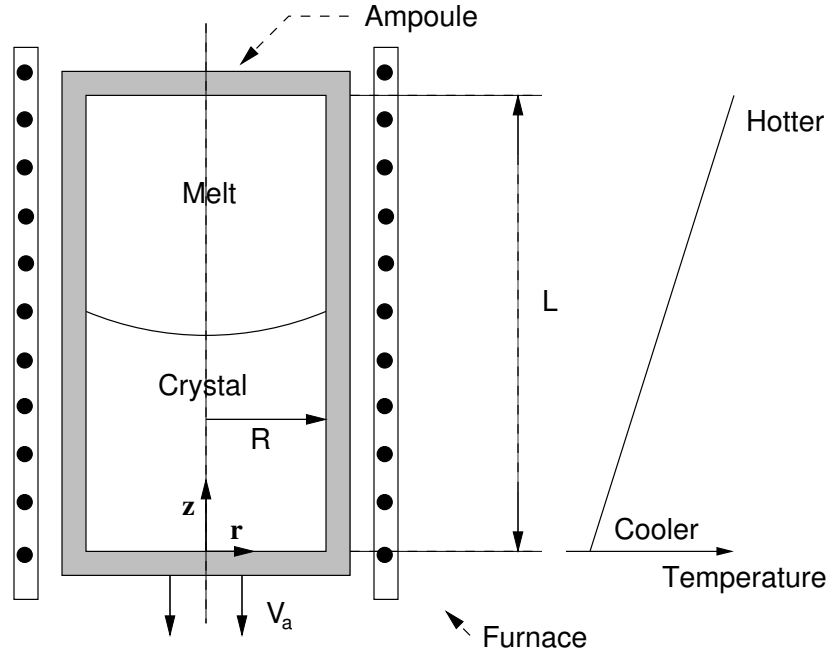


Figure 2.1: Schematic of the vertical Bridgman crystal growth system

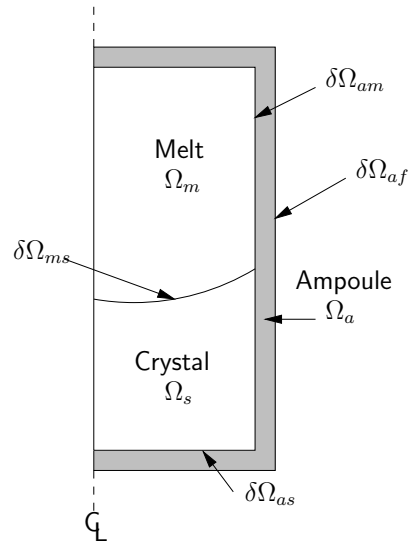


Figure 2.2: Schematic of the computational domain

Table 2.2: Characteristic values

Symbol	Description	Scale Factor	Comments
L_o	Length	R_i	Ampoule inner radius
T_o	Temperature	ΔT	Furnace temperature gradient * L
V_o	Velocity	$\sqrt{\beta g L_o T_o}$	based on vertical heated plate in an infinite bath
t_o	Time	$\frac{\rho L_o^2}{\mu}$	viscous dissipation of melt
p_o	Pressure	$\frac{\mu V_o}{L_o}$	based on melt dynamic viscosity
c_o	Concentration	c_i	initial melt concentration

Table 2.3: Dimensionless groups of CZT for $L_o = 2.54$ cm and $\Delta T_f = 2.33$ K/cm

Group	Symbol	Definition	Phase
Grashof number	Gr	$\rho^2 \beta g T_o L_o^3 / \mu^2$	7.4327×10^5 melt
Marangoni number	Ma	$\gamma_T L_o / \mu V_o$	-6.7150×10^1 melt
Prandtl number	Pr_m	ν / α_m	4.25×10^{-1} melt
	Pr_s	ν / α_s	7.25×10^{-1} crystal
	Pr_a	ν / α_a	$4, 84 \times 10^{-1}$ ampoule
Thermal Peclet number	$Pe_{t,m}$	$V_o L_o / \alpha_m$	3.6629×10^2 melt
Stefan number	Sf	$H_f / C p_m T_o$	1.8903×10^2 melt
Schmidt number	Sc_m	ν / D_m	8.0×10^1 melt
	Sc_s	ν / D_s	1.6×10^4 crystal
Mass Peclet number	$Pe_{m,m}$	$V_o L_o / D_m$	6.8971×10^4 melt

freezes in place [193]. Symmetry conditions are applied along the system centerline, $r = 0$,

$$\mathbf{v} \cdot \mathbf{e}_r|_{r=0} = 0 \quad (2.5)$$

$$\left. \frac{\partial(\mathbf{v} \cdot \mathbf{e}_z)}{\partial r} \right|_{r=0} = 0 \quad (2.6)$$

where $\mathbf{e}_i$ is the unit vector in either the r - or z -direction.

In many systems, there is a gaseous overhead within the ampoule above the melt phase. A more specialized boundary condition is applied in such cases, which embodies the effects of surface tension at the interface. We use a simplified condition by assuming that the stress induced by the gas phase is negligible, such that the momentum balance in the gas phase is ignored and its stress term is set equal to an ambient pressure (here set to zero). We further assume that gravity keeps the free boundary nearly flat, in which case the normal momentum balance is automatically satisfied, and only the tangential component must be accounted for. The force balance at the melt/gas interface becomes:

$$\mathbf{n} \cdot \mathbf{T}|_l = (\text{Mat} \cdot \nabla T)\mathbf{t} \quad (2.7)$$

where $\text{Ma} \equiv \gamma_T T_o / \mu V_o$ is the Marangoni number, $\mathbf{t}$ is the unit tangent vector, μ is the melt dynamic viscosity, and γ_T is the linear variation of surface tension with respect to temperature. This tangential stress caused by the surface tension gradient is known as the Marangoni effect [193].

2.2.2 Conservation of Energy

Energy transport occurs by conduction and convection within the melt, and conduction within the crystal and ampoule. The non-dimensional form of the energy conservation equation is

$$\text{Pr}_m \frac{\partial T}{\partial t} + \text{Pe}_{t,m} \mathbf{v} \cdot \nabla T = \nabla^2 T \quad (2.8)$$

in the melt, and

$$\text{Pr}_i \frac{\partial T}{\partial t} = \nabla^2 T \quad (2.9)$$

in all other materials, where the index m indicates the melt phase, and i designates the solid crystal s or ampoule a . The melt energy equation assumes that transport is dominated by

natural convection versus diffusion-induced flow, which is valid for the vertical Bridgman crystal growth system. The Prandtl number is defined as $Pr_i \equiv \nu_m/\alpha_i$ where ν_m is the kinematic viscosity of the melt, and $\alpha_i \equiv k_i/\rho_i C p_i$ is the thermal diffusivity of phase i , and used as a measure of the viscous diffusivity to the heat diffusivity. The thermal Peclet number, $Pe_{m,i} \equiv VL/\alpha_i$, is a measure of the convective heat transport to the diffusive heat transport. The characteristic values of velocity, V_o , and length, L_o are defined in Table 2.2.

Latent heat of fusion is released at the melt-crystal interface, $\delta\Omega_{ms}$, upon crystallization as a function of the local interface velocity. Therefore, the Stefan or latent-heat boundary condition is applied:

$$\mathbf{n} \cdot (-\kappa \nabla T|_s + \nabla T|_m) = \Lambda Pr_m Sf_m \mathbf{n} \cdot \mathbf{V}_{ms} \quad \text{at} \quad \delta\Omega_{ms} \quad (2.10)$$

where the variables and operators are in non-dimensional. $\kappa = k_s/k_m$ is the ratio of thermal conductivities, $\Lambda = \rho_s/\rho_m$ is the density ratio, and $\mathbf{V}_{ms} = \partial \mathbf{x}/\partial t$ is the velocity of the melt-crystal interface. As can be seen in Equation (2.10), $\partial \mathbf{x}/\partial t$ is non-dimensionalized by ν/L_o . The Stefan number, $Sf_i \equiv H_f/C_{p,i}T_o$, is the ratio of latent heat to heat capacity, where H_f is the latent heat and $C_{p,i}$ is the heat capacity of phase i .

In addition to the Stefan condition, a melting-point isotherm condition is applied at the melt-crystal interface, which implicitly sets the shape of this boundary. The isotherm condition sets the location of the interface as that of the melting point temperature T_{mp} . Since the temperature is defined relative to T_{mp} , the boundary condition becomes

$$T = T_{mp} = 0 \quad \text{at} \quad \delta\Omega_{ms} \quad (2.11)$$

Furnace contributions to heat transfer are included by a boundary condition along the external ampoule wall $\delta\Omega_{ag}$. In experimental vertical Bridgman growth, many different furnace temperature profiles are utilized in crystal growth. For illustrative purposes, we will employ a furnace temperature profile that is linear with axial position:

$$T_f = T_{mp} + \frac{T_H - T_L}{z_H - z_L}(z - z_{mp}) + V_a t \quad (2.12)$$

where T_f is the furnace temperature, z_H is the axial position of the top of the ampoule, z_L is the bottom of the ampoule, and T_H and T_L are the corresponding temperatures ($T_L < T_H$). A reference position, z_{mp} , is set as the axial location of the melting point temperature. The time dependent term simulates translation of the ampoule through the furnace, with V_a as the ampoule translation rate. Heat flux at the external ampoule wall

can be approximated using Newton's law of cooling:

$$\mathbf{n} \cdot \nabla T = h(T - T_f) \quad \text{at} \quad \delta\Omega_{af} \quad (2.13)$$

where $\mathbf{n}$ is the unit normal vector, h is the dimensionless heat transfer coefficient (Biot number), and $T_f = z$ is the applied, dimensionless, furnace temperature.

As with the momentum balance, a symmetry condition is applied at the system centerline:

$$\left. \frac{\partial T}{\partial r} \right|_{r=0} = 0 \quad (2.14)$$

Temperature is assumed to be continuous across all system boundaries.

2.2.3 Conservation of Mass

Similar to the energy transport equation, mass transport is described by a convection-diffusion equation. Zinc is assumed to be a dilute species in CdTe, and is subject to pseudo-binary conservation equations:

$$\text{Sc}_m \frac{\partial c}{\partial t} + \text{Pe}_{m,m} \mathbf{v} \cdot \nabla c = \nabla^2 c \quad (2.15)$$

in the melt, and

$$\text{Sc}_s \frac{\partial c}{\partial t} = \nabla^2 c \quad (2.16)$$

in the CdTe solid, where c is dimensionless concentration of zinc. The Schmidt number, $\text{Sc}_i \equiv \nu/D_i$, is a measure of viscous to species diffusivity. The mass Peclet number, $\text{Pe}_m, i \equiv VL/D$, measures the convective species transport of the diffusive species transport. In both dimensionless quantities, i indicates phase.

Segregation boundary conditions that are analogous to the Stefan and the isotherm boundary condition are imposed at the melt-crystal interface:

$$\mathbf{n} \cdot (-\nabla c|_m + D\nabla c|_s) = \text{Sc}_m(1 - K)\mathbf{n} \cdot \mathbf{V}_{ms} \quad \text{at} \quad \delta\Omega_{ms} \quad (2.17)$$

$$c|_s = Kc|_m \quad \text{at} \quad \delta\Omega_{ms} \quad (2.18)$$

where $D = D_s/D_m$ and K is the equilibrium segregation coefficient.

Again, a symmetry condition is applied at the system centerline

$$\left. \frac{\partial c}{\partial r} \right|_{r=0} = 0. \quad (2.19)$$

Furthermore, the melt-ampoule interface, crystal-ampoule interface, ampoule are assumed to be impermeable to solute transport:

$$\mathbf{n} \cdot \nabla c = 0 \quad \text{on} \quad \delta\Omega_{am} \text{ and } \delta\Omega_{as} \quad (2.20)$$

$$c = 0 \quad \text{on} \quad \Omega_a. \quad (2.21)$$

2.3 Numerical Implementation

The equations governing macroscopic transport in the vertical Bridgman growth system are nonlinear, coupled, partial differential equations (PDE). Due to the complexity of the equations, it is not possible to solve them analytically, therefore they are solved numerically. The computational domain is spatially discretized, or partitioned into small elements. The PDEs are spatially discretized based on the domain discretization. The Galerkin Finite Element Method (GFEM) is used to discretize the PDEs and time is integrated using numerical techniques [76]. Discretization of the equations transforms the equations from PDEs to ordinary differential equations (ODE). The goal of this thesis is not to be an account of the many intricacies of GFEM, but on how this method can be used to model a crystal growth system. For a more detailed exposition on how to apply GFEM to fluid dynamics problems, the reader is referred to [71, 76].

2.3.1 Discretization of Computational Domain

To solve PDEs using the finite element method, the computation domain is first discretized. Figure 2.3 shows how a one dimensional domain is discretized. The domain is divided into smaller subdomains called elements while the endpoints of each element are defined by nodes. The collection of elements and nodes is called a “mesh”. Discretizing a two dimensional computational domain is an extension of the one dimensional example. For quadrilateral elements, each element is defined by four corner nodes. Additional nodes can be added to the interior of an element to improve computational accuracy.

The computational domain of the vertical Bridgman growth system has an added complication of different materials making up the domain. Also, not all transport mechanisms apply throughout the domain. Therefore, the domain is divided into large subdomains representing different regions of the problem, i.e. the melt, crystal, and ampoule which are in turn divided by elements.

As the number of elements is increased, or the mesh is further refined, the accuracy of

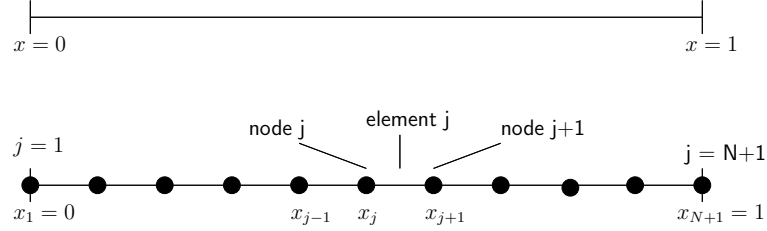


Figure 2.3: Example of a discretized domain in one dimension

the approximate solution increases. In the limit as the elements become infinitely small, the “true” solution is achieved. The increase in accuracy comes at the cost of computational resources and time. Therefore, a compromise must be made between accuracy and available resources.

2.3.2 Discretization of Unknowns

The unknowns are also discretized according to the discretization of the computational domain. The unknowns (velocity, pressure, temperature, and concentration) are represented as a linear combination of finite element basis functions. These basis functions, ϕ^i and ψ^i , are contained within the Sobolev space $\mathcal{H}$ [76] defined as:

$$\mathcal{H} = \{w : \int_{\Omega} |\nabla w|^2 dx < \infty \text{ and } w = 0 \text{ for } w \in \Omega\} \quad (2.22)$$

where Ω is a domain, $\delta\Omega$ its boundary, and w a basis function. The basis functions used here are Lagrange polynomials

- Q_2 polynomials: biquadratic and continuous
- P_1 polynomials: linear and continuous
- P_{-1} polynomials: linear and discontinuous

which belong to the Sobolev space.

The basis functions are associated with a specific node in the finite element mesh. Furthermore, the basis function is non-zero only in the elements that share a given node, and zero over the rest of the domain making a set of nearly orthogonal set of functions. At the node itself ($node^i$), the basis function is unity and zero at all other nodes. The velocity

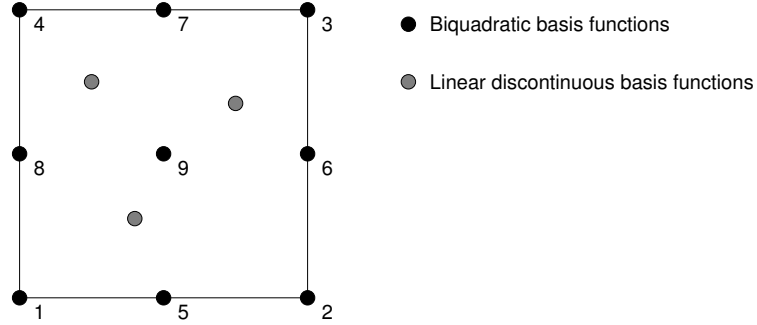


Figure 2.4: A sample element and its nodes. Dark nodes are associated with biquadratic basis functions, and light nodes are associated with linear discontinuous basis functions.

field, $\mathbf{v}$, is then approximated as

$$\mathbf{v}(t, \mathbf{x}) \approx \hat{\mathbf{v}} = \sum_{j=1}^{N_v} \mathbf{v}^j(t) \phi_v^j(\mathbf{x}) \quad (2.23)$$

where $\mathbf{v}^j$ is the unknown velocity at the j^{th} node, N_v is number of nodes in the melt subdomain, and $\mathbf{x}$ is the position vector. The concentration is approximated as

$$c(t, \mathbf{x}) \approx \hat{c} = \sum_{j=1}^{N_c} c^j(t) \phi_c^j(\mathbf{x}) \quad (2.24)$$

where c^j is the concentration at the j^{th} node and N_c is the number of nodes in the melt and crystal subdomains. Similarly, the temperature is approximated as

$$T(t, \mathbf{x}) \approx \hat{T} = \sum_{j=1}^{N_T} T^j(t) \phi_T^j(\mathbf{x}) \quad (2.25)$$

where T^j is the temperature at the j^{th} node, and N_T is the number nodes over the entire domain. The basis functions used to represent velocity, concentration, and temperature are biquadratic. The domain is represented by nine nodes per element.

The pressure cannot be approximated in the same manner as the above field variables. Pressure does not appear in the continuity equation (Equation (2.2)), and only its gradient appears in the momentum equation (Equation (2.1)). Therefore, the pressure cannot be solved explicitly and creates an underlying constraint of the momentum equation. The constraint is overcome by “enriching” the velocity space by either adding extra velocity

Table 2.4: Types of basis functions

Unknown	Symbol	Basis Function	Type
$\mathbf{v}$	velocity	ϕ_v	continuous, biquadratic
p	pressure	ψ_p	discontinuous, linear
T	temperature	ϕ_T	continuous, biquadratic
c	concentration	ϕ_c	continuous, biquadratic
ξ	nodal position	ζ	continuous, linear
η	nodal position	ζ	continuous, linear

nodes or minimizing the number of pressure nodes. This technique is known as the “inf-sup” or “Babuska-Brezzi” condition. Elements that pass the stability test will converge “optimally” and without spurious pressure behavior. Those failing the test *may* not converge [71]. With quadrilateral elements, the better choice is to minimize the pressure space. This work uses linear discontinuous basis functions, ψ^i , and three nodes per elements which are associated with the value of the pressure and pressure gradient at the element center,

$$p(t, \mathbf{x}) \approx \hat{p} = \sum_{j=1}^{N_p} p^j(t) \psi_p^j(\mathbf{x}). \quad (2.26)$$

Figure 2.4 shows a schematic of a sample element and its associated nodes.

The spatial and temporal dependencies of the unknowns are divided between the basis functions and their coefficients. For example, temperature is discretized as in Equation (2.25). The temperature’s spatial dependency is captured by the basis functions $\phi_T^j(x)$, and its temporal dependency by the coefficients $T^j(t)$. Therefore, the spatial and time derivatives of the temperature are approximated as:

$$\frac{\partial T}{\partial x}(t, \mathbf{x}) \approx \sum_{j=1}^{N_T} T^j \frac{d\phi_T^j}{d\mathbf{x}} \quad (2.27)$$

$$\frac{\partial T}{\partial t}(t, \mathbf{x}) \approx \sum_{j=1}^{N_T} \frac{dT^j}{dt} \phi_T^j \quad (2.28)$$

The discretized unknowns and their derivatives are substituted into the governing equations resulting in a set of discretized PDEs. Table 2.4 summarizes the types of basis functions used in this model.

2.3.3 Weighted Residual Equations

Finite element methods comprise a subset of the method of weighted residuals [63], the goal of which is to obtain an approximate solution by minimizing the error between the exact solution and the numerical approximation. To accomplish this, the equation in residual form (all of the terms are moved to the same side) is made orthogonal to a set of weighting functions that are mutually orthogonal. If the residual is orthogonal to a complete set of weight functions, then the residual itself is equal to zero and the solution to the equations is found [76]. The weighted residual, $\mathcal{R}$ has the form

$$\mathcal{R} = \int w f(x) dx = 0 \quad \forall w \quad (2.29)$$

where w is the weight function, and $f(x)$ is the residual.

It is not always feasible to use a complete set of mutually orthogonal weighting functions, so a nearly complete, finite set $\mathcal{W}$ is used instead. As mentioned above, the basis functions used to discretize the unknowns is a set of nearly orthogonal basis functions. GFEM uses these basis functions as weighting functions [76]: $w = \phi^i$ or ψ^i .

The transport equations are put into residual form, multiplied by the weight functions, and integrated over the computational domain. The weighted residuals of the governing equations are:

$$\mathcal{R}_{\mathbf{v}}^k = \int_V \phi_v^k \left(\frac{\partial \hat{\mathbf{v}}}{\partial t} + \sqrt{\text{Gr}} \mathbf{v} \cdot \nabla \hat{\mathbf{v}} - \nabla \cdot \hat{\mathbf{T}} - \sqrt{\text{Gr}}(1 - \hat{T})\mathbf{g} \right) dV = 0 \quad (\text{momentum}) \quad (2.30)$$

where $\hat{\mathbf{T}} = -\hat{p}\mathbf{I} + (\nabla \hat{\mathbf{v}} + (\nabla \hat{\mathbf{v}})^T)$

$$\mathcal{R}_p^k = \int_V \psi_p^k (\nabla \cdot \mathbf{v}) dV = 0 \quad (\text{continuity}) \quad (2.31)$$

$$\mathcal{R}_T^k = \int_V \phi_T^k \left(\text{Pr}_i \frac{\partial \hat{T}}{\partial t} + \text{Pe}_{t,i} \hat{\mathbf{v}} \cdot \nabla \hat{T} - \nabla^2 \hat{T} \right) dV = 0 \quad (\text{energy}) \quad (2.32)$$

$$\mathcal{R}_c^k = \int_V \phi_c^k \left(\text{Sc}_i \frac{\partial \hat{c}}{\partial t} + \text{Pe}_{m,i} \hat{\mathbf{v}} \cdot \nabla \hat{c} - \nabla^2 \hat{c} \right) dV = 0 \quad (\text{species}). \quad (2.33)$$

where V indicates volume integrals in real space. Note that the weighting functions in the continuity equation are the pressure basis functions. The pressure acts as a Lagrange multiplier for the incompressibility constraint by “adjusting itself” instantaneously in time and everywhere in space [71].

The next step in the GFEM is to recast the above equations into the weak form, also known as the variational form. Converting the equations to the weak form reduces the order of the derivatives which increases the accuracy of the equation, however information is lost with every derivative taken. The equations are put into the weak form using the identity chain rule:

$$\nabla \cdot \phi_T^k \nabla \hat{T} = \nabla \phi_T^k \cdot \nabla \hat{T} + \phi_T^k \nabla^2 \hat{T}, \quad (2.34)$$

and the divergence theorem of Green, Gauss, and Ostrogradskii (GGO):

$$\int_V \nabla \cdot \phi_T^k \nabla \hat{T} dV = \int_A \phi_T^k \mathbf{n} \cdot \nabla \hat{T} dA. \quad (2.35)$$

where A indicates an area integral. The volume integral over $\phi \nabla^2 \hat{T}$ becomes:

$$\int_V \phi_T^k \nabla^2 \hat{T} dV = - \int_V \nabla \phi_T^k \cdot \nabla \hat{T} dV + \int_A \phi_T^k \mathbf{n} \cdot \nabla \hat{T} dA. \quad (2.36)$$

This procedure is analogous to integration by parts in one dimension. The governing equations in weak form are:

$$\begin{aligned} \underline{\mathcal{R}}_{\mathbf{v}}^k = \int_V \phi_v^k \left(\frac{\partial \hat{\mathbf{v}}}{\partial t} + \sqrt{\text{Gr}} \mathbf{v} \cdot \nabla \hat{\mathbf{v}} - \sqrt{\text{Gr}} (1 - \hat{T}) \mathbf{g} \right) dV \\ + \int_V \nabla \phi_v^k \cdot \hat{\mathbf{T}} dV - \int_A \phi_v^k (\mathbf{n} \cdot \hat{\mathbf{T}}) dA = 0 \quad (\text{momentum}) \end{aligned} \quad (2.37)$$

$$\mathcal{R}_p^k = \int_V \psi_p^k (\nabla \cdot \mathbf{v}) dV = 0 \quad (\text{continuity}) \quad (2.38)$$

$$\begin{aligned} \mathcal{R}_T^k = \int_V \phi_T^k \left(\text{Pr}_i \frac{\partial \hat{T}}{\partial t} + \text{Pe}_{t,i} \hat{\mathbf{v}} \cdot \nabla \hat{T} \right) dV \\ + \int_V \nabla \phi_T^k \cdot \nabla \hat{T} dV - \int_A \phi_T^k (\mathbf{n} \cdot \nabla \hat{T}) dA = 0 \quad (\text{energy}) \end{aligned} \quad (2.39)$$

$$\begin{aligned} \mathcal{R}_c^k = \int_V \phi_c^k \left(\text{Sc}_i \frac{\partial \hat{c}}{\partial t} + \text{Pe}_{m,i} \hat{\mathbf{v}} \cdot \nabla \hat{c} \right) dV \\ + \int_V \nabla \phi_c^k \cdot \nabla \hat{c} dV - \int_A \phi_c^k (\mathbf{n} \cdot \nabla \hat{c}) dA = 0 \quad (\text{species}). \end{aligned} \quad (2.40)$$

Equations (2.37)-(2.40) make up the set of differential algebraic equations used in

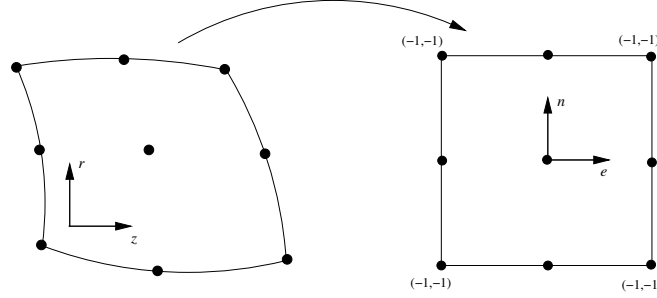


Figure 2.5: Parametric mapping to a unit square parent element

GFEM to solve the problem. Note that the residual equation for momentum is a vector, $\underline{\mathcal{R}}_{\mathbf{v}}$ with a component for each velocity direction.

The types of boundary conditions are natural (flux), essential (Dirichlet). Flux boundary conditions are used in the surface integrals of the weak form. Essential boundary conditions replace the entire residual equation. For example, applying the flux boundary condition Equation (2.20) of impermeable solute transport to the surface integral Equation (2.40)

$$\int_A \phi^k(\mathbf{n} \cdot \nabla \hat{c}) dA = 0 \quad \text{on} \quad \delta\Omega_{am}, \delta\Omega_{as}. \quad (2.41)$$

The isotherm boundary condition is an example of an essential boundary condition. In place of an equation of the form in Equation. (2.39), we have

$$\mathcal{R}_T^k = \int_A \phi^k(\hat{T} - T_{mp}) dA = 0. \quad (2.42)$$

The integrals of the weak form are solved numerically using Gaussian quadrature. Also, to speed up computation time, isoparametric mapping is used to “map” the arbitrary, computational elements in r, z to a square parent element in ξ, η coordinates, as in Figure 2.5. This mapping is given by:

$$(z, r) = \sum_{k=1}^{N_{el}} (z^k, r^k) \phi^k(\xi, \eta) \quad (2.43)$$

where $\phi^k(\xi, \eta)$ are bilinear basis function as described in [76], r^k, z^k are the nodal coordinates in the computational domain, and N_{el} is the number of nodes per element.

Isoparametric mapping also allows us to transform volume and area integrals from real

space to parent element space using:

$$\int_V (\dots) dV = \int_{-1}^1 (\dots) r J d\xi d\eta; \quad J = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}, \quad (2.44)$$

where $r = 1$ for planar coordinates and $r = \text{radius}$ for cylindrical coordinates. The transformation Jacobian, J is computed using Equation (2.43), and the integrals are evaluated using 3×3 Gaussian quadrature [81].

Area integrals on boundaries of constant η are transformed using:

$$\int_A (\dots) dA = \pm \int_{-1}^1 (\dots) r S_\xi d\xi; \quad S_\xi = \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2} \quad (2.45)$$

The leading sign is positive for $\eta = -1$, and negative for $\eta = 1$. Area integrals on boundaries of constant ξ are transformed using:

$$\int_A (\dots) dA = \pm \int_{-1}^1 (\dots) r S_\eta d\eta; \quad S_\eta = \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2} \quad (2.46)$$

where the leading sign is positive for $\xi = 1$ and negative for $\xi = -1$. The signs are determined by the convention that line integrals proceed counter-clockwise over the domain. S_ξ and S_η are computed using Equation (2.43), and the area integrals are evaluated using 3 point Gaussian quadrature. By implementing isoparametric mapping, the residual integrals are computed easily and efficiently using generic code structure.

2.3.4 Deforming Boundary

Difficulty arises in accurately defining the melt-crystal interface. The interface shape is not known *a priori* and may change as the crystal grows [193]. In this work, a deforming mesh is used in order to compensate for the moving boundary. The initial mesh is generated by an algebraic method that uses quadrilateral elements. The mesh deforms according to the strategy known as elliptic mesh generation proposed by Winslow [186], and applied as by Christodoulou and Scriven [34] with boundary conditions developed by de Santos [42]. Elliptic mesh generation parameterizes the node locations using a pair of elliptic partial

differential equations. For two dimensions,

$$\nabla \cdot D_\xi \nabla \xi = 0 \quad (2.47)$$

$$\nabla \cdot D_\eta \nabla \eta = 0 \quad (2.48)$$

where D_ξ and D_η are node diffusivities that control the refinement of the mesh. For a non-uniform mesh, the diffusivities are not constant. The diffusivities can be calculated from an initial mesh that is generated algebraically:

$$D_\xi = \frac{\partial S}{\partial \xi} = \sqrt{\left(\frac{\partial r}{\partial \xi}\right)^2 + \left(\frac{\partial z}{\partial \xi}\right)^2}, \quad D_\eta = \frac{\partial S}{\partial \eta} = \sqrt{\left(\frac{\partial r}{\partial \eta}\right)^2 + \left(\frac{\partial z}{\partial \eta}\right)^2} \quad (2.49)$$

In this manner, the relative distribution of elements sizes is preserved while the domain deforms.

The residual elliptic mesh equations, Equations (2.47) and (2.48), are also put into weighted residual form. The basis functions used in isoparametric mapping are used as the weight functions:

$$\int_V \zeta^k \nabla \cdot (D_\xi \nabla \xi) dV = 0 \quad (2.50)$$

$$\int_V \zeta^k \nabla \cdot (D_\eta \nabla \eta) dV = 0 \quad (2.51)$$

The residual equations are written such that the computational coordinates (ξ, η) are dependent on the physical coordinates (z, r) . In practice, it is a necessary to perform a coordinate transformation so that (ξ, η) are the independent variables. There are two possible boundary conditions for Equations (2.47) and (2.48). One boundary condition assigns boundary values for D_ξ and D_η . The second boundary condition keeps the boundary nodes on the boundary.

The time derivatives also need to be rewritten to accommodate the moving mesh. In the continuous governing equations, the time derivatives are taken with respect to the computational domain reference frame or Eulerian frame. On the other hand, the time derivatives of the basis function coefficients in the discretized equations are taken with respect to the parent element frame which moves at the velocity of the nodes. The time derivatives in the discretized frame are converted from the Eulerian frame to the parent

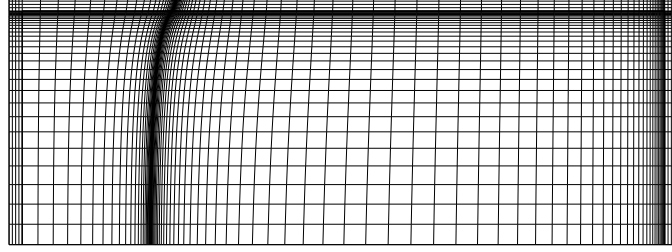


Figure 2.6: Sample mesh with deformed boundary at melt-crystal interface.

element frame:

$$\frac{\partial \hat{\mathbf{v}}}{\partial t} = \dot{\mathbf{v}} - \dot{\mathbf{x}} \cdot \nabla \hat{\mathbf{v}}, \quad (2.52)$$

$$\frac{\partial \hat{T}}{\partial t} = \dot{T} - \dot{\mathbf{x}} \cdot \nabla \hat{T}, \quad (2.53)$$

$$\frac{\partial \hat{c}}{\partial t} = \dot{c} - \dot{\mathbf{x}} \cdot \nabla \hat{c}, \quad (2.54)$$

where the overdot indicates time derivatives with respect to the parent element frame which is analogous to the material derivative in Reynolds transport theorem. Figure 2.6 shows an example of a mesh with a deformed boundary at the melt-crystal interface.

The number of degrees of freedom (DOF) in the FEM framework approximating the governing equations is dependent on the number of elements used to describe the system geometry. The DOF is calculated as:

$$DOF = N_n \times V + N_i \times 2 + N_{em} \times 3 \quad (2.55)$$

where N_n is the total number of nodes in the domain, V is the number of unknowns other than pressure ($\mathbf{v}$, T , c), N_i is the number of elements used for mesh deformation, and N_{em} is the number of elements in the melt. The DOF correspond to the coefficients of the basis functions used to approximate the unknowns.

2.3.5 Time Integration

The unknowns (velocity, pressure, temperature, and concentration) have both spatial and temporal dependencies. By approximating the unknowns as a linear combination of spatially dependent basis functions and temporally dependent coefficients, the dependencies are decoupled. However, the spatial and temporal dependencies in the discretized govern-

ing equations need to be separated. Then, instead of solving an initial value/boundary value problem (IBVP), the equations are of the initial value problem (IVP) form:

$$\mathbf{M} \frac{d\mathbf{q}}{dt} = \mathbf{F}(\mathbf{q}) \quad (2.56)$$

where $\mathbf{q}$ is the vector of unknowns

$$\mathbf{q}^j = \begin{bmatrix} v_r^j \\ v_z^j \\ p^j \\ T^j \\ c^j \\ \xi^j \\ \eta^j \end{bmatrix} \quad (2.57)$$

for $j = 1 \dots N$ where N equals the total number of nodes. In the crystal solid and ampoule domains, $v_r^j = v_z^j = p^j = 0$. $\mathbf{M}$ is known as the mass matrix and $\mathbf{F}$ is a nonlinear vector function of the terms that do not have time derivatives of the unknowns in $\mathbf{q}$.

With the equations rewritten as an IVP, time is integrated using a numerical scheme. Integration, though, is difficult because the mass matrix, $\mathbf{M}$ is singular due to the lack of time derivatives for the pressure and the nodal positions. Additional difficulty arises from the nonlinearity of the equations. Therefore, an implicit time integration method is used. The method used here is the implicit trapezoid rule (TR) with a fixed timestep which is second-order accurate. This technique, adapted from Gresho [70], is unconditionally stable with respect to time step. Since TR depends on the derivative of the previous time step, the first time integration step of Equation (2.56) is done with Implicit Euler (IE):

$$\mathbf{M}(\mathbf{q}_1) \left[\frac{1}{\Delta t} (\mathbf{q}_1 - \mathbf{q}_0) \right] - \mathbf{F}(\mathbf{q}_1) = 0 \quad (2.58)$$

where the subscripts refer to time steps. The initial condition, $\mathbf{q}_0$, is obtained from a quasi-steady state solution of the governing equations $\mathbf{F}(\mathbf{q}_0) = 0$. The velocity of the melt/solid interface, V_{ms} is approximated as the ampoule translation rate V_a . In the subsequent steps, TR is used:

$$\mathbf{M}(\mathbf{q}_n) \left[\frac{2}{\Delta t} (\mathbf{q}_n - \mathbf{q}_{n-1}) - \left(\frac{d\mathbf{q}}{dt} \right)_{n-1} \right] - \mathbf{F}(\mathbf{q}_n) = 0 \quad (2.59)$$

For $n = 2$, the derivative $(d\mathbf{q}/dt)_{n-1}$ is calculated using IE:

$$\left(\frac{d\mathbf{q}}{dt}\right)_1 = \frac{1}{\Delta t}(\mathbf{q}_1 - \mathbf{q}_0) \quad (2.60)$$

while for $n > 2$ the derivative $(d\mathbf{q}/dt)_{n-1}$ is based on TR:

$$\left(\frac{d\mathbf{q}}{dt}\right)_{n-1} = \frac{2}{\Delta t}(\mathbf{q}_{n-1} - \mathbf{q}_{n-2}) + \left(\frac{d\mathbf{q}}{dt}\right)_{n-2} \quad (2.61)$$

The solution vector, $\mathbf{q}_n$, in Equations (2.58) and (2.59) is solved for using modified Newton-Raphson method. Newton-Raphson is an iterative method based on Newton's method of root approximation, but generalized for matrices. The governing equations, after discretization by the method of lines, are rearranged into residual form. As with the weighted residuals, we want to have an expression of the error, or residual, and minimize this error. Equations (2.58) and (2.59) are already written in residual form. At steady state, the residual equation is

$$\mathbf{R}_{ss} = \mathbf{F}(\mathbf{x}_{ss}) \quad (2.62)$$

where $\mathbf{R}$ is the vector of residual equations. These nonlinear residuals are linearized about a specific vector $\mathbf{q}_n^{(k)}$:

$$\mathbf{J}(\mathbf{q}_n^{(k)})\delta^{(k+1)} = -\mathbf{R}(\mathbf{q}_n^{(k)}) \quad (2.63)$$

where n denotes the time step value, and k denotes the iteration count at time step n . The Jacobian matrix, $\mathbf{J}(\mathbf{q}_n^{(k)})$, is defined as:

$$J_{ij} = \frac{\partial R_i}{\partial q_j} \quad (2.64)$$

where the subscripts i and j denote row and column indices, respectively; R_i is the i^{th} component of vector $\mathbf{R}$; and q_j is the j^{th} component of vector $\mathbf{q}$. The solution update vector, $\delta^{(k+1)}$, is defined by

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \delta^{(k+1)} \quad (2.65)$$

The converged solution of the previous time step is used as the initial guess for each iteration. The iterations continue until the L_2 norm of the residual vector and the solution

update vector are both less than specified error tolerances:

$$\|\mathbf{R}(\mathbf{q}_n^{(k+1)})\|_2 = \sqrt{\sum_{i=1}^N |\mathbf{R}_i(\mathbf{q}_n^{(k+1)})|^2} < \epsilon_1, \quad (2.66)$$

$$\|\delta^{(k+1)}\|_2 = \sqrt{\sum_{i=1}^N |\delta_i^{(k+1)}|^2} < \epsilon_2 \quad (2.67)$$

The linearized matrix problem in Equation (2.63) is put into LU-decomposition form using a direct solver method called the frontal solver developed by Hood [74]. The Jacobian matrix $\mathbf{J}$ is sparse due to the fact that the basis functions are nearly orthogonal. The product of the basis functions and their derivatives are non-zero only when multiplying basis functions corresponding to adjacent nodes. Gauss elimination can be performed as soon as all of the rows and columns involving the current element are constructed. In this manner, Gauss elimination is confined to the submatrix of rows and columns corresponding to nodes not yet eliminated but interact with the node being assembled. These nodes, called active variables, form a front that moves along the region hence the name frontal method.

Modified Newton-Raphson method uses a single LU-decomposition for several iterations and for several time steps. A new LU-decomposition is done when the ratio of the L_2 norms of consecutive residual vectors is less than a tunable parameter (≤ 1). Typically iterations continue until the L_2 norm of both the residual and solution update vectors are less than 10^{-4} times the L_2 norm of the solution vector.

2.3.6 Continuation

The governing equations depend on many parameters such as the Grashof number, Schmidt number, and the Prandtl number. It is useful to have a set of equations at different values for these parameters for comparison purposes. Also, the equations may be difficult to solve at desired parameter values. The parameters can be viewed as another set of independent variables which can be integrated in parameter space to the desired value,

$$\mathbf{F}(\mathbf{q}, \alpha) = 0. \quad (2.68)$$

where α is the desired parameter. This method used to determine a system's dependency on a parameter is called continuation. Continuation is also employed to study the stability of a system with respect to specific parameters. The analysis is performed on a steady

state solution.

There are three common continuation techniques: zero-order, first-order, and arc-length continuation. Zero-order continuation uses the solution at the previous parameter value as the initial guess for the new parameter value. This method is limited by the “step length” in parameter space: the new parameter value must be close to the previous. First order continuation, or Euler-Newton continuation, can take larger step lengths within the parameter space. The steady state solution, $\mathbf{q}(\alpha_o)$ at parameter value α_o is computed using modified Newton-Raphson. An Euler step is used to predict the solution:

$$\mathbf{q}(\alpha_o + \Delta\alpha) = \mathbf{q}(\alpha_o) + \left(\frac{\partial \mathbf{q}}{\partial \alpha} \right) \Big|_{\alpha_o} \Delta\alpha \quad (2.69)$$

The update vector $(\partial \mathbf{q} / \partial \alpha)$ is calculated as:

$$\mathbf{J}(\mathbf{q}(\alpha_o)) \left(\frac{\partial \mathbf{q}}{\partial \alpha} \right) \Big|_{\alpha_o} = - \left(\frac{\partial \mathbf{R}}{\partial \alpha} \right) \Big|_{\alpha_o} \quad (2.70)$$

where $\mathbf{R}$ is the steady state residual and $\mathbf{J}$ is its Jacobian given in Equations (2.62) and (2.64) respectively.

First order continuation can fail at turning points, i.e., points where the solution turns back on itself giving multiple solutions for a given parameter. At such a point, the update vector has an infinite slope. Another example of when first order continuation fails is bifurcation. Continuation can only track one solution, the other solution(s) are found using solution perturbation. Arc-length continuation, on the other hand, is unaffected by the presence of critical points. This method, first proposed by Keller [85], parameterizes the solution $\mathbf{q}$ and the parameter set α in terms of the arc-length along the path through parameter space. In other words, the arc-length is a solution path coordinate. For an in depth treatment of arc-length continuation see [128].

2.4 Conclusion

The vertical Bridgman model presented in this chapter was generated using the Crystallization and Transport Simulator (Cats2D) code, developed by Andrew Yeckel and Ralph Goodwin [195]. In Chapters 3-4, we simulate the growth of cadmium zinc telluride based on furnace temperature calculations from Pandey [135]. A simple random-restart hill climbing routine is added to the code for post-processing of zinc concentration data as discussed in Chapter 3. Proportional control is implemented to control the furnace temperature

gradient in Chapter 4. Additional software is used to model the entire crystal growth furnace using CrysVUn, a sophisticated radiation and conducted heat transfer code developed by Müller et al. [8,97]. Chapter 5 focuses on coupling Cats2D with CrysVUn to produce a multi-scale crystal growth model. An iterative solver technique based on a Schur complement method to solve the interface location problem is presented in Chapter 6.

Part II

Crystal Growth Modeling

Chapter 3

Tilting of Ampoule to Decrease Lateral Zinc Segregation

Contents

3.1	Chapter Summary	43
3.2	Introduction	44
3.3	Modeling Approach	46
3.4	Simulation Results	48
3.4.1	Flow and Mass Transfer in the Melt	49
3.4.2	Axial and Lateral Segregation of Zinc in the Crystal	53
3.4.3	Overall Segregation of Zinc in the Crystal	55
3.5	Discussion and Conclusion	58

3.1 Chapter Summary

We use a two-dimensional, planar computational model to study the effects of ampoule tilting on zinc segregation during the growth of cadmium zinc telluride in a low-gradient vertical Bridgman system. Tilting the axis of the ampoule away from the direction of gravity strongly affects the structure and strength of melt convection, which, in turn, modifies the overall heat transfer and the shape of the melt-crystal interface. While these effects have relatively little influence on axial segregation, the pattern of lateral segregation is significantly changed by ampoule tilting. Optimal tilt angles are found that flatten locally

lateral compositional profiles, and that decrease a measure of overall segregation compared to that grown in a perfectly vertical ampoule.

3.2 Introduction

One of the classical challenges of melt crystal growth is obtaining macroscopic, single crystals of uniform composition. In the vertical Bridgman process, as with most growth configurations, partitioning of a solute between the solid and melt naturally results in a crystal of inhomogeneous composition, a phenomenon known as segregation. Early analyses of segregation focused on how a solute would be distributed axially during uni-direction solidification. In the limit of no convective mixing, Tiller et al. [175] showed that diffusion balances dopant rejection at the melt-crystal interface giving rise to a uniform composition within the crystal except for initial and final transients. Growth toward this limit was the goal of initial space-flight experiments, where reduced levels of acceleration would drive much weaker convective flows [16, 44]. At the other extreme of complete mixing of dopant in the melt, axial composition is described by the Scheil equation [155] and leads to maximum axial segregation [65]. Under normal, earth-bound conditions, this mixing is typically driven by strong buoyant convection. The model proposed by Burton, Prim, and Schlichter [25], or the BPS model, incorporates both diffusive and convective mass transport in the melt, and is able to correlate a wide range of axial segregation behaviors between the complete-mixing and no-mixing limits.

Axial segregation, however, is only one component to overall dopant composition. The advent of numerical models for melt crystal growth enabled the study of radial segregation effects that occur in Bridgman systems. Brown and co-workers [23, 24] provide still-relevant discussions of how the spatial structure and intensity of melt flows, along with the shape of the melt-crystal interface, determine axial and radial segregation. A general outcome of these studies is that radial segregation is especially sensitive to the degree of mixing provided by flows near the interface. Due to incomplete mixing at lower levels of flow, the compositional difference across a crystal reaches a maximum at intermediate levels of convection. This behavior is different from that of axial segregation, which gets steadily worse as convection intensity increases.

An interesting optimization scenario is suggested by systems that display melt flows that are intense enough to lead to significant axial segregation and near-maximum radial segregation. In these systems, *increasing* the strength of melt flow near the interface may *decrease* the level of radial segregation without significantly affecting axial segregation.

Indeed, many vertical Bridgman procedures used to grow cadmium zinc telluride may fall into this category. This is so because, unlike classical semiconductors (e.g., silicon and gallium arsenide) whose melts exhibit a metallic character and a relatively low viscosity, the molten phase of cadmium telluride is semiconducting and exhibits high viscosity [69]. Under low thermal gradients, the buoyant flows in its melt phase are weaker than those in other semiconductor melts.

The question we want to address is whether a modification of buoyant flows in the melt by tilting the axis of an ampoule away from the gravitational vector could favorably influence the radial segregation of zinc during growth of CZT in a vertical Bridgman system. There have been few prior analyses of this scenario, primarily because the cylindrical symmetry inherent to the vertical Bridgman system method is broken when its axis is tilted away from the gravitational axis, thereby strictly requiring an expensive three-dimensional modeling approach. Xiao et al. [187] were among the first to compute mass transfer and segregation along with three-dimensional flows for a model vertical Bridgman system whose axis is tilted away from the gravitational vector. Liang and Lan [107] presented three-dimensional, steady-state models for vertical Bridgman growth that considered ampoule tilting effects. Lan and co-workers [101, 103–105] developed and applied three-dimensional models to describe small-scale, vertical Bridgman systems and the effects of ampoule rotation on the flows. Vizman et al. [182] studied ampoule tilting during Bridgman growth of semitransparent crystals, and concluded that three-dimensional flows were stronger due to radiation effects. Plaza et al. [141] investigated the influence of ampoule tilting on dopant distribution in the melt for growth of gallium antimonide. More recently, Yeckel et al. [189] modeled various imperfections leading to three-dimensional effects in vertical Bridgman systems. In agreement with previous observations by Xiao et al. [187], Yeckel et al. [189] showed that three-dimensional flows generated by tilting promote better mixing of the solute and could reduce radial segregation in the melt. However, all of these prior analyses employed quasi-steady-state formulations, such that they could not simultaneously consider the effects of axial segregation during growth.

To incorporate both radial and axial segregation in an affordable transient computation, we choose to model the vertical Bridgman system using a planar geometry instead of a cylindrical geometry. Upon tilting of the ampoule in such a model system, the flows remain two-dimensional in a planar geometry instead of producing three-dimensional flows as in cylindrical geometries. In this manner, we are still faithful to the idea of self-consistently modeling the coupled effects of heat and mass transfer, melt convection, and interface shape, but in a cost-effective approach so that we are able to conduct transient simula-

tions and model crystal composition with growth. Such an approximation is not new; for example, Garandet et al. [67] this method to study the effect of a transient, transverse acceleration on segregation during a space-flight Bridgman experiment.

3.3 Modeling Approach

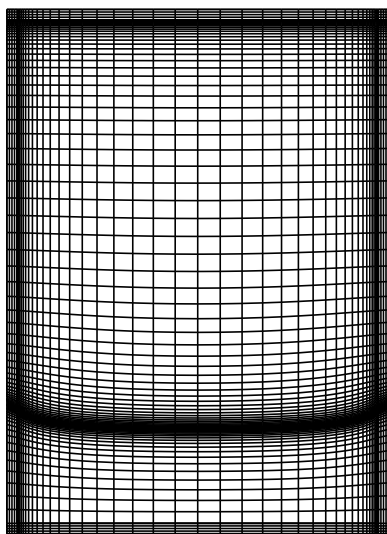
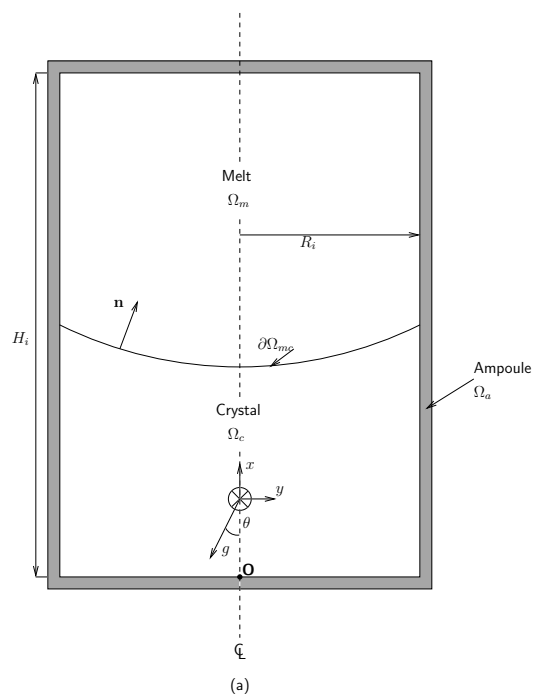
We consider the effects of ampoule tilting in a vertical Bridgman growth system. The crystal and melt are contained within the ampoule that stands in a vertical gradient furnace which translates relative to the ampoule to direct solidification from the bottom of the ampoule to the top. Our model consists of three regions: a crystal region, a melt region, and an ampoule region as seen in Figure 3.3(a). The regions are denoted Ω_i , where the subscript i is m for melt, c for crystal, and a for ampoule. The boundaries of region i are similarly labeled $\partial\Omega_i$ and its outward pointing unit normal is denoted $\mathbf{n}_i$. An example of the computational mesh is give in Figure 3.3(b).

We model two-dimensional, time-dependent phenomena which arise during growth by modeling the ampoule and its contents using equations of continuum transport of heat, mass, and momentum, coupled with physically appropriate boundary and initial conditions as described in §2.2.1-2.2.3. The computational domain of our model is assumed to be planar $\mathbf{x} = (x, y)$, with the coordinate origin fixed to the inner, bottom, center of the ampoule. The x -axis is aligned with the axial direction of the ampoule, and the y -axis is aligned with the lateral direction. The angle θ describes the amount of ampoule titling away from vertical, and gives rise to the following expression for the gravitational vector, $\mathbf{g}$, in the coordinates of the model:

$$\mathbf{g} = -g(\mathbf{e}_y \sin \theta + \mathbf{e}_x \cos \theta) \quad (3.1)$$

where g is the gravitational constant, and $\mathbf{e}_x$ and $\mathbf{e}_y$ are unit coordinate vectors associated with the Cartesian coordinate system fixed to the ampoule. If the ampoule is perfectly aligned with the vertical gravitational axis, the angle of tilt θ is zero and $\mathbf{g} = -g\mathbf{e}_x$. Appropriate changes are made to the governing equations to transform from cylindrical to planar geometries.

The furnace temperature corresponds to an assumed linear profile for the inner bore of the furnace. This profile also moves axially in time to drive the growth of the solid phase. The entire furnace is assumed to tilt along with the ampoule such that the ampoule and furnace share the same axis of symmetry.



(b)

Table 3.1: System operating parameters

Description	Symbol	Value	Unit
Ampoule inner radius	R_{in}	2.54	cm
Ampoule outer radius	R_{out}	2.68	cm
Ampoule inner height	H_{in}	6.98	cm
Ampoule outer height	H_{out}	7.26	cm
Ampoule wall thickness	t	0.14	cm
Ampoule translation rate	V_a	1.0	mm/h
Axial temperature gradient	∇T_f	2.33	K/cm
Gravitational constant	g	9.81×10^2	cm/s ²
Heat transfer coefficient	h	0.1 (estimated)	W/cm ² K
Initial Zn concentration	c_o	0.1	mole fraction

The initial condition corresponds to steady-state heat transfer and convection determined for a fixed ampoule location and no translation. The ampoule is placed with respect to the furnace temperature profile so that the melt-crystal interface is located near the bottom of the ampoule. We make no attempt to predict the initial nucleation of solid material from the melt, which is beyond the scope of our model. At time zero, we set the zinc concentration to a uniform value in the melt and the crystal, and initiate motion of the furnace temperature profile which drives ensuing growth.

3.4 Simulation Results

We consider the growth of nominally 2-in diameter CZT ingots in a low-gradient, vertical Bridgman furnace which is tilted off-axis. Tables 2.1 in Chapter 2 and 3.1 list the physical properties and operating parameters of the model system. Each growth simulation is begun with the ampoule at the same position in the furnace, and ended when the melt-crystal interface first encounters the end of the ampoule. Due to differences in the evolution of the shape of the interface, the total growth times vary slightly ($t = 50.4 - 51.4$ h) with tilt angle.

The accuracy of our numerical solutions was verified by grid refinement studies of the base (no-tilt) case. The melt domain was discretized with 600 (20×30), 2400 (40×60), and 4800 (60×80) elements. The flow velocity calculated using the different discretizations was compared along several cross-sections in the lateral direction at $x \pm 0.5$ in. and $x = 0.75$ in., and in the axial direction at $y = 0.8$ in. and $y = 2.5$ in. The error was calculated as a percent deviation from the finest mesh. Table 3.2 gives the maximum error of the two

Table 3.2: Mesh comparison

Mesh	DOF	$x = 0.5$	$x = 0.75$	$y = 0.8$	$y = 2.5$
20×30	4851	16.10%	21.25%	8.13%	1.62%
40×60	19,109	0.50%	0.45%	3.09%	0.21%

coarser meshes with respect to the finest mesh at the given cross-sections. The maximum errors for the coarse mesh of 600 elements ranged from 1.6% to 21.1%, while the medium mesh of 2400 elements produced relative errors ranging from 0.2% to 3.1%. Based on these measurements, the mesh with 40×60 elements in the melt (19,109 total degrees of freedom) was deemed accurate enough for the goals of this study, and was chosen for all computations shown here.

For the transient simulations, a dimensionless time step of $\Delta t = 0.2$ was used. A single factorization of the Jacobian matrix required approximately 5.8 s on a PC with a processor speed of 2.0 GHZ. The use of the modified Newton's method, where a single factorization is reused for several time steps, reduced the average time to compute a single time step to approximately 5.6 s. The computational time needed for a single transient computation was typically about 2 h.

3.4.1 Flow and Mass Transfer in the Melt

Transient simulations were computed for six cases with tilt angles ranging from the base case value of $\theta = 0^\circ$ to 30° . Figures 3.1 and 3.2 show melt flows and concentration contours for these cases after a short period of growth (at $t = 11.2$ h). Dashed contours in Figure 3.1 represent contour-clockwise rotation, while solid contours indicate clockwise rotation. In Figure 3.2, dashed contours indicate concentration within the solid while solid contours represent concentration within the melt.

