

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN
DÉPARTEMENT DE MÉCANIQUE



Numerical Simulation of Czochralski Bulk Crystal Growth Process: Investigation of Transport Effects in Melt and Gas Phases

Liang Wu

*Thèse présentée en vue de l'obtention du grade de
Docteur en Sciences Appliquées*

Advisor :

Prof. F. Dupret (MEMA, UCL)

President of the Jury :

Prof. J.-C. Samin (PRM, UCL)

Members of the Jury :

Prof. E. Dick (University of Gent)

Dr. Y. Ryckmans (Laborelec)

Dr. V. Regnier (FEMAGSoft)

Dr. N. Van den Bogaert (FEMAGSoft)

Prof. G. Winckelmans (TERM, UCL)

Prof. V. Legat (MEMA, UCL)

August 2008

Numerical Simulation of Czochralski Bulk Crystal Growth Process: Investigation of Transport Effects in Melt and Gas Phases

Liang Wu

*Thèse présentée en vue de l'obtention du grade de
Docteur en Sciences Appliquées*

Abstract

The current requirement of wafers of larger and larger size and extremely high quality (with respect to impurity, defect and shape control, etc.) for the semiconductor industry, together with the requirement of cost-effective wafers with long carrier lifetimes for the solar market, have made the design of the crystal growth furnace and the whole growth system a very challenging task. Global numerical simulation of bulk crystal growth becomes an indispensable and powerful tool to design, predict and optimize the crystal growth process.

Based on more than 20 years of R&D experience on crystal growth modeling at the CESAME research center of UCL, the main objective of this thesis aims at developing a new generation of software products, in order to obtain a fully automatic simulator predicting the entire Czochralski process while handling correctly the switches between the different growth stages (including poly-crystal melting, seeding, conical growth, shouldering, body growth, tail-end stage and after growth cooling).

To achieve this goal, new efficient, robust and high-quality automatic unstructured mesh generation algorithms with enough flexibility for any complex geometry were implemented, including a 1D mesh generator by global grade-adaptive method, a 2D initial triangulation algorithm by improved sweep line technique and an automatic 2D shape-quality unstructured mesh generator by modified incremental Delaunay refinement technique.

The convection problem in Czochralski crystal growth represents one of the biggest challenges of crystal growth modeling. In this thesis, a Finite Element Navier-Stokes solver based on unstructured meshes was firstly developed and validated by solving the well-known 2D lid-driven cavity flow problem at different Reynolds numbers. Then, enhanced turbulence models based on the classical

mixing-length or $k-l$ model improved by an appropriate treatment of the mixing-length boundary layers, together with a generic transformation method to avoid negative k when solving the turbulent kinetic energy equation by the Newton-Raphson iterative method were introduced and implemented.

In practice, gas convection exerts an important effect on the global heat transfer and oxygen concentration both in the silicon melt and the silicon crystal in modern Czochralski crystal growth processes. Therefore, laminar and turbulent mathematical models governing the gas convection, thermal distribution and oxygen concentration were developed, and Finite Element numerical methods to solve these governing equations on unstructured meshes were investigated. In order to analyze the gas effects on the silicon melt flow, oxygen distribution and global heater transfer, appropriate numerical approaches to capture the wall shear stress exerted by the gas flow and experienced by the silicon melt were implemented and analyzed.

Finally, a series of numerical experiments devoted to investigate the industrial Czochralski crystal growth process under various growth conditions are presented based on all the developments implemented. Comparisons of the simulation results with literature and available experimental observations are also presented, and conclusions are drawn based on these simulation results and observations.

Contents

Abstract	iii
Acknowledgements	ix
1 Introduction	1
1.1 Preliminary remarks	1
1.2 Czochralski bulk crystal growth	1
1.3 Computer Modeling of Bulk Crystal Growth	2
1.4 Challenges of modeling and simulation of Bulk Crystal Growth	5
1.5 Thesis structure	7
2 Mesh Generation for Bulk Crystal Growth Processes	11
2.1 Introduction	11
2.2 Mesh generation strategy for crystal growth	12
2.3 Unstructured 2D mesh generation	14
2.3.1 Data structure	16
2.3.2 Initial triangulation	18
2.3.3 Incremental Delaunay refinement algorithm	28
2.4 Grade-adaptive 1D boundary mesh generation	38
2.5 Mesh deformation for quasi-steady and time-dependent simulations	43
2.6 Extend to surface triangulation	45
2.7 Conclusions	48
3 Modeling of Melt Convection	51
3.1 Melt convection in Czochralski growth	51
3.2 Modeling of melt convection in Czochralski silicon growth . . .	54

3.2.1	Governing equations	56
3.2.2	Boundary conditions	58
3.3	Numerical modeling	59
3.3.1	Discretization method and formulation	59
3.3.2	Numerical method	62
3.4	Numerical benchmark testing on 2D cavity problem	63
3.5	From laminar to turbulent flow	68
3.6	Turbulence modeling	71
3.6.1	Reynolds averaged Navier-Stokes equations	71
3.6.2	Closure problem and turbulent-viscosity hypothesis	73
3.6.3	Turbulence models and simulations	75
3.7	The mixing-length turbulence model	78
3.7.1	Mixing-length modeling	80
3.7.2	Enhanced mixing-length model	82
3.8	Turbulence $k - l$ model	86
3.8.1	Modeling of the turbulent kinetic energy equation	86
3.8.2	Numerical method	89
3.9	Generic transformation method to avoid negative k	91
3.9.1	Transformation scheme and discretization	92
3.9.2	Numerical experiments and discussions	96
3.10	Conclusions	97
4	Modeling of Gas Convection and Oxygen Transport	101
4.1	Introduction	101
4.2	Modeling of gas convection	104
4.2.1	Governing equations	104
4.2.2	Boundary Conditions	106
4.2.3	Turbulence models for gas convection	108
4.3	Modeling of oxygen transport	109
4.4	Capture of the wall shear stress using Lagrange Multipliers	111
4.4.1	Discretization and numerical method	113
4.5	Conclusions	120
5	Industrial Applications of Czochralski Crystal Growth Process	123
5.1	Introduction	123
5.2	Global modeling and iterative strategy	124
5.3	Description of the industrial Leybold EKZ 1300 Czochralski silicon puller	126
5.4	Quasi-steady simulations on two different meshes	126

Contents

5.5	Quasi-steady simulation with turbulence mixing-length and $k-l$ models	131
5.6	Influence of the pulling rate on the melt and crystal interface .	135
5.7	Influence of convection on the Czochralski crystal growth process	137
5.7.1	Influence of melt/gas convection on the Czochralski silicon growth process	137
5.7.2	Influence of gas convection on Czochralski silicon growth process under varied crucible rotation and gas flow rate	142
5.8	Conclusions	158
6	Conclusions	161
A	Discretization of generic transformed turbulent kinetic energy equation	165
	Bibliography	171

Acknowledgments

First of all, I would like to take this opportunity to express my deep and sincere gratitude to my advisor, Professor Dupret François, for his careful guidance, enormous encouragement, continuous support and endless patience throughout this work. His scientific endeavor, overly enthusiasm and integral view on research as well as his hard-work, wide knowledge and superb analytical skill have been of great value for me.

I would also like to thank the members of the jury, Professor Erik Dick from University of Gent, Dr. Yves Ryckmans from Laborelec, Professor Grégoire Winckelmans, Professor Vincent Legat and Professor Jean-Claude Samin from Université Catholique de Louvain for their detailed review, insightful comments, helps and suggestions during the thesis preparation.

I would also like to extend my sincerely thanks to Dr. Nathalie, Van den Bogaert, presently the CEO of *FEMAGSoft* S.A. She always was there when I needed helps, advices and discussions, which were not only limited to specific scientific issues, but also my daily life in Louvain-la-Neuve. I'm also indebted to Dr. Vincent Regnier, the manager of R&D team at *FEMAGSoft*, who provided fruitful helps, suggestions and discussions throughout this work, particularly on the theory aspects of turbulence modeling.

I gratefully acknowledge cooperations, discussions and helps from the following colleagues in our research group at CESAME, UCL: Roman Rolinsky, Brieux Delsaute, Olivier Magotte, Fabrice Loix, Nicolas Van Geothem, François Bioul. More specifically, I would like to thank Roman and Oliver, for their numerous and insightful discussions when I was implementing the unstructured mesh generation algorithm and codes for the “geometrical project”, and Brieux who provided a lot of helps and discussions on Finite Element Method and fluid mechanics, Fabrice, the “sportsman” at CESAME, who organized a lots of sport activities in our spare time.

Also thanks to Michèle Sergant, our secretary at MEMA, who provide enor-

mous logistics helps during my PhD research period. Mr. Arnaud de Potter from *FEMAGSoft*, where numerical simulation work in Chapter 5 were performed, provided me helps in various way.

In addition, thanks to Professor Martin Held, the head of the Computational Geometry and Computer Graphics Lab at the Universität Salzburg, Austria, who provided me precious random polygon data set for Poly2Tri's robustness and efficiency testing.

And last but not least, I owe my loving thanks to my wife Ruijun Liu, my sons Bohan and Boxuan, who have accompanied me at Louvain-la-Neuve. My special gratitude is due to my parents, brother and my sisters for their loving, long-time encouragement and support. Without their encouragement and understanding it would have been impossible for me to finish this work.

Liang Wu
Louvain-la-Neuve
August, 2008

1.1 Preliminary remarks

The whole industrial world currently relies critically on bulk-grown crystals of a variety of materials. These industries range from information technology (IT) based on silicon, through radiofrequency applications using gallium arsenide, etc., to telecommunications and lighting based on III-V compounds, to infrared imaging based on cadmium mercury telluride, and to high-energy physics and medical imaging using scintillator materials. Of which, silicon is the most important semiconductor material. Silicon based systems on “chips” form the basis of the huge information technology industry [(Capper 2005)]. Nowadays, almost all large-scale integration (LSI) chips are fabricated on silicon wafers except the high-speed and optical devices fabricated on III-V compound semiconductor crystals [(Hibiya and Hoshikawa 2005)].

The Silicon crystal growth technology has rapidly advanced during the past decades. One of its striking features has been the diameter race which was driven by the demand for cost reduction in the device industry [(von Ammon 2004)]. Currently silicon wafer size has been converted from 200mm to 300mm in diameter. However, the current transition from 200mm to 300mm wafers has been slower and more painful than expected and there are strong indications that this almost periodic increase of crystal diameter will considerably slow down in the future because of the dramatically rising costs of crystal growth processes and device manufacturing equipments. The 300mm wafer is expected to stay for at least one decade and the transition from 300mm to 450mm wafer probably will take place around 2012~2014 [(Hibiya and Hoshikawa 2005)] or even later.

1.2 Czochralski bulk crystal growth

The development of the silicon crystal growth technology has been driven by two major challenges, i.e. the demand for larger wafer diameters, as previously

mentioned, and the need to improve the bulk quality of the crystals. The Czochralski method has proven to be the most suitable to follow this rapid development. It is estimated that more than 20,000 tons of bulk crystals are produced per year, while 50% of them are semiconductor silicon crystals, of which 98% are produced by the Czochralski method, and the remaining silicon crystals are manufactured by Floating Zone method, which is mainly used to grow high purity silicon for power and high-frequency electronic devices.

The Czochralski single crystal growth method was originally developed by Jan Czochralski [(Czochralski 1918)], while modern Czochralski crystal growing systems are more or less still the same as what was developed for industrial applications by Teal and Little in the 1950's [(Hurle 1993)]. Figure (1.1) shows a large Czochralski crystal growth apparatus and a diameter of 400mm single silicon crystal ingot with weight 438kg and body length 1,100mm produced by [(Shiraishi et al. 2001) in 2001]. The Czochralski silicon crystal growth process consists of several basic steps as follows. First, a precise amount of polycrystalline silicon is charged in a silica crucible, and the growth chamber is then back-filled with low pressure, high purity Argon or Nitrogen gas (in order to protect the growth system and the silicon at high temperature). After silicon is molten by a graphite heater, the single crystal is grown gradually with a controlled shape from the silicon melt contained in the quartz crucible. Generally, a sequence of crystal-growth process steps, like seeding, necking, shouldering, body growing, etc., are performed to grow anticipated size, high quality crystals (dislocation-free with low concentration of point defects and appropriate level of oxygen concentration, etc.). Moreover, different kinds of magnetic fields are widely employed in nowadays crystal growth systems to improve the quality and oxygen distribution in the crystal.

1.3 Computer Modeling of Bulk Crystal Growth

The extremely high quality requirements of silicon wafers as well as the crystal purity, defect and shape control requests, and the demand for large-size wafers, have made the design of the furnace and the whole crystal growth system a very challenging task, and full collaborations between chemical and process engineers, thermodynamicists, hydrodynamists, electrical and mechanical/machine engineers, material scientists, numerical specialists, physicists and crystallographers are absolutely necessary because of the multidisciplinary nature of the crystal growth technology [(Scheel 1998)]. The complex chemical and physical theories involved in crystal growth render the understanding of the



(a) A Czoehralski crystal puller with magnet



(b) A 400mm silicon crystal ingot

Figure 1.1: A Czoehralski puller and silicon ingot [(Shiraishi et al. 2001)]

growth phenomena and of the interaction between the many factors influencing the final product quality, a very complicated task. However, to perform any extensive experiments in the hostile crystal growth environment with high temperature, opaque materials and high purity requirements, etc., is proved to be time consuming and extremely expensive. Therefore, numerical modeling and simulation has become an essential and indispensable tool for understanding, developing and optimizing the crystal growth processes and equipments.

The primary objectives of numerical simulation of bulk crystal growth are first to estimate the various processing conditions, such as heater power, pull rate, crystal and crucible rotation rates and magnetic field intensity, and then to predict the evolution of the temperature field taking all furnace constituents into account, to calculate the evolution of the melt/gas flows, and to determine the shapes of meniscus and crystal/melt interfaces, etc.. All such kind of processing parameters, as well as the evolutions of the temperature field, melt/gas convection and growth interfaces will provide a better understanding of the various factors which may affect the quality of the crystal during the growth process. The ultimate goal of numerical modeling is to control and optimize the growth processes.

At the present time, crystal growth numerical modeling is based on introducing some basic hypotheses to approximate and simplify the growth processes. For example, the temporal behavior of the crystal growth system is approximated by the quasi-steady assumption. Another well-known and widely used assumption is that the crystal growth system is perfectly axisymmetrical. These two important assumptions have proved to be highly successful in the past decades both in academic researches and in industrial applications, although experiments show that the temperature field, as well as the melt convection are far from axisymmetrical and that there exists complex interactions between the melt driving forces. Hence the real melt flow has proved to be three-dimensional, oscillatory and turbulent due to the size of modern crystal growth systems, and to the temporal behavior of these crystal growth systems.

The numerical modeling approach at CESAME was initiated in the early eighties by Crochet and co-workers using the finite element technique [(Crochet et al. 1983a)][(Crochet et al. 1989)], while Dupret and co-workers have been extending and developing these methods for decades to simulate the melt convection and global heat transfer in Czochralski, Bridgman, Floating zone and VGF crystal growth processes [(Dupret et al. 1990)][(Dupret and Van den Bogaert 1994)]. The bases of the global quasi-steady model, which includes axisymmetric, diffuse radiative heat transfer [(Dupret et al. 1990)] between the furnace gray surfaces, were introduced by Wouters in the mid-

eighties [(Wouters 1985)]. The development of the global model was pursued by Ryckmans who implemented laminar melt convection in the global heat transfer and applied it to the Czochralski growth of germanium and the Vertical Bridgman Growth of indium phosphide [(Ryckmans 1989)], and by Nicodeme, who extended the model to take into account the particularities of Liquid Encapsulated Czochralski furnaces, and developed thermo-elastic and visco-plastic models to compute the thermally induced stress field responsible of dislocation generation inside the growing crystal [(Nicodème 1990)]. Furthermore, laminar flow computations and the development of a simplified approach to take into account laminar melt flow effects during dynamic global simulations were performed by Lecomte [(Lecomte et al. 1992)]. Moreover a global dynamic Czochralski heat transfer model taking into account the heat capacity and the geometrical evolution of the puller constituents was developed by Van den Bogaert [(Van den Bogaert 1993)][(Van den Bogaert and Dupret 1997a)][(Van den Bogaert and Dupret 1997b)]. Finally Assaker [(Assaker et al. 1997)][(Assaker 1998)] extended the global quasi-steady and dynamical heat transfer algorithm by introducing two axisymmetric eddy-viscosity turbulence models for melt convection possibly taking the effect of axial or cusp magnetic fields into account. The related commercialized software products, such as FEMAG-CZ/FEMAG-FZ/FEMAG-VGF, are currently used by major crystal growth companies around the world.

1.4 Challenges of modeling and simulation of Bulk Crystal Growth

Most crystal growth processes involve a large number of coupled physical phenomena and require to master the related theories, such as, heat transfer by conduction, convection and radiation; mass transfer by convection and diffusion; fluid dynamics with combined effect of natural and forced convection; multi-phase surface dynamics; chemical reactions in the solid and molten silicon; point defects dynamics and thermomechanics, etc.. For all such kinds of physical behaviors, coupling with the continually deforming melt/crystal geometry, leads to a set of highly nonlinear, time-dependent partial differential equations governing the growth process to be solved. Considering the fact that these physical phenomena occur over a vast range of time scales and length scales, a numerical model that includes all these physical phenomena is far beyond the capabilities of today's computers. Even if the computing capacity were available, the existing state of theory remains inadequate to build a

comprehensive, realistic and usable model to predict the whole crystal growth process [(Yeckel and Derby 2005)].

One essential challenge comes from the overall furnace deforming geometry and the associated high-quality automatic mesh generation issues both for quasi-steady and time-dependent simulations. These issues play an essential role in numerical simulation of bulk crystal growth. Indeed, not only do several constituents (crucible, pedestal, crystal, and pulling rod) move, but also do crystal and melt continually change shape. In particular, the solid is very small during seeding and subsequently becomes larger and larger, while the melt volume decreases and often takes a particular shape during tail-end stage. Various moving and deforming interfaces delimiting the melt and the crystal must hence be computed for the process dynamic simulation, and very complex problems of numerical geometry are thereby posed. Therefore efficient, automatic mesh generation algorithms with enough flexibility to control and optimize the mesh shape, size and density for complex deforming geometries have to be developed. Also if the geometrical changes become too important when switching to a subsequent growth stage, an automatic remeshing procedure has to be performed in order to provide optimal mesh shapes.

However, the biggest challenge for the numerical simulation of Czochralski single crystal growth comes from melt convection modeling. Indeed melt convection strongly affects the various mixing mechanisms in many growth systems, thereby affecting the growth rate, melt/crystal interface shape, chemical composition of the crystal and subsequent defect formation. Therefore the accurate simulation of heat transfer, oxygen segregation and the formation of crystal defects in large-size Czochralski growth systems highly relies on the accuracy of turbulent convection modeling in the silicon melt. However, with nearly two decades of development, melt convection simulation remains one of the most difficult challenges to overcome in the crystal growth modeling research community, and probably will continue to be one of the most active research topics in the next decades. First of all, from the aspect of physics, the flow of the molten semiconductor in a Czochralski crucible is dauntingly complicated due to a complex combination of natural and forced convection. Natural convection is attributed to the buoyancy and capillary forces acting in the melt and on its interface. The buoyancy and capillary forces are due to the melt density and surface tension dependence upon temperature and solute concentration. On the other hand, the forced convection due to crystal and/or crucible rotation rates (in order to grow axisymmetric cylindrical crystals) strongly complicates the melt flow pattern. Finally the shear stress exerted on the melt/gas (meniscus) interface due to gas convection also plays an important role on the melt flow

pattern. The combination of all the above driving forces leads to a complex three-dimensional and time dependent flow structure. Ample evidence shows that the melt flow driven by the above mentioned forces is at least chaotic and mostly probably turbulent [(Lipchin and Brown 1999)], even for those growth systems with relatively small-scale crucibles [(Kim and Langlois 1991)]. From the viewpoint of modeling and numerical algorithms, turbulence proves to be the most difficult problem of fluid mechanics and it is notoriously difficult to develop an accurate tractable turbulent mathematical model because of its random, multi-dimensional and time-dependent nature. There are no prospects of a simple analytic theory [(Lipchin and Brown 1999)]. Although the basic equations that describe turbulence are well-known and simple, however the solutions are incredibly complex and the computer power needed to find these solutions would transcend an imaginable computer if direct simulations were to be performed.

Let us mention that there are other challenges today which still have not yet been well solved, such as the fully time-dependent 3D global modeling with continuously deforming geometries, the theory and modeling of defect generation, the radiation modeling of participating media, etc..

1.5 Thesis structure

The main objective of this thesis aims at developing the next generation of the FEMAG-CZ software product, a fully automatic simulator predicting the entire Czochralski process while handling correctly the switches between the different growth stages, including poly-crystal melting, seeding, conical growth, shouldering, body growth, tail-end stage and after growth cooling, although most of the developments in this thesis are directly or with minor adaptations applicable to other crystal growth processes, such as Vertical Bridgman (VB) and Floating Zone (FZ) process.

However, in the previous FEMAG-1 software generation, structured quadrilateral meshes were used, and the resulting requirement of intensive interactions from the users made it hard for fully automatic time-dependent simulations. Other big disadvantages of structured quadrilateral meshes are found in the difficulty of handling complex geometries with holes, and in the lack of flexibility to control the mesh density (boundary or internal node spacing). Therefore, new efficient and high-quality automatic unstructured mesh generation algorithms with enough flexibility for complex deforming geometries had to be implemented firstly, which will be the main topic of our chapter 2. In this chapter, we will firstly introduce a general strategy of unstructured mesh generation

for quasi-steady and dynamic simulations of bulk crystal growth processes, and then we will focus on 2D shape-quality unstructured mesh generation by a modified incremental Delaunay refinement algorithm and a grade-adaptive 1D mesh generation on fixed geometries. Finally we will close this chapter by presenting the associated 1D and 2D mesh deformation techniques for quasi-steady and dynamic simulations.

In chapter 3, we will focus on the convection problem in Czochralski crystal growth processes, one of the biggest challenges in crystal growth modeling. In this chapter, we will first describe the origin of the melt flow features in Czochralski crystal growth systems, then detail the governing equations, boundary conditions and numerical methods both for the laminar, turbulent mixing-length, and turbulent $k - l$ models. In order to validate our FEM based, unstructured Navier-Stokes solver, benchmark testings on the 2D lid-driven cavity problem will be performed at different Reynolds numbers, and comparisons of our solutions with other highly accurate results obtained by the FEM or other numerical methods will also be given. Moreover, an enhanced turbulent mixing-length model based on the existing mixing-length model improved by an appropriate treatment of the mixing-length boundary layers will be introduced. Finally, a generic transformation method to avoid negative k when solving the turbulent kinetic energy equation by the Newton-Raphson iterative method will be introduced, and numerical experiments and comparisons of different transformation schemes, such as the \sqrt{k} scheme and $\log k$ scheme, will be presented and discussed.

In chapter 4, we will firstly address the modeling of gas convection and oxygen transport issues in Czochralski silicon single crystal growth processes. Two ways to capture the shear stress experienced by the melt flow due to gas convection, by the direct method and the Lagrange Multipliers method will also be presented, and the results obtained from these two methods will also be discussed.

The main purpose of chapter 5 will be to investigate the numerical applications of all the developments presented in the previous chapters to industrial Czochralski crystal growth using the FEMAG quasi-steady global simulation technique. Therefore, the FEMAG global modeling concepts and techniques will be summarized firstly in chapter 5, and then we will focus on industrial applications to Czochralski silicon growth processes by means of turbulent mixing-length, enhanced mixing-length and $k - l$ models without taking gas convection into account. Finally, in order to fully analyze the effect of inert argon gas convection on the Czochralski growth process, a series of numerical experiments taking gas convection into account are performed with varied

growth conditions and operating parameters, and comparison of the simulation results (such as the heater power, temperature difference in the silicon melt etc.) with literature and available experimental observations are also presented, while the influences of gas convection on the global heat transfer, melt convection, interface position and oxygen concentration in the silicon melt at different operating conditions are detailed and the appropriate conclusions are drawn.

Chapter 2

Mesh Generation for Bulk Crystal Growth Processes

2.1 Introduction

Meshing can be defined as the process of breaking up a complex physical domain into smaller and simpler pieces (elements) in order to facilitate further numerical processing[1]. This technique can be used for a wide range of applications, such as computational geometry, computer graphics, solid modeling, geographic information systems (GIS), robotics, and others. The principal application of interest is the Finite Element Method (FEM). For example, in the FEM world, a physical domain is decomposed into elements, typically triangles in 2D applications or tetrahedra in 3D applications. Partial differential equations (PDEs) representing some physical phenomena, such as heat transfer or fluid convection, can then be approximated by using functions that are piecewise polynomial within each element.

According to the type of mesh generated, there are two major kinds of meshes: structured and unstructured. Structured meshes, which are commonly called “grids”, offer certain advantages over unstructured meshes. Firstly, they are simpler and also more convenient for use with the simpler finite difference method (FDM) or finite volume method (FVM). Also, because of the ease of determining each node neighbors, even the FEM method on structured meshes is often simpler and faster than on unstructured meshes. Furthermore, the regular topology of structured meshes makes their use straightforward for parallel computation [(Schewchuk 1997)]. However, the big disadvantage of structured meshes is their lack of flexibility when fitting complex and irregular domains, whereas unstructured meshes can be flexibly tailored to any domain shape. And there are cases in which using unstructured meshes is preferable or even indispensable. For example, in Science and Engineering area, several physical phenomena with sharp feature changes are modeled by PDEs defined on irregularly shaped domains and then need dynamic adaptive, anisotropic unstructured meshes to capture these sharp changes with acceptable accuracy,

time and memory requirements. Although unstructured mesh generation is a relatively new field, it is gaining increasing popularity in diverse fields and tremendous progress has been made in the last decade.

In this chapter, we will firstly introduce the general strategy we have selected for unstructured mesh generation in order to perform quasi-steady and dynamic simulations of bulk crystal growth processes. We will then focus on 2D shape-quality unstructured mesh generation as based on a modified incremental refinement algorithm and a grade-adaptive 1D boundary mesh generation. Furthermore, 1D and 2D mesh deformation techniques for quasi-steady and dynamic simulations will also be addressed, and application examples of our method to bulk single crystal growth processes will be presented as well. Finally, the possibility to extend our algorithm to surface mesh generation will be shortly discussed.

2.2 Mesh generation strategy for crystal growth

Numerical modeling of Czochralski silicon growth requires to develop suitable dynamic geometrical algorithms in order to accurately represent the various – moving or not – constituents of the furnace, together with computing the evolution of the different deforming system interfaces (solidification front, melt/gas interface with crystal/melt and crucible/melt menisci, and crystal wall).

The objective of launching the FEMAG-2 software generation was to provide a fully automatic simulator predicting the entire Czochralski process while handling correctly the switches between the different growth stages, including poly-crystal melting, seeding, conical growth, shouldering, body growth, tail-end stage and after growth cooling. The difficulty to overcome resulted from the important evolution of the system geometry during growth. Indeed, not only do several constituents (crucible, pedestal, crystal, and pulling rod) move, but also do crystal and melt continually change shape. In particular, the solid is very small during seeding and subsequently becomes larger and larger, while the melt volume decreases and often takes a particular shape during tail-end stage. Various moving and deforming interfaces delimiting the melt and the crystal must hence be computed for the process dynamic simulation, and very complex problems of numerical geometry are thereby posed. Therefore, new powerful geometrical methods were developed to allow easy calculation of the system free surfaces, through a single mathematical formulation, valid for all possible configurations, and new efficient and automatic unstructured mesh generation algorithms with high flexibility were implemented in order to perform easy time-dependent simulations even for stages of the process where important ge-

ometrical changes occur (note that structured quadrilateral meshes were used in the previous FEMAG-1 software generation).

FEMAG-2 simulations can be quasi-steady or time-dependent. For each growth stage the so-called quasi-steady problem can be solved on a fixed geometry (except the shape of melt/crystal interface), which means that geometrical calculations are completely separated from field calculations. The decoupled iterative strategy adopted to perform FEMAG-2 quasi-steady simulations can be summarized as follows :

- ❶ The global heat transfer is solved in the overall furnace by assuming given melt flow and geometry – the latter being defined by crystal and crucible vertical positions, and by the shapes of crystal/melt, crystal/gas, and melt/gas interfaces;
- ❷ The non-isothermal melt flow is solved assuming given geometry and thermal conditions along the melt boundary;
- ❸ The geometry is updated using the results obtained in steps ❶ and ❷.

This sequence of operations is iteratively repeated until convergence is achieved, and hence a coupled solution is provided by this decoupled scheme. In case of dynamic simulations, the procedure is subdivided in two principal steps. First, the melt flow is calculated at several stages of the growth by means of quasi-steady simulations. Secondly, interpolation between these quasi-steady results provides the melt flow at every time step of the dynamic simulation. At each time step, calculations are iteratively performed in a similar way as in the quasi-steady case (without however melt flow calculations since flow interpolation is applied). Recently, however, a new numerical technique was implemented to calculate the melt flow at each time step.

Unstructured Finite Element meshes are initially generated to cover each constituent, or “macro-element”, of the global furnace (viz. the crystal, the melt, the crucible, the heater, the insulator, the pedestal..., all the 1D radiative enclosures, and the 1D external steel shell). This initial mesh generation procedure is illustrated in Figure (2.1). Subsequently, the global mesh is automatically deformed for each iteration of the quasi-steady calculations and at each time step of the dynamic simulations. This operation is based on using a 1D mesh adaptor and a 2D mesh conditioner to optimize the deformed mesh while keeping mesh topology consistency. In addition, re-meshing is performed only if geometrical changes become too important when switching to a subsequent growth stage.

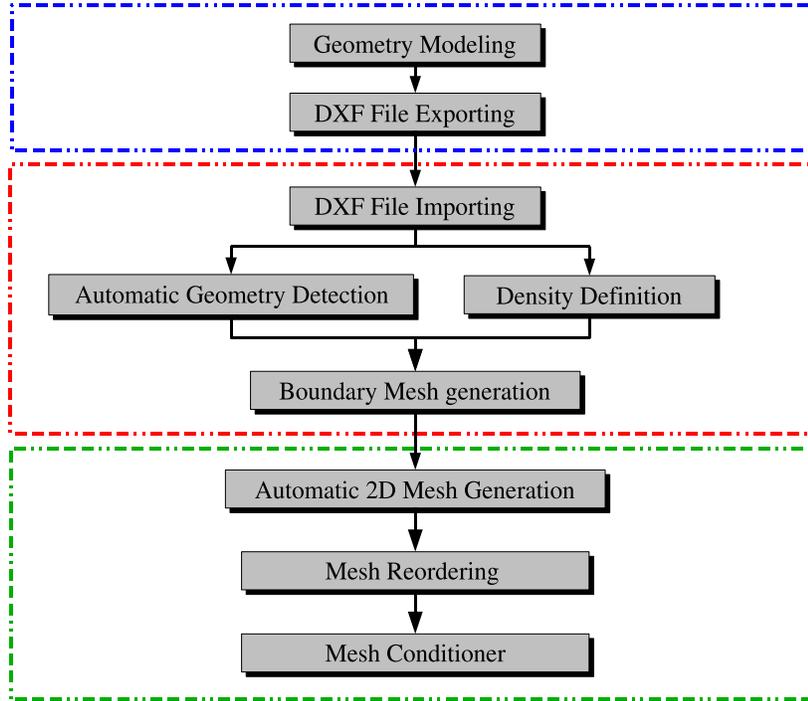


Figure 2.1: Initial mesh generation procedures.

2.3 Unstructured 2D mesh generation

Triangles (in 2D space) and tetrahedra (in 3D space) are by far the most common forms of elements for unstructured mesh generation while most techniques currently in use can fit into one of the following three main categories [(Owen 1998)]: ❶ Octree technique, ❷ Delaunay technique, and ❸ Advancing Front technique, of which, the Delaunay method is the most popular technique employed in the mesh generation community. The unstructured mesh generation process using the Delaunay criterion is also referred to as Delaunay Triangulation. The latter has the property that the circumcircle or circumsphere of every element (triangle or tetrahedron) doesn't contain any node of the triangulation (refer to Figure 2.2) besides the element vertices while it maximizes the minimum angle and minimizes the maximum circumcircle or circumsphere of all the elements. There are many Delaunay triangulation algorithms, for example, the divide-and-conquer algorithm proposed by Lee and

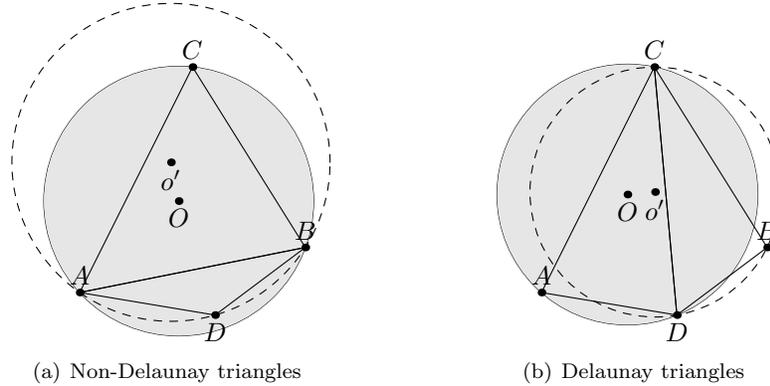


Figure 2.2: *Delaunay criterion*

Schachter [(D.T.Lee and B.J.Schachter 1980)], the plane-sweep algorithm presented by Fortune [(Fortune 1987)] and the incremental insertion algorithm proposed by Lawson [(Lawson 1977)]. Fortune[(Fortune and Wyk 1993)], Su and Drysdale [(Su and Drysdale 1995)] and Shewchuk [(Shewchuk 1996b)] have presented experimental comparisons of a number of these algorithms. Among them, the incremental insertion algorithm performs poorly since most of its time is spent on point location. On the other hand, the incremental insertion algorithm is simple to understand and implement and is more competitive than other algorithms in most practical applications.

In this section, we will present a modified incremental insertion algorithm based on the algorithm proposed by Ruppert [(Ruppert 1995)]. However, important differences between our implementation and the one by Ruppert and others can be highlighted as follows [(Wu et al. 2003)]:

1. Unlike the implementation by Ruppert, splitting the encroached boundary line segments is ignored during the triangulation process in order to keep boundary consistency (as required by FEMAG-2 software constraints).
2. An innovative triangle-based data structure is proposed for the incremental Delaunay refinement process. Based on this elegant data structure, a new, simple but efficient, linear point location algorithm is presented. The main advantage of our point location algorithm is that the associated time complexity is independent of the mesh size. Furthermore, a so called “split and flip” generic algorithm to construct the updated

Delaunay triangulation is proposed and implemented.

3. Similarly, as in the Triangle code developed by Schewchuk, a splay-tree sorting/searching algorithm is used to quickly locate the “bad triangles”, while two sorting keys are employed in our implementation for different refinement stages, as associated to the triangle minimum angle and maximum area constraints.
4. Three methods are used to control the mesh shape quality and density: 1D boundary mesh density, minimum angle and maximum area-scale control. The 1D boundary mesh density plays a prominent role to keep mesh quality, mesh density and boundary consistency in our algorithm. The second criterion relies on the minimum element angle, which is commonly used by unstructured triangulation codes. The third one is the maximum area-scale criterion, which is extremely useful when triangulating flow boundary layers (with expected large variations in triangle size).

Before addressing the implementation detail of our algorithm, let us first summarize the basic steps of the Delaunay triangulation process: an initial triangulation for a given input geometry is firstly generated, then all these initial triangles are transformed into Delaunay triangles (without Steiner point insertion) by means of flipping operations, and finally all these triangles are refined by applying given criteria (minimum angle, maximum area scale, etc.). These basic procedures are depicted in Figure 2.3. It should be noted that:

- ❶ The input data for our 2D mesh generation module are simple polygons with a sequence of given points oriented in the counter-clockwise sense, whereas the boundary points of holes if any are oriented in the clockwise sense,
- ❷ Steiner points are only inserted in the Delaunay refinement process.

2.3.1 Data structure

The efficiency of a mesh generator rests on the efficiency of the triangulation algorithm and the data structures it employs [(Shewchuk 1996b)]. In our incremental Delaunay refinement algorithm, a triangular data structure is used for each triangle. In this structure, each triangle contains three pointers to vertices, three pointers to edges, three pointers to neighboring triangles, and one pointer to its circumcircle (see Figure 2.4). In order to improve efficiency,

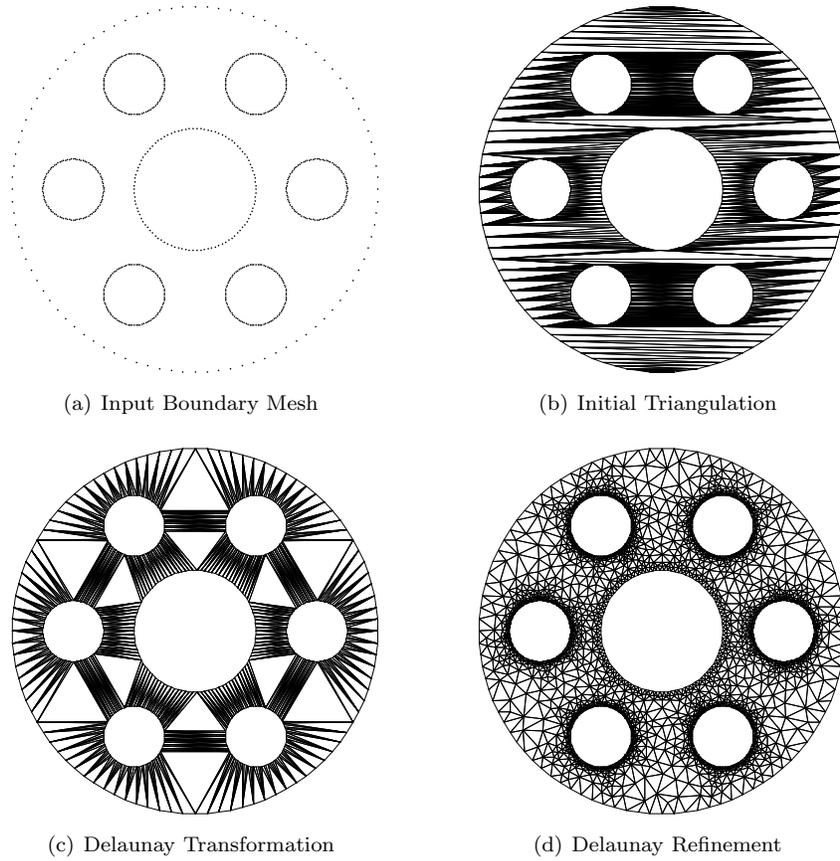


Figure 2.3: Basic steps of Delaunay Triangulation

the circumcircle is calculated only when necessary. Furthermore, for each triangle, $\angle A$ is always its largest angle, and $\angle C$ is always its smallest angle. This also means that $\text{Length}(BC) \geq \text{Length}(AC) \geq \text{Length}(AB)$ for all the triangles constructed. We will explain later why this property is extremely important for our refinement algorithm. However, it should be noted that the triangles of the final mesh may not satisfy this rule since the vertices of all triangles are eventually reordered in counterclockwise orientation for numerical purpose.

Very similar to the Triangle program developed by Shewchuk, our mesh generator also uses a splay tree data structure to store all the generated triangles, while two ordering/searching keys are employed for different refinement stages. A splay tree is a kind of self-adjusting balanced binary search tree. It

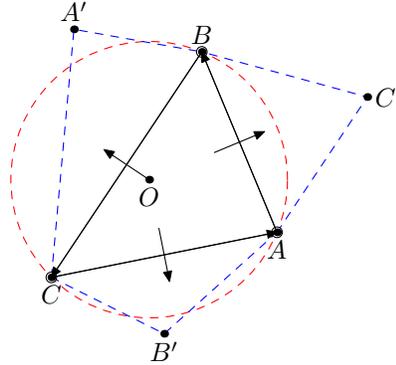


Figure 2.4: Data structure for single triangle.
for each triangle, $\angle A \geq \angle B \geq \angle C$

guarantees that those frequently accessed items are located near the top of the tree by rotations. All splay tree operations run in $O(\log n)$ time on average. Particularly for our refinement process, only searching the triangles with minimum angle (or maximum area) is needed, so the average time efficiency will not be so bad compared with an AVL tree, but more space efficient. However, since the splay tree ordering/searching keys may be equivalent for different inserted triangles, slightly decreasing or increasing these keys will be needed when a new triangle with duplicated key is inserted into the splay tree.

2.3.2 Initial triangulation

As previously mentioned, the input data are a simple polygon with a sequence of given points oriented in the counter-clockwise sense, whereas the boundary points of holes are oriented in the clockwise sense. This initial triangulation problem is also called the polygon triangulation problem, which can be stated as follows: for a sequence of n given points, find $n - 3$ diagonals that partition this polygon into $n - 2$ triangles (for a polygon with m holes, the total number of triangles will be $(n - 2) + 2 * m$). Polygon triangulation is a classic problem in computational geometry with great appeal for nearly one century, and it has received widespread interest over the last two decades. The first simple polygon triangulation “algorithm” was proposed by Lennes [(Lennes 1911)] in 1911 via recursive diagonal insertion with $O(n^2)$ expected running time [(O’Rourke 2000)]. In 1975, Meisters [(Meisters 1975)] presented an $O(n^3)$ algorithm based on the ear-cutting method. Garey et al [(Garey

et al. 1978]) were the first to publish an $O(n \log n)$ algorithm via polygon decomposition and sweeping in 1978. Four years later another algorithm with the same complexity was presented by Chazelle [(Chazelle and Incerpi 1984)] using the divide-and-conquer technique. Later, Hertel and Mehlhorn [(Hertel and Mehlhorn 1983)] presented their $O(n + r \log r)$ algorithm in 1983, where r is the number of reflex vertices, and Tarjan et al [(L.J. Guibas et al. 1987)] [(Tarjan and Wyk 1988)] [(Kirkpatrick et al. 1990)] proposed an $O(n \log \log n)$ algorithm via “involved” data structures. In 1988, Toussaint [(Toussaint 1991)] proposed an $O(n(1 + t_0))$ algorithm using the sleeve searching method, where t_0 is the number of “free triangles” in the output triangulation. In 1993, ElGindy, Everett and Toussaint [(ElGindy et al. 1993)] found that Meister’s algorithm can be performed in $O(n^2)$ time with the “prune and search” technique, Kong, Everett and Toussaint [(Kong et al. 1990)] also implemented an $O(kn)$ algorithm by Graham’s scan method, where $k - 1$ is the number of concave vertices in the polygon. A breakthrough in polygon triangulation algorithms was made by Chazelle [(Chazelle 1991)] in 1990, whose algorithm showed that a nearly linear-time polygon triangulation can be reached through randomized trapezoidation techniques. However, since the original algorithm is quite complicated and very difficult to implement, several improvements of this method were published by Seidel [(Seidel 1991)] and Amato [(Amato et al. 2000)] in 1991 and 2000 respectively. “Is there a deterministic, linear-time polygon triangulation algorithm significantly simpler than that of Chazelle?” is still an open problem in the computational geometry community.

There are at least three popular polygon triangulation algorithms: the recursive ear cutting algorithm as improved by Toussaint [(Toussaint 1991)] with $O(kn)$ time complexity, where k is the number of concave vertices of the polygon; the sweep line algorithm presented by Garey [(Garey et al. 1978)] and the incremental randomized algorithm by Seidel [(Seidel 1991)] and Amato [(Amato et al. 2000)], both with $O(n \log n)$ time complexity. Among them, the recursive ear cutting algorithm has the worst performance and also exhibits difficulties in the handling of polygon with holes, although it is very easy to implement compared with other complicated algorithms. On the other hand, the incremental randomized algorithm has better performance, but the improved algorithm presented by Amato [(Amato et al. 2000)] is very difficult to implement. The first linear algorithm proposed by Chazelle [(Chazelle 1991)] is extremely difficult and no implementation of this algorithm was reported up to now according to the author’s knowledge. The sweep line algorithm is easier to implement when compared with Chazelle’s linear algorithm and has gained popularity in nowadays applications due to its efficiency, robustness and capability to handle

polygons with holes.

In this subsection, we will first briefly review the three popular simple polygon triangulation algorithms addressed here above, and then we will focus on the sweep line algorithm improvements and implementation. Finally, we will finish this subsection by comparing our implementation with other major popular codes available from the web. Before reviewing the three popular polygon triangulation algorithms, several basic definitions and theories have to be introduced.

2.3.1. DEFINITION. A *simple polygon* P is a polygon with no pair of non-consecutive intersecting edges.

A simple polygon P can be represented by n consecutive points v_1, v_2, \dots, v_n oriented in the counterclockwise or clockwise sense. These points are called the vertices of polygon P . Then $(v_1, v_2), (v_2, v_3), \dots, (v_n, v_1)$ are n line segments, called the edges of polygon P . Every simple polygon (without holes) with n vertices ($n \geq 4$) may be partitioned into $n - 2$ triangles by adding $n - 3$ diagonals. This conclusion was proved by Meisters [(Meisters 1975)]. To proof this statement, Meisters introduced the concept of **ears** of a simple polygon as follows.

2.3.2. DEFINITION. A vertex v_i of a simple polygon P is called an *ear* if the line segment (v_{i-1}, v_{i+1}) that bridges v_i is a diagonal.

The so called **Two-Ears Theorem** states that every simple polygon has at least two non-overlapping ears, except triangles.

The above definition and theorems imply that a straightforward algorithm to triangulate a simple polygon consists in searching and cutting off the ears of the polygon until the left polygon is a triangle. In this algorithm, all the vertices of polygon P are stored in a circular doubly-linked list. The algorithm starts at vertex v_2 and checks whether the previous vertex v_1 is an ear. If this is not the case, the current vertex is moved forward. If the previous vertex is an ear, it is cut off and removed from the polygon. The current vertex is not moved forward in this case except if it is the vertex following v_0 in order to prevent v_0 from being cut off. The algorithm stops when it reaches v_0 . This algorithm has a worst $O(n^2)$ running time complexity and its real running time complexity is $O(kn)$, where k is the number of concave vertices of the polygon. If the polygon has few concave vertices the time complexity becomes nearly linear. If all the vertices of a simple polygon P are convex, then this polygon can be triangulated in $O(1)$ time complexity. Actually, a simple polygon without any

concave vertex can be triangulated by simply drawing a diagonal from a given vertex v_i to all the other vertices that are not neighbors of v_i .

Certain classes of polygons can be triangulated easily, for example, the convex polygons. The question arises whether it is possible that a simple polygon can be partitioned into convex polygons. Answering this question is not a easy task. The first algorithm for finding an optimal polygon convex partitioning with diagonals was due to Greene [(Greene 1983)] with $O(n^4)$ time complexity and was subsequently improved to $O(n^3 \log n)$ by Keil [(Keil 1985)]. However, both of these algorithms are not efficient enough for triangulation. Nevertheless, another class of polygons that can be easily triangulated in linear time is the class of monotone polygons.

2.3.3. DEFINITION. *A simple polygon is called **monotone** with respect to a line l if for any line l' perpendicular to l the intersection of the polygon with l' is connected. A polygon that is monotone with respect to the y -axis is called **y -monotone**.*

We have to pay special attention to vertices with equal y -coordinates. One way to avoid this problem consists in slightly rotating the whole polygon in the clockwise direction with respect to the coordinate system. However, a more reasonable way to solve this problem is obtained by defining the **below** and **above** notions as follows:

2.3.4. DEFINITION. *a vertex p is **below** another vertex q if $p_y < q_y$ or $p_y = q_y$ and $p_x > q_x$, and p is **above** q if $p_y > q_y$ or $p_y = q_y$ and $p_x < q_x$.*

2.3.5. DEFINITION. *An **interior cusp** of a polygon is a vertex v whose adjacent vertices are either both at or above, or both at or below, v .*

2.3.1. LEMMA. *If a polygon P has no interior cusp, then it is monotone.*

2.3.2. LEMMA. *A strictly y -monotone polygon with n vertices can be triangulated in linear time.*

To overcome the poor performance of the recursive ear-cutting algorithm and handle polygons with holes, the trapezoidal decomposition algorithm was proposed by Seidel [(Seidel 1991)] and the sweep line algorithm was presented and improved by by Garey et al [(Garey et al. 1978)] respectively. Both the trapezoidal decomposition algorithm and the sweep line algorithm are based on the fact that all simple polygons can be partitioned into monotone parts, which can be triangulated efficiently in linear time. The difference between these two algorithms is the way the polygons are decomposed. The former

algorithm employs a horizontal or vertical trapezoidation line through every vertex of the polygon to decompose the polygon into trapezoids, and then introduces diagonals to remove the interior cusps. On the contrary, in the sweep line algorithm, the diagonals are immediately introduced at all turn vertices (defined here below) when a horizontal or vertical line sweeps through the polygon.

In the sweep line algorithm, all the literature distinguishes five types of vertices in order to decompose the polygon into monotone pieces. Four of these types are **turn vertices**: the **start vertices**, the **split vertices**, the **end vertices**, and the **merge vertices**. The vertices that are not turn vertices are **regular vertices**. These types of vertices are defined as follows:

2.3.6. DEFINITION. *A vertex v is a **start vertex** if its two neighbors lie below it and the interior angle at v is lower than 180° ; if the interior angle is higher than 180° , then v is a **split vertex**. A vertex is an **end vertex** if its two neighbors lie above it and the interior angle at v is lower than 180° ; if the interior angle is higher than 180° then v is a **merge vertex**.*

Theory [(de Berg et al. 2000)] shows that a polygon is y-monotone if it has no split vertices or merge vertices. Therefore, the goal of the y-monotone decomposition is to get rid of the polygon split and merge vertices by adding a diagonal going upwards from each split vertex and a diagonal going downwards from each merge vertex. More specifically, for each split vertex v , a diagonal is inserted from it to the lowest vertex above it. On the contrary, for each merge vertex, a diagonal is inserted to the highest vertex below it. This lowest/highest vertex is often called the **helper** of direct left edge e_i , or simply **helper**(e_i), where e_i is the direct left edge of vertex v_i . Mark et al and O'Rourke fully summarized the sweep line algorithm in [(de Berg et al. 2000)] and in [(O'Rourke 2000)] respectively and this issue will not be addressed here. However, there is a big disadvantage in the current algorithm: for each regular vertex v_i , we still have to know whether the interior of the polygon lies to the right or to the left of v_i when the sweeping line reaches the regular vertex v_i . This means that additional computations and storage space to correctly handle all regular vertices are needed. To improve the efficiency and decrease the storage space, we classify the **regular vertices** into **regular down vertices** and **regular up vertices** as follows (see Figure 2.5):

2.3.7. DEFINITION. *A **regular down vertex** is a regular vertex whose previous neighbor vertex v_i^- lies above it, while the next neighbor v_i^+ lies below it. On the contrary, a **regular up vertex** is a regular vertex whose previous neighbor v_i^- lies below it, while the next neighbor v_i^+ lies above it.*

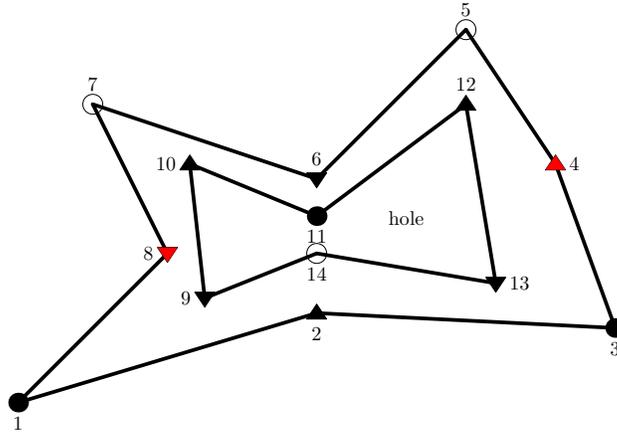
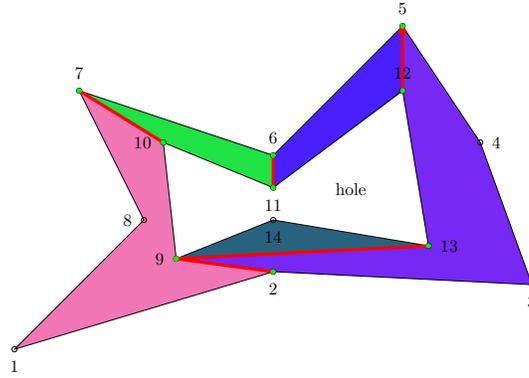


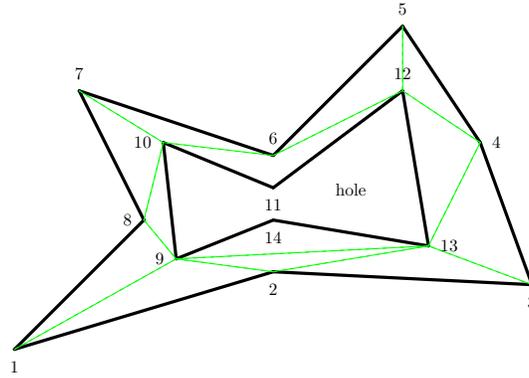
Figure 2.5: Six types of vertices: Start Vertex (5,7,14), End Vertex (1,3,11), Split Vertex (2,10,12), Merge Vertex (6,9,13), Regular Up Vertex (4), and Regular Down Vertex (8).

This classification is based on the fact that for a polygon oriented in the counter-clockwise direction (with holes in the clockwise direction), the interior of the polygon is always located to the left of the regular up vertices, and to the right of the regular down vertices. Therefore any additional computation and storage space for such kind of situation is not needed anymore when handling a regular vertex.

When the sweeping line reaches vertex v_i , a basic operation is to find the edge located to the left of this vertex. Therefore, the edges of the polygon intersecting the sweep line are stored in a dynamic splay tree, and the sorting/searching keys are the x coordinates of the intersections. Note that all the sorting/searching keys of the edges in the splay tree have to be dynamically updated at each event vertex. Once a diagonal is inserted, an auxiliary diagonal is inserted at the same time in the opposite direction (note that polygon edges and diagonals are directed line segments in our algorithm) for monotone piece searching purpose only. All the monotone pieces can be easily constructed using these auxiliary diagonals since each diagonal is always shared by two adjacent monotone polygons. Finally, all the constructed monotone polygons can be triangulated in linear time by the algorithm described in [(de Berg et al. 2000)][(O'Rourke 2000)]. It should be noted that correct calculation of the interior angle of a vertex v is extremely important for the robustness of the sweep line algorithm, hence the arbitrary precision floating-point arithmetics and fast robust geometry predicates from Shewchuk [(Shewchuk 1996a)] are



(a) Monotone polygons obtained by adding diagonals



(b) Final triangulation

Figure 2.6: Sweep line algorithm

used to ensure the robustness of our implementation. Figure 2.7 and Figure 2.8 show polygons with/without holes and the associated shadowed monotone pieces, polygon triangulation and polygon Delaunay triangulation, respectively.

Practical experience and numerical experiments show that our initial triangulation code ¹ implemented using the sweep line algorithm is efficient and robust due to our algorithm improvements and well-designed data structures, and

¹Poly2tri, one of the fastest robust simple polygon triangulation code open to public and using the sweep line algorithm, is now used by some academic projects from Europe and North American. Poly2tri is available in C++, Java on Linux/Unix and Microsoft Windows platforms from <http://www.mema.ucl.ac.be/~wu/Poly2Tri/poly2tri.html>

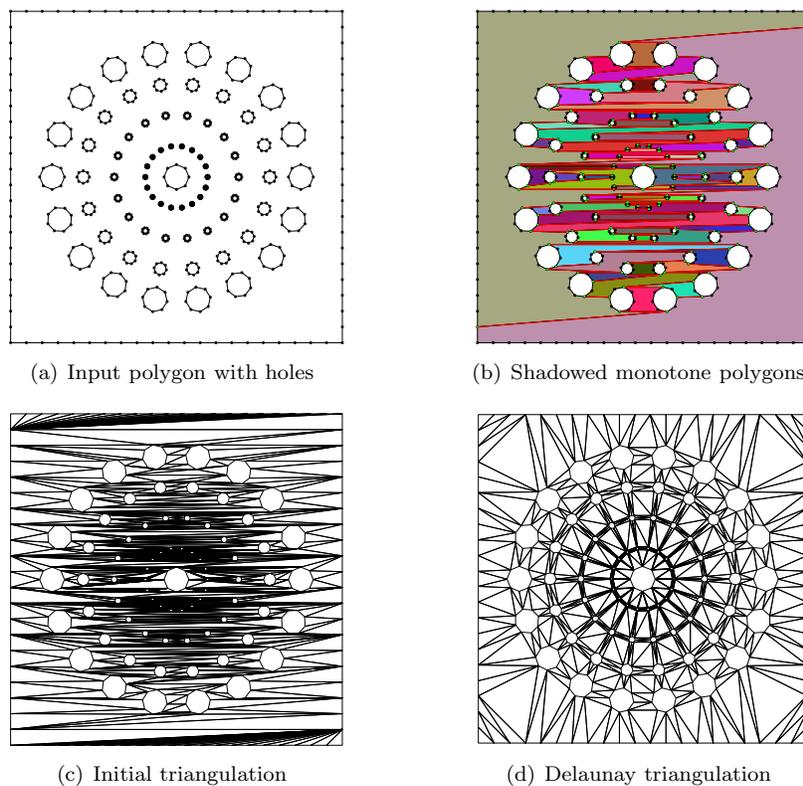


Figure 2.7: Sweep line algorithm for polygon with holes

the use of exact floating-point arithmetic predicates. Comparisons with other popular triangulation codes like “Triangle” as developed by Jonathan with divide-and-conquer, incremental and sweep line algorithms [(Shewchuk 1996b)], “FIST” as developed by Held with “fancy” ear-clipping algorithm [(Held 2001)], and “triangulation” as implemented by Narkhede and Manocha using Seidel’s incremental randomize algorithm [(Narkhede and Manocha 1995)] are presented in Table 2.3.2. Our primary comparisons show that our implementation is very competitive and tends to be faster than several other above mentioned popular codes. However, the performance of our implementation is extremely poor compared with FIST and Triangle for those random polygons that zig-zag widely. The reason is simple: to triangulate such kinds of polygons, the size of the splay tree (to locate the direct left edge of an event vertex) is very much larger than for “smoother” polygons with small edges, and hence most of the time is spent

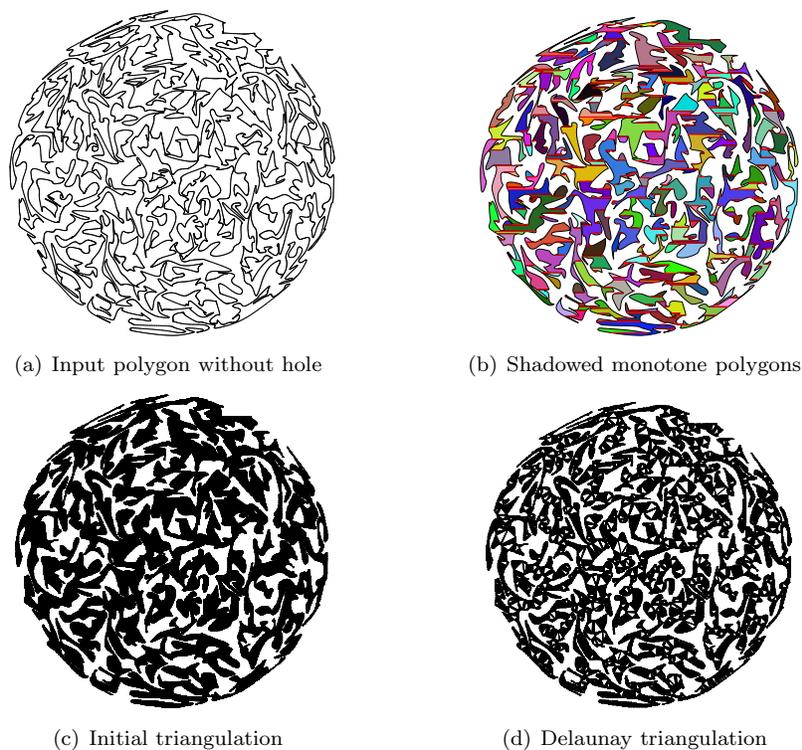


Figure 2.8: Sweep line algorithm for polygon without hole

on splay tree transversing and search-key updating. However, in practice, our high-quality 1D mesh generator can always guarantee that the input polygons (generated from 1D boundary meshes) are “smooth” enough for our real Finite Element applications.

Vertices	triangulation	Poly2Tri	FIST	Triangle			
				divide-and-conquer	incremental	Sweepline	vertical cuts only
857	0.007(s)	0.003(s)	0.008(s)	0.005(s)	0.015(s)	0.006(s)	0.005(s)
6937	0.086(s)	0.040(s)	0.094(s)	0.113(s)	0.773(s)	0.136(s)	0.133(s)
10366	0.139(s)	0.063(s)	0.159(s)	0.155(s)	1.480(s)	0.173(s)	0.166(s)
20670	0.285(s)	0.135(s)	0.445(s)	0.286(s)	5.554(s)	0.329(s)	0.307(s)
68372	1.065(s)	0.601(s)	1.875(s)	0.747(s)	95.251(s)	0.959(s)	0.797(s)
103079	1.614(s)	0.785(s)	5.173(s)	1.153(s)	179.331(s)	1.458(s)	1.264(s)
573440*	failed	7.487(s)	55.641(s)	8.451(s)	>2.7(h)	8.596(s)	9.008(s)
1048576*	failed	13.942(s)	155,725(s)	14.682(s)	>2.7(h)	17.300(s)	18.482(s)
1756160*	failed	32.177(s)	132.725(s)	22.629(s)	>2.7(h)	32.208(s)	29.517(s)
32768*	failed	24.795(s)	5.216(s)	1.692(s)	1.410(s)	1.967(s)	1.451(s)

Table 2.1: Running time comparisons with other popular triangulation codes.
Platform: Mandrake Linux 10.1 with Intel PIII 1.2GHZ Mobile processor, 512M SDRAM and compiled by gcc 3.4.1 with -O2 option. The last four polygonal raw data (30~120 Megabytes) gcc 3.4.1 with -O2 option. The last four polygonal raw data (30~120 Megabytes) marked by “*” are four different random polygons contributed by Prof. Martin Held.

2.3.3 Incremental Delaunay refinement algorithm

The initial mesh generation algorithms reviewed in the previous subsection are not suitable for the Finite Element Method (FEM), since in many applications the numerical stability and convergence are strongly affected by the element shapes, and excessively “long and skinny” elements can lead to undesirable behavior. Delaunay refinement algorithms were developed to offer a shape guarantee on the generated elements, such as to provide non-obtuse triangles, or all triangles with a bounded *aspect ratio*. The *aspect ratio* of a triangle is the highest ratio of its edge length divided by the associated altitude. A fairly general measure of a triangle shape is given by its minimum angle α , since this gives a $\pi - 2\alpha$ bound to its maximum angle and guarantees an *aspect ratio* between $1/\sin \alpha$ and $2/\sin \alpha$. Compared with other refinement techniques, the Delaunay refinement method is arguably the most popular method due to its theoretical guarantee and practical performance.

The incremental Delaunay refinement algorithm inserts one Steiner point at a time while maintaining a Delaunay triangulation until all triangles meet the imposed constraints on element quality and size. The basic idea of Ruppert’s Delaunay refinement algorithm is to make local improvements by inserting Steiner points at the circumcircle of skinny triangles while maintaining the Delaunay property. Moreover, if the inserted Steiner point *encroaches upon* any input boundary segment, then the boundary segment will be split at its *diametral circle* center. However unlike in Ruppert’s algorithm, splitting the encroached boundary segments is ignored during the refinement process in our algorithm. This comes from two reasons. Firstly, the thermal, stress and flow calculations of the different furnace components (or “macro-elements”) are decoupled in our Finite Element model, and hence we need to separately generate unstructured meshes for the different crystal growth furnace components. Moreover, at some stage of the algorithm, coupling the whole system is needed. Therefore, avoiding to split the boundary line segments is necessary in order to keep the boundary consistency of these components. Secondly, we need a better approach to exactly control the mesh density along the boundaries (in particular in order to well-handle the melt boundary layers). This issue will be addressed in the next subsection.

When a Steiner point is inserted, a good way to maintain the Delaunay property is to use the “delete and build” algorithm introduced by Bowyer/Waston [(Bowyer 1981)][(Waston 1981)]. In the Bowyer/Waston algorithm, each triangle whose circumcircle encloses the new inserted Steiner point is no longer “Delaunay” and hence should be deleted. All the other triangles remain Delau-

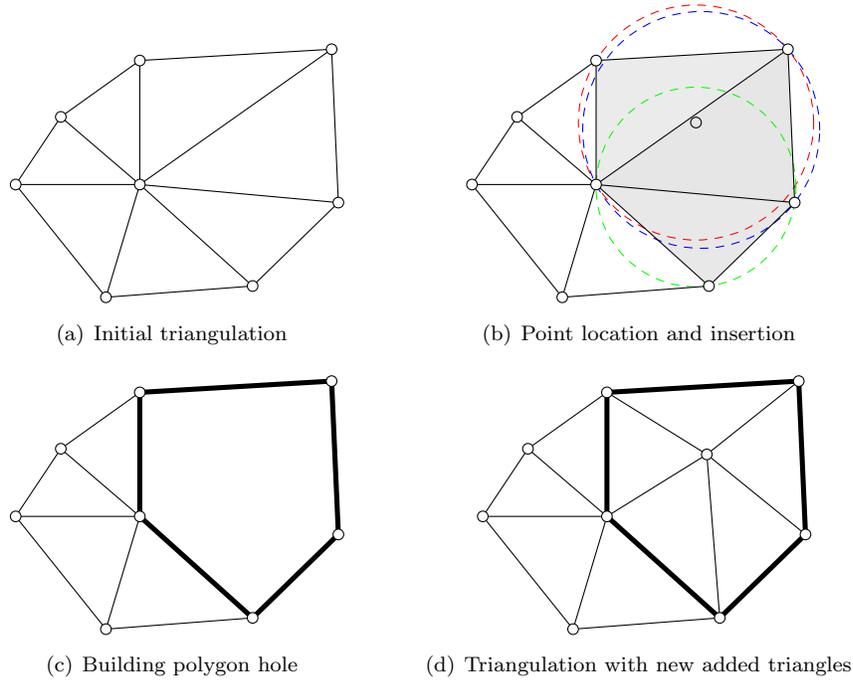


Figure 2.9: Waston's "delete-and-build" algorithm

may and are left undisturbed. The set of deleted triangles forms a "polygonal hole" in the triangulation. New triangles are then inserted by connecting each vertex of this polygon to the newly created vertex with a new edge. This algorithm is illustrated in Figure 2.9.

Compared with Bowyer/Waston's "delete and build" algorithm, a so called "split and flip" algorithm has been proposed and implemented. In our "split and flip" algorithm, building the "polygonal hole" is not necessary. However, a Steiner point is inserted at the bad triangle circumcircle center and the triangle which contains this Steiner point is simply removed and split while three new triangles are inserted into the Delaunay triangulation. Finally, any set of triangles which break the Delaunay criterion will be transformed into Delaunay triangles by flipping operations. This "split and flip" algorithm is illustrated in Figure 2.10. The highest advantage of this algorithm is that a simple generic refinement algorithm can be implemented by different refinement criteria.

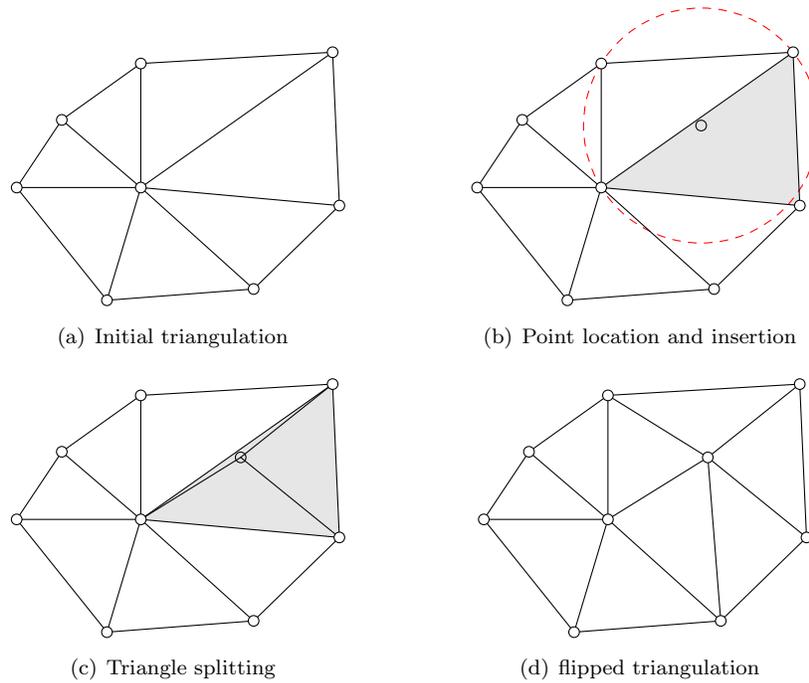


Figure 2.10: Our “split-and-flip” algorithm

```

Function Refinement_by_Criterion(given criterion)
  found bad triangle=true;
  while(found bad triangle)
  {
    if(triangle  $\mathcal{T}$  that doesn't meet given criterion is found)
    {
      Search the triangle  $\mathcal{T}'$  which encloses the circumcircle center  $\mathcal{O}$  of  $\mathcal{T}$ ;
      Remove triangle  $\mathcal{T}'$ ;
      Split  $\mathcal{T}'$  by  $\mathcal{O}$  and add new split triangles  $\mathcal{T}'_1, \mathcal{T}'_2, \mathcal{T}'_3$ ;
      Validate new inserted triangles  $\mathcal{T}'_1, \mathcal{T}'_2, \mathcal{T}'_3$ 
    } else found bad triangle=false;
  }
End

```

According to our refinement criterion, there are four kinds of bad triangles:

1. **Needle**, a triangle with very small minimum angle but whose circumcircle center is located inside itself, see Figure 2.11(a).

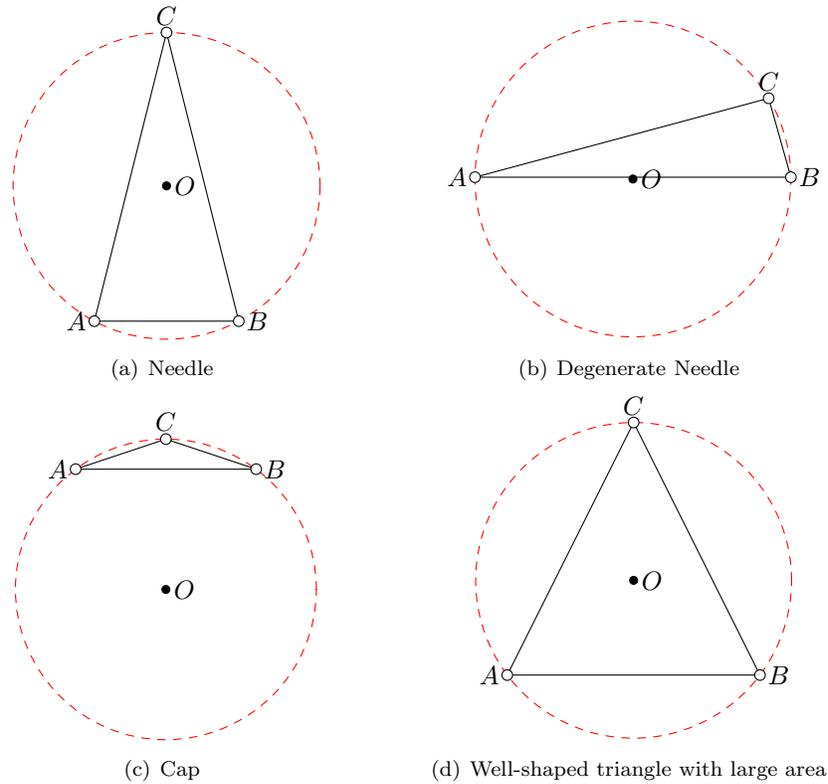


Figure 2.11: Three Categories of bad triangles and a good one with large area

2. **Degenerate needle**, Figure 2.11(b) can be viewed as a degenerate case of Figure 2.11(a), where the circumcircle center is located at the triangle edge unoccasionally.
3. **Cap**, a triangle with very large maximum angle, and whose circumcircle center is located outside the triangle itself, see Figure 2.11(c).
4. **Large area triangle**, Figure 2.11(d) is a well-shaped triangle but with very large area. Such kinds of triangles with a relatively large area sometimes should also be avoided in numerical simulations for accuracy reasons, although they are not bad triangles according to the shape quality criterion.

All kinds of above described bad triangles should be removed during the Delaunay refinement process in order to generate shape and size quality trian-

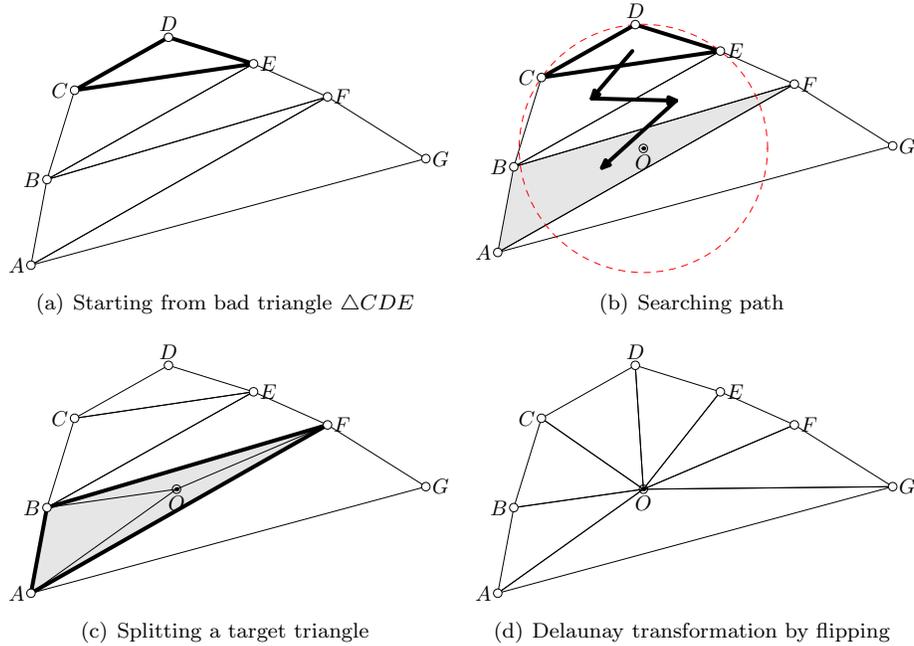


Figure 2.12: Typical point query and refinement process

gles. Figures 2.11(a) and 2.11(c) are easy to handle cases during a Delaunay refinement process based on our “split and flip” algorithm. Bad triangles like in Figure 2.11(b) should be carefully handled since their circumcircle center is located unoccasionally on their edge. Using our “split and flip” algorithm will immediately let the triangulation process fail since a degenerate triangle with three collinear vertices will be generated. In practical implementations, a slight deviation of the circumcircle center can avoid such kind of failure.

However for bad triangles like in Figure 2.11(d), we need to find where the circumcircle center is located. This is a so-called point location problem, which is a key issue for the incremental Delaunay refinement algorithm. Generalized point location algorithms such as the “bucketing” algorithm due to Asano et al [(Asano et al. 1985)] achieve optimal logarithmic time complexity and require some extra processing, additional storage and complicated data structures. On the other hand, the algorithms based on trapezoidal maps firstly have to compute the trapezoidal map in $O(n \log n)$ expected time and then do query for each point in $O(\log n)$ expected time. Finally, the randomized algorithm by P.Mucke et al [(Mucke et al. 1996)] presents a popular technique with $O(n^{1/3})$

time complexity for each query without using any additional preprocessing time and storage space, but as a bonus, it cannot be easily implemented. Here, we propose the first linear point location algorithm specifically developed for the incremental Delaunay refinement process without using any additional preprocessing, storage space and complicated data structures, while being easy to implement at the same time. In this algorithm, each query can be performed in a constant time and the time complexity for this query is independent of the mesh size. The basic idea of our algorithm is quite the same as the one from P.Mucke et al. However the big difference is that we do not need to generate a randomized “good” starting point while the query begins immediately from the bad triangle found. The target triangle can be found step by step by walking across the largest edges of the adjacent triangles. A typical point query and refinement process is illustrated in Figure 2.12. The point location problem can be easily solved on the basis of the triangular data structure and the following lemmas.

2.3.3. LEMMA. *For a Delaunay triangulation \mathcal{DT} , let \mathcal{T} be a skinny triangle with maximum angle larger than 90° , \mathcal{T}'_1 be the adjacent triangle along the largest edge of \mathcal{T} , and \mathcal{T}'_2 be the adjacent triangle along the largest edge of \mathcal{T}'_1 , while \mathcal{O} and \mathcal{O}'_1 are the circumcircle centers of triangles \mathcal{T} and \mathcal{T}'_1 respectively. Then the following **theorems** hold:*

- ① *If the circumcircle center \mathcal{O} of \mathcal{T} is located outside triangle \mathcal{T}'_1 , the maximum angle of \mathcal{T}'_1 is larger than 90° .*
- ② *If the circumcircle center \mathcal{O} of \mathcal{T} is located outside triangle \mathcal{T}'_2 , the circumcircle center \mathcal{O}'_1 of \mathcal{T}'_1 is also located outside triangle \mathcal{T}'_2 .*
- ③ *If \mathcal{O} is located outside triangle \mathcal{T}'_2 , the maximum angle of \mathcal{T}'_2 is also larger than 90° .*

Proof: To prove the first conclusion, let $\triangle CDE$ be the skinny triangle \mathcal{T} with maximum angle $\angle CDE$ larger than 90° and the adjacent triangle \mathcal{T}'_1 on the largest edge CE be $\triangle BCE$, see Figure 2.13(a). Connect line segment $E\mathcal{O}$ and extend it until it intersects the circumcircle of \mathcal{T} at E' , and then connect C with E' , connect line segment $C\mathcal{O}$ and extend it until it intersects the circumcircle of \mathcal{T} at C' , and finally connect E with C' . Since \mathcal{O} is located outside $\triangle BCE$ while $\triangle CDE$ and $\triangle BCE$ are Delaunay triangles, so the only possible locations of point B are the shadow regions illustrated in Figure 2.13(a). Then either $\angle BCE > \angle ECE' = 90^\circ$ or $\angle BEC > \angle CEC' = 90^\circ$. It should be noted that the inverse of this conclusion is not valid. In Figure 2.13(b) for example, $\triangle BCE$ is a triangle whose maximum angle $\angle BCE$ is larger than 90° , whereas the circumcircle center of $\triangle CDE$ is located inside $\triangle BCE$.

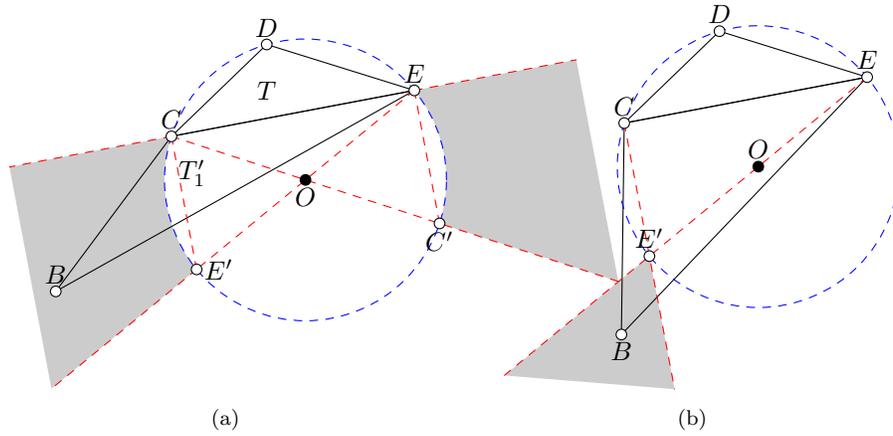


Figure 2.13: If O outside triangle T' , the maximum angle of $T' > 90^\circ$

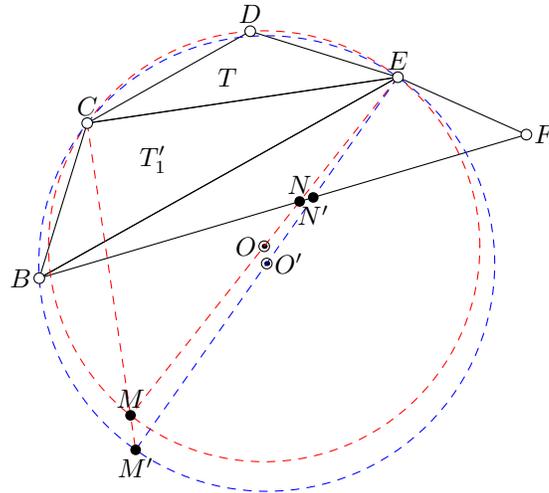


Figure 2.14: If O' outside $\triangle BEF$, so does O

To prove the second conclusion, see Figure 2.14. Connect line segment EO and extend it until it intersects the circumcircle of T at M , and then connect line segment EO' and extend it until it intersects the circumcircle of T_1' at M' . The line segment BF intersects EM and EM' at N and N' respectively. Since the maximum number of intersections of two circumcircles is two, C and E

are two such known intersections while B is located outside the circumcircle of triangle \mathcal{T} , and hence $EM' > EM$, which means that the circumcircle radius of triangle \mathcal{T}'_1 is larger than that of triangle \mathcal{T} . Also $\angle NEC < \angle N'EC < \angle MCE = 90^\circ$, and hence $NE > N'E$. Since \mathcal{O} is located outside triangle \mathcal{T}'_2 while $\mathcal{O}E < \mathcal{O}'E$, then \mathcal{O}' is located outside triangle \mathcal{T}'_2 ($\triangle BEF$).

The third conclusion can be derived from the two previous ones: since \mathcal{O} is located outside triangle \mathcal{T}'_2 , we know that \mathcal{O}' is also located outside triangle \mathcal{T}'_2 . Since $\angle BCE$ is larger than 90° , we can immediately conclude that $\angle BEF$ is larger than 90° according to the first conclusion.

2.3.4. LEMMA. *For a Delaunay triangulation \mathcal{DT} without holes, assume that \mathcal{T} is a skinny triangle with its maximum angle larger than 90° , \mathcal{T}'_1 is the adjacent triangle along the largest edge of \mathcal{T} , \mathcal{T}'_2 is the adjacent triangle along the largest edge of \mathcal{T}'_1 , and so on. If the triangle \mathcal{T}'_n encloses the circumcircle center \mathcal{O} of \mathcal{T} , then there always exists a shortest path from \mathcal{T} to its circumcircle center \mathcal{O} as obtained by walking across the largest edges of \mathcal{T} , \mathcal{T}'_1 , \mathcal{T}'_2 , ..., \mathcal{T}'_n .*

Proof: We first have to prove the existence of this path. Actually, the corresponding dual graph $\mathcal{DG}(\mathcal{T})$ of the Delaunay triangulation \mathcal{DT} is a planar, undirected and connected graph. Therefore, there always exists a path from any site P_i to site P_j in $\mathcal{DG}(\mathcal{T})$. The fact that the path walking across the largest edges is the shortest path can be proved by induction, but this is very complicated and will not be detailed here.

Finally, to determine the time complexity of the algorithm, we have the following lemma:

2.3.5. LEMMA. *For a Delaunay triangulation \mathcal{DT} with minimum angle α for a domain without holes, if α and β are the minimum and maximum angles of triangle \mathcal{T} and $\beta > 90^\circ$, then the length n of the shortest searching path to the circumcircle center of \mathcal{T} (as obtained by walking across the largest adjacent triangles of the triangulation) is bounded by the following equation:*

$$1 \leq n \leq \lfloor \frac{\beta - 90^\circ}{\alpha} + 1 \rfloor \quad (2.1)$$

Proof: Based on our previous lemmas, we know that, if a triangle \mathcal{T} has a maximum angle β larger than 90° , then its circumcircle center \mathcal{O} will lie outside \mathcal{T} . It is straightforward to show that there always exists adjacent triangles whose minimum angle is larger than α on the longest edge of \mathcal{T} and which enclose \mathcal{O} . Now let us see the worst case illustrated in Figure (2.15). Assume that the minimum angle α of \mathcal{DT} is $\angle ECD$ and that the maximum

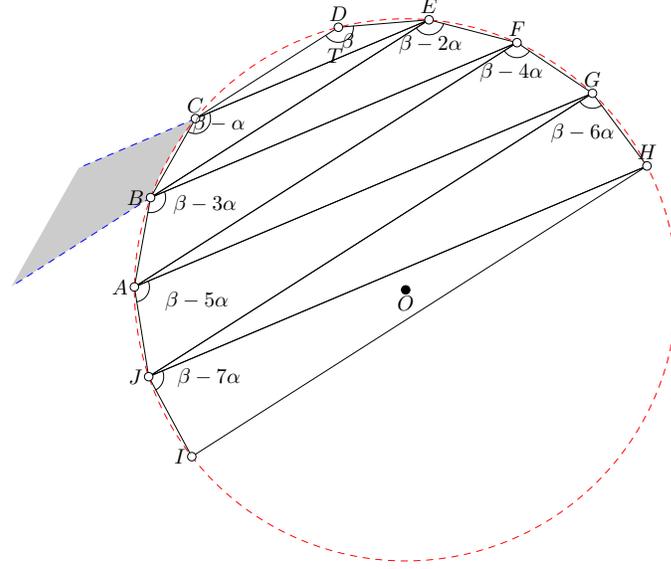


Figure 2.15: Point location searching path: the worst-case

angle β of the skinny triangle $\triangle CDE$ is $\angle CDE$. Draw a parallel line to CD from E , which intersects the circumcircle of $\triangle CDE$ at B , and then extend EB and EC . If B lies in the shadow region, then the minimum angle of \mathcal{DT} will be smaller than α and the worst skinny triangle will be $\triangle BCE$, not $\triangle CDE$. If B lies on the extension of line segment EB or below the shadow region, then $\triangle BCE$ or its adjacent triangles will be closer to the circumcircle center \mathcal{O} . Do the same from B, F, \dots , until a triangle is found which encloses \mathcal{O} . This is the longest searching path (as obtained by walking across the largest adjacent triangles of the triangulation) from $\triangle CDE$ to \mathcal{O} . If the circumcircle center \mathcal{O} lies inside the n^{th} triangle, it is easy to show that for the $(n-1)^{\text{th}}$ triangle, the maximum angle will be $\beta - (n-1)\alpha$, in such a way that

$$\begin{cases} \beta - n\alpha < 90^\circ \\ \beta - (n-1)\alpha > 90^\circ \end{cases}$$

that is,

$$\frac{\beta - 90^\circ}{\alpha} < n < \frac{\beta - 90^\circ}{\alpha} + 1 \quad (2.2)$$

or

$$n = \lfloor \frac{\beta - 90^\circ}{\alpha} + 1 \rfloor \quad (2.3)$$

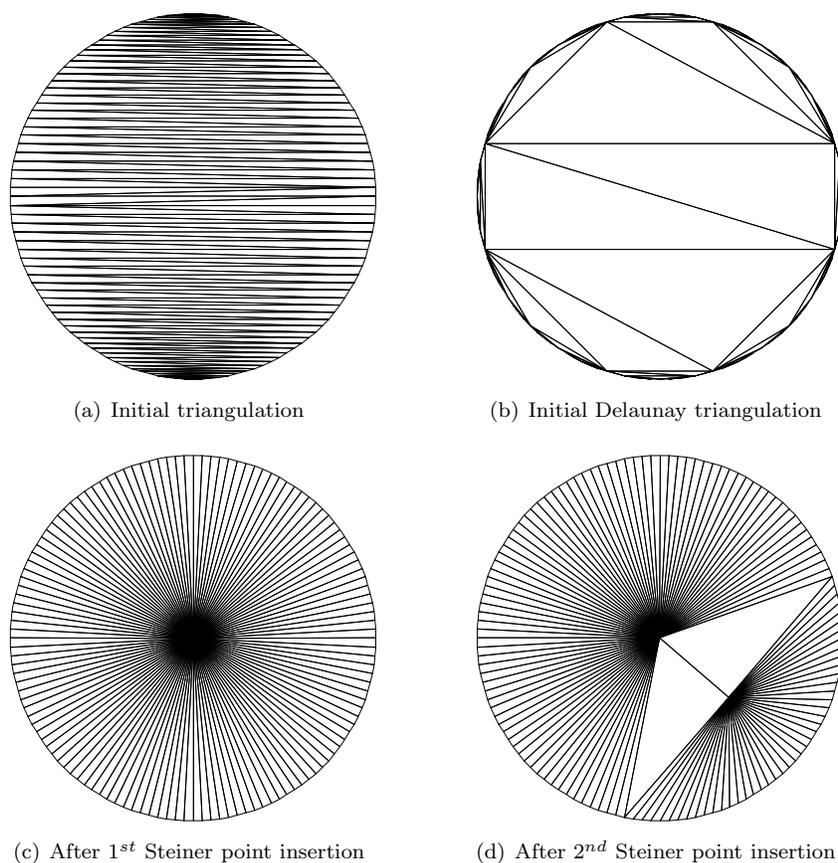


Figure 2.16: *Delaunay refinement process*

The above lemma shows that the point location problem can be efficiently solved on the basis of our previously introduced triangle-based data structure and splay-tree searching/sorting algorithms. For example, letting $\alpha = 12.5^\circ$, $\beta = 150^\circ$, then the length of the shortest searching path in the worst case is $n = \lfloor (150 - 90)/12.50 + 1 \rfloor = 5$, which means that for this skinny triangle, only 6 incircle testings are needed in the worst case. As another example, letting $\alpha = 30.5^\circ$, then $\beta \leq 119^\circ$, and $n = \lfloor (119 - 90)/30.50 + 1 \rfloor = 1$. Comparing these two examples, we can see that in the beginning of the refinement process, our point location algorithm is normally slow due to a very small minimum angle. However, the algorithm is much faster after several refinement iterations.

Let us also average some testing examples and draw some average conclusions. The first example is a circle with 128 boundary points. Figure 2.16(a) is the initial triangulation without Delaunay transformation and Figure 2.16(b) is the initial Delaunay triangulation as obtained by flipping operations. In the first iteration, a very bad triangle with minimum angle 1.40621° and maximum angle 177.188° is found. The worst-case searching time for this triangle is 64 with our theory. Our practical testing shows that it took 7 incircle testings to locate the triangle which encloses the circumcircle center. Figure 2.16(c) is the mesh after insertion of the first Steiner point. However, in the second iteration, it only took 1 test to find the target triangle. Figure 2.16(d) is the mesh obtained after the second Steiner point has been inserted. Our numerical testings have also shown that after completion of the refinement iterations, it took around 1 or 2 incircle testings to locate the target triangles while the average testing time per Steiner point is around 2.0. Our other numerous practical numerical testings and experiments have shown that 1 or 2 incircle testings in average are enough for each inserted Steiner point throughout the refinement process.

There are two main advantages in our algorithm. Firstly, the running time complexity of the point location algorithm is independent of the mesh size, which means that it is an ideal algorithm for large-scale mesh generation. Secondly, our point location method does not need any additional storage space and complicated data structure. Therefore our method is easy to implement and to adapt to other mesh generation codes using triangle-based data structure and Delaunay incremental refinement algorithm. If additional storage space is used to save the maximum angle of each triangle, the algorithm can be further improved since for those triangles with maximum angle smaller than 90° , we will immediately know that the bad triangle is the target triangle and hence no incircle testing will be necessary. Figure(2.17) compares the total number of incircle testings between our algorithm and the one proposed by P.Mucke with $O(n^{4/3})$ time complexity as implemented by Schewchuk in Triangle [(Shewchuk 1996b)].

2.4 Grade-adaptive 1D boundary mesh generation

As previously addressed, boundary segment splitting is not performed in our incremental refinement algorithm. This comes from two reasons. Firstly, thermal, stress and flow simulations for the different macro-elements are decoupled

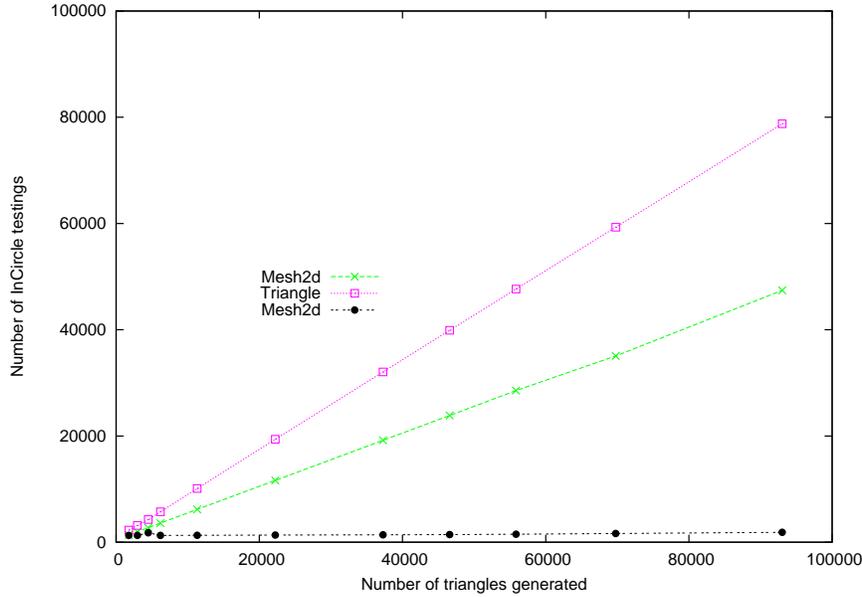


Figure 2.17: Comparison of Point Location algorithms:

$\dots \square \dots$, $O(n^{4/3})$ in Triangle;

$\dots * \dots$, $O(n)$ in Mesh2d;

$\dots \bullet \dots$, $O(n)$ in Mesh2d with additional storage.

in FEMAG software in such way that the unstructured meshes associated with different crystal growth furnace components are generated separately. However, at some stage, coupling the whole system is needed. In order to keep boundary consistency of the various components, it is necessary to ignore the boundary segment splitting operations during the mesh refinement process. Secondly, we need a better approach to exactly control the mesh density along all boundaries (to well control the boundary layers meshing) in single crystal growth processes.

Because of the above-described reasons, obtaining a high quality 1D boundary mesh with grade-adaptive density will play a prominent role to keep final 2D mesh quality and density, especially for large graded meshes arising either from geometric or physical considerations. Boundary mesh generation is also important for other mesh generation algorithms which preserve the boundary mesh and use them to generate new elements inside the domain. For example, the advancing front method requires a well-sized boundary mesh to generate high quality final elements.

In the FEMAG-2 software generation, special computer-aided geometrical

design techniques have been developed to define the furnace component shapes. For example, in Czochralski single crystal growth, the crucible shape is often defined by surfaces of different curvatures, including conical, spherical and toroidal zones. Therefore, it is generally impossible to represent such surfaces by means of analytical functions, while a simple piecewise linear curve approximation is not appropriate in view of the computational problems resulting from the flow behavior in the melt domain. However, Bézier curves [(Farin 1998)] represent a well-known tool in computational geometry, as having a simple formulation, which is also easy to implement and powerful enough to represent complex shapes. We have based all our FEMAG-2 developments on the use of cubic rational Bézier curves [(Farin 1998)], which are parametric 2D curves defined in the projective 3D space by a weighted sum of 4 control points, with the weights equal to Bernstein polynomials:

$$\begin{aligned} Q(t) &= \sum_{i=0}^3 P_i B_{i,3}(t) \\ &= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3 \end{aligned} \quad (2.4)$$

where P_0, P_1, P_2, P_3 are 4 control points, $0 \leq t \leq 1$ and $i = 0, 1, 2, 3$.

In FEMAG-2, rational Bézier curves form segments of “multi-curves”[11], and the resulting computational objects are used for the definition of the macro-element mesh boundaries, with the possibility of building complex dynamic models with deforming shapes and topological changes. In particular, these objects can efficiently support 1-D meshes, as used to form the boundaries of the 2-D finite element meshes of the system macro-elements.

A general simple method to generate grade-adaptive 1D meshes for such kinds of “multi-curves” consists in discretizing the curves with a given density (for example, the number of nodes per unit length) taking their curvature into account, in such as way that the density function σ is expressed as:

$$\sigma = \sigma(k, k', \rho) \quad (2.5)$$

where k is the given density, ρ the curvature and k' a curvature impact factor. Therefore the total number of nodes U on each “multi-curve” can be found

from the following relation:

$$\begin{aligned}
U &= \int_0^n \sigma(t) \sqrt{x'(t)^2 + y'(t)^2} dt \\
&= \sum_{i=0}^{n-1} \left(\int_i^{i+1} \sigma(t) \sqrt{x'(t)^2 + y'(t)^2} dt \right) \\
&= \sum_{i=0}^{n-1} u_i
\end{aligned} \tag{2.6}$$

and where n is the number of Bézier curves on the “multi-curve” and u_i is the number of nodes on the i^{th} Bézier curve. Assuming that the j^{th} discretized point is located on the m^{th} Bézier curve, in order to locate this point on the “multi-curve”, the unknown t_j should be determined as follows:

$$\begin{aligned}
j \cdot U / [U] &= \int_0^{t_j} \sigma(t) \sqrt{x'(t)^2 + y'(t)^2} dt \\
&= \sum_{i=0}^{m-1} u_i + \int_m^{t_j} \sigma(t) \sqrt{x'(t)^2 + y'(t)^2} dt
\end{aligned} \tag{2.7}$$

where $j = 1, 2, \dots, [U] - 1$. The above nonlinear equation is an inverse integration problem, which can be solved by Newton-Raphson method for each unknown t_j .