Streamlines representing the flow of the base case, which is perfectly aligned with gravity ($\theta = 0^\circ$), are shown in Figure 3.1(a). This solution exhibits a four-cell flow structure that is symmetric about the system centerline. Analogous flow structures are typically seen in axisymmetric, vertical Bridgman systems. The flow cells nearest to the melt-crystal interface circulate downward along the ampoule walls and upward along the system centerline. These flow cells are driven by the radial temperature gradient (hot in the middle and cooler at the wall) caused by the release of latent heat at the melt-crystal interface. The upper cells flow upward along the ampoule walls and downward along the system centerline. The fluid flows in the upper cells are driven by heating from the furnace. Due

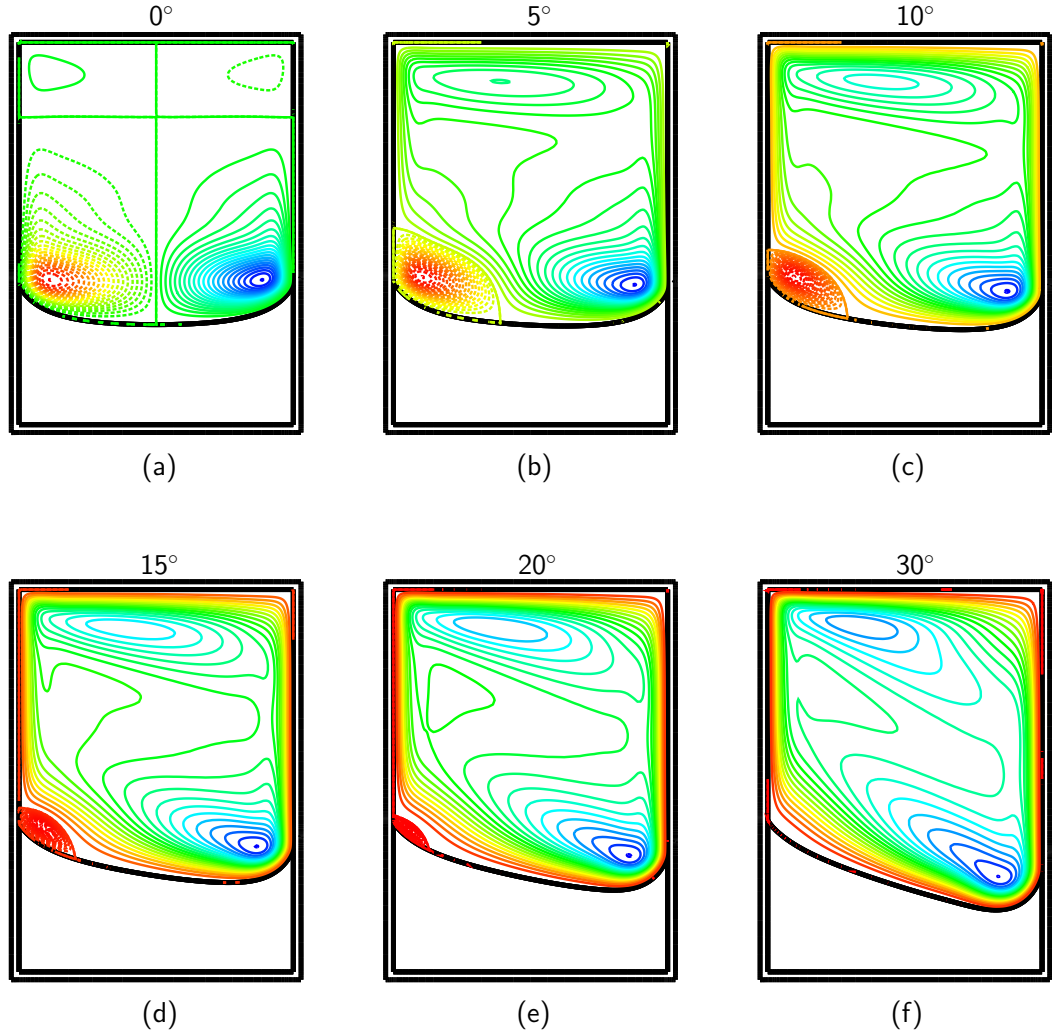


Figure 3.1: Streamline contours for various tilt angles: (a) 0° , (b) 5° , (c) 10° , (d) 15° , (e) 20° , and (f) 30° . Dashed streamline contours indicate counter-clockwise rotation at intervals $\Delta\psi = \psi_{max}/20$. Solid contours indicate clockwise rotation at intervals $\Delta\psi = \psi_{min}/20$. $\psi_{max} =$ (a) 3.00×10^{-2} , (b) 1.88×10^{-2} , (c) 9.66×10^{-3} , (d) 3.45×10^{-3} , (e) 5.16×10^{-4} , and (f) 4.29×10^{-8} cm^2/s . $\psi_{min} =$ (a) -2.99×10^{-2} , (b) -4.27×10^{-2} , (c) -5.66×10^{-2} , (d) -7.20×10^{-2} , (e) -8.86×10^{-2} , and (f) -1.26×10^{-1} cm^2/s .

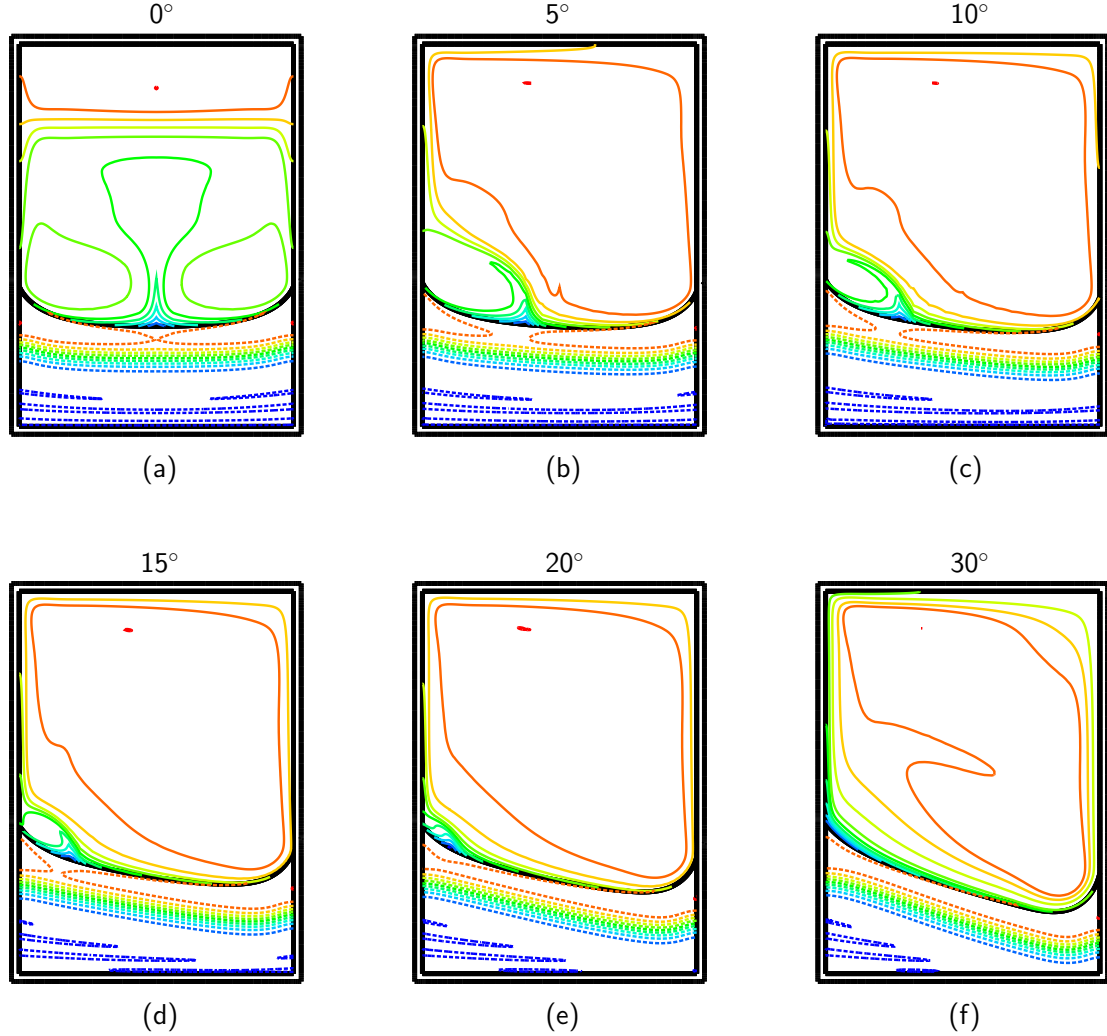


Figure 3.2: Mole fraction contours of zinc at different angles of tilt: (a) 0° , (b) 5° , (c) 10° , (d) 15° , (e) 20° , and (f) 30° . Dashed contours are used in the crystal at intervals $\Delta c = (c_{c,max} - c_{c,min})/10$. Solid contours are used in the melt at intervals $\Delta c = (c_{m,max} - c_{m,min})/10$. $c_{c,max} =$ (a) 1.32, (b) 1.32, (c) 1.32, (d) 1.32, (e) 1.32, and (f) 1.30. $c_{c,min} =$ (a) 1.00, (b) 1.00, (c) 1.00, (d) 1.00, (e) 1.00, and (f) 1.00. $c_{m,max} =$ (a) 0.97, (b) 0.96, (c) 0.96, (d) 0.96, (e) 0.96, and (f) 0.96. $c_{m,min} =$ (a) 0.94, (b) 0.93, (c) 0.93, (d) 0.93, (e) 0.93, and (f) 0.94.

to the low axial gradients in the system, the flows near the interface are stronger than those in the upper portion of the melt.

Tilting the ampoule from θ equal to 5° to a maximum of 30° progressively changes these flows. Even with a relatively small tilt angle of 5° , shown in Figure 3.1(b), the symmetry across the centerline of the base case is broken, which agrees with the results of previous three-dimensional simulations [187, 189]. Furthermore, the number of flow cells decreases from four to two as the lower flow cell on the right side of the system (the downward side of the titled system) enlarges and merges with the upper-left cell. The weak upper-right cell has disappeared, while the lower-left cell is pushed toward the interface and ampoule wall in the lower left corner of the melt. As the ampoule is tilted further, the merged single cell grows stronger and larger. Conversely, the remnants of the lower-left flow cell near the interface grows smaller and weaker, until it becomes negligible at $\theta = 30^\circ$. The remaining flow exhibits a complicated, clockwise sense of rotation with two centers.

To characterize the physical nature of these flows, we define non-dimensional Reynolds and Grashof numbers as

$$\text{Re} = \frac{R_{in}|v_{max}|}{\nu}, \quad (3.2)$$

$$\text{Gr} = \frac{l^3 g \beta \Delta T_{max}}{\nu^2}, \quad (3.3)$$

where R_{in} is half the lateral width of the ampoule, ν is the melt kinematic viscosity, $|v_{max}|$ is the maximum velocity in the melt, g is the gravitational acceleration, and β is the thermal expansivity of the melt. The characteristic scales that represent the nature of buoyant forces are necessary to consider the strength of driving forces for free convection that act parallel to the direction of the gravitational vector and across a horizontal lever-arm orthogonal to this direction. Therefore, the Grashof number is computed using a characteristic length scale, l , defined as half the length of a chord cut horizontally across the ampoule with respect to gravity, $l = R_{in} \sec \theta$. Furthermore, ΔT_{max} is defined to be the maximum temperature difference along a similar chord emanating from the highest point of intersection of the melt-crystal interface with the ampoule wall and continuing to its midpoint.

In each growth simulation, the intensity of the flow increases through an initial transient, then decreases as the volume of melt diminishes. From the 0° tilt case, the Reynolds number reaches a maximum of $\text{Re} = 37.0$, then decreases to $\text{Re} = 25.0$ by the end of the simulation. Tilting the ampoule gives rise to stronger driving forces. The $\theta = 30^\circ$ case

displays a maximum Reynolds number of $Re = 160$, falling to $Re = 93.7$. These values of the Reynolds number indicated that inertia is important in these flows, yet they remain laminar. Their nonlinear nature is exhibited by complicated, multiple-vortex structures with boundary and internal shear layers. The greater buoyant forces arising from tilting are reflected by computed Grashof numbers of $Gr = 7.76 \times 10^4$ for the 0° case and $Gr = 3.66 \times 10^5$ for the 30° tilt case.

The evolution of the melt flow structure with ampoule tilting has significant effects on the shape of the melt-crystal interface. Namely, as the down flow along the right side of the ampoule becomes stronger with tilt angle, it carries more heat via convection toward the interface, skewing it strongly toward the right.

Figure 3.2 shows concentration contours of the different systems at $t = 11.2$ h. For the base cases of no ampoule tilt, shown in Figure 3.2(a), the interaction of the flow near the melt-crystal interface is evident as the flow streams upward at the centerline, the zinc depletion layer is carried along with it. The lower flow cells act to mix the melt away from the interface, while the upper flow cells produce a second, nearly completely mixed region. These two mixing regions are separated by a horizontal diffusion layer that falls exactly over the shear flow between the upper and lower flows. Breaking the symmetry of the 0° base case by tilting acts to mix the melt more globally.

The nature of the flow near the interface significantly impacts lateral segregation. The minimum concentration along the melt-crystal interface skews toward the left with ampoule tilt, corresponding to the decreasing strength of the counter-clockwise flow cell. Even during the early stages of growth, these effects are reflected in different compositional fields within the crystal in the individual cases. Note that the oscillations in the composition field near the bottom of the ampoule represent extremely small oscillations around the composition of the initial solid. They are simply remnants of small errors in computing the start-up transient and are not significant.

3.4.2 Axial and Lateral Segregation of Zinc in the Crystal

Axial and lateral segregation of zinc in the crystal are compared for the different cases by looking at the concentration field in the crystal at the end of growth. Axial segregation is compared by measuring zinc concentration along the system centerline, $y = 1.0$ in. in Figure 3.3. Approximately the first quarter of the crystal exhibits a profile that reflects the starting “seed” crystal and initial transient. After this (for x greater than approximately 0.7 in), the concentration of zinc decreases with length, stemming from the depletion of

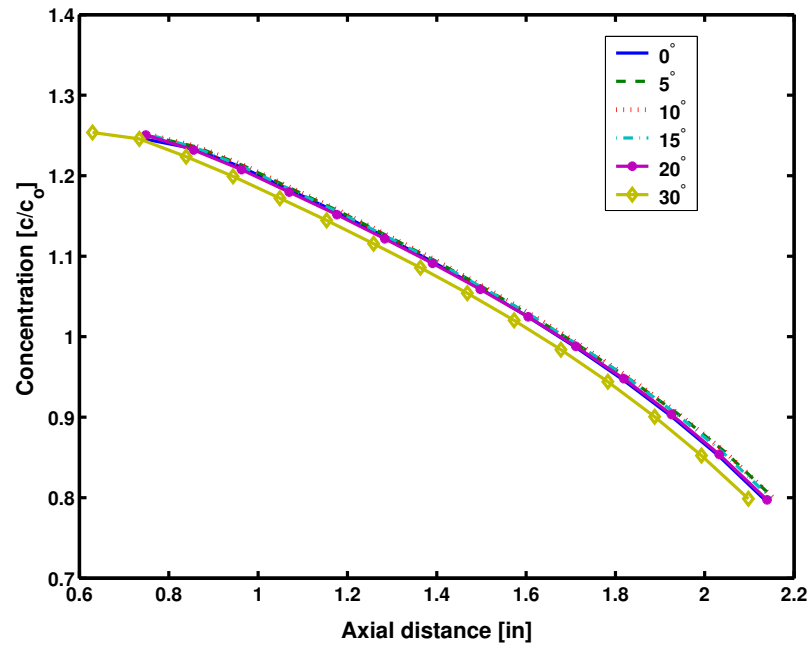


Figure 3.3: Axial zinc concentration is plotted along the centerline ($y = 0.0$ in) for different angles of tilt.

zinc in the melt during growth. Tilting has little effect on axial segregation of zinc. The concentration profiles at each angle have the same shape and nearly the same absolute concentration at the same axial location. Only the concentration profile of the $\theta = 30^\circ$ tilt case does not lie on top of the others, but its shape is the same. Its profile is shifted lower in the crystal, most likely due to the lower initial position of the melt-crystal interface; see Figure 3.1(f).

Figure 3.4 shows the lateral concentration profiles of the different cases at the same axial height in each crystal, chosen here to be $x = 1.50$ in. The lateral concentration profile of the $\theta = 0^\circ$ case is symmetric across the system centerline. As the ampoule is tilted, the location of the minimum point in the profile initially shifts toward the left ($\theta = 5^\circ$) and then toward the right ($\theta = 15^\circ$ and greater). While significant curvature of the concentration profiles remains in each case, at $\theta = 10^\circ$ tilt the profile flattens near the center. Figure 3.5 shows a direct comparison between the concentration profiles at $\theta = 0^\circ$ and $\theta = 10^\circ$ of tilt. The flattening of the profile suggests that there may be advantages of purposefully growing CZT in a tilted ampoule.

3.4.3 Overall Segregation of Zinc in the Crystal

Assessing the combined effects of axial and lateral segregation is made difficult in these computations due to the two-dimensional nature of the concentration field. To do this, we define a root mean square deviation in a specific area of the crystal:

$$c_{RMS} = \sqrt{\int_A (c - c_{mean})^2 dA} \quad (3.4)$$

where c_{mean} is the mean concentration in the area dA . We imagine that this area represents the best, i.e., most uniform in composition, $0.6 \text{ in} \times 0.6 \text{ in}$ square (a crystal dimension used in the manufacture of bulk CZT radiation detectors) than can be harvested from this ingot.

To find the minimum c_{RMS} in the crystal, we use a simple random-restart hill climbing routine (RRHC) to search the sample space [149]. Hill climbing is an iterative improvement algorithm that makes changes that improve the current state. In this case, the algorithm moves in the direction of decreasing c_{RMS} based on an initial state. The parameters that define the initial state are the x-, y-coordinates of the center of the square and a rotation angle, (x_o, y_o, ϕ_o) . Basic hill climbing has the drawback of making no progress at local extrema, plateau, or ridges. To overcome these difficulties, the RRHC algorithm is started again and again at a new, randomly selected initial states. The search is confined to a

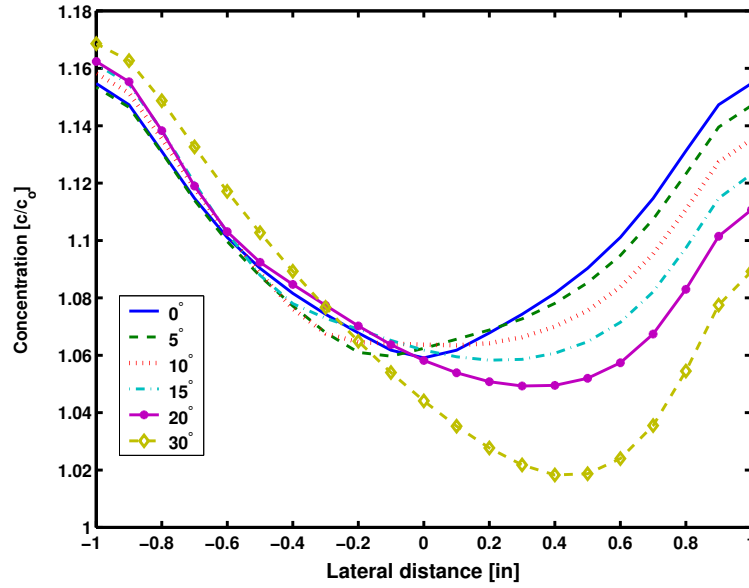


Figure 3.4: Lateral zinc concentration is plotted across the width of the crystal at $x = 1.5$ in. for different tilt angles.

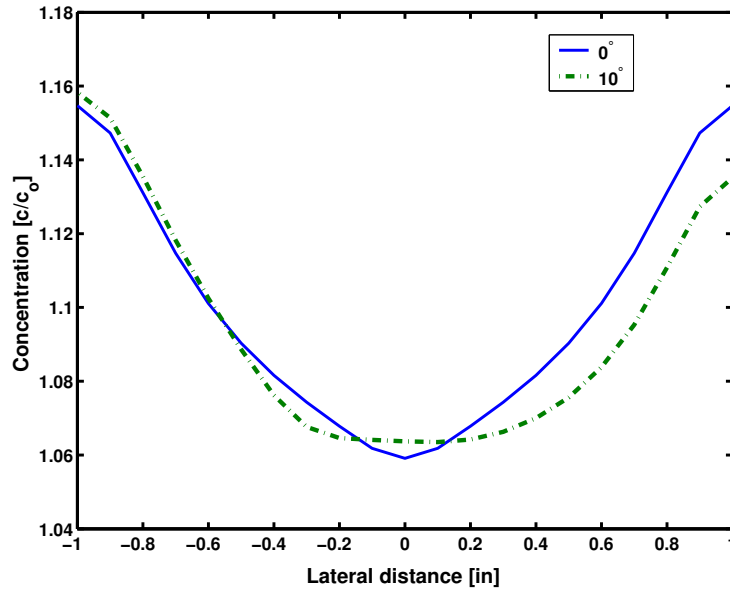


Figure 3.5: Lateral zinc concentration profiles for tilt angles of 0° and 10° at an axial height in the crystal of $x = 1.5$ in.

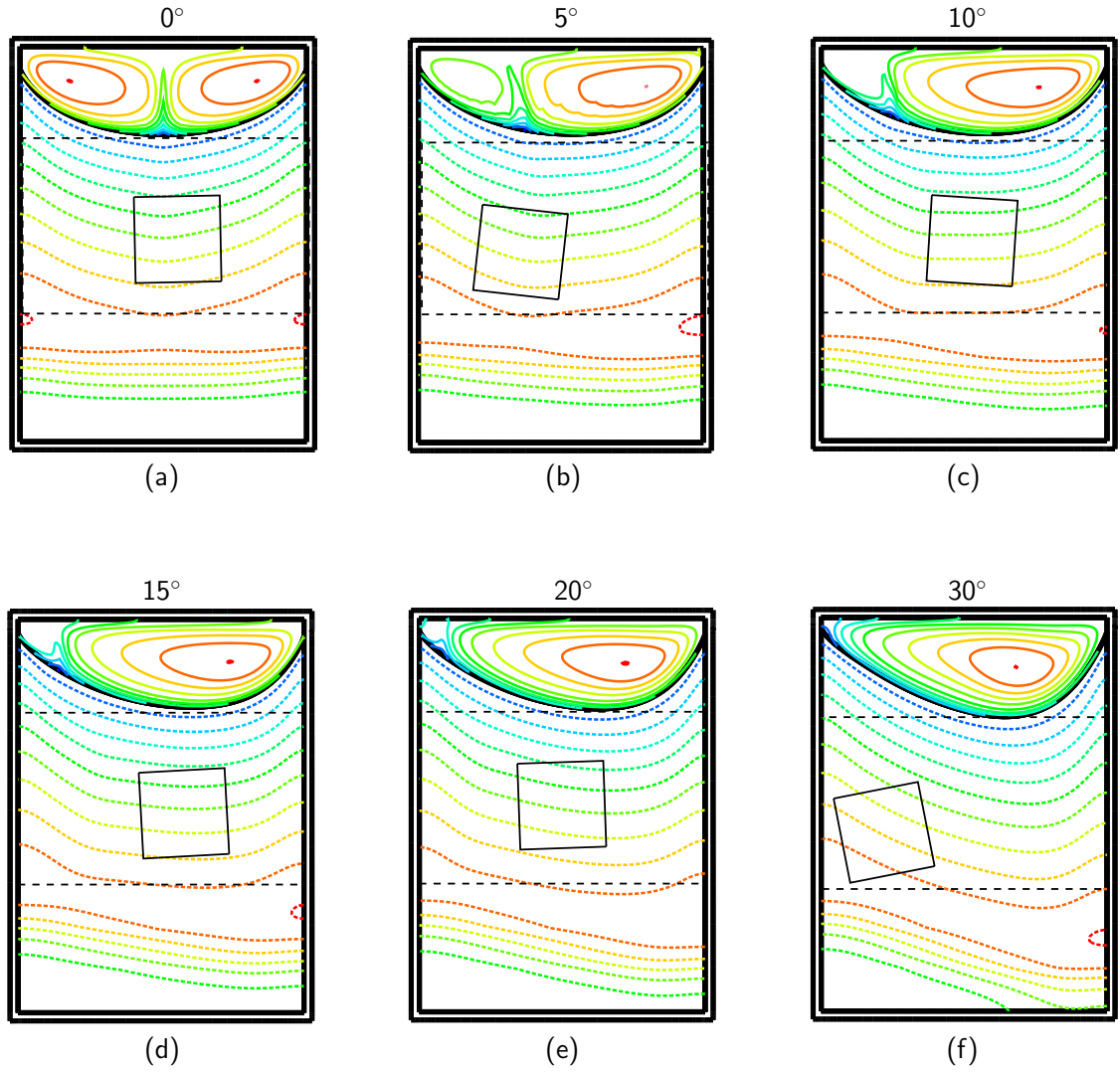


Figure 3.6: Location of $0.6 \text{ in} \times 0.6 \text{ in}$ square with optimal c_{RMS} , defines in § 3.4.3, for each case.

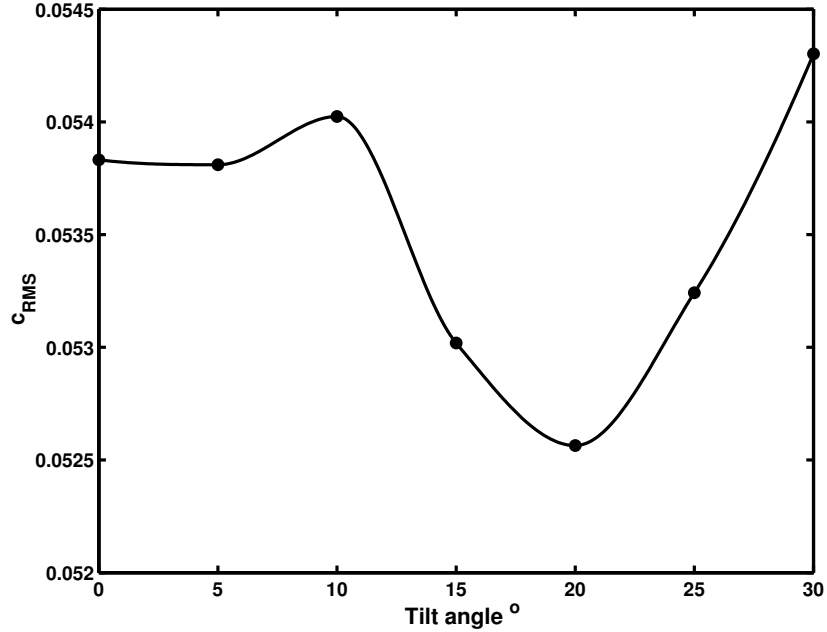


Figure 3.7: Optimal c_{RMS} plots as a function of ampoule tilt angle; c_{RMS} is defined in § 3.4.3 and is computer over the optimal square area indicated in Figure 3.6 for each grown crystal.

specific section of the crystal as marked by the horizontal, dotted lines in Figure 3.6. The area of the search space is the same for each tilt angle (2.1 in $\times$ 2.0 in), and is defined to avoid the initial transient in each crystal. The resulting optimal volume is indicated by the square drawn over each crystal in Figure 3.6

Figure 3.7 shows this measure of the minimum overall segregation, c_{RMS} , as a function of tilt angle. At $\theta = 5^\circ$ of tilt, the minimum c_{RMS} decreases slightly compared to the control case of $\theta = 0^\circ$. Then the minimum c_{RMS} increases at $\theta = 10^\circ$, which is the case with the flattened lateral concentration profile. Next the optimal c_{RMS} goes through a minimum point at $\theta = 20^\circ$, and then increases again as the degree to ampoule tilting is increased.

3.5 Discussion and Conclusion

We have presented transient, two-dimensional simulations to analyze the effects ampoule titling with respect to gravity on transient melt flows, mass transfer, interface shape,

and segregation during the vertical Bridgman growth of CZT crystals. Previous studies [107,141,182,187,189] on tilting employed quasi-steady state, three-dimensional models that focused on zinc concentration in the melt. Instead of the typical cylindrical geometry configuration, our model simplifies the system as a two-dimensional planar geometry allowing us to conduct transient simulations of the growth process. The ability to conduct transient simulations allows us to predict the overall composition of a crystal grown in a tilted vertical Bridgman system, a prediction not possible using quasi-steady-state models.

As in previous three-dimensional models, our two-dimensional simulation shows that the flow and zinc melt distribution are sensitive to tilting. In the base case $\theta = 0^\circ$, the flow is symmetric across the system centerline, and is characterized by two upper and two lower flow cells. However, when the ampoule is tilted, the shear layer between the upper and lower cells is broken, giving rise to just two flow cells, one larger and one smaller. As the ampoule is tilted further, the size of the smaller flow cell continues to shrink until its effect on the flow field is negligible. Because the flow structure in the titled ampoules cases is more global, the concentration field also becomes more uniform across the melt with ampoule tilting (see Figure 3.2).

Tilting showed little to no effect on zinc segregation in the axial direction, even though melt convection intensifies with tilt angle. This suggests, even when the ampoule is completely vertical, that convection is intense enough to approach the Scheil limit of complete mixing.

Lateral segregation, on the other hand, is greatly affected by tilting, as shown in Figure 3.4, due to changes in the melt flow near the interface and changes of the interface shape. At the lowest tilt angles, $\theta = 0^\circ$ and $\theta = 5^\circ$, the location of the minimum concentration point in the lateral direction coincides with the stagnation point at the solidifying interface (i.e., where the zero flow streamline between the two flow cells intersects the interface). This outcome is not strongly coupled to changes in the interface shape. At larger tilt angles, $\theta = 15^\circ$ to 30° , the lateral concentration profile is influenced by the shape of the melt-crystal interface. The minimum, lateral concentration point shifts toward the lowest point of the melt-crystal interface. Interestingly, at $\theta = 10^\circ$, the influence of the location of the stagnation point and the interface shape nearly balance each other, giving a lateral concentration profile that is flattened in the center. However, even for this case, the change in the average lateral composition is not significant. The effect of tilt is localized to the region near the system centerline.

Using c_{RMS} , our parameter of merit based on an optimal-composition, crystal area, we determine that a tilt angle of $\theta = 20^\circ$ gives the best balance between axial and lateral

segregation (see Figure 3.7. This angle is different than the best lateral segregation case of $\theta = 10^\circ$, suggesting a more complicated picture of zinc segregation than indicated by separate consideration of the vertical and lateral concentration profiles. However, using some measure of overall segregation is probably a more valid approach to assess whether tilting is favorable from the point of view of process yield.

While the model presented here is highly idealized (e.g., two-dimensional and has not considered any effects of crystal orientation or defects), and considers a specific system, the results do give an idea of how ampoule tilting influences dopant segregation in the crystal. As speculated in previous work [187, 189], tilting can lead to a more uniform crystal composition. At specific tilt angles in this model system, we find improved lateral segregation at $\theta = 10^\circ$, and overall segregation (within a $0.6 \text{ in} \times 0.6 \text{ in}$ square) at $\theta = 20^\circ$. The improvements brought on by ampoule tilting alone are not dramatic for this system. However, tilting does provide a means to change the behavior of the system, and its effects may be more pronounced for larger-diameter and higher-gradient systems where buoyant flow is stronger than considered here. Possibly the purposeful introduction of a combination of factors, such as ampoule tilt, furnace heating modification, ampoule rotation, etc., could be used to engineer process conditions that improve the vertical Bridgman process. To truly test this idea requires the ability to probe a multi-parameter operating space using more realistic process models. This prompts research in developing a multi-scale crystal growth model that is presented in Chapter 5.

Chapter 4

Control of Interface Shape Via Furnace Gradient

Contents

4.1	Chapter Summary	61
4.2	Introduction	62
4.3	Base Case Model	65
4.4	Open-Loop Study	66
4.4.1	Model Formulation	68
4.4.2	Results and Discussion	70
4.5	Closed-Loop Control Study	78
4.5.1	Model Formulation	78
4.5.2	Results and Discussion	80
4.6	Concluding Remarks	85

4.1 Chapter Summary

Numerical simulations are performed to assess the dependency of CZT interface shape on thermal gradients, and the feasibility of closed-loop control to control the interface shape in electro-dynamic gradient (EDG) furnaces. Open-loop results explain how larger furnace gradients in these systems tend to flatten the shape of the melt-crystal interface. Convection dominates heat transfer through the melt, and mixing acts to radially homogenize the

melt temperature field. These features do not significantly change with gradient conditions. In contrast, changing the rate of heat conduction through the solid, notable via changes to the furnace profile adjacent to the crystal, have an overriding influence on the interface shape. Increased gradients increase the ability for latent heat to be transported axially through the crystal and flatten the interface; however, the sensitivity of the shape of the interface to details of the furnace heating profile decreases as overall gradients increase.

Based on the results of the open-loop study, a simple proportional control algorithm is applied to a quasi-steady-state model to control the interface shape by adjusting the thermal gradient of the furnace temperatures. Three scenarios are enacted: actuation along the entire external boundary, along the solid portion only, and along the melt portion only. Actuation along the solid only proved to be the most successful of the three scenarios, and produced the unexpected benefit of a favorable convex interface shape.

4.2 Introduction

The macroscopic shape of the melt-crystal interface of crystals grown in vertical Bridgman systems has long been associated with crystal quality. An interface curved such that it is concave with respect to the melt is more likely to propagate adverse interface-ampoule interactions inward toward the crystal, potentially resulting in twinning or higher dislocation density [173]. An interface growing convex to the melt minimizes the potential for defects or spurious crystals nucleated at the ampoule wall to propagate toward the bulk crystal [21, 173]. In addition, the limit of a flat interface shape is consistent with very low radial temperature gradients and nearly constant axial temperature gradients near the interface, thereby minimizing thermal stresses [138].

In the absence of kinetic effects (as is often the case during the melt growth of semiconductors), the interface shape is determined by the melting-point isotherm which, in turn, is set by heat transfer. Much work, both experimental and theoretical, has been done to study the effect of the heat transfer environment on interface shape. Chang and Wilcox [31] developed an approximate heat transfer analysis for an idealized system with no melt convection and the same properties in both phases. Their analysis predicted that the shape of the interface depended solely on its axial location in the furnace. They postulated that the qualitative effects of buoyant convection in the melt would tend to make the interface more concave, consistent with the action that a greater melt thermal conductivity would have on interface shape. Sen and Wilcox [159] furthered this analysis by adding an ampoule to the system, using numerical techniques to solve the heat transfer problem. They showed that

the ampoule conductivity should be close to the conductivities of the melt and crystal to minimize radial temperature gradients within the system. Fu and Wilcox [66] showed that adding an insulated zone (the so-called adiabatic zone) between the hot and cold zones of the furnace flattened isotherms in the interface region and decreased the sensitivity of the interface shape to perturbations. Naumann [129] developed a two-dimensional analytical model of heat transfer in Bridgman crystal growth; which he used to study the effects of varying different operating parameters (e.g., adiabatic zone length and ampoule pull rate) on the axial gradients in the system as well as on the melting-point isotherm shape and position. Naumann and Lehoczky [130] illustrated the effect of unequal melt and crystal thermal conductivities on the shape and position of the melt-crystal interface. Chin and Carlson [33] also showed that interface curvature is very sensitive to the ratio of crystal to melt conductivities and the thermal gradient.

The effects of conductivity mismatch near the melt-crystal interface were further investigated by Jasinski and Witt [80] in their analysis of the growth of semiconductor crystals. They demonstrated the strong influence of ampoule thermal mass on the thermal field and the *interface effect*, where the thermal conductivity mismatch among melt, crystal, and ampoule coupled with the release of latent heat resulted in significant interface deflection at the ampoule wall. Brandon and Derby [20] demonstrated how the effects of internal radiation in the crystal could amplify such an interface deflection by modifying the heat fluxes near the ampoule wall, terming this the *radiative interface effect*. Adornato and Brown [1] studied the effect of different furnace profiles and ampoule materials on interface shape and favorably compared their results of the growth experiments of Wang and Witt [183]. Dutta et al. [56,57] were able to grow GaSb crystals with a planar interface under certain ratios of furnace temperature gradient to ampoule pull rate. Udayashankar et al. [180] calculated the critical ratio for flat-interface growth of GaInSb from their growth experiments. Arafune et al. [2] found that decreasing the thermal gradient decreased interface concavity over the growth run of InSb grown in an horizontal Bridgman furnace.

Nearly all of the preceding theoretical and experimental works studied the growth of semiconductor materials whose melts exhibit large thermal conductivities. Such materials are characterized by dimensionless Prandtl numbers much less than unity, typically $Pr \sim \mathcal{O}(10^{-2})$, where $Pr \equiv \nu/\alpha$, with ν representing the melt kinematic viscosity and α its thermal diffusivity. In such systems the effect of melt convection on heat transfer is relatively minor, so the balance of heat flows that determines the shape of the melt-crystal interface primarily involves latent heat release and conductive fluxes in both the melt and crystal.

Here, we are interested in how the melt-crystal interface can be effectively modified during the vertical Bridgman growth of CZT, and investigate the use of closed-loop control to improve interface shape during growth. As noted in Chapter 1, cadmium telluride and its alloys have unusual thermophysical properties which make their growth very different from other semiconductor crystals [146]. Notably, cadmium telluride has a high melt viscosity and a low melt thermal conductivity, lead to a Prandtl number of $Pr \approx 0.4$, much higher than typical. This behavior results from a liquid phase that is semiconducting [69], rather than metallic, as are the melts of more common electronic materials such as silicon and gallium arsenide. What this means is that convective heat transfer through the molten phase of CZT is much more important than conduction heat transfer under typical growth conditions. In such systems, the conditions affecting interface shape are a balance among latent heat, conduction in the solid, and *convection* in the melt. Strategies known to modify interface shape for classical semiconductor crystal growth may be rendered ineffectual for CZT growth because of this importance of convective heat transfer in the melt.

Prior analyses have shed some light on interface shape during CZT grown via Bridgman processes. Kuppurao et al. [92,93] analyzed a 75 mm diameter vertical Bridgman system for CZT and found the convection levels in the melt were intense enough to result in a relatively flat interface away from the ampoule. Later computations [95] showed that furnace design could control the interface shape, even made convex toward the melt, during early growth through the cone region of the ampoule, when convective effects were not strong. Yeckel and Derby [190,191] showed how transient flows driven by the accelerated crucible rotation technique (ACRT) affected mass transfer and interface shape in both large- and small-scale CZT Bridgman systems. Yeckel et al. [189] considered three-dimensional effects during small-scale, low-gradient CZT growth and found that interface shape could be affected by asymmetric heating by the furnace. Yeckel and Derby [192] also considered the effects of a submerged heater on shaping the interface during CZT growth. Pandey et al. [136] developed a sophisticated model of CZT grown via a low-pressure, electrodynamic gradient freeze (EDG) furnace used by eV Products, Inc. This approach employed a global furnace model coupled with a local model of the growing crystal to achieve a self-consistent solution of the entire growth system.

While much work has been done to improve the vertical Bridgman process and the growth of CZT, little work has been done to incorporate closed-loop control into the process. The sheer nature of vertical Bridgman growth inhibits the applicability of closed-loop control. Measurements of the system during growth are difficult, since opaque materials are often used for the ampoule. Furthermore, high temperatures are required for crystal

growth, and the ampoule is often sealed due to volatile raw materials. Chin and Carlson [33] and Seidensticker et al. [157] used open-loop simulations to investigate the effects of temperature boundary conditions on interface shape and their suitability to a control algorithm. Azuma et al. [6] developed an automatic feedback control system to keep the temperature at the interface of silicon germanium, which is dependent on composition, during growth to control bulk composition. The ampoule translation rate was updated based on the position of the interface, which was detected automatically by images captured by an in situ monitoring system via a charge-coupled-device camera. Sonda et al. [165,167] studied the feasibility of using feedback control to suppress oscillatory flows via crucible rotation with numerical modeling on indium antimonide. Batur and co-workers [11,12,168] fashioned a transparent, multi-zone furnace for the production of lead bromide. A video camera and real-time imaging software was able to observe the melt-crystal interface shape through the transparent furnace. The temperature profile was adjusted by comparing the observed interface shape to the desired shape. A good overview of recent work of active growth control is provided by Lan [102].

This study has two goals. The first is to perform open-loop studies to examine how furnace gradients affect the melt-crystal interface of CZT grown in a vertical Bridgman system using numerical simulations. The second goal is to implement closed-loop control to the simulation to control the melt-crystal interface by adjusting the gradient of the applied furnace temperatures.

4.3 Base Case Model

Our model endeavors to assess the effects of modifying the furnace design and the applicability of closed-loop control on CZT interface shape in the low-pressure, electro-dynamic gradient (EDG) system developed by eV Products, Inc. A base case is obtained by utilizing the multi-scale, coupled model developed by Pandey et al. [136] corresponding to conditions employed in the low-pressure EDG system. Work is proceeding to improve the robustness and efficiency of the coupled approach (see Chapter 5); however, its utilization remains involved and computationally expensive. Here, we employ a simpler modeling strategy for examining changes in behavior due to increasing the furnace temperature gradient.

For the open-loop model, we begin with the base state, then we transform the furnace temperature profile using simple scaling factors and solve for the resulting changes in heat transfer, melt convection, and interface shape in the system. In our closed-loop control study, instead of using arbitrary values factors to transform the furnace gradient, the scaling

factor is updated based on the current interface deflection using a control algorithm.

The base case is a steady-state model comprised of two parts: a global model representing the furnace, and a local model consisting of the melt, crystal, and melt-crystal interface. A schematic diagram of the meridional views of the two-dimensional, axisymmetric model is shown in Figure 4.1. Two separate codes are used to solve the global and local models. CrysVUn, used for the global model, was developed by Müller et al. [7, 96] specifically to model radiation heat transfer occurring in high-temperature, crystal growth furnaces. This code used the finite volume method on an unstructured grid to discretize the furnace heat transfer equations, which are solved via a quasi-Newton iterative method. For the local model, we use Cats2D [195] as discussed in Chapter 2. The top of the melt is not enclosed, and is modeled as a free surface using Equation (2.7).

The global and local models share a common boundary, see Figure 4.1, along which energy is conserved through strict continuity of the temperature field and norm heat fluxes:

$$T_{global}(\mathbf{x}) = T_{local}(\mathbf{x}) \quad (4.1)$$

$$q_{global}(\mathbf{x}) = q_{local}(\mathbf{x}) \quad (4.2)$$

where $\mathbf{x}$ denotes the boundary coordinates and q is the heat flux normal to the boundary. Obtaining a self-consistent solution of these coupled models that rigorously satisfies both Equations (4.1) or (4.2) is not an easy task and is dependent upon the details of how iterations are conducted between the two models. The base case presented here was developed using a Block Gauss-Seidel iterative approach to solve for quasi-steady-state solutions as described in References [136, 196].

4.4 Open-Loop Study

The purpose of the open-loop study is twofold. The first is to examine how the furnace temperature gradient affects the interface shape of CZT, and the interface shapes response to perturbation in the furnace temperatures. The second goal is to reconcile the difference in interface shapes predicted for the low-pressure EDG system and the high-pressure EDG pressure process (HP-EDG), also developed by eV Products, Inc., by Szeles et al. [170]. For the nominal growth conditions in the low-pressure system considered by Pandey et al. [136], significant interface deflection is predicted. However, Reed et al. [144] predicts a relatively flat interface for the HP-EDG system. The HP-EDG is distinguished from the low-pressure system by a larger diameter and a higher temperature gradient. The computational ap-

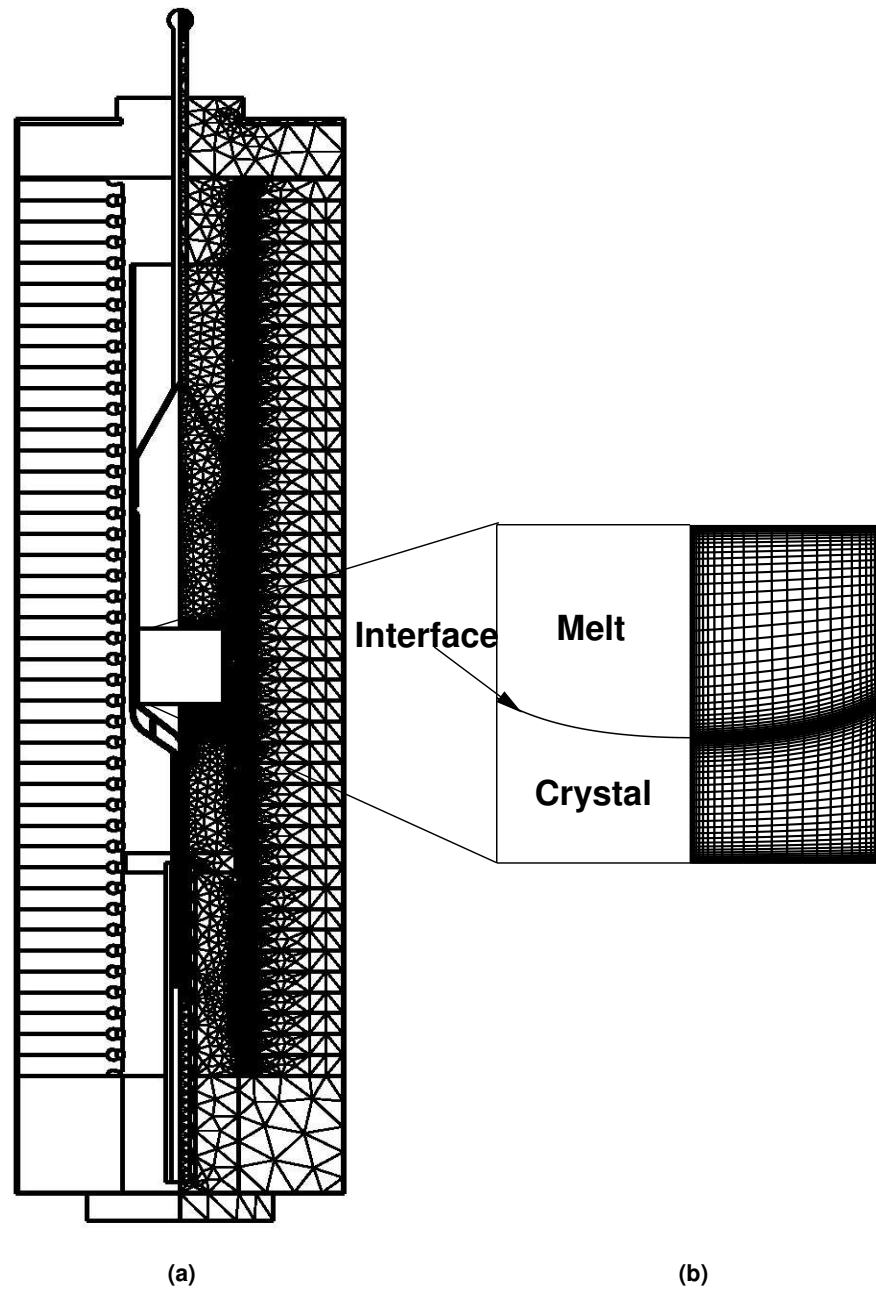


Figure 4.1: Schematic diagram of the computational domains and meshes of the (a) global model and (b) local model.

proach taken here aims to connect these seemingly disparate results, and investigate more general phenomena affecting interface shape during CZT growth.

4.4.1 Model Formulation

We stretch the conditions of the base case to represent different furnace configurations under higher gradients to assess the sensitivity of interface shape to furnace changes. In this manner, we transform the temperature gradient present in the low-pressure EDG system to that of the HP-EDG system. To do this, we take the self-consistent solution computed by the coupled model described above as our base case and assume that changes to design of the furnace can be represented by slight modifications of the base-case temperature along the coupling boundary, namely the exterior domain boundary indicated in Figure 4.1(b). With this approach, we are able to solve the local model, Cats2D, independently, and different parameter studies are computed quickly and efficiently.

In the open-loop study, a Dirichlet temperature boundary condition is applied to the local model. This boundary conditions is an algebraic stretching of the self-consistent, base-case temperature profile:

$$T(\mathbf{x}) = \alpha * (T_o(\mathbf{x}) - T_i) + T_i \quad (4.3)$$

where T is the new temperature, α is the gradient magnification factor, T_o is the temperature along the shared boundary calculated by the coupled model, and T_i is the interfacial temperature. This approach generates a family of temperature profiles that exhibit increasing axial gradients as the gradient magnification factor, α , is increased from unity. The profile of the temperature along the outer vertical boundary of the local model is shown in Figure 4.2 for a number of different values of α . The axial temperature gradient near the interface is approximately 2 K/cm for the base case; the axial gradients from the other cases then correspond to approximately 2α K/cm.

We also examine the sensitivity of the interface shape to perturbations to the furnace heating profile. This is accomplished by adding the following perturbation to the furnace profile:

$$dT = \beta \left(\frac{(z_i/2)^2 - (z - z_i/2)^2}{(z_i/2)^2} \right) \quad (4.4)$$

where dT is the perturbation, β is the perturbation maxima, z is the axial position with respect to the bottom of the crystal, and z_i is axial position of the interface along the boundary. We add this perturbation to the temperature profile only below the melting-

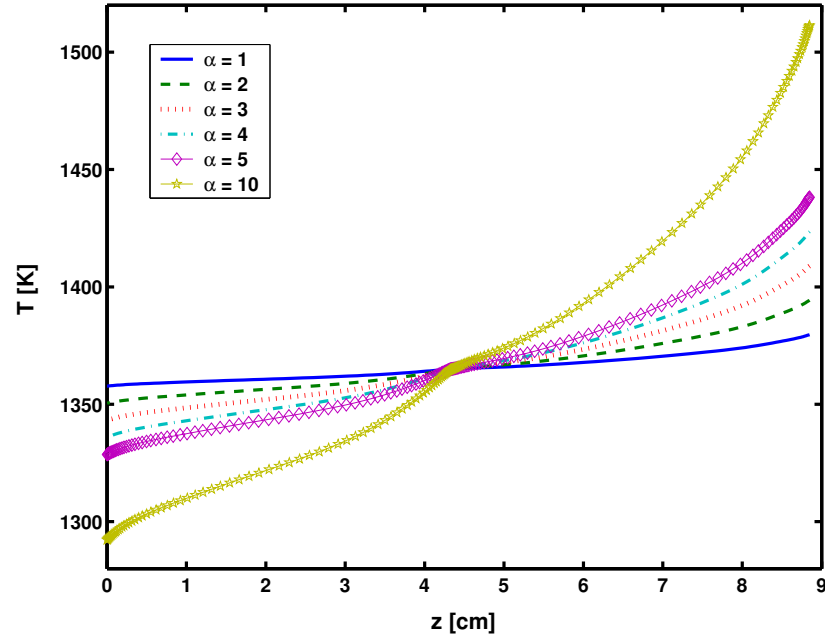
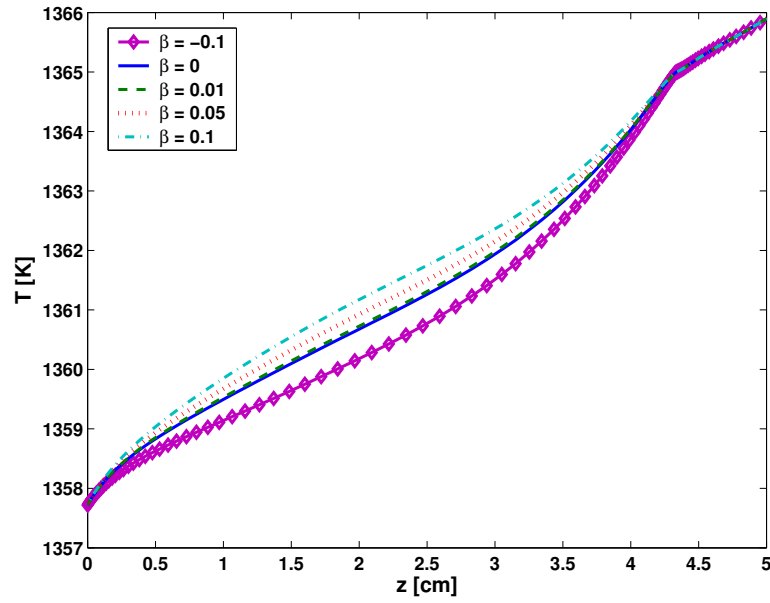
Figure 4.2: Axial furnace temperature gradient for different values of α .Figure 4.3: Perturbation of furnace temperature gradient for different values of β .

Table 4.1: Physical properties of CdTe [92], system dimensions, and operating parameters.

Parameter	Phase	Symbol	Value	Unit
Thermal conductivity	melt	k_m	1.085×10^{-2}	W/cm K
	solid	k_s	9.07×10^{-3}	W/cm K
Sample radius		R	4.625	cm
Sample height		H	8.85	cm
Ampoule translation rate		V_a	0.6336	cm/s

point temperature, so that the new boundary condition around the domain is :

$$\text{at } z > z_i : \quad T = T, \quad (4.5)$$

$$\text{at } z \leq z_i : \quad T = T + dT \quad (4.6)$$

where T is the temperature from Equation (4.3). Therefore, the temperatures along the melt are unaffected. A positive value of β corresponds to additional heating supplied to the crystal. Figure 4.3 shows a representation of how the perturbation β affects the temperature boundary condition.

4.4.2 Results and Discussion

The physical properties used to represent CZT are listed in Table 2.1 in Chapter 2, but with values for melt and solid conductivities agreeing with those used in References [135, 136]. These conductivity values along with the base case operating parameters are listed in Table 4.1. The accuracy of the numerical solution was tested by comparing the solution for steady-state flow at $\alpha = 10$, the strongest flow case, for different mesh discretization of the melt phase. Three meshes comprising 1400 elements, 2400 elements, and 4000 elements were employed. When the finest mesh solution was used as a benchmark, the worst-case error in predicting the magnitude of the velocity along a radial line through the melt near the melt-crystal interface was 2.1% for the 1400-element solution and 1.1% for the 2400-element solution. The worst-case error for computing the velocity along a vertical line drawn midway between the centerline and ampoule wall was 29.4% for the 1400-element solution and 4.1% for the 2400-element solution. For the purposes of this study, we concluded that the 2400-element mesh (representing a total of 28,765 degrees of freedom) was sufficient in accuracy. The results of the accuracy tests are summarized in Table 4.2. A full Newton step, dominated by the cost of a single factorization of the Jacobian matrix, took approximately 19 s on a PC with a 2.0GHz processor speed. Typically, four to six

Table 4.2: Mesh size comparison of percent deviation of velocity magnitude at various cross-sections within the melt.

Mesh	DOF	$r = 0.46$ cm	$r = 2.31$ cm	$z = 4.42$ cm	$z = 7.96$ cm
20×30	15,925	50.8%	39.4%	2.15%	81.2%
30×40	28,765	2.86%	4.09%	1.12%	7.13%

Newton iterations were needed to obtain a fully converged solution.

Effect of Axial Gradients on Melt Flow and Interface Shape

We consider the response of heat transfer in the melt and crystal, flow in the melt, and interface shape as the axial gradient is increased in this system. The quasi-steady-state (interface assumed to translate at ampoule pull rate to calculate latent heat) model solutions correspond to a point approximately half-way through the growth run. Figure 4.4 shows melt streamlines, temperature isotherms, and interface shapes for the different cases with the system centerline along the left-hand side. Dashed streamline contours indicate clockwise rotation, and solid contours indicate counter-clockwise rotation. The base case for $\alpha = 1$ shows a clockwise circulation enclosing two vortices, one near the melt-crystal interface and one near the free surface. This flow is driven by two dominant effects. First, the release of latent heat from the solidification causes the center of the melt to be warmer than near the ampoule wall, driving fluid up the centerline and down along the wall. Second, Marangoni forces along the free, upper surface of the melt result in an outward-directed surface stress, adding another force to drive a clockwise circulation. In the melt away from the melt-crystal interface, the flow is strong enough to effectively flatten the temperature isotherms, leading to a temperature field that is almost one-dimensional (only a thin boundary layer is present along the ampoule wall). The melt-crystal interface bows significantly to allow for radial conduction of heat through the solid.

With an increased axial gradient applied to the system, cases (b)-(f) in Figure 4.4, the total amount of heat flowing through the charge increases and the axial gradients through the system increase. The intensity of the flow increases, as shown in Figure 4.5, where the maximum magnitude of the streamfunction, ψ_{max} , which represents the total volumetric circulation of the melt, is plotted as a function of α . The flow strength scales asymptotically with $\alpha^{1/2}$, as indicated in the figure by the functional curve fit of ψ_{max} . This form is consistent with a balance of dominant forces of inertia (which is proportional to the velocity squared) and buoyancy (linearly proportional to the temperature gradient)

for these nonlinear melt flows.