Based on equation (2.5), we can define the density function as follows:

$$\sigma = \frac{k + k' \rho^2}{1 + k'' \rho^2} \tag{2.8}$$

It is clear that the density function σ is bounded by $[k, k'/k'']$ (if $k < k'/k''$) or $[k'/k'', k]$ (if $k'/k'' < k$), since

$$\lim_{\rho \rightarrow \infty} \sigma = \frac{k'}{k''} = \hat{k} \tag{2.9}$$

and

$$\lim_{\rho \rightarrow 0} \sigma = k \tag{2.10}$$

Therefore, k is the density at low curvature, and \hat{k} the density at high curvature. Since more discretization intervals (points) are needed in order to better fit high-curved boundaries, we have $k < k'/k''$. Equation (2.8) can be rewritten as follows:

$$\sigma = \frac{k + k' \rho^2}{1 + k'' \rho^2} = \hat{k} - \frac{\hat{k} - k}{1 + \rho^2/k_*^2} \tag{2.11}$$

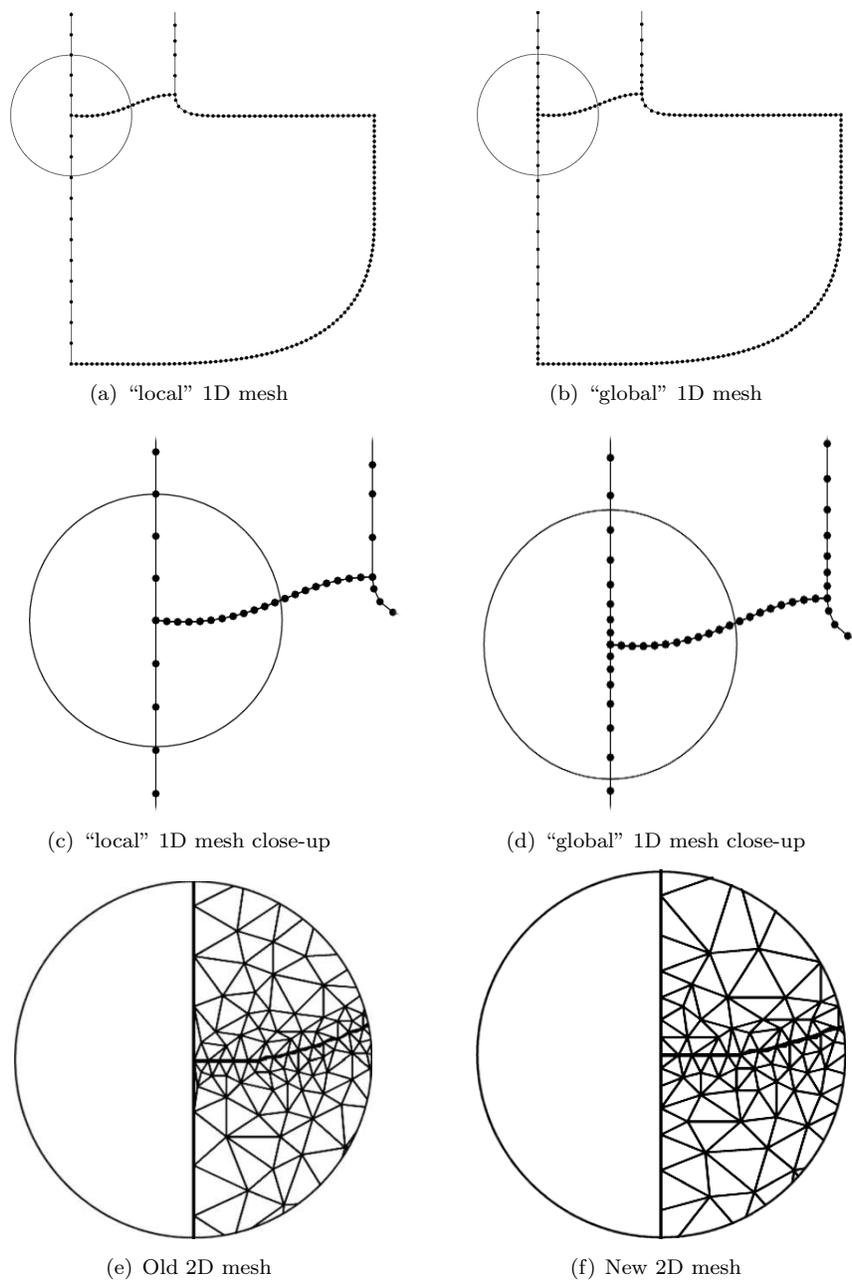


Figure 2.18: Comparison of "local" and "global" 1D mesh and correspondent 2D mesh

where $k^* = \sqrt{1/k''}$, gives the curvature providing the average density $(k + \hat{k})/2$.

However, the density function defined in equation (2.8) is a “local” function which does not take the adjacent multi-curves into account. When there are very large curvature or mesh density changes from one multi-curve to the adjacent multi-curves, very bad 1D meshes with sharp mesh gradient changes will be generated, and hence very skinny triangles will be constructed during the 2D mesh generation and refinement process, see Figure 2.18(a) and Figure 2.18(e). To solve this problem, a new “global” density function has been introduced as follows:

$$\sigma' = \frac{k + \frac{\hat{k}\rho^2}{k_*^2} + \frac{1}{k_*^2} \sum_{i=1}^2 \left(\frac{\sigma_{max}^i}{d_i^2} \right)}{1 + \frac{\rho^2}{k_*^2} + \frac{1}{k_*^2} \sum_{i=1}^2 \left(\frac{1}{d_i^2} \right)} \quad (2.12)$$

where $\sigma_{max}^1 = \max(\sigma_1^1, \sigma_2^1, \dots, \sigma_p^1)$, $\sigma_{max}^2 = \max(\sigma_1^2, \sigma_2^2, \dots, \sigma_q^2)$, representing the maximum densities at the two extremities, and where p and q are the two multi-curve endpoint degrees of freedom and d_1 and d_2 are the distances to these two endpoints respectively. The 1D mesh generated by the new “global” density function and the corresponding 2D mesh are illustrated in Figure 2.18(b) and Figure 2.18(f). Comparing the 1D and 2D mesh generated by means of the “local” density function with the meshes generated by means of the “global” density function, the quality of the meshes generated with our new method is significantly improved.

2.5 Mesh deformation for quasi-steady and time-dependent simulations

During the crystal growth process, the geometry of the melt and crystal are continuously moving and deforming. For example, the crystal solid is very small during seeding and subsequently becomes larger and larger, while the melt volume decreases and often takes a particular shape during tail-end stage. Various moving and deforming interfaces delimiting the melt/crystal and the melt/gas (meniscus) boundary must hence be computed, and the technique to deform the melt and crystal meshes for dynamic simulations has to be developed. Notice that even in quasi-steady simulations, the interface between the melt and crystal is continuously deforming during the global iterations, although the assembled melt and crystal geometry as a whole is fixed.

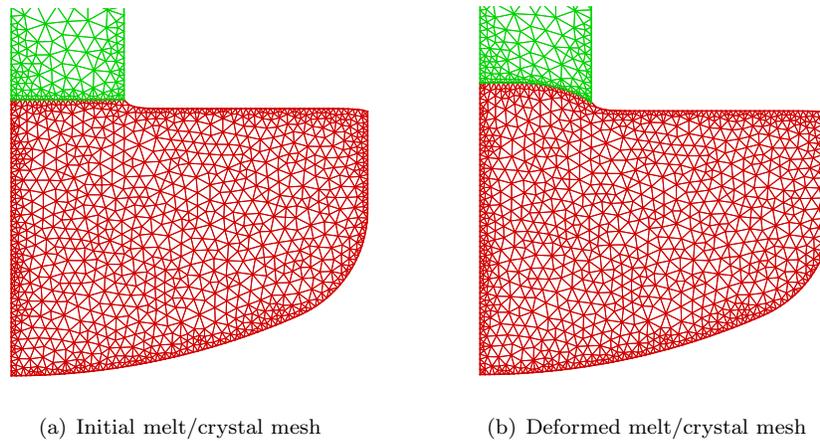


Figure 2.19: Mesh deformation due to solidification front change

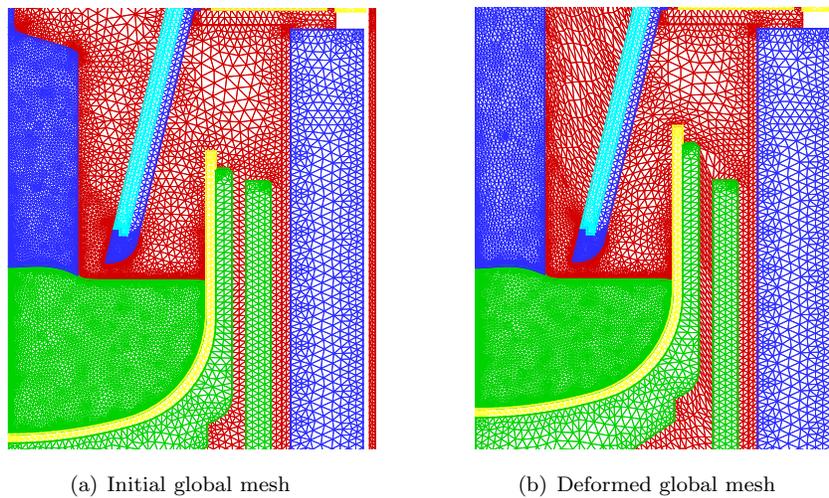


Figure 2.20: Global mesh deformation for time-dependent simulation

The mesh deformation technique we employ here is based on the method proposed by Dupret et al [(Berghezan and Dupret 1994)]. In this method, the boundary and interface nodes are fixed after displacement, while the internal nodes are moved to a more appropriate place at each iteration until the mesh is acceptable. This result can be obtained by exerting on any internal node

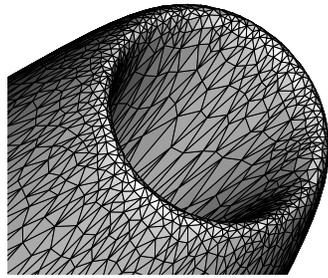
an appropriate pseudo-force, which tends to correct ill-behaved elements and which vanishes for a satisfactory solution. Figure (2.19) shows the deformed melt/crystal mesh in a quasi-steady simulation. This quasi-steady simulation begun with an initial flat melt/crystal interface, then this interface was deformed to the appropriate place due to the isothermal condition of the solidification front when taking the global heat transfer and/or melt convection into account. Figure (2.20) shows the global deformed mesh due to the lengthening of crystal and lift of crucible. It should be noted that for time-dependent simulations, re-meshing is needed if geometrical changes become too important when switching to a subsequent growth stage. This can be easily performed by our unstructured mesh generation algorithm.

2.6 Extend to surface triangulation

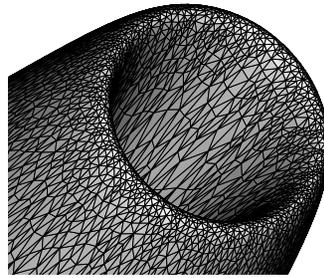
Currently, single crystal growth simulation are transiting from two-dimension axisymmetric models to three-dimensional models, although we believe that this transition will take quite a long time because of the limits of nowadays computing power and the slow progress in algorithm development for large sets of algebraic equations. Therefore fully three-dimensional time-dependent simulations for industrial applications will not be available in the near future. The first big difficulty arising in 3D simulations is the requirement of automatic unstructured mesh generation for the whole crystal growth system, particularly for the melt/crystal “macro-elements” which exhibit a continuously deforming geometry.

One basic step of 3D mesh generation is to generate initial meshes on the boundary surfaces, as typically represented by NURBS or analytical functions. Surface mesh generation algorithms can be classified as either belonging to parametric space methods or to direct 3D methods. The former approach first generates elements in the two-dimensional parametric space and then maps the $u - v$ coordinates back to $x - y - z$ space. On the other hand, direct 3D methods generate elements directly on the geometry surface without regard to the parametric representation of the underlying geometry [(Owen 1998)]. Here, we try to extend our algorithm to surface triangulation by the parametric space method. We first generate and refine a 2D mesh in the parametric space by the minimum angle, maximum area-scale criteria and the newly introduced surface approximation tolerance criterion. When the refinement process has converged, the final surface mesh can be obtained by mapping the 2D mesh onto the surface space. However, a mapping with bad properties can easily spoil the final triangulation. Therefore, the key point of this method is to find a “good”

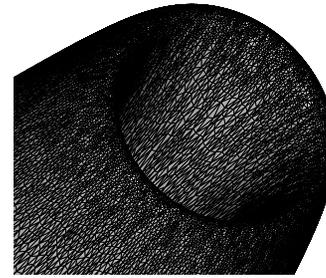
mapping from 2D parametric space to 3D surface space while keeping the correctness of mesh topology and quality. This is another hot topic in parametric surface visualization and surface meshing research community. Fortunately, nearly all important geometries of the crystal growth system, particularly the crucible, the melt and the crystal, etc., are perfectly axisymmetric in order to grow cylindrical crystals, and therefore it is not so difficult to find a “good” mapping between 2D parametric space and 3D surface space. Several surface triangulations and the corresponding meshes in the 2D parametric space are presented in Figure 2.21 with different minimum angle, maximum area-scale and surface approximation tolerance criteria.



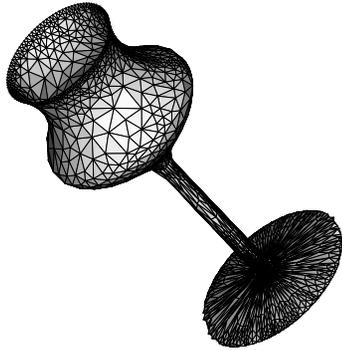
(a) $\alpha = 20$, $scale = 10^4$, $\epsilon = 10^{-3}$



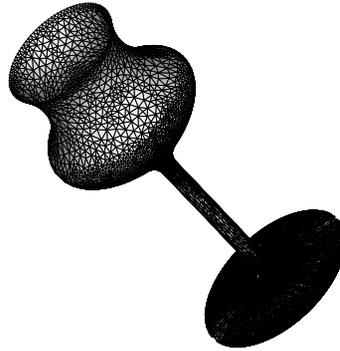
(b) $\alpha = 20$, $scale = 10^4$, $\epsilon = 2.0e^{-4}$



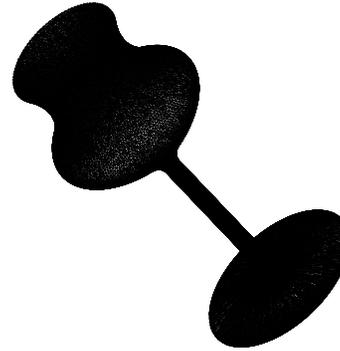
(c) $\alpha = 20$, $scale = 10^4$, $\epsilon = e^{-4}$



(d) $\alpha = 20$, $scale = 5.0e^3$, $\epsilon = 10^{-2}$



(e) $\alpha = 20$, $scale = 5.0e^3$, $\epsilon = 2.0e^{-3}$



(f) $\alpha = 20$, $scale = 5.0e^3$, $\epsilon = 4.0e^{-4}$

Figure 2.21: Surface triangulation examples

2.7 Conclusions

This chapter aimed at addressing the unstructured mesh generation for single crystal growth processes. We first introduced a general strategy of unstructured mesh generation for quasi-steady and dynamic simulations of bulk crystal growth processes. Then we focused on 2D shape-quality unstructured mesh generation by a modified incremental Delaunay refinement algorithm and a grade-adaptive 1D mesh generation algorithm on fixed geometries, followed by the introduction of 1D and 2D mesh deformation techniques for quasi-steady and dynamic simulations. Finally, we discussed the possibility to extend our algorithm and implementation to surface/3D unstructured mesh generation and closed this chapter by presenting some applications to single crystal growth processes.

The first step of our 2D shape-quality unstructured mesh generation algorithm is to generate an initial triangulation for a given 1D boundary mesh. This initial triangulation is also called the simple polygon triangulation, which is a classic problem in computational geometry. We have reviewed three simple popular polygon triangulation algorithms and summarized their advantages and disadvantages for real applications, then we have focused on the algorithm improvements and implementation by the sweep line method. Comparisons between our implementation and other popular polygon triangulation codes available from the web were also presented. Our practical experience and numerical experiments have showed that our implementation is very competitive and prove to be faster than other popular codes.

The next step of our algorithm is to transform the initial triangulation into an initial Delaunay triangulation by flipping operations. Finally the triangulation is refined by a modified incremental refinement algorithm. Three criteria were used to control the mesh shape/size quality and mesh density. Unlike the implementation by Ruppert, splitting the encroached boundary line segments is ignored during the different refinement stages in order to keep boundary consistency between the different “macro- elements”. When a Steiner point is inserted into the triangulation, one key issue is to find where this point is located. To solve this so-called point location problem, we have proposed and implemented the first linear point location algorithm for incremental Delaunay refinement without using any additional preprocessing, storage space and complicated data structure, while providing an easy to implement algorithm at the same time. After one Steiner point is inserted and the target triangle is found, a so-called “split-and-flip” algorithm to construct an updated Delaunay triangulation has been proposed and implemented.

Since the 1D boundary mesh plays a prominent role in our Delaunay refinement algorithm, we have proposed and implemented a “global” grade-adaptive 1D mesh generator taking the density, curvature, curvature impact factor and adjacent curves into account and hence the quality of the meshes generated by our new method was significantly improved.

In order to handle deformed geometries both for quasi-steady and time-dependent simulations, appropriate techniques to deform and optimize the mesh were summarized at the end of this chapter and the possibility to extend our algorithm to surface and 3D unstructured mesh generation was also discussed.

Chapter 3

Modeling of Melt Convection

3.1 Melt convection in Czochralski growth

It is well known that convection of the molten semi-conductor strongly affects mass, heat and momentum mixing in several growth systems, thereby affecting the temperature gradient in the melt, the melt/crystal interface shape, the crystal growth rate, and the chemical composition and defect density of the crystal. Hence, understanding the melt flow pattern and characteristics can help the crystal grower to control, improve and optimize the crystal growth process. However, the molten semi-conductor flow in a Czochralski crucible is dauntingly complicated [(Hurle 1993)] due to a complex combination of natural and forced convection, and the main characteristics of the melt flow in the Czochralski crystal growth process are still not well understood.

Natural convection results from the buoyancy and capillary forces acting in the melt and on its interface. The origin of these two kinds of forces is due to the melt density and surface tension dependence upon temperature and solute concentration. In all growth processes, a temperature gradient is necessary in order to grow the crystal from the melt. Therefore, natural thermal convection is always present due to this temperature gradient between the hot crucible wall and the relatively cold solidification interface. If the temperature gradient is sufficiently low, a flow pattern with steady, laminar and axisymmetric characteristics can be observed, but when the temperature gradient is larger than a critical value, the flow becomes unstable and time-dependent, or even turbulent and asymmetric (see Figure 3.1). Natural solutal convection is due to the solute concentration gradients, and hence the fluid density gradients, generated by segregation along the solidification interface and across the melt surface.

The thermosolutal dependence of melt/gas surface tension induces fluid flow along this interface from the low surface tension area (usually located in the hot part of the interface, close to the crucible wall) to the high surface tension region (usually located in the cold part of the interface, close to the tri-junction). The

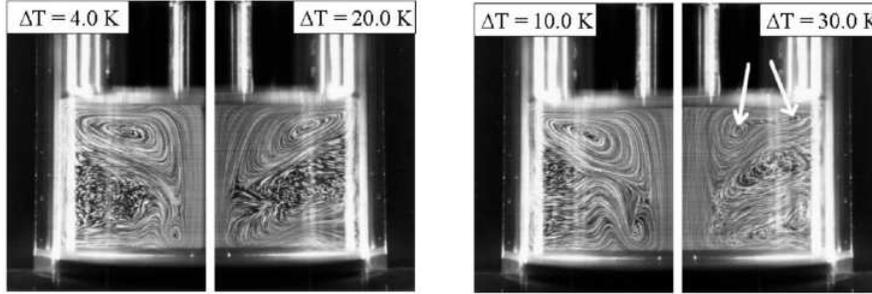


Figure 3.1: Experiments of temperature gradient influence on liquid silicon oil convection without rotation of the dummy crystal: with a sufficiently high temperature difference, a multiroll and “blurred” structure is observed and the “blurred” effect is due to the unsteady flow features [(Hintz et al. 2001)].

surface tension gradient due to concentration variations along the liquid/gas interface can also drive the flow. In general, the surface tension induced flow, also called Marangoni convection, is generated at the free surface and usually has the same direction as the buoyancy driven flow. Marangoni convection increases the flow velocity at the melt surface, and hence, in Czochralski silicon growth, influences the rate of evaporation of oxygen from the free surface significantly.

“Forced” convection denotes the flow induced by crystal and crucible rotations in order to grow axisymmetric cylindrical crystals. The rotation of the crucible generates the so-called Coriolis force, which reduces the vertical convection due to angular momentum conservation, hence stabilizing the fluid flow [(Kishida and Okazawa 1999)]. However, the Coriolis force with increasing rotation rate will complicate the convection structure and increase the curvature of the fluid particle paths, resulting in a special fluid motion called baroclinic instability. Further increasing the crucible rotation rate, a transition from baroclinically stable flow to unstable, geostrophic turbulence will occur at high Taylor numbers (as defined in Table 3.2). On the other hand, above a critical crystal rotation rate, a typical wave/vortex pattern can be observed at the melt free surface, see Figure (3.2). This pattern is a flow instability induced by the interaction between the flow outgoing from the center of the melt domain due to crystal rotation, and the incoming buoyancy driven flow [(Assaker 1998)]. The “forced” convection induced by crystal and/or crucible rotations has great influence on the solidification interface shape and the concentration of impuri-

ties in the crystal, and therefore is often used as a key parameter to establish the desired interface shape, and a homogeneous distribution of impurities as well.

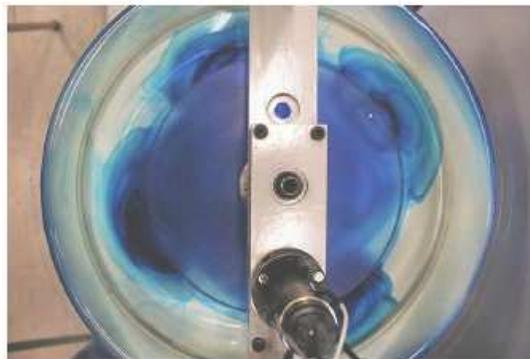


Figure 3.2: Wave/Vortex pattern in a model experiment simulating the behaviour of a Czochralski silicon puller [(Delsaute et al. 2006, Bamberg, Germany)].

The shear stress exerted on the melt/gas interface or meniscus due to gas convection is another kind of force driving the flow, which also plays an important role, therefore influencing oxygen transport and rate of evaporation. The shear stress due to gas convection usually has an opposite direction to the buoyancy and capillary driven forces. Numerical simulations of Kalaev et al [(Kalaev et al. 2003)] show that, at low crucible and/or crystal rotation rates, the gas flow is insufficient to generate a separate flow cell in the melt and only slightly damps out the melt flow driven by buoyancy and Marangoni forces. However, at high crucible and/or crystal rotation rates, the melt flow becomes much more sensitive to shear stress and a separate clockwise cell probably appears beneath the melt/gas interface.

In addition, in modern Czochralski crystal growth systems, magnetic fields are widely used to damp out the melt flow and temperature fluctuations. In this case, the melt flow experiences the Lorentz electromagnetic force induced by the interaction between electrical current and magnetic field. Accordingly the flow and temperature oscillations are strongly reduced.

The combination of all the above driving forces leads to a complex three-dimensional and time-dependent flow structure. Ample evidence shows that the melt flow driven by the above mentioned forces is oscillating and most often chaotic and turbulent [(Lipchin and Brown 1999)], even for those semi-

conductor growth systems with relatively small-scale crucibles [(Kim and Langlois 1991)]. Such complex flow patterns can only be predicted by using equally complex models and numerical methods.

3.2 Modeling of melt convection in Czochralski silicon growth

The modeling of melt convection during Czochralski silicon crystal growth has represented a major objective in Czochralski growth modeling [(Dupret and Van den Bogaert 1994)]. In fact it was one of the most active research topics in the last three decades, and probably will continue to so in the next decades.

In the early 1980s, the efforts to understand the mechanisms of melt convection were based mainly on using finite difference methods [(Stewart and Weinberg 1972)][(Langlois 1977)][(Langlois 1982)][(Langlois 1983)] [(Kobayashi 1980)]. Reviews of these efforts can be found in [(Pimpputkar and Ostrach 1981)][(Polezhaev 1984)][(Langlois 1985)]. Afterwards, the finite element method was introduced by various authors [(Ettouney and Brown 1983)][(Crochet et al. 1983b)] [(Chang and Brown 1984)][(Derby and Brown 1987)][(Sackinger et al. 1989)], in order to calculate highly non-linear and coupled melt convection, and to solve melt/crystal interface problems in Czochralski growth processes. By the end of the 1980s, the modeling of crystal growth melt convection for two-dimensional stationary problems, together with significant progress in global heat and mass transport modeling and simulation [(Wouters 1985)][(Dupret et al. 1986a)][(Dupret et al. 1986b)][(Atherton et al. 1987)][(Nicodeme et al. 1988)][(Sackinger et al. 1989)] [(Derby et al. 1989)][(Ryckmans 1989)][(Dupret et al. 1990)] [(Bornside et al. 1990)] in moderately complicated geometries had reached a more accurate level. However, due to extremely large computational costs and limited computer resources a few decades ago, most studies on the melt convection during this period were based on the use of a laminar model with simplified or moderately complicated silicon melt geometries.

Tremendous progress in the computer modeling of bulk crystal growth had been achieved at the end of the 1980s and in the 1990s. On the one hand, the time-dependent global models developed by [(Atherton et al. 1987)] (using an explicit integration technique), and by [(Van den Bogaert 1993)][(Van den Bogaert and Dupret 1997a)][(Van den Bogaert and Dupret 1997b)] (using an implicit technique) were initially introduced to perform crystal growth simulations. On the other hand, more complicated numerical techniques and turbulence models to predict the flow behavior and heat transfer in the silicon

melt were developed, and coupled with the global quasi-steady or even time-dependent heat transfer. For example, a global time-dependent heat transfer model firstly taking melt convection with the mixing-length or $k-l$ turbulence models to predict melt convection in any complicated two-dimensional geometry was developed by [(Assaker et al. 1997)][(Assaker 1998)], and great success was witnessed when applying these models to industrial Czochralski silicon growth processes. Furthermore, the standard turbulence $k-\epsilon$ model, the low-Reynold $k-\epsilon$ turbulence model of Chien [(Chien 1982)] and Jones and Launder [(Jones and Launder 1972)] have been widely used by other crystal growth research groups, such as in the work of [(Kinney and Brown 1993)][(Chung et al. 1996)][(Zhang et al. 1996)][(Kalaev et al. 2002)], etc.. A special study was conducted by [(Lipchin and Brown 1999)] to understand which turbulence $k-\epsilon$ model is the best for melt convection simulation. Their numerical experiments were performed on a simplified silicon melt geometry (with a cylindrical crucible and flat meniscus) with uniform thermal boundary conditions and using the finite volume method, and they concluded that the low-Reynolds model of Jones and Launder is preferable with respect to the prediction of bulk flow and impurity transport through the melt. Later on, they used this turbulence model to calculate the global heat and mass transfer in a Czochralski system using a hybrid finite volume - finite element method [(Lipchin and Brown 2000)]. Recently, the employment of hybrid LES (Large-eddy simulation) and Reynolds averaged approach for two-dimensional and/or three-dimensional melt convection was presented by [(Evstratov et al. 2001)][(Kalaev et al. 2002)][(Evstratov et al. 2002)].

The mid-1990s was a period of great optimism in the hope that accurate three-dimensional simulations, including the prediction of global time-dependent phenomena, would soon become routine [(Yeckel and Derby 2005)]. However, in the past few years there appears to have been a slowdown of the developments in crystal growth numerical modeling. This slowdown is mainly due to the great difficulties and extremely expensive computational costs experienced when solving the three-dimensional turbulent melt convection problem arising in crystal growth processes, particularly for time-dependent growth processes. For such kind of problems, an iterative solver with parallelized implementation is nearly mandatory, and practical three-dimensional applications maybe will become routine and accurate enough in the next decade only if robust and efficient iterative preconditioning techniques develop at current or more rapid pace. Therefore, the transition from two-dimensional axisymmetric calculations to fully three-dimensional industrial applications maybe will take another decade, and crystal growth numerical modeling with the axisymmet-

ric assumption still represents the most economic way to predict, control and optimize the industrial single crystal growth processes.

3.2.1 Governing equations

Based on three basic assumptions: ❶ the flow of the silicon melt is incompressible, ❷ molten silicon is a Newtonian fluid, and ❸ the effect of melt density variations is negligible except in the buoyant force (Boussinesq approximation), the motion of the silicon melt in the crucible is governed by the momentum and continuity equations, and the temperature field of the melt is governed by the energy equation. The momentum equation, which describes the fluid particle acceleration DU/Dt due to the surface stresses and body forces experienced by the fluid, writes as follows:

$$\rho \frac{D\mathbf{U}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} \quad (3.1)$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla \quad (3.2)$$

The molecular-originated stresses are described by the stress tensor $\underline{\underline{\sigma}}$, which is symmetric, that is $\sigma_{ij} = \sigma_{ji}$. For constant density Newtonian fluids, the stress tensor is given by:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu(T)\mathbf{D} \quad (3.3)$$

where

$$\mathbf{D} = \frac{1}{2}(\nabla\mathbf{U} + \nabla\mathbf{U}^T) \quad (3.4)$$

is the symmetric, deviatoric rate-of-strain tensor, p is the pressure, \mathbf{I} is the identity tensor and $\mu(T)$ the dynamic shear viscosity as a function of fluid temperature T .

The body forces can be gravity, electromagnetic Lorentz force or any other force per unit volume. For industrial Czochralski crystal growth systems, the gravity force is normally present and can be described by:

$$\mathbf{f} = \rho(T)\mathbf{g}, \quad (3.5)$$

where \mathbf{g} is the gravity acceleration vector. According to the Boussinesq approximation, the effect of melt density variations in the buoyant force term is not negligible. Assuming a linear temperature dependence, $\rho(T)$ can be written as:

$$\rho(T) = \rho_0 \left(1 - \beta(T - T_0) \right), \quad (3.6)$$

where

$$\beta = -\frac{1}{\rho_0} \left(\frac{\partial \rho}{\partial T} \right), \quad (3.7)$$

is the (assumed constant) thermal expansion coefficient, T the temperature field and ρ_0 the fluid density at the reference temperature T_0 .

Various kinds of magnetic fields are widely employed to damp out the melt convection in modern large-size Czochralski crystal growth systems. When a magnetic field is applied, the Lorentz force can be expressed as:

$$\mathbf{f} = \mathbf{J} \times \mathbf{B} \quad (3.8)$$

where \mathbf{J} is the electric current density, and \mathbf{B} the magnetic induction. Introducing equations (3.4) to (3.8) into (3.1), the final form of the momentum equation is obtained:

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right) = -\nabla p + \nabla \cdot \left(\mu(T) (\nabla \mathbf{U} + \nabla \mathbf{U}^T) \right) + \rho \mathbf{g} (1 - \beta(T - T_0)) + \mathbf{J} \times \mathbf{B} \quad (3.9)$$

Assuming that the melt flow is quasi-steady, and defining the modified pressure P as:

$$P = p + \rho g z \quad (3.10)$$

where z is the vertical coordinate, the momentum equation for quasi-steady melt convection can finally be rewritten as follows:

$$\rho (\mathbf{U} \cdot \nabla \mathbf{U}) = -\nabla P + \nabla \cdot \left(\mu(T) (\nabla \mathbf{U} + \nabla \mathbf{U}^T) \right) + \rho \beta \mathbf{g} (T - T_0) + \mathbf{J} \times \mathbf{B} \quad (3.11)$$

Here, the gravity force term exists implicitly in the momentum equation. Therefore the gravity force experienced by the fluid has the same effect as the isotropic pressure p and this conservative force has no effect on the velocity. It should also be noted that, in equations (3.9) and (3.11), the reference density ρ_0 has been replaced by ρ for simplicity reasons.

The continuity equation, which ensures mass conservation, is given by:

$$\frac{D\rho}{Dt} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.12)$$

For incompressible fluids, equation (3.12) can be simplified as follows:

$$\nabla \cdot \mathbf{U} = 0 \quad (3.13)$$

The temperature field T is governed by the energy equation:

$$\rho c(T) \frac{DT}{Dt} = -\nabla \cdot \mathbf{q} + w \quad (3.14)$$

where w is the volumetric heat source and q the heat flux given by Fourier's law:

$$q = -k(T)\nabla T \quad (3.15)$$

Plugging equations (3.4) to (3.15) into equation (3.14) and assuming that there is no heat source in the melt and crystal of a Czochralski furnace ($w = 0$), we obtain the energy equation as follows:

$$\rho c(T) \left(\frac{\partial T}{\partial t} + \mathbf{U} \cdot \nabla T \right) = \nabla \cdot (k(T)\nabla T) \quad (3.16)$$

where ρ , $c(T)$ and $k(T)$ are the density, specific heat and thermal conductivity of the melt or crystal, respectively. Notice that both the specific heat and thermal conductivity can be temperature dependent.

3.2.2 Boundary conditions

The boundary of the melt domain Γ consists of two complementary subsets Γ_{du} and Γ_{nu} , on which given Dirichlet-type (with imposed velocity vector $\bar{\mathbf{U}}$) and Neumann-type (with imposed stress vector $\bar{\lambda}_n$) boundary conditions apply. More specifically, along all solid surfaces including the melt/crystal solidification front Γ_s and the crucible bottom wall Γ_c , the melt should stick to the corresponding solid boundary and no-slip boundary conditions (Dirichlet-type) are thus applied:

$$U_i = \bar{U}_i \quad \text{on} \quad \Gamma_s \cup \Gamma_c \quad (3.17)$$

Along the melt/gas interface (the so-called meniscus), capillary normal stress and zero normal velocity boundary conditions are imposed:

$$\mathbf{U} \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma_m \quad (3.18)$$

and

$$\sigma_n = \chi\sigma - P_a \quad \text{on} \quad \Gamma_m \quad (3.19)$$

where σ and χ denote the meniscus surface tension and curvature respectively, while P_a stands for the gas pressure. The condition (3.19) is used to determine the meniscus shape. Concerning the tangential stress, it is imposed to be zero as well if the Marangoni effect is negligible. However, if Marangoni effect is taken into account, the tangential stress along the meniscus will not be zero. This tangential force is induced by the gradient of surface tension σ from low σ to high σ along the meniscus. The value of σ is sensitive to temperature and solute concentration. Neglecting this latter effect and assuming a linear surface tension thermal behavior, σ can be described as:

$$\sigma = \sigma_0 + \gamma_T(T - T_0) \quad (3.20)$$

where σ_0 is the surface tension at the reference temperature T_0 , T the actual temperature and $\gamma_T = \frac{\partial\sigma}{\partial T}$ the surface tension coefficient, which is negative for liquid silicon where the surface tension decreases with increasing temperature. The meridional thermocapillary force exerted along the meniscus is then given as follows:

$$f_\sigma = \frac{\partial\sigma}{\partial s} = \gamma_T \frac{\partial T}{\partial s} \quad (3.21)$$

where s stands for the curvilinear abscissa along the meniscus.

Note that the melt convection below the meniscus has a negligible effect on its shape due to the low capillary and Weber numbers characteristic of silicon melt flow [(Brown 1998)]. Therefore the meniscus shape calculation can be decoupled from the equations governing melt convection.

3.3 Numerical modeling

3.3.1 Discretization method and formulation

To obtain effective numerical simulation, the problems governed by partial differential equations (PDEs) have to be converted to a set of algebraic equations which can be solved by direct or iterative solvers. The most common techniques to discretize PDEs are the finite-volume method (FVM) and the finite-element method (FEM), although the boundary-element method (BEM), the finite-difference method (FDM) and the spectral method are also widely used.

In FDM method, the discretization is based on the differential form of the PDEs to be solved and all the derivatives are approximated by finite difference formulas. The computational domain is usually subdivided into structured quadrangular or hexahedral cells and the resulting sparse system of equations can be efficiently solved by direct or iterative solvers. Comparatively, in FVM method, the discretization is based on the integral form of the PDEs to be solved. The computational domain can be discretized into structured or unstructured finite volumes and the resulting solution typically consists of variables placed at finite volume centroids (rather than at nodal points with FDM or FEM methods). On the other hand, the FEM discretization is based on a piecewise representation of the solution in terms of specified, usually polynomial, basis functions. The computational domain is subdivided into smaller domains (the finite elements) and the solution in each element is constructed locally from the basis (or trial) functions while the equation are made discrete by mean of appropriate test functions. Finally, in the spectral method, a complete family (or basis) of global smooth functions is employed to minimize the

weighted residuals of the equations. It is difficult to compare these discretization methods. Generally speaking, for the same problem with the same order of discretization, the spectral method provides the most accurate results in the absence of singularities, closely followed by the other methods provided strongly refined meshes are used [(Dupret and Van den Bogaert 1994)].

The FVM method has been the most widely used numerical method in the CFD community because of its low computational cost compared with the FEM method while keeping enough flexibility for complex geometries. However, since all the equations governing the crystal growth process are coupled, nonlinear and extremely complicated, the Galerkin FEM method is particularly suitable for such kinds of nonlinear problems on complex deforming geometrical domains which mix equations of different types. Therefore, the Galerkin FEM discretization method has been traditionally chosen in the FEMAG software to discretize all the governing equations, and hence it has been employed throughout our work.

Let \mathcal{T}_h be a partition of the domain Ω into finite elements (triangles, quadrangles, etc.), while the elements are denoted by Ω_e with boundary Γ_e . Letting $H^1(\Omega)$ be the Hilbert space of square-integrable functions with square-integrable first-order derivatives, then the finite dimensional subspace $H^{1h} \subset H^1(\Omega)$ is defined as the space spanned by piecewise, usually polynomial, C^0 continuous basis functions ϕ^h over the discretization \mathcal{T}_h :

$$H^{1h} = \left\{ \phi^h \mid \phi^h \in C^0(\Omega), \phi^h|_{\Omega_e} \in P^k, \forall \Omega_e \in \mathcal{T}_h \right\} \quad (3.22)$$

where P^k represents the set of k^{th} -order polynomials. A set of functions belonging to H^{1h} then can be written as a linear combination of nodal basis functions:

$$\phi^h(x, y) = \sum_j \phi_j \psi_j(x, y) \quad (3.23)$$

where the coefficients ϕ_j are the so called degrees of freedom (or nodal values) of ϕ^h , and $\psi_j(x, y)$ is the j^{th} basis function (associated with node j). Note that $\psi_j(x_i, y_i) = \delta_{ij}$ for any two nodes i and j of the grid, where δ_{ij} is the Kronecker delta. Letting $\mathbf{u}^h = (u^h, v^h, w^h)$ denote the trial solution for velocity and p^h the trial solution for pressure, then the velocity and pressure fields are

approximated as follows on each element:

$$u^h(r, z) = \sum_j^{n_v} u_j \psi_j^v(r, z) \quad (3.24)$$

$$v^h(r, z) = \sum_j^{n_v} v_j \psi_j^v(r, z) \quad (3.25)$$

$$w^h(r, z) = \sum_j^{n_v} w_j \psi_j^v(r, z) \quad (3.26)$$

$$p^h(r, z) = \sum_j^{n_p} p_j \psi_j^p(r, z) \quad (3.27)$$

where n_v and n_p are the numbers of velocity and pressure nodes on each element, while ψ_j^v and ψ_j^p stand for the velocity and pressure basis functions respectively. In order to obtain an algebraic equation for each unknown, a so-called weak weighted residual formulation is applied. Letting \mathbf{v}^h and q^h denote test functions for the momentum equation (3.11) and the continuity equation (3.13), respectively, the classical Galerkin formulation can be written as follows:

$$\int_{\Omega} \left[\rho(\mathbf{U} \cdot \nabla \mathbf{U}) - \nabla \cdot (\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T) - p\mathbf{I}) - \rho\beta\mathbf{g}(T - T_0) \right] \cdot \mathbf{v}^h d\Omega = 0 \quad (3.28)$$

and

$$\int_{\Omega} [(\nabla \cdot \mathbf{U})q^h] d\Omega = 0 \quad (3.29)$$

If the test-functions are chosen to be the shape functions, which is typical of the Galerkin FEM method, then plugging the test-function set:

$$\mathbf{v}^h = \begin{cases} (\psi_j, 0, 0) & j = 1, \dots, n_v, \\ (0, \psi_j, 0) & j = 1, \dots, n_v, \\ (0, 0, \psi_j) & j = 1, \dots, n_v, \end{cases} \quad (3.30)$$

and

$$q^h = \varphi_j, \quad j = 1, \dots, n_p, \quad (3.31)$$

into equation (3.28) and equation (3.29), for each element we obtain appropriate contributions to a set of $3 * n_v + n_p$ nonlinear algebraic equations governing the nodal velocity and pressure fields. After assembly with contributions from other elements, these equations can be written as:

$$\begin{cases} F_u(u, v, w, p) = 0 \\ F_v(u, v, w, p) = 0 \\ F_w(u, v, w, p) = 0 \\ F_p(u, v, w, p) = 0 \end{cases} \quad (3.32)$$

It should be noted that, in order to respect the so-called LBB condition [(Babuška 1971)][(Babuška 1973)] [(Brezzi 1974)] the degree of the velocity field basis functions was chosen as 2 with C^0 continuity, while for the pressure field this degree is chosen as 1 with the same continuity as for the velocity field.

3.3.2 Numerical method

The resulting set of discretized algebraic equations (3.32) is highly nonlinear. This nonlinearity comes from the nonlinear advection terms and the viscous terms if turbulent viscosity is taken into account. Due to this nonlinearity, the Newton-Raphson iterative method is chosen to linearize and solve these algebraic equations. The choice of this method has been mainly motivated by its quadratic convergence rate when the current guess is close enough to the actual solution [(Assaker 1998)]. At each Newton-Raphson iteration, a sparse linear system of the following form has to be solved:

$$\begin{bmatrix} \frac{\partial F_u^i}{\partial u_j} & \frac{\partial F_u^i}{\partial v_j} & \frac{\partial F_u^i}{\partial w_j} & \frac{\partial F_u^i}{\partial p_j} \\ \frac{\partial F_v^i}{\partial u_j} & \frac{\partial F_v^i}{\partial v_j} & \frac{\partial F_v^i}{\partial w_j} & \frac{\partial F_v^i}{\partial p_j} \\ \frac{\partial F_w^i}{\partial u_j} & \frac{\partial F_w^i}{\partial v_j} & \frac{\partial F_w^i}{\partial w_j} & \frac{\partial F_w^i}{\partial p_j} \\ \frac{\partial F_p^i}{\partial u_j} & \frac{\partial F_p^i}{\partial v_j} & \frac{\partial F_p^i}{\partial w_j} & \frac{\partial F_p^i}{\partial p_j} \end{bmatrix} \begin{bmatrix} \delta u_j \\ \delta v_j \\ \delta w_j \\ \delta p_j \end{bmatrix} = - \begin{bmatrix} F_u^i \\ F_v^i \\ F_w^i \\ F_p^i \end{bmatrix} \quad (3.33)$$

The above linear system of equations can be solved by means of a direct solver or an iterative solver. Historically, direct methods based on variants of the Gaussian elimination (such as, the frontal method), are proved to be robust and efficient for two-dimensional problems. However, for large scale applications, using a direct solver is not the optimum choice because of large memory requirements and low performance, especially for three-dimensional problems, and also because this method is very difficult to parallelize. Therefore, for large-scale or three-dimensional problems, iterative methods are almost mandatory. Let us here note that iterative methods have experienced striking progress in the past decades and gained popularity in many areas of scientific computing because of the emergence of **1** preconditioned conjugate gradient-like methods or **2** multigrid methods for solving linear systems. In practice, it was found that the combination of pre-conditioning and Krylov subspace iterations could provide efficient and simple “general purpose” procedures that could compete

with direct solvers and approach their quality [(Saad 2000)]. However, when iterative solvers are used, particular attention has to be paid to transport modeling (such as in crystal growth) since the significant nonlinearity resulting from the convection terms, might greatly slow down or sometimes even completely halt the convergence to a solution [(Yeckel and Derby 2005)].

3.4 Numerical benchmark testing on 2D cavity problem

In order to validate our Navier-Stokes solver based on unstructured meshes (structured quadrilateral meshes were used in the previous FEMAG-1 software generation), benchmark testings on the 2D lid-driven cavity problem are performed at different Reynolds numbers. Comparisons of our solutions with other highly accurate results will be presented thereafter.

The 2D lid-driven cavity flow is probably one of the most studied fluid problem in the field of computational fluid dynamics due to simple geometry and associated boundary conditions. At the same time, the cavity flow retains a rich fluid physics manifested by multiple counter-rotating recirculating regions near the cavity corner depending on the Reynolds number. There exists a variety of experimental and numerical results for this classic problem, which hence can be used as a benchmark test for our unstructured steady incompressible Navier-Stokes solver.

The problem configuration with a square domain $\Omega = [0, 1] \times [0, 1]$ and the corresponding anisotropic unstructured mesh with 10,372 elements are illustrated in Figure (3.3). The associated boundary conditions for this problem are: along the top boundary $(0, 1) \rightarrow (1, 1)$, the velocity vector is imposed as $(1, 0)$, whereas the velocity vector along any other boundary is imposed as $(0, 0)$. Therefore the boundary conditions are discontinuous at the top left and right corners, which represent singularities of the solution. The Reynolds number of this problem is given by:

$$Re = \frac{\rho}{\mu} \quad (3.34)$$

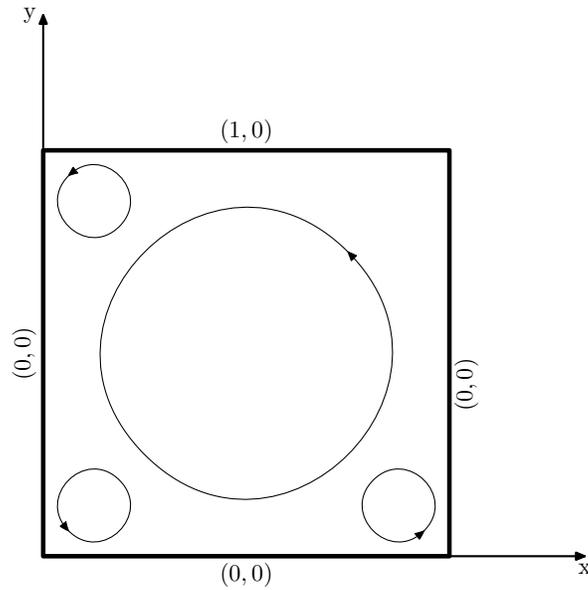
where, ρ is the fluid density and can be imposed as 1.0 for simplicity, and μ denotes the dynamic viscosity.

In all our numerical experiments, computations are performed for the flow regimes with $Re = 1, 100, 400, 1,000, 5,000, 10,000, 12,500, 15,000, 20,000$ and 25,000 using the same mesh in Figure (3.3), while the corresponding streamline contours can be found in Figure (3.4) and Figure (3.5). It should

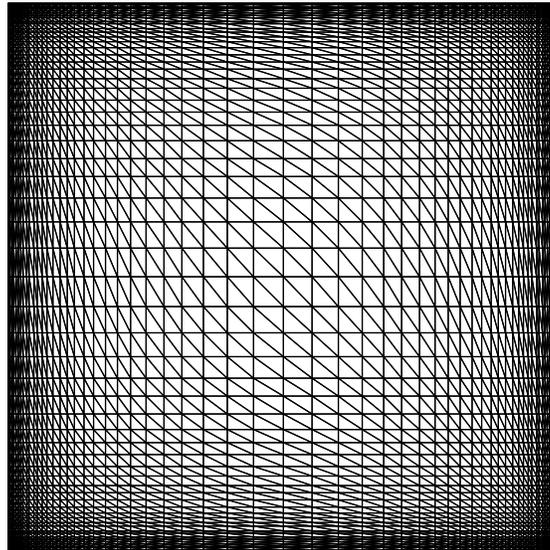
Re	present work	Ghia et al	Erturk et al	Medić et al
100	(0.6229, 0.7340)	(0.6172, 0.7344)	–	(0.6100, 0.7500)
400	(0.5534, 0.6013)	(0.5547, 0.6055)	–	(0.5800, 0.6150)
1000	(0.5267, 0.5534)	(0.5313, 0.5620)	(0.5300, 0.5650)	(0.5450, 0.5600)
5,000	(0.5267, 0.5267)	(0.5117, 0.5352)	(0.5150, 0.5350)	(0.5300, 0.5300)
10,000	(0.5000, 0.5267)	(0.5117, 0.5333)	(0.5117, 0.5300)	(0.5250, 0.5300)

Table 3.1: Comparison of the primary vortex location.

be noted that the maximum Reynolds number achieved on this specific mesh is 25,000. According to author’s knowledge, this is also the highest Reynolds number obtained for the 2D steady incompressible flow in a lid-driven cavity. The results of the present computations are compared with the highly accurate benchmark solutions obtained by [(U. Ghia and Shin 1982)] using a multi-grid strategy for a Reynolds number as high as $Re = 10,000$ on meshes with 129×129 and 257×257 nodes, by [(E. Erturk and Gokcol 2005)] by means of a high-order Finite Difference method up to a Reynolds number of 21,000 on a uniform 601×601 grid size, and by Medić et al [(Medić and Mohammadi 1999)] as based on Chorin’s project method by means of the Finite Element Method on unstructured meshes with 5,000 and 10,000 elements. The comparison of the primary vortex location shows a very good agreement between our solution and the above-mentioned literature (see table 3.4), while a maximum difference of 2.93% is found with respect to the results of [(U. Ghia and Shin 1982)] at $Re = 5,000$.