Interestingly, and perhaps counter-intuitively, even while the overall flow strength in the melt increases with the applied thermal gradient, the strength of the flow vortex near the interface *decreases* as α is increased. An explanation is provided by plotting the radial temperature profile near the ampoule wall and slightly above the interface, as shown in Figure 4.6. Here, it is evident that the temperature gradient near the wall does decrease as the applied axial gradient is increased, leading to a simultaneous reduction of the buoyant force driving the lower vortex as α is increased. The profiles for $\alpha = 5$ and 10 are not shown in Figure 4.6, since the radial gradient actually reverses near the wall for these cases. This effect opposes the clockwise circulation of the lower cell, eventually leading to a weak counter-clockwise flow adjacent to the interface for the $\alpha = 10$ case.

The same interactions responsible for the decreasing, radial, melt gradient and the weakening lower flow cell, discussed above, alter the relative balance of axial and radial heat fluxes along the melt-crystal interface. There is a notable flattening of the interface shape with increases in α . Significantly, the progression of interface shapes displayed by these calculations captures the qualitative change between the low-pressure EDG system computed by Pandey et al. [136] and the HP-EDG system modeled by Reed et al. [144]. Indeed, the interface shape of cases $\alpha = 2$ and 3 are quite similar to that predicted in References [144, 170] for the HP-EDG system. For yet higher gradients, the interface becomes even flatter.

We probe the causes of the flattening of the interface with increased gradient by concentration on the case of $\alpha = 2$ and performing two additional computations. Figure 4.7(a) shows the original base case corresponding to $\alpha = 1$. Next to it, in Figure 4.7(b), is the case of $\alpha = 2$, showing a flatter interface and a weaker flow in the melt near the interface. Figure 4.7(c) shows a new computation where the magnification factor of $\alpha = 2$ is applied only to the melt. This is carried out by applying the boundary condition specified in Equation (4.3) with $\alpha = 2$ for $T \geq T_i$ and for $\alpha = 1$ for $T < T_i$. with larger temperature gradients in the melt, the flow is everywhere increased in strength, including the vortex near the interface. This increased flow drives warmer melt toward the center of the interface and slightly increases its deflection from the base case. Figure 4.7(d) displays the result when only the crystal profile is magnified, namely $\alpha = 1$ is used for $T < T_i$ and $\alpha = 2$ is applied for $T \geq T_i$. In this case, the interface is dramatically flattened and the melt flow near the interface is significantly diminished.

Clearly, the effect of changing the temperature gradient applied to the crystal is far more important than changing the gradient applied to the melt. This underscores the balance

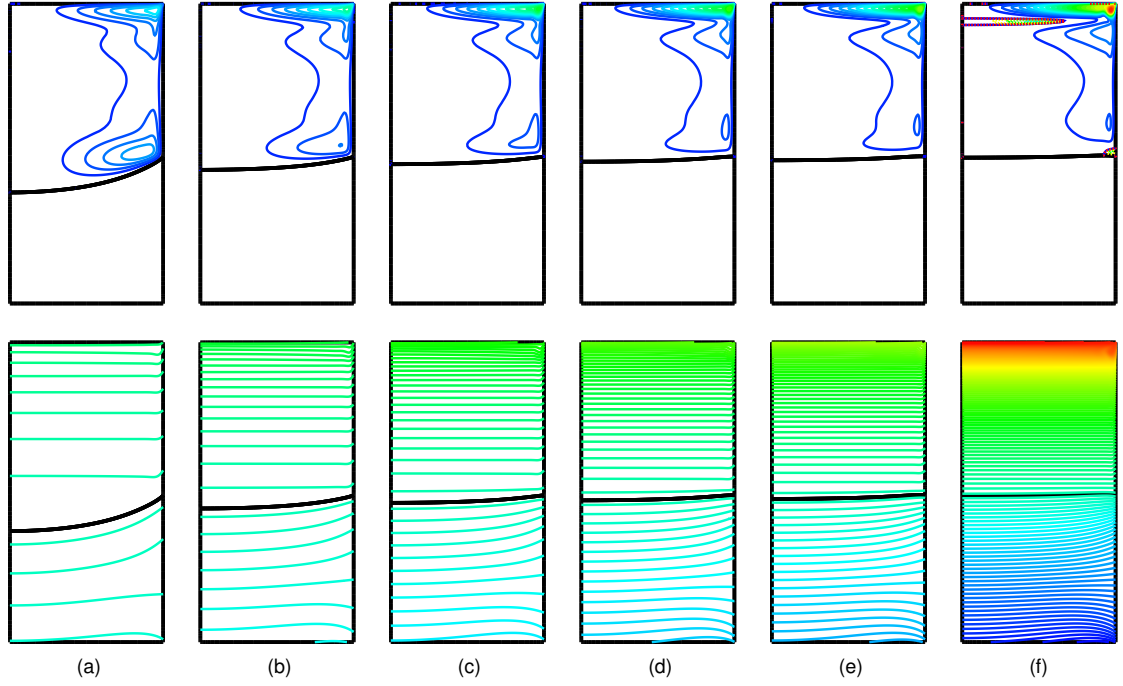
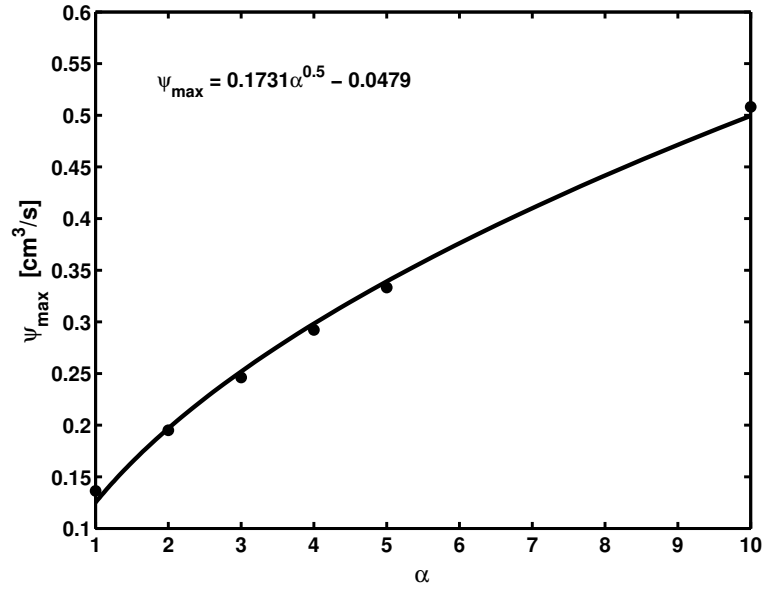
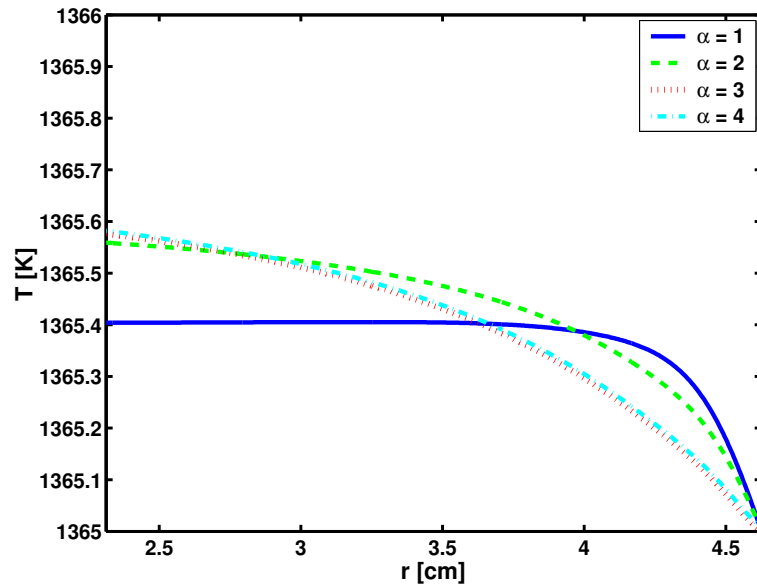


Figure 4.4: Streamline and temperature contours for each case as a function of the gradient magnification factor: (a) $\alpha = 1$, (b) $\alpha = 2$, (c) $\alpha = 3$, (d) $\alpha = 4$, (e) $\alpha = 5$, and (f) $\alpha = 10$. Isotherms are space at regular intervals of $\Delta T = 2$ K. The streamline contours indicate increasing flow intensity as the furnace temperature gradient is increased. For plot (a)-(e), flow is in the clockwise direction indicated by the dashed contours spaced at intervals of $\Delta\psi = 2 \times 10^{-2} \text{ cm}^3/\text{s}$. In plot (f), dashed contours indicate clockwise flow at intervals used previously, and solid contours indicate counter-clockwise flow at intervals of $\Delta\psi = -4 \times 10^{-3} \text{ cm}^3/\text{s}$.

Figure 4.5: Maximum value of streamfunction versus α .Figure 4.6: Radial temperature in the melt plotted slightly above the height of the solid interface for various values of α .

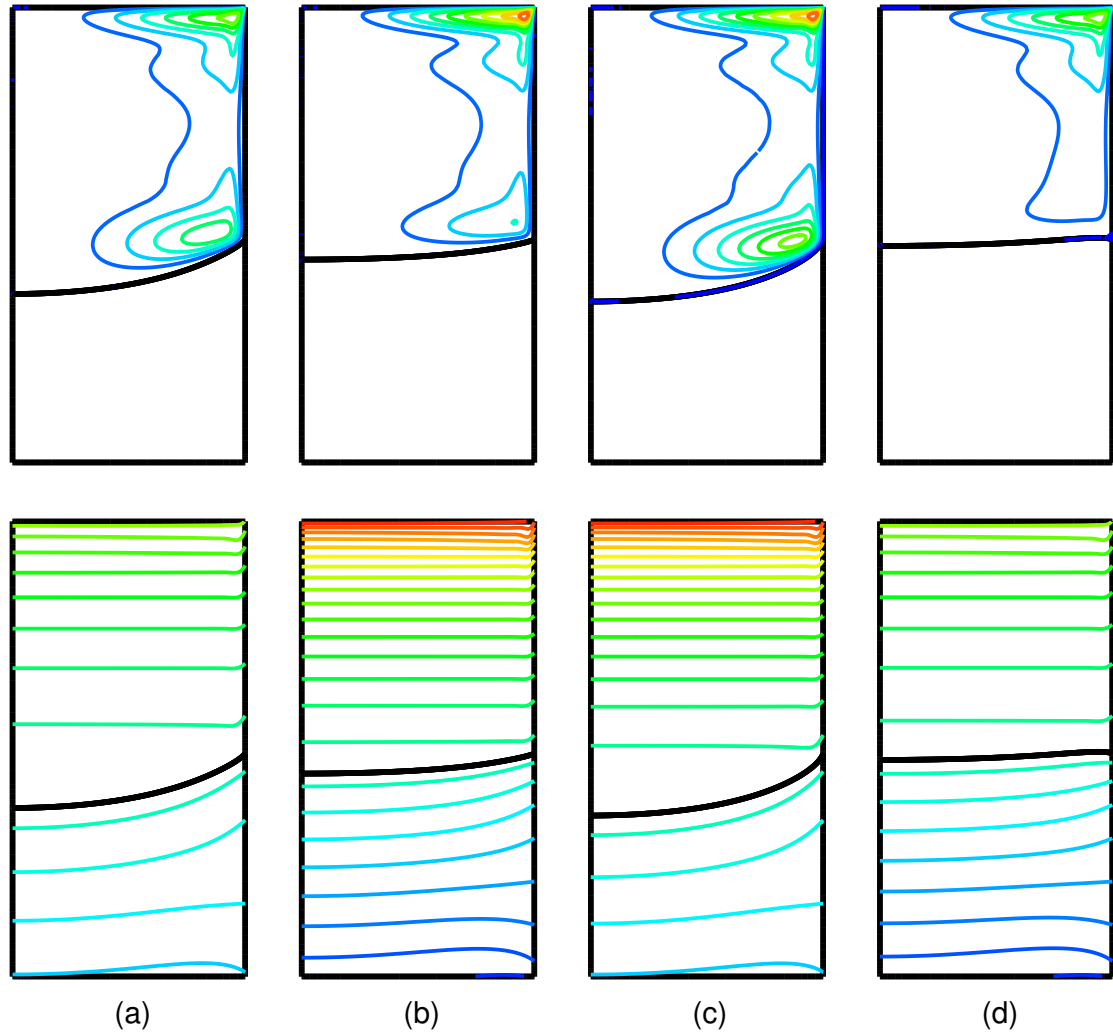
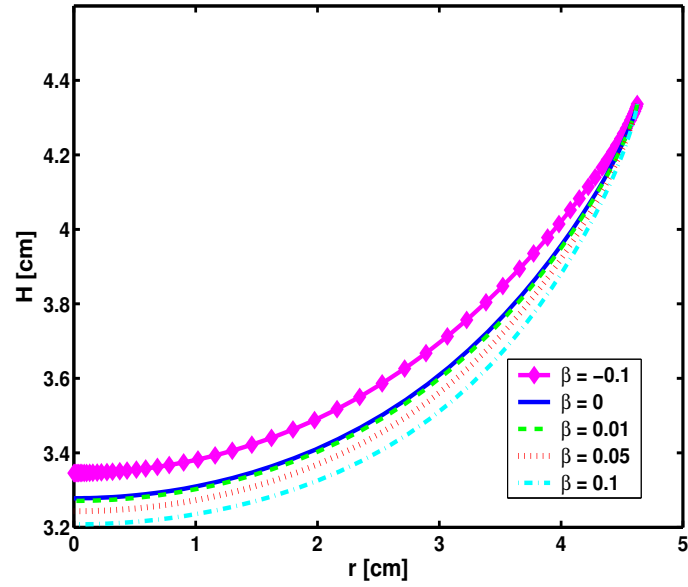
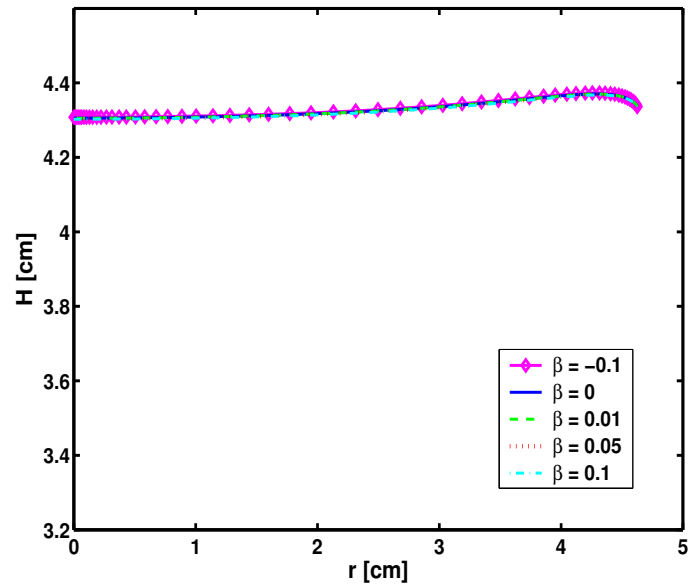


Figure 4.7: Streamline and temperature contours for: (a) base case, $\alpha = 1$, (b) total gradient doubled $\alpha = 2$, (c) doubled temperature gradient in the melt only, and (d) doubled temperature gradient in the crystal only. Isotherms are spaced at regular intervals of $\Delta T = 2$ K. Dashed streamline contours indicate flow in the clockwise direction spaced at intervals of $\Delta\psi = 2 \times 10^{-2} \text{ cm}^3/\text{s}$.



(a)



(b)

Figure 4.8: Axial height of interface versus radius for (a) $\alpha = 1$ and (b) $\alpha = 10$ at different values of β

among heat flows in this systems that are dominated by convection in the melt, conduction in the crystal, and latent heat at the interface. Changing overall system gradients, such as achieved by scaling the entire temperature profile in the cases shown in Figure 4.4, does not markedly change the nature of heat transfer through the melt, since it is dominated by convective effects. However, allowing for greater temperature gradients through the crystal dramatically affects conductive heat flows, especially in the axial direction. Latent heat, released at the growing interface, can only flow down a temperature gradient. If the axial flux of heat through the crystal is insufficient, the interface must deflect to allow some of the latent heat to flow radially outward. Promoting a greater axial heat flux by larger temperature differences along the crystal more readily accommodates the axial flow of latent heat, thereby flattening the interface.

Effect of Furnace Perturbations on Interface Shape

Finally, we wish to address the possibility of modifying the shape of the interface by slight changes in the furnace heating profile, an action that is made feasible by the design of multiple-zone EDG furnaces. We specifically consider the response of the interface shape to the furnace perturbation previously described by Equation (4.4) and represented in Figure 4.3. Figure 4.8 shows the shape of the interface for various values of the perturbation parameter, β , for the case of the low axial gradient, $\alpha = 1$, and the high axial gradient, $\alpha = 10$. Applying this perturbation to the temperature field scarcely changes flows and heat transfer through the melt. However, increasing the value of β increases the temperature in the crystal underneath the interface, thereby decreasing the ability for conduction to dissipate the latent heat through the crystal. Interestingly, a negative value of β has the converse effect. By improving the driving force for conduction out of the crystal, latent heat is more easily removed and the interface flattens.

The two cases clearly demonstrate that the interface is more sensitive to the furnace perturbation under lower gradient conditions, Figure 4.8(a), than at higher gradients, Figure 4.8(b). In fact, the absolute movement of the centerline interface position caused by changing the value of β is over 20 times greater for the low-gradient system compared to the high-gradient case.

4.5 Closed-Loop Control Study

Closed-loop control is added to the model for CZT grown in a low-pressure EDG system based on ideas presented in the open-loop study. Three control scenarios are enacted: actuation along the entire external boundary, along the solid portion only, and along the melt portion only.

4.5.1 Model Formulation

In the closed-loop model, the magnification factor α is determined using a control algorithm instead of being chosen arbitrarily as with the open-loop analysis. The simulation is performed under a quasi-steady-state approximation, which is defined differently than in the open-loop study. For a typical profile translation rate of 1 mm/hr, the value used here, the time scale of crystal growth is two orders of magnitude greater than that of heat transfer ($t_{heat} = R^2 \rho C_p / k_m \approx 35$ min and $t_{growth} = V_a / R \approx 4370$ min). This allows us to safely ignore the effect of growth on heat transfer. However, interface motion does produce latent heat that must be accounted for. Therefore, we assume that the interface is moving at a constant rate equal to that of the furnace translation rate. From another perspective, if the interface moves at the same rate as the furnace temperature profile, the interface always sees the same thermal environment, ignoring end effects. We can view this as changing our frame of reference from the fixed furnace to the moving frame of reference of the interface. In this manner, we apply an alternate quasi-steady-state approximation where the furnace temperature profile is fixed with respect to the interface while the transport equations are integrated in time. The effect of translating the temperature profile must still be accounted for. This is accomplished by adding the translation rate to the rate of change of the interface position:

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} + V_a \quad (4.7)$$

where $d\mathbf{x}/dt$ is overall rate of change of the interface position, $\dot{\mathbf{x}}$ is the rate of change due to transforming the temperature profile, and V_a is the translation rate of the temperature profile. The quantity $\dot{\mathbf{x}}$ can be viewed as a perturbation to the interface shape caused by transforming the temperature profile through control.

The base case model is used as the initial condition of the system. We then use closed-loop control to go from this initial state, where the interface has significant interface deflection, to a state in which the interface is flat. Interface deflection is expressed as the difference in axial heights of the interface at the external boundary, X_H , and at the system

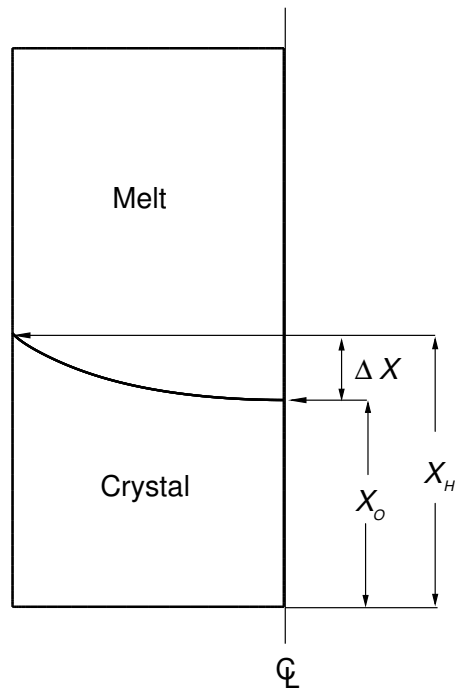


Figure 4.9: Diagram depicts interface deflection ΔX as the difference in axial height of melt-crystal interface at the external boundary to that at the centerline relative to the bottom of the computational domain.

centerline, X_O , as shown in Figure 4.9. The gradient magnification factor α is updated based upon the value of $\Delta X = X_H - X_O$. The following procedure is applied to the model:

1. Measure $\Delta X = X_H - X_O$,
2. Update α based on ΔX ,
3. Time integrate transport equations until next controller reading,

Step 1-3 are repeated until the set-point value of ΔX is reached. Afterward, the value of α is held constant. Since a proportional controller is used, we expect some offset in the final value of ΔX . While the choice of interface deflection as the controlled output is not always possible in a real system, this limitation is overlooked in that our immediate goal is to assess closed-loop control of a vertical Bridgman system in a general sense.

In this study, we use a simple proportional control algorithm to update the manipulated input α :

$$\alpha = \alpha_{nominal} + K\Delta X \quad (4.8)$$

where K is the proportional gain, and $\alpha_{nominal}$ is the value of α when $\Delta X = 0.0$ cm. The appropriate value of $\alpha_{nominal}$ is not known, so it is approximated as the previous value of α calculated by the algorithm. No upper limit is placed on α . In addition, it is assumed there is no time-lag in the response of the furnace to the controller, and that ΔX can be measured or accurately estimated.

Three different scenarios of actuation of the furnace temperature profile are examined. First, the new value of α is applied along the entire external boundary. Second, the new value of α is applied only in the solid region, with $\alpha = 1$ in the melt region. Finally, α is updated only in the melt region, with $\alpha = 1$ in the solid. A benefit of multi-zone furnaces, such as the EDG system, is that they are able to approximate such scenarios.

4.5.2 Results and Discussion

We consider the response of interface deflection under three different control scenarios. See Ref. [110] for physical property data and furnace geometry details. In each scenario, the controller updates the gradient multiplication factor, α , every 2 min, and is time integrated to a final time of 16 hrs. A set-point of $\Delta X = 0.0$ corresponding to a flat interface is used for all controller gains. Results of the three scenarios are shown in Figure 4.10 for gains of $K = 0.1, 0.2, 0.5, 1.0$, and 2.0 . Figures 4.10(a) and (d) show the response of the interface deflection, ΔX , and the corresponding value of α , respectively, for the case of control along

the entire external boundary. As expected, the response of the interface deflection is faster with higher gains. Only the case of $K = 2.0$ reaches closely the set-point value $\Delta X = 0.0$ cm after 16 hrs, with a final deflection value of $\Delta X = -4 \times 10^{-3}$ cm indicating a slightly convex shape. Offset from the set-point value is due to the use of a simple proportional controller. In the other cases, the interface deflection asymptotically approaches the set-point value but does not reach it. As seen in Figure 4.10(d), the value of α increases rapidly at initial times, but this increase slows down significantly as the interface deflection approaches the set-point value. In the cases where $K < 2.0$, α continues to slowly increase with time because the set-point has not been reached. When $K = 2.0$, a final value of $\alpha = 19.3$ is reached for an overall temperature gradient of 38.6 K/cm, a large value but one attainable by the furnace. The results imply that the controller should be tuned using a gain of at least $K = 2.0$ in order to reach the set-point value. For values of $K < 2.0$, the response of the controller is not aggressive enough to overcome the competing forces that produce a concave interface. These competing forces are further discussed below.

At long times, the difference in final values of ΔX for the different gains is relatively small compared to the difference in the final values of α . The difference in final values of α for $K = 2.0$ and $K = 0.1$ is $\Delta\alpha = 13.6$ with a corresponding difference in deflection of 0.098 cm, or 2% with respect to the radius. This indicates that effectiveness of increasing α diminishes, and that there is an optimal value for α when actuation is applied along the entire external boundary.

The results for the case of actuation along the solid portion of the boundary alone are shown in Figures 4.10(b) and (e). In this scenario, the controller is not only able to flatten the interface for every controller gain tried, but also produces the favorable convex interface shape for every controller gain through offset from the proportional controller. Producing a convex interface shape is an unexpected benefit from the controller. The response of ΔX is faster with higher gains as before, as seen in Figure 4.10(b), but the amount of offset increases with increasing gain. From another point of view, the convexity of the interface increases with increasing gain. Figure 4.10(e) shows that α initially increases rapidly for each gain, and then stays at a constant value once the set-point value is reached. The interface deflection reaches the set-point $\Delta X = 0.0$ faster than in the previous scenario and for smaller final values of α , indicating that control targeted on the solid is more effective than control along the entire boundary. The behavior agrees with the results of the open-loop study, where it was shown that the interface shape is more sensitive to the amount of axial heat conduction available through the solid. By increasing the axial temperature gradient in the solid region, more latent heat can be conducted away from the

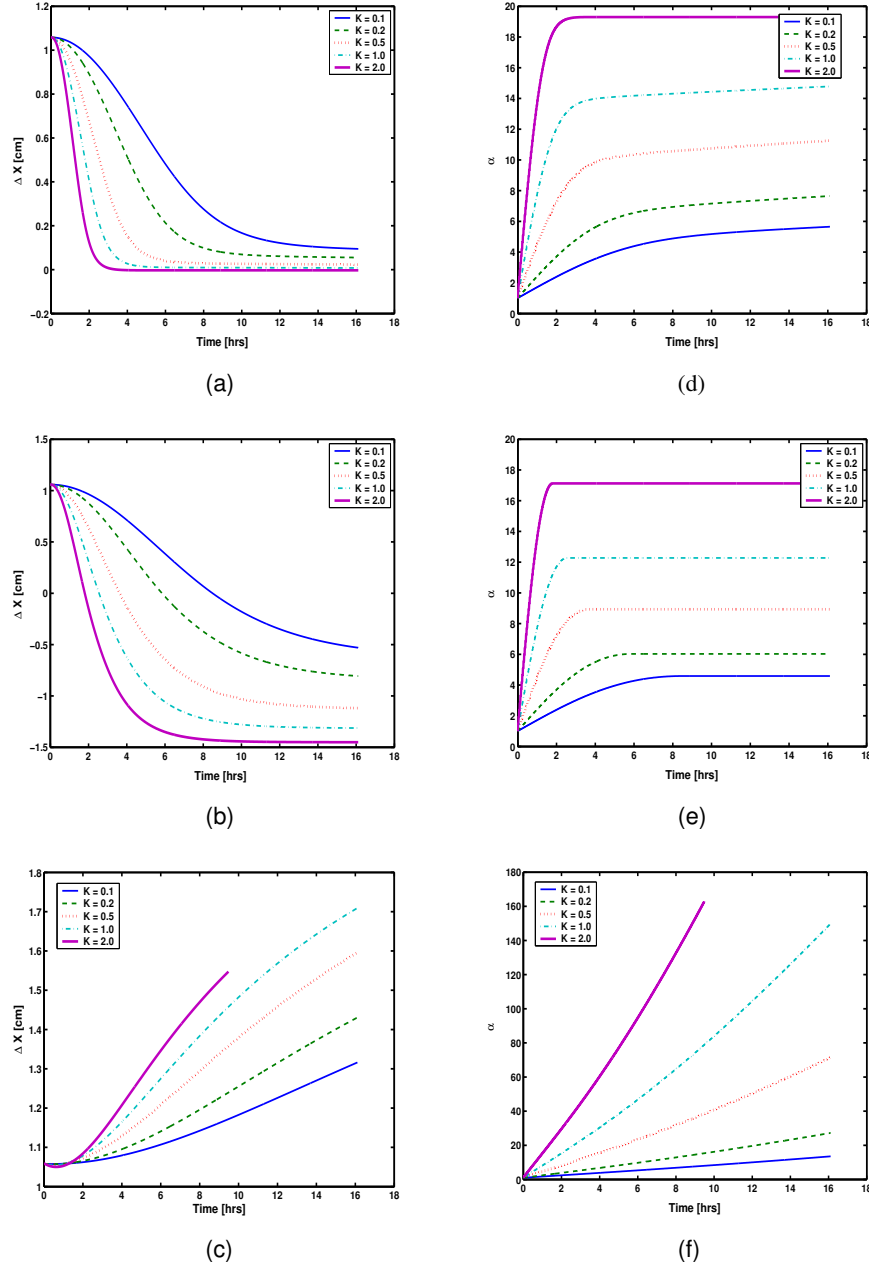


Figure 4.10: Proportional control simulation results for control along (a) & (d) the entire boundary, (b) & (e) crystal only, and (f) & (g) melt only. Plots (a)-(c) are of interface deflection ΔX versus time, and plots (e)-(g) are of gradient multiplication factor α versus time.

interface axially giving a flatter interface.

Unlike the previous case, there is a significant difference in the final interface deflections relative to the corresponding difference in final values of α . The difference in final values of α for $K = 2.0$ and $K = 0.1$ is $\Delta\alpha = 12.5$ with a corresponding difference in deflection of 0.92 cm, or 20 % with respect to the radius, an order of magnitude larger than before. This suggests that interface shape is more sensitive to increased temperature gradients in the solid than when actuation is applied along the entire external boundary, or possibly the range where ΔX becomes less sensitive to α was not reached with the gains used here. Figure 4.11 shows representative results for temperature and streamline contours at $t = 16$ hrs for the scenario of control along the solid boundary only, $K = 2.0$. The interface clearly has a convex shape. The isotherms in the melt are nearly flat, as are the isotherms in the solid farther away from the interface as seen in Figure 4.11(a). Solid streamline contours, Figure 4.11(b), near the interface indicate flow in the counter-clockwise direction such that fluid flows up the furnace boundary, and down the system centerline. Therefore, the system is *cooler* in the center than at the wall, which is not the case under typical growth conditions where the release of latent heat overwhelms the radial gradient from the furnace.

Figures 4.10(c) and (f) show the results for when control is applied only to the melt portion of the external boundary. At time $t > 1$ hr, the interface deflection decreases slightly, but begins to increase after $t > 1$ hr leading to an unstable situation. For $K = 2$, time integration failed at approximately $t \approx 10$ hrs due to deformation of the computational mesh. While the open-loop study showed that increasing the axial temperature gradient in the melt did not improve the interface deflection in a steady-state, there was no indication that this would cause an unstable behavior. Increasing the axial temperature gradient in the melt causes an increase in the overall thermal load that must be conducted away through the solid for proper heat balance. Since the axial gradient through the solid is fixed at the relatively small 2 K/cm, heat must be conducted away radially through the solid, causing the interface to curve more. As the temperature gradient in the melt is further increased, or α increased, the deflection, ΔX also increases giving rise to the unstable behavior.

By comparing the three cases, it can be seen when actuation is applied along the entire boundary, two competing forces are applied to the interface. Increased axial heat transfer through the solid flattens the interface, which is counteracted by the response to increased temperature gradient in the melt. If actuation is applied along the entire external boundary, large gains are needed which give relatively large final values of α . In

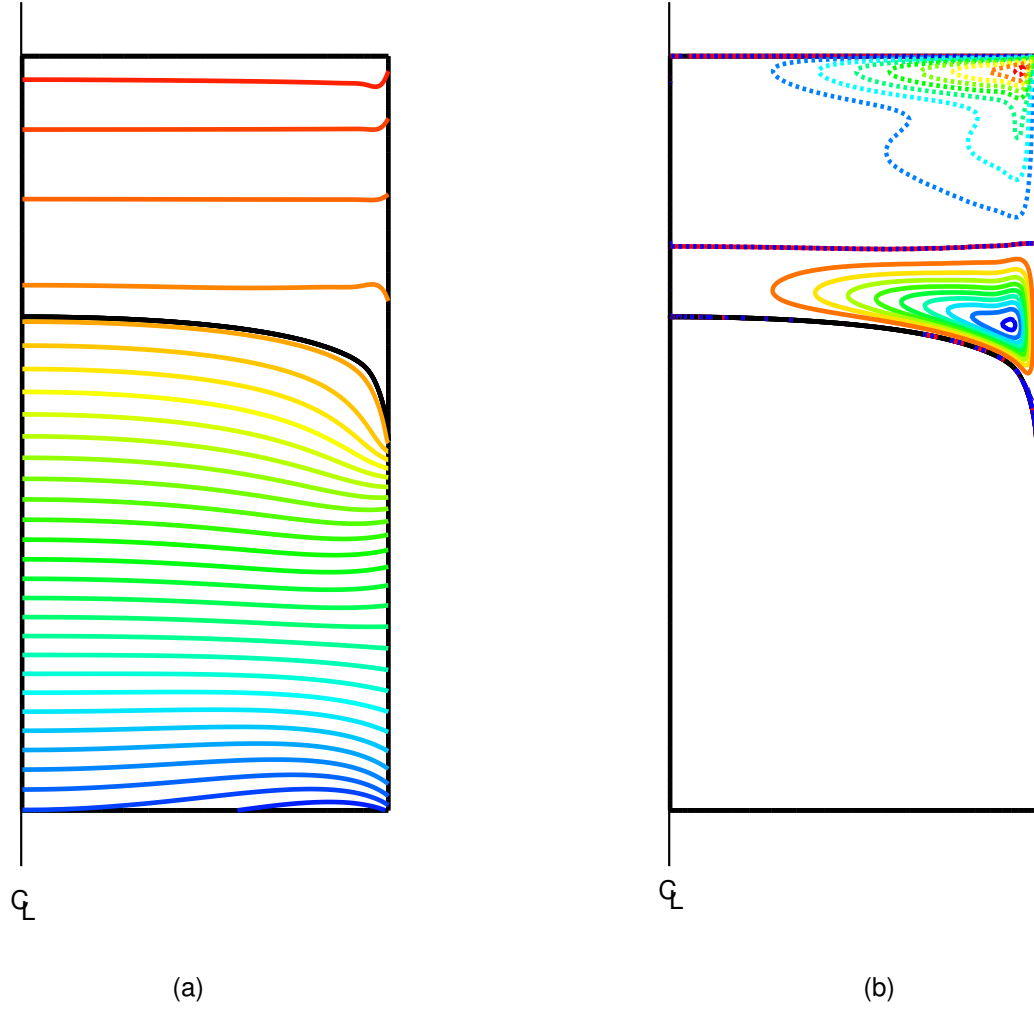


Figure 4.11: Representative system results at $t = 16$ hrs for closed-loop control solely along the solid portion of the furnace boundary. Temperature contours, shown in (a), are spaced at $\Delta T = 5$ K, where $T_{min} = 1234$ K and $T_{max} = 1389$ K. Streamline contours, shown in (b), are plotted at intervals of $\Delta\psi = 2 \times 10^2$ cm³/s, $\psi_{min} = -1.8 \times 10^{-1}$ cm³/s, and $\psi_{max} = 1.6 \times 10^{-1}$ cm³/s. Solid streamline contours indicate flow in the counter-clockwise direction, while dashed contours indicate flow in the clockwise direction.

conventional growth of CZT, a large gradient is applied along the entire outer boundary. As shown here, however, the ability of a large value of α to flatten the interface diminishes as α is further increased. Alternatively, control action targeted on the solid gives the fastest and most effective response, and showed the surprising benefit of a convex interface shape. By targeting future furnace design on increasing axial heat transfer through the solid only, it may be possible to improve significantly the melt-crystal interface shape.

4.6 Concluding Remarks

We have employed a numerical model to assess how furnace design affects the growth of CZT crystals via vertical Bridgman processes, and to explore the feasibility of closed-loop control to improve these crystals. Our approach is a cost-effective model that has been adapted from the results of the multi-scale, self-consistent model of Pandey et al. [136]. With this modeling approach, we are able to examine the open-loop response of the system from low- to high-growth conditions and the effects of perturbations to the thermal profile on the melt-crystal interface shape. We also compare the growth conditions applied in eV Products' low-pressure EDG system to that of their larger, higher-gradient HP-EDG system [144] with the open-loop model. In addition, we incorporate a control methodology that changes the furnace design by updating the furnace temperature gradient to flatten the interface shape.

Our open-loop results show that the different interface shapes of the low-pressure and high-pressure growth systems is explained by a continuous evolution from a significantly deflected shape [136] to a much flatter shape [144] that is driven by the temperature gradients in the system. The physical mechanism behind this evolution is explained by the balance among heat fluxes through the melt (dominated by convection) and crystal (by conduction only) and, especially, the heat released at the melt-crystal interface by phase change (latent heat). Interestingly, changing the gradient in the melt had a relatively minor effect on the system, since radial heat transfer is dominated by strong, nonlinear flows and convective mixing in these high-Prandtl-number liquids. Of overriding importance in these systems is heat conduction through the crystal, which is strongly affected by the temperature profile of the furnace below the melting point. Interface flatness is primarily determined by the ability of latent heat to be conducted away from the interface, through the crystal, in the axial direction. If axial conduction is too little, as is the case when gradients are small, the interface deflects so that radial conductive fluxes can dissipate latent heat. On the other hand, a higher applied gradient promotes the axial conduction

of latent heat, and the interface flattens.

Consistent with these mechanisms, our results also demonstrated that small changes to the furnace profile will have a stronger effect on changing the interface shape in lower-gradient systems. Here, small perturbations to the temperature profile applied to the crystal were able to move the interface, but this effect was only significant in the low-gradient system.

The close-loop response model was developed based on the results obtained from the open-loop model. The results of the closed-loop model show that a simple proportional controller is able to flatten the interface by changing the overall gradient of the furnace temperature profile. The system responds faster with higher gains, as expected. However, a larger gain also gives a relatively large final value of the multiplication factor α compared to small differences in the final interface deflection. This shows there is limited effectiveness in further increasing the temperature gradient along the entire boundary to change the interface shape.

When the controller algorithm acts only on a targeted area, the response of the interface changes. In the case where the furnace temperature profile in the melt section is fixed and updated in the solid portion, the interface deflection reaches the set-point $\Delta X = 0.0$ faster than if the entire profile is updated. An unexpected boon of a convex interface shape was achieved because of offset from the set-point in the proportional controller. If, however, the furnace temperature profile is updated in only the melt portion, the response is unstable, and the interface deflection continues to increase.

Achieving a convex interface shape is exciting and wholly unexpected. The results indicate that it may be possible to improve the furnace design to produce a convex interface shape for CZT growth. Current practice applies a large temperature gradient along the entire length of the crystal. By keeping the temperature gradient small in the melt region, and increasing axial heat conduction through the solid, it may be possible to achieve a convex interface. The results from both open- and closed-loop models are consistent with those computed by Kuppurao et al. [95], where the relative balance of axial and radial fluxes, as affected by the design of the ampoule support, was shown to affect interface shape of CZT in the cone region of the ampoule.

The controller presented here is relatively simplistic, and more of a proof-of-concept test than a rigorous control implementation. The next step is to include integral action to the control law for proportional-integral (PI) control. A more complicated control strategy such as controlling the individual heaters of the furnace would be a more realistic study instead of solely adjusting the gross furnace temperature gradient. In sophisticated multi-

zone furnaces such as the EDG system, heaters have their own power inputs making it possible to control individual heater powers. Nonlinear control is another option for such a highly nonlinear problem.

Additional complexities can be added to the model for a more realistic simulation. Including the material in the cone section of the ampoule would eliminate transient end effects for a fully transient simulation. With a transient model, we can investigate the ability of closed-loop control to keep the interface deflection flat, or at a specific value, during growth. Optimization of the furnace temperature profile could be achieved by solving the steady-state problem inversely to calculate the necessary temperature boundary condition, such as $\alpha_{nominal}$, to have a flat interface. Another possibility is to employ an adaptive algorithm to estimate $\alpha_{nominal}$ such as a secant method that updates $\alpha_{nominal}$ based on difference in the previous values. Control of melt crystal growth systems is a rich area of study, and work is ongoing to explore its potential.

Part III

Algorithm Developments

Chapter 5

Multi-Scale Model via an Approximate Block Newton Method

Contents

5.1	Chapter Summary	89
5.2	Introduction	90
5.3	Approximate Block Newton Method	96
5.4	Example Problems	107
5.4.1	One-Dimensional Model	107
5.4.2	Two-Dimensional Model	114
5.4.3	Electrodynamic Gradient Freeze Furnace Model	127
5.5	Concluding Remarks	134

5.1 Chapter Summary

Modular coupling of numerical codes will enable the development of increasingly more sophisticated numerical models that maximize effectiveness, capability, and flexibility while keeping development costs down. In this chapter, we present a modular coupling approach of a melt crystal growth system, where the code CrysVUn is used to compute furnace-scale heat transfer and the code Cats2D is used to calculate melt fluid dynamics and phase-change phenomena. Previously a coupling scheme known as the block Gauss-Seidel method

was used to couple these two codes together which proved to be unreliable with regards to convergence [135, 136, 196]. Here, a strategy called the Approximate Block Newton method is applied to overcome the shortcomings of the block Gauss-Seidel method. A direct comparison of the two methods is demonstrated via a one-dimensional heat transfer problem that is representative of melt-crystal growth. In addition, results of a two-dimensional melt crystal growth model are presented, where the two codes, CrysVUn and Cats2D, are coupled using an Approximate Block Newton method. Finally, initial results of a complex, multi-scale model of an industrial gradient freeze furnace constructed using the Approximate Block Newton method are presented. The Approximate Block Newton method proves to be far superior to the block Gauss-Seidel algorithm in terms of convergence and efficiency.

5.2 Introduction

Computational modeling has become increasingly more sophisticated as the technology advances, as has the problems we are interested in solving. To incorporate added complexity and level of detail to existing modeling capability, we can either become experts in additional physical phenomena or incorporate work done by others into existing models. A model that simultaneously represents all chosen phenomena in all domains in one large mathematical model is referred to as a *direct-coupling*, *monolithic*, or *analytic* approach [29]. This course of action may be generally preferred, since all interactions can be accounted for simultaneously and self-consistently. However, such an approach may be undesirable or infeasible, e.g., when the computational or software development costs for solving all of the assembled governing equations is too high or when subprocess models exist but employ commercial software, preventing access to the source code.

Due to these challenges in developing *direct-coupling* models, there is an increasing need for innovative ways to couple existing software that solve specific problems, especially for multi-scale and multi-physics problems [84]. Therefore, researchers have resorted to a *partitioned*, *modular*, or *synthetic* approach in which different models and solvers are used together [29, 62]. In this manner, software tools that have the desired added technology can be exploited to enhance existing capabilities.

Partitioned coupling is generally split into two categories: *weak* and *strong* coupling [115, 116]. The distinction between the two is primarily due to the manner in which time integration is handled. In the *weak*, also known as *loose* coupling, the partitioned subproblems are time integrated in an explicit manner, the simplest of which is known as *staggering*. For more examples of this method, see [62]. The *strong*, or *tight*, coupling

method uses implicit time integration between the subproblems. The definitions of weak and strong coupling can also be applied to steady-state problems. In this context, coupling is along the lines of solver iteration instead of time steps. Block Gauss-Seidel is a form of weak coupling of steady-state problems, and the Approximate Block Newton method is a form of strong coupling. Matthies and Steindorf [115] give a detailed treatment of the distinction between *weak* and *strong* coupling.

Modular coupling of numerical software is applicable to fields that deal with multi-scale and multi-physics issues. Much work on coupling such problems has been in the field of fluid-structure interaction problems [10, 15, 50, 52, 120, 122], with applications in aeroelasticity [61, 121, 139, 140] and biomechanics [53]. Coupled models have also been developed to study thermomechanical response [3], soil-pore fluid interaction [200], turbomachinery problems [49], conjugate heat transfer [35, 73, 137, 161, 162], combustion [36], and many others. In this work we apply a *strong* modular coupling method to develop a multi-scale melt-crystal growth model.

Crystal growth incorporates a wide range of physical phenomena over a large extent of length (atoms to an entire furnace) and time (picoseconds to months) scales [43, 46]. Our goal is to develop a quantitatively predictive model for melt-crystal growth processes by coupling two independent models, specifically, models of the melt-growth furnace and the growing crystal. The furnace, or *global* model, computes high-temperature heat transfer by radiation and conduction, and is typified by complex geometry. Within the furnace, crystal growth occurs in a container known as an ampoule. The model of the growing crystal, or *local* model focuses on the physics within the ampoule during growth: melt convection, heat transfer, and melt-crystal interface shape. While the geometry of the ampoule is relatively simple—often a cylinder—the physics occurring within the ampoule exhibit strong interactions. Monolithic models of such problems have been attempted infrequently due to the challenges in portraying all of the necessary features realistically.

Research to develop models of crystallization in melt-growth furnaces can be divided along two general paths: developing a global model of heat transfer in the furnace and concentrating on transport phenomena occurring on the local level in the growing crystal. There are several noteworthy accomplishments to construct global models of melt-growth furnaces. Dupret et al. developed the numerical code FEMAG [55] for predicting heat transfer in Czochralski (CZ) crystal growth; and Müller et al. produced the code CrysVUn [8, 97], originally for predicting heat transfer in vertical gradient freeze processes. Even though both codes have been enhanced to include phase-change phenomena and convection effects, the codes expertise is primarily the simulation of radiation heat transfer in high tempera-

ture systems on the global level. Crystal growth models on the local level have typically focused on solving interacting transport phenomena with an emphasis on fluid dynamics and interfacial phenomena, while treating heat exchange with the furnace environment in a simplified manner. Excellent overviews of such models are the still-relevant review by Brown [23] and more recent reviews by Müller [124], Lan [102], and Yeckel and Derby [193].

Constructing a multi-scale, melt-growth model by coupling a global code for furnace heat transfer with a local code for melt convection has been attempted previously. Baumgartl et al. [14] applied temperature conditions from a global heat transport simulation to compute melt convection in a Czochralski crystal growth process. Ouyang and Shyy [133, 134] developed a two-level approach to incorporate heat transfer in a Bridgman system into a crystal growth model. The first level of this model represented the furnace with a coarse discretization of the ampoule and contents. The second level applied a finer discretization only on the model representing crystal growth within the ampoule. Tomás et al. [176] used a similar two-level approach to model cadmium telluride growth in vertical Bridgman systems, as did Rule [148] to model cadmium zinc telluride growth. Virbulis et al. [181] employed FEMAG [55] and the commercial code CFD-ACE in a coupled manner to analyze silicon CZ flow. Virbulis et al. [181] employed FEMAG [55] and the commercial code CFD-ACE in a coupled manner to analyze silicon CZ flow. Lukanin et al. [109] presented a coupled model for two-dimensional global heat transfer code with three-dimensional turbulent flow computations for silicon Czochralski growth analysis. Yeckel et al. [189, 191] have applied two-dimensional furnace heat transfer analyses to supply boundary conditions to three-dimensional local computations.

However, it is unclear in most of these studies if the final model is self-consistent such that the temperatures and heat fluxes are in agreement throughout the entire domain. Some models only used one-way coupling where temperature boundary conditions are supplied to the local model after computing furnace heat transfer with a coarse-scale representation of the melt and crystal [14, 133, 134, 176, 189, 191]. One-way coupling ignores the change in the furnace heat load caused on the local level such as melt convection and latent heat generation at the melt-crystal interface. Others [109, 181] have reported iterating between global and local models, but supply little details about the information exchanged or the convergence tolerances used to determine if the final solutions actually satisfied the governing equations of each model.

In recent work, Pandey et al. [136] presented a self-consistent, multi-scale model to study the growth of single crystal cadmium zinc telluride in an industrial electrodynamic gradient freeze furnace. This model coupled a global model of the furnace made with the

commercial code CrysVUn [8, 97] with a local model of growing crystal simulated with the code Cats2D [195]. The two computational domains were joined along a shared, one-dimensional boundary with no overlapping of the separate computational meshes. The codes were treated as two black boxes that shared the boundary temperature and heat flux data. The study examined the errors produced in the final solutions introduced by one-way coupling compared to a fully self-consistent solution. This study also showed that the convergence of the iterative computations depended upon how the two models were coupled, specifically on how the boundary conditions along the shared boundary were formulated. This work presents an alternative coupling algorithm to link the codes CrysVUn and Cats2D that is more robust than that used by Pandey et al. [136].

Before beginning our in-depth discussion, we wish to make some comments on the mathematical notation used in this chapter. Matrices are denoted by boldface upper-case Roman letters, and vectors by boldface lower-case Roman letters. Scalar quantities are represented by lower-case Greek letters. Function evaluations are indicated by parentheses ($\cdot$). Multiplication of quantities is denoted by curly brackets, $\{\cdot\}$.

The coupling method employed by Pandey et al. [136] is known as the block Gauss-Seidel (BGS). As a general example, we consider a problem that can be divided into two subproblems that are coupled along a boundary. These subproblems can be denoted as

$$\mathbf{r}_1(\mathbf{x}, \tilde{\mathbf{y}}) = \mathbf{0} \quad (5.1)$$

$$\mathbf{r}_2(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbf{0} \quad (5.2)$$

where $\mathbf{r}_i$ is a residual vector associated with problem i , $\mathbf{x}$ denotes the unknowns of problem 1, and $\mathbf{y}$ denotes the unknowns of problem 2. The tilde indicates that these variables are from the other model, and fixed. We begin by assuming each subproblem has its own solver, and that these solvers can be written in fixed-point form with convergent internal iterations

$$\mathbf{x}^{(k+1)} = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}) \quad (5.3)$$

$$\mathbf{y}^{(k+1)} = \mathcal{G}(\tilde{\mathbf{x}}, \mathbf{y}^{(k)}) \quad (5.4)$$

where the solvers of r_1 and r_2 are denoted by the script letters $\mathcal{F}$ and $\mathcal{G}$, respectively. The variables designated with the tilde are held constant during the iteration. The index k indicates the *global*, or outer, iteration number corresponding to a converged solution of each subproblem. Each subsolver performs its own internal iterations to calculate the

updated global iterate, but we do not explicitly denote these iterations.

The BGS algorithm for solving Equations (5.1) and (5.2) is:

1. $\mathbf{x}^{(k+1)} = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}^{(k)})$, with $\tilde{\mathbf{y}}^{(k)}$ held constant.
2. $\mathbf{y}^{(k+1)} = \mathcal{G}(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k)})$, with $\tilde{\mathbf{x}}^{(k+1)}$ held constant.
3. Repeat until the global model is converged. Global convergence is achieved when both original residuals are satisfied, (i.e., the norms $\|\mathbf{r}_1(\mathbf{x}^{(k+1)}, \tilde{\mathbf{y}}^{(k+1)})\| < \epsilon_1$ and $\|\mathbf{r}_2(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k+1)})\| < \epsilon_2$).

The appeal of the block Gauss-Seidel method is that it is relatively easy to understand and to implement.

Note that each subproblem does not need to be solved such that it satisfies some internal convergence criteria. One may choose to define Equations (5.1) and (5.2) as completely converged solutions or as the solutions of one internal iteration of their respective sub-solvers. For example, let's assume Newton's method is used by both solvers. Then each solver performs one internal Newton iteration before exchanging information. In either definition, the same global convergence criteria is applied ($\|\mathbf{r}_1(\mathbf{x}^{(k+1)}, \tilde{\mathbf{y}}^{(k+1)})\| < \epsilon_1$ and $\|\mathbf{r}_2(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k+1)})\| < \epsilon_2$) to each subproblem to judge overall convergence.

Solving the CrysVUn-Cats2D model shown in Reference [136] with the BGS method was difficult to achieve. Complicating the issue was the large number of degrees of freedom in each submodel and the interaction of many different governing physical phenomena such as radiation and momentum transport. These issues made it difficult to pinpoint why BGS did not work consistently, and barred deeper exploration of the mechanics involved in the coupling method.

In a companion piece to Pandey et al. [136], Yeckel et al. [196] explored the issues of convergence of the block Gauss-Seidel method by viewing the overall problem as a single fixed-point iteration. A simple one-dimensional problem representative of crystal growth that had an analytical solution was developed specifically to examine convergence behavior. Convergence of the block Gauss-Seidel method to solve the coupled model was found to be strongly dependent on the boundary conditions applied at the shared boundary. These boundary conditions are a mixture of temperatures and heat fluxes that are dependent on "mixing" parameters, one for each subproblem (see Equations (5.70) and (5.71)). The values chosen for the mixing parameters, in effect, determined whether or not the coupled model would converge. However, appropriate values for mixing parameters to achieve

convergence were difficult to chose *a priori*, and would actually change depending on the nonlinearity of the test problem.

Implementation of under-relaxation [39] within a block Gauss-Seidel iteration by modifying the above procedure is also examined in Derby et al. [45], (see Appendix A). With the addition of under-relaxation, convergence characteristics of the one-dimensional, linear test problem improved slightly but actually performed worse for the nonlinear case. In both cases the extra cost in computation time outweighed the marginal gain in performance.

Ideally, full Newton-Raphson method would be applied to the overall system of equations, which is robust with respect to convergence in that Newton-Raphson is guaranteed to converge if the initial guess is suitably close to the solution [41]. To see how Newton-Raphson can be applied to the coupled system, we form a block residual “vector” made of Equations (5.1) and (5.2)

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix}. \quad (5.5)$$

A full Newton-Raphson step of this residual is then:

$$\mathbf{J}\Delta\mathbf{u} = -\mathbf{r} \quad (5.6)$$

where $\mathbf{J}$ is the Jacobian matrix of residual vector $\mathbf{r}$, and $\Delta\mathbf{u}$ is the update vector of the unknowns. Equation 5.6 can be rewritten in block form

$$\begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} \quad (5.7)$$

where $\mathbf{J}_{ij}$ is the block Jacobian matrix corresponding to $\mathbf{r}_i$ with respect to either $\mathbf{x}$ or $\mathbf{y}$. For example, $\mathbf{J}_{11}$ is defined the partial derivative of vector $\mathbf{r}_1$ with respect to vector $\mathbf{x}$. The $(i, j)^{th}$ component of matrix $\mathbf{J}_{11}$ is then the partial derivative of the i^{th} component of vector r_1 with respect to the j^{th} component of vector $\mathbf{x}$:

$$\mathbf{J}_{11}^{i,j} = \frac{\partial \mathbf{r}_1^i}{\partial \mathbf{x}^j} \quad (5.8)$$

where the superscripts indicate matrix/vector indices. From now on we will write this partial derivative using short-hand notation such as $\mathbf{J}_{11} \equiv \partial\mathbf{r}_1/\partial\mathbf{x}$ and $\mathbf{J}_{12} \equiv \partial\mathbf{r}_1/\partial\mathbf{y}$. The block vectors $\Delta\mathbf{x} \equiv \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ and $\Delta\mathbf{y} \equiv \mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}$ are the update vectors of subproblems 1 and 2, respectively.

In a modular setup, however, the off-diagonal blocks, $\mathbf{J}_{12}$ and $\mathbf{J}_{21}$ are inaccessible

because we do not know the sensitivity of subproblem 1, $\mathbf{r}_1$, to the variables of subproblem 2, $\mathbf{y}$ and vice versa. If these off-diagonal blocks are set to zero, we revert back to the block Jacobi method:

$$\begin{bmatrix} \mathbf{J}_{11} & 0 \\ 0 & \mathbf{J}_{22} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix}. \quad (5.9)$$

and the two submodels are solved in a decoupled manner. Equation 5.9 becomes the BGS method if the two parts are solved sequentially. However, these methods converge slowly or solutions cannot even be obtained, as shown in Yeckel et al. [196]. By including the information provided in the off-diagonal blocks, we hope to improve the robustness with respect to convergence of the multi-scale model. With this goal in mind, we present the Approximate Block Newton method.

5.3 Approximate Block Newton Method

The Approximate Block Newton (ABN) method approximates a global, coupled Newton iteration using the solvers of the individual systems. The ABN method was first proposed by Chan [30] as a means of performing continuation in parameter space. Artlich and Mackens [4] expanded the method to be used as solver for a multi-physics problem with a non-sparse Jacobian, while Menck [117] proves convergence and further proposes a control mechanism to optimize efficiency. Matthies and Steindorf [115, 116] base their implementation of ABN method on that of Artlich and Mackens [4] to couple two separate software codes to solve a transient fluid-structure interaction problem. We propose using the ABN method to solve a self-consistent, multi-scale, steady-state, crystal growth problem, and illustrate the method using one- and two-dimensional example problems representative of crystal growth. We also present a two-dimensional, coupled model of an electric dynamic gradient (EDG) freeze furnace similar to that presented in [136] that is solved using the ABN method.