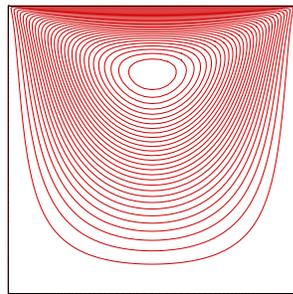
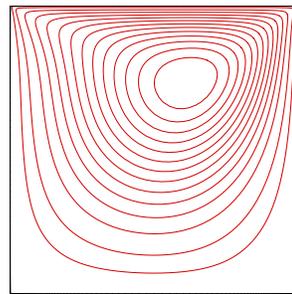
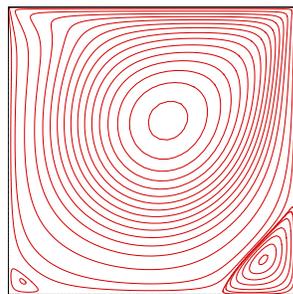
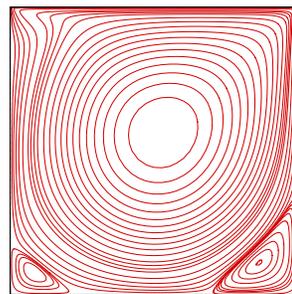
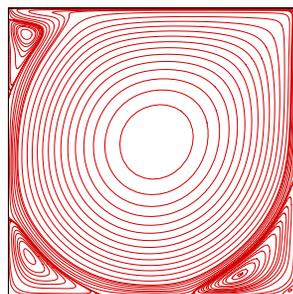
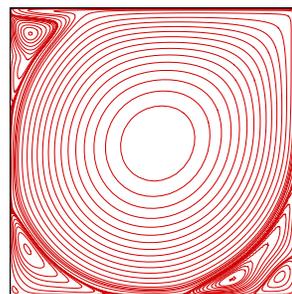


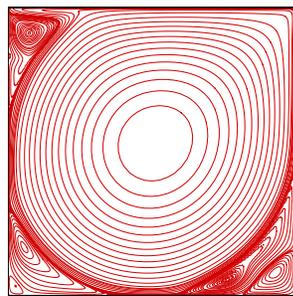
(a) 2D cavity problem geometry



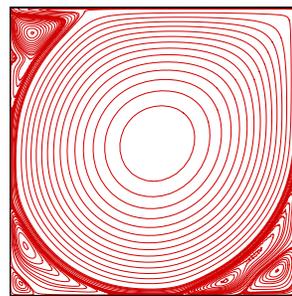
(b) Computational mesh: 10,372 elements, 5331 vertices.

Figure 3.3: 2D cavity problem

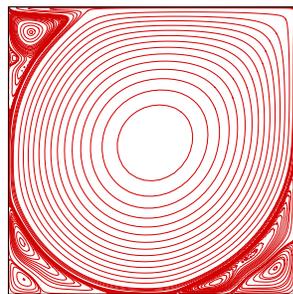
(a) $Re = 1$ (b) $Re = 100$ (c) $Re = 400$ (d) $Re = 1,000$ (e) $Re = 5,000$ (f) $Re = 10,000$ **Figure 3.4:** Streamlines for different Reynolds numbers



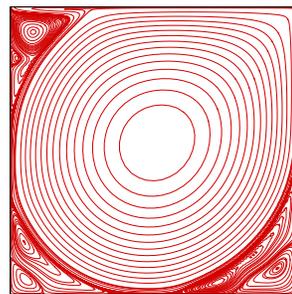
(a) $Re = 12,500$



(b) $Re = 15,000$



(c) $Re = 20,000$



(d) $Re = 25,000$

Figure 3.5: Streamlines for different Reynolds numbers, continued

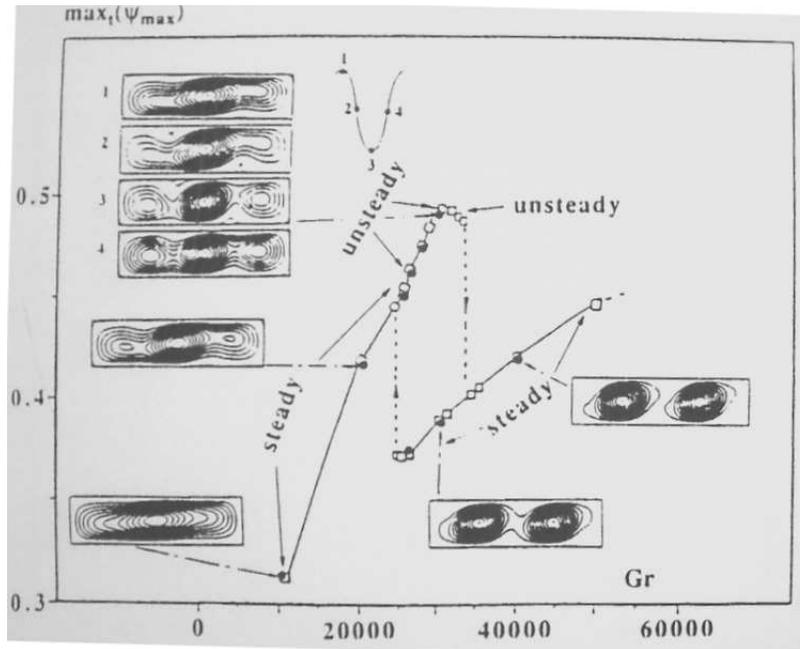


Figure 3.6: Maximum value of the stream function, as a function of the Grashof number, for the case of a two-dimensional flow in a rectangle cavity by Pulicani et al [(Pulicani et al. 1990)].

3.5 From laminar to turbulent flow

As detailed earlier, the molten semi-conductor flow in a Czochralski system is dauntingly complicated due to a complex combination of natural and forced convection, as associated with the buoyancy resulting from the temperature dependence of the density, the Coriolis forces resulting from the rotation of the crucible, the centrifugal pumping due to crystal rotation, the shear stress due to gas convection along the meniscus and the thermocapillary forces due to the temperature dependence of surface tension along the meniscus, etc.. The combination of the above driving forces leads to a complex three-dimensional and time dependent flow structure, such that the melt flow is chaotic and turbulent. For example, the axisymmetric time-dependent simulations of Kim and Langlois [(Kim and Langlois 1991)] on relatively small-scale crucibles show chaotic flow patterns.

Based on similarity analysis, hydrodynamic flows can usually be analyzed

Symbol and Description	Definition
Re_c , crucible rotation Reynolds number	$Re_c = \frac{R_c^2 \Omega_c}{\nu}$
Re_s , crystal rotation Reynolds number	$Re_s = \frac{R_s^2 \Omega_s}{\nu}$
Gr , Grashof number	$Gr = \frac{\beta g \Delta T L^3}{\nu^2}$
Pr , Prandtl number	$Pr = \frac{\nu}{\kappa}$
Ma , Marangoni number	$Ma = \frac{ \gamma_T (R_c - R_s)\Delta T}{\mu\kappa}$
Ra , Rayleigh number	$Ra = Gr \times Pr = \frac{\beta g \Delta T L^3}{\nu\kappa}$
Ta , Taylor number	$Ta = 4\Omega_c^2 R_c^4 / \nu^2$
We , Weber number	$We = \frac{\rho U^2 L}{\sigma}$
Ar , melt aspect ration	$Ar = \frac{H}{R_c}$
Rr , crucible to crystal radius ratio	$Rr = \frac{R_c}{R_s}$

with,

β , the thermal expansion coefficient	g , the acceleration of gravity
ν , the silicon melt kinematic viscosity	R_c , the crucible radius
μ , the silicon melt dynamic viscosity	R_s , the crystal radius
σ , the silicon melt surface tension	Ω_s , the crystal rotation rate
ΔT , a characteristic temperature difference	Ω_c , the crucible rotation rate
$\kappa = k/(\rho c)$, the silicon melt heat diffusivity	L , a characteristic length scale
U , the velocity scale in silicon melt	

Table 3.2: Important dimensionless numbers of Czochralski crystal growth

by several dimensionless numbers. In the crystal growth modeling community, the Czochralski bulk flow can be described by means of several independent dimensionless parameters as indicated in Table (3.2).

However, one must be very careful when computing these dimensionless numbers before using them to draw any general conclusions concerning the flow

structure and nature. For example, the Grashof number is a very important dimensionless number describing the characteristics of the melt flow. However calculating the exact value of this number is not an easy task since it is difficult to define the related length scale and temperature scale due to the constantly evolving melt/crystal geometry, operating conditions and thermal environment. On the other hand, even assuming that the Grashof number is correct, in some cases a completely misleading conclusion can be drawn when using this parameter to describe the flow structure. This was confirmed by the numerical investigations from Crespo del Arco et al [(Crespo del Arco et al. 1997)] [(Crespo del Arco et al. 1989)], Pulicani et al[(Pulicani et al. 1990)] and B. Roux et al [(R. Boux 1990)]. The numerical experiments were conducted on a zero Prandtl number liquid in a rectangular cavity, where a horizontal temperature gradient was imposed, see Figure (3.6). One can observe from their results that, when the Grashof number is increased, the steady one-cell initial structure of the streamlines gives rises to an alternatively oscillating or quasi-periodic flow. A sudden change of the flow regime is observed at a critical Grashof number of about 3.5×10^4 , and a steady two-cell flow is then formed. However, when the Grashof number is decreased, this structure persists down to a Grashof number of about 2.5×10^4 , and a rapid switch to the initial one-cell flow regime is then found.

Although the use of these dimensionless numbers can be sometimes misleading, they still help understand some general flow characteristics and features. The following Table (3.3) gives an order of magnitude of the principal dimensionless numbers for 4 different silicon crystal diameters. Let us notice that, even for a 4" crystal diameter, the Grashof number has a magnitude of 10^9 , which is much larger than the above-mentioned critical Grashof number. Therefore, the melt flow in Czochralski silicon growth is surely asymmetric, unsteady and three-dimensional for nearly all practical production conditions, and the flow in small crucibles is transitional, while for modern large size Czochralski crystal growth systems, the flow is weakly or even fully turbulent.

Crystal diameter	Re_s	Re_c	Gr	Ra	Ma
4"	1.1×10^4	2.3×10^4	2.0×10^9	2.9×10^7	3.8×10^3
8"	4.5×10^4	9.3×10^4	1.6×10^{10}	2.3×10^8	7.6×10^3
12"	1.1×10^5	2.1×10^5	5.6×10^{10}	7.8×10^8	1.1×10^4
16"	1.8×10^5	3.7×10^5	1.3×10^{11}	1.8×10^9	1.5×10^4

Table 3.3: Order of magnitude of the dimensionless parameters for different size of crystal [(Assaker 1998)].

3.6 Turbulence modeling

The previously summarized Navier-Stokes equations governing the melt flow provide vast details about the flow turbulence, from the largest to the smallest length and time scales. Unfortunately, as a consequence of this completeness, using a direct approach (DNS) to solve the Navier-Stokes equations is almost impossible. A natural alternative to DNS is to use a statistical approach associated with the chaotic properties of turbulence.

3.6.1 Reynolds averaged Navier-Stokes equations

Starting from the Navier-Stokes equations, which govern the underlying instantaneous turbulent velocity $\mathbf{U}(x, t)$, it is possible to derive equations for the evolution of the mean and fluctuating velocity fields by means of the Reynolds decomposition. For each turbulent variables $\phi(x, t)$, one can decompose it into an averaged value and some fluctuations around this value:

$$\phi'(x, t) \equiv \phi(x, t) - \langle \phi(x, t) \rangle \quad (3.35)$$

where $\langle \phi(x, t) \rangle$ is the ensemble average (over N repetitions) of variable $\phi(x, t)$, defined as follows:

$$\langle \phi(x, t) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \phi_n(x, t) \quad (3.36)$$

Here, $\phi_n(x, t)$ is the value of ϕ at position x and time t for the n^{th} experiment. One can replace ensemble averaged values by time averaged values (over a time interval T) when assuming a statistically steady turbulence and the validity of the ergodic hypothesis [(Speziale 1995)]: for any n :

$$\langle \phi(x, t) \rangle = \bar{\phi}(x, t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t'}^{t'+T} \phi_n(x, t) dt \quad (\text{for any } n) \quad (3.37)$$

The above defined ensemble on time averaging procedures obey several basic rules as follows:

$$\bar{a} = a \quad (3.38)$$

$$\bar{\phi}' = 0 \quad (3.39)$$

$$\overline{a\phi + b\psi} = a\bar{\phi} + b\bar{\psi} \quad (3.40)$$

$$\overline{\frac{\partial \phi}{\partial x_i}} = \frac{\partial \bar{\phi}}{\partial x_i} \quad (3.41)$$

$$\overline{\frac{\partial \phi}{\partial t}} = \frac{\partial \bar{\phi}}{\partial t} \quad (3.42)$$

$$\overline{\nabla\phi} = \nabla\bar{\phi} \quad (3.43)$$

$$\overline{\nabla \cdot \Phi} = \nabla \cdot \bar{\Phi} \quad (3.44)$$

where a , b are constants, x_i denotes the spatial coordinates, and ϕ , ψ and Φ are three generic variables.

According to the ensemble on time averaging rules, the mean of the continuity equation (3.13) is simply:

$$\overline{\nabla \cdot \mathbf{U}} = \nabla \cdot \bar{\mathbf{U}} = 0 \quad (3.45)$$

Based on the Reynolds decomposition, both the mean velocity \bar{U} and the velocity fluctuation u' are solenoidal, since

$$\nabla \cdot \mathbf{U} = \nabla \cdot (\bar{\mathbf{U}} + \mathbf{u}') = 0 \quad (3.46)$$

while, by subtraction:

$$\nabla \cdot \mathbf{u}' = 0 \quad (3.47)$$

The mean momentum equation can be derived from equation (3.11) as follows. Since, applying Einstein's summation convention on dummy indices,

$$\frac{DU_j}{Dt} = \frac{\partial U_j}{\partial t} + \frac{\partial(U_i U_j)}{\partial x_i} \quad (3.48)$$

in such a way that the mean of the substantial derivative is

$$\frac{\overline{DU_j}}{Dt} = \frac{\partial \bar{U}_j}{\partial t} + \frac{\partial(\overline{U_i U_j})}{\partial x_i} \quad (3.49)$$

while

$$\begin{aligned} \overline{U_i U_j} &= \overline{(\bar{U}_i + u'_i)(\bar{U}_j + u'_j)} \\ &= \overline{\bar{U}_i \bar{U}_j + u'_i \bar{U}_j + u'_j \bar{U}_i + u'_i u'_j} \\ &= \bar{U}_i \bar{U}_j + \overline{u'_i u'_j} \end{aligned} \quad (3.50)$$

plugging equation (3.50) into equation (3.49), provides the mean of the substantial derivative as follows:

$$\begin{aligned} \frac{\overline{DU_j}}{Dt} &= \frac{\partial \bar{U}_j}{\partial t} + \frac{\partial}{\partial x_i} (\bar{U}_i \bar{U}_j + \overline{u'_i u'_j}) \\ &= \frac{\partial \bar{U}_j}{\partial t} + \bar{U}_i \frac{\partial \bar{U}_j}{\partial x_i} + \frac{\partial}{\partial x_i} (\overline{u'_i u'_j}) \end{aligned} \quad (3.51)$$

Defining the mean substantial derivation operator as:

$$\frac{\bar{D}}{\bar{D}t} \equiv \frac{\partial}{\partial t} + \bar{\mathbf{U}} \cdot \nabla \quad (3.52)$$

it results that:

$$\frac{D\bar{U}_j}{Dt} = \frac{\bar{D}\bar{U}_j}{\bar{D}t} + \frac{\partial}{\partial x_i} (\overline{u'_i u'_j}) \quad (3.53)$$

Applying the Reynolds decomposition to the other terms of the momentum equations, like viscous stresses, isotropic pressure and/or any body forces, finally the mean momentum, or the so called Reynolds-averaged Navier-Stokes equations (RANS) are obtained as follows:

$$\rho \frac{D\bar{U}_j}{Dt} = \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \right] - \frac{\partial \bar{p}}{\partial x_i} - \rho \frac{\partial}{\partial x_i} (\overline{u'_i u'_j}) + \rho \beta g_i (\bar{T} - T_0) \quad (3.54)$$

The Navier-Stokes equation (3.1) and the above derived RANS equations are quite the same in appearance. However, there is a crucial difference between them, since an additional so called Reynolds stress term $-\rho \overline{u'_i u'_j}$ is introduced due to the fluctuating velocity field.

The same treatment can be applied to the energy equation and we obtain the mean energy equation as follows:

$$\rho c(T) \frac{\bar{D}\bar{T}}{\bar{D}t} = - \frac{\partial}{\partial x_i} (\bar{q}_i + q_i^{Re}) \quad (3.55)$$

where

$$\bar{q}_i = -k(\bar{T}) \frac{\partial \bar{T}}{\partial x_i} \quad (3.56)$$

is the mean heat flux and q_i^{Re} the turbulent heat flux due to temperature and velocity fluctuations [(Assaker 1998)]:

$$q_i^{Re} = \rho c(T) \overline{u'_i T'} - \overline{(k(T) - k(\bar{T})) \frac{\partial T}{\partial x_i}} \quad (3.57)$$

Here, it was supposed that the thermal conductivity can depend upon temperature. A similar decomposition technique can be applied to the momentum equations if the viscosity is temperature dependent.

3.6.2 Closure problem and turbulent-viscosity hypothesis

For a general statistical flow, there are 4 independent equations governing the mean velocity and pressure fields, namely the 3 components of the Reynolds

equations (3.61) together with the mean continuity equation (3.45). However, these 4 equations contain more than 4 unknowns due to the newly introduced 6 Reynolds stress and 3 Reynolds flux components (note that the energy equation is decoupled in our iterative strategy). Such a set of equations with more unknowns than equations is said to be unclosed and cannot be solved unless the Reynolds stresses are somehow determined. Generally speaking, it is useless to determine the instantaneous, fluctuating velocity and pressure fields in practice. More practical and of more important relevance are the statistical quantities given by the mean values of these variables, which are solution of equations (3.61) and (3.45)

To close the problem governed by the mean continuity and RANS equations, either an additional mathematical model that describes the additional unknown variables as a function of the basic ones, or an additional set of equations governing the new unknowns must be developed.

The turbulent-viscosity hypothesis, which was introduced by Boussinesq in 1877, is mathematically analogous to the stress-rate-of-strain relation for a Newtonian fluid. According to this hypothesis, the deviatoric Reynolds stress

$$a_{ij} = (-\rho \overline{u'_i u'_j} + \frac{2}{3} \rho k \delta_{ij}) \quad (3.58)$$

is proportional to the mean rate of strain,

$$-\rho \overline{u'_i u'_j} + \frac{2}{3} \rho k \delta_{ij} = \rho \nu_t \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \quad (3.59)$$

where the positive scalar coefficient ν_t is the turbulent eddy viscosity, while k is the turbulent kinetic energy, which is defined as follows:

$$k = \frac{1}{2} \overline{u'_i u'_i} \quad (3.60)$$

The mean-momentum equations incorporating the turbulent-viscosity hypothesis can be re-expressed as follows:

$$\rho \frac{\bar{D} \bar{U}_j}{\bar{D} t} = \frac{\partial}{\partial x_i} \left[(\mu + \mu_t) \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \right] - \frac{\partial}{\partial x_j} \left(\bar{p} + \frac{2}{3} \rho k \right) - \rho \frac{\partial}{\partial x_i} (\overline{u'_i u'_j}) + \rho \beta g_i (\bar{T} - T_0) \quad (3.61)$$

where $(\mu + \mu_t)$ is called the effective viscosity (μ_{eff}). This equation is the same as the Navier-Stokes equation with \bar{U} and μ_{eff} in place of U and μ , and $\bar{p} + \frac{2}{3} \rho k$ (the modified mean pressure) in place of p .

It is important to note that the turbulent-viscosity hypothesis has been introduced without justification or criticism so far. For many flows the accuracy of this hypothesis is poor or even wrong for very simple flows. The

turbulent-viscosity hypothesis can be viewed in two steps. First, there is an intrinsic assumption that the Reynolds stress anisotropy a_{ij} is determined by the mean velocity gradients at each point and time. Second, there is a specific assumption that the relationship between a_{ij} and the velocity gradients is governed by equation (3.59). However, according to the wind-tunnel experiment on an axisymmetric contraction by Uberoi [(Uberoi 56)] and Tucker [(Tucker 1970)], together with the DNS results from Lee and Reynolds [(Lee and Reynolds 1985)], the Reynolds stresses are not determined by the rate of strain, but by the total amount of mean strain experienced by the turbulence, and at the same time the turbulent fluid does not behave like a viscous fluid, but more like an elastic solid [(Crow 1968)]. Therefore the turbulent viscosity hypothesis is qualitatively incorrect. Nevertheless, the concept of eddy viscosity often works well in practice and forms the basis of nowadays popular turbulence models in the computational fluid dynamics (CFD) modeling community.

3.6.3 Turbulence models and simulations

The CFD models for turbulent flows can be classified into three main categories: direct numerical simulation (DNS) models, large eddy simulation (LES) models and Reynolds-averaged Navier-Stokes (RANS) models. Actually, DNS does not include any modeling at all, apart from the required numerical approximation and grid resolution technique, and hence DNS can be treated as accurate, or even more accurate than experiments. However, the DNS method is too expensive since the computational cost increases rapidly with the Reynolds number (approximately as Re^3), and hence can only be applied to flows with low or moderate Reynolds number. On the other hand, the use of cut-off in wave-number space enables higher Reynolds number flows to be simulated by LES method. Depending on the approach used to close the mean continuity and RANS equations, the RANS models can be subdivided in two major categories, viz the turbulent-viscosity models, which are based on the intrinsic assumption that the deviatoric Reynolds stresses a_{ij} are locally determined by $\frac{\partial \bar{U}_i}{\partial x_j}$, and the Reynolds stress models, which model the Reynolds stresses directly by Reynolds-stress transport equations. Generally speaking, Reynolds stress models are computationally more expensive and experience more difficulties to converge compared with turbulent viscosity models [(Pope 2000)].

The concept behind any turbulent viscosity model is that the unknown Reynolds stresses are modeled by using known flow parameters, for example the average rate of strain S_{ij} , together with an eddy viscosity. Algebraic or

zero-equation models use an algebraic equation to model the eddy viscosity, while one-equation models, such as the turbulence $k - l$ model, use a turbulent quantity governed by a transport equation (usually the turbulent kinetic energy) and a second turbulent quantity (a turbulent length or time scale) obtained from an algebraic expression or an equation independently by of the first turbulent quantity. Two-equation models employ 2 transport equations to model the eddy viscosity. For two-equation models, the eddy viscosity is typically modeled by a product of powers of the turbulent kinetic energy k and a secondary turbulent quantity ϕ as follows:

$$\nu_t \sim k^a \phi^b \quad (3.62)$$

Based on dimensional analysis, the turbulent viscosity to be modeled has the following form:

$$k - l \quad model : \quad \nu_t \sim \sqrt{kl} \quad (3.63)$$

$$k - \epsilon \quad model : \quad \nu_t \sim \frac{k^2}{\epsilon} \quad (3.64)$$

$$k - \omega \quad model : \quad \nu_t \sim \frac{k}{\omega} \quad (3.65)$$

However, for turbulent-viscosity models, several comments are useful:

- Although the above turbulent-viscosity models are all based on the turbulent viscosity hypothesis (remember that there is no sound physical foundation for this hypothesis), these models still perform reasonably well in many flows and are extensively used in commercial CFD applications and academic researches as well.
- From the modeling aspect, both the two-equation $k - \epsilon$, and $k - \omega$ and the Reynolds stress models are complete and can be applied to any turbulent flow theoretically. However not all models are applicable to all flows in practice. Each model is originally developed for specific kinds of flows with some special terms or calibrated modeling constants. For example, the $k - \epsilon$ model has been very successful in a large variety of flow situations, but is inaccurate for flows with adverse pressure gradients and extremely difficult to integrate across the viscous sublayer [(Wilcox 1998)]. Therefore, many $k - \epsilon$ models incorporate ad-hoc “damping functions” which have been used to fit the model to experimental or DNS computational data. The Wilcox $k - \omega$ model is more accurate for two-dimensional boundary layers with variable pressure gradients [(Wilcox 1998)][(Menter 1992)]. However Wilcox’s $k - \omega$ model is sensitive

to free-stream boundary conditions for free shear flows [(Menter 1992)]. The SST $k - \omega$ model developed by Menter [(Menter 1992)] incorporates a damped cross-diffusion derivative term in the ω equation and fine-tunes the modeling constants to overcome the shortcomings of Wilcox $k - \omega$ model for adverse pressure gradient flows, airfoil flows, transonic shock waves, etc.

- From a numerical viewpoint, two-equation models, especially the $k - \epsilon$ model, are very difficult to solve. Firstly two-equation models often incorporate highly nonlinear damping functions. The behavior of these functions cannot be easily controlled by conventional linearization techniques and can therefore interfere with the algorithm convergence properties [(Wilcox 1998)]. Secondly, the final solution of a two-equation model can be very sensitive to the flow boundary conditions. For example, both ϵ and ω do not tend to zero along a no-slip boundary. As pointed out by Wilcox [(Wilcox 1998)], the k , ϵ and ω quantities satisfy the following equations near the wall:

$$k = \frac{u_\tau^2}{\sqrt{\beta^*}}, \quad \omega = \frac{\sqrt{k}}{(\beta^*)^{1/4} \kappa y}, \quad \epsilon = (\beta^{*3/4}) \frac{k^{3/2}}{\kappa y}$$

Here y is the distance from the wall boundary, $u_\tau = \sqrt{t_w/\rho}$ the friction velocity, which is defined from the wall shear stress t_w and fluid density ρ , while $\kappa = 0.41$ is the von Karman constant and β^* an empirical constant. Also Menter [(Menter 1992)] suggested that for the $k - \omega$ model, it is much easier and as accurate to implement the following boundary condition for ω :

$$\omega = 10 \frac{6\nu}{\beta_1(\Delta y^2)} \quad \text{at} \quad y = 0 \quad (3.66)$$

where Δy is the distance to the next point away from the wall. However, in order to eliminate any numerical errors and to obtain results that are not sensitive to the boundary conditions provided by the above equations, Wilcox and Menter suggest that $7 \sim 10$ grid points are needed for $y^+ = y/\delta_v < 2.5$ with $\delta_v = \nu/u_\tau$. Considering that the viscous sub-layer thickness is of the order of 10^{-2} mm to 10^{-1} mm for large-size crystal growth systems, this is a very stringent condition and an extremely refined mesh near the solid-liquid interface and the crucible wall is indispensable in order to obtain grid-independent, boundary insensitive solutions. Recalling that actual flows in Czochralski melts are also strongly rotating, time-dependent and three-dimensional, therefore both $k - \epsilon$ and

$k - \omega$ models are too computationally expensive for nowadays large-size crystal growth simulations, particularly for time-dependent and/or three-dimensional simulations.

- Both the mixing-length and one-equation $k - l$ turbulence model are incomplete since the mixing-length l_m has to be specified a priori. The appropriate specification of l_m is however dependent on the flow geometry. For a complex flow that has not been studied before, the specification of l_m requires a large measure of guesswork. However for flows that have been studied extensively, appropriate specification of the mixing-length can be well established and quite accurate results can be expected. Therefore, specifying the appropriate mixing-length is the key issue of mixing-length and $k - l$ models and the accuracy of the modeling results will be strongly dependent on the specified mixing-length. The mixing-length model is arguably the simplest turbulence model and rarely causes unexpected numerical difficulties. On the other hand, for the one-equation $k - l$ model, an additional turbulent kinetic energy transport equation is introduced, and therefore the computational cost is slightly higher. Also the $k - l$ model has convergence difficulties when the mesh is too coarse or the initial guess value is too far away from the solution. A comparison of model predictions with experimental data by Wilcox shows that the one-equation $k - l$ model has a modest advantage in accuracy over the mixing-length model. Both the mixing length and $k - l$ models are attractive for practical global and time-dependent crystal growth simulations due to their much lower CPU costs compared with other models and their reasonable accuracy as long as the mixing-length is appropriately specified.

3.7 The mixing-length turbulence model

The mixing length model is the simplest turbulence model to close the system governed by the mean continuity equation (3.45) and RANS equations (3.61). The mixing length hypothesis was initially proposed in 1925 by Prandtl for unidirectional boundary-layer flows parallel to a flat wall. In fact the mixing length is the macroscopic analogous of the mean free path introduced in the kinetic theory of gas molecular momentum transport [(Wilcox 1998)]. According to the Boussinesq eddy-viscosity approximation, the turbulent viscosity ν_t

is then defined in shear flow in such a way that the Reynolds stress is given by

$$-\overline{u'v'} = \nu_t \frac{\partial \bar{U}}{\partial y} \quad (3.67)$$

where the mean flow has a velocity $\bar{U}(y)$ in x direction, whose perturbations are denoted by u and v , while the turbulent eddy-viscosity can be expressed as the product of a velocity scale u^* and a lengthscale l_m (the mixing length):

$$\nu_t = u^* l_m \quad (3.68)$$

Prandtl postulated that this velocity scale u^* (the mixing velocity) can be written as:

$$u^* = l_m \left| \frac{\partial \bar{U}}{\partial y} \right| \quad (3.69)$$

Finally the turbulent eddy-viscosity can be rewritten as:

$$\nu_t = l_m^2 \left| \frac{\partial \bar{U}}{\partial y} \right| \quad (3.70)$$

As mentioned earlier, the Prandtl mixing length hypothesis was originally proposed for unidirectional flows parallel to a flat plate. Several generalizations of equation (3.70) have been introduced in order to provide applications of the mixing length hypothesis for all flows. On the basis of the mean rate-of-strain \bar{d}_{ij} , Smagorinsky [(Smagorinsky 1963)] proposed the following model:

$$\begin{aligned} \nu_t &= l_m^2 \dot{\gamma} \\ &= l_m^2 \sqrt{2\bar{d}_{ij}\bar{d}_{ij}} \end{aligned} \quad (3.71)$$

whereas, based on the mean rate-of-rotation $\bar{\Omega}_{ij}$, Baldwin and Lomax [(Baldwin and Lomax 1978b)] proposed that ν_t can be described as follows:

$$\nu_t = l_m^2 \sqrt{2\bar{\Omega}_{ij}\bar{\Omega}_{ij}} \quad (3.72)$$

Here, \bar{d}_{ij} is an objective symmetric, deviatoric tensor:

$$d_{ij} = \frac{1}{2} \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \quad (3.73)$$

while $\bar{\Omega}_{ij}$ is a non-objective antisymmetric tensor:

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial \bar{U}_i}{\partial x_j} - \frac{\partial \bar{U}_j}{\partial x_i} \right) \quad (3.74)$$

Both equation (3.71) and equation (3.72) reduce to equation (3.70) when $\frac{\partial \bar{U}_1}{\partial x_2}$ is the only non-zero mean velocity gradient.

We have based our mixing length model on Smagorinsky's formula. The only unknown to close the mixing length model is the length scale l_m , which is a key parameter of our model. Recalling that Prandtl's mixing length hypothesis was proposed on the basis of an analogy with gas molecular momentum transport nearly one century ago, it should, however, be observed that the fundamental mechanisms of molecular and turbulent eddy diffusions are different. Therefore the validity of the analogy between these processes is quite flimsy. Even at current stage the value and physical interpretation of the mixing length still remains unknown. On the other hand, although the length scale is not rigorously defined, the Prandtl hypothesis greatly simplifies the turbulence closure problem and surprisingly sometimes does an excellent job for reproducing some experimental measurements. Moreover, the model can be easily calibrated for specific classes of flows.

3.7.1 Mixing-length modeling

The only unknown in equation (3.71) is l_m , which is a semi-empirical and flow dependent quantity. For example, the mixing length is constant across the mixing layer and proportional to the width of the layer for free shear flows. However, for flows near solid boundaries, turbulence behaves differently. Prandtl originally postulated that, for the flow near a solid boundary, the mixing length is proportional to the distance from the surface:

$$l_m = Cd \quad (3.75)$$

where d is the distance to the flow boundary and C is an empirical constant which should be determined experimentally for each particular flow configuration. Prandtl's postulate is proved to be consistent with the well-known law of the wall, which has been observed for a wide range of wall-bounded flows. For the wall-bounded flows observed in crystal growth, it can be found experimentally that C equals the von Karman constant κ , or $C = \kappa = 0.41$.

A smooth approximation of the distance to the boundary d can be obtained by solving the following equation, ref Lecomte et al [(Lecomte et al. 1992)] and Assaker[(Assaker 1998)]:

$$\Delta \Delta d = 0 \quad \text{in } \Omega_L \quad (3.76)$$

where Δ is the Laplacian operator. At the melt boundary of a Czochralski crystal growth system, the boundary conditions along the melt/crucible interface

Γ_c and melt/crystal interface Γ_s are as follows:

$$\begin{cases} d = 0 \\ \frac{\partial d}{\partial n} = 1 \end{cases} \quad \text{on } \Gamma_s \cup \Gamma_c \quad (3.77)$$

while along the melt/gas interface Γ_m , either a wall same as equation (3.77) or a mirror boundary condition as follows can be chosen:

$$\begin{cases} \frac{\partial d}{\partial n} = 0 \\ \frac{\partial \Delta d}{\partial n} = 0 \end{cases} \quad \text{on } \Gamma_m \quad (3.78)$$

where n is the normal to the corresponding boundary.

Note that the equation governing the distance to the boundary d is a function of the melt geometry only, in such a way that it can be solved decoupled from other physical unknowns like the temperature, velocity and pressure fields. In order to solve the above differential equation by the FEM method, we introduce an additional temporary variable λ , which is defined as follows:

$$\Delta d = \lambda \quad (3.79)$$

and hence equation (3.76) becomes:

$$\begin{cases} \Delta d - \lambda = 0 \\ \Delta \lambda = 0 \end{cases} \quad (3.80)$$

Using the same discretization for d and λ as for the velocity field,

$$\lambda = \sum_j^N \lambda_j \psi_j(r, z) \quad (3.81)$$

$$d = \sum_j^N d_j \psi_j(r, z) \quad (3.82)$$

where N is the number of nodes and choosing the same test functions as the shape functions ψ_j , we obtain the weak formulation of these equations as follows:

$$\int_{\Omega} (\nabla d \cdot \nabla \psi_i + \lambda \psi_i) d\Omega = \int_{\Gamma} \frac{\partial d}{\partial n} \psi_i d\Gamma \quad (3.83)$$

$$\int_{\Omega} (\nabla \lambda \cdot \nabla \psi_i) d\Omega = \int_{\Gamma} \frac{\partial \lambda}{\partial n} \psi_i d\Gamma \quad (3.84)$$

Plugging equations (3.81) and (3.82) into the weak formulations, we obtain the following set of linear equations without taking yet the boundary conditions into account:

$$\begin{bmatrix} \int_{\Omega} (\nabla\psi_i \cdot \nabla\psi_j) d\Omega & \int_{\Omega} (\psi_i\psi_j) d\Omega \\ 0 & \int_{\Omega} (\nabla\psi_i \cdot \nabla\psi_j) d\Omega \end{bmatrix} \begin{bmatrix} d_j \\ \lambda_j \end{bmatrix} = \mathbf{0} \quad (3.85)$$

The final local matrices and right-hand side vectors are dependent on the boundary conditions applied to the crucible wall, melt/crystal interface, or meniscus. We will not address the details here. All these local matrices and right-hand vectors can be assembled together and the resulting system can easily be solved by direct or iterative solvers.

3.7.2 Enhanced mixing-length model

The differential equation (3.76) governing the distance d proposed by Lecomte et al [(Lecomte et al. 1992)] provides a smooth, accurate and unique value of the distance between the finite element nodes and the melt domain boundary. This is particularly important in the vicinity of the domain corners where more than one boundary side, and thus more than one distance, coexist [(Assaker 1998)]. The highest accuracy in distance evaluation is reached close to the boundary, see equation (3.77). However, there are two main drawbacks in equation (3.76): firstly, the mixing-length estimation gets poorer in the core of the melt and, secondly, it does not take the mixing length boundary layer thickness into account. For example, when applying the differential equation (3.76) to the 1D case, the analytical solution of equation (3.76) is:

$$d = y(1 - \frac{1}{L}y) \quad (3.86)$$

where L is the length of the 1D domain. Selecting $L = 1$ for simplicity, the approximate mixing-length l_m is:

$$l_m = \kappa y(1 - y) \quad (3.87)$$

and hence for $y \rightarrow 0^+$, the above equation becomes:

$$l_m = \kappa y \quad (3.88)$$

This result is consistent with the Prandtl mixing length hypothesis and the well-known universal law of the wall for wall-bounded flows. However, experimental

studies show that in the turbulence core (far away from the wall), the mixing length l_m only slightly changes with the flow geometry and is limited by a peak value. Therefore different variants of the mixing length models were proposed by Van Driest [(Driest 1956)], Smith and Cebeci [(Smith and Cebeci 1967)], Baldwin and Lomax [(Baldwin and Lomax 1978a)] to fit experimental data. For example, in Van Driest's model, the mixing length is multiplied by a damping function near the wall as follows:

$$l_m = \kappa y \left(1 - e^{(-y^+/A_0^+)} \right) \quad (3.89)$$

where A_0^+ is a constant and $A_0^+ = 26$. Escudier [(Escudier 1966)] found that the model predictive accuracy is improved by limiting the peak value of the mixing length as follows:

$$(l_m)_{max} = 0.09\delta \quad (3.90)$$

where δ is the boundary-layer thickness. Cebeci and Smith [(Smith and Cebeci 1967)], and Baldwin and Lomax [(Baldwin and Lomax 1978a)] also introduced two-layer mixing length models by differently defining the mixing length in those two layers. In the inner layer, the mixing length is the same as the one proposed by Van Driest, while in the outer layer, the turbulent viscosity is limited by a peak value described by some empirical closure coefficients and flow dependent characteristics, for example the boundary layer thickness. Details of these models are summarized by Wilcox [(Wilcox 1998)].

Actually, the distance to boundary equation (3.76) can be considered as a model where an infinite mixing length boundary layer thickness near the melt/crucible and melt/crystal interfaces is used. However, according to Karman's similarity solution (see also [(Colley et al. 1999)] [(Owen and Rogers 1989)]), the boundary layer thickness along a rotating disk is given by

$$\delta = 5.5 \left(\frac{\nu}{2\pi f_D} \right)^{1/2} \quad (3.91)$$

where f_D is the rotation frequency in Hertz. In order to grow purely axisymmetric crystals, the crystal and crucible are normally rotating at different and opposite rates. Therefore, there are at least two boundary layers with different boundary layer thicknesses near the melt/crucible, and melt/crystal walls and it is necessary to take these two different boundary layers into account.

To overcome the two above mentioned drawbacks, enhanced equations governing the effective distance to the boundary are proposed as follows on the basis of an idea of Winkelmanns (private communication):

$$\begin{cases} \Delta \Delta d + \nabla \cdot (\alpha \nabla d) = 0 \\ \Delta \alpha = 0 \end{cases} \quad (3.92)$$

The boundary conditions for these two equations must be detailed: first, along the melt/crucible and melt/ crystal interfaces, the boundary conditions for d and α are:

$$\begin{cases} d = 0 \\ \frac{\partial d}{\partial n} = -C/\kappa \\ \alpha = -\left(\frac{C}{P}\right)^2 \end{cases} \quad (3.93)$$

where C and P may be selected as constants (which can however differ along the melt/crucible and melt/crystal interfaces). It should be noted that the asymptotic properties of equation (3.92) at low values of P/C show that the the mixing length boundary layer thickness can be described by combining these two constants as follows (see Figure 3.7 and Figure 3.9):

$$\delta_l = P/C \quad (3.94)$$

and if $P \rightarrow \infty$, the enhanced equations (3.92) will thus degenerate into the

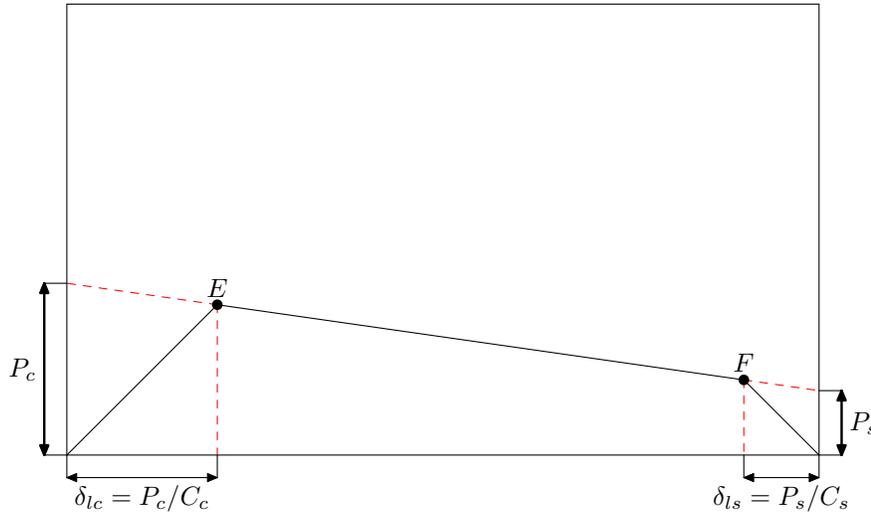


Figure 3.7: *Enhanced mixing-length model.*

original governing equation (3.76) proposed by [(Lecomte et al. 1992)].

Finally, as boundary condition for α at the melt free surface, we have:

$$\frac{\partial \alpha}{\partial n} = 0 \quad (3.95)$$

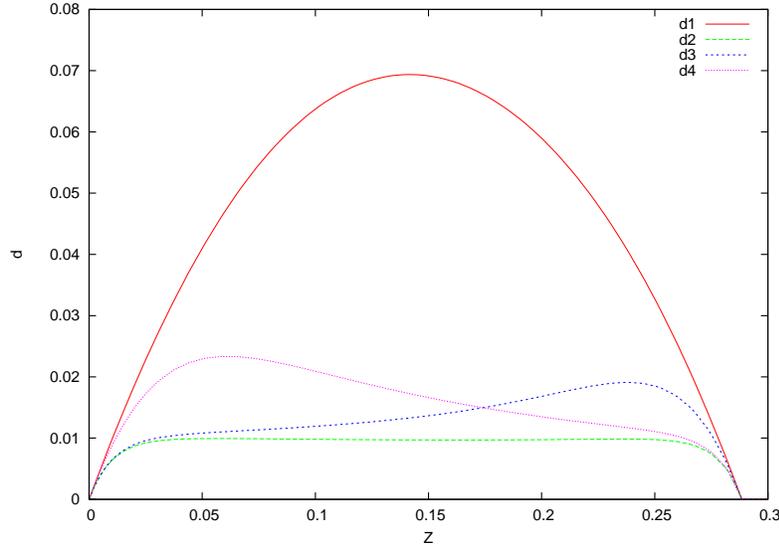


Figure 3.8: Distance profiles along the symmetry axis:

$$C_s = C_c = 0.41 \text{ for all tests;}$$

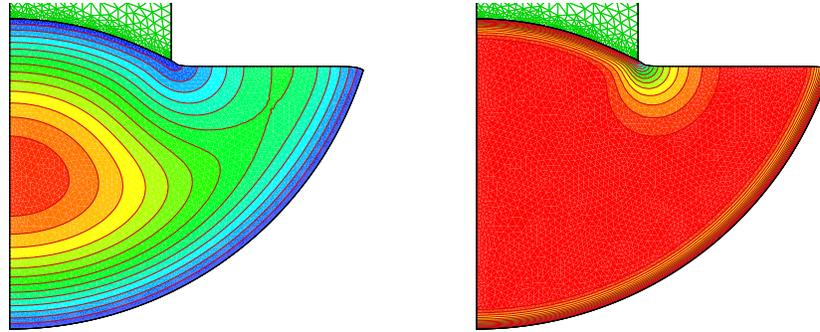
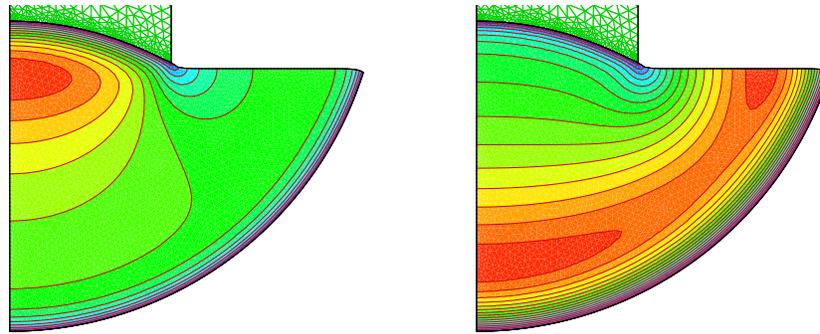
$$d_1 (P_s = P_c = \infty); d_2 (P_s = P_c = 1.0 \times 10^{-3} m);$$

$$d_3 (P_s = 1.0 \times 10^{-1} m, P_c = 1.0 \times 10^{-3} m);$$

$$d_4 (P_s = 1.0 \times 10^{-3} m, P_c = 1.0 \times 10^{-1} m)$$

and for the boundary conditions of d on the melt free surface, they are exactly the same as in equation (3.78). The mixing length l_m is again calculated from d by equation (3.75).

The enhanced distance to boundary equation (3.92) can be solved by direct or iterative solvers using the same procedures as described previously for equation (3.76). However the governing equation (3.92) is nonlinear since d and α are coupled, and therefore using Newton-Raphson's method to linearize the discretized equations is possible. Figure (3.8) shows the profiles of d along the melt symmetry axis with different parameters. Figure (3.9) shows the complete corresponding distance profiles in the melt domain. From these numerical tests, we can see that the maximum of d is reached in the core of the melt, and also that the thickness of the boundary layers can be easily controlled by the enhanced mixing length model introduced.

(a) $P_s = P_c = \infty$, d step=4.3mm(b) $P_s = P_c = 10^{-3}m$, d step=0.6mm(c) $P_s = 10^{-1}m$, $P_c = 10^{-3}m$, d step=1.2mm (d) $P_s = 10^{-3}m$, $P_c = 10^{-1}m$, d step=0.15mm**Figure 3.9:** Profiles of enhanced d with different parameters, $C_s = C_c = 0.41$

3.8 Turbulence $k - l$ model

3.8.1 Modeling of the turbulent kinetic energy equation

As explained earlier, on the basis of dimensional analysis the turbulent viscosity is given by:

$$\nu_t = C_\mu^{1/4} l \sqrt{k} \quad (3.96)$$

where C_μ is a constant (whose value will be discussed later), l the length scale of the large turbulent eddies and k the turbulent kinetic energy. In order to close the problem, a turbulent kinetic energy transport equation has to be modeled. The exact turbulent kinetic energy transport equation directly derived from Navier-Stokes equations is as follows:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j} \left(\nu \frac{\partial k}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left(\frac{1}{2} \overline{u'_i u'_i u'_j} + \frac{\overline{p u'_j}}{\rho} \right) - \overline{u'_i u'_j} \frac{\partial \bar{U}_i}{\partial x_j} - \nu \frac{\partial u'_i}{\partial x_k} \frac{\partial u'_i}{\partial x_k} - \beta g_i \overline{T u'_i} \quad (3.97)$$

The left-hand side is a transient and convective transport term, while the right-hand terms can be interpreted as a viscous diffusion term, a turbulence and pressure diffusion term, a production term, a dissipation term and a buoyancy term, respectively. Unfortunately, the derived equation introduces new additional unknown fluctuation correlations. A model k equation therefore should be developed to represent the unknown correlations as approximate functions of the mean or kinetic energy variables.

DNS results from Mansour, Kim and Moin [(Mansour et al. 1988)] show that the pressure diffusion term is quite small when compared with the turbulent diffusion term and therefore can be neglected. In analogy with the eddy viscosity concept, the right-hand turbulent diffusion term grouped with the pressure diffusion term can be represented as follows:

$$\frac{1}{2} \overline{u'_i u'_i u'_j} + \frac{\overline{p u'_j}}{\rho} \approx - \frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \quad (3.98)$$

where σ_k is an empirical constant. Since the Reynolds stress term can be written as:

$$\overline{u'_i u'_j} = 2\nu_t d_{ij} - \frac{2}{3} k \delta_{ij} \quad (3.99)$$

for incompressible flows, the production term can be rewritten as:

$$-\overline{u'_i u'_j} \frac{\partial \bar{U}_i}{\partial x_j} = 2\nu_t d_{ij} d_{ij} = P_k \quad (3.100)$$

At high Reynolds number, the dissipation rate ϵ scales as $(u^*)^3/l$, where u^* and l are the velocity scale and lengthscale respectively. Consequently, it is reasonable to model ϵ as

$$\epsilon = C_d k^{3/2} / l_m \quad (3.101)$$

where C_d is a model constant, and l_m the mixing-length. Here we still need a prescription for the turbulence length-scale in order to close the system. As for the mixing-length model, we still assume that this length-scale is proportional

to the distance d to the boundary, as governed by the enhanced distance to boundary equation (3.92) presented in the previous section.

Substituting equation (3.57) for the buoyancy term, we obtain:

$$G_k = \beta g_i \overline{T u_i'} = k_t \beta g_i \frac{\partial \bar{T}}{\partial x_i} \quad (3.102)$$

According to the Prandtl number definition, the turbulent thermal conductivity in the energy equations can be expressed as follows:

$$k_t = \frac{c}{Pr_t} \mu_t = \frac{C_\mu^{1/4}}{Pr_t} \rho c l \sqrt{k} \quad (3.103)$$

where Pr_t and c are the turbulent Prandtl number and heat capacity of the fluid respectively. According to Rodi [(Rodi 1985)], experiments have shown that the Prandtl number only varies slightly in any given flow and also slightly from flow to flow.

Combining all the above equations, the k equation becomes:

$$\frac{Dk}{Dt} = \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right) + P_k - C_d k^{3/2} / l_m + G_k \quad (3.104)$$

To completely close the turbulent kinetic energy equation, the empirical constants C_μ , σ_k and C_d and the turbulent Prandtl number Pr_t must be specified. The choice of the empirical constants used in our model was provided by Hanjalic and Launder for fully developed axisymmetric flow in a plane channel [(Hanjalic and Launder 1972)]:

$$C_\mu = 0.09 \quad \sigma_k = 1.0 \quad \text{and} \quad C_d = C_\mu^{3/4} = 0.164 \quad (3.105)$$

Therefore,

$$C_\mu^{1/2} = 0.3 \quad \text{and} \quad C_\mu^{1/4} = 0.548 \quad (3.106)$$

However, the optimal set of constants for a Czochralski flow is not yet available [(Assaker 1998)] and additional experimental and numerical work is thus necessary to provide the appropriate set of constants in a crystal growth system.

There are two different boundary conditions associated with the turbulent kinetic energy equation. Along the solid walls, for example the melt/crucible and the melt/crystal interfaces Γ_c and Γ_s , k is simply imposed to be zero. On the melt free surface, the following mirror boundary condition is imposed:

$$\frac{\partial k}{\partial n} = 0 \quad (3.107)$$

3.8.2 Numerical method

The set of equations governing the turbulence $k-l$ model will be the momentum equation (3.11), the continuity equation (3.13), the turbulent kinetic energy equation (3.104) and the energy equation (3.16). The turbulent kinematic viscosity ν_t , turbulent conductivity k_t and mixing length l_m are governed by equation (3.96), equation (3.103) and equation (3.92) respectively. As earlier mentioned, the mixing length is a function of melt geometry only and can be pre-calculated before the other physical unknowns. However for the newly introduced turbulent kinetic energy equation, the convective transport term and the production term P_k both are functions of the velocity field, while the turbulent viscosity in the momentum equation is also a function of k . Therefore we choose to solve the turbulent kinetic energy k together with the velocity and pressure fields in a coupled way. Nevertheless, since the same strategy is used in the laminar model, the energy equation governing the temperature field is decoupled from all other equations.

The weak formulation and the discretization of equations (3.11) and (3.13) have already been presented in the previous section, so here we only focus on the weak form of the turbulent kinetic energy equation (3.104). Letting k' denote the test functions, the weak formulation of (3.104) is:

$$\int_{\Omega} \left(\frac{Dk}{Dt} - \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right) - P_k + \epsilon - G_k \right) k' d\Omega = 0 \quad (3.108)$$

where $P_k = 2\nu_t d_{ij} d_{ij}$, $\epsilon = C_d k^{3/2} / l_m$ and $G_k = k_t \beta g_i \frac{\partial \bar{T}}{\partial x_i}$. For k the same discretization is used on each element as for the mean velocity field,

$$k(r, z) = \sum_j^{n_k} k_j \psi_j(r, z) \quad (3.109)$$

where n_k is the number of nodes per element and k_j the turbulent kinetic energy at node j . Choosing the test-functions as equal to the shape functions, plugging equations (3.100), (3.101), (3.102) and (3.103) into the above equation and performing Gauss integration by part, the final weak formulation for quasi-steady flow becomes:

$$\begin{aligned} & \int_{\Omega} (U \cdot \nabla k \psi_i) d\Omega + \int_{\Omega} \left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \cdot \nabla \psi_i d\Omega \\ & - \int_{\Omega} (\nu_t 2d : d) \psi_i d\Omega + \int_{\Omega} (C_d k^{3/2} / l_m) \psi_i d\Omega - \int_{\Omega} (c\nu_t \mathbf{g} \cdot \nabla T) \psi_i d\Omega \\ & = \int_{\Gamma} \left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial n} \psi_i d\Gamma \end{aligned} \quad (3.110)$$

Coupling together the above equation with the equations governing the velocity and pressure fields and applying the corresponding boundary conditions, finally we obtain after assembly a set of nonlinear algebraic equations governing the nodal velocity, pressure and turbulent kinetic energy fields on the domain as follows:

$$\begin{cases} F_u(u, v, w, p, k) = 0 \\ F_v(u, v, w, p, k) = 0 \\ F_w(u, v, w, p, k) = 0 \\ F_p(u, v, w, p, k) = 0 \\ F_k(u, v, w, p, k) = 0 \end{cases} \quad (3.111)$$

At each Newton-Raphson iteration, the following linearized set of equations has to be solved:

$$\begin{bmatrix} \frac{\partial F_u^i}{\partial u_j} & \frac{\partial F_u^i}{\partial v_j} & \frac{\partial F_u^i}{\partial w_j} & \frac{\partial F_u^i}{\partial p_j} & \frac{\partial F_u^i}{\partial k_j} \\ \frac{\partial F_v^i}{\partial u_j} & \frac{\partial F_v^i}{\partial v_j} & \frac{\partial F_v^i}{\partial w_j} & \frac{\partial F_v^i}{\partial p_j} & \frac{\partial F_v^i}{\partial k_j} \\ \frac{\partial F_w^i}{\partial u_j} & \frac{\partial F_w^i}{\partial v_j} & \frac{\partial F_w^i}{\partial w_j} & \frac{\partial F_w^i}{\partial p_j} & \frac{\partial F_w^i}{\partial k_j} \\ \frac{\partial F_p^i}{\partial u_j} & \frac{\partial F_p^i}{\partial v_j} & \frac{\partial F_p^i}{\partial w_j} & \frac{\partial F_p^i}{\partial p_j} & \frac{\partial F_p^i}{\partial k_j} \\ \frac{\partial F_k^i}{\partial u_j} & \frac{\partial F_k^i}{\partial v_j} & \frac{\partial F_k^i}{\partial w_j} & \frac{\partial F_k^i}{\partial p_j} & \frac{\partial F_k^i}{\partial k_j} \end{bmatrix} \begin{bmatrix} \delta u_j \\ \delta v_j \\ \delta w_j \\ \delta p_j \\ \delta k_j \end{bmatrix} = - \begin{bmatrix} F_u^i \\ F_v^i \\ F_w^i \\ F_p^i \\ F_k^i \end{bmatrix} \quad (3.112)$$

A key issue results from the fact that the Reynolds equations are highly nonlinear due to the advection term. The newly introduced turbulent viscosity $\nu_t = C_\mu^{1/4} l \sqrt{k}$ in the Reynold equations makes the nonlinearity even worse. In addition, the coupling of the highly nonlinear turbulent kinetic energy equation to the Reynolds equations dramatically increases the numerical difficulties encountered when trying to simulate the melt flow. These difficulties come from two reasons. First, due to the dominating nonlinear transport terms, the equations are numerically of hyperbolic nature and Galerkin's method is thus non-optimal [(Assaker 1998)]. Secondly, the square-root of k present in the turbulent viscosity expression leads to numerical failure when the turbulent kinetic energy k becomes negative during the Newton-Raphson iterative scheme and therefore the quadratic convergence rate is not guaranteed. The methods to overcome these difficulties will be addressed in the next section.

3.9 Generic transformation method to avoid negative k

The nonlinear algebraic system coupling the Reynolds equations with the k equation is very difficult to solve due to its numerically hyperbolic nature. Furthermore, the k equation governing the turbulent kinetic energy is quite similar to a generic advection-diffusion system. When the viscous diffusion term dominates the turbulent kinetic energy transport process, numerical difficulties (with oscillations or even negative k) will often happen during the Newton-Raphson iterative process if the mesh is too coarse or if the initial guess solution is too far away from the real one. Numerical oscillations and negative k values will dramatically slow down the Newton-Raphson convergence rate and even may lead to numerical failure. These difficulties also exist for the solution of any two-equation (such as $k - \epsilon$ and $k - \omega$ model) by FEM or FVM method.

Several procedures were investigated by Assaker [(Assaker 1998)], Habets and Nyevelt [(Habets and Nyevelt 1994)] to overcome the above mentioned numerical difficulties. Firstly, the initial flow field and k distribution of the $k-l$ model were always generated by means of the mixing length model. Secondly, to avoid negative k values, appropriate variable transformations of the form,

$$k = f(\tilde{k}) > 0 \quad (3.113)$$

were introduced and the k equation to be solved then became:

$$F_k(k) = F_k(f(\tilde{k})) = 0 \quad (3.114)$$

The simplest transformation consists in adding an appropriate positive number to the actual k value. Another way to avoid negative value of k consists in using an adaptive relaxation to guarantee

$$k^{r+1} = k^r + \alpha^r \delta k^r > 0 \quad (3.115)$$

at each Newton-Raphson iteration, where α^r is the relaxation coefficient. Although this method can avoid negative k values at all nodes, the k values at Gauss integration points may still be negative due to quadratic interpolation. Moreover, all the above trials showed an unacceptable decrease of the convergence rate [(Assaker 1998)]. Therefore, more complicated transformations to overcome the numerical difficulties seem to be necessary.

In the following section, a generic transformation algorithm is presented in detail in order to overcome the two numerical difficulties encountered in the Newton-Raphson iterative scheme.

3.9.1 Transformation scheme and discretization

Let the new unknown L be defined as:

$$L = nk^{1/n} \quad (3.116)$$

in such a way that

$$k = \left(\frac{L}{n}\right)^n \quad (3.117)$$

It is now L (and not k) which is discretized as follows:

$$L = \sum_j^{n_L} L_j \psi_j \quad (3.118)$$

where $n_L = n_k$. A family of test-functions,

$$k' = k^{(1-1/m)} \psi_i = \left(\frac{L}{n}\right)^{(1-1/m)n} \psi_i \quad (3.119)$$

is considered in order to investigate the effect of the parameter m , whose value is varied from 1 (the simplest case) to ∞ (a "self-adjoint" model for diffusive terms). Plugging equation (3.117) into the k equation (3.104), then we obtain the transformed turbulent kinetic energy equation in weak formulation as follows:

$$\begin{aligned} & \int_{\Omega} \left[\left(U \cdot \nabla \left(\frac{L}{n} \right)^n \right) \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) \right] d\Omega \\ & + \int_{\Omega} \left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla \left(\frac{L}{n} \right)^n \cdot \nabla \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) d\Omega \\ & - \int_{\Omega} (\nu_t 2d : d) \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) d\Omega \\ & + \int_{\Omega} (C_d k^{3/2} / l_m) \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) d\Omega - \int_{\Omega} (c \nu_t \mathbf{g} \cdot \nabla T) \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) d\Omega \\ & = \int_{\Gamma} \left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial n} \left(\left(\frac{L}{n} \right)^{n(1-1/m)} \psi_j \right) d\Gamma \end{aligned} \quad (3.120)$$

where m and n are integer constants and

$$\nu_t = C_{\mu}^{1/4} l_m \sqrt{k} = C_{\mu}^{1/4} l_m \left(\frac{L}{n} \right)^{n/2} \quad (3.121)$$

In each Newton-Raphson iteration, the original set of linearized equations for the $k-l$ model is transformed into a linear system as follows:

$$\begin{bmatrix} \frac{\partial F_u^i}{\partial u_j} & \frac{\partial F_u^i}{\partial v_j} & \frac{\partial F_u^i}{\partial w_j} & \frac{\partial F_u^i}{\partial p_j} & \frac{\partial F_u^i}{\partial L_j} \\ \frac{\partial F_v^i}{\partial u_j} & \frac{\partial F_v^i}{\partial v_j} & \frac{\partial F_v^i}{\partial w_j} & \frac{\partial F_v^i}{\partial p_j} & \frac{\partial F_v^i}{\partial L_j} \\ \frac{\partial F_w^i}{\partial u_j} & \frac{\partial F_w^i}{\partial v_j} & \frac{\partial F_w^i}{\partial w_j} & \frac{\partial F_w^i}{\partial p_j} & \frac{\partial F_w^i}{\partial L_j} \\ \frac{\partial F_p^i}{\partial u_j} & \frac{\partial F_p^i}{\partial v_j} & \frac{\partial F_p^i}{\partial w_j} & \frac{\partial F_p^i}{\partial p_j} & \frac{\partial F_p^i}{\partial L_j} \\ \frac{\partial F_L^i}{\partial u_j} & \frac{\partial F_L^i}{\partial v_j} & \frac{\partial F_L^i}{\partial w_j} & \frac{\partial F_L^i}{\partial p_j} & \frac{\partial F_L^i}{\partial L_j} \end{bmatrix} \begin{bmatrix} \delta u_j \\ \delta v_j \\ \delta w_j \\ \delta p_j \\ \delta L_j \end{bmatrix} = - \begin{bmatrix} F_u^i \\ F_v^i \\ F_w^i \\ F_p^i \\ F_L^i \end{bmatrix} \quad (3.122)$$

and in each Newton-Raphson iteration δL , but not δk is to be solved.

However, in practice we never introduce the new variable L into the system in order to keep the solver consistency and simplicity. In other words, the nodal values L_i are not stored, but are directly obtained from the nodal values k_i by equation (3.116). Therefore, some key issues are to be addressed. Firstly, in order to discretize the nonlinear system, the derivatives of all terms involving L have to be developed as a function of the nodal value k_i , including the Reynolds equations associated with the turbulent viscosity. Secondly, an appropriate incrementation algorithm for Newton-Raphson iterations has to be implemented. Finally, the boundary conditions related to the k equation have to be slightly changed.

The discretization of L and calculation of the associated derivatives are quite complicated and will not be addressed here (refer to appendix A). Concerning the incremental method for Newton-Raphson iterations, we have:

$$k_i^{r+1} = k_i^r + \delta k_i^r \quad (3.123)$$

while solving the linear system provides δL_i^r , with

$$\delta L_i^r = n(k_i^{r+1})^{1/n} - n(k_i^r)^{1/n} \quad (3.124)$$

or

$$\frac{k_i^{r+1}}{k_i^r} = \left[1 + \left(\frac{\delta L_i^r}{n} \right) (k_i^r)^{-1/n} \right]^n \quad (3.125)$$

The right-hand side of equation (3.125) can be approximated by a Padé fraction whose numerator and denominator are both second degree polynomials. Letting

$$\left(1 + \frac{x}{n} \right)^n = \frac{1 + \alpha x + \beta x^2}{1 + \alpha' x + \beta' x^2} \quad (3.126)$$

a Mac-Laurin development of the left-hand side provides the expansion:

$$1 + \alpha x + \beta x^2 = (1 + \alpha x' + \beta' x^2) \left(1 + x + \frac{n-1}{2n} x^2 + \frac{(n-1)(n-2)}{6n^2} x^3 + \frac{(n-1)(n-2)(n-3)}{24n^3} x^4 + \dots \right) \quad (3.127)$$

Therefore, to reach the maximal order in equation (3.127), the following equations have to be solved:

$$\begin{cases} \alpha = \alpha' + 1 \\ \beta = \beta' + \alpha' + \frac{n-1}{2n} \\ \beta' + \frac{n-1}{2n} \alpha' + \frac{(n-1)(n-2)}{6n^2} = 0 \\ \beta' + \frac{n-2}{3n} \alpha' + \frac{(n-2)(n-3)}{12n^2} = 0 \end{cases} \quad (3.128)$$

whose solution is:

$$\alpha = \frac{n+2}{2n} \quad \beta = \frac{(n+1)(n+2)}{12n^2} \quad \alpha' = -\frac{n-2}{2n} \quad \beta' = \frac{(n-1)(n-2)}{12n^2} \quad (3.129)$$

Finally, the incremental method develops as follows:

$$\frac{k_i^{r+1}}{k_i^r} = \frac{1 + \frac{n+2}{2n} \left(\frac{\delta L_i^r}{(k_i^r)^{1/n}} \right) + \frac{(n+1)(n+2)}{12n^2} \left(\frac{\delta L_i^r}{(k_i^r)^{1/n}} \right)^2}{1 - \frac{n-2}{2n} \left(\frac{\delta L_i^r}{(k_i^r)^{1/n}} \right) + \frac{(n-1)(n-2)}{12n^2} \left(\frac{\delta L_i^r}{(k_i^r)^{1/n}} \right)^2} \quad (3.130)$$

Since the discriminants of the quadratic polynomials are negative:

$$\rho = \alpha^2 - 4\beta = -\frac{(n+2)(n-2)}{12n^2} \leq 0 \quad \forall n \geq 2 \quad (3.131)$$

$$\rho' = \alpha'^2 - 4\beta' = -\frac{(n+2)(n-2)}{12n^2} \leq 0 \quad \forall n \geq 2 \quad (3.132)$$

equation (3.130) provides a positive left-hand side:

$$\frac{k_i^{r+1}}{k_i^r} > 0 \quad \forall n \geq 2 \quad (3.133)$$

This algorithm is a generic one since we can choose different integer constants n and m to implement different complex transformation schemes without changing any code implementation details.

In addition, a special incremental algorithm has to be developed when $n \rightarrow \infty$. In this case, the transformation provides the so called $\log k$ method, since

$$\lim_{n \rightarrow \infty} (1 + L/n)^n = e^L = \lim_{n \rightarrow \infty} e^{nk^{1/n}} = k \quad (3.134)$$

Therefore, for $\log k$ method,

$$k_i^{r+1} = e^{L_i^r + \delta L_i^r} = k_i^r e^{\delta L_i^r} \quad (3.135)$$

Because δk is an exponential function of δL , if the initial guess solution is too far away from the exact solution, any quite large δL will dramatically decrease the Newton-Raphson convergence rate. Therefore, the following function is introduced to approximate the exponential $e^{\delta L}$ while limiting its variations:

$$e^{\delta L_i^r} \approx f(\delta L_i^r) + \sqrt{1 + f^2(\delta L_i^r)} \quad (3.136)$$

where $f(x)$ is an approximation function, whose role is to cut-off the overshoots. For this approximation function, the following functions have been tested:

$$f_1(x) = \left(1.5x + \sqrt{1 + \left(\frac{3x}{2}\right)^2}\right)^{1/3} \quad (3.137)$$

$$f_2(x) = \frac{2\alpha}{\pi} \arctan(\pi x/2) \quad (3.138)$$

$$f_3(x) = \alpha \tanh\left(\frac{x}{\alpha}\right) \quad (3.139)$$

Also when $n \rightarrow \infty$, the relationship between the value of k and the element nodal values k_j can be re-expressed in terms of the shape functions as follows:

$$k = e^L = \exp\left(\sum_{j=1}^{n_k} L_j \phi_j\right) = \prod_j^{n_k} k_j^{\phi_j} \quad (3.140)$$

Finally, the use of essential boundary conditions for k is adapted to the transformation scheme. As mentioned a little bit earlier, for each Newton-Raphson iteration, not δk but δL is selected as the unknown. Correspondingly, the essential boundary conditions should apply to L after transformation. According to equation (3.116), the essential boundary conditions become for L :

$$nk^{1/n} - n\bar{k}^{1/n} = 0 \quad (3.141)$$

or

$$\log k - \log \bar{k} = 0 \quad (3.142)$$

for $\log k$ method when $n \rightarrow \infty$, with \bar{k} the imposed boundary k value in both cases. Also for the $\log k$ method, \bar{k} should be modified by adding an appropriate small positive value to the actual k if $\bar{k} = 0$.

3.9.2 Numerical experiments and discussions

A summary of the transformation schemes that we have tested on the Czochralski flow problem, whose geometry is shown in Figure (5.1) (chapter 5), is listed in Table (3.4), while the results obtained by these transformation schemes are presented in Table (3.5) with the crystal rotation rate at 20 *rpm*, the crucible rotation at -5 *rpm* and the crystal pulling rate at 1.0 *mm/min*. Please notice that in Table (3.5), u_{min} , u_{max} , v_{min} , v_{max} , ψ_{min} , ψ_{max} , k_{min} , k_{max} , $\mu_{t_{min}}$ and $\mu_{t_{max}}$ denote the minimum and maximum values of the radial and axial velocities, the Stokes stream function, the turbulent kinetic energy, and the turbulent viscosity respectively. One can observe that the solutions provided both by \sqrt{k} and $\log k$ transformation schemes (with m equals to 1 for both two schemes) are very close to the original $k-l$ solution and the generic $k-l$ solution (with both m and n equal to 1). It should be noted that our practical numerical experiments also showed that the original $k-l$ solution will fail when the initial guess values of k are far from the real ones, while both the \sqrt{k} and $\log k$ succeed to converge to the final solution, with however much more iterations needed. On the other hand, both \sqrt{k} and $\log k$ transformation scheme could not reach convergence when m is chosen to be 2 for \sqrt{k} and ∞ for $\log k$, respectively, even when the initial solution is close enough to the real solution.

m	n	Scheme	Test function	Increment approximation
1	1	$L = k$	ψ_i	$k_i^{r+1} = k_i^r + \delta k_i^r$
1	2	$L = 2\sqrt{k}$	ψ_i	$k_i^{r+1} = \left(1 + 2\left(\frac{\delta L_i}{\sqrt{k_i^r}}\right) + \frac{1}{4}\left(\frac{\delta L_i}{\sqrt{k_i^r}}\right)^2\right)k_i^r$
2	2	$L = 2\sqrt{k}$	$\sqrt{k}\psi_i$	$k_i^{r+1} = \left(1 + 2\left(\frac{\delta L_i}{\sqrt{k_i^r}}\right) + \frac{1}{4}\left(\frac{\delta L_i}{\sqrt{k_i^r}}\right)^2\right)k_i^r$
1	∞	$L = \log k$	ψ_i	$f(x) + \sqrt{1 + f^2(x)}$
∞	∞	$L = \log k$	$k\psi_i$	$f(x) + \sqrt{1 + f^2(x)}$

Table 3.4: Summary of the transformation schemes tested

	$k - l$	generic $k - l$	\sqrt{k}	$\log k$
$u_{min}(m/s)$	-0.05178	-0.05290	-0.05409	-0.005285
$u_{max}(m/s)$	0.016004	0.016557	0.016932	0.0150702
$v_{min}(m/s)$	-0.02219	-0.02299	-0.02340	-0.02137
$v_{max}(m/s)$	0.050869	0.052991	0.054388	0.055562
$w_{min}(m/s)$	-0.06	-0.06	-0.06	-0.06
$w_{max}(m/s)$	0.191775	0.192716	0.193080	0.197928
$\psi_{min}(m^3/s)$	-1.8724e-5	-1.8724e-5	-1.9466e-5	-1.5655e-5
$\psi_{max}(m^3/s)$	8.3130e-4	8.3130e-4	8.5100e-4	7.7749e-4
$k_{min}(m^2/s^2)$	0*	0*	0	0
$k_{max}(m^2/s^2)$	0.0018277	0.0018277	0.0018615	0.00181636
$\mu_{t_{min}}(kg/m \cdot s)$	0	0	0	0
$\mu_{t_{max}}(kg/m \cdot s)$	2.4730	2.4730	2.4945	2.4658

Table 3.5: Comparisons of the results obtained for the problem described in Figure (5.1) with different transformation schemes, where m was chosen as 1 both for \sqrt{k} and $\log k$ schemes.

3.10 Conclusions

The present chapter aimed at addressing one of the biggest challenging problems in the Czochralski crystal growth process – namely the melt convection modeling, which also has represented a major objective in bulk crystal growth modeling, and is of primary importance for global heat and mass transfer in the crystal growth process.

We first described the origin of the silicon melt flow in the Czochralski growth system. Then, the modeling efforts in the past three decades to understand the mechanisms governing Czochralski melt convection were reviewed. Next, the mathematical model and associated boundary conditions governing the melt flow and thermal field were introduced, and the Finite Element numerical method to solve the Navier-Stokes equation and the energy equation governing the melt flow were presented as well. Our finite element quasi-steady Navier-Stokes solver based on unstructured meshes was validated by solving the well-known 2D lid-driven cavity flow problem at different Reynolds numbers up to 25,000 with other highly accurate benchmark solutions.

The rest of this chapter focused on turbulence melt convection modeling in the Czochralski crystal growth process. Our main objective was to develop relatively simple turbulence models, but with reasonable accurateness in order to simulate efficiently the average axisymmetric flow in the Czochralski melt as

coupled to the whole quasi-steady or time-dependent growth problem. With this objective in mind, we first derived the evolution of the mean and fluctuating velocity fields by means of the Reynolds decomposition. Some turbulence dynamics aspects, the eddy viscosity/conductivity concepts and the models required to close the system of equations governing the mean quantities were reviewed and introduced. Then we have detailed the turbulence mixing-length and $k-l$ models and the numerical method developed at the CESAME crystal growth research group in the past decades, where the mixing-length l is approximated by a smooth and continuous distance to the boundary function governed by double Laplace equations. Based on the Laplace equations governing the distance to the boundary, an enhanced model taking into account different turbulent boundary layers along the melt/crucible and melt/crystal interfaces was presented, and the modeling parameters governing the distance profiles to the boundary in the silicon melt were primarily given.

It should be noted that when a very small mixing-length boundary layer thickness $\delta_l = P/C$ is chosen, convergence problems might occur when solving the mixing-length governing equation (3.92). Even when convergence is reached, it might remain very difficult to solve the melt convection problem, since the melt convection boundary layer could not be well captured by using not so highly refined isotropic meshes. Therefore the melt convection boundary layer is underresolved in practice. However, the enhanced mixing-length model still provides an ideal tool to fit the convection results obtained by numerical simulation with experimental data. To solve the above mentioned numerical problem, highly refined anisotropic meshes near the wall boundaries would be needed, and appropriate wall models to accurately capture the melt convection boundary layer are mandatory.

On the other hand, the coupling of an additional nonlinear equation to the Reynolds equations increases dramatically the numerical difficulties encountered when trying to simulate the melt flow by the turbulence $k-l$ model. These difficulties either come from the hyperbolic nature of the nonlinear transport terms or from the square-root of k present in the turbulent viscosity expression in the Reynolds equations and turbulent kinetic energy equation. Both of these difficulties, particularly the negative k issue will let fail the Newton-Raphson iterative scheme (or dramatically slow down it if negative values of k are set to zero). Therefore a generic transformation method to overcome these difficulties was proposed and implemented, and our primary numerical experiments have showed that the negative k problem was well solved by the specific transformation schemes we have chosen. Another big advantage of our generic transformation method is that we can choose different transformation constants

to implement different complex transformation schemes without changing any code implementation.

Chapter 4

Modeling of Gas Convection and Oxygen Transport in Czochralski Silicon Crystal Growth Process

4.1 Introduction

During the Czochralski growth process, the gas inside the growth chamber has to be evacuated after loading a precise amount of polycrystal charge. The growth chamber is then back-filled with an inert gas to inhibit the entrance of atmosphere gas into the melt during the growth process. The inert gas in the growth chamber plays important roles in nowadays crystal growth processes. The primary role of the inert gas is to protect the molten silicon melt against oxidation. Another important role of the inert gas consists in evacuating the evaporated oxygen away from the melt free surface by convection, hence affecting the oxygen distribution in the melt and its concentration in the grown crystal. Moreover, the inert gas convection also exerts a shear stress on the melt free surface, therefore driving the melt flow from the crystal/melt/gas tri-junction to the crucible wall. This shear stress due to gas convection usually has the opposite direction than the buoyancy and Marangoni forces. Finally, in modern crystal growth furnaces, “gas flow control” is widely used to adjust the gas flow pattern near the melt free surface, and hence to increase the gas-phase mass transfer coefficient as well as the gas-driven thermal effects.

On the other hand, oxygen, as one of the most important impurities in the Czochralski silicon growth process, strongly affects the final quality of the single crystal. Oxygen precipitates in the bulk of the wafer are used to act as gettering centers for metal impurities and the dissolved interstitial oxygen also hardens the silicon lattice [(Falster and Bergholz 1990)]. However, too high density of oxygen precipitates will warp the silicon wafer and degrade the characteristics of the electronic devices [(Moerschel et al. 1977)]. Therefore, a convenient oxygen concentration is needed to increase the mechanical properties of the silicon wafer [(Hu and Pateick 1975)]. It is found that an oxygen concentration in the

range of $5 \times 10^{17} \sim 8 \times 10^{17} (\text{atom}/\text{cm}^3)$ in the silicon wafer is beneficial for the production of nowadays large-scale integrated circuits [(Mühe et al. 1999)].

The oxygen in the Czochralski silicon crystal originates from the dissolved silica crucible. During the growth process, a large amount of oxygen dissolves from the silica crucible into the silicon melt and is transported to the core of the melt by diffusion due to a violent concentration gradient across a diffusion layer [(Hoshikawa and Huang 2000)]. Then, the oxygen is transported to the melt free surface and melt/crystal interface by diffusion and convection. Most of the dissolved oxygen (more than 99%) evaporates in the form of SiO to the gas phase at the melt free surface and the rest is incorporated from the silicon melt into the crystal at the melt/crystal interface by segregation [(Hoshikawa et al. 1981)]. Some of these processes are illustrated in Figure (4.1).

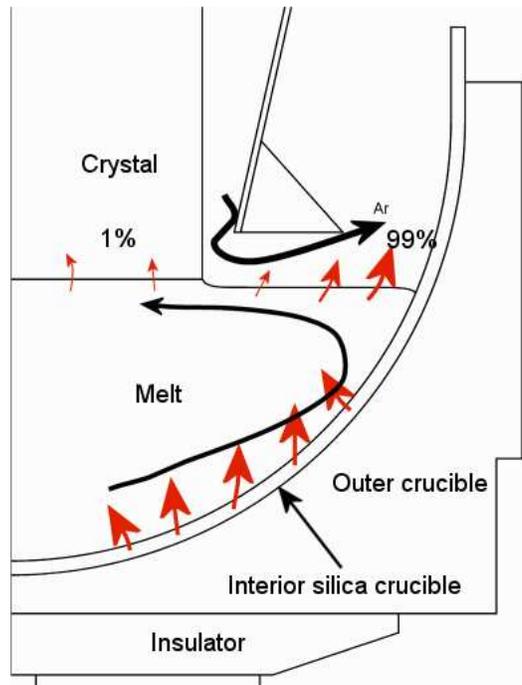


Figure 4.1: Oxygen transport process.

Numerical simulation of crystal growth becomes an indispensable tool to understand, predict and optimize the crystal growth process. During the last three decades, much effort has been made to develop and verify mathemati-

cal models describing global heat transfer, melt flow, impurity transport and defect evolution in the silicon crystal. However, gas convection in the furnace chamber has often been neglected due to the very complicated gas chamber structure (with associated mesh generation difficulties) and an expensive gas flow computational cost. With the development of computer power and computational methods, recently more and more attention is being focused on the effects of gas convection on the global heat transfer, melt convection and oxygen concentration during the crystal growth process. Machida and co-workers [(Machida et al. 1998)][(Machida et al. 2000)] examined the effects of argon gas flow rate, furnace pressure and magnetic field pattern on the oxygen concentration in Czochralski silicon crystal growth through experiments, and these experiments revealed that the gas flow pressure and the inlet gas flow rate have significant effects on the oxygen concentration in the crystal. Kalaev and co-workers [(Kalaev et al. 2003)] presented a model of global heat transfer in Czochralski systems for the growth of silicon crystals taking the inert gas flow into account by a laminar model. Their numerical results showed that at low crucible and/or crystal rotation rates, the gas flow is insufficient to generate a separated flow cell in the melt and only slightly damps out the melt flow driven by buoyancy and Marangoni forces. However, at high crucible and/or crystal rotation rates, the melt flow becomes much more sensitive to shear stresses and a secondary clockwise melt flow cell, which is opposite to the natural melt convection cell, probably appears beneath the melt/gas interface.

Concerning the modeling and numerical simulation of oxygen transport in silicon growth, it is well known that oxygen transport in the melt is dominated by convection and diffusion, and therefore that any physical phenomena and material properties as well as operating conditions that affect melt convection and diffusion will change the oxygen distribution both in the melt and the crystal. In recent years, several experiments and numerical analyses have been conducted on oxygen transport in the silicon melt. For example, A. Mühe et al [(Mühe et al. 1999)] used an electrochemical oxygen sensor to measure the oxygen distribution in production-scale Czochralski silicon melts and, comparing the results with numerical simulations, found that the modeling assumptions in the literature are not adequate to predict quantitative results. The oxygen transport and its distribution both in the melt and the final crystal are closely related to gas convection. Let us here recall the work of Machida et al [(Machida et al. 1998)][(Machida et al. 2000)], Xu [(Xu 1999)], Watanabe et al [(Watanabe et al. 2002)], Kakimoto et al [(Kakimoto et al. 1996)][(Kakimoto et al. 2002)], Sim et al [(Sim et al. 2005)] and Li et al [(Li, Li, Imaishi, Akiyama and Tsukada 2004)][(Li, Imaishi, Akiyama, Peng, Wu and Tsukada 2004)], who

investigated the effects of Marangoni convection, magnetic fields and gas convection on oxygen concentration in the Czochralski crystal growth system.

In this chapter, we will first present the laminar and mixing-length turbulence models for gas convection and gas heat transfer, and then introduce the modeling of oxygen transport in the silicon melt taking the turbulent melt flow into account. Finally, we will focus on the ways to capture the shear stress along the melt free surface due to gas convection by means of direct and Lagrange Multipliers methods, and the primary results captured by these two methods will be presented and analyzed. However, the effects of gas convection on the melt flow, the melt/crystal interface shape, and the oxygen distribution as obtained on the basis of FEMAG quasi-steady simulations will be delayed to the next chapter (chapter 5), and will not be presented and discussed here.

4.2 Modeling of gas convection

4.2.1 Governing equations

Employing the same strategy as for melt convection, the energy equation in the gas domain is solved by applying a decoupled algorithm from gas convection calculations and any other geometrical unknowns (however, the converged solution is of course coupled). The temperature field T is governed by the energy equation as follows:

$$\rho c_p \frac{DT}{Dt} = \nabla \cdot [(k + k_t) \nabla T] \quad (4.1)$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla \quad (4.2)$$

is the substantial time derivative operator, ρ , c_p , k and k_t are the density, specific heat at constant pressure, thermal conductivity and additional turbulent thermal conductivity respectively. We always assume that there is no heat source in the gas domain. It should be noted that the thermal conductivity k and specific heat c_p can be temperature dependent. Such as in the melt, k_t is expressed as follows:

$$k_t = \frac{\nu_t}{P_r^t} \quad (4.3)$$

with ν_t the kinetic turbulent viscosity and P_r^t the turbulent Prandtl number, which is generally assumed as being quite close to 1.

Similarly as for melt convection, several basic assumptions for the gas flow have to be firstly introduced as follows: ❶ the argon gas in the furnace chamber is an ideal gas; ❷ the argon gas is a weakly compressible, Newtonian fluid; and

⊕ the argon gas flow is weakly turbulent. Momentum conservation, which expresses Newton's second law, is governed by the equation

$$\rho \frac{D\mathbf{U}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} \quad (4.4)$$

where \mathbf{U} denotes the velocity field, ρ is the fluid density, $\boldsymbol{\sigma}$ the stress tensor and \mathbf{f} the body force acting on the fluid. Assuming a zero bulk viscosity by Stokes' law, the constitutive equation of the stress tensor for a viscous, variable-density and Newtonian fluid is given by

$$\sigma_{ij} = -p\delta_{ij} + (\mu + \mu_t)(2d_{ij} - \frac{2}{3}\delta_{ij}\frac{\partial U_k}{\partial x_k}) \quad (4.5)$$

where

$$d_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \quad (4.6)$$

is the deformation rate tensor, p the pressure, δ_{ij} Kronecker's delta, μ the dynamic shear viscosity and μ_t the additional turbulent dynamic shear viscosity. For the convection in a Czochralski furnace, the gas flow is only subject to the buoyant force given by:

$$f_i = \rho' g_i \quad (4.7)$$

where ρ' is the density fluctuation ($\rho' = \rho - \rho_0$), ρ_0 the gas density at given temperature and pressure, and g_i the gravity acceleration.

The continuity equation, which ensures mass conservation, is given by

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{U} = 0 \quad (4.8)$$

For an ideal gas

$$mP = \rho RT \quad (4.9)$$

and

$$mP_0 = \rho_0 RT_0 \quad (4.10)$$

where m is the gas molar mass, R the ideal gas constant, P and T the pressure and temperature respectively, while P_0 and T_0 denote the inlet gas pressure and temperature.

The pressure P is further decomposed as follows:

$$P = P_0 + p' \quad (4.11)$$

with p' the dynamic pressure due to gas motion. Considering that in the Czochralski furnace, $P \gg p'$, therefore,

$$P = P_0 + p' \simeq P_0 \quad (4.12)$$

and

$$mP_0 = \rho RT = \text{constant} \quad (4.13)$$

for which, it results that

$$\frac{1}{\rho} \frac{D\rho}{Dt} = -\frac{1}{T} \frac{DT}{Dt} = -\frac{1}{T} \mathbf{U} \cdot \nabla T \quad (4.14)$$

in the steady case. Plugging equations (4.9), (4.10), (4.11), (4.12) and (4.13) into the momentum equation (4.4), the continuity equation (4.8), and the energy equation (4.1), and replacing p' by p , the final governing equations are then obtained in tensor form as follows:

$$m \frac{P_0}{RT} \frac{D\mathbf{U}}{Dt} = -\nabla p + \nabla \cdot \left[(\mu + \mu_t) (2\mathbf{d} - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I}) \right] + m \frac{P_0 \mathbf{g}}{R} \left(\frac{1}{T} - \frac{1}{T_0} \right) \quad (4.15)$$

$$\nabla \cdot \mathbf{U} - \frac{1}{T} \mathbf{U} \cdot \nabla T = 0 \quad (4.16)$$

$$m \frac{P_0}{RT} c_p \frac{DT}{Dt} = \nabla \cdot [(k + k_t) \nabla T] \quad (4.17)$$

where, \mathbf{I} is the identity tensor and \mathbf{d} the deformation rate tensor defined in equation (4.6). This is the so-called weakly compressible approximation.

4.2.2 Boundary Conditions

To complete the description of the mathematical model governing the temperature and velocity fields in the gas domain, one must add the conditions that should be applied to the associated boundaries. The same no-slip boundary condition described in the previous chapter for melt convection is also applied to the gas domain solid boundaries, that is:

$$U_i = \bar{U}_i \quad (4.18)$$

More specifically, all normal and tangential velocity components along the gas solid boundaries will be imposed to zero, while the azimuthal velocity w on the crystal/gas interface, internal crucible/gas and external crucible/gas interfaces is imposed to:

$$w = \Omega r \quad (4.19)$$

where Ω is the crystal or crucible rotation rate, and r the radial boundary coordinate. It should be noted that along the melt/gas interface, the order of magnitude of the melt velocity is much smaller than that of the neighbouring gas. Therefore, it is reasonable to impose a no-slip boundary condition with

zero normal and tangential velocities for the gas convection along this interface, while the azimuthal velocity can be approximated by a Couette flow as follows:

$$w = Ar + B/r \quad (4.20)$$

where r is the axial coordinate along the melt/gas interface, and A , B are constants governed by the following equations:

$$\begin{cases} AR_s + B/R_s = \Omega_s R_s \\ AR_c + B/R_c = \Omega_c R_c \end{cases} \quad (4.21)$$

Here R_s and R_c are the radius of crystal and crucible, while Ω_s and Ω_c are the rotation rates of crystal and crucible, respectively. This procedure allows us to decouple gas flow calculations from melt flow calculations in the algorithm.

A free boundary condition can be imposed along the outlet boundary, which means that the three velocity components are free while the extra-stress vector is imposed to vanish. Along the inlet boundary, the radial velocity u and azimuthal velocity w are imposed to 0. However, the axial velocity profile v depends on the furnace design and other operating conditions, such as the diameter of the gas inlet sections, the inlet gas pressure and flow rate, etc.. In order to evaluate this velocity profile, a reasonable assumption is that the inlet gas flow is steady, laminar and fully developed, and hence that the imposed velocity profile along the inlet section is given by the following analytical solution (see Figure 4.2 left):

$$v(r) = -\frac{2Q}{\pi} \frac{[1 - (r/R)^2]}{R^2} \quad (4.22)$$

where Q is the inlet gas flow rate, R the gas inlet section radius, and r the axial distance to the furnace symmetry axis. However, in an industry furnace, there always exists a thin pulling/seeding rod. Taking this pulling rod into account, the inlet velocity profile along the inlet gate is governed by the following analytical solution (see Figure 4.2 right):

$$v(r) = -\frac{2Q}{\pi} \frac{r^2 - a \ln r + b}{\left[r^4 + 2br^2 + ar^2(1 - 2 \ln r) \right]_{R_{min}}^{R_{max}}} \quad (4.23)$$

where

$$a = \frac{R_{max}^2 - R_{min}^2}{\ln R_{max} - \ln R_{min}} \quad (4.24)$$

$$b = \frac{R_{max}^2 \ln R_{min} - R_{min}^2 \ln R_{max}}{\ln R_{max} - \ln R_{min}} \quad (4.25)$$

while R_{min} and $R_{max} = R + R_{min}$ are the radii of the pulling rod and the inlet section respectively.

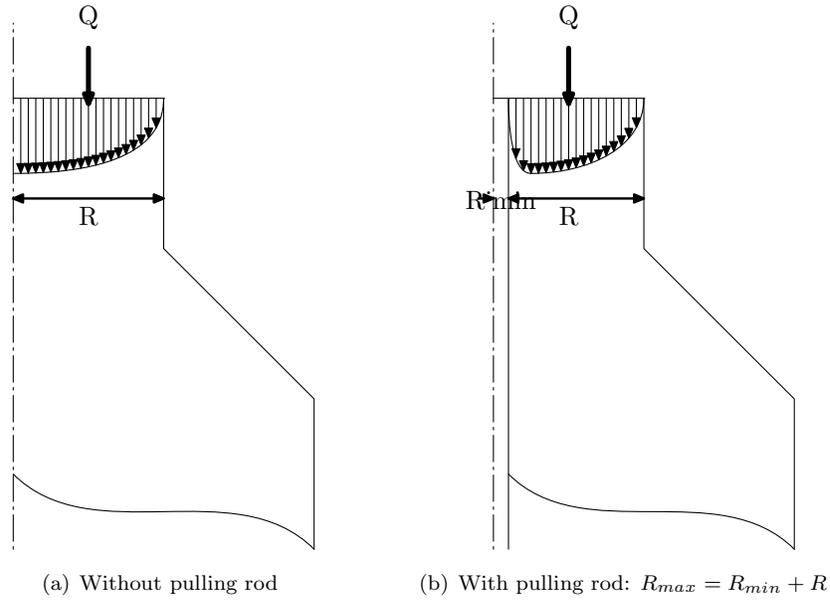


Figure 4.2: Inlet velocity profile

4.2.3 Turbulence models for gas convection

In modern Czochralski crystal growth furnaces, a “gas flow control” device or “heat shield” is widely used to control the radiation heat transfer to the crystal, and also the gas flow velocity near the melt free surface, hence changing the melt flow pattern, heat transfer and oxygen evaporation. However, due to this apparatus, the local Reynolds number in the gas phase near the melt free surface is much higher than in any other region of the furnace chamber. Therefore a laminar model is no longer suitable to predict the gas flow in the vicinity of this region, and a turbulence model has to be used. Considering the fact that the mesh size for the gas chamber is quite large, only the mixing-length model developed in the previous chapter for melt convection has been used to predict the gas flow at a reasonable computational cost. Also, since the gas domain for each quasi-steady simulation is fixed, the mixing-length l_m can be evaluated a priori.

4.3 Modeling of oxygen transport

Oxygen transport in the silicon melt is governed by the following equation:

$$\frac{DC}{Dt} = \nabla \cdot \left((D_0 + D_t) \nabla C \right) \quad (4.26)$$

where D_0 is the oxygen diffusion coefficient in the silicon melt, D_t the additional diffusion coefficient due to turbulence, which can be described as follows:

$$D_t = \frac{\nu_t}{S_c^t} \quad (4.27)$$

with ν_t is the kinetic turbulent viscosity and S_c^t the turbulent Schmidt number. Recalling that the Schmidt number Sc is defined as:

$$Sc = \frac{\nu}{D_0} \quad (4.28)$$

then the turbulent Schmidt number plays the same role in turbulent species diffusion as the turbulent Prandtl number Pr^t in turbulent heat diffusion. It is important to note that both Sc^t and Pr^t are close to 1 (or at least $O(1)$), while this is generally not the case for Sc and Pr in a liquid.

The equilibrium dissolution oxygen concentration at the crucible wall reported by Hirata et al [(Hirata and Hoshikawa 1990)] is used for the boundary condition as follows:

$$C = 3.99 \times 10^{23} \times \exp^{(-2.0 \times 10^4 / T)} (\text{atom/cm}^3) \quad (4.29)$$

where T is the absolute temperature of the crucible wall.

On the other hand, at the melt/gas interface, a first approximation consists in imposing the oxygen concentration to be 0 since 99% oxygen is evaporated and immediately transported away from the melt/gas interface by gas flow. More accurate conditions taking the adjacent gas flow boundary layer are under investigation.

The boundary condition along the melt/crystal interface is a little bit complex due to the effect of oxygen segregation along this interface. The oxygen segregation coefficient is defined as:

$$k = \frac{C_o^s}{C_o^l} \quad (4.30)$$

where C_o^s and C_o^l are the oxygen concentrations in the crystal and silicon melt near the melt/crystal interface. In addition, the oxygen flux across the melt/crystal interface is governed by the following balance equation:

$$\cos\theta(-v_{pull})C_o^l - D_o^l \frac{\partial C_o^l}{\partial n} = \cos\theta(-v_{pull})C_o^s - D_o^s \frac{\partial C_o^s}{\partial n} \quad (4.31)$$

where v_{pull} is the crystal pull rate, θ the melt/crystal interface angle with respect to the horizontal direction, and \underline{n} the normal to the melt/crystal interface, while D_o^s and D_o^l are the oxygen diffusion coefficients in the silicon crystal and melt. Since $D_o^s \ll D_o^l$ and $C_o^s = kC_o^l$, equation (4.31) can be simplified to:

$$-D_o^l \frac{\partial C_o^l}{\partial n} = \cos\theta(-v_{pull})(k-1)C_o^l \quad (4.32)$$

The segregation coefficient k is an important parameter governing oxygen incorporation along the melt/crystal interface. However, the reported results, which include those calculated from thermodynamic data and those obtained from the BPS theory of Burton et al [(Burton et al. 1958)] or the theory of Yen and Tiller [(Yen and Tiller 1991)][(Yen and Tiller 1992)] are inconsistent and range widely from 0.21 to more than one. In addition, Xu [(Xu 1999)] presented a theoretical model to investigate oxygen segregation in silicon crystal growth and showed that k is smaller than 1 but very close to 1. This coefficient is finally suggested to be 0.8 ± 0.1 by Hoshikawa et al [(Hoshikawa and Huang 2000)] after comparison of the oxygen concentration in the silicon crystal and the oxygen saturation concentration in the silicon melt by means of experiments.

In general, it should be pointed out that the oxygen dissolved from the silica crucible wall diffuses across a thin diffusion layer to the bulk silicon melt due to a very low D_o , and therefore that there exists a strong concentration gradient in the melt near the crucible wall. At the same time, another similar diffusion layer near the melt/crystal solidification front also exists when $k \neq 1$ due to the same reason. Hence an extremely highly refined mesh in the melt along the crucible and the solidification front is required if these boundary layers are to be captured. In particular, capturing the segregation boundary layer when $k \neq 1$ requires a huge computational effort, which is unfortunately completely lost since the real value of k is basically unknown. Indeed, as mentioned earlier, the experimental measurements only show that k is quite close to 1, and probably a little bit lower than 1, and hence that its effect is reduced. In such a case, it is better to approximate the solution by neglecting the diffusion layer near the melt/crystal interface, which is simply obtained by setting $k = 1$ in equation (4.30) and (4.32)[(Hurle 1994)].

Similarly, the evaporation of oxygen from the melt/gas interface certainly will require a high mesh refinement near this interface if any accurate boundary condition is modeled, since turbulent mixing tends to vanish in the normal direction close to the free surface.

As a conclusion, improving our simple oxygen transport model represents a very difficult and challenging task.