First, we give a detailed explanation of the ABN method, and then introduce the example problems. The procedure that we outline here is an alternative derivation from both Artlich and Mackens [4] and Matthies and Steindorf [115, 116]. We then explore properties of the ABN method and compare with the convergence of block Gauss-Seidel method.

Here, we wish to consider two problems, which may be nonlinear, that are coupled along a common boundary. In this coupling scheme, we consider only information that passes between the two models along the common boundary. We begin with the two subproblems,

Equations (5.1) and (5.2), and apply Newton-Raphson to solve these equations as outlined in Equation (5.7).

We choose to present our derivation in terms of in the original residual equations, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2)^T$, and assume that each subsolver uses Newton's method. Artlich and Mackens [4] use a similar approach, but assume each solver uses an arbitrary solution method. Furthermore, they rely on the identities

$$\mathbf{r}_1(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathcal{F}(\mathbf{x}, \mathbf{y}) \quad (5.10)$$

$$\mathbf{r}_2(\mathbf{x}, \mathbf{y}) = \mathbf{y} - (G)(\mathbf{x}, \mathbf{y}), \quad (5.11)$$

holding, which is true at convergence. We do not use such a restriction. Matthies and Steindorf [115, 116] present their derivation in terms of the fixed-point solvers $\mathbf{r} = (\mathbf{f}, \mathbf{g})^T$, where

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathcal{F}(\mathbf{x}, \mathbf{y}) = \mathbf{0} \quad (5.12)$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{y} - \mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathbf{0}.$$

without stating anything about the nature of their solution methods. We choose to present our derivation in an explicit manner, as we are interested in coupling codes that specifically use Newton's method.

As stated before, we begin by apply Newton's method to Equations (5.1) and (5.2) to get Equation (5.7). Both subsolvers use Newton's methods to give

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \mathbf{J}_{11}^{-1} \mathbf{r}_1 = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}) \quad (5.13)$$

$$\mathbf{y}^{(m+1)} = \mathbf{y}^{(m)} - \mathbf{J}_{22}^{-1} \mathbf{r}_2 = \mathcal{G}(\tilde{\mathbf{x}}, \mathbf{y}^{(k)}) \quad (5.14)$$

where l and m indicate the iteration counters local to each problem. The block-diagonal Jacobian matrices, $\mathbf{J}_{11}$ and $\mathbf{J}_{22}$ are the Jacobian matrices from the two subsolvers. The above equations can be rewritten in the form of Equation 5.12 such that

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathcal{F}(\mathbf{x}, \mathbf{y}) = \mathbf{J}_{11}^{-1} \mathbf{r}_1 \quad (5.15)$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{y} - \mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathbf{J}_{22}^{-1} \mathbf{r}_2 \quad (5.16)$$

where the subsolvers each take only one local, internal Newton iteration.

The global Jacobian is then preconditioned with a block diagonal preconditioner based

on the solvers for the subproblems:

$$\begin{bmatrix} \mathbf{J}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{22}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{J}_{11}^{-1} \mathbf{r}_1 \\ \mathbf{J}_{22}^{-1} \mathbf{r}_2 \end{bmatrix}. \quad (5.17)$$

When multiplied through, the block matrix equation becomes:

$$\begin{bmatrix} \mathbf{I} & \mathbf{J}_{11}^{-1} \mathbf{J}_{12} \\ \mathbf{J}_{22}^{-1} \mathbf{J}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{J}_{11}^{-1} \mathbf{r}_1 \\ \mathbf{J}_{22}^{-1} \mathbf{r}_2 \end{bmatrix}. \quad (5.18)$$

Next, Block-Gauss elimination is applied to Equation (5.18) to yield:

$$\begin{bmatrix} \mathbf{I} & \mathbf{J}_{11}^{-1} \mathbf{J}_{12} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{J}_{11}^{-1} \mathbf{r}_1 \\ \mathbf{r}_S \end{bmatrix} \quad (5.19)$$

where

$$\mathbf{S} \equiv \mathbf{I} - \mathbf{J}_{22}^{-1} \mathbf{J}_{21} \mathbf{J}_{11}^{-1} \mathbf{J}_{12} \quad (5.20)$$

is the Schur complement, and

$$\mathbf{r}_S \equiv \mathbf{J}_{22}^{-1} \mathbf{r}_2 - \mathbf{J}_{22}^{-1} \mathbf{J}_{21} \mathbf{J}_{11}^{-1} \mathbf{r}_1 \quad (5.21)$$

is its modified right-hand side. These last two equations can be further simplified by defining

$$\mathbf{C} \equiv \mathbf{J}_{11}^{-1} \mathbf{J}_{12}, \quad (5.22)$$

$$\mathbf{q} \equiv -\mathbf{J}_{11}^{-1} \mathbf{r}_1. \quad (5.23)$$

While it appears that the inverses of the block Jacobians must be calculated, we will show that this preconditioning step need not be carried out explicitly. Block backward substitution is used to solve Equation (5.19) for $\Delta \mathbf{x}$ and $\Delta \mathbf{y}$:

$$\Delta \mathbf{y} = -\mathbf{S}^{-1} \mathbf{r}_S \quad (5.24)$$

$$\Delta \mathbf{x} = \mathbf{q} - \mathbf{C} \Delta \mathbf{y} \quad (5.25)$$

The following procedure is used to solve one iteration of the ABN iteration, evaluated

at $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$:

1. Solve $\mathbf{J}_{11}\mathbf{q} = -\mathbf{r}_1$ for $\mathbf{q}$
2. Compute the modified right hand side $\mathbf{r}_S = \mathbf{J}_{22}^{-1}(\mathbf{r}_2 + \mathbf{J}_{21}\mathbf{q})$
3. Solve $\mathbf{S}\Delta\mathbf{y} = -\mathbf{r}_S$ for $\Delta\mathbf{y}$
 - (a) Solve $\mathbf{J}_{11}\mathbf{C} = \mathbf{J}_{12}$ for $\mathbf{C}$
 - (b) Compute the Schur complement $\mathbf{S} = \mathbf{I} - \mathbf{J}_{22}^{-1}\mathbf{J}_{21}\mathbf{C}$
4. Compute $\Delta\mathbf{x} = \mathbf{q} - \mathbf{C}\Delta\mathbf{y}$
5. Update unknowns, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}$ and $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \Delta\mathbf{y}$

We will now explain each step in detail.

In the first step, it is easily seen that vector $\mathbf{q}$ can be viewed as the update vector in one Newton-Raphson step to solve $\mathbf{r}_1(\mathbf{x}, \mathbf{y}^{(k)}) = \mathbf{0}$ for the $\mathbf{x}$ unknowns with fixed $\tilde{\mathbf{y}} = \mathbf{y}^{(k)}$

$$\mathbf{q} = -\mathbf{J}_{11}^{-1}\mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}). \quad (5.26)$$

Only one Newton iteration is taken to solve for $\mathbf{q}$, and the linear system of equations is solved using whatever method is internal to subproblem 1. Vector $\mathbf{q}$ can be viewed as the update vector from one Newton step of solver 1 if subproblem 1 is solved on its own. Equation (5.26) can be rewritten in terms of an arbitrary fixed-point solver using the definition supplied in Equation (5.15)

$$\mathbf{q} = \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathbf{x}^{(k)}. \quad (5.27)$$

We view $\mathcal{F}$ as a black box which we use to mimic the Newton step of Equation (5.26). Again, only one local iteration of $\mathcal{F}$ is required.

Before going into detail of Step 2, we wish to introduce the concept of a Jacobian-Free method. In the ABN method, we have many instances of matrix-vector multiplication of a Jacobian matrix multiplied by some vector. For example, Step 2 has the matrix-vector product of $\mathbf{J}_{21}\mathbf{q}$. This action can be approximated by a finite difference of the residual associated with the Jacobian matrix.

Theorem 5.1 *Jacobian-Free method*

Consider a general nonlinear problem

$$\mathbf{r}(\mathbf{x}) = \mathbf{0} \quad (5.28)$$

whose solution is calculated via Newton's method by

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{r}(\mathbf{x}^{(k)}) \quad (5.29)$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x} \quad (5.30)$$

where $\mathbf{J}$ is the Jacobian matrix. Suppose we wish to evaluate the multiplication of $\mathbf{J}$ on some arbitrary vector $\mathbf{u}$, or $\mathbf{J}\mathbf{u}$. The Jacobian matrix can be approximated by a finite difference such that

$$\mathbf{J}\mathbf{u} = \mathbf{r}(\mathbf{x}^{(k)} + \mathbf{u}) - \mathbf{r}(\mathbf{x}^{(k)}). \quad (5.31)$$

As proof, we take the Taylor series expansion of $\mathbf{r}$ around $\mathbf{x}^{(k)}$ and evaluated at $\mathbf{x}^{(k)} + \mathbf{u}$:

$$(\mathbf{x}^{(k)} + \mathbf{u}) = \mathbf{r}(\mathbf{x}^{(k)}) + \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \left\{ \mathbf{x}^{(k)} + \mathbf{u} - \mathbf{x}^{(k)} \right\} + \dots \quad (5.32)$$

where $\partial \mathbf{r} / \partial \mathbf{x}$, is a matrix of the partial derivative of vector $\mathbf{r}$ with respect to vector $\mathbf{x}$ evaluated at $\mathbf{x} = \mathbf{x}^{(k)}$. Ignoring higher terms, we substitute the Taylor series expansion of $\mathbf{r}(\mathbf{x}^{(k)} + \mathbf{u})$ into the finite difference and cancel terms:

$$\mathbf{r}(\mathbf{x}^{(k)} + \mathbf{u}) - \mathbf{r}(\mathbf{x}^{(k)}) \approx \mathbf{r}(\mathbf{x}^{(k)}) + \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \mathbf{u} - \mathbf{r}(\mathbf{x}^{(k)}) \quad (5.33)$$

$$= \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \mathbf{u} \quad (5.34)$$

$$= \mathbf{J}\mathbf{u}. \quad (5.35)$$

With this approximation, the Jacobian matrix is not even required. The matrix-vector product is approximated by application of the above finite difference formula. However, for this approximation to be valid, $\mathbf{u}$ must be “small” with respect to $\mathbf{x}$.

In the second step, we are solving the modified right hand side of the Schur complement problem, $\mathbf{r}_S$. First, let's look at the term $\mathbf{J}_{21}\mathbf{q}$ in the equation to solve $\mathbf{r}_S$, where there is a dependence on the off-diagonal block $\mathbf{J}_{21}$ multiplied by the vector $\mathbf{q}$. We approximate $\mathbf{J}_{21}$ using the procedure outlined in Theorem 5.1. To see how this Jacobian-Free method is put into practice, we go through the individual steps as applied to Step 2. First we perform a Taylor series expansion of $\mathbf{r}_2$ around $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ with respect to $\mathbf{x}^{(k)}$ and evaluated at

$(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)})$:

$$\mathbf{r}_2(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) = \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}} \left\{ \mathbf{x}^{(k)} + \mathbf{q} - \mathbf{x}^{(k)} \right\} + \dots \quad (5.36)$$

Ignoring higher order terms and rearranging the equation, we get

$$\frac{\partial \mathbf{r}_2}{\partial \mathbf{x}} \mathbf{q} = \mathbf{r}_2(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) - \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \quad (5.37)$$

where $\partial \mathbf{r}_2 / \partial \mathbf{x} \equiv \mathbf{J}_{21}$ in our short-hand notation. Substituting this approximation into the definition of $\mathbf{r}_S$, we get

$$\begin{aligned} \mathbf{r}_S &= \mathbf{J}_{22}^{-1} \{ \mathbf{r}_2 + \mathbf{J}_{21} \mathbf{q} \} \\ &\approx \mathbf{J}_{22}^{-1} \left\{ \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \mathbf{r}_2(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) - \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\} \\ &= \mathbf{J}_{22}^{-1} \mathbf{r}_2(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}). \end{aligned} \quad (5.38)$$

Hence, the equation for $\mathbf{r}_S$ only depends on $\mathbf{r}_2$ and its Jacobian $\mathbf{J}_{22}$. Vector $\mathbf{q}$ is “small” with respect to $\mathbf{x}$ as it is the update vector from one Newton step for solving $\mathbf{r}(\mathbf{x}, \tilde{\mathbf{y}}) = 0$. Another way to look at this step is to view vector $\mathbf{r}_S$ as the update vector in a single Newton step to solve the residual equation $-\mathbf{r}_2(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}) = \mathbf{0}$ for $\mathbf{y}$. Note, though, that the Jacobian matrix $\mathbf{J}_{22}$ is evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$, whereas the residual $\mathbf{r}_2$ is evaluated at $(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)})$. We assume that $\mathbf{J}_{22}(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) \approx \mathbf{J}_{22}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ such that instead of taking a full Newton step, we are taking a modified Newton step. Returning to our definition laid out in Equation (5.16), we can rewrite $\mathbf{r}_S$ in terms of the arbitrary solver $\mathcal{G}(\mathbf{x}, \mathbf{y})$. The vector $\mathbf{r}_S$ rewritten in terms of $\mathcal{G}(\mathbf{x}, \mathbf{y})$ is:

$$\mathbf{r} = \mathbf{y}^{(k)} - \mathcal{G}(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}). \quad (5.39)$$

Again, only one iteration of $\mathcal{G}$ is required.

In the third step, we need to compute the Schur complement matrix $\mathbf{S}$ to find our update vector $\Delta \mathbf{y}$. Note that the Schur complement matrix is not needed explicitly, but simply its action on a vector. Therefore, an iterative solution technique, such as a Krylov subspace method, can be used to solve the linear equation $\mathbf{S} \Delta \mathbf{y} = -\mathbf{r}_S$ for $\Delta \mathbf{y}$. A suitable candidate solution is constructed from the Krylov space defined by $K \equiv \{\mathbf{w}, \mathbf{S}\mathbf{w}, \mathbf{S}^2\mathbf{w}, \dots, \mathbf{S}^m\mathbf{w}\}$ where $\mathbf{w} = -\mathbf{r}_S$. Furthermore, the Schur complement, $\mathbf{S}$, depends on the off-diagonal block Jacobians $\mathbf{J}_{12}$ and $\mathbf{J}_{21}$ which can be approximated using finite differences as in

Theorem 5.1. This method of using a Krylov subspace solver, such as GMRES, along with approximating the Jacobian matrix with finite differencing is known as the Jacobian-free, Newton-Krylov (JFNK) method [91].

Theorem 5.2 *Jacobian-Free Newton Krylov method*

Consider a general nonlinear problem:

$$\mathbf{r}(\mathbf{x}) = \mathbf{0} \quad (5.40)$$

whose solution via Newton's method is calculated by

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{r}(\mathbf{x}^{(k)}) \quad (5.41)$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}. \quad (5.42)$$

Krylov subspace methods are applied to solve the linear algebraic system of equations of a single Newton step. A suitable candidate solution is constructed from the Krylov subspace defined by $K \equiv \{\mathbf{w}, \mathbf{J}\mathbf{w}, \mathbf{J}^2\mathbf{w}, \dots, \mathbf{J}^m\mathbf{w}\}$ where, typically, $w = -\mathbf{r}(\mathbf{x}^{(k)})$. In this manner, the Jacobian matrix is not factorized. Only matrix-vector products, $\mathbf{J}\mathbf{w}$ are required, and we can apply Theorem 5.1 to approximate the Jacobian matrix using a finite difference approximation to give:

$$\mathbf{J}\mathbf{w} = \mathbf{r}(\mathbf{x}^{(k)} + \mathbf{w}) - \mathbf{r}(\mathbf{x}^{(k)}). \quad (5.43)$$

However, the Krylov vector could be “large” in comparison to $\mathbf{x}$. Therefore, the Krylov vector must be scaled appropriately for the finite difference to adequately approximate the Jacobian matrix. This is done by multiplying the Krylov vector w by a small, scalar, perturbation parameter $\epsilon \ll 1$ to get:

$$\mathbf{J}\mathbf{w} = \frac{1}{\epsilon}\mathbf{J}\epsilon\mathbf{w} = \frac{1}{\epsilon} \left\{ \mathbf{r}(\mathbf{x}^{(k)} + \epsilon\mathbf{w}) - \mathbf{r}(\mathbf{x}^{(k)}) \right\}. \quad (5.44)$$

With this approximation, the Jacobian matrix is not even required and the Krylov subspace is approximated by application of the above finite difference formula.

To see how the JFNK method is used to solve the Schur complement problem, we examine the matrix-vector product of the Schur complement matrix $\mathbf{S}$ multiplied by the Krylov vector $\mathbf{w}$:

$$\mathbf{S}\mathbf{w} = \frac{1}{\epsilon}\mathbf{S}\epsilon\mathbf{w} = \frac{1}{\epsilon} \left\{ \epsilon\mathbf{w} - \mathbf{J}_{22}^{-1}\mathbf{J}_{21}\mathbf{J}_{11}^{-1}\mathbf{J}_{12}\epsilon\mathbf{w} \right\} \quad (5.45)$$

where the Krylov vector $\mathbf{w}$ is scaled by the scalar perturbation factor $\epsilon \ll 1$. Evaluating

this matrix-vector product is broken into two parts. Step 3(a) computes $\mathbf{C} \equiv \mathbf{J}_{11}^{-1} \mathbf{J}_{12}$, which is then substituted into the definition of the Schur complement in Step 3(b).

We see in Equation (5.45) block Jacobian matrix $\mathbf{J}_{12}$ is multiplied by the perturbed Krylov vector $\epsilon \mathbf{w}$. Therefore, we evaluate the matrix-vector product $\mathbf{C} \epsilon \mathbf{w}$ instead of computing the standalone matrix $\mathbf{C}$:

$$\mathbf{C} \epsilon \mathbf{w} = \mathbf{J}_{11}^{-1} \mathbf{J}_{12} \epsilon \mathbf{w}. \quad (5.46)$$

Theorem 5.1 is used to approximate the matrix-vector product $\mathbf{J}_{12} \epsilon \mathbf{w}$. The Taylor series expansion of $\mathbf{r}_1$ is taken around $\mathbf{y}^{(k)}$ and evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w})$ to give

$$\mathbf{J}_{12} \epsilon \mathbf{w} = \frac{\partial \mathbf{r}_1}{\partial \mathbf{y}} \epsilon \mathbf{w} \approx \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) - \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}). \quad (5.47)$$

Substituting this approximation into Equation (5.46) we get

$$\begin{aligned} \mathbf{C} \epsilon \mathbf{w} &\approx \mathbf{J}_{11}^{-1} \left\{ \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) - \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\} \\ &= \mathbf{J}_{11}^{-1} \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) - \mathbf{J}_{11}^{-1} \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ &= \mathbf{J}_{11}^{-1} \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) + \mathbf{q} \end{aligned} \quad (5.48)$$

and we reuse vector $\mathbf{q} = \mathbf{J}_{11}^{-1} \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ from Step 1. The first term of $\mathbf{C}$, $\mathbf{J}_{11}^{-1} \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w})$ is determined as in Step 1 except the residual vector $\mathbf{r}_1$ is evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w})$ with one Newton step of solver 1. However, the Jacobian matrix $\mathbf{J}_{11}$ is evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ instead of $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w})$. We approximate the Jacobian matrix as $\mathbf{J}_{11}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) \approx \mathbf{J}_{11}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ taking a modified Newton step instead of a full Newton step. Equation 5.48 rewritten in terms of generic solver $\mathcal{F}$ becomes

$$\mathbf{C} \epsilon \mathbf{w} = \mathbf{x}^{(k)} - \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) + \mathbf{q} \quad (5.49)$$

where solver $\mathcal{F}$ performs one internal iteration at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w})$.

In Step 3(b), the matrix-vector product $\mathbf{C} \epsilon \mathbf{w}$ is substituted into Equation (5.45) to give:

$$\frac{1}{\epsilon} \mathbf{S} \epsilon \mathbf{w} = \frac{1}{\epsilon} \left\{ \epsilon \mathbf{w} - \mathbf{J}_{22}^{-1} \mathbf{J}_{21} \mathbf{C} \epsilon \mathbf{w} \right\}. \quad (5.50)$$

Here we have the matrix-vector product $\mathbf{C} \epsilon \mathbf{w}$, which is itself a vector, that is multiplied with block Jacobian matrix $\mathbf{J}_{21}$. We again approximate this product using finite differencing by

taking a Taylor series expansion of $\mathbf{r}_2$ about $\mathbf{x}^{(k)}$ and evaluated at $(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)})$

$$\mathbf{J}_{21}\mathbf{C}\epsilon\mathbf{w} = \frac{\partial \mathbf{r}_2}{\partial \mathbf{x}}\mathbf{C}\epsilon\mathbf{w} \approx -\left\{\mathbf{r}_2(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)}) - \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\right\}. \quad (5.51)$$

Equation (5.51) multiplied by the inverse of block Jacobian matrix $\mathbf{J}_{22}$ is then

$$\begin{aligned} \mathbf{J}_{22}^{-1}\mathbf{J}_{21}\mathbf{C}\epsilon\mathbf{w} &\approx -\mathbf{J}_{22}^{-1}\left\{\mathbf{r}_2(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)}) - \mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\right\} \\ &= -\mathbf{J}_{22}^{-1}\mathbf{r}_2(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)}) + \mathbf{J}_{22}^{-1}\mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{aligned} \quad (5.52)$$

which can be viewed as two separate Newton solves of problem 2, the first evaluated at $(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)})$ and the second at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. However, the Jacobian matrix J_{22} is evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ such that the first term is actually a modified Newton solve while the second term is a full Newton solve. Only one Newton iteration at either vector location is necessary. Equation 5.45 is then approximated as

$$\mathbf{S}\mathbf{w} = \frac{1}{\epsilon}\mathbf{S}\epsilon\mathbf{w} \approx \mathbf{w} + \frac{1}{\epsilon}\left\{\mathbf{J}_{22}^{-1}\mathbf{r}_2(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)}) - \mathbf{J}_{22}^{-1}\mathbf{r}_2(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\right\}. \quad (5.53)$$

Equation (5.53) can be written in terms of the arbitrary solver $\mathcal{G}$ using the relation given in Equation (5.16) such that

$$\mathbf{S}\mathbf{w} \approx \mathbf{w} - \frac{1}{\epsilon}\left\{\mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\right\} \quad (5.54)$$

where only one internal iteration of $\mathcal{G}$ is needed in each instance. The entirety of Step 3 is repeated until enough Krylov vectors are attained for a satisfactory approximation of update vector $\Delta\mathbf{y}$ to $\mathbf{S}\Delta\mathbf{y} = -\mathbf{r}_S$.

In the final step, Step 4, for update vector $\Delta\mathbf{x} = \mathbf{q} - \mathbf{C}\Delta\mathbf{y}$, we need the action of $\mathbf{C}$ on vector $\Delta\mathbf{y}$. Calculating $\mathbf{C}\Delta\mathbf{y}$ is done in the same manner as calculating $\mathbf{C}\epsilon\mathbf{w}$ in Step 3(a). We again approximate the action of the off-diagonal block Jacobian $\mathbf{J}_{21}$ on $\Delta\mathbf{y}$ with finite differencing:

$$\mathbf{J}_{12}(\Delta\mathbf{y}) = \frac{\delta \mathbf{r}_1}{\delta \mathbf{y}}(\Delta\mathbf{y}) \approx \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) - \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}). \quad (5.55)$$

The equation for $\Delta \mathbf{x}$ then becomes

$$\begin{aligned}\Delta \mathbf{x} &= \mathbf{q} - \mathbf{C}\Delta \mathbf{y} \\ &\approx -\mathbf{J}_{11}^{-1} \left\{ \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y}) - \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\} \\ &= -\mathbf{J}_{11}^{-1} \left\{ \mathbf{r}_1(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y}) \right\}\end{aligned}\tag{5.56}$$

or one modified Newton step of problem 1 evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y})$, where $\mathbf{J}_{11}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y}) \approx \mathbf{J}_{11}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$. Rewritten in terms of generic solver $\mathcal{F}$, we get

$$\Delta \mathbf{x} = \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y}) - \mathbf{x}^{(k)}.\tag{5.57}$$

where $\mathcal{F}$ need only perform one internal iteration.

Steps 1-4 constitute one block Newton step of the coupled problem. The ABN method is repeated until each subproblem is deemed converged by some internal criterion as with the BGS method, namely $\|\mathbf{r}_1(\mathbf{x}^{(k+1)}, \tilde{\mathbf{y}}^{(k+1)})\| < \epsilon_1$ and $\|\mathbf{r}_2(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k+1)})\| < \epsilon_2$.

Each subsolver is called multiple times to solve the global, block Newton step: $n + 2$ times where n is the number of iterations needed to solve the Schur complement problem. Subsolver 1, $\mathcal{F}$, is called once in Step 1, n times in Step 3(a), and once in Step 4 for a total of $n + 2$ calls. Subsolver 2, $\mathcal{G}$, is called once in Step 2 and $n + 1$ in Step 3(b) for a total of $n + 2$ calls. In addition, the subsolvers can be iterated internally as many times as deemed necessary by the user. GMRES or other iterative solution methods is used to approximate the solution of the Schur complement problem. The derivation presented here assumes that each subsolver is based on Newton's method also. However, each step can be reduced in terms of black-box solvers, $\mathcal{F}(\mathbf{x}, \mathbf{y})$ and $\mathcal{G}(\mathbf{x}, \mathbf{y})$, the use any arbitrary, fixed-point solution method. If the subsolvers are based on methods other than Newton's method, overall quadratic convergence could be compromised since the different steps of the ABN method are based on both solvers using Newton's method. Different methods would only approximate a Newton step of a subproblem.

As mentioned above, the derivation presented here differs in its basic construction from that presented in References [4, 115, 116]. In these previous works, arbitrary, fixed-point solvers are introduced at the beginning of the derivation instead of at the end. Working through each step, however, produces the same algorithm, although subtle differences do remain in the calculation of the Schur complement. Working through this step as given in

References [4, 115, 116], the Schur complement is approximated by

$$\frac{1}{\epsilon} \mathbf{S}(\epsilon \mathbf{w}) = w + \frac{1}{\epsilon} \left\{ \mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}(\epsilon \mathbf{w}), \mathbf{y}^{(k)} + \epsilon \mathbf{w}) \right\}. \quad (5.58)$$

The approximation includes an extra perturbation added to the $\mathbf{y}$ unknowns in Equation (5.58), term $\mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}(\epsilon \mathbf{w}), \mathbf{y}^{(k)})$ that is absent in Equation (5.50). If the equations in solver $\mathcal{G}(\mathbf{x}, \mathbf{y})$ are linear, and assuming $\mathcal{G}$ is based on Newton's method, one Newton step solves the equations exactly regardless of the initial guess of y , i.e. $\mathbf{g}(\mathbf{x}, \mathbf{y} + \epsilon \mathbf{w}) = \mathbf{g}(\mathbf{x}, \mathbf{y})$, where $\mathbf{g}$ is defined in Equation (5.16). However, if the equations are nonlinear, a single Newton iteration of a nonlinear problem can depend on $\mathbf{y}$ making the two derivations different.

A work around is to iterate solver $\mathcal{G}(\mathbf{x}, \mathbf{y})$ until subproblem 2 is deemed converged, i.e., perform multiple internal Newton steps instead of just once. Each subproblem can be viewed as a set of nonlinear boundary value problems, whose solutions depend on the boundary values. By the way we choose to partition the two problems, information between the two problems are exchanged only along a shared boundary. Subproblem 2, $\mathbf{y}$ unknowns, only sees the variables of subproblem 2 that are on the shared boundary, $\mathbf{x}$ along the boundary, and vice versa. In a sense, the values of $\mathbf{x}$ on the boundary set the solution of subproblem 2 because everything internal to subproblem 2 is fixed. Therefore, if $\mathbf{x}$ does not change the ultimate converged solution of subproblem 2 does not change even if $\mathbf{y}$ has been perturbed. Then upon convergence of subproblem 2 $\mathcal{G}(\mathbf{x}, \mathbf{y}) \equiv \mathcal{G}(\mathbf{x})$, thus eliminating the difference in the two derivations but at a cost of algorithm efficiency. What role this additional perturbation plays on convergence of the algorithm is not addressed in this paper, but will be investigated in future work.

We conclude our description of the ABN method with a caveat on how we treat the unknowns $\mathbf{x}$ and $\mathbf{y}$. We have chosen to couple our two problems along a shared boundary that is one-dimension less than the spatial dimension of the overall problem, i.e. a point in a one-dimensional problem, and a line in a two-dimensional problem. In this manner, the solvers $\mathcal{F}$ and $\mathcal{G}$ do not explicitly depend on the other code's internal unknowns. Subproblem 1 uses its solution $\mathbf{x}$ to derive a set of boundary data $\mathcal{B}(\mathbf{x})$, that it hands to subproblem 2. Similarly, subproblem 2 hands back a set of boundary data $\mathcal{A}(\mathbf{y})$. Whenever solution data is passed between the two solvers, the solutions $\mathbf{x}$ and $\mathbf{y}$ are parameterized as sets of boundary data, $\mathcal{A}(\mathbf{y})$ and $\mathcal{B}(\mathbf{x})$. The action of the solvers is described as $\mathcal{F}(\mathbf{x}, \mathcal{A}(\mathbf{y}))$ and $\mathcal{G}(\mathcal{B}(\mathbf{x}), \mathbf{y})$. Upon implementation, the boundary data $\mathcal{B}(\mathbf{x})$ and $\mathcal{A}(\mathbf{y})$ must be updated each time $\mathbf{x}$ and $\mathbf{y}$ are updated to keep the internal representation of the solution and its

boundary parameterization consistent with one another.

5.4 Example Problems

To evaluate the robustness of convergence of the ABN iterative method compared to the BGS method, we consider a one-dimensional, steady-state, heat transfer problem that is representative of melt crystal growth. This one-dimensional problem was used previously to understand the convergence characteristics of the BGS method in [135, 196], and with under-relaxation in [44]. Additionally, a simple, two-dimensional, steady-state heat transfer problem is used to investigate the rate of convergence of the ABN method for a more complicated model. Finally, the initial solution to a complex, two-dimensional model of an electrodynamic gradient freeze furnace solved using the ABN method, similar to that in [136], is presented.

5.4.1 One-Dimensional Model

In the one-dimensional case studies, we define a steady-state heat transfer problem over a domain that spans from $x = 0$ to $x = 2$, where x is a non-dimensional length, as shown in Figure 5.1(a). This domain is further divided into two subdomains: Ω_1 ($0 \leq x \leq 1$) which is a one-dimensional analog of a crystal melt, and Ω_2 ($1 \leq x \leq 2$) the analog of the crystal growth furnace.

Problem Formulation

The non-dimensional, governing equations of the problem are heat conduction and convection in Ω_1 and heat conduction only in Ω_2 :

$$\frac{d^2 T}{dx^2} - c \frac{dT}{dx} = 0, \quad \text{in } \Omega_1 \quad (5.59)$$

$$\kappa \frac{d^2 T}{dx^2} = 0, \quad \text{in } \Omega_2 \quad (5.60)$$

where T is the non-dimensional temperature and x is the non-dimensional length defined by dividing the dimensional variables by characteristic scales $\tilde{T}$ and $\tilde{x}$ respectively. The parameter $\kappa \equiv k_1/k_2$ is the ratio of thermal conductivities between the two domains, and $c \equiv \rho C_p \tilde{v} \tilde{x}/k_1$ is the Peclet number of the system, defined by the characteristic melt velocity $\tilde{v}$, the melt density ρ , and heat capacity of the melt C_p in the subdomain Ω_1 .

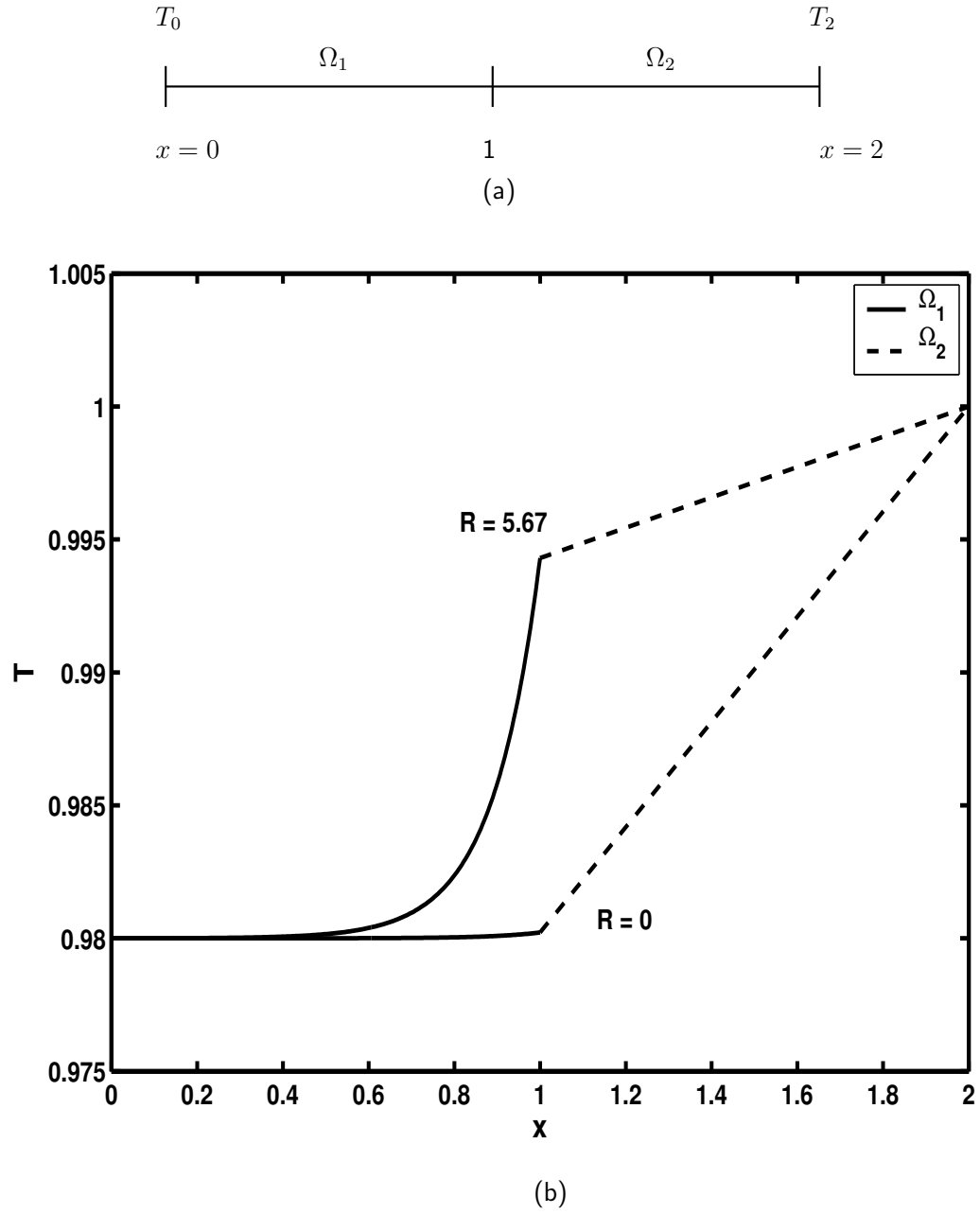


Figure 5.1: (a) Schematic representation of the one-dimensional model problem. (b) Steady-state temperature profiles of the one-dimensional model.

Heat transfer in a crystal growth furnace is often dominated by radiation effects due to the high operating temperatures of the system. Radiation effects are included in the one-dimensional problem by treating the subdomain Ω_2 as transparent to radiation, with the boundary at $x = 1$ exchanging radiation with the boundary at $x = 2$. At the shared boundary coordinate of $x = 1$, we demand continuity of the temperature and heat flux, the later of which accounts for both conductive and radiative terms. To simplify the notation we will use $T_L \equiv T_{x=1-}$ and $T_R \equiv T_{x=1+}$ where L and R mean left and right, respectively. The continuity equations are then:

$$T_L = T_R \quad (5.61)$$

$$q_L = q_R \quad (5.62)$$

where q are the non-dimensional heat fluxes given by

$$q_L = - \left. \frac{dT}{dx} \right|_{x=1-} \quad (5.63)$$

$$q_R = -\kappa \left. \frac{dT}{dx} \right|_{x=1+} + R(T_R^4 - T_2^4) \quad (5.64)$$

where $R \equiv \sigma \tilde{T}^3 \tilde{x} / k_1$ is the non-dimensional radiation number with σ as the Stefan Boltzmann constant, and T_2 is the temperature at $x = 2$. Note we assume that both external surfaces at $x = 1$ and $x = 2$ are black such that their emissivities are unity. Dirichlet boundary conditions are specified on the external ends of the system, $x = 0$ and $x = 2$:

$$T = T_0, \quad \text{at } x = 0 \quad (5.65)$$

$$T = T_2, \quad \text{at } x = 2. \quad (5.66)$$

The above problem can be solved analytically [135, 196] to give

$$T = \frac{(T_1 - T_0 e^c) + (T_0 - T_1) e^{cx}}{1 - e^c}, \quad \text{in } \Omega_1 \quad (5.67)$$

$$T = (T_2 - T_1)(x - 2) + T_2, \quad \text{in } \Omega_2 \quad (5.68)$$

where T_1 is the temperature at $x = 1$ given by

$$R(T_1^4 - T_2^4) + \frac{(T_0 - T_1) c e^c}{1 - e^c} - \kappa(T_2 - T_1) = 0. \quad (5.69)$$

The goal is to determine the value of T_1 which satisfies the continuity conditions, Equations (5.61) and (5.62). The value of T_1 then sets the temperature in both the melt and furnace domains.

For this example problem, we use a set of parameter values that are representative of a crystal growth system: $c = 9$, $\kappa = 0.1$, $T_0 = 0.98$, and $T_2 = 1.0$. Two cases are considered regarding radiation effects. The first case sets $R = 0$, thereby eliminating radiation heat transfer and making the problem linear. The second case sets $R = 5.67$, a representative value for radiation effects, making radiation a significant part of the computation. Steady-state temperature profiles through both domains are plotted in Figure 5.1(b) for both the linear and nonlinear cases. The relatively large value of the Peclet number, c , results in a flattened thermal profile through the melt domain Ω_1 . When $R = 0$, this profile connects to a linear temperature profile through the furnace domain Ω_2 . Addition of radiation heat flux at $x = 1$ by setting $R = 5.67$ considerably alters the solution from the linear problem without radiation.

Coupling Equations

As stated above, the temperature and heat flux must match at the coupling point $x = 1$ to have a converged, self-consistent solution to the governing equations. This matching condition, however, does not specify how information is to be passed between the two models via boundary conditions at $x = 1$. In prior work [135, 196], we developed a set of mixed boundary conditions to be used by each subproblem during local iterations:

$$q_L = q_R + \frac{1 - \alpha}{\alpha}(T_L - T_R) \quad \text{at } x = 1^- \quad (5.70)$$

$$q_R = q_L + \frac{1 - \beta}{\beta}(T_R - T_L) \quad \text{at } x = 1^+ \quad (5.71)$$

where α and β are chosen coupling parameters and vary from 0 to 1. Equation (5.70) is applied as a boundary condition when solving Equation (5.59) for the melt domain Ω_1 , and Equation (5.71) is applied as a boundary condition when solving Equation (5.60) for the furnace domain Ω_2 . By applying these conditions, the temperature and flux continuity requirements are satisfied for any values of $\alpha \neq \beta$. When either $\alpha = 0$ or $\beta = 0$, the corresponding boundary condition reduces to Dirichlet condition of temperature continuity. Similarly, when $\alpha = 1$ or $\beta = 1$, we have Neumann conditions specify flux continuity. Other values of α and β produce mixed, or Robin, boundary conditions. When $\alpha = \beta$, the two independent boundary conditions, Equations (5.70) and (5.71), become identical, making

the resulting global problem for the temperature field under-specified.

Since we treat each subproblem as a black box, in essence our residual equations are Equations (5.70) and (5.71). The Equations (5.59) and (5.60) are within the black boxes, and are used to calculate the temperatures and fluxes at the shared boundary $x = 1$. The coupling equations may appear to be under-specified with two equations and four unknowns (two temperatures and two fluxes), but the heat fluxes are quantities derived from the temperature field of their respective domains. From the analytic solutions of the temperature in the two subdomains, Equations (5.67) and (5.68), along with the definitions of heat flux, Equations (5.63) and (5.64), the coupling equations can be rewritten as:

$$T_L = a + bT_L + cT_R - RT_R^4 \quad (5.72)$$

$$T_R = a + dT_L + eT_R - RT_R^4 \quad (5.73)$$

where

$$\begin{aligned} a &= fT_0 + \kappa T_2 + RT_2^4, & b &= -\frac{1-\alpha}{\alpha} - f, & c &= \frac{1-\alpha}{\alpha} - \kappa, \\ d &= -\frac{1-\beta}{\beta} - f, & e &= \frac{1-\beta}{\beta} - \kappa, & f &= -\frac{ce^c}{1-e^c}, \end{aligned}$$

and $R = 0$ in the linear case.

The two unknowns of the coupled problem are $\mathbf{x} = T_{x=1-}$ and $\mathbf{y} = T_{x=1+}$. Equations (5.72) and (5.73) are our residual equations analogous to Equations (5.1) and (5.2) as outlined in the previous section:

$$r_1 = T_L - a - bT_L - cT_R + RT_R^4 \quad (5.74)$$

$$r_2 = T_R - a - dT_L - eT_R + RT_R^4. \quad (5.75)$$

The Jacobian matrix, $\mathbf{J}$ of this problem is a (2×2) matrix and the submatrices are (1×1) matrices:

$$\begin{aligned} J_{11} &= \frac{\partial r_1}{\partial T_L} = 1 - b, & J_{12} &= \frac{\partial r_1}{\partial T_R} = -c + 4RT_R^3 \\ J_{21} &= \frac{\partial r_2}{\partial T_L} = -d, & J_{22} &= \frac{\partial r_2}{\partial T_R} = 1 - e + 4RT_R^3 \end{aligned}$$

The solvers $\mathcal{F} = J_{11}^{-1}r_1$ and $\mathcal{G} = J_{22}^{-1}r_2$ in Equations (5.3) and (5.4) use Newton's method

to solve of the residual equations for T_1 . At convergence $T_L = T_R$ and $q_L = q_R$.

Results

The convergence behaviors of the BGS method and the ABN method when solving the above problem are compared. When solving with either method, we begin with a suitable initial guess for the interface temperature, $T_1^{(0)}$. In iteration k of the BGS method first $\mathcal{F}$ solves for T_L with $T_R^{(k)}$ using Newton's method. Then $\mathcal{G}$ solves for $T_1^{(k+1)} = T_R$ using the value of T_L computed by $\mathcal{F}$. Both solvers were iterated until internal convergence was reached per each BGS step. Solving the problem using the ABN method is more complicated than the BGS method. The ABN algorithm is solved using the residual and Jacobian equations outlined above. Only one internal Newton step per solver call was taken.

Convergence behavior of the two methods are compared in Figures 5.2(a) and 5.2(b), respectively, for different (α, β) pairs and for $R = 0$ and $R = 5.67$. The dark spots on these plots indicate (α, β) pairs that are non-convergent after 1,000 BGS or ABN iterations. As stated in [44, 196], the dark areas correspond to conditions where the iteration sequence will never converge, regardless of the value of the initial guess. For a detailed description on how the block Gauss-Seidel results were computed, see [196].

Convergence of the block Gauss-Seidel method is highly dependent on both the chosen (α, β) pair and the nonlinearity of the problem indicated by the value of R , the radiation number. As nonlinearity is increased from $R = 0$ to $R = 5.67$, the convergence map changes significantly. Virtually all of the of the successful choices of α and β for the linear $R = 0$ case fail for the nonlinear $R = 5.67$ case. Also, the area of the unshaded region changes very little between the linear and nonlinear cases, indicating that about half of all (α, β) pairs would be successful.

The ABN method, on the other hand, does successfully converge for nearly all chosen (α, β) pairs, as shown by the plots in Figure 5.2(b). For the linear $R = 0$ case, the only instances when the ABN fails is when $\alpha = \beta$ which causes a singular situation of one governing equation. These pairings are always non-convergent no matter what modular coupling algorithm is used. The nonlinear $R = 5.67$ case has additional non-convergent pairings, but is still vastly superior to the results of the corresponding block Gauss-Seidel case. Overall, the robustness of the ABN method far exceeds that of the block Gauss-Seidel method.

Figure 5.3 shows additional convergence maps of the ABN method for various values

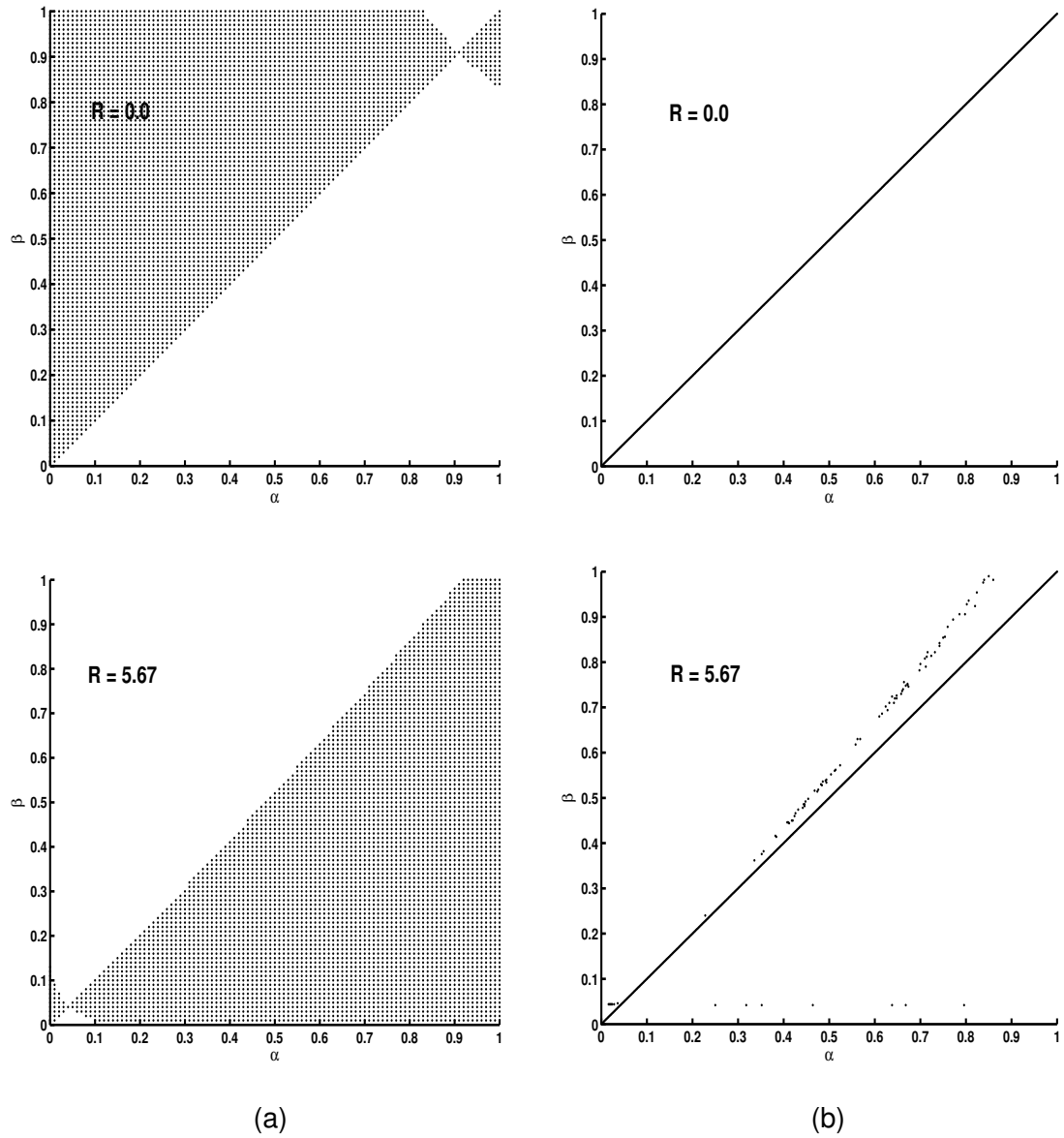


Figure 5.2: Comparison between convergence maps of the one-dimensional model as a function of α and β of (a) block Gauss-Seidel and (b) the Approximate Block Newton methods for $R = 0$ and $R = 5.67$.

of R . As the nonlinearity of the problem increases with the value of R , the number of non-convergent (α, β) pairs does not increase correspondingly. For the case of $R = 0.3$ over 98% of the pairings are convergent and for $R > 0.3$ over 99% are convergent. The additional non-convergent pairings in the nonlinear cases compared to the linear case are due to the finite differencing step used in the Jacobian-free Newton-Krylov approximation of the off-diagonal block Jacobians.

The performance of the two algorithms is compared in Figure 5.4, where (a) percentage of converged cases and (b) average number of iterations are plotted versus the radiation number R . As R increases, over 98% of the (α, β) pairs converge with the ABN method, while just under 50% converge with the block Gauss-Seidel method. Furthermore, the average number of iterations for the method to reach convergence stays relatively constant with increasing R for the ABN method, while the block Gauss-Seidel method needs to perform more and more iterations as the nonlinearity of the problem increases. This iteration count, however, is of the outer block iterations. In each method, the subsolvers are called two or more times during each block iteration. The block Gauss-Seidel method calls each solver once during each block iteration, and solves each subproblem until convergence. The ABN method calls each solver multiple times dependent on iterative method used to solve the Schur complement problem; in this example we performed only one iteration of each solver.

We also compare the convergence rates of the ABN method with an exact Newton solver of Equations (5.72) and (5.73) for two different (α, β) pairs and values of ϵ . The two (α, β) pairs are designated as “normal” where $(\alpha, \beta) = (1, 0)$, and “nearly non-singular” where $(\alpha, \beta) = (0.3, 0.32)$, as shown in Figures 5.5(a) and 5.5(b) respectively. In these figures, the residual norm of the coupled problem is plotted against iterations number for $R = 5.67$, where $norm = \sqrt{R_1^2 + R_2^2}$. For the “normal” case, Figure 5.5(a), convergence is super-linear to quadratic depending on the value of ϵ used in the finite difference step to approximate the off-diagonal blocks. At $\epsilon = 10^{-8}$, the convergence of the “normal” case is as good as the exact Newton’s method. For the “nearly singular” case, the ABN method converges much slower than the exact Newton’s method, but still ultimately works. Note that the BGS algorithm would not converge if $(\alpha, \beta) = (0.3, 0.32)$.

5.4.2 Two-Dimensional Model

We now consider a more realistic two-dimensional representation of a melt-crystal growth system. The domain is again split into two subdomains: a “local” domain and a “global”

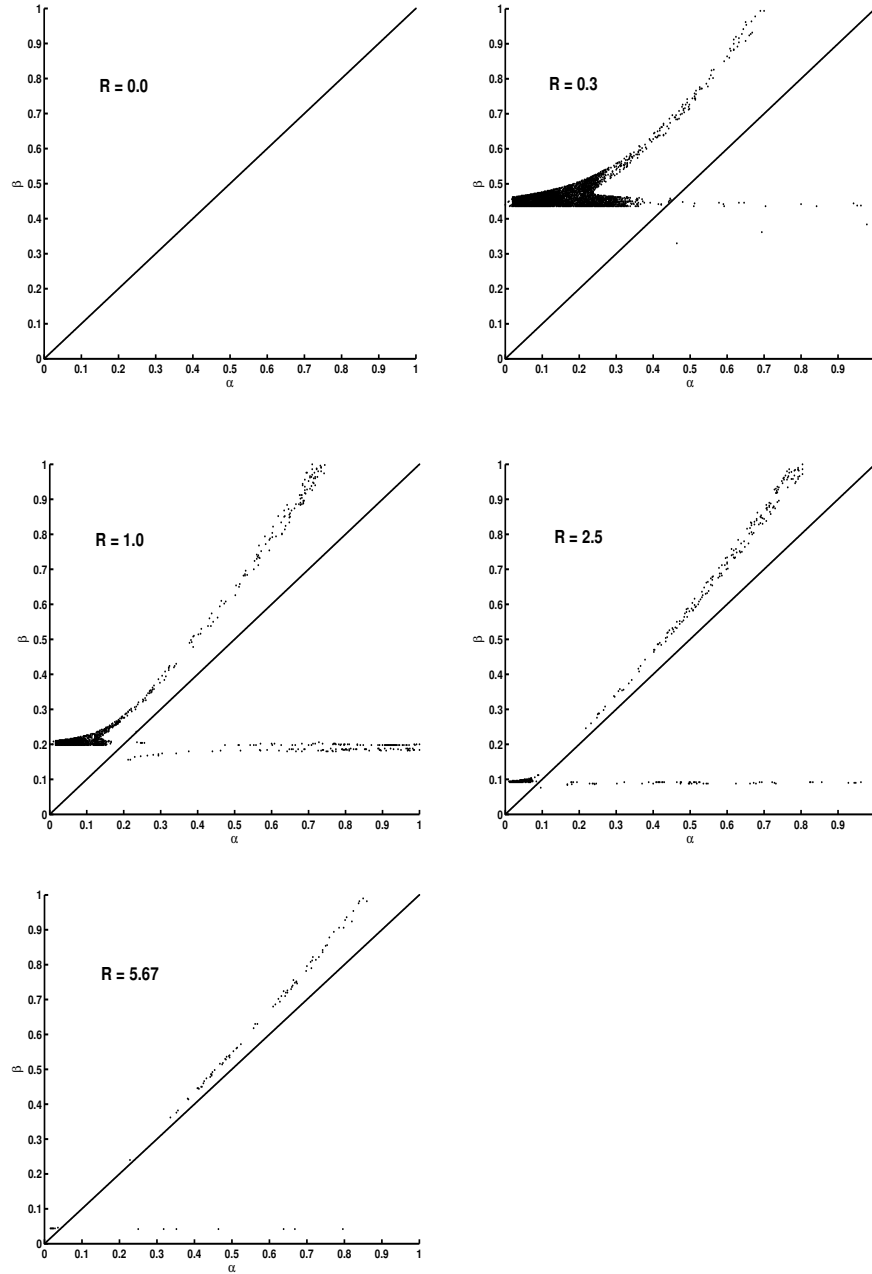
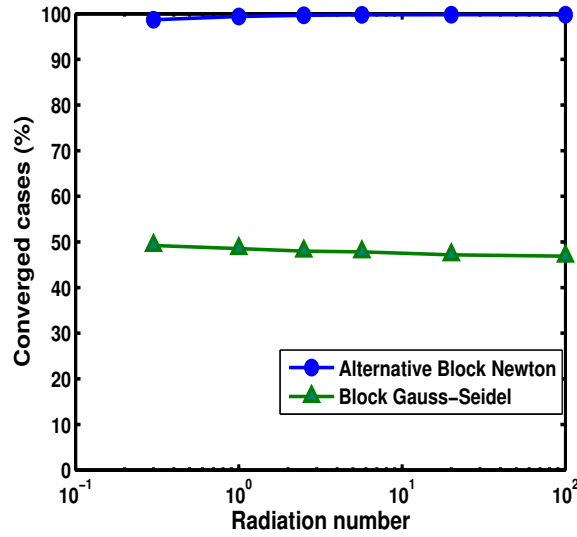
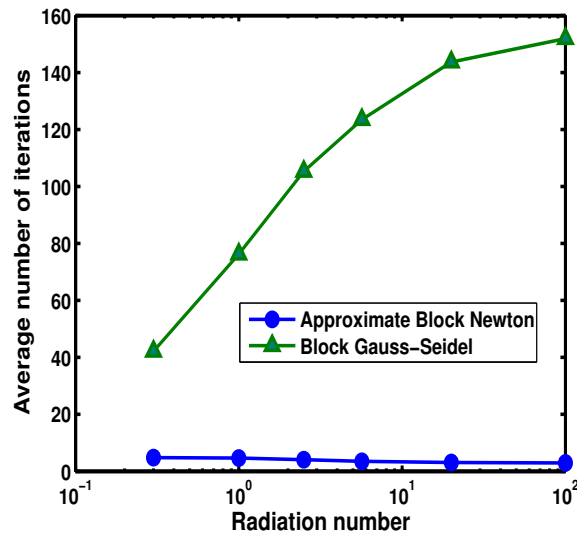


Figure 5.3: Convergence maps of the one-dimensional model using the approximate Block Newton method for different values of radiation number R .