4.4 Capture of the wall shear stress using Lagrange Multipliers

On the melt/gas interface or the “meniscus”, the molten silicon experiences a viscous shear stress, which is opposite to the effect of buoyancy and Marangoni forces and drives the melt flow from the crystal/melt/gas tri-junction to the crucible wall. This shear stress is due to the gas velocity gradient near the meniscus and can be expressed as follows:

$$f_w = (\underline{\underline{\tau}} \cdot \underline{n}) \cdot \underline{s} \quad (4.33)$$

where, \underline{n} and \underline{s} are the unit normal and tangent vectors along the meniscus, while $\underline{\underline{\tau}}$ is the viscous traceless extra-stress tensor defined as follows:

$$\underline{\underline{\tau}}^d = \underline{\underline{\tau}} - \frac{2}{3} (tr(\underline{\underline{\tau}})) \underline{\underline{\mathbf{I}}} \quad (4.34)$$

with

$$\underline{\underline{\tau}} = \mu(\nabla \underline{U} + \nabla \underline{U}^T) \quad (4.35)$$

where μ is the gas dynamic viscosity. For the axisymmetric case, the viscous stress tensor can be expressed as:

$$\underline{\underline{\tau}} = \mu \begin{bmatrix} \tau_{rr} & \tau_{rz} & \tau_{rw} \\ \tau_{zr} & \tau_{zz} & \tau_{zw} \\ \tau_{wr} & \tau_{wz} & \tau_{ww} \end{bmatrix} = \mu \begin{bmatrix} 2\frac{\partial u}{\partial r} & \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} & \frac{\partial w}{\partial r} - \frac{w}{r} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} & 2\frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \\ \frac{\partial w}{\partial r} - \frac{w}{r} & \frac{\partial w}{\partial z} & 2\frac{u}{r} \end{bmatrix} \quad (4.36)$$

Hence, the boundary shear stress can be directly calculated from the boundary velocity gradient as follows:

$$\begin{aligned} f_w &= \mu \left(\begin{bmatrix} \tau_{rr} & \tau_{rz} & \tau_{rw} \\ \tau_{zr} & \tau_{zz} & \tau_{zw} \\ \tau_{wr} & \tau_{wz} & \tau_{ww} \end{bmatrix} \cdot \underline{n} \right) \cdot \underline{s} \\ &= \mu [-n_r n_z \tau_{rr} + (n_x^2 - n_y^2) \tau_{rz} + n_r n_z \tau_{zz}] \end{aligned} \quad (4.37)$$

However, another way to capture the boundary shear stress consists in using the theory of Lagrange Multipliers with the objective of avoiding a complete boundary layer mesh resolution in the melt and gas phases. The theory of Lagrange Multipliers applied to the Finite Element Method was initially studied

and implemented in the 1970s to avoid the difficulty of fulfilling essential boundary conditions in elasticity problems [(Babuška 1971)][(Babuška 1973)][(Brezzi 1974)]. Based on the theory of Lagrange Multipliers, a general diffusion model problem governed by a partial differential equation with a Dirichlet-type boundary condition

$$-\Delta C + C = f \quad \text{on } \Omega \quad (4.38)$$

$$C = \bar{C} \quad \text{on } \Gamma_{du} \quad (4.39)$$

can be re-expressed in classical Galerkin formulation by introducing Lagrange Multipliers λ as follows:

$$\int_{\Omega} (\nabla C \cdot \nabla C' + C C') d\Omega = \int_{\Omega} (f C') d\Omega + \int_{\Gamma_{du}} (\lambda C') d\Gamma + \int_{\Gamma_{du}} \lambda' (C - \bar{C}) d\Gamma \quad (4.40)$$

where Γ_{du} is the boundary with a Dirichlet-type condition, $\lambda = \frac{\partial C}{\partial n}$ is the flux along Γ_{du} , and C' and λ' are arbitrary test-functions belonging to the appropriate functional spaces. In this formulation, the test-functions C' are no longer constrained to vanish on the boundary Γ_{du} and hence the essential boundary conditions (4.39) are imposed in a weak sense on Γ_{du} .

The formal equivalence between the strong formulation (4.38) and the weak formulation (4.40) can be established for smooth solutions C and λ using an integration by parts. Applying the theory of Lagrange Multipliers to the Navier-Stokes equation, then λ becomes the viscous stress exerted by the fluid on the boundary Γ_{du} :

$$\underline{\lambda} = \mu(\nabla \underline{U} + \nabla \underline{U}^T) \cdot \underline{n} = \underline{\underline{\tau}} \cdot \underline{n} \quad (4.41)$$

and the wall shear stress experienced by the Dirichlet-type boundary becomes:

$$f_w = \underline{\lambda} \cdot \underline{s} = (\underline{\underline{\tau}} \cdot \underline{n}) \cdot \underline{s} \quad (4.42)$$

Finally, the Navier-Stokes equation with Lagrange Multipliers has to be solved in weak formulation as follows:

$$\begin{aligned} & \int_{\Omega} \left[\rho(\underline{U} \cdot \nabla \underline{U}) \cdot \underline{U}' - p \nabla \cdot \underline{U}' + (\mu(\nabla \underline{U} + \nabla \underline{U}^T) : \nabla \underline{U}') \right] d\Omega \quad (4.43) \\ & = \int_{\Gamma_{du}} \left[(\underline{\lambda} - p \underline{\mathbf{I}}) \cdot \underline{U}' + \lambda' \cdot (\underline{U} - \bar{\underline{U}}) \right] d\Gamma \end{aligned}$$

Mixed boundary conditions combining normal velocity conditions and tangential stress conditions, or vice-versa, can be treated in a similar way by use of Lagrange Multipliers. Additional integrals have to be added to the right-hand side of equation (4.43) if natural stress boundary conditions are imposed on a given part of the boundary.

4.4.1 Discretization and numerical method

As we can see from equation (4.43), for each node with a Dirichlet-type boundary condition, three additional unknowns (the normal components of the viscous stress tensor) are introduced in the Navier-Stokes equation, while the velocity fields of the gas and melt flows are approximated by piecewise-quadratic interpolations in our discretization and numerical method.

It should be recalled that, in our model, the gas flow is pre-calculated by assuming a Couette-like velocity distribution along the melt surface. Therefore, the boundary conditions for gas flow calculation are of essential type on the meniscus. The melt flow is calculated in a second step, using the shear stress provided by the gas flow as boundary condition. This approach is devoted to avoid a useless and expensive coupling between melt and gas flow calculations. Correspondingly, three additional equations expressing the Dirichlet-type boundary condition on the meniscus in a weak sense are established in gas flow calculations. Hence the system represented by equation (4.43) is closed and can be discretized and solved by a classical iterative technique. If 2^{nd} -order piecewise polynomials for all three components are chosen, then the final left-hand side local stiffness matrix with Lagrange Multipliers $\underline{\lambda}$ for each element by Newton-Raphson iterative method will be:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \quad (4.44)$$

where, \mathbf{A} is exactly the same local stiffness matrix (left-hand side) as in equation (3.33), and \mathbf{B} is constructed to make discrete the right-hand of equation (4.43) from the following matrix \mathbf{L} :

$$\mathbf{L} = \begin{bmatrix} \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_1) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_3) d\Gamma \\ \int_{\Gamma_{du}} (\hat{\psi}_2 \hat{\psi}_1) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_2 \hat{\psi}_2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_2 \hat{\psi}_3) d\Gamma \\ \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_1) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_3) d\Gamma \end{bmatrix} \quad (4.45)$$

with the indices referring to the local velocity and Lagrange Multiplier nodes on the element boundary. Therefore, with an appropriate numbering of the

velocity nodes, the matrix \mathbf{B} writes as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.46)$$

If exact nodal velocities are expected to result from gas flow calculations along the meniscus, the same shape functions are to be used for Lagrange Multiplier and velocity discretizations on the boundary. Therefore the following piecewise 2^{nd} -order polynomial set has been chosen in our first numerical experiments on the 1D parent element ($-1 \leq \xi \leq 1$),

$$\begin{cases} \hat{\psi}_1 &= (1 - \xi)(1 - 2\xi) \\ \hat{\psi}_2 &= 4\xi(1 - \xi) \\ \hat{\psi}_3 &= \xi(2\xi - 1) \end{cases} \quad (4.47)$$

However, our numerical experiments have shown that strong oscillations are observed if all three components of $\underline{\lambda}$ are approximated by 2^{nd} -order piecewise polynomials, even for flows at very low Reynolds number. Hence, the way to discretize the Lagrange Multipliers $\underline{\lambda}$ had to be chosen carefully. To avoid the above-mentioned drawback, we decided to discretize the viscous stresses by a so-called mixed approximation as follows:

$$\underline{\lambda}^m = \underline{\lambda}^{n(q)} + \underline{\lambda}^{t(\ell)} \quad (4.48)$$

where $\underline{\lambda}^{n(q)}$ is the normal stress vector, which is approximated by piecewise 2^{nd} -order polynomials, while the tangential stress vector $\underline{\lambda}^{t(\ell)}$ is approximated by piecewise linear polynomials. More specifically,

$$\underline{\lambda}^{n(q)} = \underline{\lambda}_{i-1}^{n(q)} \psi_{i-1} + \underline{\lambda}_i^{n(q)} \psi_i + \underline{\lambda}_{i+1}^{n(q)} \psi_{i+1} \quad (4.49)$$

$$\underline{\lambda}^{t(\ell)} = \underline{\lambda}_{i-1}^{t(\ell)} \phi_{i-1} + \underline{\lambda}_{i+1}^{t(\ell)} \phi_{i+1} \quad (4.50)$$

where, $(\psi_{i-1}, \psi_i, \psi_{i+1})$ and (ϕ_{i-1}, ϕ_{i+1}) denote the quadratic and linear shape functions, respectively, while index (i) refers to the boundary element midside node and indices $(i-1)$ and $(i+1)$ refer to the element extremity nodes.

If we choose the piecewise 2^{nd} -order polynomial set to be:

$$\begin{cases} \hat{\psi}_{i-1} &= (1 - \xi)(1 - 2\xi) \\ \hat{\psi}_i &= 4\xi(1 - \xi) \\ \hat{\psi}_{i+1} &= \xi(2\xi - 1) \end{cases} \quad (4.51)$$

on the parent element, and the piecewise linear polynomial set to be:

$$\begin{cases} \hat{\phi}_{i-1} &= (1 - \xi)(1 - 2\xi) \\ \hat{\phi}_{i+1} &= \xi(2\xi - 1) \end{cases} \quad (4.52)$$

on the parent element, then we have:

$$\begin{cases} \hat{\phi}_{i-1} &= \hat{\psi}_{i-1} + \hat{\psi}_i/2 \\ \hat{\phi}_{i+1} &= \hat{\psi}_{i+1} + \hat{\psi}_i/2 \end{cases} \quad (4.53)$$

Also, to define a mixed approximation, the following decomposition was used:

$$\underline{\lambda}^m = \underline{\lambda} - \underline{\lambda}^d \quad (4.54)$$

where, $\underline{\lambda} = (\lambda_r, \lambda_z, \lambda_\theta)$ represents the viscous stress approximated by piecewise 2^{nd} -order polynomials for each component, while $\underline{\lambda}^d$ represents the difference between $\underline{\lambda}$ and the mixed approximation $\underline{\lambda}^m$, which has to be forced to zero in a weak sense. Therefore,

$$\begin{aligned} \underline{\lambda}^d &= \underline{\lambda} - \underline{\lambda}^{n(g)} - \underline{\lambda}^{t(\ell)} \\ &= (\underline{\lambda}_i^{t(\ell)} - \underline{\lambda}_{i-1}^{t(\ell)} - \underline{\lambda}_{i+1}^{t(\ell)})\hat{\psi}_i \\ &= \Delta \underline{\lambda}_i^{t(\ell)} \hat{\psi}_i \end{aligned} \quad (4.55)$$

Based on the above-described mixed discretization method, finally the Navier-Stokes equation can be represented by the following equation in weak form:

$$\begin{aligned} &\int_{\Omega} \left[\rho(\underline{U} \cdot \nabla \underline{U}) \cdot \underline{U}' - p \nabla \cdot \underline{U}' + (\mu(\nabla \underline{U} + \nabla \underline{U}^T) : \nabla \underline{U}') \right] d\Omega \\ &= \int_{\Gamma_{nu}} \left[(\underline{\lambda} - p \underline{\mathbf{n}}) \cdot \underline{U}' + \underline{\lambda}' \cdot (\underline{U} - \bar{\underline{U}}) - \underline{\lambda}^d \cdot \underline{U}' - \underline{\lambda}'^d \cdot (\underline{U} - \bar{\underline{U}}) + \underline{\lambda}^d \cdot \underline{\lambda}'^d \right] d\Gamma \end{aligned} \quad (4.56)$$

where the test-functions $\underline{\lambda}'^d$ are introduced to force $\underline{\lambda}^d$ to vanish in a weak sense. Therefore, the final left-hand side local stiffness matrix with Lagrange Multipliers $\underline{\lambda}$ using mixed approximation for each element by Newton-Raphson iterative method will be:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}' \\ \mathbf{B}'^T & \mathbf{C} \end{bmatrix} \quad (4.57)$$

where, with an appropriate numbering,

$$\mathbf{B}' = \begin{bmatrix} \mathbf{L} - n_y^2 \mathbf{D} & n_x n_y \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ n_x n_y \mathbf{D} & \mathbf{L} - n_x^2 \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} - \mathbf{D} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.58)$$

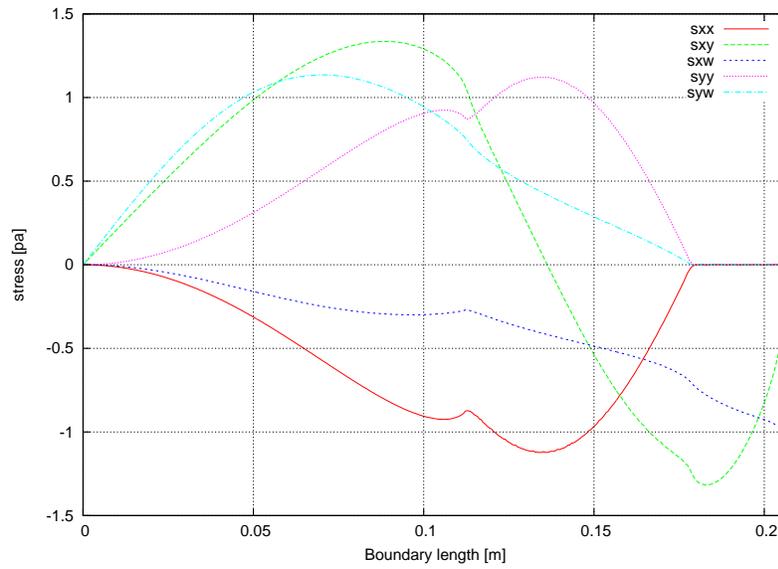
$$\mathbf{C} = \begin{bmatrix} (n_x^2 + n_y^2) n_y^2 \mathbf{E} & -(n_x^2 + n_y^2) n_x n_y \mathbf{E} & \mathbf{0} \\ -(n_x^2 + n_y^2) n_x n_y \mathbf{E} & (n_x^2 + n_y^2) n_x^2 \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix} \quad (4.59)$$

$$\mathbf{E} = \int_{\Gamma_{du}} \psi_2^2 d\Gamma \cdot \begin{bmatrix} 0.25 & -0.50 & 0.25 \\ -0.50 & 1.0 & -0.50 \\ -0.25 & -0.50 & 0.25 \end{bmatrix} \quad (4.60)$$

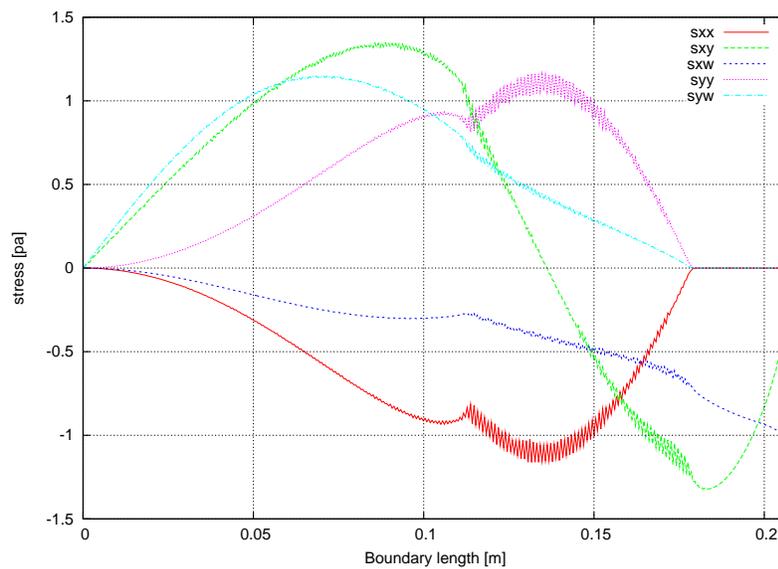
$$\mathbf{D} = \begin{bmatrix} -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_2) d\Gamma & -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_1 \hat{\psi}_2) d\Gamma \\ -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_2^2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_2^2) d\Gamma & -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_2^2) d\Gamma \\ -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_2) d\Gamma & \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_2) d\Gamma & -\frac{1}{2} \int_{\Gamma_{du}} (\hat{\psi}_3 \hat{\psi}_2) d\Gamma \end{bmatrix} \quad (4.61)$$

A large number of simulations were performed with mixed Lagrange Multipliers in order to calculate the shear stress along the melt free surface and melt/crucible interface. From these numerical experiments the following conclusions can be drawn:

- Special care must be given to the Lagrange Multipliers discretization at the interface extremities (possibly located on the axis, the melt/crystal/gas tri-junction and melt/crucible/gas tri-junction points) in order to avoid some side effects as mass balance locking.
- Even when all these side effects are removed from the model the oscillations observed with fully quadratic Lagrange Multipliers are reduced but do not completely disappear when mixed Lagrange Multipliers are used.

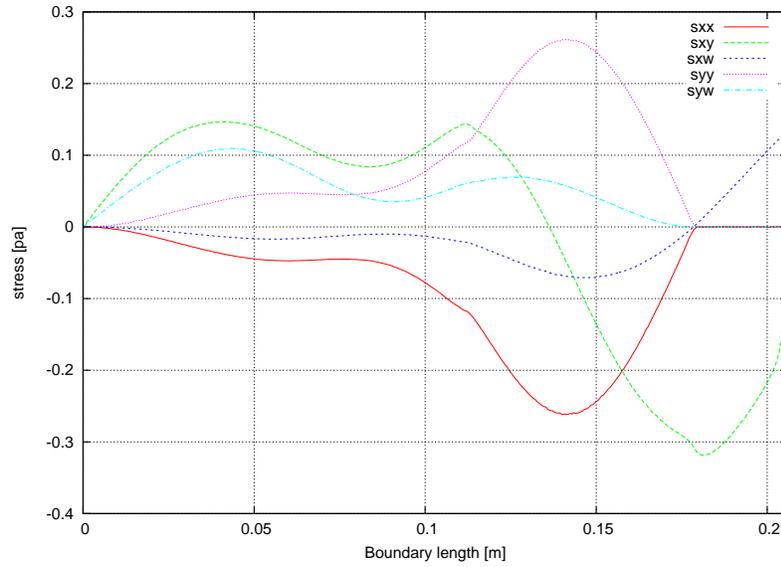


(a) Direct Method

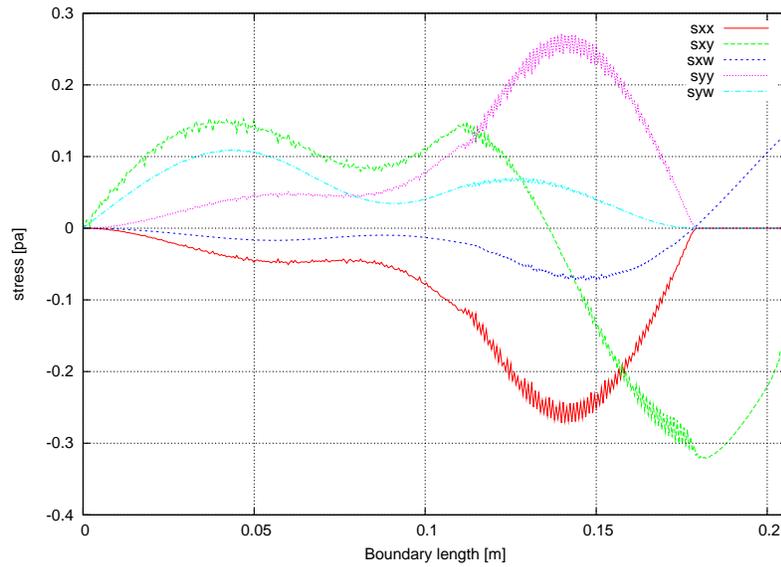


(b) Lagrange Multipliers method

Figure 4.3: Comparison of captured stress tensor along melt/crucible boundary at $\mu = 8.22\text{kg/m.s}$ by Direct Method and Lagrange Multipliers method with 2nd-order approximation.

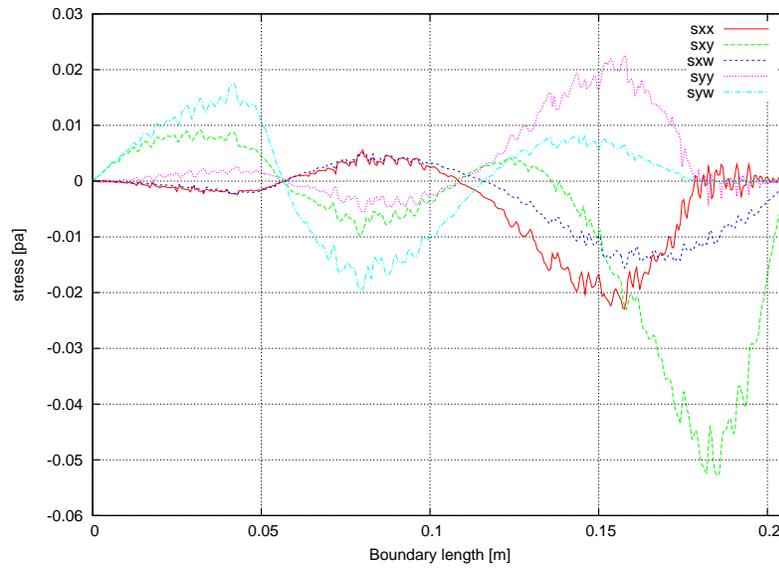


(a) Direct Method

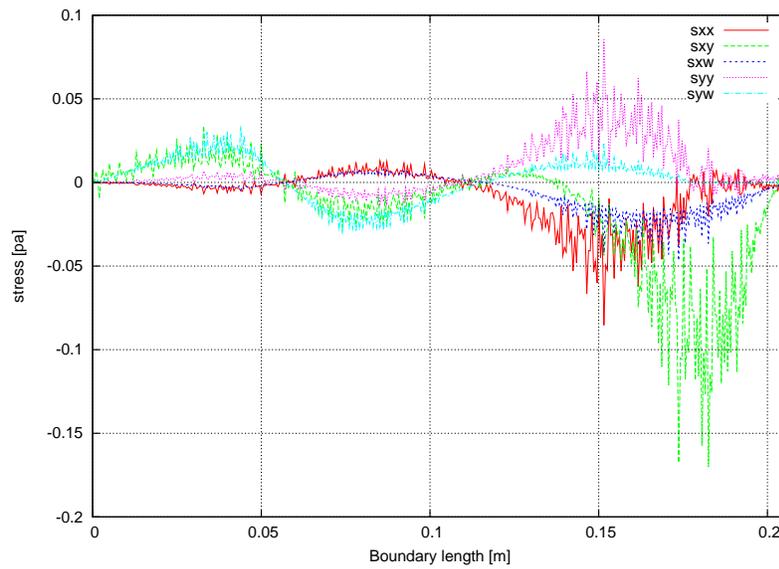


(b) Lagrange Multipliers method

Figure 4.4: Comparison of captured stress tensor along melt/crucible boundary at $\mu = 8.22 \times 10^{-2} \text{ kg/m.s}$ by Direct Method and Lagrange Multipliers method with 2nd-order approximation.



(a) Direct Method



(b) Lagrange Multipliers method

Figure 4.5: Comparison of captured stress tensor along melt/crucible boundary at $\mu = 8.22 \times 10^{-4} \text{ kg/m.s}$ by Direct Method and Lagrange Multipliers method with 2nd-order approximation.

One cause of the above-mentioned problem (but not necessarily the only one) was identified as resulting from an erroneously continuous discretization of the Lagrange Multipliers on the interface. Indeed, considering that the Lagrange Multipliers $\underline{\lambda}$ are expected to approximate the normal viscous stresses $\underline{\underline{\tau}} \cdot \underline{n}$, it can immediately be observed that $\underline{\underline{\tau}}$ is continuous along the interface, whereas this is not the case for the outer normal \underline{n} when rectilinear elements are used to make discrete a curved boundary. Consistency of the numerical scheme requires that the exact analytic solution of the problem satisfies the approximate (discrete) equations, which obviously is impossible with continuous discrete Lagrange Multipliers.

A possible solution to this problem is to use discontinuous Lagrange Multipliers. However, with a quadratic continuous velocity discretization on the boundary, the degree of discontinuous Lagrange Multipliers is maximum one while, as it was already explained, theory shows that the use of a lower degree such as zero for the Lagrange Multipliers should always give better results. In fact this provides a quite poor discretization. Moreover, discontinuous Lagrange Multipliers are not well-adapted to continuous pressure discretization. This is why this approach was not investigated further.

The last approach we investigated in order to calculate interface shear stresses by means of Lagrange Multipliers was to calculate the complete extra stress tensor $\underline{\underline{\tau}}$ (equation (4.35)) on the boundary. Since this tensor is traceless, 3 of its 5 independent components can be calculated in the same way as $\underline{\lambda}$ was, while the 2 remaining components can easily be estimated in a weak sense from the velocity tangential derivatives.

Typical results obtained by this method are shown in Figure (4.3), Figure (4.4) and Figure (4.5), and compared with the results of direct calculations. Clearly, unacceptable oscillations remain present when Lagrange Multipliers are used. However, the theory was not investigated further and we believe that more investigation are needed to draw definite conclusions.

4.5 Conclusions

Gas convection exerts an important effect on the global heat transfer and the distribution of oxygen both in the silicon melt and silicon crystal during modern Czochralski crystal growth processes. Therefore the main purpose of this chapter has been devoted to understand the effect of gas convection on the global heat transfer and oxygen concentration through mathematical and numerical modeling, however without investigating dedicated applications to industrial Czochralski growth systems, which will be covered in the next chapter.

In this chapter, we have firstly summarized the role of the gas flow in nowadays Czochralski crystal growth systems, and the role and origin of oxygen transport in Czochralski silicon crystal growth processes, and then we have reviewed the available experimental observations, modeling tools and numerical experiments on gas convection and oxygen transport in the crystal growth research community. Next, the laminar and turbulent mathematical models governing the gas convection, thermal distribution and oxygen concentration have been developed, and associated boundary conditions for these equations have been introduced. Employing the same strategy as presented in Chapter 3, the Finite Element numerical method on unstructured meshes was used to solve the gas convection and oxygen transport problems, and the velocity and temperature fields were solved in a decoupled way for gas flow calculations. Moreover, the oxygen transport equation is solved only after all the unknowns governing the Czochralski growth system have converged to a final solution.

In order to analyze the effects of the gas flow on silicon melt flow, oxygen distribution and global heater transfer, the rest of this chapter focused on investigating appropriate numerical approaches to capture the wall shear stress exerted by the gas flow and experienced by the silicon melt. First, we have introduced and implemented a direct way to capture the viscous stress tensor along any boundary. Next, we have tried to capture the viscous stress by means of Lagrange Multipliers through varied discretization approaches. The viscous stresses captured by these two methods have been compared and discussed. Our primary numerical experiments have shown that, at a very small Reynolds number, the viscous shear stresses obtained by these two methods are very similar. However, at a medium or large Reynolds number, the Lagrange Multipliers failed to capture the stresses due to strong oscillations of the normal stress. Further investigations are still needed to draw definite conclusions.

Nevertheless, the interest of Lagrange Multipliers remains high since this technique could provide an efficient method to avoid a complete boundary layer resolution near the interface in turbulent flows. Indeed, relative stress variations across the boundary layer remain much more limited than relative velocity variations, because stresses are momentum fluxes and that the momentum which can be communicated to a very thin boundary layer is low. This is way a huge number of numerical experiments were conducted in order to capture the boundary shear stress in the melt along the melt/gas or melt/crystal interface by the Lagrange Multipliers technique. Unfortunately strong oscillations appeared in all cases and the method still cannot be used without important improvements.

Finally, it should be pointed out that we did not conduct numerically investigations on the effect of the gas flow on the way oxygen evaporates at the melt

surface. However this effect is important and further research should be devoted to investigate this effect.

Chapter 5

Industrial Applications of Czochralski Silicon Crystal Growth Process

5.1 Introduction

The FEMAG software has switched from FEMAG-1 to FEMAG-2 generation on the basis of the use of unstructured meshes. Recalling that structured quadrilateral meshes were used in the previous FEMAG generation, all the modules associated with structured meshes had consequently to be adapted to unstructured meshes coordinately. It is necessary and important to validate this adaptation by means of comparisons with literature and available experimental results.

The main purpose of this chapter is devoted to the numerical application of all the developments presented in the previous chapters about the simulation of industrial Czochralski crystal growth processes. First of all, the FEMAG global modeling concepts and techniques will be summarized, since these concepts and techniques as developed by [(Dupret et al. 1986a)][(Dupret et al. 1990)][(Kakimoto et al. 1991)][(Van den Bogaert and Dupret 1997a)][(Van den Bogaert and Dupret 1997b)][(Assaker et al. 1997)] in the past three decades at CESAME form the basis of all our numerical simulations. Then we will focus on the industrial application of these methods to Czochralski silicon growth by means of turbulent mixing-length (or $m-l$), enhanced mixing-length and $k-l$ models without taking gas convection into account. The influences of the modeling parameters on the growth process, and particularly on melt convection, melt heat transfer and melt/crystal interface shape will be presented. Finally, in order to fully analyze the effect of inert argon gas convection on the Czochralski growth process, a series of numerical experiments taking gas convection into account will be performed with varied growth conditions and operating parameters. Comparisons of the simulation results (such as heater power, temperature difference in the silicon melt, etc.) with literature and available experimental observations will also be presented, and the influences of gas convection on the global heat transfer, melt convection, interface position, and

oxygen concentration in the silicon melt for different operating conditions will be detailed and conclusions will be drawn.

5.2 Global modeling and iterative strategy

Numerical simulation of crystal growth processes requires use of a global modeling technique in order to accurately predict the entire thermal behavior of the growth system. The global simulation technique consists in analyzing heat transfer taking all the furnace constituents (or macro-elements) into account. From physical aspects, there are three types of macro-elements in the crystal growth system:

- ❶ **Radiative macro-elements**, which are assumed to be transparent and where the heat transfer is essentially radiative;
- ❷ **Solid macro-elements**, which are one- or two-dimensional, and result from the discretization of solid components such as heater, insulator, crucible, etc., where heat transfer is due to conduction;
- ❸ **Two-dimensional convection macro-elements**, where both the conductive and convective heat transfer should be considered, such as the melt and gas macro-elements. The interface between the different macro-element forms the furnace skeleton.

Global simulations can be either quasi-steady or time-dependent. In quasi-steady simulations, the time dependency of the furnace geometry, and of the heat and mass transfer are not taken into account in the governing equations. Only the release or absorption of latent heat proportionally to the imposed pulling rate and the associated modifications of the melt/crystal interface are taken into account. Although quasi-steady simulations provide cheap and valuable information for any selected specific Czochralski crystal growth stages, however only time-dependent simulations can provide full information on the entire crystal growth process, including poly-crystal melting, seeding, conical growth, shouldering, body growth, tail-end stage and after growth cooling. The major interests of time-dependent simulations are firstly to capture the transient effects during critical growth stages, such as the shouldering or tail-end stages, and secondly, that developing efficient control strategies requires one to analyze the dynamic response of the system to perturbations of the input parameters [(Van den Bogaert and Dupret 1997a)], such as heater power or pull rate. In time-dependent simulations, besides the transient effects of heat

transfer in all the furnace components (including conductive and convective heat transfer in the melt/crystal and gas macro-elements), all the geometrical effects induced by crystal lengthening and crucible lift are also taken into account, together with the heat capacity of every constituent, the motion of the solid-liquid and melt/gas interfaces, etc.. Therefore, time-dependent simulations are not only very complex, but also computationally expensive.

Global simulations also can be performed either in a direct or in an inverse way. In a direct problem, the heater power is imposed as a function of time, while the crystal shape is calculated by means of the crystal radius evolution at the tri-junction. On the contrary, in an inverse problem, a natural process output is given (the crystal diameter history), while the heater power evolution is calculated. In this chapter, only the inverse problem will be considered due to the fact that its modeling is much easier to handle than for a direct problem [(Dupret et al. 1990)]. In the next section, we will focus on our numerical experiments and industrial applications by means of efficient inverse global quasi-steady iterative algorithms. The full description of these algorithms can be found in the above-mentioned corresponding literature and will not be detailed here. However, to better understand the global iterative strategies employed in this chapter, a quick summary is useful.

The Czochralski global quasi-steady algorithm is based on a decoupled, global-local iterative strategy. Typically, a quasi-steady simulation starts by performing a static condensation on all the furnace macro-elements (except the radiative macro-elements). By this way, all the macro-element nodal unknowns are expressed as a function of the macro-element boundary unknowns. The next step consists in assembling the different macro-element contributions, thus leading to a nonlinear system of equations governing the skeleton temperatures and the heater power, which can be iteratively solved by using Newton-Raphson's method. The computed skeleton temperatures and heat fluxes can then be used as thermal boundary conditions for the melt/crystal and gas macro-elements, where the local nonlinear equations governing the convection and heat transfer described in the previous chapters are iteratively solved (in the case of the melt/crystal macro-element, the solid/liquid interface position will also be computed). Iterations are pursued until the local tri-junction temperature matches the melting temperature. When convergence is reached, back-substitution is performed in order to recover the temperature field inside the furnace constituents.

In the case of a dynamic simulation, the procedure is subdivided in two principal steps. First, the melt flow is calculated at several growth stages by means of quasi-steady simulations. Secondly, interpolation between these

quasi-steady results provides the melt flow at every time step of the dynamic simulation. At each time step, calculations are iteratively performed in a similar way as in the quasi-steady case (without however melt flow calculations since flow interpolation is applied in order to reduce the computational cost).

5.3 Description of the industrial Leybold EKZ 1300 Czochralski silicon puller

All our numerical simulations are based on the industrial Leybold EKZ-1300 Czochralski silicon puller for the growth of 100mm diameter crystals. The main reason to select this Czochralski silicon puller is that many numerical and experimental results based on this apparatus can be found from the literature [(Seidl 1995)][(Dornberger et al. 1997)][(Vizman et al. 2000)][(Gräbner et al. 2001)][(Kalaev et al. 2003)]. The global sketch of this puller with a 300mm diameter crucible and 50mm-length single crystal is illustrated in Figure (5.1). The material properties of all the corresponding furnace constituents are given in Table (5.1), and the growth operating parameters used in all our simulations are summarized in Table (5.2), while the other growth parameters not listed in this table are varied in the calculations. The argon properties are presented in Table (5.3) when the inert argon gas convection is taken into account, while the thermal properties of silicon melt and crystal are summarized in Table (5.4) (where the molten silicon dynamic viscosity and thermal conductivity are approximated as constants).

5.4 Global quasi-steady simulations with turbulence mixing-length model on two different meshes

In order to analyze the quality of our global quasi-steady simulation results with turbulence models, a series of numerical experiments on two finite element meshes with different densities (particularly the densities along the melt/crucible, melt/gas and melt/crystal interface), are performed. The coarse mesh in Figure 5.2(a) (left) has 1423 elements (4148 nodes) in the silicon melt, and 496 elements in the solid crystal, while the slightly refined mesh in Figure 5.2(a) (right) has 2752 elements (7744 nodes) and 843 elements in the silicon melt and solid crystal, respectively. The major differences between these two meshes is that the density along the melt/crucible, melt/gas and melt/crystal

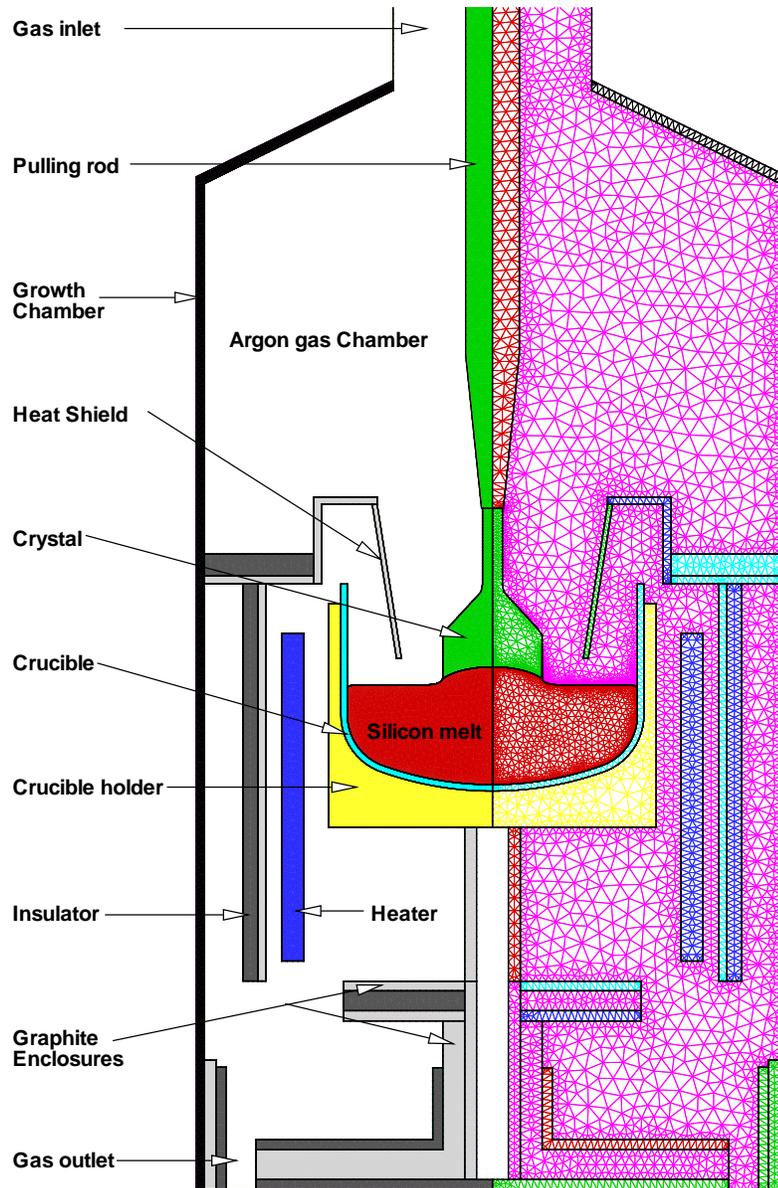


Figure 5.1: Global sketch of the Czochralski silicon crystal 4" puller (Leybold EKZ 1300). Left: the different puller constituents, Right: global finite element mesh

Quartz	
Thermal conductivity	4.0W/mK
Specific heat	1000.0J/(kgK)
Emissivity	0.85
Graphite	
Thermal conductivity	$76.3 \times e^{(-6.695 \times 10^{-4}T)} W/mK$
Specific heat	2100.0J/(kgK)
Emissivity	0.8
Felt	
Thermal conductivity	$7.679e^{-3} + 5.449e^{-5}T - 4.35e^{-11}T^3 W/mK$
Specific heat	1000.0J/(kgK)
Emissivity	0.9
Molybdenum	
Thermal conductivity	10W/mK
Emissivity	0.18
Steel	
Thermal conductivity	15W/mK
Specific heat	470.0J/(kgK)
Emissivity	0.45

Table 5.1: Material properties of the furnace constituents: Steel (enclosure walls, pulling rod), Quartz (crucible), Felt (insulators), Molybdenum (heat shield), Graphite (crucible holder, enclosures, heater, heat shield holder)

Crystal diameter	4"
Crystal rotation rate	20 rpm
Coolant temperature on the wall	300K
Gas reference pressure	15 mbar

Table 5.2: Operating parameters of the growth process

Conductivity	$0.01 + 2.5 \times 10^{-5}T (W/m \cdot K)$
Specific heat	521(J/K · kg)
Dynamic viscosity	$5.0 \times 10^{-5} (kg/m \cdot s)$
Gas constant	8.314(J/K · mol)

Table 5.3: Physical properties of argon gas

interfaces in the refined mesh is nearly tripled with respect to the coarse mesh, although the mesh densities in the silicon melt core are only slightly different in these two meshes. Detailed comparisons of the simulation results based on

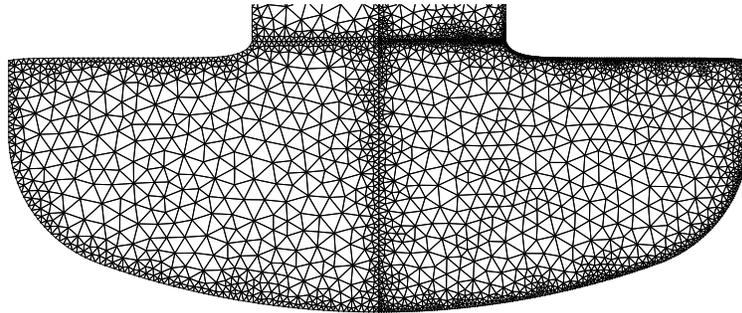
Silicon melt		
Density		$2.530 \times 10^3 (kg/m^3)$
Dynamic viscosity		$8.22 \times 10^{-4} (kg/m \cdot s)$
Thermal expansion coefficient		$1.32 \times 10^{-4} (K^{-1})$
Latent heat of fusion		$4.140 \times 10^3 (J/m^3)$
Melting temperature		$1685 (K)$
Thermal conductivity		$42.9 (W/m \cdot K)$
Specific heat		$942.727 (J/kg \cdot K)$
Emissivity		0.3
Wetting angle		11°
Solid silicon		
Density		$2.34 \times 10^3 (kg/m^3)$
Thermal conductivity		$21.6 (W/m \cdot K)$
Thermal expansion coefficient		$5.20 \times 10^{-6} (K^{-1})$
Emissivity		0.7

Table 5.4: Silicon melt and crystal physical properties

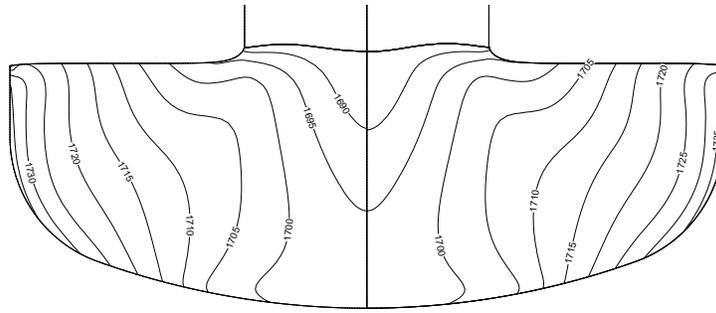
	$T_{max} (K)$	$T_{min} (K)$	$u_{max} (m/s)$	$u_{min} (m/s)$
coarse mesh	1737.73	1109.76	0.02578	-0.01744
refined mesh	1738.65	1109.92	0.04169	-0.02036
	$v_{max} (m/s)$	$v_{min} (m/s)$	$w_{max} (m/s)$	$w_{min} (m/s)$
coarse mesh	0.01537	-0.02583	0.1047	-0.0948
refined mesh	0.02428	-0.0260	0.1047	-0.0960
	$\Psi_{max} (m^3/s)$	$\Psi_{min} (m^3/s)$	$P_{max} (pa)$	$P_{min} (pa)$
coarse mesh	2.11e-5	-2.78e-6	47.3277	-553.6187
refined mesh	2.07e-5	-3.08e-6	40.6181	-562.0627
	$k_{max} (m^2/s^2)$	$\mu_{t_{max}} (kg/ms)$	heater power	deflection
coarse mesh	7.15e-4	0.19731	58.13 KW	1.6700mm
refined mesh	7.84e-4	0.18293	58.04 KW	1.8090mm

Table 5.5: Comparisons of computed results on two meshes with mixing-length model

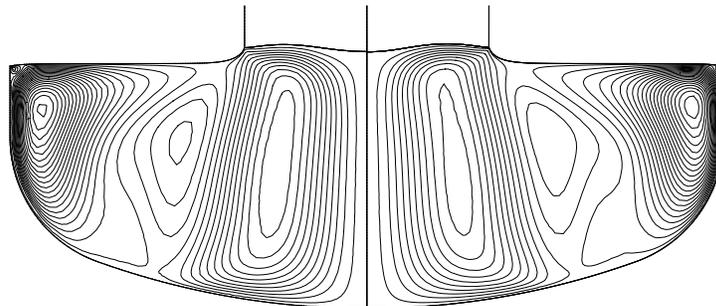
the two different meshes are given in Table (5.5), and the comparison of the temperature profiles, and stream function contour lines in the silicon melt can be found in Figure (5.2(b)) and Figure (5.2(c)), respectively. One can see that the computed melt flow, temperature field and melt/crystal interface deflection (as obtained using the mixing-length model), and the heater power required to grow the prescribed crystal are quite similar for these two different meshes.



(a) Left: coarse mesh / Right: refined mesh



(b) Temperature profiles by steps of 5K, Left: coarse mesh / Right: refined mesh



(c) Stream functions by step of $1.0e-6m^3/s$, Left: coarse mesh / Right: refined mesh

Figure 5.2: Comparison of results on two different meshes. The crystal pulling rate is $1mm/min$, the crystal/crucible rotation rates are $20/-5$ rpm, while gas convection is not taken into account. The mixing-length models is used.

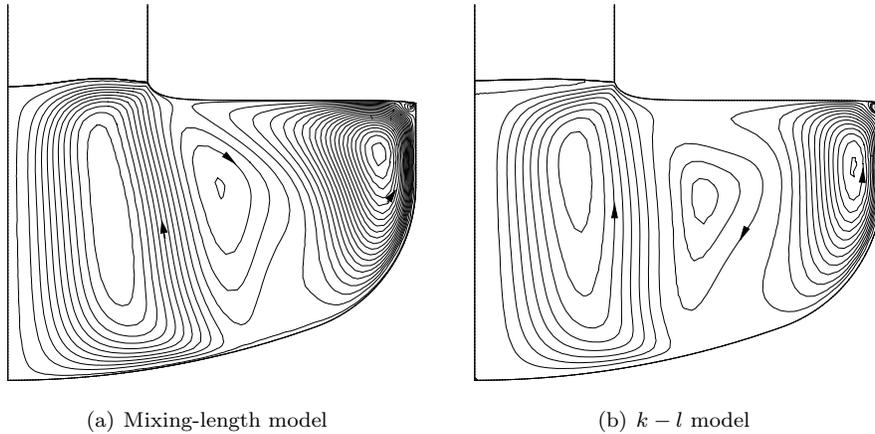


Figure 5.3: Stream function contour lines by steps of $1\text{cm}^3/\text{s}$ obtained by the mixing length and $k-l$ models.

5.5 Global quasi-steady simulation with turbulence mixing-length and $k-l$ models

This section is devoted to comparing the melt convection and temperature results obtained by using the turbulence mixing-length and $k-l$ models developed in the previous chapters. First of all, two global quasi-steady simulations with standard mixing-length and $k-l$ models will be performed, and the results predicted by these two models will be shortly analyzed. Then we will examine the effect of the modeling parameters on the melt flow and melt/crystal interface deflection based on our calculations by means of the enhanced mixing-length model. Finally, three additional calculations will be conducted using the generic $k-l$ transformed method in order to simulate $k-l$, \sqrt{k} and $\log k$ schemes. For all the calculations in this section, the growth conditions and parameters are fixed as follows: the pulling rate is $1\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20\text{rpm}/-5\text{rpm}$. When gas convection is taken into account, the gas flow rate is chosen to be $1500\text{ Standard Litre per Hour (SLH)}$, and the gas reference pressure is fixed at 15mbar .

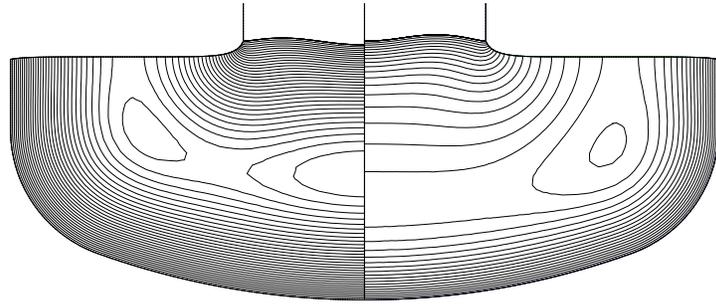
The streamfunction contour lines predicted by the standard mixing-length and $k-l$ models are given in Figure (5.3). The flow pattern in the melt obtained by these two models is quite similar, and is basically constituted by two primary counter-clockwise vortices. One primary vortex is located just beneath the melt/crystal interface due to the rotation of the crystal and the

effect of Coriolis force, while another primary vortex is located under the melt free surface, as due to the combined effects of buoyancy and crucible rotation. These two vortices tend to merge into a single one when increasing the von Karman modeling parameter, as observed by Assaker [(Assaker 1998)] through numerical experiments. Our numerical experiments (as will be seen afterwards) show that decreasing the crucible rotation rate results in the same effect. Furthermore, there are two or three additional secondary clockwise vortices under these specified growth conditions. One of these vortices is located under the melt free surface but close to the melt/gas/crucible tri-junction point (upper right corner in the melt) due to the Marangoni effect, while a second vortex is located between the two primary counter-clockwise vortices. The third vortex, as due to centrifugal force, is only observed with the $k-l$ model just under the melt/crystal interface. Generally speaking, the melt/crystal interface predicted by the $k-l$ model tends to be more concave from the crystal. Finally, the intensities of all the vortices predicted by the mixing-length model are stronger than that of the $k-l$ model due to a lower turbulent viscosity obtained by the mixing-length model.

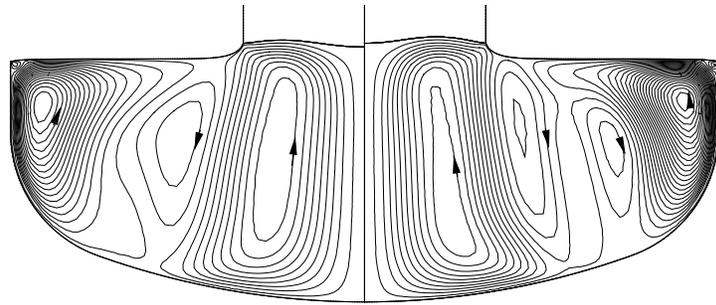
In order to examine the influences of the modeling parameters on the melt flow and the melt/crystal interface deflection, four additional calculations were performed by means of the enhanced mixing-length model without taking gas convection into account. According to equation (3.91) in Chapter 3, the boundary layer thickness of a rotating disk is a linear function of $1/\sqrt{f_d}$, where f_d is the rotation rate of the crystal or the crucible in Hertz. Therefore, the modeling parameter P_s along the melt/crystal interface is chosen to be uniform and equal to half the parameter P_c along the crucible wall, while the parameters C_s and C_c are fixed at 0.41 for all the four calculations.

The contour lines of the distance to the boundary d , the stream function ψ and the turbulent viscosity μ_t with $P_s = 4.1 \times 10^{-1}m$ and $P_s = 4.1 \times 10^{-3}m$ are presented in Figure (5.4). One can observe from Figure 5.4(b) that the flow pattern in the silicon melt core is much more complicated due to the damped turbulent viscosity with much smaller P_s and P_c , and that an additional clockwise vortex is found near the primary vortex due to the rotation of the crystal. Coordinately, the temperature difference in the silicon melt increases from $52.731K$ to $53.90K$ (as compared to $52.73K$ in the reference simulation) as a consequence of the reduced turbulent heat conductivity and damped interaction of the two primary vortices due to crystal/crucible rotational effects and the buoyancy. However the main characteristics of the melt flow near the melt boundaries keep quite the same.

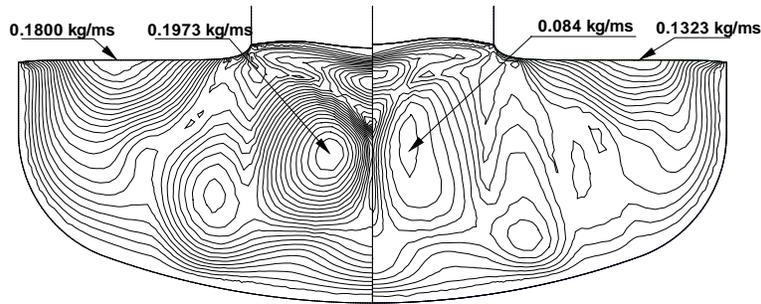
The influence of the modeling parameters on the melt/crystal interface is



(a) Distance to boundary by steps of 0.75mm



(b) Stream functions by steps of $1.2e-6\text{ m}^3/\text{s}$



(c) turbulent viscosity by steps of $8.25e-3\text{ kg/ms}$

Figure 5.4: Use of the enhanced mixing-length model: contour lines of the distance to the boundary, the stream function and the turbulent viscosity with different modeling parameters: $P_c = 2 \times P_s = 8.1 \times 10^{-1}m$ (left), $P_c = 2 \times P_s = 8.1 \times 10^{-3}m$ (right). The crystal pulling rate is $1\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20/ - 5\text{rpm}$.

given in Figure (5.5). Our preliminary numerical experiments show that decreasing P_s and P_c lets the melt/crystal interface deflection become less concave or more convex to the crystal, and also that interface deflection is much more sensitive to these parameters at low P_s and P_c .

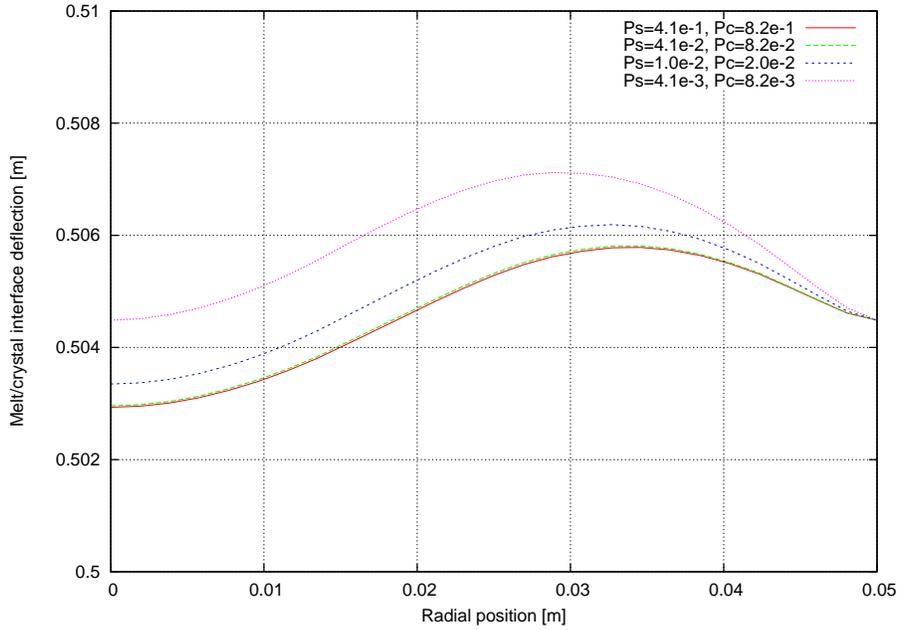
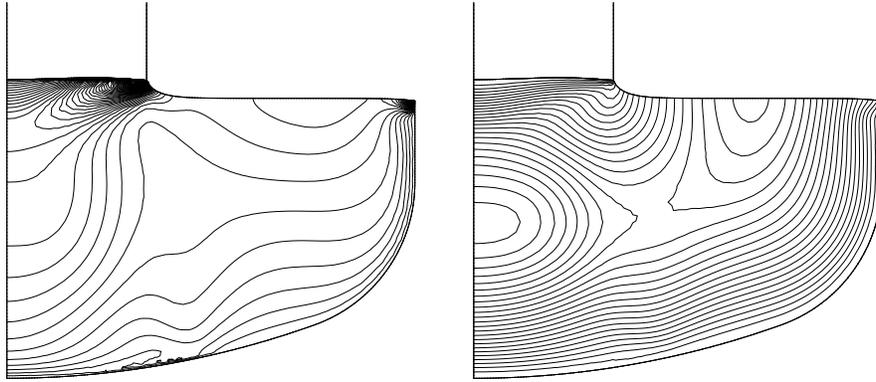


Figure 5.5: Effect of the modeling parameters P_s and P_c (in meters) on the melt/crystal interface shape with enhanced mixing-length model.

It should be observed that the turbulent viscosities obtained by the mixing-length model (Figure 5.4) and the $k-l$ model (Figure 5.6) are quite different, together with the resulting melt/crystal interface deflections. Whereas the effect of the core turbulent viscosity is known to be limited in Czochralski melt flows, the role of the melt/crucible and melt/crystal interface boundary layers appear to be crucial. Since our Lagrange Multipliers technique failed to correctly capture the melt shear stress along these interfaces with an underresolved boundary layer mesh, additional (and very expensive) numerical experiments should be conducted with the appropriate mesh refinement in the boundary layers before drawing definite conclusions on the model selection.

Finally, three additional global quasi-steady simulations were performed based on our generic $k-l$ transformed method in order to compare $k-l$, \sqrt{k} and $\log k$ schemes, and all our calculations with different transformed schemes



(a) Contour lines of turbulent kinetic energy k (b) Contours lines of turbulent viscosity μ_t

Figure 5.6: *Quasi-steady simulation results with log k transformation scheme.* a. Turbulent kinetic energy k by steps of $1.0 \times 10^{-5} m^2/s^2$
 b. Turbulent viscosity μ_t by steps of $1.5 \times 10^{-2} m^2/s^2$.

gave very similar results when compared with the standard $k-l$ model without transformation. The contour lines of the turbulent viscosity μ_t and turbulent kinetic energy with log k scheme are presented in Figure (5.6) (the results obtained by other transformed schemes are not illustrated here due to the very similar results obtained). It should be noted that more Newton-Raphson iterations are needed to reach convergence when \sqrt{k} or log k scheme is chosen, particularly when the initial guess value is too far away from the exact solution, although the negative k problem is well solved by the \sqrt{k} or log k transformation scheme proposed.

5.6 Influence of the pulling rate on the melt and crystal interface

In order to examine the influence of the pulling rate on the melt/crystal interface deflection, four global quasi-steady numerical experiments taking melt convection into account (but without gas convection) are performed with the pulling rate set to $1mm/min$, $1.5mm/min$, $1.8mm/min$ and $2.0mm/min$, while all other operating conditions and parameters are fixed. The interface deflection for these different pulling rates is given in Figure (5.7). One can observe that the interface is more convex to the melt when a lower pulling rate is applied to the crystal. This phenomenon is related to the heat balance

between the solid and the liquid across their interface. Moreover, the temperature difference in the silicon melt for the four given growth rates is $53.65K$, $49.56K$, $45.90K$ and $44.15K$, respectively, which means that the temperature distribution in the silicon melt is more homogeneous with a larger pulling rate, and therefore that a weaker natural convection due to buoyancy and a more convex melt/crystal interface are observed. This conclusion was also confirmed by other experimental observations and numerical simulations. Finally, in practice, in inverse simulations, a larger crystal growth rate always results in a lower heater power to keep the same crystal radius, and hence the expected heater power will be slightly decreased with an increasing pulling rate. Our numerical experiments on the heater power required to grow the prescribed crystal provide $58.12KW$, $57.10KW$, $56.23KW$ and $55.81KW$ for the four given growth rates accordingly. The results also well agree with the measured experimental result from [(Seidl 1995)] with a heater power of $52.0KW$ and a growth rate of $1.5mm/min$.

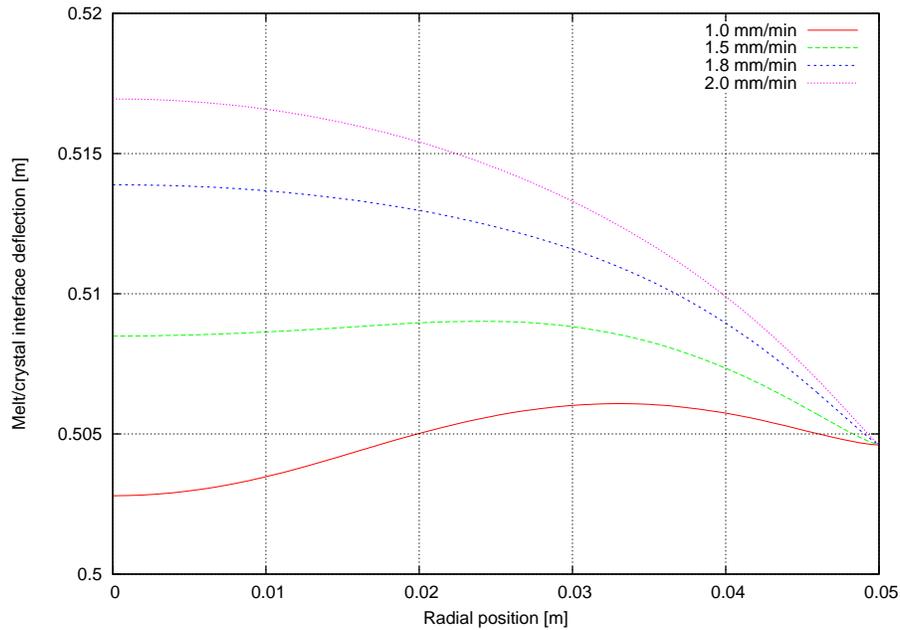


Figure 5.7: Influence of crystal growth rate on the melt and crystal interface shape.

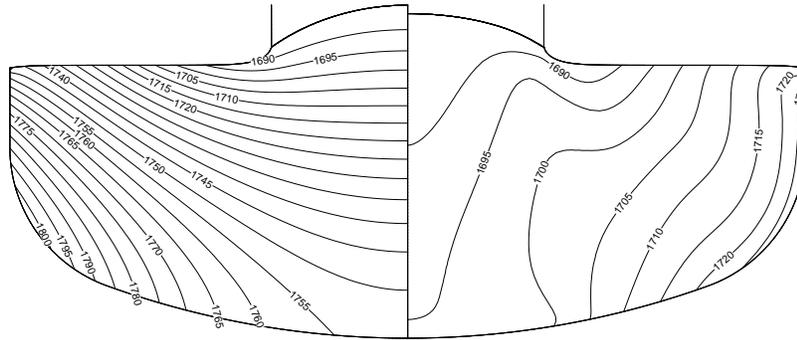
5.7 Influence of convection on the Czochralski crystal growth process

In order to fully analyze the influence of the gas flow on the global heat transfer, melt convection, oxygen distribution, and melt/crystal interface position, a series of calculations are performed by means of FEMAG inverse global quasi-steady algorithm. That is, the temperature field and heater power necessary to grow the prescribed crystal shape are accurately computed taking radiative, conductive and convective heat transfer into account. It should be noted that all our numerical experiments are performed based on the refined mesh, which can be found on the right of Figure 5.2(a), while the mixing-length turbulence model was chosen when gas convection and/or melt convection are taken into account. These series of calculations can be grouped as follows:

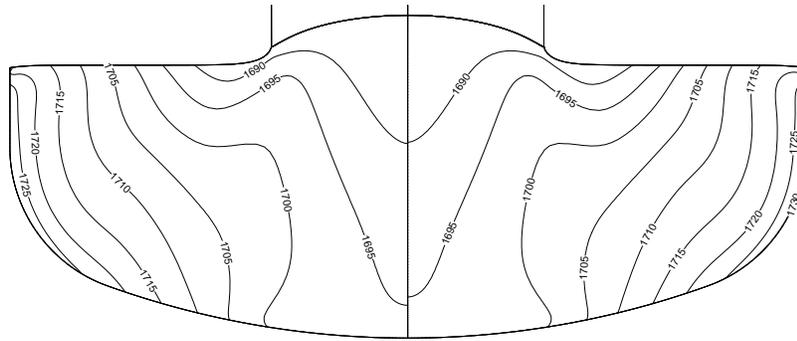
- **Group 1:** Global quasi-steady calculations without gas and melt convection, with melt convection only, and with both uncoupled and coupled gas/melt convection. The main purpose of this group of numerical experiments is to examine the temperature distribution in the silicon melt with or without taking melt convection and gas convection effects into account during the growth process. Comparisons of the melt/crystal interface position and the heater power required to grow the prescribed crystal will also be presented.
- **Group 2:** Global quasi-steady calculations at different gas flow rates with fixed or varied crucible rotation rate, but with all the other operating parameters remaining unchanged. The main purpose of these calculations is to fully examine the gas convection effect on the Czochralski silicon growth process. We are particularly interested in the gas convection effect on the melt flow pattern, the melt/crystal interface deflection and the oxygen distribution in the silicon melt at different gas flow rates and varied crucible rotation rates.

5.7.1 Influence of melt/gas convection on the Czochralski silicon growth process

The main purpose of this section is to examine and discuss the influences of convection on the Czochralski silicon growth process based on the calculated results of group 1. In this group, four numerical experiments are performed: global quasi-steady calculation with pure heat conduction, with melt convection only, and with uncoupled and coupled melt/gas convection. The temperature



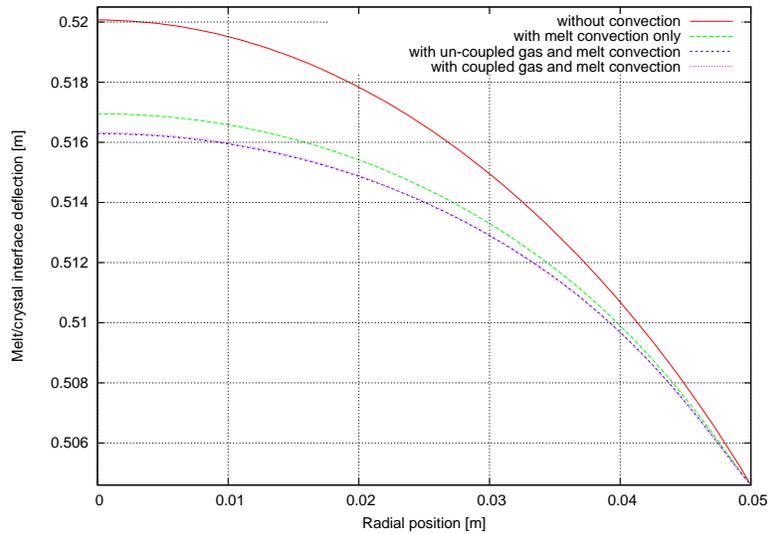
(a) Without convection (left) / with melt convection only (right)



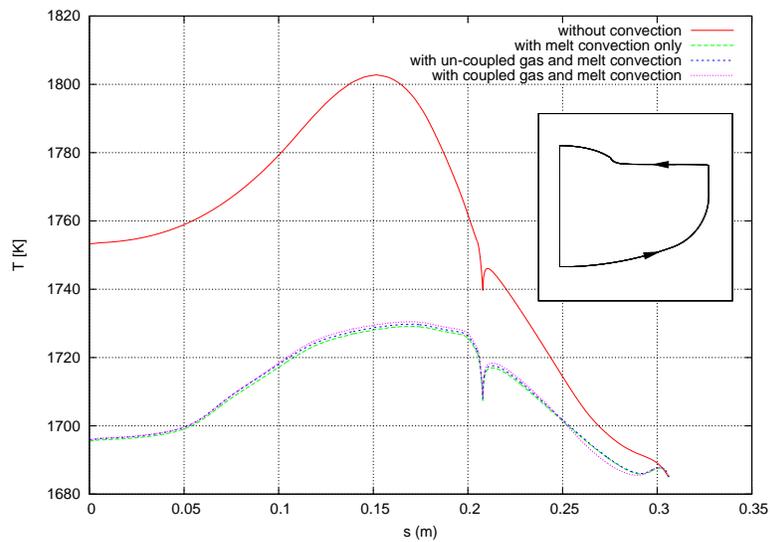
(b) With uncoupled gas & melt convection (left) / with coupled gas & melt convection (right)

Figure 5.8: Simulation group 1. Temperature profiles in the silicon melt by steps of 5K. The crystal pulling rate is 2mm/min, the crystal/crucible rotation rates are 20/-5 rpm, the argon gas pressure and flow rate are 15 mbar and 1500 SLH respectively, and the crystal/crucible rotation rates are 20/ - 5rpm.

profiles of these numerical experiments in the silicon melt are given in Figure (5.8), while the corresponding melt/crystal interface deflection, and the temperature profile along the melt/crucible and melt/gas interfaces are illustrated in Figure (5.9). The calculated temperature distribution in the entire furnace taking into account the coupled silicon melt and argon gas convection is presented in the left part of Figure (5.10), while the argon velocity vectors in the entire gas chamber are plotted on the right. Very similar results can be found in [(Kalaev et al. 2003)].



(a) Interface position



(b) Temperature profile along the melt-crucible and melt-gas interfaces

Figure 5.9: Simulation group 1. Interface position and temperature profile along the melt/crucible and melt/gas interfaces. The crystal pulling rate is $2\text{mm}/\text{min}$, the crystal/crucible rotation rates are $20/ - 5\text{rpm}$, and the argon gas pressure and flow rate are 15 mbar and 1500 SLH , respectively.

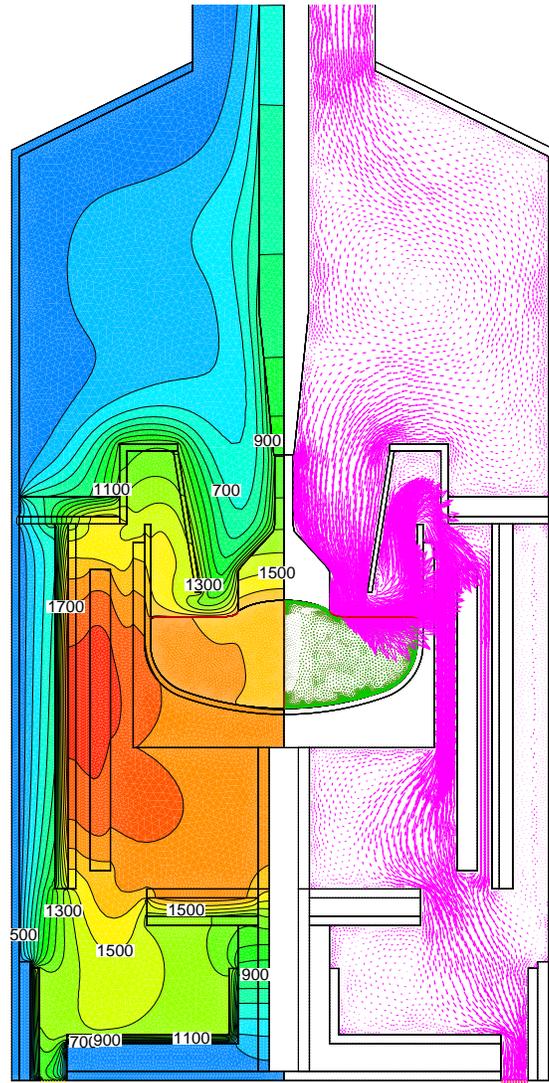


Figure 5.10: Simulation group 1. Temperature distribution and velocity vectors in the entire furnace: pulling rate is $2\text{mm}/\text{min}$, argon gas pressure and flow rate are 15 mbar and 1500 SLH, respectively, the crystal/crucible rotation rates are 20/-5 rpm.

From Figures (5.8) and (5.9), one can see that melt convection has great influence on the temperature distribution in the silicon melt, the melt/crystal interface deflection, and the temperature profile along the melt/crucible and melt/gas interfaces as well. When pure heat conduction is assumed, an over-predicted temperature difference ($\Delta T = 117.78K$) with $T_{max} = 1802.78K$ in the silicon melt is found. In this case, the heater power is also too large ($W = 62.5KW$) due to the over-predicted temperature profile in the silicon melt. When melt convection is taken into account, a much more reasonable and realistic temperature difference ($44.15K$) in the melt is found, which (i) is well agreeing with the numerical result ($45K$) from Gräbner et al [(Gräbner et al. 2001)] as obtained by using the low Reynolds number turbulence model of Chien [(Chien 1982)], and ii) is a little bit lower than the experimental difference ($50K$) reported in [(Seidl 1995)]. When gas convection is taken into account, our numerical experiments show that the temperature differences in the melt with uncoupled and coupled melt/gas convection are $44.78K$ and $45.50K$ respectively, whereas the result of Kalaev et al [(Kalaev et al. 2003)] for the same growth parameters with coupled gas convection was found to be $30K$, which the authors believe as underestimated.

On the other hand, our numerical results for melt/crystal interface deflection with pure heat conduction, with melt convection only, and with uncoupled and coupled melt/gas convection are found to be $15.467mm$, $12.346mm$, $11.681mm$ and $11.707mm$, see Figure (5.9(a)), while the heater power necessary to grow the prescribed crystal shape is $62.5KW$, $55.81KW$, $56.40KW$ and $56.43KW$, respectively, which is in good agreement with the experimentally required power of $62KW$ for this puller[(Dornberger et al. 1997)], but with nearly 20% difference with respect to the simulation result ($67.05KW$) of Kalaev et al [(Kalaev et al. 2003)]. The over-predicted result from Kalaev et al can mainly be attributed to the underestimated temperature difference in the silicon melt, as the authors stated. Therefore, based on the numerical experiments of group 1, it can be concluded that:

- ❶ The quasi-steady global heat transfer model with the mixing-length turbulence model well predicts the temperature difference in the silicon melt, while the pure heat conduction global model does not.
- ❷ For these growth conditions, gas convection has a very limited influence on the temperature difference in the silicon melt, the melt/crystal interface, and the heater power as well, whether the melt convection is, or not, coupled with gas convection.

5.7.2 Influence of gas convection on Czochralski silicon growth process under varied crucible rotation and gas flow rate

The purpose of the numerical experiments of group 2 is to fully examine the various effects of the gas flow rate on the Czochralski silicon growth process at different crucible rotation rates, such as typically the global heat transfer, the melt convection, the oxygen distribution, the melt/crystal interface deflection, etc.. To achieve this goal, a set of numerical experiments are performed with the gas flow rate set at 500 SLH, 1500 SLH, 3000 SLH and 4500 SLH, respectively, and the crucible rotation rate chosen to be -1 rpm, -5 rpm and -10 rpm for each gas flow rate. Moreover, for the gas flow rate of 1500 SLH, three additional simulations are performed with the crucible rotation rate set to -2 rpm, -3 rpm and -7.5 rpm, respectively. All the other operating conditions and growth parameters remain unchanged for all the numerical experiments in this section.

The temperature distributions in the whole Czochralski puller at different flow rates with the crucible rotation rate of -5 rpm are presented in Figure (5.11), and the correspondent stream function contourlines in the convection macro-elements (gas and silicon melt) are given in Figure (5.12). The effects of gas convection on the global heat transfer can be observed in Figure (5.11), where it appears that the temperature gradient in the upper growth chamber is dramatically increased with an increasing gas flow rate. In particular, strong thermal boundary layers are found along the boundary of the pulling rod and the crystal, and along the melt/gas free surfaces. On the other hand, the gas flow pattern, particularly in the upper growth chamber, is quite sensitive to the gas flow rate, which can be observed from Figure (5.12). At a low flow rate, there exists only one primary clockwise vortex in the upper growth chamber, but another counter-clockwise vortex appears near the puller shoulder when the flow rate is increased. When further increasing the flow rate, the primary vortex becomes counter-clockwise, and two secondary clockwise vortices coexist on the left- and right-hand side of the heat shield. However, the major characteristics of the gas flow pattern at the bottom part of the growth chamber remain unchanged, except the intensities of those vortices.

However, the influence of gas convection on the temperature profile along the melt/crucible and melt/gas interfaces is very limited, which can be observed from Figure (5.13) and Table (5.6). When the gas flow rate is increased from 500 SLH to 1500 SLH and 3000 SLH, the maximum temperature along the melt/crucible interface is found to be 1729.6 K, 1730.5 K and 1731.0 K, respectively. When further increasing the flow rate to 4500 SLH, the maximum

	500 SLH	1500 SLH	3000 SLH	4500 SLH
$T_{max}(K)$	1729.6	1730.5	1731.0	1729.7
$T_{min}(K)$	1102.3	1095.8	1090.2	1085.8
ΔT (melt)	44.6	45.5	46.0	44.7

Table 5.6: Comparisons of computed temperature results along melt/crucible and melt/crystal interface with varied argon gas flow rates. The crystal rotation rate is 20 rpm and crucible rotation rate is -5 rpm.

temperature along the melt/crucible interface decreases to 1729.7 K, but the average temperature along the crucible bottom is slightly increased, which can be observed in Figure (5.13(a)) or Figure (5.13(b)). However, the average temperature along the melt/gas interface at different gas flow rates behaves differently and is much more complicated than on the melt/crucible interface, although at a low crucible rotation rate, both the maximum and average temperature along the melt/crucible and melt/gas interface increase with increasing gas flow rate, as shown in Figure (5.22(b)).

The melt/crystal interface for different gas flow rates with the crucible rotation rate at -1 rpm, -5 rpm and -10 rpm is presented in Figure (5.14(a)), Figure (5.14(b)) and Figure (5.15(a)), while a comparison of the melt/crystal interfaces obtained for a gas flow rate at 1500 SLH with varied crucible rotation rate is given in Figure (5.15(b)). One can observe that interface deflection is much more sensitive to the gas flow rate at a higher crucible rotation rate, while for a specified crucible rotation rate, the interface becomes less convex with an increasing gas flow rate, which is consistent with the results in [(Kalaev et al. 2003)]. Generally speaking, the melt/crystal interface deflection decreases with increasing crucible rotation rate. However, when gas convection is taken into account, the melt/crystal interface shape becomes more complex due to the combination of forced convection, buoyancy and gas shear stress, and there is no direct proportionality between the crucible rotation rate and the melt/crystal interface shape, since in particular intersections are found between the melt/crystal interfaces at different crucible rotation rates, see Figure (5.15(b)).

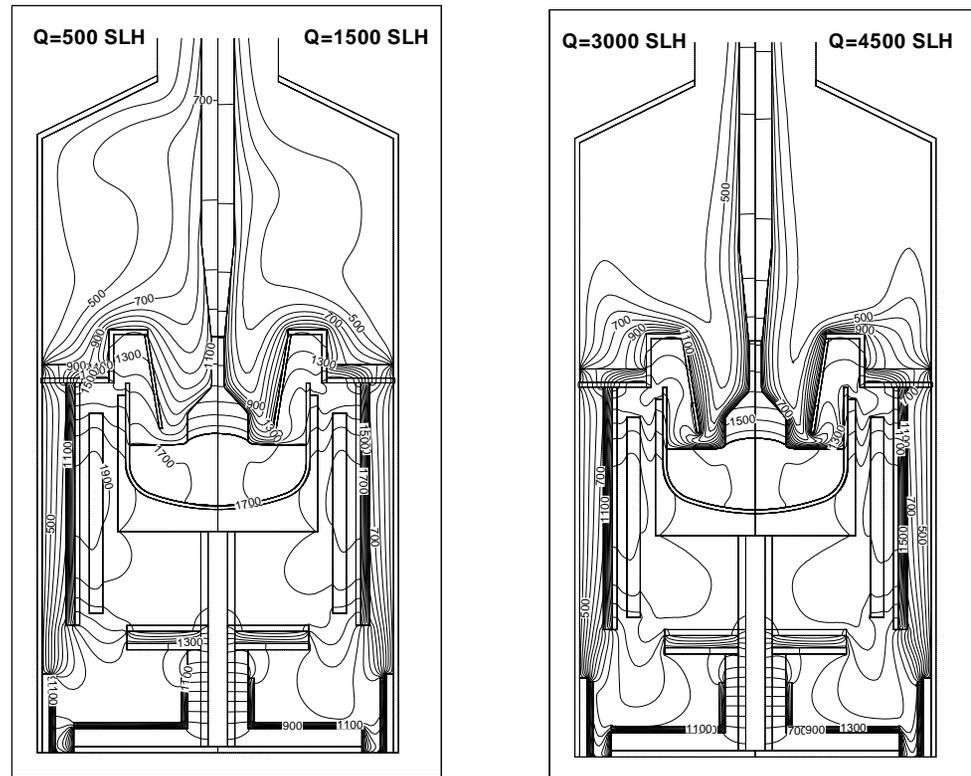


Figure 5.11: Simulation group 2. Temperature distribution by steps of 100K in the entire Czochralski puller: the crystal growth rate is 2mm/min, the argon gas pressure is 15 mbar, and the crystal/crucible rotation rates are 20/-5 rpm.

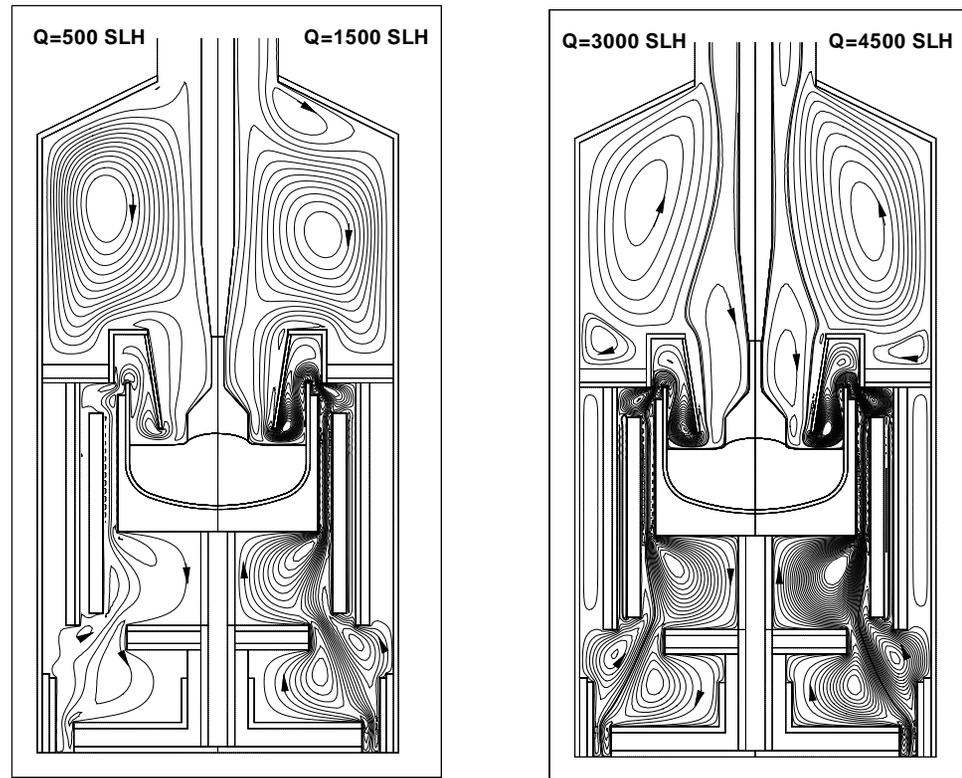
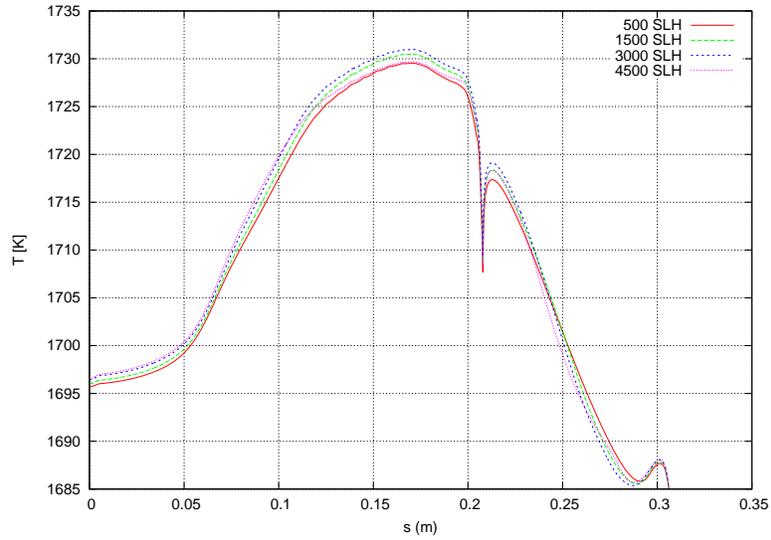
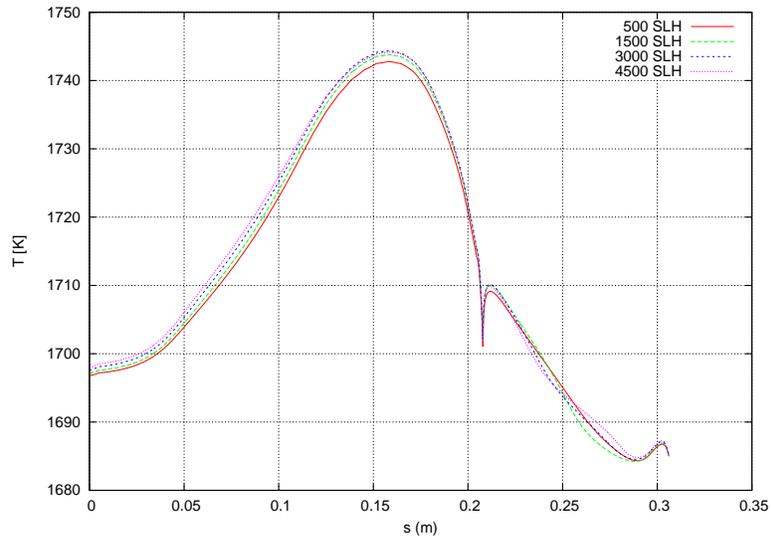


Figure 5.12: Simulation group 2. Stream function contourlines by steps of $0.001\text{m}^3/\text{s}$ (left) and $0.0025\text{m}^3/\text{s}$ (right): the crystal growth rate is $2\text{mm}/\text{min}$, the argon gas pressure is 15 mbar , and the crystal/crucible rotation rates are $20/-5\text{ rpm}$.

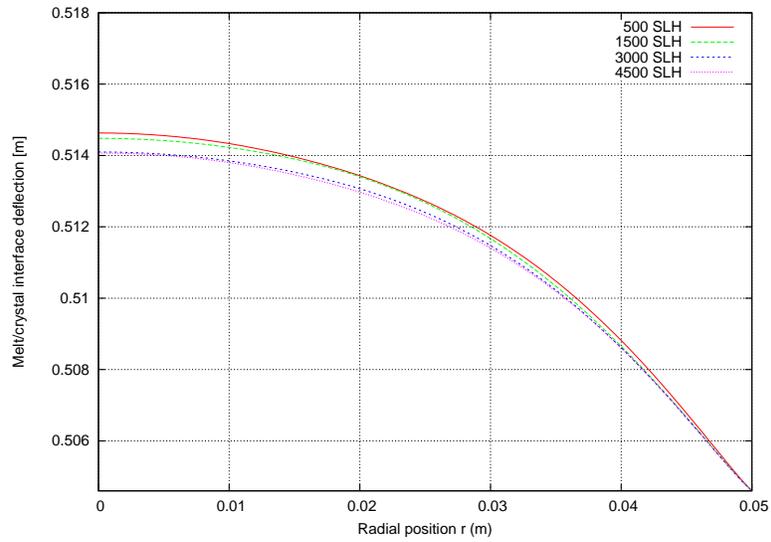


(a) Crucible rotation rate at -5 rpm.

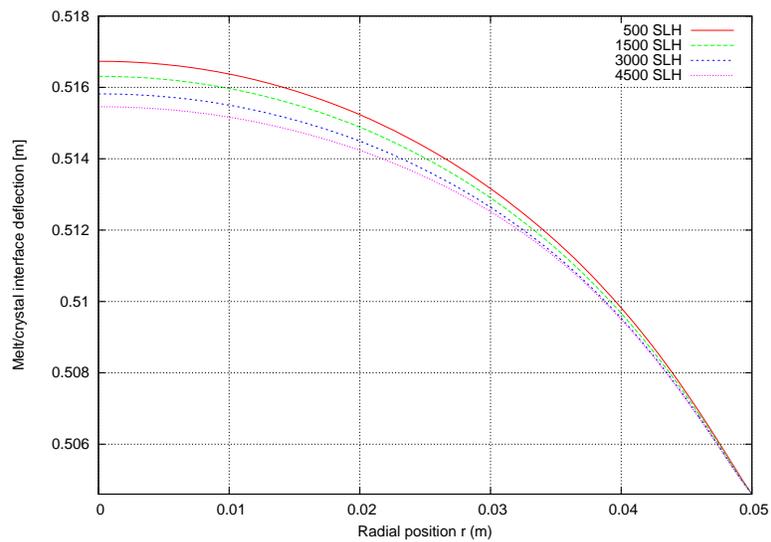


(b) Crucible rotation rate at -10 rpm.

Figure 5.13: Simulation group 2. Temperature profiles along the melt/crucible and melt/gas interfaces at different crucible rotation rates with varied argon gas flow rates. The crystal rotation rate is 20 rpm and crystal pulling rate is 2mm/min.

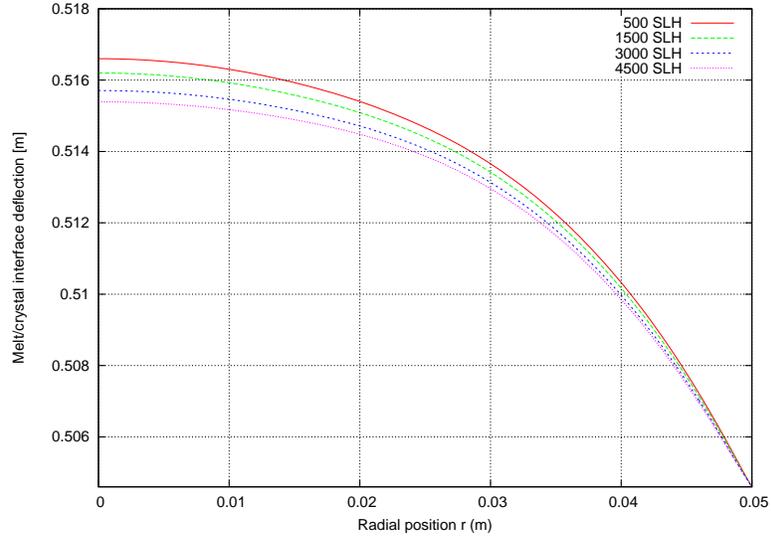


(a) Crucible rotation rate at -1 rpm

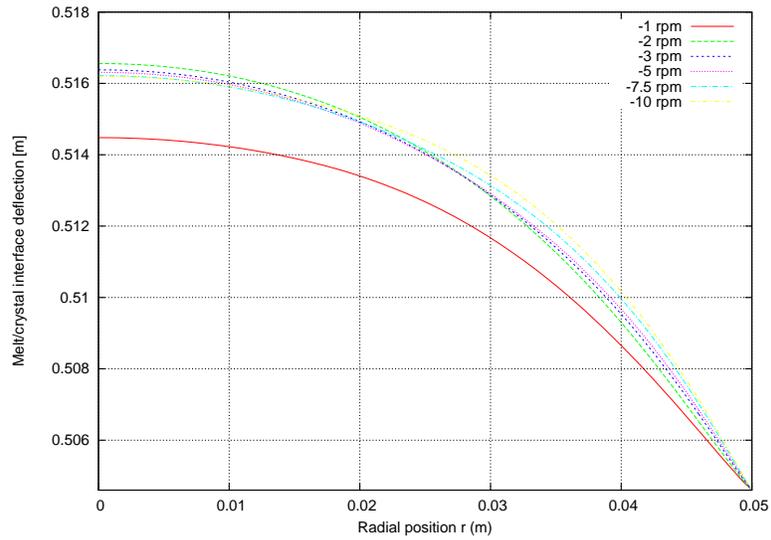


(b) Crucible rotation rate at -5 rpm

Figure 5.14: Melt/crystal interface shape at different gas flow rates and crucible rotation rates. The crystal pulling rate is 2mm/min.



(a) Crucible rotation rate at -10 rpm

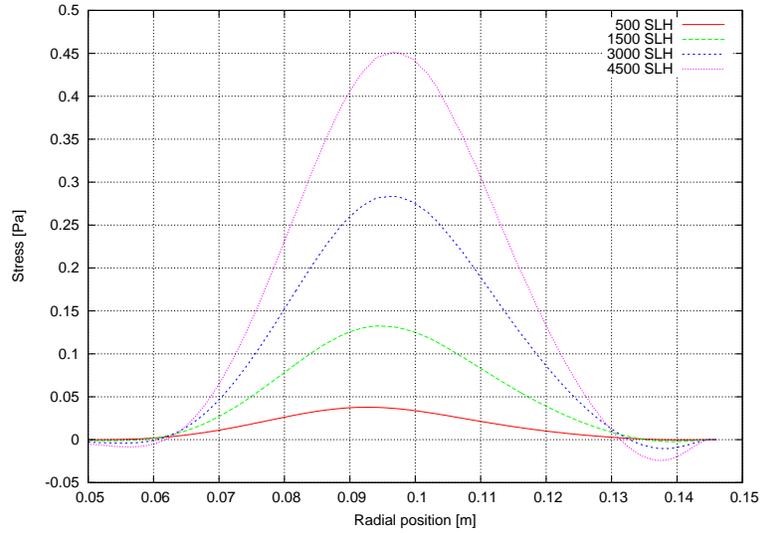


(b) Different crucible rotation rates with the gas flow rate at 1500 SLH

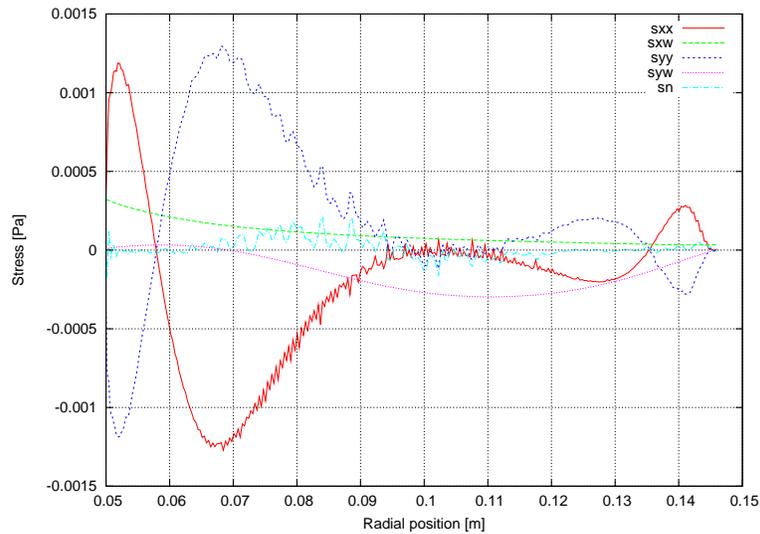
Figure 5.15: Melt/crystal interface shape at different gas flow rates and crucible rotation rates. The crystal pulling rate is 2mm/min.

The forced argon gas flow also exerts a shear stress on the silicon melt, and therefore influences the melt/crystal interface, together with the convection and oxygen distribution in the silicon melt. The captured shear stress along the melt/gas interface at different flow rates is given in Figure (5.16(a)). One can see that the maximum boundary shear stress is exactly beneath the heat shield, where the highest gas velocity gradient is encountered. It should be noted that the location where the maximum boundary shear stress occurs is slightly displaced and closer to the melt/crucible/gas triple point with increased flow rate. Very similar results were obtained by Kalaev et al [(Kalaev et al. 2003)]. However, the results of Kalaev are slightly lower (3.40% at flow rate 1500 SLH, and 10.05% at 3000 SLH) than the results we have captured. One possible reason is that a uniform inlet velocity is assumed in [(Kalaev et al. 2003)], therefore the maximum inlet velocity and finally the maximum velocity gradient beneath the heat shield are hence underestimated. Furthermore, at higher flow rate, negative values of the shear stress due to the clockwise gas motion in the corner near the melt/crucible/gas triple point probably generate another secondary flow pattern in the silicon melt when acting together with the Marangoni stress, particularly when the boundary shear stress is comparable with the value of the Marangoni stress. Finally, let us note that the normal shear stress and other components of the viscous stress tensor along this interface are presented in Figure (5.16(b)), where oscillations are found for the viscous stress tensor component σ_{rr} , σ_{zz} , and the normal shear stress σ_{nn} . The oscillatory nature of the stress tensor with an underresolved mesh can be one of the reasons why we have experienced convergence problems when trying to capture the shear stress by the Lagrange multipliers method.

The effect of gas convection on silicon melt convection at different flow rates with varied crucible rotation are given in Figures (5.17), (5.18) and (5.19). One can observe from these figures that at low flow rate (for example at 500 SLH and 1500 SLH), gas convection has very limited influence on the melt flow, since only the vortex beneath the melt/gas interface driven by the buoyancy and Marangoni forces is slightly damped. However, when further increasing the flow rate (for example at 3000 SLH), a separated clockwise vortex due to the gas shear stress occurs, and the intensity of this newly generated vortex increases with the gas flow rate. At a very high flow rate, for example at 4500 SLH, not only the vortex due to the buoyancy and Marangoni effects, but also the secondary vortex in the melt core (between the primary vortex due to crystal rotation and the vortex due to buoyancy and Marangoni effect) is damped, see Figure (5.18(d)) or Figure (5.19(d)).



(a) Boundary shear stress due to gas flow



(b) Components of stress tensor (flow rate at 1500 SLH)

Figure 5.16: Shear stress and stress tensor components along the melt/gas interface due to gas convection. The crystal pulling rate is $2\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20/ - 5\text{rpm}$.

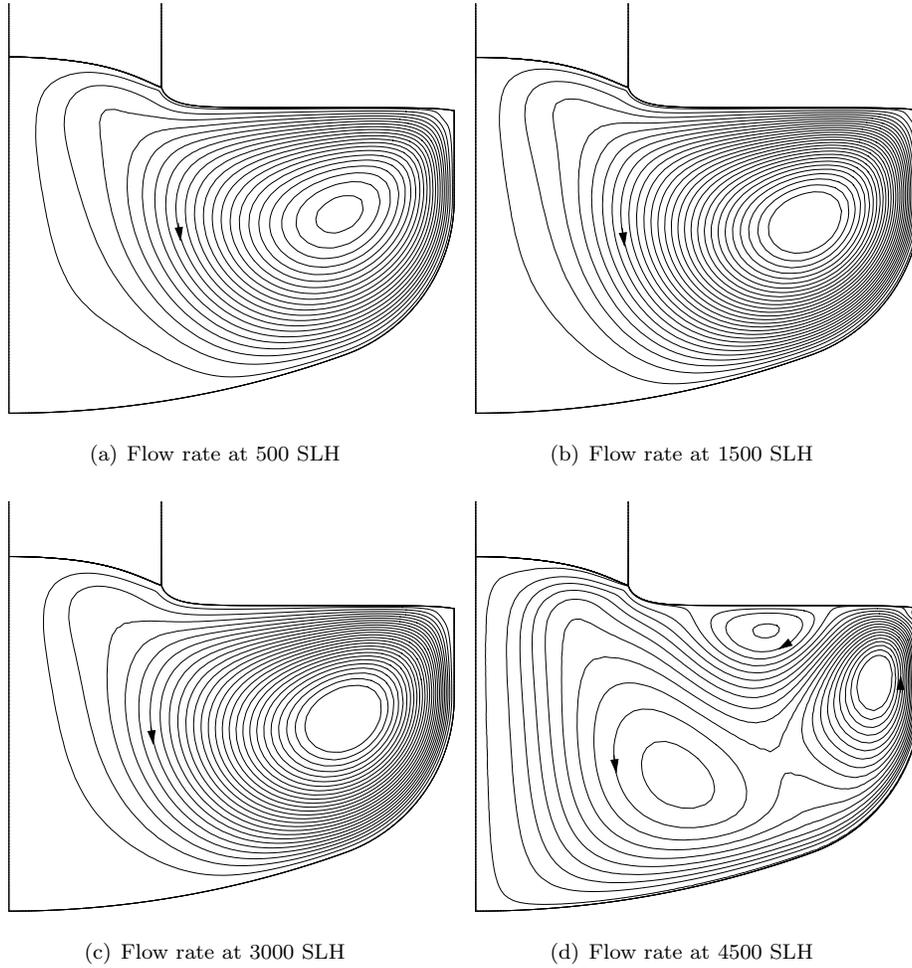


Figure 5.17: Stream function contour lines in the silicon melt by steps of $1\text{cm}^3/\text{s}$ at different gas flow rates. The crystal pulling rate is $2\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20/ - 1\text{rpm}$.