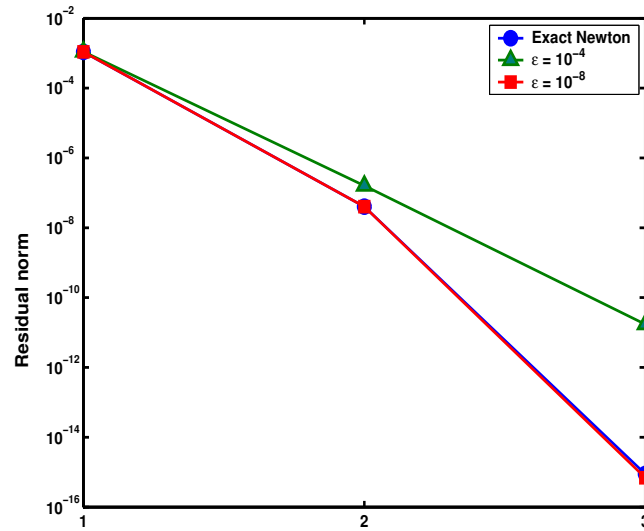


(a)

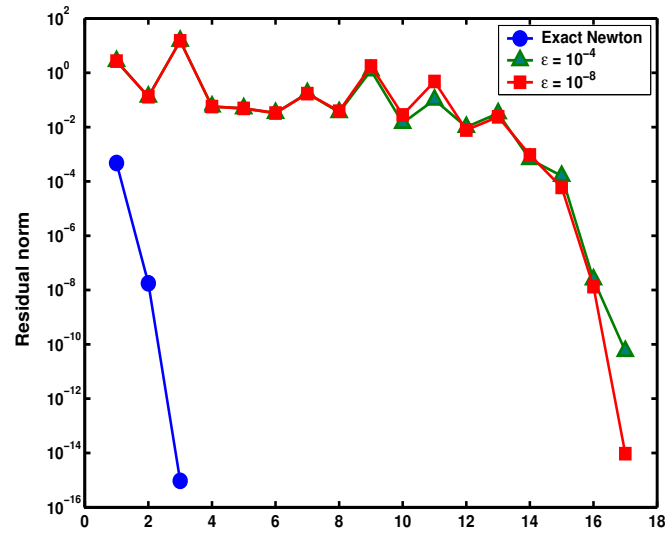


(b)

Figure 5.4: Comparison of the one-dimensional model between the approximate Block Newton and block Gauss-Seidel methods for (a) percentage of converged cases and (b) average number of iterations as a function of radiation number R .



(a)



(b)

Figure 5.5: Comparison of the residual norm versus iteration number one-dimensional model between the exact Newton and the Approximate Block Newton methods for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-8}$ for (a) $\alpha, \beta = (1, 0)$ and the (b) $\alpha, \beta = (0.3, 0.32)$ at $R = 5.67$.

model as seen in Figure 5.6. The “local” domain is analogous to the crystal melt in a growth system, where we model heat conduction, convection, and buoyancy driven flow. The “global” model is analogous to the crystal growth furnace, where we model heat conduction only. The domain is cylindrical in shape, which we assume is axisymmetric to reduce the dimensional order from three to two. This overall model is a gross simplification of a crystal growth furnace, but it includes the key features and physical phenomena of such systems. With a simple two-dimensional example, we are able to test the ABN method, specifically its rate of convergence, in a time efficient manner.

Two separate computer codes are employed to model the subdomains: Cats2D for the local model, and CrysVUn for the global model. Cats2D [195], designated as solver 1, uses the finite element method solved via a modified-Newton scheme and a frontal solver. CrysVUn [8, 97], solver 2, is a commercial code based on the finite volume method and solved via quasi-Newton iteration schemes. We again look at a linear and nonlinear case of the two-dimensional problem. In this example, the nonlinearity arises from the Navier-Stokes equation for fluid flow coupled the energy balance equation within the local model.

Local Model

As mentioned above, the local model, submodel 1, represents the molten phase of a crystal grown via the melt-crystal method. In this submodel, we consider heat conduction, convection, and buoyancy driven flow at steady-state. The governing equations of the local model are:

$$\nabla^2 T - \text{Pe}_T \mathbf{u} \cdot \nabla T = 0 \quad (5.76)$$

$$\text{Re} \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \mathbf{T} - \text{Ra}(1 - T)\mathbf{g} = 0 \quad (5.77)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (5.78)$$

where

$$\mathbf{T} \equiv -p\mathbf{I} + (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (5.79)$$

is the stress tensor and $\mathbf{I}$ is the idem factor. The non-dimensional velocity $\mathbf{u}$, non-dimensional temperature T , and the non-dimensional coordinates (r, z) are defined as the dimensional variables divided by characteristic scales of V , T_c , and L , respectively. Equation (5.76) includes the thermal Peclet number, $\text{Pe}_T \equiv \rho C_p L V / k_{\text{local}}$, where ρ is the melt

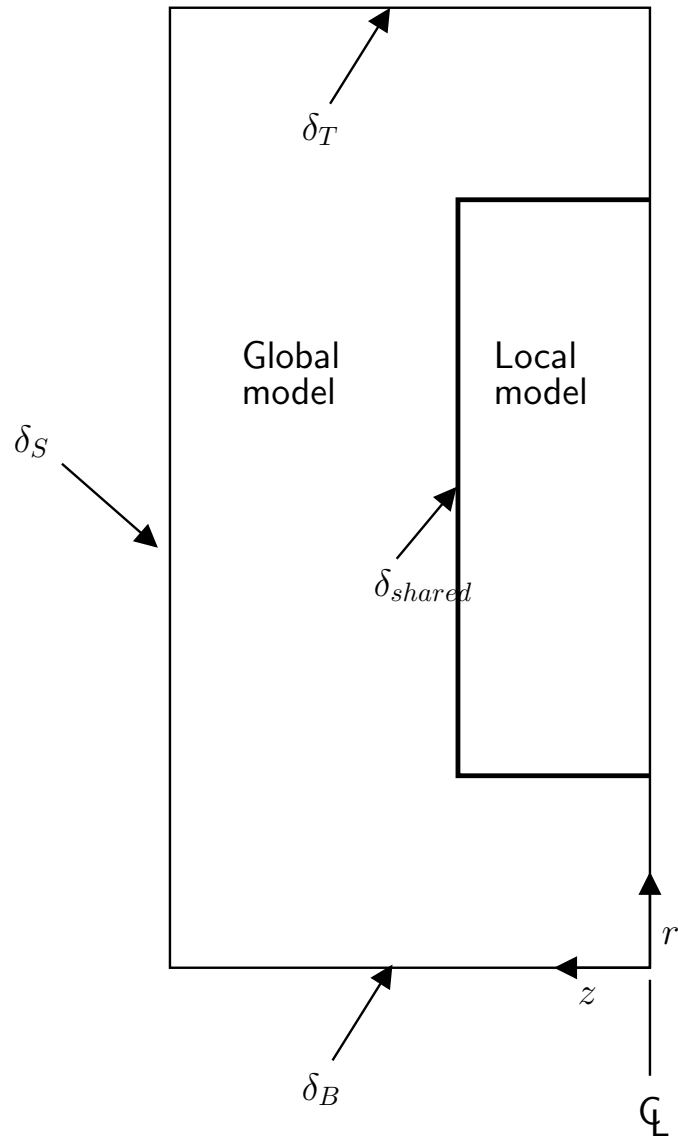


Figure 5.6: Schematic representation of the two-dimensional model problem.

density, C_p is the melt heat capacity, and k_{local} is the melt thermal conductivity. The parameter $Re \equiv \rho V L / \mu$ in Equation (5.77) is the Reynolds number, where μ is the melt viscosity. In the buoyancy term is the Rayleigh number, $Ra \equiv \rho g L^2 \beta T_c / \mu V$ where g is the gravitational constant and β is the thermal compressibility. Fluid flow is driven only by natural convection through the buoyancy term in the momentum balance equation. Symmetry conditions are applied along the system centerline.

The equations are discretized using Galerkin's finite element method, and the resulting set of nonlinear algebraic equations are solved using Newton's method. The local model's residual vector and its Jacobian matrix correspond to $\mathbf{r}_1$ and $\mathbf{J}_{11}$ of the ABN method, respectively.

Global Model

The global subdomain of the model, submodel 2, is analogous to a melt-crystal furnace, but extremely simplified. The global model consists of a single material in one phase that encloses the local model on all sides; essentially a heat bath. Only heat conduction is considered in the global model

$$k_{global} \nabla^2 T = 0 \quad (5.80)$$

where k_{global} [W/K·m] is the conductivity of the heat bath. Unlike Cats2D, CrysVUn uses dimensional values for its variables—temperature is degrees K and length is in meters. When passing information back and forth between the two codes, the variables must be converted appropriately. The global model has three external boundary conditions:

$$T = 300 \text{ K} \quad \text{on } \delta_T \quad (5.81)$$

$$\mathbf{n} \cdot \mathbf{q} = 0 \text{ W/m}^2 \quad \text{on } \delta_S \quad (5.82)$$

$$T = 350 \text{ K} \quad \text{on } \delta_B. \quad (5.83)$$

In this situation, the furnace is hotter at the bottom and cooler at the top—similar to a Rayleigh-Bérnard cell [17, 143].

The governing equations are discretized using the finite volume method, and the resulting set of linear algebraic equations is solved using Newton's method. The global models' residual and Jacobian matrix correspond to $\mathbf{r}_2$ and $\mathbf{J}_{22}$ of the ABN, respectively.

As in the one-dimensional model, we use a set of parameter values that are representative of crystal growth systems, namely $k_{global} = 2 \text{ W/K}\cdot\text{m}$, $k_{local} = 1 \text{ W/K}\cdot\text{m}$, $Pe_T = 1.0$, and $Re = 1.0$. Nonlinearity is introduced to the problem by adding buoyancy driven flow

to the local model by varying the Rayleigh number. The problem is linear when $Ra = 0$ (no flow), and nonlinear when $Ra > 0$. We use $Ra = 6 \times 10^3$ as the representative value for the nonlinear case.

Coupling Equations

As with the one dimensional models previously discussed, continuity of temperature and normal heat flux along the shared boundary must be enforced to make the overall problem well-posed

$$T_{local} = T_{global} \quad (5.84)$$

$$q_{local} = q_{global} \quad (5.85)$$

where T_i , $i = local$ or $global$, are the temperatures along the shared boundary, δ_{shared} , and q_i are the heat fluxes normal to the shared boundary. We construct mixed boundary conditions along the shared boundary for each domain

$$q_{local} = q_{global} + \frac{1 - \alpha}{\alpha} (T_{local} - T_{global}) \quad (5.86)$$

$$q_{global} = q_{local} + \frac{1 - \beta}{\beta} (T_{global} - T_{local}) \quad (5.87)$$

where, again, α and β are arbitrary coupling parameters that vary from 0 to 1. Due to the options available for boundary conditions in CrysVUn, however, we have set $\beta = 0$, reducing the boundary in global domain to the temperature continuity condition, Equation (5.84), or Dirichlet boundary condition. As in the one dimensional example, α cannot equal β for well-posedness. Note that a pure flux boundary condition for the local model, $\alpha = 1$, solves the temperature to within an arbitrary constant. To calculate the correct absolute temperature, either the temperature must be specified at a point within the local model or a value of $\alpha < 1$ must be used. Since we do not know the temperature in the local model *a priori*, α is set to less than 1 for a Robin boundary condition. In the calculations shown here, $\alpha = 0.9$. We were not able to solve the problem using BGS because of the limitations imposed on the (α, β) space.

Results

Many different parameters govern convergence of the ABN method. For the iterative solution of the Schur complement problem, we use GMRES settings of maximum of 50

Krylov vectors and a convergence tolerance of 1×10^{-6} . The overall problem is deemed converged when both subproblems are considered converged with residual L_2 norms less than 1×10^{-4} , and the L_2 norm of the difference in boundary temperatures is less than 1×10^{-4} . Again, a value of $\alpha = 0.9$ is chosen, while $\beta = 0$ is set by CrysVUn boundary conditions. A value of $\epsilon = 1 \times 10^{-4}$ is used for the JFNK approximation of the Schur complement.

Figure 5.7 shows temperature contours of (a) the global and (b) the local domains case spaced at intervals of $\Delta T = 1$ K for the linear $Ra = 0$. Heat conduction only is considered in both the local and global domains, with a zero initial condition. Figure 5.8 shows temperature contours of (a) the global model and (b) temperature and streamline contours of the local model of the nonlinear $Ra = 6 \times 10^3$ case with heat conduction in the global domain, and heat conduction, convection, and buoyancy driven flow in the local domain. The temperature contours of both the global and local domains are spaced at $\Delta T = 1$ K. An initial condition of $Ra = 3 \times 10^3$ is used. Comparing Figure 5.7(a) and Figure 5.8(a), slight differences in the shape of the global temperature contours can be seen. The temperature profile of the local domain, Figure 5.8(b) is highly influenced by convection and is dramatically different than the temperature contour plot in 5.7(b). The streamline plot shows two flow cells rotating counter-currently. The dashed contours in the streamline plot indicate flow in the clock-wise direction at contours of $\psi_{max}/10$, where $\psi_{max} = 1.6579$ is the dimensionless maximum value of the stream function. Similarly, the solid contours indicate flow in the counter-clock-wise direction at contours of $\psi_{min}/10$, where $\psi_{min} = -1.6576$ is the dimensionless minimum of the stream function.

Temperature and heat fluxes along the shared boundary as a function of the arc length of said boundary are compared between the local and global models in Figure 5.9. The arc length is calculated relative to the radius of the local domain. In both the linear and nonlinear cases, there is good matching between the profiles for both the temperature and heat flux. The slight mismatch of the heat flux at the corners between $\delta_B - \delta_S$ and $\delta_S - \delta_T$ is due to numerical error defining the normal vector of a corner. Figure 5.9 also shows how the addition of fluid flow and heat convection affects the boundary thermal profile, and especially the heat flux. The addition of fluid flow greatly changes the heat transfer characteristics of the entire problem. Use of simple one-way coupling is unable to account the change in the heat load from convection.

We also tested the rate of convergence dependence of the ABN method on the number of Krylov subspace vectors used by the iterative solver GMRES to approximate the off-diagonal block Jacobians. With each additional Krylov space vector, each subsolver is

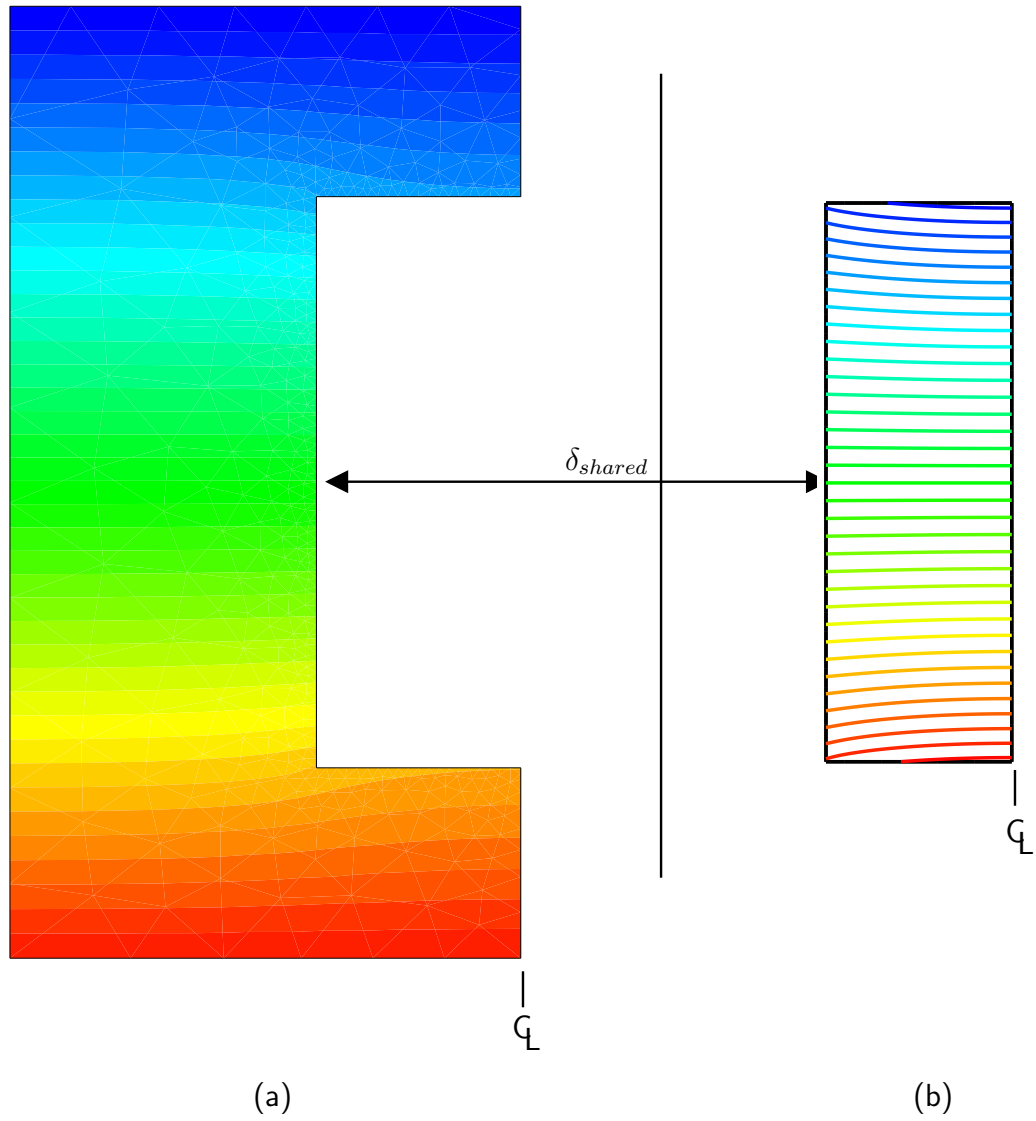


Figure 5.7: Temperature contours of the (a) global and (b) local domains of the linear two-dimensional model, $Ra=0$, where $\Delta T = 1$ K.

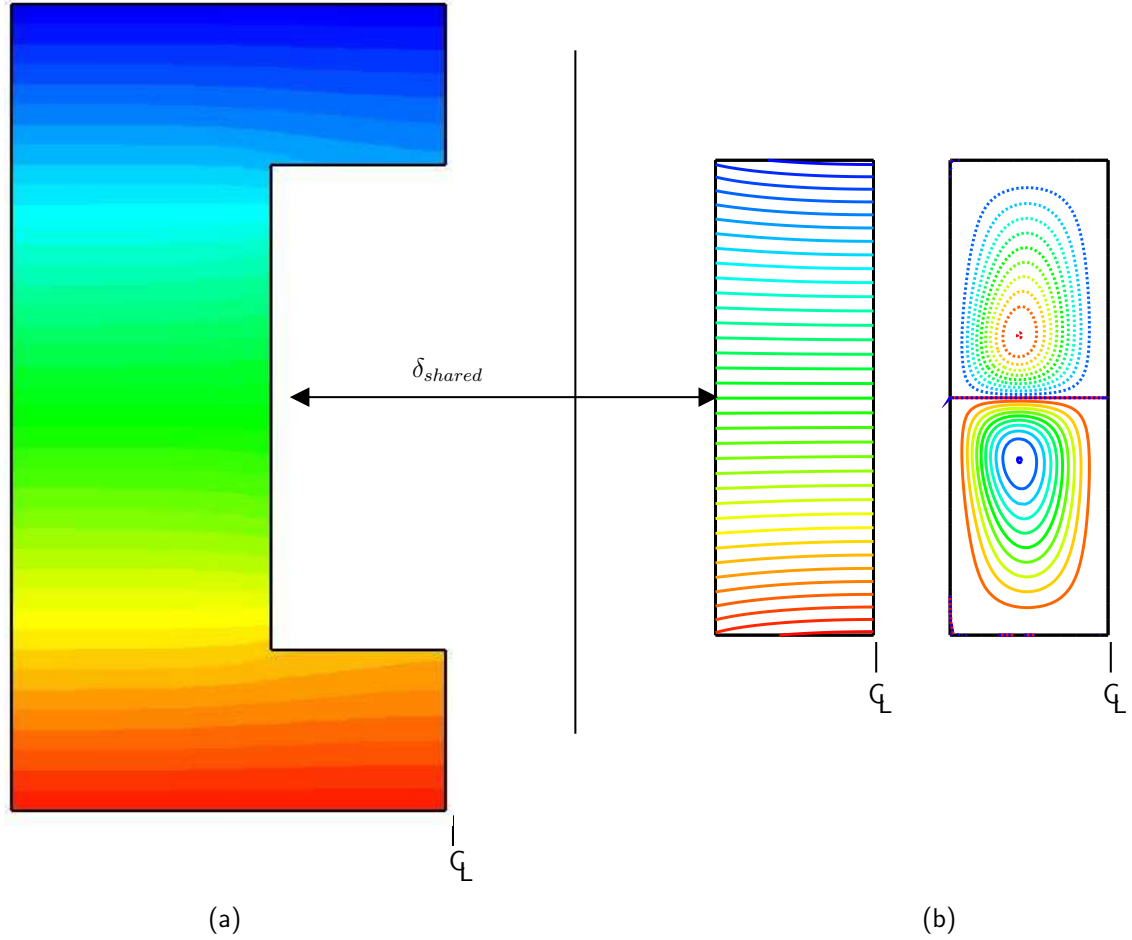
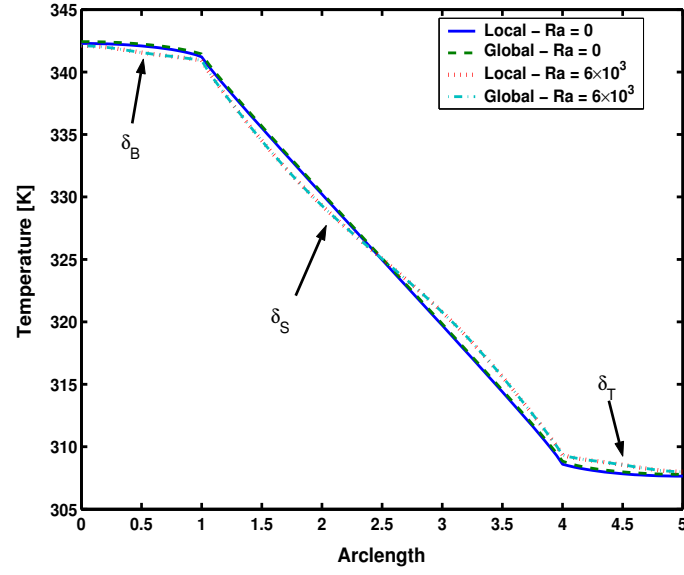
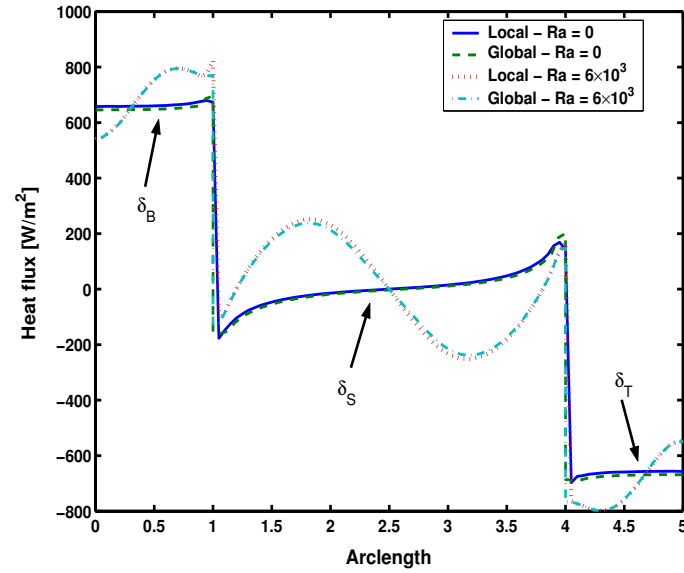


Figure 5.8: (a) Temperature contours of the global domain, and (b) temperature and streamline contours of the local domain of the nonlinear two-dimensional model, $Ra = 6 \times 10^3$. Temperature contours are spaced as $\Delta T = 1$ K. Dashed streamline contours indicate clockwise flow at $\Delta\psi = \psi_{max}/10$, and solid streamline contours indicate counter-clockwise flow at $\Delta\psi = \psi_{min}/10$, where $\psi_{max} = 1.6579$ and $\psi_{min} = -1.6576$, dimensionless.

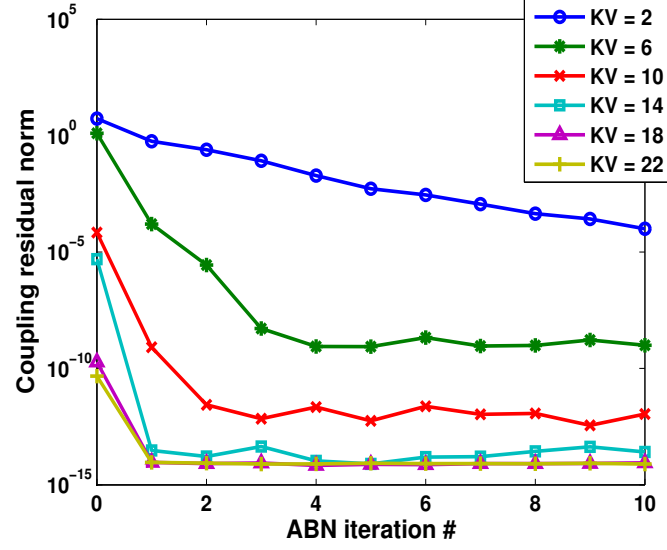


(a)

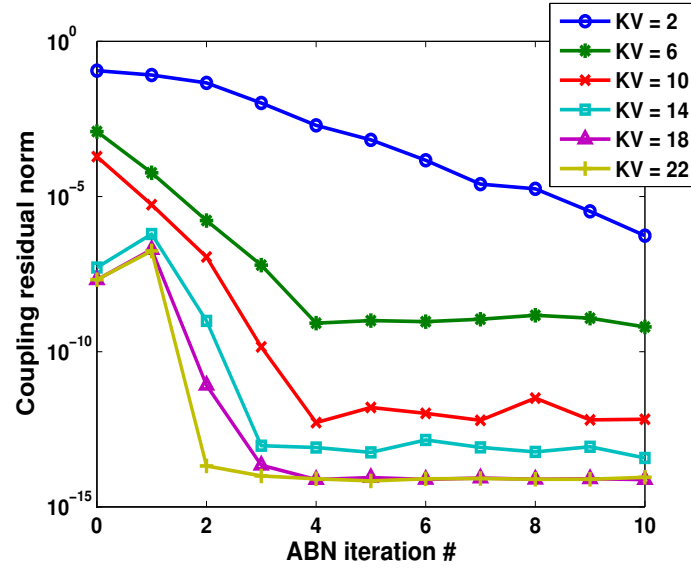


(b)

Figure 5.9: Comparison of (a) temperature [K] and (b) heat flux [W/m^2] as a function of arc length (with respect to the ampoule radius) along the shared boundary for $Ra = 0$ and $Ra = 6 \times 10^3$ for the two-dimensional model.



(a)



(b)

Figure 5.10: Residual norm of boundary data versus block Newton iteration number at different number of Krylov vectors for (a) $Ra = 0$ and (b) $Ra = 6 \times 10^3$ of the two dimensional model.

called once, and each iterate internally multiple times. In Figure 5.10, the residual norm of the shared boundary is plotted against the number of outer, block iterations for the (a) linear and (b) nonlinear cases. We define the residual norm as the L_2 -norm of the difference in temperatures calculated by CrysVUn and Cats2D along the shared boundary. The initial shared boundary condition for the linear case is based on a monolithic solution of the entire system computed by CrysVUn. The initial condition of the nonlinear case is based on the self-consistent boundary condition of the linear case with $Ra = 6 \times 10^3$ as calculated by the uncoupled local model, Cats2D.

If the linear case is solved monolithically, the model converges in one iteration. With the ABN method, we approximate a monolithic model. As the number of Krylov vectors used to solve the Schur complement problem for $\mathbf{S}\Delta\mathbf{y} = \mathbf{r}_S$ we approach the convergence behavior of a monolithic model. The larger the number of Krylov vectors, the better the iterative solution approximation of $\Delta\mathbf{y}$. This can be seen in the difference in residual norm on the first iteration for the different cases. In this relatively simple two-dimensional model, only six Krylov vectors are needed for a residual norm $< 1 \times 10^{-4}$. As the ABN method continues to iterate, the residual norm stalls due to accumulation of round-off error for the cases of $KV \geq 6$.

In the nonlinear case, we again see a difference in the residual norm of the first iteration for the different cases which is due to the increased accuracy in the approximation of $\Delta\mathbf{y}$ with increase number of Krylov vectors. The solution also converges faster when the number of Krylov vectors increased until we reach $KV \geq 14$. For these cases, the solution would be deemed converged on the first iteration since the L_2 -norm is $< 1 \times 10^{-4}$. On the second iteration, the residual norm increases and then decreases again. For the cases of $KV \geq 6$, the residual norm eventually stalls as the method continues to iterate due to round-off error.

5.4.3 Electrodynamic Gradient Freeze Furnace Model

The final model we present is a multi-scale, steady-state, modular model to study the growth of single crystal cadmium zinc telluride (CZT) in an industrial electrodynamic gradient freeze (EDG) furnace. This model is similar to that presented in [136] with some simplifications to the furnace geometry and physical properties for CZT given in Chapter 2, and solved using the ABN algorithm. The complete model is divided into two domains. Submodel 1 is the local model of the melt, crystal, and melt-crystal interface which uses Cats2D as the solver. Submodel 2 is the global model of the furnace using CrysVUn as

the solver.

A schematic diagram of the system and the computational meshes are shown in Chapter 4, Figure 4.1, where meridional views are shown of the two-dimensional, axisymmetric domains. The global model is constructed to include all of the details of the furnace and ampoule assembly. The local model consists only of the melt, interface, and a portion of the crystal. There are 7,115 degrees of freedom in the global model, and 28,825 degrees of freedom in the local model. We assume that a quasi-steady-state (QSS) model appropriately describes the behavior of heat transfer and fluid flows within the system. See References [8, 92, 97, 195] and Chapter 2 and for complete model descriptions. Again, Cats2D is designated solver 1 and CrysVUn as solver 2.

Figure 5.11 shows results for a typical furnace geometry and operating parameters. A temperature image plot of the furnace is given in Figure 5.11(a). A Dirichlet boundary condition sets the temperature along the external boundary of the furnace to room temperature. Temperature and streamline contours of the local model, or growing crystal, are shown in Figure 5.11(b). Temperature contours, or isotherms, are spaced at intervals of $\Delta T = 1$ K. Dashed streamline contours at intervals of $\Delta\psi = 5 \times 10^{-8}$ m³/s indicate flow in the clockwise direction; and solid contours at intervals of $\Delta\psi = 5 \times 10^{-9}$ m³/s indicate flow in the counter-clockwise direction. The temperature field in the solid shows a two-dimensional thermal field, with heat flowing downward and outward which causes the melt-crystal interface to be strongly concave with respect to the melt. The isotherms in the melt are flattened in the bulk due to strong convective mixing throughout most of the melt domain. Thermal boundary layers in the melt occur near the ampoule wall. The streamline plot shows that the larger flow cell rotates in the clockwise direction. Buoyant forces in the radial direction drive this recirculating flow where warm fluid rises along the centerline and cooler fluid flows downward along the wall. The majority of Bridgman systems exhibit flow in the opposite direction. This system, however, is at an extremely low temperature gradient such that the effect of latent heat is large enough to reverse the direction of the radial temperature gradient through the bulk of the melt. Surface tension gradients caused by radial thermal gradients along the free surface at the top of the melt lead to Marangoni forces also in the clockwise direction. A flow cell rotating in the counter-clockwise direction develops between the two flow cells rotating in the clockwise direction.

The temperatures and heat fluxes calculated by the two codes, CrysVUn and Cats2D, are compared in Figures 5.12(a) and 5.12(b) by plotting them as a function of arc length of the shared boundary. The arc length is non-dimensionalized with respect to the ampoule radius. The top, side, and bottom of the boundary are indicated by δ_T , δ_S , and δ_B

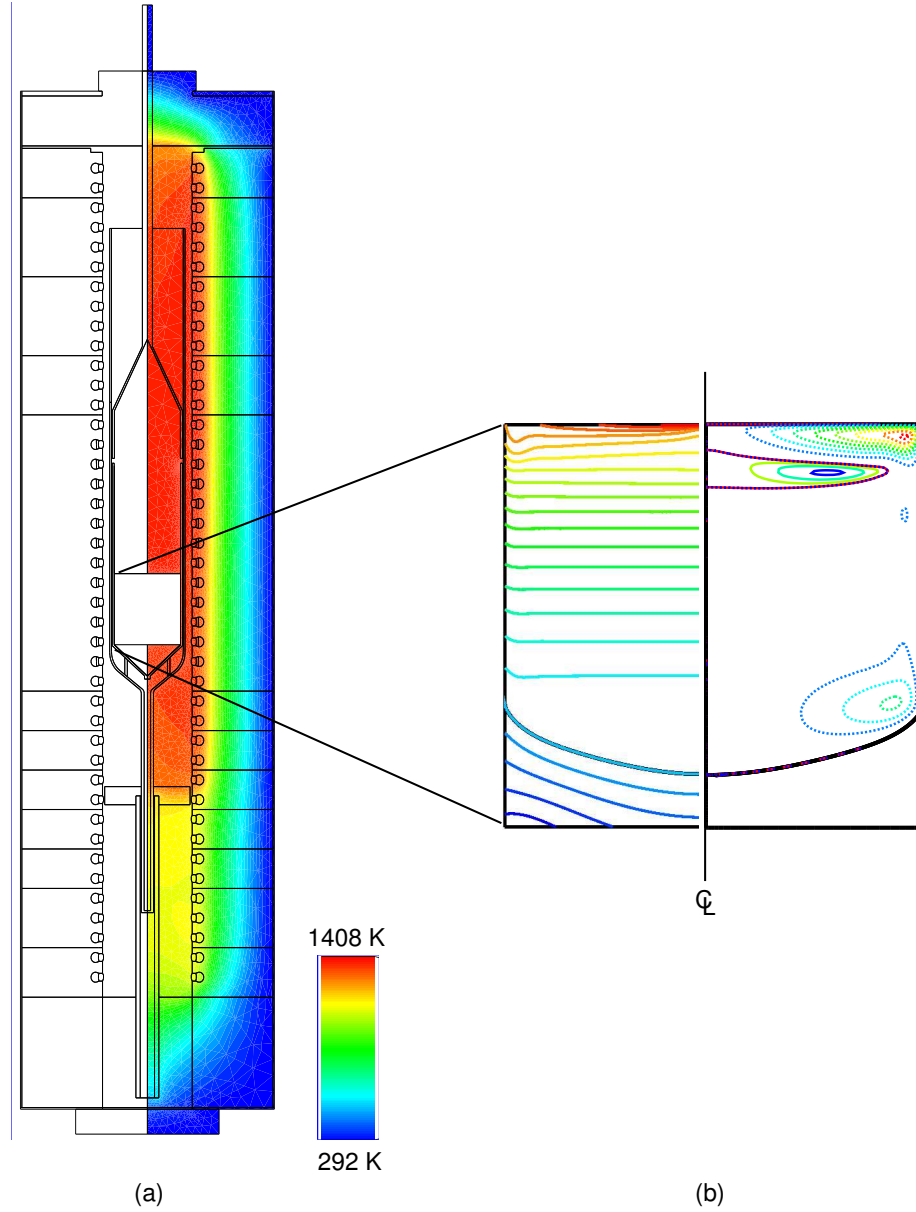


Figure 5.11: Results of the self-consistent model of an EDG furnace: (a) temperature image plot of global furnace calculated by CrysVUn, scale indicated, and (b) temperature and streamline contours of the local crystal calculated by Cats2D. The temperature contours of the local model are spaced at $\Delta T = 1$ K. Dashed streamline contours indicate clockwise flow at intervals of $\Delta\psi = 5 \times 10^{-8} \text{ m}^3/\text{s}$, and solid streamline contours indicate counter-clockwise flow at intervals of $\Delta\psi = 5 \times 10^{-9} \text{ m}^3/\text{s}$.

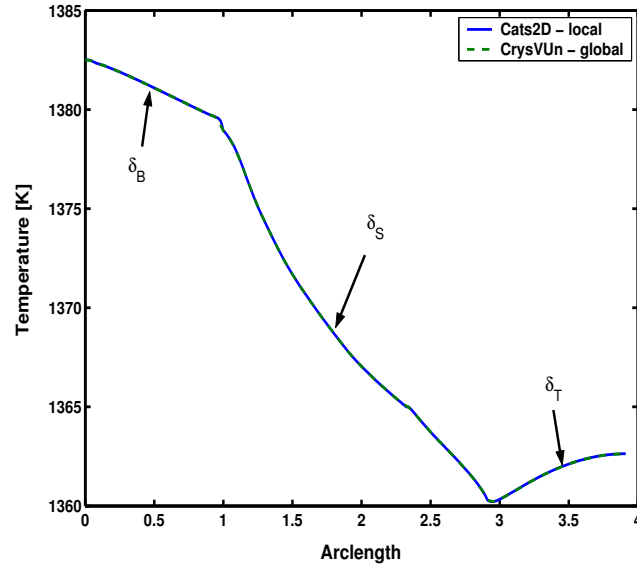
respectively. Both the temperatures and heat fluxes show good agreement at the shared boundary. As in the simple two-dimensional model, we see discontinuities of the heat flux at the corners of the domain due to the non-uniqueness of the outward drawn normal. There is also a small peak in the heat flux along δ_S where the melt-crystal interface meets the ampoule wall due to the heat conductivity mismatch between the melt phase, solid phase, and ampoule material.

The rate of convergence for this complicated, multi-scale model is examined by plotting the residual L_2 -norm of the temperature difference along the share boundary versus block Newton iteration number, Figures 5.13-5.15. Both the effects of the number of Krylov vectors used by GMRES and the parameter ϵ used in the finite difference approximation of the Schur complement on rate of convergence are examined.

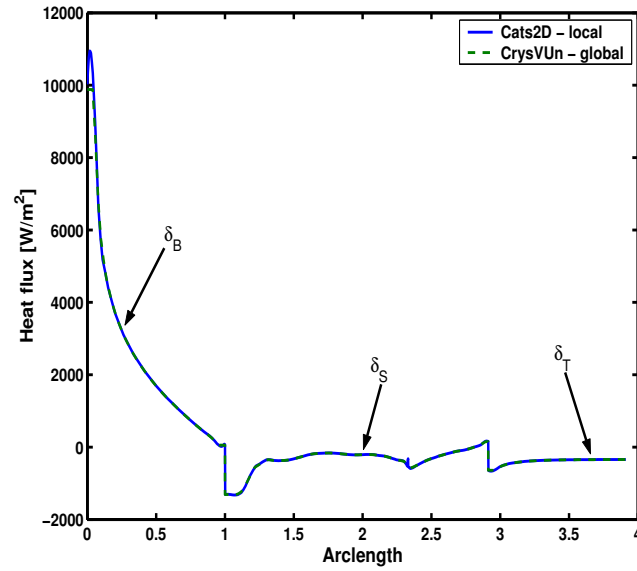
For our initial condition, we use a fully coupled model where the latent heat coefficient, $\text{Pr}_m \text{Sf}_m$ in Equation (2.10), is equal to 3.1375×10^1 , where the final value is 3.1678×10^1 . All other non-dimensional quantities in the governing equations are at their full values. This initial condition was chosen because we were able to reach the final value of the latent heat coefficient without using continuation from this point.

Figure 5.13 shows the sensitivity of the rate of convergence for different numbers of Krylov vectors with $\epsilon = 1 \times 10^{-4}$. The rate of convergence increases with the number of Krylov vectors used by GMRES which agrees with the behavior seen in the simple, two-dimensional model. In the case of two Krylov vectors, $KV = 2$, the residual norm displays a general downward trend with iteration number. The cases of $KV = 10$ and 14 exhibit quadratic convergence on the first three iterations, but then the method stagnates and no further improvement in the residual norm is seen. For the case of $KV = 26$, the residual norm actually increases after the first iteration, and then achieves superlinear rate of convergence. After the fifth iteration, the residual norm then begins to steadily increase. We suspect the increase may be due to round-off error accumulation in the GMRES solve of the Schur complement problem. As in the simple two-dimensional case, there is a difference in L_2 -norm on the first iteration for the different number of Krylov case. Again, this norm is dependent on the accuracy of the solution of the Schur complement problem that is solved using an iterative method.

The rate of convergence is also sensitive to the value of ϵ chosen for the finite differencing approximation. Figure 5.14 shows the effect of ϵ on the rate of convergence using 18 Krylov vectors. In the case of a relatively large value of $\epsilon = 1 \times 10^{-2}$, the finite difference approximation is cruder, and displays a lower rate of convergence than the other cases. The cases of $\epsilon = 1 \times 10^{-3}$ - 1×10^{-5} , the method again stagnates with increasing iterations



(a)



(b)

Figure 5.12: Comparison of (a) temperature [K] and (b) heat flux [W/m²/s] as a function of arc length (with respect to ampoule radius) along the shared boundary of the EDG furnace model.

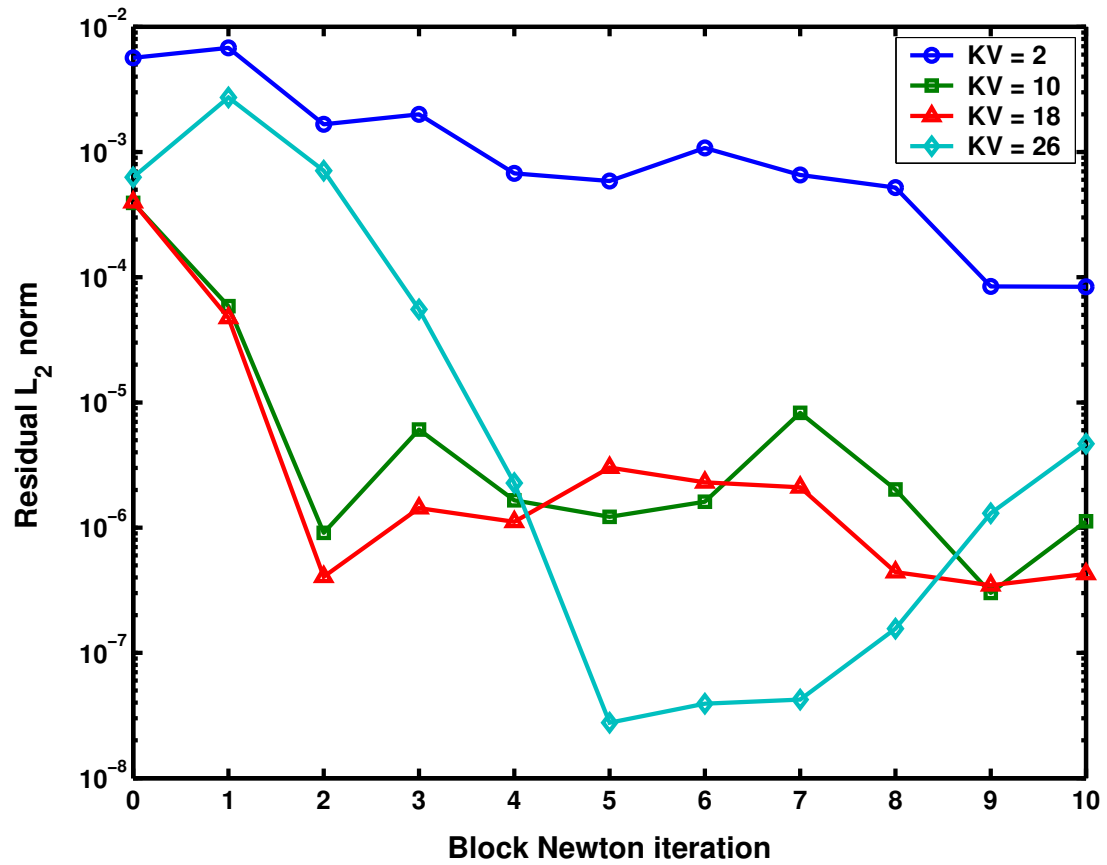
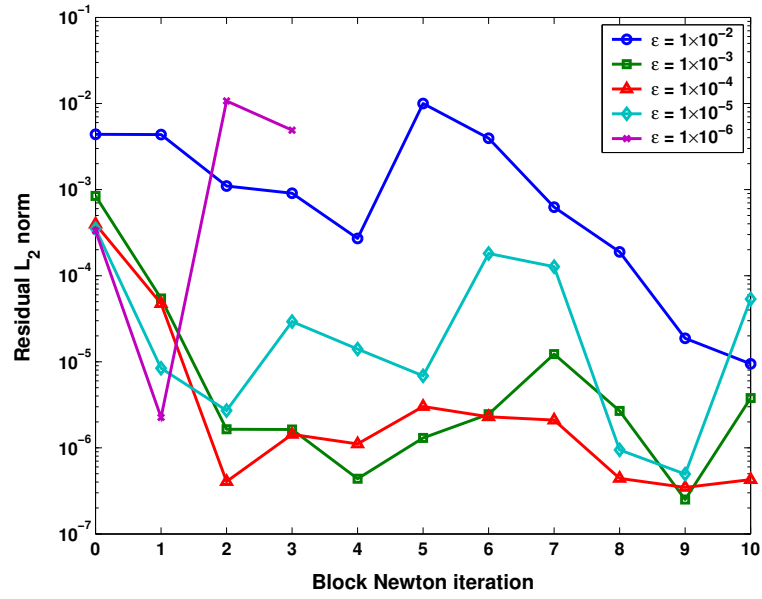
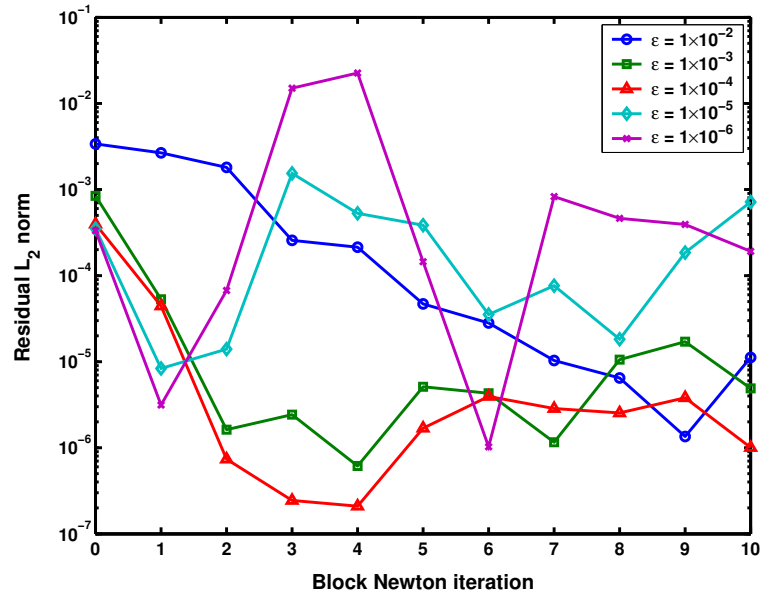


Figure 5.13: The dependence of rate of convergence on Krylov space size is determined by plotting the residual L_2 norm of the difference in Cats2D and CrysVUn boundary data versus block Newton iteration number for $\epsilon = 1 \times 10^{-4}$.

Figure 5.14: The dependence of rate of convergence on ϵ for $KV = 18$.Figure 5.15: The dependence of rate of convergence on ϵ for $KV = 22$.

as displayed when varying the number of Krylov vectors with constant ϵ . Of these three cases, the rate of convergence of $\epsilon = 1 \times 10^{-5}$ fluctuates the most. When $\epsilon = 1 \times 10^{-6}$, the method actually fails in that the local mesh deformed such that the subsolver Cats2D has a fatal error. At smaller values of ϵ , the finite differencing step may not capture enough information to be a good approximation.

The sensitivity test of rate of convergence on ϵ was repeated for $KV = 22$, shown in Figure 5.15. The method behaves similarly at the different values of ϵ for $KV = 18$ and 22, but the fatal error does not occur for $\epsilon = 1 \times 10^{-6}$ when $KV = 22$. However, at $\epsilon = 1 \times 10^{-5}$, the residual norm fluctuates more when $KV = 22$ than at $KV = 18$. In both cases, quadratic convergence was achieved only during the first three iterations when $\epsilon = 1 \times 10^{-4}$. The results suggest that the success of the method depends on some interplay between the number of Krylov vectors used and an appropriate value of ϵ . In general, though, the value of ϵ should not be too large or too small.

5.5 Concluding Remarks

We have presented an Approximate Block Newton iteration algorithm as a means to develop modular, coupled models that represent steady-state crystal growth from the melt. These coupled models are divided into two subdomains where temperature and heat fluxes are exchanged along a shared, physical boundary. Each domain is characterized by its own independent model that only interacts with its partner model at the shared boundary. Previous work [44, 136, 196] employed a much simpler technique, the block Gauss-Seidel method, as the coupling method. This method ignores the cross-term dependencies of a model's equations on the other model's variables leading to poor convergence behavior (see Equation (5.9)). Under-relaxation of the block Gauss-Seidel method also proved ineffective at improving the robustness, in terms of overall model convergence, of the algorithm as shown in Appendix A and [44]. This behavior leads us to implement a method that includes these cross-term dependencies, namely the ABN method.

The ABN method is an approximation of full Newton's method; however, it still retains the vital information provided by the cross-term dependencies. This information is kept by using a Jacobian-free, Newton-Krylov method where the off-diagonal block Jacobians are approximated using finite differences and an iterative solver to solve the needed matrix-vector products. The ABN method is much more complicated conceptually and to implement than the block Gauss-Seidel method, but the expected payoff in terms of robustness outweighs this expenditure.

Using a one-dimensional problem that is representative of crystal growth, we were able to directly compare the performance of the block Gauss-Seidel method and the ABN method. Just under half of the (α, β) pairings gave conditions in which the block Gauss-Seidel method would converge. Furthermore, convergent pairings changed when nonlinearity was introduced into the problem through radiation effects which is more indicative of high temperature melt growth processes of interest. In comparison, the ABN method achieved convergence for $> 98\%$ of the cases regardless of nonlinearity. Moreover, the effort expended to reach convergence was much less with the ABN method than the block Gauss-Seidel method. The rate of convergence of the ABN method was compared to that of full Newton's method, and showed that quadratic convergence could be achieved under certain conditions.

We also developed a simple, two-dimensional model where we coupled the computer codes CrysVUn and Cats2D. The final results showed excellent matching between temperatures and heat fluxes along the shared boundary. The rate of convergence of the ABN method was shown to be strongly dependent on the number of Krylov vectors used in the iterative solver steps for both linear and nonlinear case studies. Quadratic convergence was attained with ≥ 20 Krylov vectors. With each additional Krylov vector, both subsolvers are called once each. The number of Krylov vectors can be viewed as a trade-off between efficiency and accuracy.

Finally, we applied the ABN method to a modular, multi-scale, steady-state melt growth model of an industrial gradient freeze furnace. Again, the final result showed excellent agreement between the temperatures and heat fluxes along the shared boundary. At best, though, super-linear convergence was achieved for this model. The rate of convergence still depends on the number of Krylov vectors, but as the ABN method continues to iterate the residual norm starts to fluctuate in the cases of ≥ 6 Krylov vectors. The value of ϵ chosen for the finite difference approximation is also significant. This value cannot be too large or small. If ϵ is too large, the finite difference is no longer a good approximation. When the value is too small, the finite difference does not reflect the sensitivity of the off-diagonal block Jacobians. While there are still areas to be explored about the behavior of the ABN method and its implementation, we see this method as an exciting new way to couple our existing models to build more realistic simulations.

Chapter 6

Schur complement formulation for solving a free-boundary crystal growth problem

Contents

6.1	Chapter Summary	136
6.2	Introduction	137
6.3	Model Formulation	138
6.3.1	Governing Equations	139
6.3.2	Schur Complement Formulation	141
6.3.3	Approximations and Preconditioning	146
6.4	Results and Discussion	148
6.4.1	Comparison of Convergence Rate	149
6.4.2	Scaling of Computational Effort	152
6.4.3	Effectiveness of Approximations and Preconditioning	163
6.5	Concluding Remarks	166

6.1 Chapter Summary

A Schur complement formulation is used to solve a two-dimensional, free-boundary, melt-crystal growth problem by exploiting the underlying structure of the discretized equation

set. Instead of applying a direct or iterative technique to solve the full matrix equation resulting from Newton’s method, the equation set is segregated into block form separating the field unknowns and the variables describing the location of the melt-crystal interface. The Schur complement of the Jacobian matrix is formed using block Gauss elimination. The resulting Schur complement problem can be solve using either direct or iterative methods, here we use full LU decomposition and GMRES respectively. We achieve quadratic convergence with the direct solver, and quadratic convergence is possible with GMRES depending on its convergence criterion. The Schur complement formulation is computationally slower than solving Newton’s method with either a direct or iterative method, however, but scales with problem size in a manner similar to iterative techniques. Estimates on how the methods scale in three-dimensional problems are also made. Reductions in computation time can be made through approximations and preconditioning of the Schur complement problem.

6.2 Introduction

For a model of a melt-crystal growth system to be truly predictive, it must accurately portray the two-phase structure of a solidifying melt. Specifically, determining the macroscopic location of a free-boundary caused by the phase transition between a melt and solid phase is crucial to producing a physically realistic solution. This can be a challenging calculation especially in a three-dimensional model. Various methods have been used to tackle such problems. For example, the problem domain can be discretized into two sub-domains on a structured, computational mesh and the boundary separating the two fields deforms such that the temperature along that boundary is the melting-point temperature [193]. Another method is to use an unstructured mesh where the location of the melt-crystal interface is interpolated between the nodes in a diffuse manner. Phase-field models, a type of diffuse interface model, have been of great interest recently. With this method the phase-field varies rapidly, but continuously, from solid to liquid over a thin interfacial region [60].

The path we have chosen to solve such problems is to use a deforming, structured mesh where the melt-crystal interface location is set implicitly by the melting point temperature. For our simulations, we simultaneously solve the heat transfer equations and the governing equations of the deforming mesh using a direct, frontal solver with Newton’s method. A PDE-based method is used to numerically generate the grid. In this manner, we have been able to calculate robustly and efficiently two-dimensional crystal growth models described by continuum level transport phenomena [92, 93, 110–112, 166, 190–192, 194]. Extending

this method toward a three-dimensional problem has only garnered limited success due to the large number of degrees of freedom to compute, making the use of a direct solver prohibitively expensive. Instead, iterative solver techniques, such as GMRES, are used, but solutions are difficult to achieve with current computing resources. In previous work, Yeckel et al. [189] had to decouple the field and mesh equations in order to calculate the free-boundary while solving with GMRES. This procedure is usually successful in the problems examined, but only a linear rate of convergence was reached at best.

We propose an alternative means of solving the linear system of equations arising from Newton’s method that may be an effective tool in solving a free-boundary, melt-crystal growth problem in three-dimensions in an effort to recapture quadratic convergence. With this method, we reorder the governing equations of our system to separate the field variables from the interface unknowns such that a Newton step is rewritten into a Schur complement form. In a sense, we are ‘mapping” out the difficult part of the problem—determining the interface location—from the governing transport equations. We first solve for the interface unknowns and then back solve for the field unknowns. We intend to show that a good approximation of the interface unknowns is more critical to Newton’s method converging than the approximation of the field unknowns.

Solving free-boundary, crystal growth problems using a Schur complement formulation has not been previously explored. Schur complement methods are popular in other fields such as a preconditioner of the incompressible Navier-Stokes equations [108, 163, 179], and in domain decomposition of multigrid techniques [72, 106, 132, 152, 169]. To study the behavior of a Schur complement based solver, we have designed a simple two-dimensional, two-phase, melt-crystal growth problem. While the ultimate goal is to apply this method on three-dimensional problems, a two-dimensional model is a more cost effective means of exploring the behavior of this method. The problem domain is based on a crystal grown in a vertical Bridgman system. The next section discusses both the physical governing equations of the model and the application of a Schur complement formulation to solve the problem.

6.3 Model Formulation

The model presented here uses the Galerkin finite element method (GFEM) to discretize the governing equations, with a modified-Newton solver to solve the resulting nonlinear, algebraic equations. In addition to discretizing the field equations of heat, mass, and momentum transport, we use a deforming grid method to represent phase interfaces. In

free and moving boundary problems, the domain geometry is unknown, making it necessary to parameterize the locations of the nodes.

6.3.1 Governing Equations

For our sample problem, we solve a two-phase, heat conduction problem under cylindrical geometry with a free-boundary at the melt-crystal interface. Figure 6.1(a) shows a schematic diagram of the domain of the sample problem. The equations and geometry are non-dimensionalized as laid out in Chapter 2:

$$\nabla^2 T = 0 \quad \text{in } \Omega_m \quad (6.1)$$

$$\kappa \nabla^2 T = 0 \quad \text{in } \Omega_s \quad (6.2)$$

where T is temperature, and $\kappa \equiv k_s/k_m$ is the ratio of melt to solid heat conductivities. In our model, we apply the melting-point isotherm condition along the boundary between the melt and solid phases

$$T = T_{mp} = 0 \quad \text{at } \delta\Omega_{ms} \quad (6.3)$$

where T_{mp} is the melting-point temperature, set here to equal 0. Reference [193] gives a detailed explanation of this boundary condition. Along the external boundary, we apply Newton's law of cooling

$$\mathbf{n} \cdot \nabla T = h(T - T_f) \quad \text{at } \delta\Omega_{ext} \quad (6.4)$$

where h is a dimensionless heat transfer coefficient, and $T_f = z$ is the applied, non-dimensionalized, furnace temperature. Symmetry conditions hold at the system centerline.

Elliptical grid generation techniques (see Chapter 2) are used to govern the movement of the deforming mesh [193]. The location of the melt-crystal interface is solved implicitly by setting the location of the interface nodes at the melting-point temperature. This is done by replacing the mesh equations that govern the interface boundary in the z -direction with the isotherm condition, Equation (6.3). The r -coordinate degrees of freedom along the interface are still governed by the elliptical grid equations. The isotherm boundary condition, Equation (6.3), can be viewed as a constraint on the mesh equations, similar to how the incompressible Navier-Stokes equation is constrained by the continuity equation. Additional boundary conditions on the mesh equations set the shape of the domain, while allowing the grid points to move as needed.

GFEM is used to discretize the governing partial differential equations into a set of nonlinear algebraic equations. Newton's method is utilized to solve this set of nonlinear

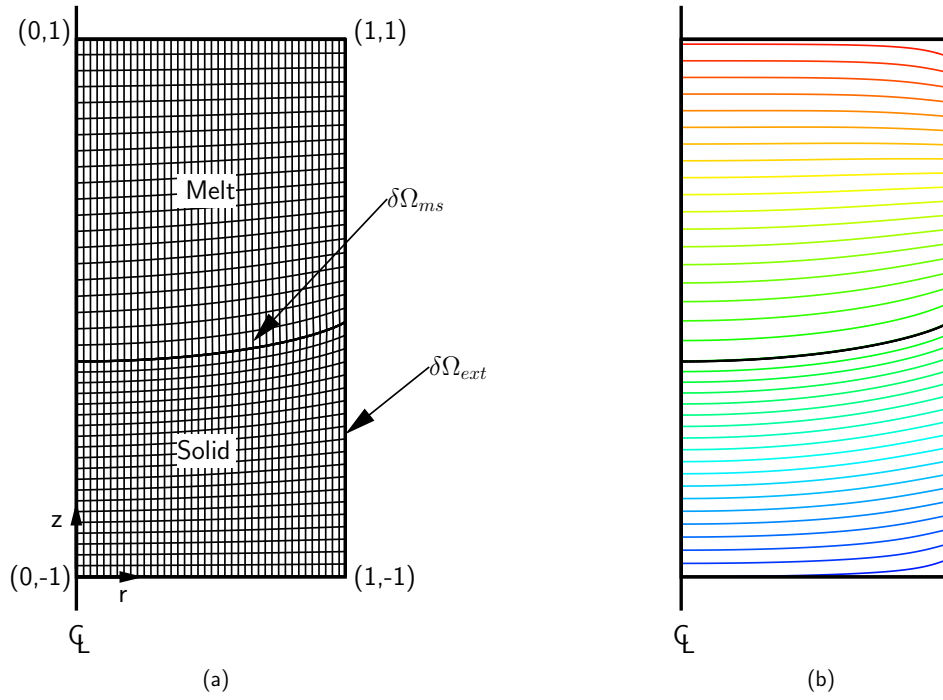


Figure 6.1: (a) Schematic of two-dimensional, two-phase sample problem discretized by a representative 40×40 element mesh, and (b) solution of the temperature field.

equations. Figure 6.1(b) gives the solution of the temperature field computed on a 40×40 element mesh where a direct, frontal solver is used to solve each Newton step.

6.3.2 Schur Complement Formulation

To reiterate, we are solving the resulting set of nonlinear equations using Newton's method, where each Newton iteration has the form

$$J\delta = -R \tag{6.5}$$

where R is the residual vector, J is its Jacobian matrix, and δ is the update vector. Instead of using a direct or iterative method to solve the linear equation set of each Newton iteration, the equations are solved using the Schur complement of the Jacobian matrix, J . First, the equations are reordered with the field and mesh variables first followed by the interface unknowns (the unknowns corresponding with Equation (6.3)). Equation (6.5) now has the following block structure

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_i \end{bmatrix} = \begin{bmatrix} -r_f \\ -r_i \end{bmatrix} \tag{6.6}$$

where A is an $n \times n$ matrix, B is an $n \times m$ matrix, C is an $m \times n$ matrix, and 0 is an $m \times m$ matrix that consists entirely of zeros. Note that m is the number of z -coordinate degrees of freedom of the interface unknowns, and is much less than n . Subscript f denotes field variables, and i denotes interface variables. The entire matrix, J , is nonsingular. Using block Gauss elimination, the system of equations can be rewritten as

$$\begin{bmatrix} A & B \\ 0 & S \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_i \end{bmatrix} = \begin{bmatrix} -r_f \\ r_S \end{bmatrix} \tag{6.7}$$

where $S \equiv -CA^{-1}B$ is the Schur complement, and $r_S \equiv -r_i + CA^{-1}r_f$ is its modified right-hand-side. Block back-substitution is used to first solve for the interface update vector δ_i , and then for the field update vector δ_f

$$S\delta_i = r_S \tag{6.8}$$

$$A\delta_f = -r_f - B\delta_i. \tag{6.9}$$

The Schur complement, S , and its right-hand-side, r_S must be determined to solve

Equation (6.8). Let's begin with the modified right-hand side, r_S :

$$r_S \equiv -r_i + CA^{-1}r_f. \quad (6.10)$$

The main difficulty in computing r_S is the inverse of matrix A . On close inspection, we see that instead of just A^{-1} we need $A^{-1}r_f$. In other words, we need to solve the linear system of equations

$$Az = r_f, \quad (6.11)$$

for the vector z , an n -dimensional vector that serves as a “placeholder”. The modified right-hand side is then easily calculated as

$$r_S = -r_i + Cz. \quad (6.12)$$

Next, we compute the Schur complement matrix

$$S \equiv -CA^{-1}B. \quad (6.13)$$

In the Schur complement, we have the action of the inverse of A , not on a vector, but on matrix B . We can use a similar technique here as we used for Equation (6.11). Here we solve m systems of linear equations of the form

$$AY = B, \quad (6.14)$$

for Y , an $(n \times m)$ placeholder matrix. The Schur complement can then be calculated by

$$S = -CY \quad (6.15)$$

and the interface update vector, δ_i , is determined by solving Equation (6.8) which is a simple $(m \times m)$ system of linear equations. The series of calculations to solve for the Schur complement, S and its right-hand side, r_S requires $m + 1$ linear solves that involve matrix A .

The whole scheme is completed by computing the field update vector, δ_f . We formally rearrange Equation (6.9) to,

$$\delta_f = -A^{-1}r_f - A^{-1}B\delta_i, \quad (6.16)$$

Algorithm 6.1 - Schur complement formulation, general approach

- I. Solve $S\delta_i = r_S$ for δ_i
 - A. Compute modified right-hand side $r_S = -r_i + CA^{-1}r_f$
 - 1. Solve $Az = r_f$ for z , save vector z for later use
 - 2. Compute $r_S = -r_i + Cz$
 - B. Compute Schur complement $S = -CA^{-1}B$ and solve for δ_i
 - 1. Solve $AY = B$ for matrix Y , save matrix Y for later use
 - 2. Form $S = -CY$
 - 3. Solve $S\delta_i = r_S$ for δ_i
 - II. **Compute** $\delta_f = A^{-1}(-r_f - B\delta_i)$
 - A. $\delta_f = -z - Y\delta_i$
-

but we realize that the first term of the right-hand-side has already been computed as $z = A^{-1}r_f$ in Equation (6.11) and the second term has already been computed as $Y = A^{-1}B$ in Equation (6.14). The final step is then trivial to compute,

$$\delta_f = -z - Y\delta_i. \tag{6.17}$$

Note that the above approach is also well known as a bordering algorithm, where typically $m \ll n$ and the blocks B , C , and 0 are considered to “border” the matrix A [85]. If LU decomposition of A is possible, typically during the solve of Equation (6.11), the decomposition can be saved and reused to obtain the m additional solutions required by Equation (6.14). Algorithm 6.3.2 outlines the general procedure of solving Newton’s method using a Schur complement form of the Jacobian matrix.

LU decomposition of A is not always feasible, especially in the three-dimensional problems we are interested in solving. Therefore, an iterative solution method can be applied to the Schur complement formulation instead of a direct solver. An iterative solver can be substituted for a direct solver as the algorithm is written above. However, we see that we need the action of A on a vector, not its explicit inverse. A smarter way of incorporating an iterative solver with the Schur complement form can be devised as discussed below.

To see how the procedure changes using an iterative solver, let’s examine the definition of the modified right-hand-side, $r_S \equiv -r_i + CA^{-1}r_f$. Again, we use a placeholder vector $z \equiv A^{-1}r_f$, and now solve Equation (6.11) iteratively for z and use its solution to form r_S .

Next, Equation (6.8), $S\delta_i = r_S$ is solved iteratively with the newly form right-hand side r_S . A suitable candidate solution for δ_i is constructed from the Krylov subspace defined by $K = \{v, Sv, S^2v, \dots\}$ where v is the initial Krylov vector, in this case set to $v = r_S$. The Schur matrix must still be determined to construct the Krylov subspace. Let's examine the definition of S multiplied by the initial Krylov vector v

$$Sv = -CA^{-1}Bv. \quad (6.18)$$

We can solve Equation 6.18 as outlined in the general procedure where we use a placeholder matrix Y , where $AY = B$, which is then multiplied with v . However, this method still requires m solves of an $(n \times n)$ system of equations. We can reduce this to one solve of the $(n \times n)$ system of equations by using a new placeholder vector, w , defined as

$$w \equiv Bv \quad (6.19)$$

that is calculated by a simple matrix vector multiplication of B with the Krylov vector v . The n -dimensional placeholder vector w can viewed as the new right-hand-side of the $(n \times n)$ system of equations instead of matrix B . The equation becomes

$$Ay = w \quad (6.20)$$

where we now solve for the placeholder $(n \times 1)$ vector y instead of the $(n \times m)$ placeholder matrix Y , saving $m - 1$ solves of the large $(n \times n)$ system of equations. The vector y is solved for using an iterative solver in the same manner the placeholder vector z in the modified right-hand-side, r_S .

Finally, we multiply the matrix C by the vector y to get the $(m \times 1)$ placeholder vector u

$$u = -Cy = -CA^{-1}Bv \equiv Sv. \quad (6.21)$$

This procedure is repeated k times until enough Krylov vectors are attained for satisfactory solution of Equation (6.8) and δ_i .

The field variables are updated by solving Equation (6.9) iteratively for δ_f . We again define a $(n \times 1)$ placeholder vector x where

$$x = -r_f - B\delta_i, \quad (6.22)$$

Algorithm 6.2 - Schur complement formulation with iterative solver

- I. Solve $S\delta_i = r_S$ for δ_i
 - A. Compute $r_S = -r_i + CA^{-1}r_f$
 1. Solve $Az = r_f$ for z using iterative solver
 2. Compute $r_S = -r_i + Cz$
 - B. Iteratively solve $S\delta_i = r_S$, where $S = -CA^{-1}B$
 1. Form $w = Bv$ where v is some arbitrary vector
 2. Solve $Ay = w$ for vector y using iterative solver
 3. Form $u = -Cy$
 4. Repeat Step B. until we satisfy $u = S\delta_i = r_S$.
 - II. Solve $A\delta_f = -r_f - B\delta_i$ for δ_f
 - A. Form $x = -r_f - B\delta_i$
 - B. Solve $A\delta_f = x$ for δ_f using iterative solver
-

and we finally solve

$$A\delta_f = x \tag{6.23}$$

using an iterative solver.

Note that this new procedure to solve Equation (6.8)–Equations (6.18)–(6.21)–requires two nested calls to the iterative solver which we label as the outer and inner iterations. The *outer* iteration is where the actual candidate solution for δ_i is constructed from the Krylov subspace $K = \{v, Sv, S^2v, \dots\}$ through repeated applications of Equation (6.18), Step I.B. in Algorithm 6.3.2. The *inner* iteration is due to solving the $(n \times n)$ system of equations in Equation (6.20) for the placeholder vector y , Step I.B.2. in Algorithm 6.3.2. The Generalized Minimal Residual (GMRES) of Saad and Schultz [151] is used as the iterative solver in the results shown here.

Stepping back, we find that we may be able to solve the entire problem using $k+2$ solves of n -dimensional systems. One solve of Equation (6.11) is needed to form the right-hand side of Equation (6.8). There are k solves of Equation (6.20) needed to form the Krylov space needed to solve Equation (6.8) for δ_i . The last solve of an n -dimensional system comes from solving Equation (6.9) for δ_f . The success of this iterative strategy depends on the amenability of Equation (6.8) toward iterative solution, namely, does $CA^{-1}B$ have

a suitable form for use with an iterative solution technique. Algorithm 6.3.2 outlines the steps taken when solving the Schur complement formulation using an iterative solution technique.

6.3.3 Approximations and Preconditioning

The majority of the computational effort comes from solving the n -dimensional system of equations involving matrix A . Recall matrix A is the block Jacobian matrix of all the field residual equations plus the equations governing the motion of the deforming mesh. To reduce the required computational effort, a much coarser, but cheaper, approximation of A can be made. Here, we try two approximations, $A \approx \tilde{A}$: $\tilde{A} = \text{diag}(A)$ where the original matrix is replaced by a matrix containing only the diagonal elements of A , and $\tilde{A} = I$ where the original matrix is replaced by the identity matrix. The inverse of the former approximation is trivial to compute, just the inverse of each of the diagonal elements. The latter approximation is equivalent to the former if diagonal scaling is conducted on the original equation set.

Matrix A appears three times in the Schur complement formulation. The first in the modified right-hand-side, $r_S = -r_i + CA^{-1}r_f$, the second in the Schur complement $S \equiv -CA^{-1}B$, and the third in the equation for the field unknown update vector Equations (6.9). The approximation $\tilde{A}$ can be substituted for A in any of these occurrences. We will approximate A only in the modified right-hand-side and in the Schur complement. Matrix A is critical to correctly computing the update vector δ_f , and will not be simplified in this step.

Substituting $\tilde{A}$ for A in Equation (6.10) changes the equation to

$$r_S \equiv -r_i + CA^{-1}r_f \approx -r_i + C\tilde{A}^{-1}r_f. \quad (6.24)$$

For $\tilde{A} = I$, multiplication by $\tilde{A}^{-1}$ can be skipped altogether to get:

$$r_S \equiv -r_i + CA^{-1}r_f \approx -r_i + Cr_f. \quad (6.25)$$

The new, approximate right-hand-side, $\tilde{r}_S$, is formed using a relatively cheap matrix-vector product in place of a linear solve of the $(n \times n)$ system of equations.

A much more significant reduction in effort would result from the same approximation of A applied to the formation of the Schur complement S , which would eliminate the k solves needed to form Krylov vectors. The substitution of $A \approx \tilde{A}$ in Equation (6.13) results

in

$$\tilde{S} \approx -C\tilde{A}^{-1}B \quad (6.26)$$

With these substitutions, the matrix S and right-hand-side vector r_S are completely known and Equation (6.8) can be solved immediately.

Another method of reducing the number of outer iterations is to develop a preconditioner for the Schur complement. In order to achieve real improvements in performance, an expensive preconditioner is needed. The ideal preconditioner is S^{-1} , but this is unknown. However, an approximation of S^{-1} can be used, similar to how A is approximated above. For our preconditioner we use:

$$\tilde{S}^{-1} = -[B(\text{diag}(A))^{-1}C]^{-1} \quad (6.27)$$

where we are again approximating A with its diagonal matrix, whose inverse is cheap to compute. The overall preconditioner $\tilde{S}$ can be computed using a series of matrix multiplications.

Here we use $\tilde{S}$ as a left preconditioner, giving

$$\tilde{S}^{-1}S\delta_i = \tilde{S}^{-1}r_S \quad (6.28)$$

as the equation to solve with GMRES, instead of Equation (6.8). The Krylov subspace becomes $K = \{v, \tilde{S}^{-1}Sv, \dots\}$, where the initial Krylov vector v is now defined as $v \equiv \tilde{S}^{-1}r_S$ [150].

The inverse of $\tilde{S}$ also has to be calculated. This can be done using either a direct or an iterative solver. Here, we use an iterative solver, as it's the method most likely used when scaling the method to three-dimensional problems. To use an iterative solver, we use an additional "inner" iteration using GMRES. Let's assume we have completed the matrix multiplication Sv , and define the $(m \times 1)$ placeholder vector $t \equiv Sv$. Next, we solve the m -dimensional system of equations

$$\tilde{S}q = t \quad (6.29)$$

where q is another $(m \times 1)$ placeholder vector. The procedure is repeated until sufficient number of Krylov vectors are accumulated for a satisfactory solution to Equation (6.28).

Using a preconditioner on the Schur complement adds an additional solve of an $(m \times m)$ system of equations. For a direct solver, this means LU decomposition of an m -dimensional equation set. With an iterative solver, we have $l < m \ll n$ additional iterations of an m -

Table 6.1: Comparison of L_2 norm of residual vector as a function of Newton iteration of the direct frontal solver and the general Schur complement formulation method. Both methods converge quadratically in three Newton iterations.

Newton Iteration	Direct solver	General Schur method
1	1.034×10^{-1}	1.034×10^{-1}
2	4.934×10^{-3}	4.934×10^{-3}
3	2.047×10^{-6}	2.047×10^{-6}

dimensional system of equations per outer iteration. Bangerth et al. [9] use a similar preconditioner to solve the Navier-Stokes equations, and saw a reduction in number of outer iterations.

6.4 Results and Discussion

A simple two-phase, heat conduction problem is used to test the Schur complement formulation as a method of solving a free interface problem. The physical properties of the two-phases are based on the melt and solid phases of cadmium zinc telluride given in Table 2.1, a material of interest in our other research, for a value of $\kappa = 0.5$. The dimensionless heat transfer coefficient is set to $h = 10$. The degree of complexity of the problem is controlled by the number of elements used to discretize the domain. We use GMRES as the iterative solver with the Schur complement formulation.

To test the viability of solving a Newton step using the Schur complement formulation, we first implement the general procedure laid out in Algorithm 6.3.2. Each step in the procedure is solved directly, and full LU decomposition is preformed on both matrix A and matrix S . We compare the number of Newton steps required to reach convergence between that of our standard, direct frontal solver and that of the general Schur complement formulation on a (20×20) element mesh where the elements are divided equally between the two sub-domains. We set our convergence requirements to when both the L_2 norms of the residual vector and the update vector relative to the solution vector are less than 1×10^{-4} . Both methods converge in a quadratic fashion in three Newton iterations. Table 6.1 compares the residual norm at each Newton step for the two methods. The general Schur complement method has the same residual norms as the direct frontal solver, to four significant digits, indicating that solving a Newton step in this fashion does not change the overall solvability of the problem and is equivalent to the direct solver.

In the following sections, we compare three different solution methods for solving a

Newton step. Method 1 (M1) is our standard, direct frontal solver which solves the field and mesh equations simultaneously. M1 is taken as the control in our tests. Solution of the coupled governing equations by GMRES is labeled Method 2 (M2). Method 3 (M3) is the solution of the Schur complement formulation in an iterative manner. As above, the solution is deemed converged when both the L_2 norms of the residual vector and the update vector relative to the solution vector are less than 1×10^{-4} . The iterative solution method GMRES has two convergence criteria: maximum number of Krylov vectors (k_{max}) and the norm of the residual vector, $\|Ax_n - b\|$, is less than a certain tolerance (ϵ) [150]. Depending on the test, either k_{max} or ϵ will be the comparison factor. The rows of the Jacobian matrix are scaled with respect to the L_2 row norm when either M2 or M3 are used. In the case of M3, row scaling is applied before diving the Jacobian into the subblocks. These methods do not converge unless row scaling is applied.

Three tests are enacted to compare the performance of the four solution methods. In the first test, we examine the rate of convergence of the three methods, in which the total number of Newton iterations to reach a converged solution are studied. A hybrid Schur method is also examined, as explained below. The second test compares the amount of computational effort required by the direct solver (M1), GMRES (M2), and the Schur method with GMRES (M3) to perform one Newton step, and how these methods scale with problem size. The final test appraises the effectiveness of the approximations and preconditioners, discussed in the previous section, in improving the performance of M3. The tests were performed on a PC with 4 Core Duo Intel Xeon processors, each subprocessor at 2.66 GHz, and 16 GB of RAM.

6.4.1 Comparison of Convergence Rate

As stated above, we propose that a good approximation of the interface unknowns is more critical than the approximation of the field unknowns for Newton's method to converge no matter what method is used to solve the Newton step. The purpose of solving a Newton step using a Schur complement form of the problem is to map out the difficult part of the problem (the interface location) and solve it separately. Therefore, we expect the convergence properties of the Schur complement method, M3, to be superior to that of GMRES, M2.

To show this, we use a (20×20) element mesh over the computational domain. Bi-quadratic basis functions are used to discretize the heat conduction equation. The mesh equations are discretized using bilinear basis functions within the domain, and with bi-

quadratic basis functions along the boundaries. The problem has 2763 total unknowns of which 1681 are the temperature (field) unknowns, and 1081 are mesh unknowns. When divided into block form, there are $m = 41$ interface unknowns, and $n = 2722$ mesh and temperature unknowns.

We compare the convergence behavior of our standard frontal solver (M1) with that of solving the entire system using GMRES (M2) and with an iterative solver (M3) by examining the total number of Newton iterations needed to reach a converged solution by the respective solvers. In addition, we examine the convergence behavior of a hybrid version of M3 where Equation (6.8) is solved exactly using a direct method and Equation (6.9) is solved iteratively. With this hybrid version, we will be able to separate the dependency of the Newton's method on the interface unknowns and the field unknowns.

The performance of M2 depends on the convergence of GMRES solving Equation (6.5), while the performance of M3 depends on the convergence of GMRES in both the outer and inner iterations. The effect of GMRES convergence on M2 is examined by varying the value of k_{max} (k_G). In addition, to study how convergence of the outer and inner iterations affect M3, we set k_{max} of the outer (k_o) and inner iterations (k_i) as our variable convergence parameters.

Figure 6.2 is a comparison of the number of Newton iterations required for the solution to reach convergence against the number of Krylov vectors. In the case of M3 this is the number of inner Krylov vectors, k_i . The direct frontal solver, M1, converges quadratically in three Newton iterations. As expected, the number of Krylov vectors used by the iterative solver affected the overall rate of convergence of Newton's method. M2, GMRES, required ≥ 181 Krylov vectors to achieve quadratic convergence, and failed at ≤ 54 Krylov vectors. Whereas, the hybrid Schur method, M3-hybrid, need ≥ 110 Krylov vectors for quadratic convergence and failed at ≤ 10 Krylov vectors. Also, for a given number of Krylov vectors < 180 , the hybrid-Schur method converged in less number of Newton iterations than GMRES. Overall, the hybrid-Schur method needed approximately half the number of Krylov vectors than GMRES for the solution to converge. This confirms our hypothesis that a good approximation of the interface unknowns is necessary for Newton's method to converge satisfactorily.

The effect of the number of outer Krylov vectors, k_o , is also examined in Figure 6.2 for the cases of $k_o = 10$ and $k_o = 40$. When $k_o = 40$, the Krylov subspace size for the outer problem is the same size (including the initial vector) as the Schur complement problem $m = 41$. For $k_o < 10$, M3 did not converge. Comparing the results for $k_o = 10$ and $k_o = 40$, we see that the additional number of outer Krylov vectors did not effect the

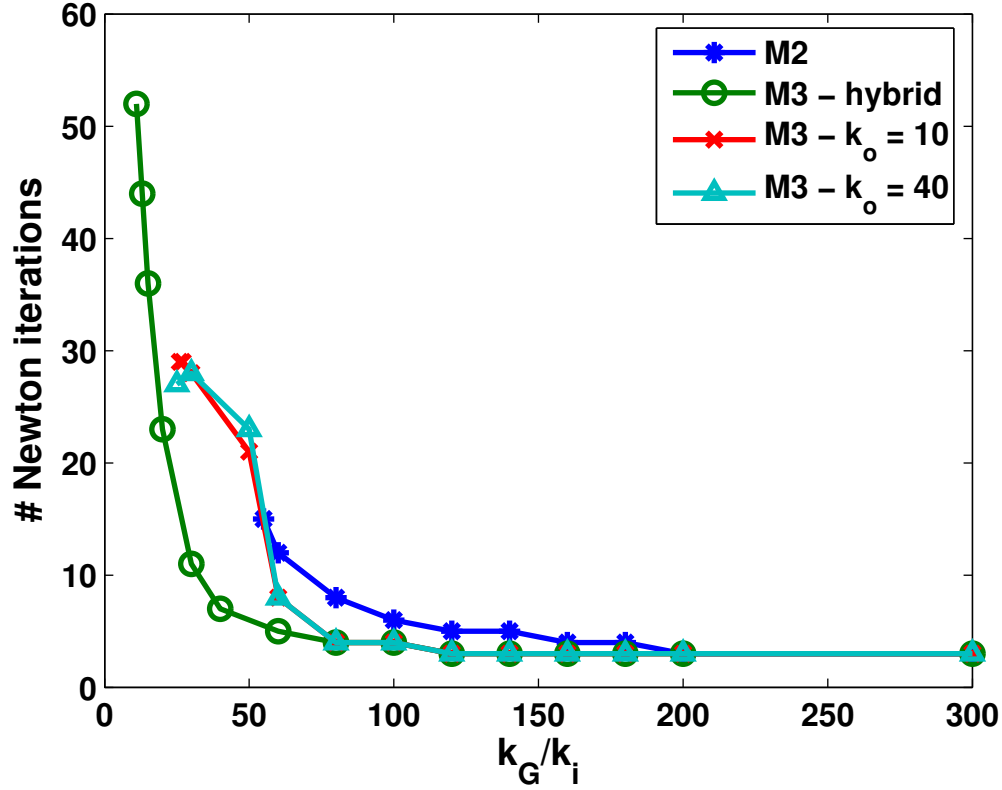


Figure 6.2: Comparison of number of Newton iterations required to reach convergence as a function of number of (inner) Krylov vectors.

overall convergence of M3. Therefore, the approximation of the interface unknowns less dependent on the number of outer iterations than the number of inner iterations, or the approximation of the Schur complement matrix. M3 also out-performed GMRES in both cases at a given number of Krylov vectors < 180 , and matched the performance of the hybrid-Schur method for $k_i \geq 60$. To note, when $k_i = 40$ in the cases of $k_o = 10$ or 40 , M3 either failed to converge or took > 200 Newton iterations to converge, but would converge for values of $25 \leq k_i < 40$. The hybrid-Schur method, however, converged in seven Newton iterations with $k_i = 40$. This anomaly is due to the nature of iterative solvers. The minimum number of Krylov vectors GMRES (M2), the hybrid version of the Schur complement method (M3, hybrid), and the iterative Schur method for $k_o = 10$ and 40 (M3, $k_o = 10$ and M3, $k_o = 40$) needed for Newton's method to converge and to converge quadratically are listed in Table tab:minK.

Table 6.2: The minimum number of (inner) Krylov vectors needed for Newton’s method to converge to the solution and to converge quadratically.

	M2	M3, hybrid	M3, $k_o = 10$	M3, $k_o = 40$
Solution to converge	55	11	26	25
Quadratic convergence	181	108	110	108

While this test problem is fairly simple and coarse, it is still revealing of the behavior of the Schur complement formulation. As shown above, solving the Schur complement form with an exact direct method works as well as our frontal solver in computing a solution. Unlike our experience in solving free-boundary problems in three dimensions with GMRES, the solution of the two-dimensional problem was able to be computed with GMRES (M2) with readily available computing resources. However, we show with the hybrid-Schur method that the approximation of the interface unknowns is more important than that of the field unknowns for Newton’s method to converge. By segregating the problem, we can focus our resources on solving the more difficult and critical portion of the problem—the interface location. The iterative version of the Schur complement formulation (M3) also converged with less number of Newton iterations for a given number of Krylov vectors than GMRES (M2). Furthermore, other than a minimum number of Krylov vectors necessary for a successful iteration, the rate of convergence, M3 does not seem to depend on k_o . More important to convergence rate is the number of inner Krylov vectors, k_i , or the accuracy of computing the action of A^{-1} on a vector. This interplay between accurately resolving the Schur complement matrix and computing the update vector δ_i is potentially quite costly in terms of computation time. In the next section we examine how the computational work of the different methods scale with problem size.

6.4.2 Scaling of Computational Effort

Not only is rate of convergence important, but also the cost of reaching a converged solution. We compare the computational effort required to solve Equation (6.5), or one Newton iteration, as a function of problem size using the frontal solver (M1), GMRES on the entire system of equations (M2), and the Schur complement formulation using an iterative solver (M3).

In this test, three different mesh sizes are examined: (20×20) , (40×40) , and (80×80) elements. We set the variable N_{el} as the number of elements in the r -direction, i.e., for a mesh of (20×20) elements, $N_{el} = 20$. Biquadratic basis functions are used for both

Table 6.3: Description of computational meshes used to determine the scaling dependencies of methods M1, M2, and M3 on problem size. N_{el} is the number of elements in the x -direction, $N_{nodes} = 2N_{el} + 1$ is the number of nodes in the x -direction, and $N_{total} = 3N_{nodes}^2$. Parameter f is the front width used by solver M1, n is the number of temperature and mesh unknowns, and m is the number of interface unknowns. N_Z is the total number of nonzero elements in the Jacobian matrix, while A_Z , B_Z , and C_Z are the number of nonzero elements of matrices A , B , and C respectively.

Mesh Size	20×20	40×40	80×80
N_{el}	20	40	80
N_{nodes}	41	81	161
N_{total}	5,043	19,683	77,763
f	195	375	735
n	5,002	19,602	77,602
m	41	81	161
N_Z	175,729	709,127	2,850,800
A_Z	173,197	704,042	2,840,590
B_Z	2,210	4,443	8,8928
C_Z	322	642	1,282

the heat transfer and mesh equations for more simplified accounting of the problem size. Therefore, the number of nodes in the r -direction is $N_{nodes} = 2N_{el} + 1$, and the total number of degrees of freedom is $N_{total} = 3N_{nodes}^2$ (a T , r , and z degree of freedom at each node) for the meshes considered here. We use parameter N_{nodes} as our measure of problem size. Table 6.3 lists the relevant problem size parameters of the different meshes.

Each solver scales differently with problem size, the all have specific solver parameters that also affect scaling. The total work of M1 to solve a N_{total} -dimensional problem, W_{M1} , scales as

$$W_{M1} \propto N_{total} f^2 \quad (6.30)$$

for large problems, where f is the front width of the Jacobian matrix [54]. In the meshes used here, $f \propto N_{nodes}$, which gives

$$W_{M1} \propto N_{nodes}^4 \quad (6.31)$$

for the relationship between W_{M1} and N_{nodes} .

The computational effort of GMRES (M2) not only depends on problem size, N_{nodes} , but also the number of nonzero elements in the Jacobian matrix, N_Z , and the number of Krylov vectors that are used to approximate the solution k_G . For the version of

GMRES used here, we have (a) a normalization of a vector with $4N_{total} + 1$ operations (b) a sparse matrix-vector product that takes $2k_G N_Z$ operations, (c) Householder vector generation and orthognolization which takes $2k_G^2 N_{total}$ operations, (d) Givens rotation of $3k_G^2 + 2k_G N_{total} + 12k_G$ operations, (e) normalization of vector with $k_G N_{total} + k_G$ operations, and (f) calculation of the solution using $k_G^2 + 2k_G N_{total} + 3k_G + 1$ operations. An additional $3N_Z + N_{total}$ operations are needed to scale the Jacobian matrix before starting GMRES. The total work for M2, W_{M2} can be rewritten in terms of N_{nodes} , k_G , and N_z :

$$W_{M2} = (2k_G + 3)N_Z + (2k_G^2 + 5k_G + 5)N_{total} + 4k_G^2 + 16k_G + 2 \quad (6.32)$$

$$= (2k_G + 3)N_Z + (6k_G^2 + 15k_G + 15)N_{nodes}^2 + 4k_G^2 + 16k_G + 2 \quad (6.33)$$

From this, we see that for large problems, W_{M2} scales as

$$W_{M2} \propto k_G^2 \quad (6.34)$$

and

$$W_{M2} \propto N_{nodes}^2 \quad (6.35)$$

for large problems.

Calculating the total work of M3 is more involved than more the previous methods; and depends on the dimensions n and m , the number of nonzero elements in matrices A , B , and C (A_Z , B_Z , and C_Z respectively), and the maximum number of Krylov vectors used in the outer and inner iterations, k_o and k_i . The addition of the dimensions n and m equal the total number of unknowns, $N_{total} = n + m$. Similarly, the number of nonzero elements of the submatrices is equal to the total number of nonzero entries in the Jacobian matrix, $N_Z = A_Z + B_Z + C_Z$.

Let's return to Algorithm 6.3.2 and look at each step in turn. We begin with Step IA, the computation of the modified right-hand-side, r_S . In this step, $Az = r_S$ is solved iteratively using GMRES with an operation count as defined above. In addition, there is (a) a sparse matrix-vector product of $2C_z$ operations, and (b) an m -dimensional vector subtraction that takes $2m$ operations. The total work for this step, W_r is

$$W_r = 2k_i A_Z + (2k_i^2 + 5k_i + 4)n + 4k_i^2 + 16k_i + 2 + 2C_Z + 2m. \quad (6.36)$$

Step IB, solving the Schur complement problem $S\delta_i = r_S$, has both an outer and inner

GMRES iteration where the inner iteration takes the place of the matrix-vector product Sv in the outer iteration. First there is (a) a sparse matrix-vector product of $2B_Z$ operations followed by (b) the iterative solution by GMRES of the n -dimensional system, $Ay = w$, with an operation count as defined for W_{M2} . We finish with (c) a sparse matrix-vector product of $2C_Z$ operations, and (d) a scalar multiplication of an m -dimensional vector for m operations. Operations (a)-(d) are repeated k_o times. Additional operations counts are due to solving the outer iteration by GMRES. This gives a total work, W_S of :

$$W_S = \{2B_Z + 2k_i A_Z + (2k_i^2 + 5k_i + 4)n + 4k_i^2 + 16k_i + 2 + 2C_Z + m\}k_o + (2k_o^2 + 5k_o + 4)m + 4k_o^2 + 16k_o + 2 \quad (6.37)$$

The quantity with the brackets, $\{\cdot\}$, is substituted in place of the matrix-vector product Sv of the outer iteration.

Finally, in Step II, the calculation of the field vector update, we have (a) a matrix-vector product of $2B_Z$ operations, and (b) a scalar multiplication of an n -dimensional vector with a vector subtraction for $2n$ operations. We finish with an iterative solve of equation $A\delta_f = x$, again, whose operation count is defined as for W_{M2} , for a total work, W_f , of

$$W_f = 2B_Z + 2n + 2k_i A_Z + (2k_i^2 + 5k_i + 4)n + 4k_i^2 + 16k_i + 2. \quad (6.38)$$

The total work of method M3 is then

$$W_{M3} = W_r + W_S + W_f + 3N_Z + N_{total} \quad (6.39)$$

including the $3N_Z + N_{total}$ operations required to scale the Jacobian matrix before applying the solution technique.

For a large problem, the total work scales with k_i as

$$W_{M3} \propto k_i^2. \quad (6.40)$$

W_{M3} also looks to depend on the square of the number of outer Krylov vectors, k_o . However, the term in Equation (6.37) enclosed in brackets accounts for a much greater number of operations than the number k_o , $\{\cdot\} \gg k_o$ giving a linear dependency of work on k_o :

$$W_{M3} \propto k_o. \quad (6.41)$$

The total work of M3 also depends linearly on submatrix dimension n , but n depends on the total problem size. The submatrix dimension m is equal to the number of nodes in the x -direction, or $m = N_{nodes}$. Therefore, submatrix dimension n is then $n = N_{total} - m = 3N_{nodes}^2 - N_{nodes} \propto N_{nodes}^2$. Therefore, the total work of the iterative Schur complement method scales as

$$W_{M3} \propto N_{nodes}^2. \quad (6.42)$$

Scaling tests were conducted by timing the amount of time a 4-duo core, 8 processor PC at x MHz per processor with 16 GB of RAM, 2 GB dedicated to each processor took to compute one Newton step by each of the methods M1, M2, and M3. No other operations were run on the computer during the timing tests. As noted above, the operation count of M2 depends on both N_{nodes} and k_G . To show the dependency of W_{M2} on N_{nodes} , the number of Krylov vectors was fixed at $k_G = 800$. Similarly, the number of nodes was fixed at $N_{nodes} = 161$ to time the dependency on k_G for values of $k_G = 600, 800$, and 1000 . The total work of method M3 depends on N_{nodes} , k_o , and k_i . Three series of tests were conducted on M3 to show how these dependencies scale. First, the number of outer and inner Krylov vectors were fixed at $k_o = 20$ and $k_i = 200$ while the mesh size was varied. In the second test, we set $k_o = 20$ and used a mesh size of $N_{nodes} = 161$ while varying $k_i = 400, 600$, and 800 . For the final test, the number of inner Krylov vectors was set to $k = 200$ on a mesh of $N_{nodes} = 161$ while varying $k_o = 60, 70$, and 80 . For both M2 and M3, the convergence tolerances of the iterative solvers, ϵ , were fixed such that the number of Krylov vectors was the only governing convergence criterion.

Figure 6.3 plots the log of operation time versus the log of N_{nodes} for the three methods. From the slope of the linear fit lines, it can be seen that the direct, frontal solver scales $W_{M1} \propto N_{nodes}^4$ as calculated in Equation (6.31). While both iterative solution methods, M2 and M3, took more time to solve a Newton step than M1, they both scale with the square of the problem size, or N_{nodes} . Eventually, methods M2 and M3 will take less time than M1 to solve a Newton step as the problem size is increased.

Figure 6.4 shows how M2 and M3 depend on the number of Krylov vectors used by GMRES. The total work of M2 increases with the square of the number of Krylov vectors, k_G . Similarly, M3 increases with the square of the number of inner Krylov vectors, k_i . Furthermore, M3 scales linearly with the number of outer Krylov vectors, k_o , as discussed above.

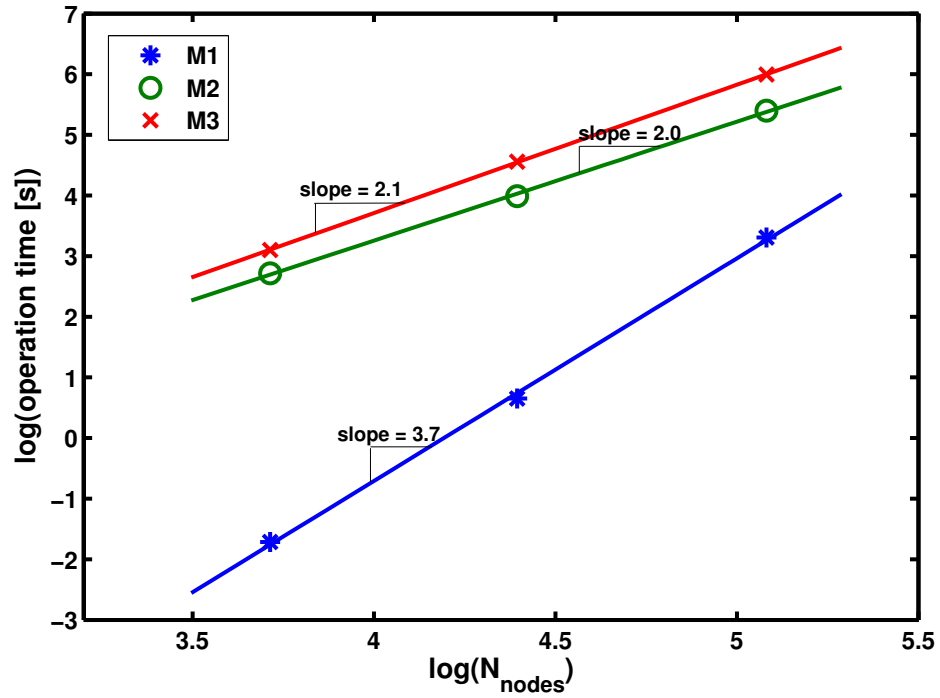


Figure 6.3: Plot of $\log(\text{operation time})$ versus $\log(N_{nodes})$ for methods M1, M2, and M3. The direct solver, M1, scales at a power of ≈ 4 with respect to N_{nodes} , while the two iterative methods, M2 and M3, scale at a power of 2.

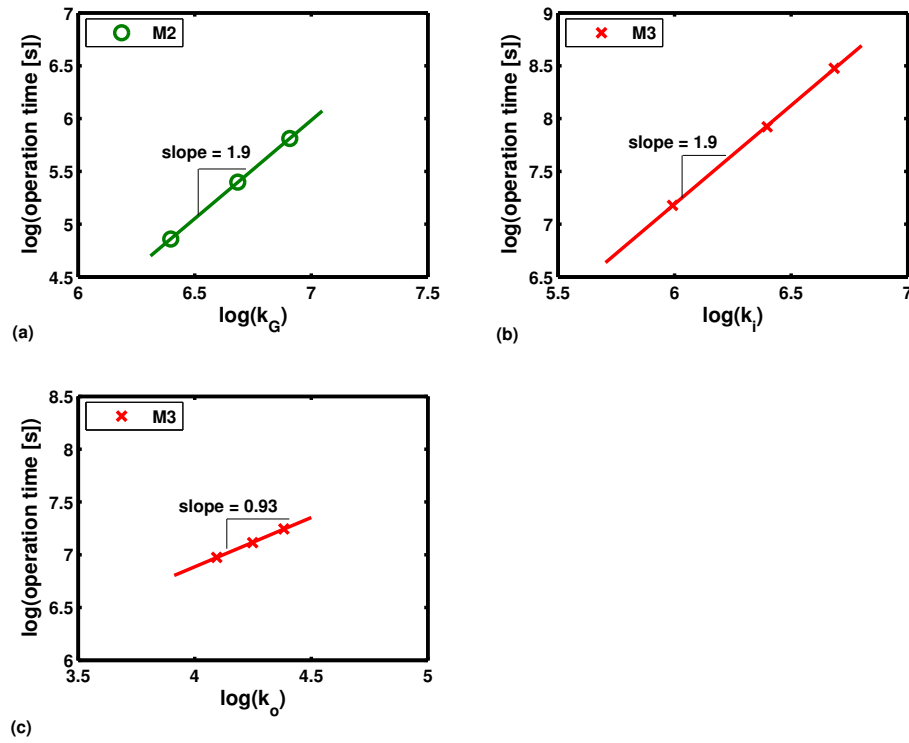


Figure 6.4: Scaling dependency of (a) method M2 on k_G , (b) method M3 on k_i , and (c) method M3 on k_o . GMRES, M2, scales with the squared of the number of Krylov vectors, k_G . The iterative Schur method, M3, scales with the square of the number of inner Krylov vectors, k_i , and linearly with the number of outer Krylov vector, k_o .

Scaling of Three-dimensional Problems

The scaling analysis performed on the two-dimensional problems can be extended to encompass three-dimensional problems to provide insight on how these methods perform on such models. We base the three-dimensional scaling argument on a mesh of $(N_{nodes} \times N_{nodes} \times N_{nodes})$ in dimension. The total number of degrees of freedom then scales as $N_{total} \propto N_{nodes}^3$.

Estimating the amount of work it takes for M1, the direct frontal solver, to solve one Newton iteration of a three-dimensional problem is straightforward. The relationship presented in Equation (6.30) still holds, except now $N_{total} \propto N_{nodes}^3$ and $f \propto N_{nodes}^2$ which gives

$$W_{M1} \propto N_{nodes}^7 \quad (6.43)$$

for three-dimensional problems.

Method M2, or GMRES, depends on the number of Krylov vectors used to approximate the solution to a Newton step and the total number of unknowns. From the leading term of Equation (6.32), we have the relationship

$$W_{M2} \propto k_G^2 N_{total}. \quad (6.44)$$

The number of Krylov vectors, k_G , depends on the total number of unknowns, N_{total} ,

$$k_G \propto N_{total}^\alpha \quad (6.45)$$

where α is some geometric scaling factor relating k_G to N_{total} . In order to determine α , some criterion must be placed on k_G as, in a sense, it can be arbitrarily chosen. We choose to evaluate the scaling analysis with k_G as the minimum number of Krylov vectors needed for the Newton iterations to converge. This value for k_G was found for different, two-dimensional mesh sizes listed in Table 6.4. From this data, the value of α was found to be 0.49, see Figure 6.5(a). This relationship is still valid for three-dimensional problems, as it relates k_G to N_{total} with no assumptions on meshing. Therefore, W_{M2} as a function of N_{total} is

$$W_{M2} \propto N_{total}^{2\alpha+1} \quad (6.46)$$

Table 6.4: Minimum number of Krylov vectors required by methods M2 and M3 for Newton's method to converge.

Mesh	N_{nodes}	N_{total}	k_G	k_i	k_o
20×20	41	2,763	55	26	10
40×40	81	10,323	174	51	16
60×60	121	22,683	177	78	15
80×80	161	39,843	229	105	16
100×100	201	61,803	284	132	17

which for two-dimensional problems becomes

$$\begin{aligned} W_{M2} &\propto N_{nodes}^{4\alpha+2} \\ &\propto N_{nodes}^{3.96} \end{aligned} \tag{6.47}$$

which is confirmed by plotting the amount of time for each case versus N_{nodes} in Figure 6.6 where the slope is 3.63. For three-dimensional problems, the dependency on N_{nodes} is

$$\begin{aligned} W_{M2} &\propto N_{nodes}^{6\alpha+3} \\ &\propto N_{nodes}^{5.94} \end{aligned} \tag{6.48}$$

where the exponent is less than that for W_{M1} , $5.94 < 7$, which makes M2 a better choice than M1 for three-dimensional problems.

Method M3 has dependencies on both the number of outer and inner Krylov vectors and the total number of unknowns with a leading term of

$$W_{M3} \propto k_o k_i^2 N_{total} \tag{6.49}$$

from Equation (6.37). The number of inner Krylov vectors scales with n , the number of field and mesh unknowns, which in turn is approximately the total number of unknowns $n \approx N_{total}$ to give

$$k_i \propto N_{total}^\beta \tag{6.50}$$

where β is some geometric scaling factor. The number of outer Krylov vectors scales with $m = N_{interface}$, the number of interface unknowns which, for the cases presented here, is equal to the number of nodes $m = N_{nodes}$ to give

$$k_o \propto N_{nodes}^\gamma \tag{6.51}$$

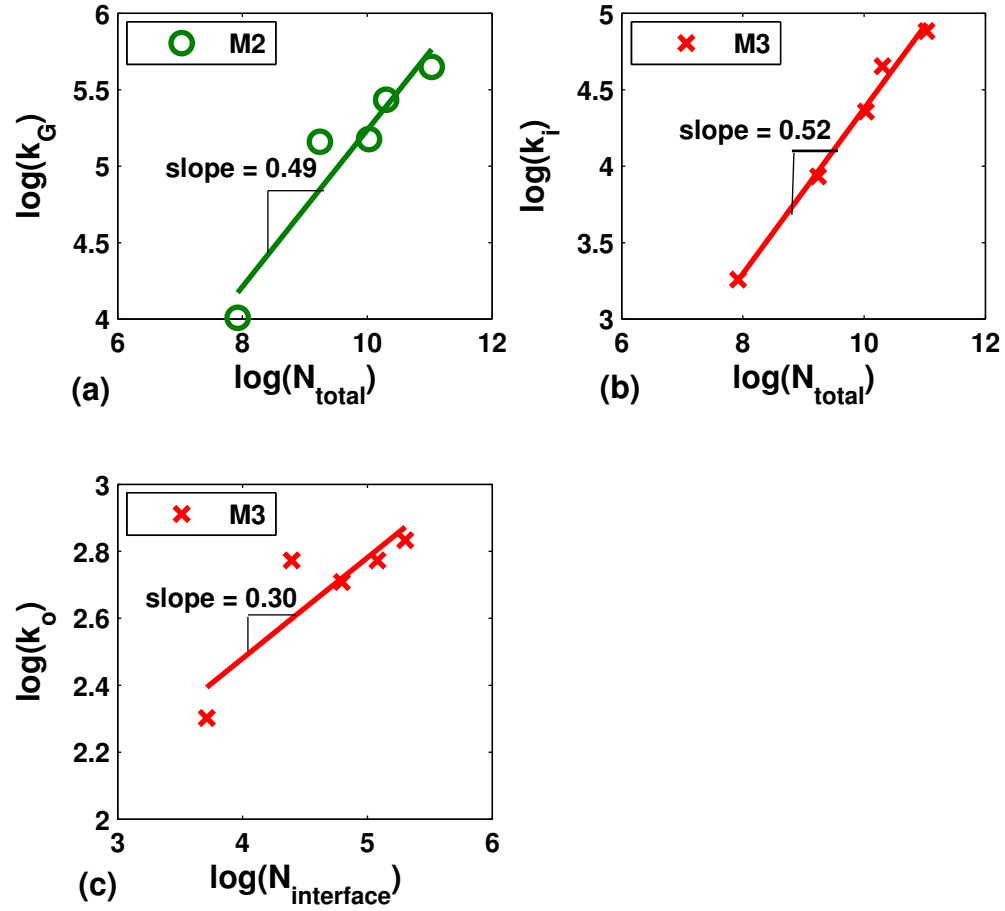


Figure 6.5: Scaling relationship between minimum number of (a) Krylov vectors used by M2 k_G , (b) inner number of Krylov vectors k_i with respect to N_{total} and (c) outer number of Krylov vectors k_o with respect to $N_{interface}$.

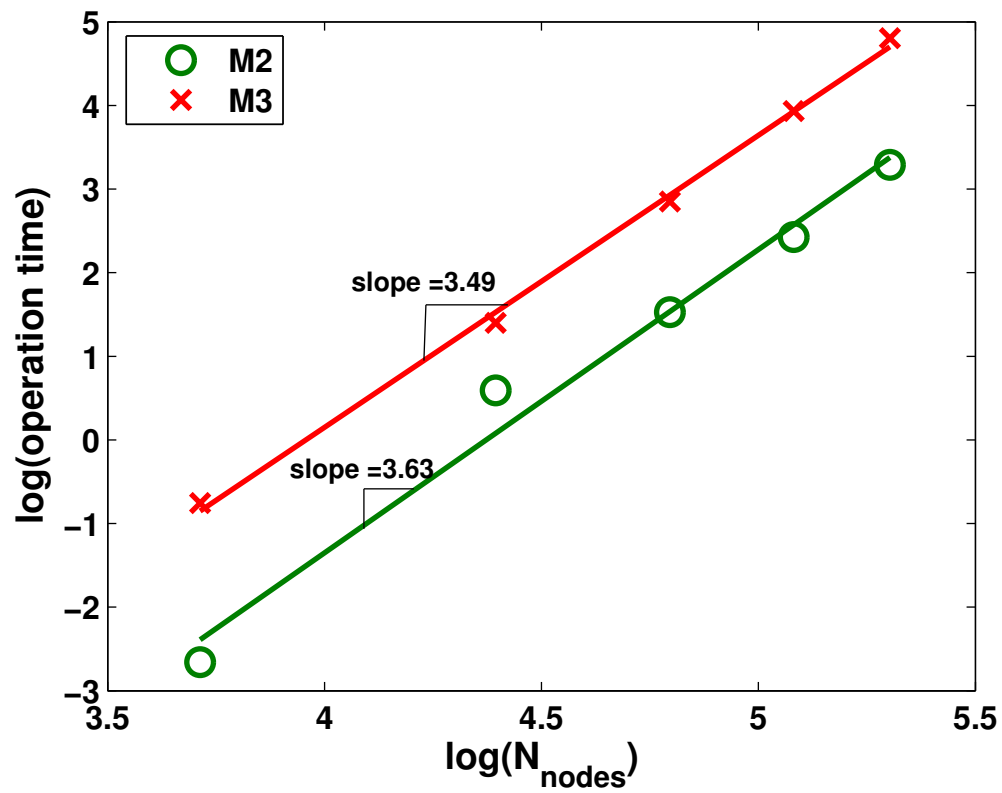


Figure 6.6: Scaling dependency of operation time with respect to N_{nodes} for M2 and M3 solving two-dimensional problems. The number of Krylov vectors for the two methods are chosen to be the minimum number required for Newton's method to converge.

where γ is some geometric scaling factor. For three-dimensional problems $m = N_{nodes}^2$. The values of k_i and k_o used to determine β and γ are chosen as the minimum number needed for the Newton iterations to converge, listed in Table 6.4. From this data, we found that $\beta = 0.52$ and $\gamma = 0.30$ as seen in Figures 6.5(b) and (c), respectively. Therefore, for two-dimensional problems, W_{M3} scales with N_{nodes} as

$$\begin{aligned} W_{M3} &\propto N_{interface}^\gamma N_{total}^{2\beta+1} \\ &\propto N_{nodes}^{4\beta+\gamma+2} \\ &\propto N_{nodes}^{4.38} \end{aligned} \tag{6.52}$$

however the timing tests predicted a relationship of $W_{M3} \propto N_{nodes}^{3.49}$, see Figure 6.6. Possibly even larger problem sizes are needed to see the predicted scaling relationship. Extending this analysis to three-dimensions gives

$$\begin{aligned} W_{M3} &\propto N_{nodes}^{6\beta+2\gamma+3} \\ &\propto N_{nodes}^{6.72} \end{aligned} \tag{6.53}$$

where the exponent is less than that for W_{M1} , $6.72 < 7$, but greater than the exponent for W_{M2} , $6.72 > 5.94$ so that it scales better than a direct solver but not as well as GMRES. The extra work for method M3 comes from the need to perform the outer iterations.

Interestingly, for large two-dimensional problems, the minimum number of outer Krylov vectors were approximately the same. There may possibly be a global minimum of outer Krylov vectors necessary to solve such problems, and may not contribute as much additional effort as even large problems are solved. Also, the (40×40) element mesh, or (81×81) nodal mesh, required a larger number of Krylov vectors for M2, k_G , and outer Krylov vectors k_o . This mesh size may represent a transition between "small" problems and "large" problems where the scaling relationships might be changing.

6.4.3 Effectiveness of Approximations and Preconditioning

While the iterative Schur complement formulation, M3, scales with problem size better than the frontal solver, M1, there is still room for improvement in the overall effort expended by the method. In this section, we present results for two ideas to reduce this effort: approximating the matrix A and applying a preconditioner to the Schur complement. The case studies are performed using a (40×40) element mesh that has 10,323 total number of unknowns of which 6,561 are temperature unknowns and 3,762 mesh unknowns with

$m = 81$ interface unknowns. Bilinear basis functions are used in the mesh equations. The rows of the Jacobian matrix are scaled with respect to the L_2 norm of each row. In these tests, we vary the convergence tolerance of the outer and inner tolerance, ϵ_o and ϵ_i respectively, to understand how either approximating A or preconditioning the Schur complement affects the number of Kyrlov vectors used by method M3. Method M1 takes three Newton iterations to solve this problem.

First we examine the results of approximating matrix A in Equation (6.8) as outlined in § 6.3.3. Recall that matrix A appears in both the right-hand- and left-hand-sides of Equation (6.8). We begin with substituting $\tilde{A}$ into the left-hand-side of Equation (6.8). The bulk of the cost of method M3 occurs from this portion of the problem from the k_o times an $(n \times n)$ dimensional system of equations is solved. The inverses of the approximations $\tilde{A} = \text{diag}(A)$ and $\tilde{A} = I$ can be computed trivially, and a direct method can be used to solve $\tilde{S}\tilde{\delta}_i = r_S$, where $\tilde{S} = -C\tilde{A}B$. Here, we use full LU decomposition of $\tilde{S}$ and Gauss elimination to solve this problem. Unfortunately, method M3 failed on the first Newton iteration when either approximation is used in place of A in the left-hand-side. A third approximation was also attempted, where $\tilde{A}$ is a tridiagonal matrix made of the three main diagonals of A , or $\tilde{A} = \text{tridiag}(A)$, whose inverse is more easily computed than that of the full matrix A . This approximation also failed, but on the second Newton iteration. From these results, we conclude that solving step I.B.4, $Ay = w$, in Algorithm 6.3.2 is critical to the overall solution of the problem. Therefore, a more accurate approximation of A , or even S , is needed if we wish to reduce the majority of the operation expended by method M3.

Conversely, approximating A on the right-hand-side of Equation (6.8) with either $\tilde{A} = \text{diag}(A)$ or $\tilde{A} = I$, method M3 still converged to a solution, but required more Newton iterations to reach the solution. At outer and inner tolerances of $\epsilon_o = \epsilon_i = 1 \times 10^{-6}$, method M3 without approximating r_S converges in three Newton iterations, while for either approximation $\tilde{A}$, the method converged in four Newton iterations. For this test case, one solve of the $(n \times n)$ system of equations per Newton iteration was saved at the expense of additional Newton iterations to reach a converged solution.

Applying a preconditioner to Equation (6.8) is the other option for reducing computational effort examine here. In a comparison test, we apply the preconditioner, $\tilde{S}$, along with approximations of the right-hand-side to see the combined effects. Table 6.5 compares the total number of Newton iterations needed to reach convergence, and the number of outer iterations used in the first Newton step for different tolerances of the Schur complement solver. The preconditioner does not contribute a significant amount of additional compu-

Table 6.5: Effect of preconditioner $\tilde{S}$ on number of Newton iterations required to reach convergence, and on number of iterations of outer solver during first Newton iteration (# Newton iterations / # Outer iterations) for different levels of accuracy of the Schur complement solver and approximations of the right-hand-side. Tests conducted on a (40×40) element mesh.. For instances where the procedure failed, the Newton iteration when failure occurred is indicated.

ϵ_o	ϵ_i	A		$\tilde{A}$		I	
		off	on	off	on	off	on
1×10^{-2}	1×10^{-3}	fail(1)/8	4/9	fail(1)/8	6/9	fail(1)/8	6/9
1×10^{-3}	1×10^{-3}	fail(2)/13	4/13	fail(2)/13	6/13	fail(3)/13	6/13
1×10^{-3}	1×10^{-4}	4/13	3/13	4/13	4/13	4/13	4/13
1×10^{-4}	1×10^{-4}	4/16	3/16	4/16	4/16	4/16	4/16
1×10^{-4}	1×10^{-5}	3/16	3/16	4/16	4/16	4/16	4/16

tation time. Application of $\tilde{S}$ takes on the order of $\leq \mathcal{O}(-2)$ seconds, while an inner solve involving S is $\mathcal{O}(-1) - \mathcal{O}(0)$ seconds on a PC with 4 Core Duo Intel Xeon processors, each subprocessor at 2.66 GHz, and 16 GB of RAM.

The use of the preconditioner did not reduce the number of outer iterations in the first Newton step for all of the cases as suggested by Bangerth et al. [9]. Either the number of outer iterations did not change with preconditioning, or increased as in the case of $(\epsilon_o = 1 \times 10^{-2}, \epsilon = 1 \times 10^{-3})$. However, the method failed in this case when preconditioning was not used. Subsequent steps were not examined because the presence of the preconditioner changes the update vector for each Newton step. The preconditioner did reduce the number of Newton iterations taken by M3 when comparing at specific combinations of ϵ_o and ϵ_i , even when approximations to r_S were made. At tighter tolerances, this reduction in number of Newton iterations disappears. Furthermore, at cases $(\epsilon_o = 1 \times 10^{-2}, \epsilon = 1 \times 10^{-3})$ and $(\epsilon_o = 1 \times 10^{-3}, \epsilon = 1 \times 10^{-3})$, M3 converged with the preconditioner “on” whereas the method failed with the preconditioner was not applied. The addition of the preconditioner with the use of the approximate $\tilde{A}$ was not able improve enough the rate of convergence of the method to have quadratic convergence at tight tolerance levels. At best four Newton iterations were need to reach convergence.

Overall, preconditioning reduces the total number of Newton iterations at relatively loose convergence tolerances. For large problems, where one cannot afford strict solution criteria because of resource limitations, preconditioning is a helpful method of reducing the overall number of Newton iterations without contributing significant additional costs per iteration.

6.5 Concluding Remarks

We have presented a new method of solving a two-phase, free-boundary, crystal growth model using a two-dimensional, heat conduction problem. The most efficient and robust method of solving such problems in two dimensions is to use a direct solver with the heat and mesh equations solve simultaneously. However, direct solvers do not scale well resource-wise when solving such problems in three-dimensions ($W \propto N_{nodes}^4$), and an iterative solver must be used which scales as $W \propto N_{nodes}^2$.

In the method presented here, we “map” away the difficult portion of the problem, calculating the location of the free-boundary, and solve each Newton step using block linear algebra or a Schur complement formulation of the problem. The Schur complement problem can be solved using either direct or iterative methods (M3). We showed that solving the Schur complement problem with an exact direct method achieves the same quadratic rate of convergence as our standard direct, frontal solver, method M1. GMRES, method M2, could also solve the problem, and converge quadratically depending on the size of the Krylov subspace. The rate of convergence of solving the Schur complement formulation, M3, using an iterative solver depends on the number of Krylov vectors used by the method. Using a hybrid solution method to solve the Schur complement formulation, we showed that overall convergence of Newton’s method depends more on the accuracy of the approximation of the interface unknowns than the field unknowns. Solving the Schur complement problem “well” with an iterative solver, however, depends more on the number of inner Krylov vectors than outer. This suggests that the overall Schur complement problem is “easy” to solve if the action of the Schur complement matrix on a vector is resolved to a certain level of accuracy.

We also computed how computational effort expended by methods M1, M2, and M3, scaled with problem size by counting the number of operations in terms of N_{nodes} , problem size, and k , number of Krylov vectors. In addition, how much time in seconds each method took to compute a Newton iteration on three different mesh sizes was timed. For the two-dimensional meshes examined here, the frontal solver, M1 was faster overall than either M2 or M3. However, the total work of M1 scales poorly with problem size compared with either M2 and M3: $W_{M1} \propto N_{nodes}^4$ versus either $W_{M2} \propto N_{nodes}^2$ or $W_{M3} \propto N_{nodes}^2$. GMRES, M2, also scales quadratically with the number of Krylov vectors, k_G . The iterative Schur complement method, M3, depends on both the number of outer (k_o) and inner (k_i) number of Krylov vectors used. M3 scales quadratically with k_i and linearly with k_o . The results of the times tests agreed with the scaling analysis. Both methods M2 and M3 perform better

than a direct solver, M1, for large problem sizes as long as the number of Krylov vectors used is much less than the number of unknowns. Furthermore, the Schur complement method converged faster than GMRES as the problem size increased, suggesting an advantage to segregating the interface equations when solving a free-boundary problem. Extending this analysis to three-dimensions, the required work for the three methods scales as $W_{M1} \propto N_{nodes}^7$, $W_{M2} \propto N_{nodes}^5$.94, and $W_{M4} \propto N_{nodes}^6$.72. Both M2 and M3 scale better than M1 for three-dimensional problems, with M2 performing better than M3.

Lastly, we presented two ideas of reducing computational effort used by the Schur complement solver, M3. The first idea was to approximate the matrix A in Equation (6.8) by either $\tilde{A} = \text{diag}(A)$ or I . The method successfully converges when either simplifications were substituted into the right-hand-side of Equation (6.8) saving one iterative solve of an n -dimensional problem, but at the expense of additional Newton iterations. However, further savings by approximating $A \approx \tilde{A}$ in the left-hand-side of Equation (6.8) was not successful, implying that the structure of the Schur complement is critical to solving for the interface location. The second idea was the application of a preconditioner, $\tilde{S}$, to Equation (6.8), where the preconditioner approximates the Schur complement (see Equation (6.27)). The use of a preconditioner did not reduce the number of outer iterations, but did not take more iterations except in the case where the method converged with preconditioning while it failed without preconditioning. However, at looser tolerance criteria, preconditioning reduced the total number of Newton steps to reach convergence, or made a failing case succeed with little additional cost. For large problems, where one cannot afford the cost of increased Krylov subspace sizes, this can be an asset as M3 will converge at looser solver tolerances.

Relatively complex elliptical mesh generation equations were used in the test problem presented here. Simpler mesh equations, such as pseudo-solid equations [154] take less time to compute and have a diagonally dominant, semi-definite Jacobian matrix. The approximations of A attempted here may work better with pseudo-solid mesh equations, as these approximations assume A is diagonally dominant.

While solving a Newton step for a free-boundary problem using a Schur complement formulation is resource heavy for a two-dimensional problem, the method does successfully compute solutions. Even though the algorithm is complex with its nested iterative solves, this provides many parameters to use to “tune” the procedure to optimally solve a given problem. The next test is to apply these ideas to solve a free-boundary, crystal growth problem in three-dimensions, where a Schur complement formulation could prove to be an important tool for solving such problems in an efficient manner.

Part IV

Concluding Remarks and
Appendix

Chapter 7

Concluding remarks

Contents

7.1	Thesis Summary	169
7.2	Directions for Future Research	172
7.2.1	Investigating the Approximate Block Newton Method	173
7.2.2	Development of time-dependent, multi-scale model for control ap- plications	175
7.2.3	Implement Schur complement formulation on 3D code	178

7.1 Thesis Summary

In this thesis we studied both the underlying physical driving forces in vertical Bridgman systems and innovative numerical methods to solve such problems. The crystal growth studies were based on an industrial process designed by eV Products Inc. to grow the type II semiconductor cadmium zinc telluride. We examined novel growth conditions that are not readily realizable in an industrial setting. The numerical methods we developed satisfy two different goals. The first was to develop a robust method of coupling separate computer codes to produce a self-consistent model of heat transfer in a furnace and bulk transport in the crystal growth domain. The second goal was to solve the free-boundary problem that occurs in crystal growth in a manner that scales better for a three-dimensional problem than a direct solver.

The equations governing momentum, energy, and mass transport in the crystal growth domain were discretized using the Galerkin finite element method. This local model in-

cluded the effects of phase change during the solidification process of a melt growth system. The effect of heat transfer from the furnace was taken as a boundary condition on the external domain boundary. For transient simulations, an implicit trapezoidal time marching scheme was employed.

The work presented in Chapter 3 studied the effect of ampoule tilting on zinc segregation in the lateral (radial) direction. In an effort to reduce the complexity of the crystal growth problem, the cylindrical ampoule was approximated as planar. Upon tilting the ampoule, the resulting fluid flow structure remained two-dimensional in a planar geometry instead of developing a three-dimensional structure as in cylindrical geometries. With a two-dimensional model, time integration to study zinc segregation in the solid phase was studied in an efficient manner. The simulations presented here were in qualitative agreement with previous three-dimensional, quasi-steady-state models [189]. We found that tilting had little to no effect on zinc segregation in the axial direction even though melt convection intensified with tilt angle. This suggested that, even when the ampoule is completely vertical, convection was intense enough to approach the Scheil [155] limit of complete mixing. Lateral segregation, however, was more strongly affected by ampoule tilting. Results suggested that at low tilt angles, lateral segregation was influenced more by melt flow topology, namely the location where the flow stagnation streamline intersected the melt-crystal interface. At higher tilt angles, lateral segregation is influenced by the shape of the melt-crystal interface. Between these two limits, the forces balanced at a certain tilt angle, $\theta = 10^\circ$, that gave a lateral concentration profile that flattened in the center. The combined effects of axial and lateral concentration was examined by calculating the minimum root mean square of concentration, c_{RMS} , at different tilt angles in a cross-sectional area (0.6 in $\times$ 0.6 in square) that corresponded to actual device used by eV Products Inc. Optimal overall segregation occurred at $\theta = 20^\circ$. Tilting, however, did not produce dramatic improvements in zinc segregation in the solid, but may be more effective in larger-diameter and higher-gradient systems where buoyant flow is stronger than considered here. Ampoule tilting could still prove to be a purposeful member of an arsenal of process modifications to improve standard vertical Bridgman growth.

In Chapter 4 we presented results of how the applied furnace temperature gradient affects the shape of the melt-crystal interface of CZT grown in a vertical Bridgman furnace. Interface shape is governed by the relative heat conductivities of the two phases, latent heat generated at the interface, and furnace design (which includes ampoule material and furnace temperature gradient). Open-loop tests were performed with the model where the furnace temperature gradient was increased by an algebraic stretching factor. The original

temperature gradient was taken from a self-consistent model developed by Pandey et al. [136] of eV Products' low-pressure, low-gradient furnace. Under these growth conditions, CZT has a concave toward the melt interface shape. As the temperature gradient was increased, the interface flattened as expected, but the intensity of flow near the interface decreased while overall flow intensity increased. This result suggested that increased mixing in the melt does not play a significant role in determining interface shape for this system. To analyze the separate effects of heat convection through the melt and heat conduction through the solid, we applied the increased temperature gradient to solely along the external melt boundary and solely along the external crystal boundary. An increased temperature gradient along the melt essentially did nothing to the interface shape compared to the base case. However, the increased gradient along only the solid had a dramatic flattening of the interface. With the increased axial temperature gradient in the solid, latent heat was conducted away axially through the solid instead of radially which flattened the interface. We also showed that the interface shape could be influenced by perturbations in the furnace temperatures, but only at low furnace gradients.

We also applied closed-loop control on the system in order to control the interface shape by actuation of the furnace temperature gradient. To account for latent heat effects, the coordinate frame of reference was fixed with the moving interface instead of the furnace. A simple proportional control algorithm was used to relate interface deflection, ΔX , to furnace temperature gradient. Time delays due to heat transfer from changes in the furnace heaters to the external domain boundary were ignored. Three control scenarios were enacted: actuation along the entire external boundary, along only the solid portion, and along only the melt portion. When the increased gradient was applied along the entire boundary, high gains were needed to reach the set-point $\Delta X = 0.0$. However, when the increased gradient is applied only to the solid portion of the boundary, the set-point was reached for all gains applied and with relatively low final temperature gradients. We even had the unexpected benefit of producing a convex interface shape. Control of the temperature gradient along only the melt resulted in an unstable situation, which was not indicated in the open-loop studies. Our closed-loop control study indicated that interface shape can be successfully controlled by focusing actuation on specific areas of actuation.

Chapter 5 presented our work on building a more realistic model of a vertical Bridgman system by coupling two disparate computer codes, one which modeled the furnace and the other which modeled the growing crystal domain. The codes were coupled through heat transfer boundary conditions along a shared boundary position. Each sub-model was treated as set of block residual equations, and were coupled using an approximate block

Newton scheme. Unlike previous work [45,135,136,196] using a block Gauss-Seidel scheme, the approximate block Newton method incorporates information from the off-diagonal blocks. As with conventional use of the methods, the block Newton method is much more complicated and involved than the Gauss-Seidel method. However, whereas block Gauss-Seidel was highly dependent on the form of the heat transfer boundary condition, this dependency was virtually eliminated with approximate block Newton. Furthermore, we were able to achieve quadratic convergence for different problems ranging from a simple one-dimensional model to a complex, two-dimensional model of eV Products' electrodynamic gradient freeze furnace.

In Chapter 6 we posed the notion of solving a crystal growth free-boundary problem using a Schur complement formulation to solve a Newton step in place of a direct or an iterative solver. In our crystal growth models, the location of the melt-crystal interface is set implicitly by the melt temperature isotherm condition, Equation (6.3). Two-dimensional problems are readily solved by direct methods, but such methods scale poorly with three-dimensional problems. The Schur complement formulation separates the isotherm equations from the rest of the governing equations, in effect mapping away the difficult portions of the problem. Using a direct solution method to solve the resulting Schur complement problem, we achieved the same rate of convergence as our frontal solver. With an iterative solver, the Schur complement formulation reached the frontal solver rate of convergence, but with an additional expense of more Krylov vectors at each sub-iteration. Overall, the Schur complement method proved to be more expensive than the frontal solver and GMRES at solving a two-dimensional free-boundary problem. Some computational effort can be saved by approximating the right-hand-side of the Schur complement problem, Equation 6.8. The bulk of the computational effort came from solving the Schur complement. Applying a preconditioner that approximated the Schur complement reduced the number of outer iterations per Newton step and overall number of Newton iterations needed to reach convergence.

7.2 Directions for Future Research

The work presented here has enriched the crystal growth field through modeling of novel processes for CZT growth and implementing closed-loop control that indicates the possibility of growing CZT with a convex interface. Additional advancements have been made in multi-scale modeling in a robust manner that is highly detailed of both the furnace and the growing crystal. Furthermore, we developed an innovative method of solving the

free-boundary problem that occurs during a phases change. Many additional questions and challenges naturally arise from this work. In this section, we highlight some future research directions that extend from the results presented here.

7.2.1 Investigating the Approximate Block Newton Method

The approximate block Newton method, discussed in Chapter 5 proved to be a powerful tool in coupling disparate computer codes to develop realistic, multi-scale models of melt-growth systems. Research is needed to understand the underlying mechanics of the ABN method. This method is complex, and there remain different interpretations on how it should be applied in practice (see § 5.3). Specifically, in our derivation, the action of the Schur complement is approximated as

$$\frac{1}{\epsilon} \mathbf{S}(\epsilon \mathbf{w}) = \mathbf{w} + \frac{1}{\epsilon} \left\{ \mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}(\epsilon \mathbf{w}), \mathbf{y}^{(k)}) \right\} \quad (7.1)$$

whereas References [4, 115, 116], the Schur complement is approximated by

$$\frac{1}{\epsilon} \mathbf{S}(\epsilon \mathbf{w}) = \mathbf{w} + \frac{1}{\epsilon} \left\{ \mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}(\epsilon \mathbf{w}), \mathbf{y}^{(k)} + \epsilon \mathbf{w}) \right\} \quad (7.2)$$

which includes an extra perturbation of the y unknowns compared to Equation (7.1). In the current setup of the algorithm, the x unknowns are assigned to Cats2D and the y unknowns of CrysVUn.

This problem may seem a question of semantics, but it also touches on improving the computational efficiency of the algorithm. In our initial implementation of the approximate block Newton method, the effect of the perturbation of y is avoided by iterating CrysVUn until the subproblem is completely converged such that $y = \mathcal{G}(x, y)$ is determined only by the applied boundary condition, or x . In mathematical terms:

$$\mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(n)}) = \mathcal{G}(\mathbf{x}^{(k)}) \quad \text{for large } n \quad (7.3)$$

where n is the internal iteration counter of $\mathcal{G}$. Cats2D only takes one Newton step, or an equivalent number of modified-Newton steps, on each call to solver $\mathcal{F}$.

Running CrysVUn until the sub-problem is an inefficient way to compute the solution of the overall problem. Each sub-solver, $\mathcal{F}$ and $\mathcal{G}$, only needs to take a smaller number of Newton steps, such as one, in the direction of the solution for the overall problem to converge. Since the furnace domain is nonlinear, perturbing its unknowns could be

significant if the problem is not iterated until convergence. To study whether or not this difference in interpretation is significant, we need access to the entire CrysVUn unknown vector instead of just the temperatures and heat fluxes along the shared boundary. The two implementations can then be evaluated by testing which application converges faster.

Another related area of interest is the implication of dealing with parameterized boundary data. As a consequence of not having access to all of CrysVUn unknowns, we only have its parameterized boundary data. Furthermore, the heat fluxes that are passed between the codes are quantities derived from the temperature data. It is unclear then what quantities should be perturbed when approximating the off-diagonal block Jacobians: both temperature and heat fluxes, just temperature, or just heat flux. It may seem obvious that the temperatures are the only true degrees of freedom, but one must recall the structure of the mixed boundary conditions, Equations (5.70) and (5.71). If either coupling parameter α or β is equal to 1, we have an all flux boundary condition. Therefore, in this situation, the degrees of freedom would be heat flux. This question was avoided in the one-dimensional problem presented in Chapter 5 because we had explicit equations for heat flux as a function of temperature. Currently we perturb both temperatures and heat fluxes. The superiority of the different scenarios can be assessed by comparing their rates of convergence.

Convergence also depends on the value of ϵ used in the finite difference approximation. A centered-difference approximation would be more accurate than the finite difference approximation for the off-diagonal block Jacobians, allowing a larger value of ϵ ; but with a possible trade-off in efficiency. Multiple values of ϵ may be needed; one value for certain variables, and another value for others.

The deforming mesh in local portion in the model of a gradient freeze furnace proved to be quite sensitive to changes in heat flux data during perturbation steps and updates to the solution vector after a block Newton step. This sensitivity allowed only for a conservative approach to reach the final solution by forcing us to take small continuation steps to reach the final values of either the melting temperature or latent heat coefficient. One proposed method to help alleviate the sensitivity is to apply zero-order continuation when updating heat flux data.

Another approach is to approximate the off-diagonal block Jacobians using a centered difference approximation, which is second order accurate, rather than finite difference approximation where Equation (5.44) is modified such that

$$\mathbf{J}\mathbf{w} = \frac{\mathbf{r}(\mathbf{x} + \epsilon\mathbf{w}) - \mathbf{r}(\mathbf{x} - \epsilon\mathbf{w})}{2\epsilon}. \quad (7.4)$$

With this alternative approximation, Steps 2-4 in the block Newton algorithm are altered. The modified right-hand-side, $\mathbf{r}_S$, in Equation (5.39) becomes

$$\mathbf{r}_S = \mathbf{y}^{(k)} - \mathcal{G} \left\{ \mathbf{x}^{(k)}, \mathbf{y}^{(k)} \right\} + \frac{1}{2} (\mathcal{G}(\mathbf{x}^{(k)} - q, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} + q, \mathbf{y}^{(k)})), \quad (7.5)$$

and the Schur complement, Equation (5.50) becomes

$$\frac{1}{\epsilon} S \epsilon \mathbf{w} = \mathbf{w} + \frac{1}{2\epsilon} \left\{ \mathcal{G}(\mathbf{x}^{(k)} + C \epsilon \mathbf{w}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} - C \epsilon \mathbf{w}, \mathbf{y}^{(k)}) \right\}. \quad (7.6)$$

Matrix C also changes form in both its application in the Schur complement, Equation (5.46),

$$\frac{1}{\epsilon} C \epsilon \mathbf{w} = \frac{1}{2\epsilon} \left\{ \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} - \epsilon \mathbf{w}) - \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon \mathbf{w}) \right\} \quad (7.7)$$

and in calculating the update vector $\Delta \mathbf{x}$ in Equation (5.57)

$$\Delta \mathbf{x} = q + \frac{1}{2} \left\{ \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} - \Delta \mathbf{y}) - \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta \mathbf{y}) \right\} \quad (7.8)$$

where q is determined in Step 1 of the algorithm.

The algorithm becomes slightly more complicated with a centered difference approximation, and the number of calls to the subsolvers is doubled in some instances. The increase in accuracy may not be significant enough in certain steps, such as r , to warrant changes to the algorithm. Future research would analyze that gains in accuracy with respect to additional effort.

7.2.2 Development of time-dependent, multi-scale model for control applications

The next step in developing our multi-scale model is to incorporate the ability to perform time-dependent calculations. With a transient calculation, we will be able to model the entire crystal growth system throughout the growth process. Large modifications of the coupling code is not needed as the approximate block Newton method has the capability of executing transient calculations embedded within its structure.

Matthies and Steindorf [115] give an excellent outline of both explicit and implicit time-stepping algorithms for solving transient problems of coupled systems. Let us assume that we have two subsystems, where the time evolution of each subsystem is described by a

differential equation of the form

$$\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, \mathbf{x}_2) \quad (7.9)$$

$$\dot{\mathbf{x}}_2 = f_2(\mathbf{x}_1, \mathbf{x}_2) \quad (7.10)$$

where the above equations are the larger evolution system of $(\mathbf{x}_1, \mathbf{x}_2)^T$. We also assume we have implicit integrators for each subsystem—as is the case with Cats2D and CrysVUn— ϕ_1 and ϕ_2 :

$$\mathbf{x}_1^{(n)} = \phi_1 \left(\mathbf{x}_1^{(n)}, \mathbf{x}_1^{(n-1)}, \mathbf{y}_2 \right) \quad (7.11)$$

$$\mathbf{x}_2^{(n)} = \phi_2 \left(\mathbf{x}_2^{(n)}, \mathbf{x}_2^{(n-1)}, \mathbf{y}_1 \right) \quad (7.12)$$

where n denotes the time point, and the functions $\mathbf{y}_j(t)$, ($j = 1, 2$) represent the variables of the other subsystem during the time interval $\Delta t = t_n - t_{n-1}$, and assumed given and their dependence on time suppressed. For simplicity, we assume Δt is the same for both subsystems. However, $\mathbf{y}_1$ and $\mathbf{y}_2$ are not known, and must be extrapolated from $\mathbf{x}_1$ and $\mathbf{x}_2$. Matthies and Steindorf [115] give two different extrapolation operators depending on whether the coupling algorithm is explicit or implicit. If the coupling algorithm is explicit, we have

$$\mathbf{y}_j = \psi_j \left(\mathbf{x}_j^{(n-1)} \right) \equiv \mathbf{x}_j^{(n-1)}, \quad j = 1, 2. \quad (7.13)$$

Substituted into Equations (7.11) and (7.12), we get

$$\mathbf{x}_1^{(n)} = \phi_1 \left(\mathbf{x}_1^{(n)}, \mathbf{x}_1^{(n-1)}, \mathbf{x}_2^{(n-1)} \right) \quad (7.14)$$

$$\mathbf{x}_2^{(n)} = \phi_2 \left(\mathbf{x}_2^{(n)}, \mathbf{x}_2^{(n-1)}, \mathbf{x}_1^{(n-1)} \right) \quad (7.15)$$

known as a staggering algorithm [62]. With explicit coupling, the coupled subsystems can be solved independently, but, as the overall system is explicitly coupled, there are limitations due to the critical time step associated with the global system. A variation on this staggering method is to update $\mathbf{x}_1$ in Equation (7.15) after Equation (7.14)

$$\mathbf{x}_2^{(n)} = \phi_2 \left(\mathbf{x}_2^{(n)}, \mathbf{x}_2^{(n-1)}, \mathbf{x}_1^{(n)} \right). \quad (7.16)$$

Implicit coupling, as with implicit integrators in general, are used to eliminate critical

time step dependencies. For an implicit formulation, the extrapolation operators are

$$\mathbf{y}_j = \psi_j \left(\mathbf{x}_j^{(n-1)} \right) \equiv \mathbf{x}_j^{(n)}, \quad j = 1, 2. \quad (7.17)$$

and Equations (7.11) and (7.12) become

$$\mathbf{x}_1^{(n)} = \phi_1 \left(\mathbf{x}_1^{(n)}, \mathbf{x}_1^{(n-1)}, \mathbf{x}_2^{(n)} \right) \quad (7.18)$$

$$\mathbf{x}_2^{(n)} = \phi_2 \left(\mathbf{x}_2^{(n)}, \mathbf{x}_2^{(n-1)}, \mathbf{x}_1^{(n)} \right). \quad (7.19)$$

Matthies and Steindorf [115] refer to this case as *strong* coupling, and is equivalent to a monolithic formulation. In this scheme, the equations must be solve simultaneously.

There are advantages and disadvantages with both methods. Explicit coupling is easier to implement and the systems can be solve independently of each other, but is limited by a critical time step. Implicit coupling does not have a critical time step, and is more robust in that the results obtained are the same if the problem were formed in a monothlithic manner. The disadvantage is that it is more difficult to implement The framework, however, for implicit coupling has been introduced through solving steady-state, multi-scale problems with a approximate block Newton method. Since both subsystems are discretized with respect to space and time, the ABN method can be applied to the time integration equations instead of the steady-state residual equations.

The case can still be made for explicit coupling. The time-scale for heat transfer in the furnace is much longer than that of the crystal, such that the furnace environment evolves slowly. It may be unnecessary to time integrate in lock-step both parts of the problem. Felippa et al. [62] give numerous examples of time integration schemes of coupled systems. One interesting idea is that of subcycling in which one subsystem (Cats2D) is integrated a smaller time step than the other subsystem (CrysVUn). Perhaps a hybrid explicit/implicit coupling method could be devised where the ABN method is used at specific time points, and the subsystems are integrated independently (explicitly) during times between these points. With a fully transient, coupled model, we would be able to simulate an entire growth run and optimize the furnace temperature profile.

In addition to optimization, we will be able to explore more complex control ideas with a transient model. Instead of changing the temperature gradient at the shared boundary, actuation could be applied directly to the furnace heat powers or set-point temperatures. The multi-scale model would also properly capture the time delay between changes in the applied temperatures and what the growing crystal sees.

Model based control in an experimental setting is another attractive idea. Due to the harsh conditions under which crystals are grown, very little, if any, observable data can be acquired from the system. Different groups have developed creative methods of obtaining real time data. Kim, Witt, and Gatos related flow intensity to thermocouple measurements [89]. Batur and co-workers [12] measured interface position by means of a transparent furnace and image processing techniques. Eddy current sensors have been shown to be effective in tracking the melt-crystal interface during vertical Bridgman growth of CZT [47, 48].

Since the availability of experimental process data is limited, state estimation methods are needed to design effective feedback controllers. The coupled, multi-scale model could be used as a state observer in an experimental crystal growth system. The complex physical behavior occurring within the furnace and crystal can only be captured by a sufficiently complex model. The model would be optimized through comparison with experiments before being used to estimate the physical system. Limited real-time data could be used to augment the model for a more accurate estimate of growth dynamics. With modern computing technology, a desktop computer with reasonable processing capability can run the code in parallel to a growth run. Also, crystal growth is on the order of millimeters per hour which gives time for a simulation to be run and actuation to occur.

7.2.3 Implement Schur complement formulation on 3D code

A Schur complement form is an alternative means of solving a free-boundary, crystal growth problem, but does not, as shown in Chapter 6 efficiently solve a Newton step of a two-dimensional problem. Direct solvers remain the most efficient and robust methods for solving problems in two-dimensions, but a Schur complement form may gain advantage in a three-dimensional problem in which solutions are solved iteratively.

Cats3D, the three-dimensional analog to Cats2D, is a FEM code for solving coupled momentum, heat, and mass transfer problems. The Galerkin-Least Squares method is used for the Navier-Stokes equations [77], and the Streamline-Upwind Petrov-Galerkin method of the energy and species equations [22]. Tetrahedral elements are used to represent the field variables. The discretized equations are solved using a portable MPI-based parallel implementation [154, 198]. The nonlinear set of algebraic equations is solved iteratively using Newton's method. At each Newton step, the linear system of equations is solved using GMRES.

An explicit front-tracking method is used in Cats3D to determine the location of the

melt-crystal interface. Unlike Cats2D where both the transport and mesh equations are solved simultaneously, Cats3D uses a decoupled procedure where the governing equations are first solved on a fixed mesh with the melting-point isotherm condition

$$T = T_{mp} = 0 \quad \text{at interface} \quad (7.20)$$

relaxed. In general, the solution obtained in this manner will not satisfy Equation 7.20. Instead, the displacement between the interface and the melting-point isotherm is estimate using a first order Taylor's expansion of the temperature field about the current axial position of the interface:

$$\delta_z = -\frac{T}{\partial T / \partial z} \Big|_i, \quad (7.21)$$

where δ_z is the displacement in the z -direction. The computational grid is generated using a pseudo-solid method [153,199] where δ_z is applied as a Dirichlet boundary condition at each node on the interface. This deforms the mesh to conform to the estimated location of the melting-point isotherm. The procedure is repeated until δ_z is suitably small.

A Schur complement form for solving the free-boundary problem requires coupling between the governing equations and the mesh equations. In a decoupled procedure, sub-matrix B of the Jacobian is zero. Therefore, the above procedure must be modified. One idea is to solve the governing and mesh equations simultaneously as is done in Cats2D. The interface unknowns would be separated from the field equation unknowns and the remaining mesh unknowns. Solving the equations simultaneously has proved to be unfeasible for a three-dimensional problem in the past, but may be doable with current computing resources and meshing techniques. Another idea is to only couple the field equations with the melting-point isotherm. The field equations would include their dependence on the location of the interface such that $B \neq 0$. First the field equations and interface location would be computed, then the mesh equations would be updated based on the interface location. This semi-coupled procedure would be repeated until the overall solution is deemed converged. Such a procedure may be more amenable to a Schur complement formulation of the problem since matrix A would not include the mesh equations but only the field equations. For a heat conduction problem, the inverse of matrix A would be better behaved and may required less iterations to solve.

Appendix A

Use of Under-Relaxation in Coupling Global and Local Crystal Growth Models

Contents

A.1 Chapter Summary	180
A.2 Introduction	181
A.3 Formulation of Iterative Algorithm	182
A.3.1 Convergence of Iterative Algorithm	183
A.4 Problem Formulation and Results	184
A.4.1 Coupling Technique	185
A.4.2 Results	185
A.5 Conclusions	189

A.1 Chapter Summary

The modular coupling of existing numerical codes to model crystal growth processes will provide for maximum effectiveness, capability, and flexibility. However, significant challenges are posed to make these coupled models mathematically self-consistent and algorithmically robust. This chapter presents sample results from a coupling of the CrysVUn code, used here to compute furnace-scale heat transfer, and Cats2D, used to calculate melt

fluid dynamics and phase-change phenomena, to form a global model for a Bridgman crystal growth system. The coupling method presented here is based on a block Gauss-Seidel method with under-relaxation. Performance of the standard block Gauss-Seidel method used to implement the CrysVUn-Cats2D coupling is unreliable and inefficient. The implementation of that includes under-relaxation is shown to be ineffective for improving the coupling performance in a model one-dimensional problem representative of a melt crystal growth model. Details of an Approximate block Newton approach toward coupling the global and local models are presented in Chapter 5.

A.2 Introduction

The modeling of melt crystal growth processes is an inherently multi-scale challenge, with relevant length scales ranging from furnace dimensions to atomic-sized features in the grown crystal [46]. A faithful depiction of the interaction of these scales is needed to advance our scientific understanding of crystal growth processes, as well as to make our process models quantitatively predictive and technologically useful. There are also mathematical and algorithmic challenges for the coupling of different models that describe these phenomena at different scales. In the following study, we address the addition of under-relaxation to the study conducted by Yeckel et al. [196] with respect to coupling of global-scale furnace heat transfer models and local-scale models for melt crystal growth.

Pandy et al. [136] presented a multi-scale model to study crystal growth in an industrial electrodynamic gradient freeze furnace. This model combined a furnace-scale model based on the commercial code CrysVUn [8, 97] coupled to our local model, Cats2D [195], that solves for heat transfer, melt flow, and melt-crystal interface shape. Pandy et al. examined the errors introduced by two different one-way couplings of the models, i.e., providing either flux or temperature boundary conditions from the furnace model to the local model without iteration, compared to a fully self-consistent solution of the multi-scale model.

This study also showed that the convergence of the iterative computations strongly depended upon how the two models were coupled. Further work by Yeckel et al. [196] examined the mathematical basis for the success or failure of the iteration techniques pioneered in [136]. Specifically, Yeckel et al. employed the framework of fixed-point iterations to assess convergence behavior for a simple model representing a coupled global-local problem and more complicated, two-dimensional problems. Significantly, this prior study proved that certain coupling implementations that were physically reasonable resulted in algorithms that will never converge, demonstrating that notions based on physical intuition

may not be useful for predicting algorithm performance.

We examine coupled models from a more general perspective, with a specific discussion of the implementation of under-relaxation in a block Gauss-Seidel iteration between sub-models. This approach is motivated to test whether under-relaxation will make our previous coupled model [136, 196] more robust with respect to convergence. The under-relaxation implementation is tested on a simple, one-dimensional problem that is representative of the challenges inherent in crystal growth modeling.

A.3 Formulation of Iterative Algorithm

We are specifically interested in coupling a furnace-scale heat transfer model with a local model representing the details of melt crystal growth to find quasi-steady-state solutions for heat transfer, convection, and interface position in a Bridgman crystal growth system. As in Chapter 5, we consider a problem that is divided into two sub-problems that are coupled along a boundary, and denoted as:

$$\mathbf{r}_1(\mathbf{x}, \tilde{\mathbf{y}}) = \mathbf{0}, \quad (\text{A.1})$$

$$\mathbf{r}_2(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbf{0}, \quad (\text{A.2})$$

where $\mathbf{r}_i$ is a residual function associated with each subproblem, $\mathbf{x}$ denotes the unknowns of problem 1, and $\mathbf{y}$ denotes the unknowns of problem 2. The tilde indicates that these variables are from the other model. Converged solutions for $\mathbf{x}$ and $\mathbf{y}$ that simultaneously satisfy both equations will represent a fully converged solution to the coupled problem.

Again, we assume each subproblem has its own solver, and that these solvers are iterative. Furthermore, we assume these solvers can be written in fixed point form and that each iteration is convergent,

$$\mathbf{x}^{(k+1)} = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}), \quad (\text{A.3})$$

$$\mathbf{y}^{(k+1)} = \mathcal{G}(\tilde{\mathbf{x}}, \mathbf{y}^{(k)}). \quad (\text{A.4})$$

Here, $\mathcal{F}$ and $\mathcal{G}$ are the solvers of $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively, and the variables designated with the tilde are held constant during the iteration. The index k denotes the global iteration number corresponding with a converged solution to each subproblem. Each solver will perform its own iterations toward reaching the updated global iterate, but we do not specifically denote these inner iterations.

The block Gauss-Seidel method employed in [136,196] is modified to incorporate under-relaxation [39] after each block iteration yield the following procedure:

- I. Solve for $\mathbf{x}_*^{(k+1)} = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}^{(k)})$, with $\tilde{\mathbf{y}}^{(k)}$ held constant and iterations performed within the solver $\mathcal{F}$ until convergence is obtained (i.e., the norm $\|\mathbf{r}_1(\mathbf{x}_*^{(k+1)}, \tilde{\mathbf{y}}^{(k)})\| < \epsilon_1$).
- II. Under-relax the outcome of the iteration so that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega(\mathbf{x}_*^{(k+1)} - \mathbf{x}^{(k)})$.
- III. Solve for $\mathbf{y}_*^{(k+1)} = \mathcal{G}(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k)})$, with $\tilde{\mathbf{x}}^{(k+1)}$ held constant and iterations performed within solver $\mathcal{G}$ until convergence is obtained (i.e., the norm $\|\mathbf{r}_2(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}_*^{(k+1)})\| < \epsilon_2$).
- IV. Under-relax the outcome of the iteration so that $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \omega(\mathbf{y}_*^{(k+1)} - \mathbf{y}^{(k)})$.
- V. Repeat steps 1–4 until the global model is converged. Global convergence is achieved when both original residuals are satisfied, (i.e., the norms $\|\mathbf{r}_1(\mathbf{x}^{(k+1)}, \tilde{\mathbf{y}}^{(k+1)})\| < \epsilon_1$ and $\|\mathbf{r}_2(\tilde{\mathbf{x}}^{(k+1)}, \mathbf{y}^{(k+1)})\| < \epsilon_2$).

Above, ϵ_1 and ϵ_2 are convergence tolerances for model 1 and model 2, respectively, and ω represents the block under-relaxation factor, taken to be $\omega \leq 1$. When $\omega = 1$, the original iteration algorithm studied in [136,196] is recovered. Note that the Gauss-Seidel character of this procedure is obtained by using updated information from the first step, namely $\tilde{\mathbf{x}}^{(k+1)}$ rather than $\tilde{\mathbf{x}}^{(k)}$, in step 3 of the iteration sequence.

A.3.1 Convergence of Iterative Algorithm

To understand the convergence characteristics of the iterative procedure discussed above, it proves illuminating to step back from the specific formulation issues of each model and their coupling to consider the overall mathematical algorithm. We define the vector of total problem unknowns $\mathbf{z}^T \equiv (\mathbf{x}, \mathbf{y})^T$ and consider the entire procedure to define a fixed-point iteration, where successive iterates of the problem unknowns are computed from,

$$\mathbf{z}^{(k+1)} = \mathbf{H}(\mathbf{z}^{(k)}), \quad (\text{A.5})$$

where $\mathbf{H}(\mathbf{z})$ is a vector function and $\mathbf{z}^{(k)}$ and $\mathbf{z}^{(k+1)}$ are successive iterates. When

$$\mathbf{z}^* = \mathbf{H}(\mathbf{z}^*), \quad (\text{A.6})$$

the iteration has converged; $\mathbf{z}^*$ is a solution to the original problem and is denoted as a fixed point for the iteration. While we know the individual solvers, $\mathcal{F}$ and $\mathcal{G}$, we do not generally know the explicit form of the global iteration function, $\mathbf{H}$, because of the shared unknowns between the two models.

We assess the convergence of this fixed-point iteration by subtracting Equation A.6 from Equation A.5, followed by applying the mean value theorem from differential calculus and the triangle inequality from linear algebra, to arrive at a result known as the contraction mapping theorem,

$$\|\mathbf{z}^{(k+1)} - \mathbf{z}^*\| \leq \|(\partial\mathbf{H}/\partial\mathbf{z})|_{\mathbf{z}^*}\| \|\mathbf{z}^{(k)} - \mathbf{z}^*\|. \quad (\text{A.7})$$

Here, $\|\cdot\|$ denotes a suitable vector-matrix norm and $(\partial\mathbf{H}/\partial\mathbf{z})|_{\mathbf{z}^*}$ is the sensitivity matrix of the mapping function $\mathbf{H}$ with respect to the unknowns $\mathbf{z}$ evaluated at the fixed point $\mathbf{z}^*$. Its form is given by coefficients, $\partial\mathbf{H}_i/\partial\mathbf{z}_j$, where i and j denote indices of the matrix coefficients and vector components.

There are two important outcomes of the above analysis. First, convergence of the iterations can *only* be attained when

$$\|(\partial\mathbf{H}/\partial\mathbf{z})|_{\mathbf{z}^*}\| < 1. \quad (\text{A.8})$$

We expect that there are iterative procedures where $\|(\partial\mathbf{H}/\partial\mathbf{z})|_{\mathbf{z}^*}\| \geq 1$ and Equation A.8 is violated. Under these conditions, the distance between the solution and the next iterate, $\|\mathbf{z}^{(k+1)} - \mathbf{z}^*\|$, grows larger with each iteration, and the iteration will never converge to the solution for *any* initial guess, including the case of an initial guess corresponding to the exact solution for the combined problems. Second, when $\|(\partial\mathbf{H}/\partial\mathbf{z})|_{\mathbf{z}^*}\| < 1$, we expect the sequence to converge and the rate of convergence to be linear, i.e., the error in iterate $(k+1)$ will be linearly proportional to the error of the prior iteration at k , as shown above by Equation A.8.

A.4 Problem Formulation and Results

A one-dimensional heat transfer problem representative of melt crystal growth is studied in-depth to assess the addition of under-relaxation to the performance of the block Gauss-Seidel algorithm. We emphasize that this model problem is overtly simple; however, its subsequent behavior in a coupled iteration, presented below, does seem to mimic many

aspects of much more complicated counterparts, such as that presented in the prior section. To avoid dealing with the computational and analytical challenges posed by two- and three-dimensional crystal growth computations, our model problem is defined as a one-dimensional, steady-state heat transfer problem over two domains. Details of the one-dimensional model are given in § 5.4.1. This problem was originally defined in [135, 196].

For this and the following one-dimensional models, we use the same set of parameter values as in § 5.4.1, which are representative of a crystal growth system, namely $c = 9$, $\kappa = 0.1$, $T_1 = 0.98$, and $T_2 = 1$. We again consider two cases regarding radiation effects. The first sets $R = 0$, thereby eliminating the radiation in the problem and making it linear. The second case chooses a representative value of $R = 5.67$, making radiation a significant part of the computation. A schematic of the computation domain, and steady-state temperature profiles through both domains for the parameters given above are shown in Figure 5.1.

A.4.1 Coupling Technique

As stated in § 5.4.1, the temperature and heat flux must match at the coupling point for a converged, self-consistent solution to the governing equations. We define a set of mixed boundary conditions to be used for each sub-problem during model iterations given in Equations 5.70 and 5.71. Note that these boundary conditions are weighted toward Dirichlet or Neumann boundary conditions by the coupling parameters α and β . These equations form the basis for deriving two residual equations,

$$r_1(T|_{x=1-}, \tilde{T}|_{x=1+}) = 0, \quad (\text{A.9})$$

$$r_2(\tilde{T}|_{x=1-}, T|_{x=1+}) = 0, \quad (\text{A.10})$$

whose detailed forms are shown in Equations 5.72 and 5.73. These residuals are analogous to Equations A.1 and A.2, with $T|_{x=1-}$ and $T|_{x=1+}$ representing the global problem unknowns. The solvers F and G in Equations A.3 and A.4 represent the solution of the governing equations and associated boundary conditions of the sub-problems defined in Ω_1 and Ω_2 . Finally, the iteration procedure follows that outlined in § A.3.

A.4.2 Results

Figures A.1(a) and (b) show the convergence behavior of the block Gauss-Seidel algorithm without under-relaxation (i.e., with $\omega = 1$) and with under-relaxation (for $\omega = 0.1$),

respectively, for different (α, β) pairs and for $R = 0$ and $R = 5.67$. The dark spots on these diagrams indicate (α, β) pairs that are non-convergent after 1,000 iterations.

The results for the unrelaxed $\omega = 1$ method, shown in Figure A.1(a), are identical to those derived analytically in [196], with the gray areas corresponding to conditions where the norm of the sensitivity matrix of the global iteration strategy is unity or greater [135, 196]. For values of the coupling parameters corresponding to these regions, the iteration sequence will never converge, regardless of the value of the initial guess. The convergence behavior of the $R = 0$ case is dramatically different from that of the $R = 5.67$ case. The specific values of α and β that allow convergence change markedly with the increasing role of radiation in the problem. Interestingly, the area of the unshaded region changes very little for these convergence maps, indicating that roughly half of all choices of α and β would be successful, yet most iterations strategies that succeed for the linear $R = 0$ case are destined to fail for the nonlinear $R = 5.67$ case.

Under-relaxation does change the convergence behavior, as shown by the plots in Figure A.1(b). For the linear $R = 0$ case, under-relaxation improves the robustness of the algorithm by slightly enlarging the region of coupling parameter combinations that achieve convergence. Most notable are combinations with $\beta \approx 0.9$ and higher that permit convergence where it was not attainable for the unrelaxed algorithm. The outcome of under-relaxation for the case of $R = 5.67$ is not so favorable. Except for a small area of new convergent combinations for $\beta \approx 0.03$ and lower, the number of convergent algorithm choices is reduced significantly.

We address the overall performance of under-relaxation for various values of the under-relaxation parameter ω in the Gauss-Seidel coupling using two measures presented in Figure A.2. The percentage of converged cases from all combinations of (α, β) are presented as functions of ω for $R = 0$ and $R = 5.67$ in Figure A.2(a). Of those converged cases, the average number of iterations needed to solve the problem to a specified level of accuracy are shown in Figure A.2(b), also as functions of the under-relaxation parameter.

Generally, the convergence characteristics for different levels of under-relaxation shown in Figure A.2(a) are similar to those results for $\omega = 0.1$ shown by Figure A.1. For the linear problem ($R = 0$), under-relaxation improves the likelihood of convergence over the unrelaxed algorithm for all values of ω , except for the lowest value of $\omega = 0.05$. However, the number of iterations needed to obtain a converged solution rises significantly as the value of ω decreases, as shown in Figure A.2(b). For the nonlinear problem with $R = 5.67$, Figure A.2(a) shows that fewer combinations of coupling parameters converge compared to the unrelaxed iterations, and smaller values of ω further reduce the robustness of the under-

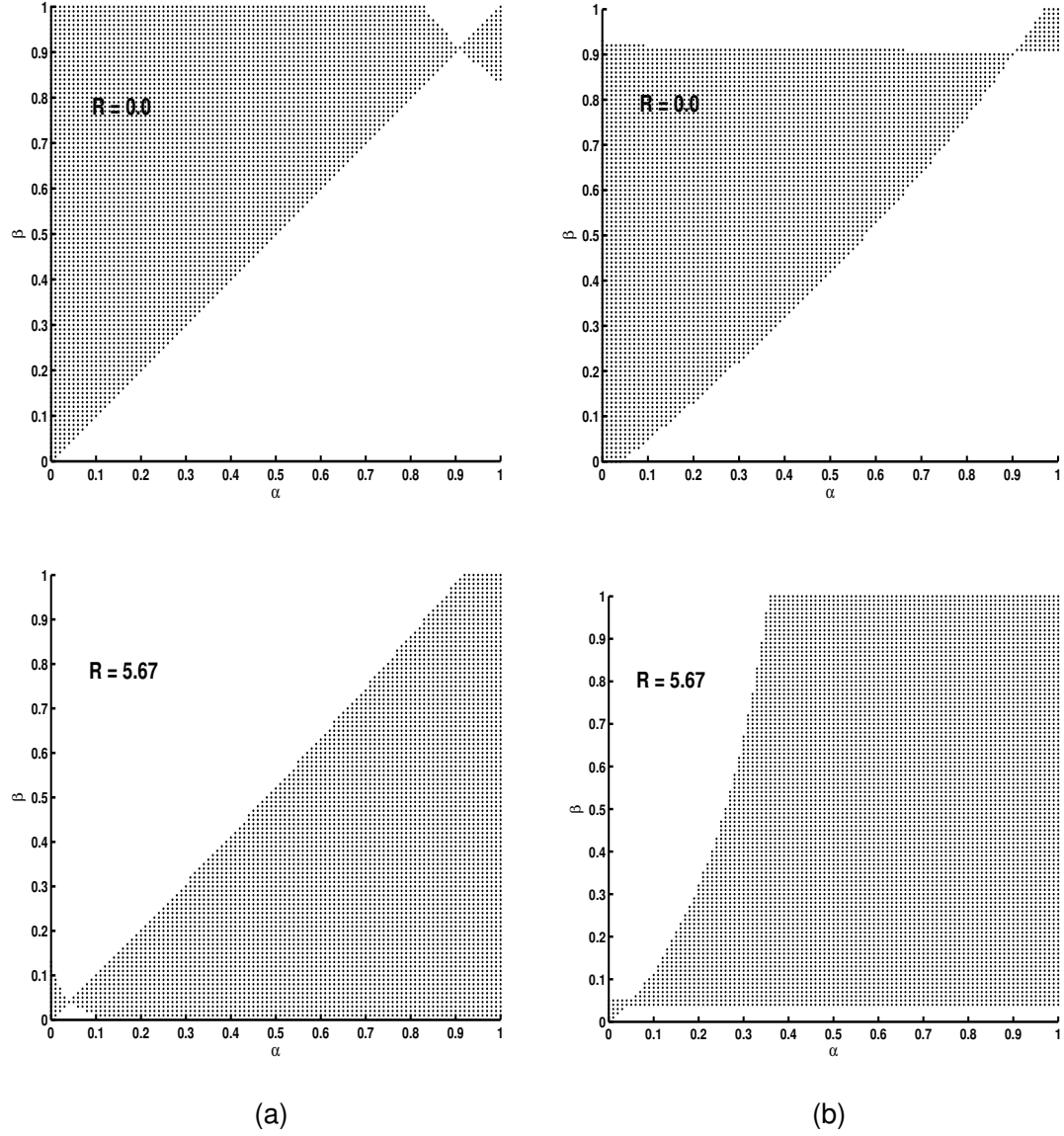
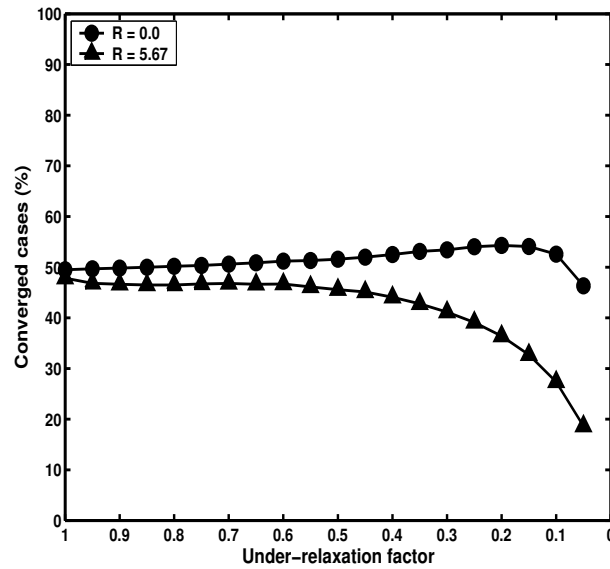
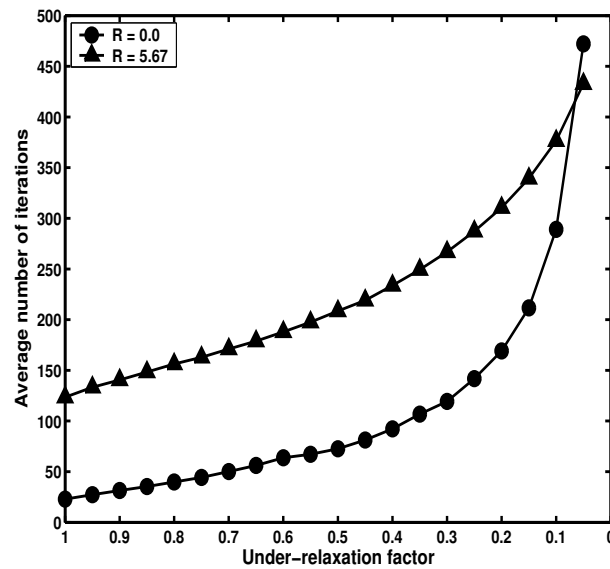


Figure A.1: Comparison between convergence of the one-dimensional model for $R = 0$ and 5.67 as a function of the coupling parameters α and β , for: (a) unrelaxed iterations, $\omega = 1$; and (b) under-relaxed iterations $\omega = 0.1$. Light areas represent algorithms that will converge to the solution for a suitable initial guess. Gray areas represent algorithms that will not converge for any initial guess to the problems solution.



(a)



(b)

Figure A.2: Performance measures of the block Gauss-Seidel iterations as a function of under-relaxation parameter, ω : (a) percentage of converged cases from all choices of α and β ; (b) average number of iterations needed for those algorithms to converge to the solution.

relaxed algorithm. Even worse, the decreased robustness of the under-relaxed algorithm is accompanied by increasing solution costs, as shown in Figure A.2(b).

A.5 Conclusions

We have presented an under-relaxed, block Gauss-Seidel iteration approach for modular, coupled models representing steady-state crystal growth from the melt. These coupled models are based on exchanging a mixture of temperatures and heat fluxes across a physical boundary separating two model domains. Each domain features its own, independent model that only interacts along a shared boundary. The primary goal of this chapter was to assess whether under-relaxation provides for better robustness, in terms of overall model convergence, compared to the unrelaxed iterative algorithms developed previously by us [135, 136, 196].

Unfortunately, under-relaxation does not provide a panacea for the ills present in our original algorithm. Notably, there remain conditions where the under-relaxed iteration sequence will *never* converge for certain values of the coupling parameters α and β , regardless of the value of the initial guess. While under-relaxation slightly improves the convergence characteristics for the linear model problem considered here, the improvement is countered by more expensive computations needed to compute a converged global solution. For the nonlinear model problem featuring radiative fluxes—a case far more relevant for the high-temperature melt growth processes of interest—under-relaxation proves to be *less* robust than the original formulation. To add insult to injury for this case, under-relaxation always resulted in a greater number of iterations and increased computational effort needed for solution.

Clearly, a better approach is needed for reliable and efficient modular coupling strategies for crystal growth models. Toward this goal, we return to our prior assertion that a monolithic model that employs Newton’s method will result in robust performance. We base this statement on the well-known fact that, unlike other fixed-point iterations, Newton’s method is guaranteed to converge to a solution provided that the solution exists and the initial guess for the iteration is close enough to it [39]. We believe that the key to a successful coupling strategy within the modular framework will be to employ a method that can well approximate the full Newton iteration. The interested reader is referred back to Chapter 5 of this thesis for an in-depth discussion on how Newton’s method can be approximated to solve a coupled crystal growth model.

Appendix B

Alternative Derivation of Approximate Block Newton Method

We present an alternative derivation of the Approximate Block Newton method as derived by Matthies and Steindorf [115,116] as a companion to the derivation presented in Chapter 5. This derivation is based on using arbitrary solution methods for the subsolvers $\mathcal{F}$ and $\mathcal{G}$ to solve time-dependent coupled problems. As the algorithm is written in a general form, steady-state problems can be solved in the same manner. The same notation of matrices denoted by boldface upper-case Roman letters, vectors by boldface lower-case Roman letters, and scalar quantities are represented by lower-case Greek letters is used here.

As in Chapter 5, we are interested in solving two coupled models represented by the residual equations

$$\mathbf{r}_1(\mathbf{x}, \tilde{\mathbf{y}}) = \mathbf{0} \tag{B.1}$$

$$\mathbf{r}_2(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbf{0} \tag{B.2}$$

where $\mathbf{r}_i$ is a residual vector associated with problem i , $\mathbf{x}$ denotes the unknowns of problem 1, and $\mathbf{y}$ denotes the unknowns of problem 2. The tilde indicates that these variables are from the other model, and fixed. We begin by assuming each subproblem has its own solver, and that these solvers can be written in fixed-point form with convergent internal

iterations

$$\mathbf{x}^{(k+1)} = \mathcal{F}(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}) \quad (\text{B.3})$$

$$\mathbf{y}^{(k+1)} = \mathcal{G}(\tilde{\mathbf{x}}, \mathbf{y}^{(k)}) \quad (\text{B.4})$$

where the solvers of $\mathbf{r}_1$ and $\mathbf{r}_2$ are denoted by the script letters $\mathcal{F}$ and $\mathcal{G}$, respectively. The variables designated with the tilde are held constant during the iteration. The index k indicates the *global*, or outer, iteration number corresponding to a converged solution of each subproblem. Each subsolver performs its own internal iterations to calculate the updated global iterate, but we do not explicitly denote these iterations.

Instead of taking Equations (B.1) and (B.2) as the equations we wish to solve, we actually seek the solution of the combined system

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{x} - \mathcal{F}(\mathbf{x}, \mathbf{y}) = \mathbf{0} \quad (\text{B.5})$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{y} - \mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathbf{0} \quad (\text{B.6})$$

which are just rewritten forms of the fixed-point forms of Equation (B.3) and (B.4). The solvers $\mathcal{F}$ and $\mathcal{G}$ are used to solve the above set of equations using an ABN method based on the work by Chan [30]. First we apply Newton's method to Equations (B.5) and (B.6) at the current position $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ to get

$$\begin{bmatrix} \mathbf{f}_{\mathbf{x}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) & \mathbf{f}_{\mathbf{y}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ \mathbf{g}_{\mathbf{x}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) & \mathbf{g}_{\mathbf{y}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{bmatrix} \quad (\text{B.7})$$

where we use the short-hand notation of

$$\begin{aligned} \mathbf{f}_{\mathbf{x}} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & \mathbf{f}_{\mathbf{y}} &= \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \\ \mathbf{g}_{\mathbf{x}} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}} & \mathbf{g}_{\mathbf{y}} &= \frac{\partial \mathbf{g}}{\partial \mathbf{y}} \end{aligned}$$

to represent the partial derivatives of the vectors $\mathbf{f}$ and $\mathbf{g}$ with respect to vectors $\mathbf{x}$ and $\mathbf{y}$ evaluated at $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$.

Using Block-Gauss elimination to the system of equations yields

$$\begin{bmatrix} \mathbf{f}_{\mathbf{x}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) & \mathbf{f}_{\mathbf{y}}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ 0 & \mathbf{S}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ \mathbf{r}_s(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{bmatrix} \quad (\text{B.8})$$

where

$$\mathbf{S} \equiv \mathbf{g}_{\mathbf{y}} - \mathbf{g}_{\mathbf{x}} \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}_{\mathbf{y}} \quad (\text{B.9})$$

is the Schur complement matrix, and

$$\mathbf{r}_s \equiv \mathbf{g} - \mathbf{g}_{\mathbf{x}} \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f} \quad (\text{B.10})$$

is its modified right-hand-side. These last two equations can be further simplified by letting

$$\mathbf{C} \equiv \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}_{\mathbf{y}}, \quad (\text{B.11})$$

$$\mathbf{q} = -\mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}. \quad (\text{B.12})$$

In the derivation presented in Chapter 5 we had inverses of the block Jacobian matrices $\mathbf{J}_{11}^{-1}$ and $\mathbf{J}_{22}^{-1}$. Here we have the inverses $\mathbf{f}_{\mathbf{x}}^{-1}$ and $\mathbf{g}_{\mathbf{y}}^{-1}$. As before, these inverses do not need to be explicitly computed as will be explained.

From these equations, we have the following algorithm:

1. Solve $\mathbf{f}_{\mathbf{x}} \mathbf{q} = -\mathbf{f}$ for $\mathbf{q}$
2. Compute the modified right hand side $\mathbf{r}_s = \mathbf{g} + \mathbf{g}_{\mathbf{x}} \mathbf{q}$
 - A. Compute the Schur complement $\mathbf{S} = \mathbf{g}_{\mathbf{y}} - \mathbf{g}_{\mathbf{x}} \mathbf{C}$
 - B. Solve $\mathbf{f}_{\mathbf{x}} \mathbf{C} = \mathbf{f}_{\mathbf{y}}$ for $\mathbf{C}$
3. Solve $\mathbf{S} \Delta \mathbf{y} = -\mathbf{r}_s$ for $\Delta \mathbf{y}$
4. Compute $\Delta \mathbf{x} = \mathbf{q} - \mathbf{C} \Delta \mathbf{y}$

We will now explain each step in detail.

In the first step, we wish to solve the equation

$$\mathbf{f}_{\mathbf{x}} \mathbf{q} = -\mathbf{f} \quad (\text{B.13})$$

for vector $\mathbf{q}$. This step can be viewed as one Newton iteration to solve the equation $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)}) = 0$ for $\mathbf{x}$. To see this, we perform a Taylor series expansion of $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)})$ about

$\mathbf{x}^{(k)}$ and evaluated at $\mathbf{x}^{(k)} + \mathbf{q}$

$$\mathbf{f}(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) = \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \mathbf{f}_{\mathbf{x}} \left\{ \mathbf{x}^{(k)} + \mathbf{q} - \mathbf{x}^{(k)} \right\} + \dots \quad (\text{B.14})$$

Ignoring higher order terms are rearranging, we then have

$$\mathbf{x}^{(k)} + \mathbf{q} \approx \mathbf{x}^{(k)} - \mathbf{f}_{\mathbf{x}}^{-1}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \approx \hat{\mathbf{x}} \quad (\text{B.15})$$

where $\hat{\mathbf{x}}$ is the solution of $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)}) = 0$. We can approximate $\hat{\mathbf{x}}$ by using the solver $\mathcal{F}(\mathbf{x}, \mathbf{y})$. At convergence of

$$z^{(j+1)} = \mathcal{F}(z^{(j)}, \mathbf{y}^{(k)}), \quad j = 0, 1, \dots, m-1, \quad m \geq 1 \quad (\text{B.16})$$

where $z^{(0)} = \mathbf{x}^{(k)}$, j is the internal iteration counter, and m is an upper iteration boundary. We can then say that

$$z^{(m)} \approx \hat{\mathbf{x}} \quad (\text{B.17})$$

and

$$\mathbf{q} = z^{(m)} - \mathbf{x}^{(k)}. \quad (\text{B.18})$$

If $m = 1$ and $\mathcal{F}$ is based on Newton's method, this solution for $\mathbf{q}$ is the as in Chapter 5. Mathematically written

$$\mathbf{q} = \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathbf{x}^{(k)} \quad (\text{B.19})$$

where solver $\mathcal{F}$ is iterated m times.

In the second step, we are solving the modified right hand side of the Schur complement problem, $\mathbf{r}_S = \mathbf{g} + \mathbf{g}_{\mathbf{x}} \mathbf{q}$. The matrix $\mathbf{g}_{\mathbf{x}}$ is analogous to the Jacobian matrix J_{21} from Chapter 5, and we treat it similarly using the Jacobian-free method. To see this, we use a Taylor expansions of $\mathbf{g}(\mathbf{x}, \mathbf{y}^{(k)}) = 0$ in the $\mathbf{x}$ -direction about $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ and evaluated at $(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)})$

$$\mathbf{g}(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) \approx \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \mathbf{g}_{\mathbf{x}}|_{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}} \mathbf{q} + \dots \quad (\text{B.20})$$

Rearranging, we see that

$$\mathbf{r}_S \approx \mathbf{g}(\mathbf{x}^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) \quad (\text{B.21})$$

or

$$\mathbf{r} = \mathbf{y}^{(k)} - \mathcal{G}(x^{(k)} + \mathbf{q}, \mathbf{y}^{(k)}) \quad (\text{B.22})$$

where $\mathcal{G}$ may be iterated internally multiple times.

In the third step, we need to compute the Schur complement matrix $\mathbf{xS}$ to find our update vector $\Delta\mathbf{y}$. As in Chapter 5, the Schur complement matrix is not needed explicitly, but simply its action on a vector. Therefore, an iterative solution technique, such as a Krylov subspace method, can be used to solve the linear equation $\mathbf{S}\Delta\mathbf{y} = -\mathbf{r}_S$ for $\Delta\mathbf{y}$. A suitable candidate solution is constructed from the Krylov space defined by $K \equiv \{\mathbf{w}, \mathbf{S}\mathbf{w}, \mathbf{S}^2\mathbf{w}, \dots, \mathbf{S}^m\mathbf{w}\}$ where $\mathbf{w} = -\mathbf{r}_S$. The JFNK method is used to approximate this step in a similar manner as discussed in Theorem 5.1 in Chapter 5. We begin with the definition of $\mathbf{S}$:

$$\mathbf{S}\epsilon\mathbf{w} = \frac{1}{\epsilon}\mathbf{S}\epsilon\mathbf{w} = \frac{1}{\epsilon} \{ \mathbf{g}_y\epsilon\mathbf{w} - \mathbf{g}_x\mathbf{C}\epsilon\mathbf{w} \} \quad (\text{B.23})$$

The right-hand-side of the above equation can be approximated by the Taylor series of $\mathbf{g}(\mathbf{x}, \mathbf{y}) = 0$ about $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$

$$g(\mathbf{x}, \mathbf{y}) \approx g(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) + \mathbf{g}_x|_{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}} \left\{ \mathbf{x} - \mathbf{x}^{(k)} \right\} + \mathbf{g}_y|_{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}} \left\{ \mathbf{y} - \mathbf{y}^{(k)} \right\} + \dots \quad (\text{B.24})$$

We then evaluate the above equation at $(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)} + \epsilon\mathbf{w})$, ignore higher order terms, and rearrange to get

$$\mathbf{g}_y\epsilon\mathbf{w} - \mathbf{g}_x\mathbf{C}\epsilon\mathbf{w} \approx \mathbf{g}(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) - \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \quad (\text{B.25})$$

which gives

$$\frac{1}{h}\mathbf{S}\epsilon\mathbf{w} \approx \frac{1}{h} \left\{ \mathbf{g}(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) - \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\}. \quad (\text{B.26})$$

$$\frac{1}{\epsilon}\mathbf{S}(\epsilon\mathbf{w}) = \mathbf{w} + \frac{1}{\epsilon} \left\{ \mathcal{G}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \mathcal{G}(\mathbf{x}^{(k)} - \mathbf{C}\epsilon\mathbf{w}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) \right\}. \quad (\text{B.27})$$

To complete Step 3, we need the matrix $\mathbf{C}$ multiplied by the scaled Krylov vector $\epsilon\mathbf{w}$. This step is evaluated similar to how vector $\mathbf{q}$ is approximated. Using the definition of $\mathbf{C}$, we get the following equation

$$\mathbf{C}\epsilon\mathbf{w} = \mathbf{f}_x^{-1}\mathbf{f}_y\epsilon\mathbf{w} \quad (\text{B.28})$$

where the derivative $\mathbf{f}_y$ can be approximated using finite differences from a Taylor series expansion such that

$$\mathbf{f}_y\epsilon\mathbf{w} \approx \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) - \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}). \quad (\text{B.29})$$

Equation B.28 becomes

$$\begin{aligned} \mathbf{C}\epsilon\mathbf{w} &= \mathbf{f}_{\mathbf{x}}^{-1} \left\{ \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) - \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\} \\ &\quad \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) - \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ &\quad \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) + \mathbf{q} \end{aligned} \quad (\text{B.30})$$

where the first term in right-hand-side looks like a Newton step in the $\mathbf{x}$ -direction to solve $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) = 0$ evaluated at $\mathbf{x}^{(k)}$. We reuse the vector $\mathbf{q}$ in the second term of the right-hand-side. We approximate the first term using solver $\mathcal{F}$. To see how this is done, we make a new placeholder vector $\mathbf{h} \equiv \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w})$ and subtract from $\mathbf{x}^{(k)}$ to get

$$\begin{aligned} \mathbf{x}^{(k)} - \mathbf{h} &= \mathbf{x}^{(k)} - \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) \\ \hat{\mathbf{x}} & \end{aligned} \quad (\text{B.31})$$

where $\hat{\mathbf{x}}$ is the solution to $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) = 0$. As in step 1, we can approximate $\hat{\mathbf{x}}$ by using the solver $\mathcal{F}(\mathbf{x}, \mathbf{y})$. At convergence of

$$z^{(j+1)} = \mathcal{F}(z^{(j)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}), \quad j = 0, 1, \dots, m-1, \quad m \geq 1 \quad (\text{B.32})$$

where $z^{(0)} = \mathbf{x}^{(k)}$, j is the internal iteration counter, and m is an upper iteration boundary. The solution vector $\hat{\mathbf{x}}$ is approximated as $\hat{\mathbf{x}} \approx \mathbf{z}^{(m)}$. When $m = 1$, we revert back to the derivation presented in Chapter 5. Vector $\mathbf{h}$ is approximated as

$$\mathbf{h} = \mathbf{f}_{\mathbf{x}}^{-1} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) \approx \mathbf{x}^{(k)} - \hat{\mathbf{x}} \quad (\text{B.33})$$

and the quantity $\mathbf{C}\epsilon\mathbf{w}$ becomes

$$\mathbf{C}\epsilon\mathbf{w} \approx \mathbf{h} + \mathbf{q}. \quad (\text{B.34})$$

This equation can be rewritten in terms of solver $\mathcal{F}$ to get

$$\mathbf{C}\epsilon\mathbf{w} = \mathbf{x}^{(k)} - \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \epsilon\mathbf{w}) + \mathbf{q} \quad (\text{B.35})$$

where $\mathcal{F}$ may be iterated internally multiple times. The entirety of Step 3 is iterated until a satisfactory approximation of the update vector $\Delta\mathbf{y}$ is attained.

In the final step, Step 4, for update vector $\Delta\mathbf{x} = \mathbf{q} - \mathbf{C}\Delta\mathbf{y}$, we need the action of $\mathbf{C}$ on vector $\Delta\mathbf{y}$. Calculating $\mathbf{C}\Delta\mathbf{y}$ is done in the same manner as calculating $\mathbf{C}\epsilon\mathbf{w}$. We again

approximate the derivative $\mathbf{f}_y$ using a finite difference

$$\mathbf{C}\Delta\mathbf{y} = \mathbf{f}_x^{-1}\mathbf{f}_y\Delta\mathbf{y} \quad (\text{B.36})$$

$$\approx \mathbf{f}_x^{-1} \left\{ \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) - \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \right\} \quad (\text{B.37})$$

$$= \mathbf{f}_x^{-1}\mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) + \mathbf{q}. \quad (\text{B.38})$$

Therefore, $\Delta\mathbf{x}$ is

$$\Delta\mathbf{x} = -\mathbf{f}_x^{-1}\mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) \quad (\text{B.39})$$

which is a Newton step to solve $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) = 0$ evaluated at $\mathbf{x}^{(k)}$. Adding the above equation from $\mathbf{x}^{(k)}$ we have

$$\mathbf{x}^{(k)} + \Delta\mathbf{x} = \hat{\mathbf{x}} = \mathbf{x}^{(k)} - \mathbf{f}_x^{-1}\mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) \quad (\text{B.40})$$

where $\hat{\mathbf{x}}$ is the solution to $\mathbf{f}(\mathbf{x}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) = 0$. As in Step 1, we can approximate $\hat{\mathbf{x}}$ by using the solver $\mathcal{F}(\mathbf{x}, \mathbf{y})$. At convergence of

$$z^{(j+1)} = \mathcal{F}(z^{(j)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}), \quad j = 0, 1, \dots, m-1, \quad m \geq 1 \quad (\text{B.41})$$

where $z^{(0)} = \mathbf{x}^{(k)}$, j is the internal iteration counter, and m is an upper iteration boundary. The solution $\hat{\mathbf{x}}$ is approximated as $\hat{\mathbf{x}} \approx z^{(m)}$. When $m = 1$, we revert back to the derivation presented in Chapter 5. The update vector $\Delta\mathbf{x}$ can then be approximated as

$$\Delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}^{(k)} \quad (\text{B.42})$$

or in terms of solver $\mathcal{F}$ as

$$\Delta\mathbf{x} = \mathcal{F}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)} + \Delta\mathbf{y}) - \mathbf{x}^{(k)} \quad (\text{B.43})$$

where $\mathcal{F}$ can be iterated internally multiple times. The derivation presented here shows that the procedure outlined in Chapter 5 is equivalent to that of Matthies and Steindorf [115, 116].

Bibliography

- [1] P. Adornato and R. A. Brown. Convection and segregation in directional solidification of dilute and non-dilute binary alloys: Effects of ampoule and furnace design. *J. Cryst. Growth*, 80:155–190, 1987.
- [2] K. Arafune, K. Kodera, A. Kinoshita, and A. Hirata. Control of crystal-melt interface shape during horizontal Bridgman growth of InSb crystal using solutal Marangoni convection. *J. Cryst. Growth*, 249:429–436, 2003.
- [3] J. Argyris, I. S. Doltsinis, P. Pimenta, and H. W”utenberg. Thermomechanical response of solids at high strains: natural approach. *Comput. Methods Appl. Mech. Eng.*, 32:3–57, 1982.
- [4] S. Artlich and W. Mackens. Newton-coupling of fixed point iterations. In W. Hackbusch and G. Wittum, editors, *Numerical Treatment of Coupled Systems*. Braunschweig, Vieweg, 1995.
- [5] N. Aslam, E. D. Jones, T. C. Q. Noakes, J. B. Mullin, and A. F. W. Willoughby. The diffusion of zinc in cadmium telluride. *J. Cryst. Growth*, 117:249–253, 1992.
- [6] Y. Azuma, N. Usami, T. Ujihara, K. Fujiwara, G. Sazaki, Y. Murakami, and K. Nakajima. Growth of SiGe bulk crystals with uniform composition by utilizing feedback control system of the crystal-melt interface position for precise control of the growth temperature. *J. Cryst. Growth*, 250:298–304, 2003.
- [7] R. Backofen, M. Kurz, and G. Müller. Process modeling of the industrial VGF growth process using the software package CrysVUN++. *J. Cryst. Growth*, 211:202–206, 2000.

- [8] R. Backofen, M. Kurz, and G. Müller. Process modelling of the industrial VGF crystal growth process using the software package CrysVUn++. *J. Cryst. Growth*, 211:202–206, 2000.
- [9] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II *Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org>.
- [10] K. J. Bathe, H. Zhang, and M. H. Wang. Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. *Comput. Struct.*, 56:193–213, 1995.
- [11] C. Batur, W. Duval, and R. Bennett. Performance of Bridgman furnace operating under projective control. In *Proceedings of the American Control Conference*, pages 4101–4105, 1999.
- [12] C. Batur, A. Srinivasan, W. M. Duval, N. Singh, and D. Golovaty. On-line control of solid-liquid interface by state feedback. *J. Cryst. Growth*, 205:395–409, 1999.
- [13] J. Baumgartl, W. Budweisr, G. Müller, and G. Neumann. Studies of buoyancy driven convection in a vertical cylinder with parabolic temperature profile. *J. Cryst. Growth*, 97:9–17, 1989.
- [14] J. Baumgartl, A. Bune, K. Koai, G. Müller, and A. Seidl. Global simulation of heat transport, including melt convection in a Czochralski crystal growth process - combined finite element / finite volume approach. *Mater. Sci. Eng., A*, A173:9–13, 1993.
- [15] T. Belytschko and R. Mullen. Two-dimensional fluid-structure impact computations with regularization. *Comput. Methods Appl. Mech. Eng.*, 27:139–154, 1987.
- [16] K. W. Benz and P. Dold. Crystal growth under microgravity: present results and future prospects towards the International Space Station. *J. Cryst. Growth*, 237-239:1638–1645, 2002.
- [17] H. Bénard. Les Tourbillons cellulaires dans une nappe liquide transportant de la chaleur par convection en régime permanent. *Ann. Chim. Phys.*, 23:62–144, 1901.
- [18] E. Billig. Some defects in crystals grown from the melt. I. defects caused by thermal stresses. *Proc. Roy. Soc. London, Ser. A, Math. Phys. Sci.*, 235:37–55, 1956.

- [19] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, 1960.
- [20] S. Brandon and J. Derby. Heat transfer in vertical Bridgman growth of oxides: Effects of conduction, convection, and internal radiation. *J. Cryst. Growth*, 121:473–494, 1992.
- [21] J. C. Brice. *Crystal Growth Processes*. Halsted Press, 1986.
- [22] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Methods Appl. Mech. Eng*, 32:199–259, 1982.
- [23] R. A. Brown. Theory of transport processes in single crystal growth from the melt. *AIChE J.*, 34:881–910, 1988.
- [24] R. A. Brown and D. H. Kim. Modeling of directional solidification: from Scheil to detailed numerical simulation. *J. Cryst. Growth*, 109:50–65, 1991.
- [25] J. A. Burton, R. C. Prim, and W. P. Schlichter. The distributions of solute in crystals grown from the melt: theoretical. *J. Chem. Phys*, 21:1987–1991, 1953.
- [26] J. F. Butler, C. L. Lingren, and F. P. Doty. $\text{Cd}_{1-x}\text{Zn}_x\text{Te}$ gamma ray detectors. *IEEE Trans. Nucl. Sci.*, 39:605–609, 1992.
- [27] P. Capper, J. C. Brice, C. L. Jones, J. J. G. Gosney, C. K. Ard, I. Kenworthy, and A. Clark. Interfaces and flow regimes in ACRT grown $\text{Cd}_x\text{Hg}_{1-x}\text{Te}$ crystals. *J. Cryst. Growth*, 89:171–176, 1988.
- [28] J. R. Carruthers. Crystal growth in a low gravity environment. *J. Cryst. Growth*, 263:629–644, 2004.
- [29] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Eng. Comput.*, 13(6):4–30, 1996.
- [30] T. F. Chan. An approximate Newton method for coupled nonlinear systems. *SIAM Journal of Numerical Analysis*, 22(5):904–913, 1985.
- [31] C. E. Chang and W. R. Wilcox. Control of interface shape in the vertical Bridgman-Stockbarger technique. *J. of Cryst. Growth*, 21:135–140, 1974.

- [32] C. J. Chang and R. A. Brown. Radial segregation induced by natural convection and melt/solid interface shape in vertical Bridgman growth. *J. Cryst. Growth*, 63(343-364), 1983.
- [33] L.-Y. Chin and F. M. Carlson. Finite element analysis of the control of interface shape in Bridgman crystal growth. *J. Cryst. Growth*, 62:561–567, 1983.
- [34] K. N. Christodoulou and L. E. Scriven. Discretization of free surface flows and other moving boundary problems. *J. Comput. Phys.*, 99:39–55, 1992.
- [35] G. Comini, O. Saro, and M. Manzan. A physical approach to finite element modeling of coupled conduction and convection. *Numer. Heat Transfer, Part B*, 24:243–26, 1993.
- [36] R. Cònsul, C. D. Pérez-Segarra, K. Claramunt, J. Cadafalch, and A. Oliva. Detailed numerical simulation of laminar flames by a parallel multiblock algorithm using loosely coupled computers. *Combust. Theory Modell.*, 7:525–544, 2003.
- [37] S. R. Coriell and R. F. Sekerka. Lateral solute segregation during unidirectional solidification of a binary alloy with a curved solid-liquid interface. *J. Cryst. Growth*, 46:479–482, 1979.
- [38] S. R. Coriell and R. F. Sekerka. Lateral solute segregation during unidirectional solidification of a binary alloy with a curved solid-liquid interface, II. large departures from planarity. *J. Cryst. Growth*, 54:167–175, 1981.
- [39] G. Dahlquist and Å. Björk. *Numerical Methods*. Prentice-Hall, 1974.
- [40] B. Dai. *Finite element modeling solution crystal growth at a meso-scale*. PhD thesis, University of Minnesota, 2005.
- [41] H. T. Davis and K. T. Thomson. *Linear Algebra and Linear Operators in Engineering with Applications in Mathematica*. Academic Press, New York, 2000.
- [42] J. de Santos. *Two-Phase CoCurrent Downflow Through Constricted Passages*. PhD thesis, University of Minnesota, 1991.
- [43] J. J. Derby. Macroscopic transport processes during the growth of single crystals from the melt. In *Science and Technology of Crystal Growth*, pages 97–110. Kluwer Academic Publishers, Dordrecht, 1995.

- [44] J. J. Derby, Y.-I. Kwon, A. Pandey, P. Sonda, A. Yeckel, T. Jung, and G. Müller. Developing quantitative, multiscale models for microgravity crystal growth. *Ann. N.Y. Acad. Sci.*, 1077:124–145, 2006.
- [45] J. J. Derby, L. Lun, and A. Yeckel. Strategies for the coupling of global and local crystal growth models. *J. Cryst. Growth*, 303:114–123, 2007.
- [46] J. J. Derby and A. Yeckel. Modeling of crystal growth. In G. Müller, J.-J. Métois, and P. Rudolph, editors, *Crystal Growth - From Fundamentals to Technology*. Elsevier Science, Amsterdam, 2004.
- [47] K. P. Dharmasena and H. N. G. Wadley. Eddy current sensing of vertical Bridgman growth of CdZnTe. *J. Cryst. Growth*, 172:337–349, 1997.
- [48] K. P. Dharmasena and H. N. G. Wadley. Modeling multifrequency eddy current sensor interaction during vertical Bridgman growth of semiconductors. *Rev. Sci. Instruments*, 70:3125–3142, 1999.
- [49] E. Divo, E. Steinthorsson, F. Rodriguez, A. J. Kassab, and J. S. Kapat. Glenn-HT/BEM conjugate heat transfer solver for large-scale turbomachinery models. *NASA/CR2003-212195*, April 2003.
- [50] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interaction. *Comput. Methods Appl. Mech. Eng.*, 33:193–213, 1995.
- [51] F. P. Doty, J. F. Butler, J. F. Schetzina, and K. A. Bowers. Properties of CdZnTe crystals grown by a high pressure Bridgman method. *J. Vac. Sci. Technol., B*, 10:1418, 1992.
- [52] E. Dowell and K. C. Hall. Modeling of fluid structure interaction. *Annu. Rev. Fluid Mech.*, 33:445–490, 2001.
- [53] G. Dubini, R. Pietrabissi, and F. M. Montevocchi. Fluid-structure interaction in bio-fluid mechanics. *Med. Eng. Phys.*, 17:609–617, 1995.
- [54] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, 1989.

- [55] F. Dupret, P. Nicodème, Y. Ryckmans, P. Wouters, and M. J. Crochet. Global modelling of heat transfer in crystal growth furnaces. *Int. J. Heat Mass Transfer*, 33:1849–1871, 1990.
- [56] P. S. Dutta, H. L. Bhat, and V. Kumar. Numerical analysis of melt-solid interface shapes and growth rates of gallium antimonide in a single-zone vertical Bridgman furnace. *J. Cryst. Growth*, 154:213–222, 1995.
- [57] P. S. Dutta, K. S. Sangunni, H. L. Bhat, and V. Kumar. Growth of gallium antimonide by vertical Bridgman technique with planar crystal-melt interface. *J. Cryst. Growth*, 141:44–50, 1994.
- [58] K. Edwards and J. J. Derby. Understanding horizontal Bridgman shelf growth of cadmium telluride and cadmium zinc telluride I. heat and mass transfer. *J. Cryst. Growth*, 179:120–132, 1997.
- [59] K. Edwards and J. J. Derby. Understanding horizontal Bridgman shelf growth of cadmium telluride and cadmium zinc telluride II. thermoelastic stresses. *J. Cryst. Growth*, 179:133–143, 1997.
- [60] H. Emmerich. *The Diffuse Interface Approach in Materials Science: Thermodynamic Concepts and Applications of Phase-Field Models*. Springer-Verlag, Heidelberg, 2003.
- [61] C. Farhat, M. Lesoinne, and P. LeTallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput. Methods Appl. Mech. Eng.*, 157:95–114, 1998.
- [62] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Comput. Methods Appl. Mech. Eng.*, 190:3247–3270, 2001.
- [63] B. A. Finlayson and L. E. Scriven. The method of weighted residuals and its relation to certain variational principles for the analysis of transport processes. *Chem. Eng. Sci.*, 20:395–404, 1965.
- [64] A. G. Fischer. Techniques for melt-growth of luminescent semiconductor crystals under pressure. *J. Electrochem. Soc.*, 117:41C–47C, 1970.
- [65] M. C. Flemings. *Solidification Processes*. McGraw-Hill, Inc., New York, 1974.

- [66] T.-W. Fu and W. R. Wilcox. Influence of insulation on stability of interface shape and position in the vertical Bridgman-Stockbarger technique. *J. Cryst. Growth*, 48:416–424, 1980.
- [67] J. P. Garandet, S. Corre, S. Gavoille, J. J. Favier, and J. I. D. Alexander. On the effect of gravity perturbations on composition profiles during Bridgman crystal growth in space. *J. Cryst. Growth*, 165:471–481, 1996.
- [68] V. M. Glazov, S. N. Chizhevskaya, and N. N. Glagoleva. *Liquid semiconductors*, volume 2 of *Monographs in semiconductor physics*. Plenum Press, New York, 1969.
- [69] V. V. Godlevsky, J. J. Derby, and J. R. Chelikowsky. *Ab initio* molecular dynamics simulation of liquid CdTe and GaAs: semiconducting versus metallic behavior. *Phys. Rev. Lett.*, 81:4959–4962, 1998.
- [70] P. M. Gresho. Some interesting issues in incompressible fluid dynamics, both in the continuum and in numerical simulation. In *Advances in Applied Mechanics*. Academic Press, 1992.
- [71] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method, Vol. 2: Isothermal Laminar Flow*. John Wiley & Sons Ltd., New York, 1998.
- [72] W. Hackbusch, B. N. Khoromskij, and R. Kriemann. Direct Schur complement method by domain decomposition on H-matrix approximation. *Computing and Visualization in Science*, 8:179–188, 2005.
- [73] B. Hassan, D. Kuntz, and D. L. Potter. Coupled fluid/thermal prediction of ablating hypersonic vehicles. *AIAA Paper 98-0168*, 1998.
- [74] P. Hood. Frontal solution program for unsymmetric matrices. *Int. J. Numer. Methods Eng.*, 10:379–399, 1976.
- [75] A. Horowitz, D. Gazit, J. Makovsky, and L. Ben-Dor. Bridgman growth of Rb_2MnCl_4 . *J. Cryst. Growth*, 61:317–322, 1983.
- [76] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
- [77] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comp. Methods Appl. Mech. Eng.*, 73:173–189, 1989.

- [78] D. T. J. Hurle. *Crystal Growth of Electronic Materials*. Elsevier Science Publishers, 1985.
- [79] R. B. James, T. E. Schlesinger, J. C. Lund, and M. Schieber. $\text{Cd}_{1-x}\text{Zn}_x\text{Te}$ spectrometers for gamma and X-ray applications. In T. E. S. and R. B. James, editor, *Semiconductors for Room Temperature Nuclear Detector Applications*, volume 43 of *Semiconductors and Semimetals*, pages 335–381. Academic Press Inc, San Diego, CA, 1995.
- [80] T. Jasinski and A. F. Witt. On control of the crystal-melt interface shape during growth in a vertical Bridgman configuration. *J. Cryst. Growth*, 71:295–304, 1987.
- [81] A. Jeffrey. *Handbook of Mathematical Formulas and Integrals*. Academic Press, New York, 1995.
- [82] A. S. Jordan, A. R. V. Neida, and J. W. Nielsen. A comparative study of thermal stress induced dislocation generation in pulled GaAs, InP, and Si crystals. *J. Appl. Phys.*, 52:3331–3336, 1981.
- [83] O. W. D. Jr., C. T. Tsai, and R. J. DeAngelis. Dislocation dynamics during of web type silicon ribbon. *J. Cryst. Growth*, 82:50–59, 1987.
- [84] A. J. Kassab and M. N. Aliabadi, editors. *Advances in Boundary Elements: Coupled Field Problems*. Computational Mechanics, Boston, 2001.
- [85] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In *Applications of bifurcation theory*, pages 159–384. Academic Press, New York, 1997.
- [86] M. Kestigian, A. B. Bollong, J. J. Derby, H. L. Glass, K. Harris, L. Hettich, P. K. Liao, P. Mitra, P. Norton, and H. Wadley. Cadmium zinc telluride substrate growth, characterization, and evaluation. *J. Electron. Mater.*, 28:726–731, 1999.
- [87] D. H. Kim, P. M. Adornato, and R. A. Brown. Effect of vertical magnetic field on convection and segregation in vertical Bridgman crystal growth. *J. Cryst. Growth*, 89:339–356, 1988.
- [88] D. H. Kim and R. A. Brown. Modelling of the dynamics of HgCdTe growth by the vertical Bridgman method. *J. Cryst. Growth*, 114:441–434, 1991.
- [89] K. M. Kim, A. F. Witt, and H. C. Gatos. Crystal growth from the melt under destabilizing thermal gradients. *J. Electrochem. Soc.*, 119:1218–1226, 1972.

- [90] K. M. Kim, A. F. Witt, M. Lichtensteiger, and H. C. Gatos. Quantitative analysis of the effects of destabilizing vertical thermal gradients on crystal growth and segregation: Ga-doped Ge. *J. Electrochem. Soc.*, 125:475–480, 1978.
- [91] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004.
- [92] S. Kuppurao, S. Brandon, and J. J. Derby. Modeling the vertical Bridgman growth of cadmium zinc telluride, I: Quasi-steady analysis of heat transfer and convection. *J. Cryst. Growth*, 155:93–102, 1995.
- [93] S. Kuppurao, S. Brandon, and J. J. Derby. Modeling the vertical Bridgman growth of cadmium zinc telluride, II: Transient analysis of zinc segregation. *J. Cryst. Growth*, 155:103–111, 1995.
- [94] S. Kuppurao, S. Brandon, and J. J. Derby. Analysis of interrupted growth strategies for cadmium zinc telluride in an unseeded vertical Bridgman system. *J. Cryst. Growth*, 158:459–479, 1996.
- [95] S. Kuppurao and J. J. Derby. Designing thermal environments to promote convex interface shapes during the vertical Bridgman growth of cadmium zinc telluride. *J. Cryst. Growth*, 172:350–360, 1997.
- [96] M. Kurz, A. Pusztai, and G. Müller. Development of a new powerful computer code CrysVUN++ especially designed for fast simulation of bulk crystal growth processes. *J. Cryst. Growth*, 198/199:101–106, 1999.
- [97] M. Kurz, A. Pusztai, and G. Müller. Development of a new powerful computer code CrysVUn++ especially designed for fast simulation of bulk crystal growth processes. *J. Cryst. Growth*, 198:101–106, 1999.
- [98] Y.-I. Kwon and J. J. Derby. Modeling the coupled effects of interfacial and bulk phenomena during solution crystal growth. *J. Cryst. Growth*, 230:328–335, 2001.
- [99] I. Lakeenkov, V. B. Ufimtsev, N. I. Shmatov, and Y. F. Schelkin. Numerical simulation of vertical Bridgman growth of $\text{Cd}_{1-x}\text{Zn}_x\text{Te}$ melts. *J. Cryst. Growth*, 197:443–448, 1999.
- [100] C. W. Lan. Effects of ampoule rotation on flows and dopant segregation in vertical Bridgman crystal growth. *J. Cryst. Growth*, 197:983–991, 1999.

- [101] C. W. Lan. Effects of vibration on vertical zone-melting processing. *Int. J. Heat Mass Transfer*, 43:1987–1997, 2000.
- [102] C. W. Lan. Recent progress of crystal growth modeling and growth control. *Chem. Eng. Sci.*, 59:1437–1457, 2004.
- [103] C. W. Lan, I. F. Lee, and B. C. Yeh. Three-dimensional analysis of flow and segregation in vertical Bridgman crystal growth under axial and transversal magnetic fields. *J. Cryst. Growth*, 254:503–515, 2003.
- [104] C. W. Lan and M. C. Liang. Three-dimensional simulation of vertical zone-melting crystal growth: symmetry breaking to multiple states. *J. Cryst. Growth*, 208:327–240, 2000.
- [105] C. W. Lan, M. C. Liang, and J. H. Chian. Influence of ampoule rotation on three-dimensional convection and segregation in Bridgman crystal growth under imperfect growth conditions. *J. Cryst. Growth*, 212:340–351, 2000.
- [106] E. Larsson. Domain decomposition method for the Helmholtz equation in a multilayer domain. *SIAM J. Sci. Computing*, 20:1713–1731, 1999.
- [107] M. C. Liang and C. W. Lan. Three-dimensional convection and solute segregation in vertical Bridgman crystal growth. *J. Cryst. Growth*, 167:320–332, 1996.
- [108] D. Loghin and A. J. Wather. Schur complement preconditioners for the Navier-Stokes equations. *Int. J. Numer. Methods Fluids*, 40:403–412, 2002.
- [109] D. P. Lukanin, V. V. Kalaev, Y. N. Makarov, T. Wetzels, J. Virbulis, and W. von Ammon. Advances in the simulation of heat transfer and prediction of the melt-crystal interface shape in silicon CZ growth. *J. Cryst. Growth*, 266:20–27, 2004.
- [110] L. Lun, A. Yeckel, P. Daoutidis, and J. J. Derby. Decreasing lateral segregation in cadmium zinc telluride via ampoule tilting during vertical Bridgman growth. *J. Cryst. Growth*, 291:348–357, 2006.
- [111] L. Lun, A. Yeckel, J. J. Derby, and P. Daoutidis. Control of interface shape of cadmium zinc telluride grown via an electrodynamic gradient freeze furnace. In *Proceedings of IEEE 2007 Mediterranean Conference on Control and Automation (MED '07)*, Athens, Greece, June 27-29, 2007.

- [112] L. Lun, A. Yeckel, M. Reed, C. Szeles, P. Daoutidis, and J. J. Derby. On the effects of furnace gradients on interface shape during the growth of cadmium zinc telluride in EDG furnaces. *J. Cryst. Growth*, 290:35–43, 2006.
- [113] C. G. M. Marangoni. Sul principio della viscosità superficiale dei liquidi stabilita dal Signor J. Plateau (on the principle of surface viscosity of liquids by Mr. J. Plateau). *Il Nuovo Cimento, Ser. 2*, 5/6:239–273, 1871.
- [114] C. Martinez-Tomas, V. Muñoz, and R. Triboulet. Heat transfer simulation in a vertical Bridgman CdTe growth configuration. *J. Cryst. Growth*, 197:435–442, 1999.
- [115] H. G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Comput. Methods Appl. Mech. Eng.*, 195:2028–2049, 2006.
- [116] H. G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction. *Comput. Struct.*, 80:1991–1999, 2002.
- [117] J. Menck. An approximate Newton-like coupling of subsystems. *Z. Angew. Math. Mech.*, 82(2):101–114, 2002.
- [118] K. Mochizuki, K. Masumoto, and K. Miyazaki. Melt growth of CdTe single crystals with controlled deviation from stoichiometry. *Mater. Lett.*, 6:119–122, 1988.
- [119] E. M. Monberg, H. Brown, and C. E. Bonner. Dynamic gradient freeze growth of InP. *J. Cryst. Growth*, 94:109–114, 1989.
- [120] H. Morand and R. Ohayon. *Fluid-Structure Interaction*. John Wiley & Sons, Chichester, 1995.
- [121] S. A. Morton, R. B. Melville, and M. R. Visbal. Accuracy and coupling issues of aeroelastic Navier-Stokes solutions on deforming meshes. *J. Aircraft*, 35:798–805, 1998.
- [122] J. Mouro and P. LeTallec. Fluid structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Eng.*, 190:3039–3067, 2001.
- [123] G. Müller. Convective instabilities in melt growth configurations. *J. Cryst. Growth*, 128:26–36, 1993.

- [124] G. Müller. Modeling of crystal growth from the melt. In P. Vincenzini and A. Lami, editors, *Computational Modelling and Simulations of Materials*, pages 267–278. Techna Srl., 2003.
- [125] G. Müller, G. Neumann, and H. Matz. A two-Rayleigh-number model of buoyancy-driven convection in vertical melt growth configurations. *J. Cryst. Growth*, 84:36–49, 1987.
- [126] G. Müller, G. Neumann, and W. Weber. Natural convection in vertical Bridgman configuration. *J. Cryst. Growth*, 70:78–93, 1984.
- [127] W. W. Mullins and R. F. Sekerka. Stability of a planar interface during solidification of a dilute binary alloy. *J. Applied Phys.*, 35:444–451, 1964.
- [128] L. C. Musson. *Two-Layer Slot Coating*. PhD thesis, University of Minnesota, 2001.
- [129] R. J. Naumann. An analytical approach to the thermal modeling of Bridgman-type crystal growth II. Two-dimensional analysis. *J. Cryst. Growth*, 58:569–584, 1982.
- [130] R. J. Naumann and S. L. Lehoczky. Effect of variable thermal conductivity on isotherms in Bridgman growth. *J. Cryst. Growth*, 61:707–710, 1983.
- [131] W. Nernst. Theorie der Reaktionsgeschwindigkeit in heterogenen systemen (theory of reaction velocities in heterogeneous systems). *Z. Phys. Chem.*, 47:52–55, 1904.
- [132] G. P. Nikishkov, H. Kanda, and A. Makinouchi. Tuning the Schur complement computations for finite element partitions. *Adv. Eng. Software*, 31:913–920, 2000.
- [133] H. Ouyang and W. Shyy. Multi-zone simulation of the Bridgman growth process of β -NiAl crystal. *Int. J. Heat Mass Transfer*, 39(10):2039–2051, 1996.
- [134] H. Ouyang and W. Shyy. Numerical simulation of CdTe vertical Bridgman growth. *J. Cryst. Growth*, 173:352–366, 1997.
- [135] A. Pandey. *Global modeling of bulk crystal growth of cadmium zinc telluride in industrial VB/VGF systems*. PhD thesis, University of Minnesota, 2004.
- [136] A. Pandey, A. Yeckel, M. Reed, C. Szeles, M. Hainke, G. Müller, and J. J. Derby. Analysis of the growth of cadmium zinc telluride in an electrodynamic gradient freeze furnace via a self-consistent, multi-scale numerical model. *J. Cryst. Growth*, 276:133–147, 2005.

- [137] S. V. Patankar. A numerical method for conduction in composite materials, flow in irregular geometries and conjugate heat transfer. *Proc. Sixth Int. Heat Transfer Conf.*, 3:297–302, 1978.
- [138] P. Penning. Generation of imperfections in germanium crystals by thermal strain. *Philips' Research Reports*, 13:79, 1958.
- [139] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: energy transfer analysis and three-dimensional applications. *Comput. Methods Appl. Mech. Eng.*, 190:3147–3170, 2001.
- [140] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part I: model problem, theory, and two-dimensional application. *Comput. Methods Appl. Mech. Eng.*, 124:79–112, 1995.
- [141] J. L. Plaza, F. Barvinschi, and E. Diéguez. Modelling of the transport process in Bridgman grown gallium antimonide. *Cryst. Res. Technol.*, 36(7):641–650, 2001.
- [142] E. Raiskin and J. F. Butler. CdTe low level gamma detectors based on a new crystal growth method. *IEEE Trans. Nucl. Sci.*, 35(1), 1987.
- [143] L. Rayleigh. On convective currents in a horizontal layer of fluid when the higher temperature is on the under side. *Philos. Mag.*, 32:475–480, 1916.
- [144] M. D. Reed, C. Szeles, and S. E. Cameron. Computational modeling of heat transport in a multi-zone high-pressure vertical electro-dynamic gradient CdZnTe furnace. *J. Cryst. Growth*, 289:494–501, 2006.
- [145] F. Rosenberger and G. Müller. Interfacial transport in crystal growth, a parametric comparison of convective effects. *J. Cryst. Growth*, 65:91–104, 1983.
- [146] P. Rudolph. Melt growth of II-VI compound single crystals. In M. Isshiki, editor, *Recent Development of Bulk Crystal Growth*. Research Signpost, Trivandrum, India, 1998.
- [147] P. Rudolph and M. Mühlberg. Basic problems of vertical Bridgman growth of CdTe. *Mater. Sci. Eng. B*, 16:8–16, 1993.
- [148] T. Rule. *Experimental Validation of (Cd,Zn)Te Crystal Growth Model*. PhD thesis, Washington State University, 2002.

- [149] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [150] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Pub. Co., Boston, 1996.
- [151] Y. Saad and M. H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
- [152] Y. Saad and M. Sosonkina. Distributed Schur complement techniques for general sparse linear systems. *SIAM J. Sci. Computing*, 21:1337–1356, 1999.
- [153] P. A. Sackinger, P. R. Schunk, and R. R. Rao. A Newton-Raphson pseudo-solid domain mapping technique for free and moving boundary problems: A finite element implementation. *J. Comput. Phys.*, 125:83–103, 1996.
- [154] A. G. Salinger, Q. Xiao, Y. Zhou, and J. J. Derby. Massively parallel finite element computations of three-dimensional, time-dependent, incompressible flows in materials processing systems. *Comp. Methods Appl. Mech. Eng.*, 119:139–156, 1994.
- [155] E. Scheil. Bemerkungen zur schichtkristallbildung. *Z. Metall.*, 34:70–72, 1942.
- [156] K. Schwenkenbacher and P. Rudolph. Investigation of convection in the solution zone at the growth of CdTe by THM. *Cryst. Res. Tech.*, 20:1609–1613, 1985.
- [157] R. G. Seidensticker, W. R. Rosch, R. Mazelsky, R. H. Hopkins, N. B. Singh, S. R. Coriell, W. M. B. Duval, and C. Batur. Active control of interface shape during the crystal growth of lead bromide. *J. Cryst. Growth*, 198/199:988–994, 1999.
- [158] S. Sen, W. H. Konkel, S. J. Tighe, L. G. Bland, S. R. Sharma, and R. E. Taylor. Crystal growth of large-area single-crystal CdTe and CdZnTe by the computer-controlled vertical modified-Bridgman process. *J. Cryst. Growth*, 86:111–117, 1990.
- [159] S. Sen and W. R. Wilcox. Influence of crucible on interface shape, position and sensitivity in the vertical Bridgman-Stockbarger technique. *J. Cryst. Growth*, 28:26–40, 1975.
- [160] L. Shcherbak, O. Kopach, Y. Plevachuk, V. Sklyarchuk, C. Dong, and P. Siffert. The viscosity of liquid cadmium telluride. *J. Cryst. Growth*, 212:385–390, 2000.
- [161] W. Shyy. Multi-scale computation of heat transfer with moving boundaries. *Int. J. Heat Fluid Flow*, 23:278–287, 2002.

- [162] W. Shyy and J. Burke. Study of iterative characteristics of convective diffusive and conjugate heat transfer problems. *Numer. Heat Transfer, Part B*, 26:21–37, 1994.
- [163] D. Silvester, H. Elman, D. Kay, and A. Warren. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comput. Appl. Math.*, 128:261–279, 2001.
- [164] V. G. Smith, W. A. Tiller, and J. W. Rutter. A mathematical analysis of solute redistribution during solidification. *Canadian J. Phys.*, 33:723–745, 1955.
- [165] P. Sonda, A. Yeckel, P. Daoutidis, and J. J. Derby. Development of model-based control for bridgman crystal growth. *J. Cryst. Growth*, 266:182–189, 2004.
- [166] P. Sonda, A. Yeckel, P. Daoutidis, and J. J. Derby. Improved radial segregation via the destabilizing vertical Bridgman configuration. *J. Cryst. Growth*, 260:263–276, 2004.
- [167] P. Sonda, A. Yeckel, J. J. Derby, and P. Daoutidis. The feedback control of the vertical bridgman crystal growth process by crucible rotation: Two case studies. *Comput. Chem. Eng.*, 29:882–896, 2005.
- [168] A. Srinivasan, C. Batur, B. N. Rosenthal, and W. M. Duval. Solid-liquid interface shape control during crystal growth. In *Proceedings of the American Control Conference*, volume 2, pages 1270–1274, 1995.
- [169] M. Storti, L. Dalcin, R. Paz, A. Yommi, V. Sonzogni, and N. Nigro. A preconditioner for the Schur complement matrix. *Adv. Eng. Software*, 31:754–562, 2006.
- [170] C. Szeles, S. Cameron, S. Soldner, J.-O. Ndap, and M. Reed. Development of the high-pressure electro-dynamic gradient crystal-growth technology for semi-insulating CdZnTe growth for radiation detector applications. *J. Electron. Mater.*, 33:742–752, 2004.
- [171] G. K. Teal, M. Sparks, and E. Buehler. Growth of germanium single crystals containing $p - n$ junctions. *Phys. Rev.*, 81:637, 1951.
- [172] R. N. Thomas, H. M. Hobgood, G. W. Eldridge, D. L. Barrett, and T. T. Braggins. Growth and characterization of large diameter undoped semi-insulating GaAs for direct ion implanted FET technology. *Solid-State Elect.*, 24:387–399, 1981.

- [173] W. A. Tiller. Principles of solidification. In J. J. Gilman, editor, *The Art and Science of Growing Crystals*, page 276. John Wiley & Sons, Inc., 1963.
- [174] W. A. Tiller. *The Science of Crystallization: Microscopic Interfacial Phenomena*. Cambridge University Press, New York, 1991.
- [175] W. A. Tiller, K. A. Jackson, J. W. Rutter, and B. Chalmers. The redistribution of solute atoms during the solidification of metals. *Acta Metallurgica*, 1:428, 1953.
- [176] C. M. Tomás and V. Muñoz. CdTe crystal growth process by the Bridgman method: numerical simulation. *J. Cryst. Growth*, 222:435–451, 2001.
- [177] Y. S. Touloukian and C. Y. Ho. *Thermophysical Properties of Matter: Master Index to Materials and Properties*, volume 14. IFI/Plenium, New York, 1979.
- [178] J. P. Tower, S. P. Tobin, M. Kestigian, P. W. Norton, A. B. Bollong, H. G. Schaake, and C. K. Ard. CdZnTe substrate impurities and their effects on liquid phase epitaxy HgCdTe. *J. Electron. Mater.*, 24(5):497–504, 1995.
- [179] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, 1999.
- [180] N. K. Udayashankar, K. G. Naik, and H. L. Bhat. The influence of temperature gradient and lowering speed on the melt-solid interface shape of $\text{Ga}_x\text{In}_{1-x}\text{Sb}$ alloy crystals grown by vertical Bridgman technique. *J. Cryst. Growth*, 203:333–339, 1999.
- [181] J. Virbulis, T. Wetzel, A. Muiznieks, B. Hanna, E. Dornberger, E. Tomzig, A. Mühlbauer, and W. V. Ammon. Numerical investigation of silicon melt flow in large diameter CZ-crystal growth under the influence of steady and dynamic magnetic fields. *J. Cryst. Growth*, 230:92–99, 2001.
- [182] D. Vizman, I. Nicoara, and G. Müller. Effects of temperature asymmetry and tilting in the vertical Bridgman growth of semi-transparent crystals. *J. Cryst. Growth*, 212:334–339, 2000.
- [183] C. A. Wang, A. F. Witt, and J. R. Carruthers. Analysis of crystal growth characteristics in a conventional vertical Bridgman configuration. *J. Cryst. Growth*, 66:299–308, 1984.

- [184] Z. Q. Wang and D. Stroud. A simple scaling formula and Monte Carlo study of liquid semiconductor surfaces: application to $\text{Cd}_x\text{Te}_{1-x}$. *Scripta Metall. Mater.*, 32:87–92, 1995.
- [185] W. R. Wilcox. Validity of the stagnant film approximation for mass transfer in crystal growth and dissolution. *Mat. Res. Bull.*, 4:265–274, 1969.
- [186] A. Winslow. Adaptive mesh zoning by the equipotential method. Technical Report UCID-190602, Lawrence Livermore National Laboratory, 1981.
- [187] Q. Xiao, S. Kuppurao, A. Yeckel, and J. J. Derby. On the effects of ampoule tilting during vertical Bridgman growth: three-dimensional computations via a massively parallel, finite element method. *J. Cryst. Growth*, 167:292–304, 1996.
- [188] E. V. Yakovlev, V. V. Kalaev, I. Y. Evstratov, C. Frank-Rotsch, M. Neubert, P. Rudolph, and Y. N. Makarov. Global heat and mass transfer in vapor pressure controlled Czochralski growth of GaAs crystals. *J. Cryst. Growth*, 252:26–36, 2003.
- [189] A. Yeckel, G. Compère, A. Pandey, and J. J. Derby. Three dimensional imperfections in a model vertical Bridgman growth system for cadmium zinc telluride. *J. Cryst. Growth*, 263:629–644, 2004.
- [190] A. Yeckel and J. J. Derby. Effect of accelerated crucible rotation on melt composition in high-pressure vertical Bridgman growth of cadmium zinc telluride. *J. Cryst. Growth*, 209:734–750, 2000.
- [191] A. Yeckel and J. J. Derby. Bouyancy and rotation in small-scale vertical Bridgman growth of cadmium zinc telluride using accelerated crucible rotation. *J. Cryst. Growth*, 233:599–608, 2001.
- [192] A. Yeckel and J. J. Derby. Feasibility study of cadmium zinc telluride growth using submerged heater in a vertical Bridgman system. *J. Electron. Mater.*, 33:488–497, 2004.
- [193] A. Yeckel and J. J. Derby. Computer modelling of bulk crystal growth. In P. Capper, editor, *Bulk Crystal Growth of Electronic, Optical, and Optoelectronic Materials*, pages 73–119. John Wiley & Sons, West Sussex, UK, 2005.

-
- [194] A. Yeckel, F. P. Doty, and J. J. Derby. Effect of steady crucible rotation on segregation in high-pressure vertical Bridgman growth of cadmium zinc telluride. *J. Cryst. Growth*, 203:87–102, 1999.
- [195] A. Yeckel and R. T. Goodwin. *Cats2D (Crystallization and Transport Simulator), User Manual*, 2003. <http://www.msi.umn.edu/~yeckel/cats2d.html>.
- [196] A. Yeckel, A. Pandey, and J. Derby. Fixed-point convergence of modular, steady-state heat transfer models coupling multiple scales and phenomena for melt-crystal growth. *Int. J. Numer. Methods Eng.*, 67:1768–1789, 2006.
- [197] K. Zanio. *Cadmium telluride*, volume 13 of *Semiconductors and Semimetals*. Academic Press, New York, 1978.
- [198] H. Zhou and J. J. Derby. Parallel implementation of finite element method with MPI: application to three-dimensional free surface stokes flow. In *Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, June 26-29 2000.
- [199] H. Zhou and J. J. Derby. An assessment of a parallel, finite element method for three-dimensional, moving-boundary flows driven by capillarity for simulation of viscous sintering. *Int. J. Numer. Methods Fluids*, 36:841–865, 2001.
- [200] O. C. Zienkiewicz, D. K. Paul, and A. H. C. Chan. Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems. *Int. J. Numer. Methods Eng.*, 26:1039–1055, 1988.