The argon gas effect on melt convection is also dependent on other growth parameters determining the turbulence characteristics, such as the crucible rotation rate. This can be confirmed when comparing our numerical results at different crucible rotation rates. From Figure (5.17), Figure (5.18) and Figure(5.19), one can see that the generation of a new vortex due to the gas

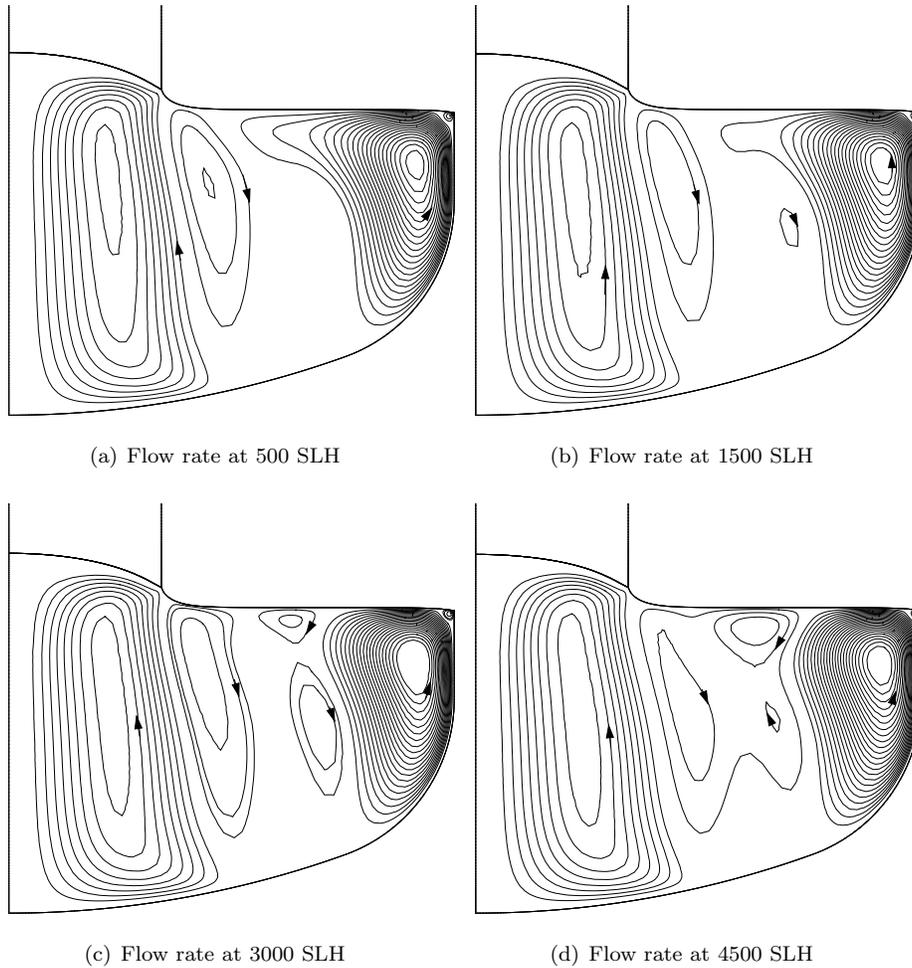


Figure 5.18: Stream function contour lines in the silicon melt by steps of $1\text{cm}^3/\text{s}$ at different gas flow rates. The crystal pulling rate is $2\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20/-5\text{ rpm}$.

shear stress is very sensitive to the crucible rotation rate. For example, when the crucible rotation rate is chosen to be -1 rpm , even the high flow rate at 3000 SLH is insufficient to generate a separate clockwise vortex. However, for the same flow rate but with a crucible rotation rate at -5 rpm (see Figure 5.18(c)) and -10 rpm , see Figure 5.19(c), an apparent separated clockwise vortex occurs due to gas convection. A similar conclusion was also drawn from other

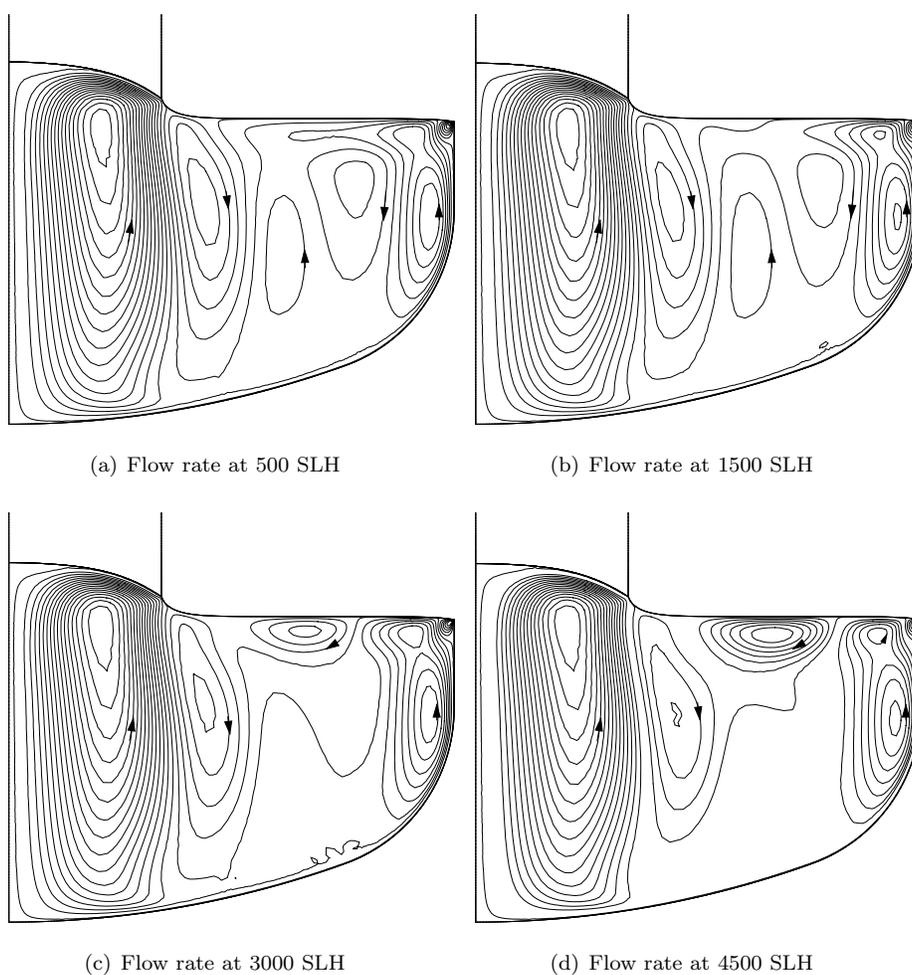


Figure 5.19: Stream function contour lines in the silicon melt by steps of $1\text{cm}^3/\text{s}$ at different gas flow rates. The crystal pulling rate is $2\text{mm}/\text{min}$, and the crystal/crucible rotation rates are $20/-10$ rpm.

numerical experiments [(Kalaev et al. 2003)].

Oxygen concentration in silicon crystals is usually adjusted by crucible rotation rate during growth, and it is well known that a high rotation rate yields a crystal with high oxygen concentration [(Togawa et al. 1996)]. Our numerical results of the oxygen distribution in the entire silicon melt at different crucible

rotation rates and gas flow rates are presented in Figure (5.21), while the oxygen concentrations along the melt axis at different crucible rotation rates and gas flow rates are given in Figure (5.22(a)) and Figure (5.20). One can clearly observe that the crucible rotation rate has a much more stronger influence on the oxygen distribution in the silicon melt than any other operating parameter, such as the argon gas flow rate or pressure, which can only be adjusted in a narrow range in industrial production. More specifically, the oxygen concentration along the silicon melt axis increases dramatically with an increasing crucible rotation rate, and this can be explained by the fact that more oxygen atoms dissolve from the crucible bottom wall and are transported to the silicon melt core by convection due to the stronger combination of crucible rotation effect (via Coriolis force) and buoyant force at a higher crucible rotation rate, see Figure (5.21(c)), while at a very low crucible rotation rate, the oxygen transport is dominated by diffusion through a thin diffusion layer along the crucible wall, as illustrated in Figure (5.21(a)). On the other hand, for a specified crucible rotation rate, a higher gas flow rate results in a significantly higher oxygen profile in comparison with a lower gas flow rate, which was also observed in experiments. For example, the experimental results obtained by Machida et al [(Machida et al. 1998)][(Machida et al. 2000)] revealed that there is a direct proportionality between argon gas flow rate and oxygen concentration, for which effect the authors gave two possible reasons. One possible reason invoked is that there is a temperature drop on the melt free surface due to the sweeping argon gas [(Machida et al. 1998)], particularly for the puller with a heat shield. A second possible reason is that the interaction between the vortex due to the gas shear stress and the vortex driven by buoyancy is suppressed, and therefore that the oxygen effective evaporation area decreases and finally the oxygen concentration in the silicon melt increases. However, our numerical results at low crucible rates cannot be explained by the two above possible reasons. Firstly, in our numerical experiments with the crucible rotation rate at -1 rpm, the effect of gas convection on the melt flow is found to be very limited. Secondly, the temperature along the melt free surface is found to be increased (and not decreased) with an increasing gas flow rate, while there is also a temperature increase along the melt/crucible interface, which can be clearly seen in Figure (5.22(b)). Therefore, we can attribute the increased oxygen concentration in the silicon melt to the increased temperature along the crucible bottom, where the oxygen dissolution rate from the silica crucible is assumed to be a function of temperature [(Kakimoto et al. 1996)].

Moreover, our numerical results show that a direct proportionality between the argon gas flow rate and the oxygen concentration only exists for low crucible

rotation rates with low gas flow rates, for example for a crucible rotation rate at -1 rpm and -5 rpm with a gas flow rate at 500 SLH, 1500 SLH or 3000 SLH (in our numerical experiments). But with a crucible rotation rate at -10 rpm, the oxygen concentration along the melt axis decreases considerably, and becomes even lower in comparison with the result obtained from a gas flow rate at 500 SLH when the gas flow rate is further increased to 4500 SLH, see Figure (5.20). This can be explained by the fact that SiO evaporation is dominated by a much more thinner diffusion layer at very high argon gas flow rate, and therefore that SiO evaporation is enhanced and finally that oxygen concentration in the silicon melt becomes lower. Even the temperature profile (see Figure 5.13(b)) along the silica crucible is a little bit higher than that with the flow rate at 3000 SLH.

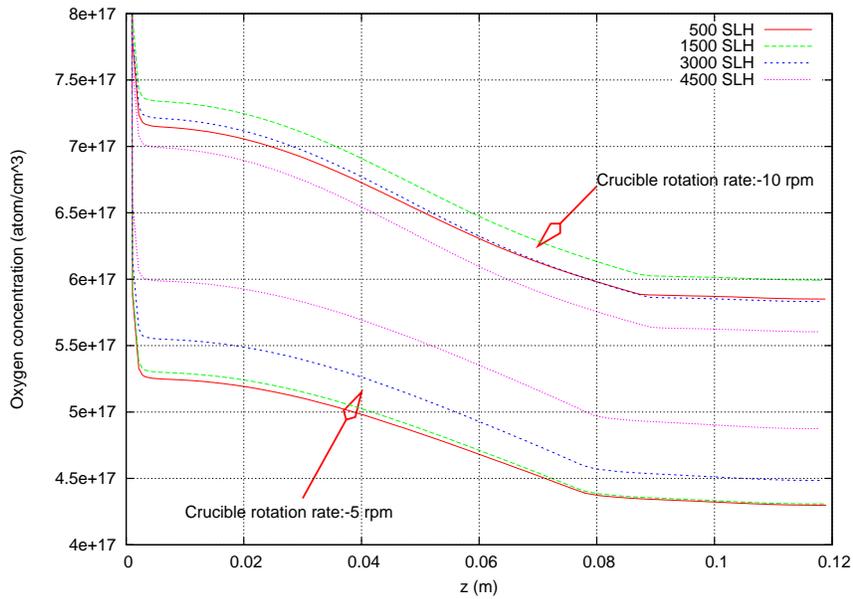
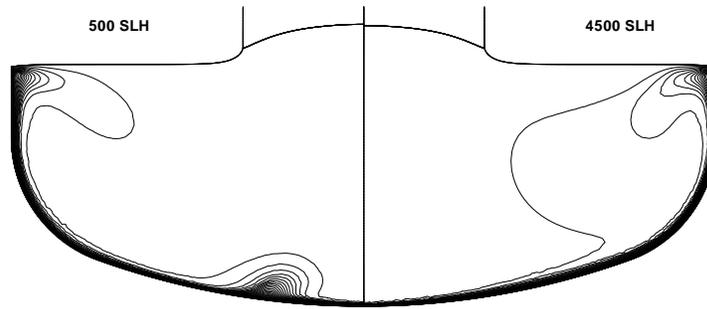
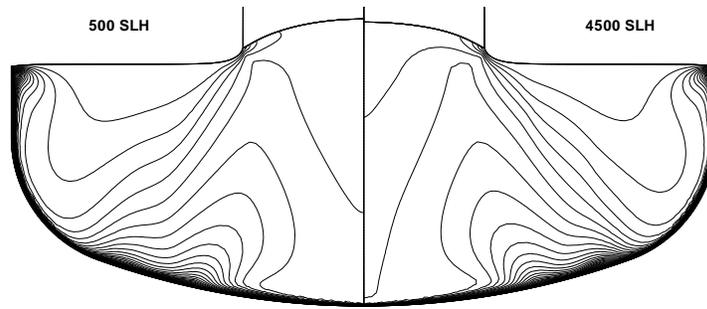


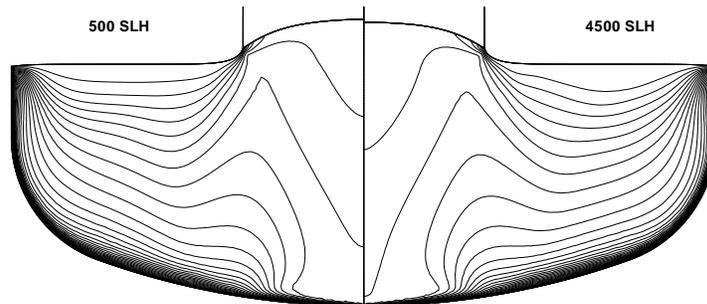
Figure 5.20: Oxygen distribution along the silicon melt axis for different gas flow rates with the crucible rotation rate at -5 rpm and -10 rpm.



(a) Crucible rotation rate at -1 rpm.

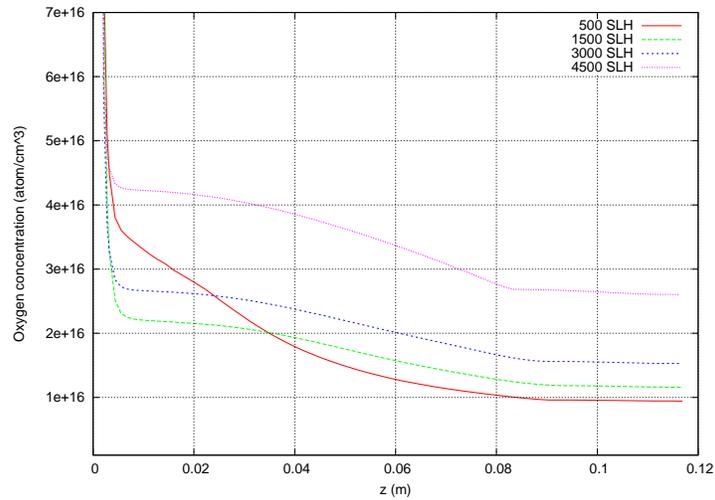


(b) Crucible rotation rate at -5 rpm.

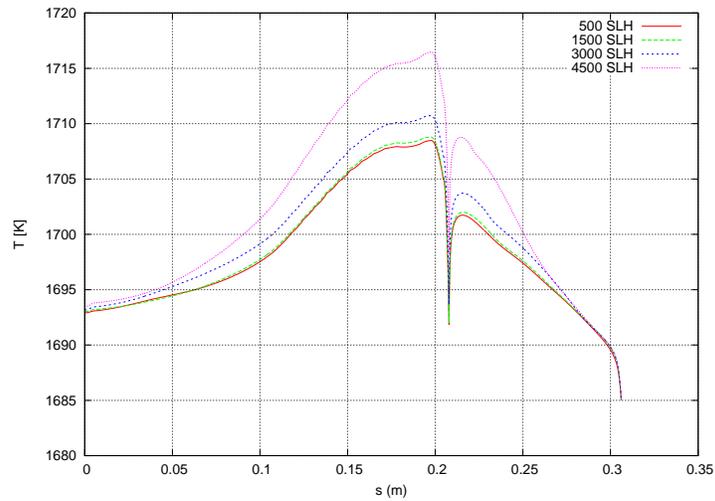


(c) Crucible rotation rate at -10 rpm.

Figure 5.21: Oxygen concentration in the melt by steps of $1.0e + 17$ atom/cm³ at different crucible rotation rates and gas flow rates. The crystal rotation rate is 20 rpm and the crystal pulling rate is 2mm/min.



(a) Oxygen concentration along the melt axis



(b) Temperature profile along the crucible bottom and the melt free surface

Figure 5.22: Simulation group 2. Influences of the gas flow rate on (a) the oxygen concentration in the silicon melt, and (b) the temperature profile along the crucible bottom and melt free surface. The crystal growth rate is 2mm/min and the crystal/crucible rotation rates are 20 rpm/-1 rpm.

5.8 Conclusions

The main purpose of this chapter was to be devoted to the numerical application of all the developments presented in previous chapters to industrial Czochralski crystal growth processes.

First of all, the FEMAG global modeling and iterative strategy employed in our numerical experiments were summarized. Then we have focused on the industrial application of Czochralski silicon growth processes by means of turbulent mixing-length, enhanced mixing-length and $k-l$ models with/without taking gas convection into account. The influences of the growth parameters, turbulence model and modeling parameters on the growth process, particularly on melt convection, melt heat transfer and melt/crystal interface shape were presented.

In order to analyze the effects of melt and gas convection on the Czochralski silicon growth process, a series of numerical experiments with pure heat conduction, with melt convection only, and with uncoupled and coupled gas convection was also carried out. Comparison of our numerical results with experiments shows that the quasi-steady global heat transfer model with the mixing-length or $k-l$ turbulence model well predicts the temperature difference in the silicon melt, while the pure heat conduction global model does not, whereas for the specified growth conditions, gas convection has a very limited influence on the temperature distribution in the silicon melt, the melt/crystal interface and heater power as well, whether melt convection is or not coupled with gas convection. Furthermore, a series of numerical experiments taking gas convection into account with varied growth conditions and operating parameters were performed, and the influences of the modeling parameters on the growth process, particularly on melt convection, melt heat transfer, melt/crystal interface and oxygen concentration, were fully given. Comparison of the simulation results (such as the heater power, temperature difference in the silicon melt, etc.) with literature and available experimental observations were also presented. Our numerical results show that:

- *The gas flow pattern, particularly the convection in the upper growth chamber, is sensitive to the gas flow rate. With increasing gas flow rate, the primary gas vortex in the upper growth chamber changes from clockwise to counter-clockwise, and the flow pattern is more complex at high gas flow rate as well. On the other hand, the major characteristics of the gas flow pattern in the bottom part of the growth chamber remain unchanged, except the intensities of those vortices.*

- *The forced argon gas flow also exerts a shear stress on the silicon melt free surface. At low gas flow rate, the influence of the gas flow on melt convection is very limited. However, at a high gas flow rate, a separated clockwise vortex due to the gas shear stress occurs. Moreover, the generation of this clockwise vortex is also sensitive to other growth conditions, such as the crucible rotation rate.*
- *For a specified crucible rotation rate, the melt/crystal interface is less convex with an increasing gas flow rate. Furthermore, the melt/crystal interface deflection is much more sensitive to the gas flow rate at higher crucible rotation rate. On the other hand, for a specified gas flow rate, the evolution of the melt/crystal interface becomes more complex and intersections between these interfaces are found at different crucible rotation rates.*
- *The oxygen distribution in the silicon melt is strongly dependent on the crucible rotation rate. At low crucible rotation rate, oxygen transport in the silicon melt is dominated by diffusion through a thin diffusion layer along the crucible wall, and the oxygen concentration in the silicon melt increases dramatically with increasing crucible rotation rate due to a strong combination of crucible rotation effects, buoyant force and gas shear stress. It is also well-known that the gas shear stress exerts a prominent effect on oxygen evaporation at the melt surface.*
- *On the other hand, our numerical experiments reveal that for a specified crucible rotation rate, a higher gas flow rate results in a higher oxygen concentration in the silicon melt mainly due to a high temperature profile along the crucible wall. However, there is no direct proportionality between the gas flow rate and the oxygen concentration at a high crucible rotation rate.*
- *As a last comment it should be recalled that underresolved boundary layers were always used in our numerical experiments. Therefore, the details of the flow and temperature distribution in the melt and gas could only be captured by means of an appropriate treatment (such as the use of wall functions, or of anisotropic mesh refinement near the boundary, or an enhanced Lagrange Multipliers technique).*

Global numerical modeling and simulation of bulk crystal growth has become an essential and indispensable powerful tool to predict, understand, develop and optimize single crystal growth processes. The main objective of this thesis aimed to develop a fully automatic Czochralski simulator predicting the entire Czochralski process taking all important physical phenomena and furnace constituents (or macro-elements) into account, including radiation heat transfer in the whole growth system, convective and conductive heat transfer in the silicon melt and inert gas chamber, silicon impurity (oxygen) transport by convection and diffusion, etc.. With this objective in mind, our major contributions towards the next generation Czochralski simulation tool are summarized as follows.

First of all, powerful, robust and efficient mesh generation algorithms with enough flexibility to control the mesh density were developed for whole bulk crystal growth processes, including a global grade-adaptive 1D mesh generation algorithm for curved boundaries, an initial simple polygon triangulation algorithm with/without holes by our improved sweep line technique, and an automatic 2D shape-quality unstructured mesh generator by incremental Delaunay refinement algorithm, where the refinement process was optimized by our firstly proposed linear point location algorithm without using any additional preprocessing, storage space and complicated data structure. The successful developments of all these algorithms form a solid basis towards the development of next generation Czochralski simulators.

Melt convection modeling remains one of the most biggest challenging problems of the Czochralski crystal growth process. In this work, a Finite Element Navier-Stokes solver based on unstructured meshes was developed, and the code was validated by solving the well-known 2D lid-driven cavity flow problem at different Reynolds numbers up to 25,000. Accordingly, the turbulence mixing-length and $k-l$ models and the numerical method developed by the CESAME crystal growth research group in the past decades were adapted to unstructured meshes.

Another of our main objectives regarding melt convection modeling was to

develop relatively simple turbulence models, but with reasonable accurateness in order to simulate efficiently the average axisymmetric flow in the Czochralski melt as coupled to the whole quasi-steady or time-dependent growth problem. Therefore, an enhanced model taking into account different turbulent boundary layers along the melt/crucible and melt/crystal interfaces was presented and analyzed.

On the other hand, the coupling of an additional nonlinear equation to the Reynolds equations increases dramatically the numerical difficulties encountered when trying to simulate the melt flow by the turbulence $k-l$ model. These difficulties either come from the hyperbolic nature of the nonlinear transport terms or from the square-root of k present in the turbulent viscosity expression in the Reynolds equations and turbulent kinetic energy equation. Both of these difficulties, particularly the negative k issue can dramatically slow down or even let fail the Newton-Raphson iterative scheme. Therefore, a generic transformation method to overcome these difficulties was proposed and implemented, and our primary numerical experiments have shown that the negative k problem was well solved by the specific transformation schemes we have chosen.

In practice, gas convection exerts an important effect on the global heat transfer and oxygen concentration both in the silicon melt and silicon crystal in modern Czochralski crystal growth processes. Therefore, laminar and turbulent mathematical models governing the gas convection, thermal distribution and oxygen concentration were developed, and Finite Element numerical methods to solve these governing equations on unstructured meshes were presented. In order to analyze the gas effects on the silicon melt flow, oxygen distribution and global heater transfer, appropriate numerical approaches to capture the wall shear stress exerted by the gas flow and experienced by the silicon melt were implemented and investigated. Our primary numerical experiments have shown that, at a very small Reynolds number, the viscous shear stresses obtained by a Direct method and the Lagrange Multipliers are very similar. However, at a medium or large Reynolds number, the Lagrange Multipliers technique always failed to capture the stresses due to strong oscillations of the stress tensor.

Finally, a series of numerical experiments devoted to investigate the industrial Czochralski crystal growth process under various growth conditions were presented based on all the developments implemented. Comparison of our numerical results with experiments shows that the quasi-steady global heat transfer model with the mixing-length or $k-l$ turbulence models well predicts the temperature difference in the silicon melt, while the pure heat conduction global model does not, whereas for the specified growth conditions investigated, gas convection has a very limited influence on the temperature distribution

in the silicon melt, the melt/crystal interface and the heater power as well, whether melt convection is or not coupled with gas convection. The effect of gas convection on oxygen evaporation at the melt surface was not investigated but certainly this effect is non-negligible. However, unfortunately, all our numerical experiments were performed with underresolved meshes in the melt boundary layers, and the Lagrange Multipliers technique failed to resolve the problem. Therefore some care should be given when considering the detail of our solutions in the melt whereas accurate global results are generally obtained.

Furthermore, a series of numerical experiments taking gas convection into account with varied growth conditions and operating parameters were performed, and the influences of the modeling parameters on the growth process, particularly on melt convection, melt heat transfer, melt/crystal interface and oxygen concentration, were fully given. Comparison of the simulation results (such as the heater power, the temperature difference in the silicon melt, etc.) with literature and available experimental observations were also presented, and conclusions were drawn based on these simulation results and observations.

Appendix A

Discretization of generic transformed turbulent kinetic energy equation by Finite Element Method

The finite element discretization of unknowns (L) in the transformed turbulent kinetic energy equation (3.120) (in weak form) is straight-forward, that is, L can be discretized like a regular unknowns as follows:

$$L = \sum_j^{n_L} L_j \psi_j \quad (\text{A.1})$$

where n_L is the number of nodes on each element. In each Newton-Raphson iteration, the transformed nonlinear turbulent kinetic energy equation has to be linearized, thus unknowns L and corresponding unknown derivatives should have to be developed and calculated a priori.

However, as mentioned earlier in chapter 3, we never introduce the new variable L into the system in order to keep the solver consistency and simplicity. That is to say, the nodal value L_j are directly obtained from the stored nodal value k_j , since

$$L = nk^{1/n} \quad (\text{A.2})$$

and

$$k = \left(\sum_j^{n_L} k_j^{1/n} \psi_j \right)^n \quad (\text{A.3})$$

when $n \rightarrow \infty$ (for *logk* scheme), the equation (A.2) and (A.3) become:

$$L = \lim_{n \rightarrow \infty} (nk^{1/n}) = \log k \quad (\text{A.4})$$

and

$$k = e^L = \exp\left(\sum_{j=1}^{n_L} L_j \phi_j\right) = \prod_j^{n_L} k_j^{\psi_j} \quad (\text{A.5})$$

Similarly, corresponding unknown derivatives can also be developed and calculated a priori based on nodal value k_j as follows:

$$\begin{aligned}
 k_{,L_j} &= \frac{\partial k}{\partial L_j} = \frac{\partial k}{\partial (nk_j^{1/n})} \\
 &= \frac{\partial}{\partial (nk_j^{1/n})} \left[n^{-1} \sum_k^{n_k} (nk_k^{1/n}) \psi_k \right]^n \\
 &= \frac{n}{n^n} \left[\sum_k^{n_k} (nk_k^{1/n}) \psi_k \right]^{n-1} \psi_j \\
 &= k^{(1-1/n)} \psi_j
 \end{aligned} \tag{A.6}$$

$$k_{,x_\alpha} = \frac{\partial k}{\partial x_\alpha} = nk^{(1-1/n)} \sum_j^{n_k} k_j^{1/n} \psi_{j,x_\alpha} = k^{(1-1/n)} L_{,x_\alpha} \tag{A.7}$$

$$\begin{aligned}
 \frac{\partial k_{,x_\alpha}}{\partial L_j} &= \frac{\partial k_{,x_\alpha}}{\partial (nk_j^{1/n})} = \frac{\partial}{\partial x_\alpha} \left[k^{(1-1/n)} \psi_j \right] \\
 &= \left(1 - \frac{1}{n}\right) k^{-1/n} k_{,x_\alpha} \psi_j + k^{(1-1/n)} \psi_{j,x_\alpha} \\
 &= \left(1 - \frac{1}{n}\right) k^{(1-2/n)} L_{,x_\alpha} \psi_j + k^{(1-1/n)} \psi_{j,x_\alpha}
 \end{aligned} \tag{A.8}$$

when $n \rightarrow \infty$ (for *logk* scheme), the equation (A.6), (A.7) and equation (A.8) become:

$$k_{,L_j} = \frac{\partial k}{\partial L_j} = \frac{\partial k}{\partial (\log k_j)} = \lim_{n \rightarrow \infty} (k^{1-1/n} \psi_j) = k \psi_j \tag{A.9}$$

$$\begin{aligned}
 k_{,x_\alpha} &= \frac{\partial k}{\partial x_\alpha} \\
 &= \lim_{n \rightarrow \infty} \left(nk^{(1-1/n)} \sum_j^{n_k} k_j^{1/n} \psi_{j,x_\alpha} \right) \\
 &= k \sum_j^{n_k} (\log k_j \psi_{j,x_\alpha}) = k L_{,x_\alpha}
 \end{aligned} \tag{A.10}$$

$$\begin{aligned}
 \frac{\partial k_{,x_\alpha}}{\partial L_j} &= \frac{\partial k_{,x_\alpha}}{\partial (\log k_j)} = \frac{\partial}{\partial x_\alpha} (k \psi_j) \\
 &= \lim_{n \rightarrow \infty} \left[\left(1 - \frac{1}{n}\right) k^{-1/n} k_{,x_\alpha} \psi_j + k^{(1-1/n)} \psi_{j,x_\alpha} \right] \\
 &= k_{,x_\alpha} \psi_j + k \psi_{j,x_\alpha}
 \end{aligned} \tag{A.11}$$

Based on the derivation of equation (A.6), (A.7), and (A.8), we choose not to discretize the formal transformed turbulent energy equation (3.120), and a better way is to discretize the original turbulent energy equation with testing functions $k^{(1-1/m)}\psi_i$ as follows:

$$\int_{\Omega} \left[(u \cdot \nabla k)(k^{(1-1/m)}\psi_i) + \left(\nu + \frac{\nu_t}{\sigma_k}\right)(\nabla k : \nabla(k^{(1-1/m)}\psi_i)) + (C_d k^{3/2}/l_m - \mu_t \dot{\gamma} + \beta g \frac{\mu_t}{Pr_t} \cdot \nabla T)(k^{(1-1/m)}\psi_i) \right] d\Omega = 0 \quad (\text{A.12})$$

Letting,

$$\begin{aligned} A &= \int_{\Omega} \left[(u \cdot \nabla k)(k^{(1-1/m)}\psi_i) \right] d\Omega \\ &= \int_{\Omega} \left[(uk_{,x} + vk_{,y})(k^{(1-1/m)}\psi_i) \right] d\Omega \\ &= \int_{\Omega} \left[k^{(2-1/n-1/m)} (uL_{,x} + vL_{,y}) \right] d\Omega \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} B &= \int_{\Omega} \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)(\nabla k : \nabla(k^{(1-1/m)}\psi_i)) \right] d\Omega \\ &= \int_{\Omega} \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) \left(k_{,x}(k^{(1-1/m)}\psi_i)_{,x} + k_{,y}(k^{(1-1/m)}\psi_i)_{,y} \right) \right] d\Omega \\ &= \int_{\Omega} \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) k^{(2-2/n-1/m)} \left[\left(1 - \frac{1}{m}\right)(L_{,x}^2 + L_{,y}^2)\psi_i + k^{1/n}(L_{,x}\psi_{i,x} + L_{,y}\psi_{i,y}) \right] \right] d\Omega \end{aligned} \quad (\text{A.14})$$

$$C = \int_{\Omega} (C_d k^{3/2} - \mu_t \dot{\gamma} - \frac{\mu_t}{Pr_t} \beta g \cdot \nabla T)(k^{(1-1/m)}\psi_i) d\Omega \quad (\text{A.15})$$

then,

$$\begin{aligned}
 \frac{\partial A}{\partial L_j} &= \int_{\Omega} \left[(u \cdot \nabla k) (k^{(1-1/m)} \psi_i) \right]_{,L_j} d\Omega \\
 &= \int_{\Omega} \left[\left(u \frac{\partial k_{,x}}{\partial L_j} + v \frac{\partial k_{,y}}{\partial L_j} \right) k^{(1-1/m)} \psi_i + (uk_{,x} + vk_{,y}) \left(1 - \frac{1}{m} \right) k^{-1/m} \frac{\partial k}{\partial L_j} \psi_i \right] d\Omega \\
 &= \int_{\Omega} \left[\left(2 - \frac{1}{m} - \frac{1}{n} \right) k^{(2-2/n-1/m)} (uL_{,x} + vL_{,y}) \psi_i \psi_j + k^{(2-1/n-1/m)} (u\psi_{j,x} \psi_i + v\psi_{j,y} \psi_i) \right] d\Omega \quad (\text{A.16})
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial B}{\partial L_j} &= \int_{\Omega} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) (\nabla k : \nabla (k^{(1-1/m)} \psi_i)) \right]_{,L_j} d\Omega \\
 &= \int_{\Omega} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \left(2 - \frac{2}{n} - \frac{1}{m} \right) k^{(2-3/n-1/m)} \left[\left(1 - \frac{1}{m} \right) (L_{,x}^2 + L_{,y}^2) \psi_i + k^{1/n} (L_{,x} \psi_{i,x} + L_{,y} \psi_{i,y}) \right] \psi_j \right. \\
 &+ \left. \left(\nu + \frac{\nu_t}{\sigma_k} \right) k^{(2-2/n-1/m)} \left[2 \left(1 - \frac{1}{m} \right) \psi_i (L_{,x} \psi_{j,x} + L_{,y} \psi_{j,y}) + \frac{1}{n} (L_{,x} \psi_{i,x} + L_{,y} \psi_{i,y}) \psi_j + k^{1/n} (\psi_{j,x} \psi_{i,x} + \psi_{j,y} \psi_{i,y}) \right] \right. \\
 &+ \left. \frac{C_4 l_m}{2\sigma_k} k^{(5/2-3/n-1/m)} \left[\left(1 - \frac{1}{m} \right) (L_{,x}^2 + L_{,y}^2) \psi_i + k^{1/n} (L_{,x} \psi_{i,x} + L_{,y} \psi_{i,y}) \right] \psi_j \right] d\Omega \quad (\text{A.17})
 \end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{C}}{\partial L_j} &= \int_{\Omega} \left[(C_d k^{3/2} / l_m - \mu_t \dot{\gamma} - \frac{\mu_t}{P_{rt}} \beta g \cdot \nabla T) (k^{(1-1/m)} \psi_i) \right]_{,L_j} d\Omega \\
&= \int_{\Omega} \left[\left(\frac{3}{2} C_d k^{1/2} k_{,L_j} / l_m - \frac{1}{2} C_4 l_m k^{-1/2} k_{,L_j} \dot{\gamma} - \frac{C_4 l_m k^{-1/2} k_{,L_j}}{2 P_{rt}} \beta g \cdot \nabla T \right) (k^{(1-1/m)} \psi_i) \right. \\
&+ \left. \left(1 - \frac{1}{m} \right) (C_d k^{3/2} / l_m - \mu_t \dot{\gamma} - \frac{\mu_t}{P_{rt}} \beta g \cdot \nabla T) k^{(-1/m)} k_{,L_j} \psi_i \right] d\Omega \\
&= \int_{\Omega} \left[\left(\frac{3}{2} C_d k^{(3/2-1/n)} / l_m - \frac{1}{2} C_4 l_m k^{(1/2-1/n)} \dot{\gamma} - \frac{C_4 l_m k^{(1/2-1/n)}}{2 P_{rt}} \beta g \frac{\partial T}{\partial y} \right) k^{(1-1/m)} \psi_i \psi_j \right. \\
&+ \left. \left(1 - \frac{1}{m} \right) (C_d k^{3/2} / l_m - C_4 l_m k^{1/2} \dot{\gamma} - \frac{C_4 l_m k^{1/2}}{P_{rt}} \beta g \frac{\partial T}{\partial y}) k^{(1-1/m-1/n)} \psi_i \psi_j \right] d\Omega \tag{A.18}
\end{aligned}$$

Since we choose to solve the turbulent kinetic energy k together with the velocity and pressure field in a coupled way, therefore, the derivatives of the turbulent kinetic energy to velocities u_j , v_j and w_j should be developed. Details of the developments and forms are exactly as the same without transformation except the way to calculate the k by equation (3.117). Furthermore, the Navier-Stokes equation to unknown L_j should also have to be developed. Let the F^u , F^v and F^w represent the three governing equations of u , v and w components in weak form, then in the axisymmetric coordinate system, we have:

$$\begin{aligned}
 \frac{\partial F_i^u}{\partial L_j} &= \int_{\Omega} \left[\rho(\nu + \nu_t)(\nabla u + \nabla u^T) : \nabla \psi_i \right]_{,L_j}' d\Omega \\
 &= \int_{\Omega} \left[\frac{1}{2} \rho C_4 l_m k^{-1/2} k_{,L_j} (\nabla u + \nabla u^T) : \nabla \psi_i \right] d\Omega \\
 &= \int_{\Omega} \left[\frac{1}{2} \rho C_4 l_m k^{(1/2-1/n)} \psi_j \left(2 \frac{\partial u}{\partial x} \frac{\partial \phi_i}{\partial x} + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial \phi_i}{\partial y} + 2 \frac{u}{x} \frac{\phi_i}{x} \right) \right] d\Omega
 \end{aligned} \tag{A.19}$$

Similarly,

$$\frac{\partial F_i^v}{\partial L_j} = \int_{\Omega} \left[\frac{1}{2} \rho C_4 l_m k^{(1/2-1/n)} \psi_j \left(\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial \phi_i}{\partial x} + 2 \frac{\partial v}{\partial y} \frac{\partial \phi_i}{\partial y} \right) \right] d\Omega \tag{A.20}$$

$$\frac{\partial F_i^w}{\partial L_j} = \int_{\Omega} \left[\frac{1}{2} \rho C_4 l_m k^{(1/2-1/n)} \psi_j \left(\left(\frac{\partial w}{\partial x} - \frac{w}{x} \right) \left(\frac{\partial \phi_i}{\partial x} - \frac{\phi_i}{x} \right) + \frac{\partial w}{\partial y} \frac{\partial \phi_i}{\partial y} \right) \right] d\Omega \tag{A.21}$$

Bibliography

- Amato, N. M., Goodrich, M. T. and Ramos, E. A.: 2000, Linear-time triangulation of a simple polygon made easier via randomization, *Symposium on Computational Geometry*, pp. 201–212.
- Asano, T., Edahiro, M., Imai, H., Iri, M. and Murota, K.: 1985, Practical use of bucketing techniques in computational geometry, *Computational Geometry* pp. 153–195.
- Assaker, R.: 1998, *Magnetohydrodynamics in Crystal Growth*, PhD thesis, Université Catholique de Louvain.
- Assaker, R., van den Bogaert, N. and Dupret, F.: 1997, Time-dependent simulation of the growth of large silicon crystals by the czochralski technique using a turbulent model for melt convection, *J. Crystal Growth* **180**, 450–460.
- Atherton, L., Derby, J. and Brown, R. A.: 1987, Radiative heat exchange in czochralski crystal growth, *J. Crystal Growth* **84**, 57–78.
- Babuška, I.: 1971, Error bounds for finite element methods, *Numerische Mathematik* **16**, 322–333.
- Babuška, I.: 1973, The finite element method with lagrangian multipliers, *Numerische Mathematik* **20**, 179–192.
- Baldwin, B. and Lomax, H.: 1978a, Thin-layer approximation and algebraic model for separated turbulent flows, *AIAA Paper* **78**, 257.
- Baldwin, B. and Lomax, H.: 1978b, Thin-layer approximation and algebraic model for separated turbulent flows, *AIAA paper* **11**, 78–257.
- Berghezan, D. and Dupret, F.: 1994, Numerical simulation of stratified coating flow by a variational method, *J. of Comput. Physics* **111**, 165–182.
- Bornside, D., Kinney, T., Brown, R. and Kim, G.: 1990, Finite element/newton method for the analysis of czochralski crystal growth with diffuse-grey radiative heat transfer, *International Journal for Numerical Methods in Engineering* **30**, 133–154.
- Bowyer, A.: 1981, Computing dirichlet tessellations, *The computer Journal* **24**, 162–166.

- Brezzi, F.: 1974, On the existence, uniqueness and approximation of saddle-point problems arising from lagrange multipliers, *Série Rouge Analyse Numérique* **8**, 129–151.
- Brown, R.: 1998, Theory of transport processes in single crystal growth from the melt, *AIChE Journal* **34**.
- Burton, J., Prim, R. and Slichter, W.: 1958, The distribution of solute in crystals grown from the melt. part i. theoretical, *J. Chem. Phys.* **21**, 1987–1991.
- Capper, P.: 2005, *Preface, in Bulk Crystal Growth of Electronic, Optical, and Optoelectronic Materials*, John Wiley and Sons Ltd.
- Chang, C. and Brown, R.: 1984, Natural convection in steady solidification: Finite element analysis of a two-phase rayleigh-bnard problem, *Journal of Computational Physics* **53**, 1–27.
- Chazelle, B.: 1991, Triangulating a simple polygon in linear time, *Discrete Comp. Geom.* **6**, 485–524.
- Chazelle, B. and Incerpi, J.: 1984, Triangulation and shape-complexity, *ACM Trans. Graph.* **3**, 125–152.
- Chien, K.: 1982, Prediction of channel and boundary-layer flows with a low-reynolds number turbulence model, *AIAA J.* **20**, 3338.
- Chung, H., Lee, S. and Yoon, J.: 1996, Numerical prediction of operational parameters in czochralski growth of large-scale si, *J. Crystal Growth* **163**, 249–258.
- Colley, A., Thomas, P., Carpenter, P. and Cooper, A.: 1999, An experimental study of boundary-layer transition over a rotating, compliant disk, *Physica of Fluids* **11**, 3340–3351.
- Crespo del Arco, E., Pulicani, J. and Bontoux, P.: 1989, *Physico-Chem. Hydrogen* **11**, 681.
- Crespo del Arco, E., Pulicani, J. and Bontoux, P.: 1997, *Proc. Conf. on Synergistics, Order and Chaos*, pp. 244–257.
- Crochet, M., Dupret, F., Ryckmans, Y., Geyling, F. and Monberg, E.: 1989, Numerical simulation of crystal growth in a vertical bridgman furnace, *J. Crystal Growth* **97**, 173–185.
- Crochet, M., Wouters, P., Geyling, F. and Jordan, A.: 1983a, Finite element simulation of czochralski bulk flow, *J. Crystal Growth* **65**, 153–165.
- Crochet, M., Wouters, P., Geyling, F. and Jordan, A.: 1983b, Finite-element simulation of czochralski bulk flow, *J. Crystal Growth* **65**, 153–165.
- Crow, S.: 1968, Viscoelastic properties of fine-grained incompressible turbulence, *J. Fluid Mech.* **33**, 1–20.
- Czochralski, J.: 1918, Ein neues verfahren zur messung der kristallisationsgeschwindigkeit der metalle, *Z. phys. Chemie* **92**, 219–221.
- de Berg, M., van Kreveld, M., Overmars, M. and Schwarzkopf, O.: 2000, *Computational Geometry: Algorithms and Applications, 2nd Edition*, Springer-Verlag, Berlin.

- Delsaute, B., Regnier, V., Houtermans, R. and Dupret, F.: 2006, Bamberg, Germany, 3d modeling of czochralski melt flow and experimental validation, *5th International Workshop on Modeling in Crystal Growth*.
- Derby, J., Atherton, L. and Gresho, P.: 1989, An integrated process model for the growth of oxide crystals by the czochralski method, *J. Crystal Growth* **97**, 792–826.
- Derby, J. and Brown, R.: 1987, On the dynamics of czochralski crystal growth, *J. Crystal Growth* **83**, 137–151.
- Dornberger, E., Tomzig, E., Seidl, A., Schmitt, S., Leister, H. J., Schmitt, C. and Müller, G.: 1997, Thermal simulation of the czochralski silicon growth process by three different models and comparison with experimental results, *J. Crystal Growth* **180**, 461–467.
- Driest, E. V.: 1956, On turbulent flow near a wall, *Journal of the Aeronautical Science* **23**, 1007.
- D.T.Lee and B.J.Schachter: 1980, Two algorithms for constructing a delaunay triangulation, *Int. J. of Comp. and info. Sci.* **9**, 219–242.
- Dupret, F., Ryckmans, Y., Wouters, P. and Crochet, M. J.: 1986a, Numerical calculation of the global heat transfer in a czochralski furnace, *J. Crystal Growth* **79**, 84–91.
- Dupret, F., Ryckmans, Y., Wouters, P. and Crochet, M. J.: 1986b, Numerical calculation of the global heat transfer in a czochralski furnace, *Proc. 6th Int. Symp. on Finite Element methods in Flow Problems*.
- Dupret, F., Ryckmans, Y., Wouters, P. and Crochet, M. J.: 1990, Global modelling of heat transfer in crystal growth furnaces, *Int. J. of Heat and Mass Transfer* **33**, 1849–1971.
- Dupret, F. and Van den Bogaert, N.: 1994, *Handbook of crystal growth: Modeling of Bridgman and Czochralski Growth*, Elsevier Science B.V.
- E. Erturk, T. C. and Gokcol, C.: 2005, Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers, *Int. J. for Num. Meth in Fluids* **48**, 747–774.
- ElGindy, H., Everett, H. and Toussaint, G.: 1993, Slicing an ear using prune-and-search, *Pattern Recogn. Letters* **14**, 719–722.
- Escudier, M.: 1966, The distribution of mixing-length in turbulent flows near walls, *Technical Report TWF/TN/12*, Imperial College.
- Ettouney, H. and Brown, R.: 1983, Finite-element methods for steady solidification problems, *Journal of Computational Physics* **49**, 118–150.
- Evstratov, I., Kalaev, V. V., Zhmakin, A. I., Makarov, Y. N., Abramov, A. G., Ivanov, N. G., Smirnov, E. M., Dornberger, E., Virbulis, J., Tomzig, E. and von Ammon, W.: 2001, Modeling analysis of unsteady three-dimensional turbulent melt flow during czochralski growth of si crystals, *J. Crystal Growth* **230**, 22–29.

- Evstratov, I., Kalaeva, V., Zhmakina, A., Makarov, Y., Abramovc, A., Ivanovc, N., Korsakovc, A., Smirnovc, E., Dornbergerd, E., Virbulisd, J., Tomzigd, E. and von Ammon, W.: 2002, Numerical study of 3d unsteady melt convection during industrial-scale cz si-crystal growth, *J. Crystal Growth* **237-239**, 1757–1761.
- Falster, R. and Bergholz, W.: 1990, The gettering of transition metals by oxygen-related defects in silicon, *J. Electrochem.* **137**, 1548.
- Farin, G.: 1998, *Curves and surfaces for CAGD, 4th Edition*, Academic Press, Boston.
- Fortune, S.: 1987, A sweepline algorithm for voronnı diagrams, *Algorithmica* **2**, 153–174.
- Fortune, S. and Wyk, C. J.: 1993, Computational geometry: an introduction, *Proceedings of 9th annual Symposium in Computational Geometry*, pp. 163–172.
- Garey, M., Johnson, D., Preparata, F. and Tarjan, R.: 1978, Triangulating a simple polygon, *Information Processing Letters* **7**, 175–179.
- Gräbner, O., Müller, G., Virbulis, J., Tomzig, E. and von Ammon, W.: 2001, Effects of various magnetic field configurations on temperature distributions in czochralski silicon melts, *Microelectronic Eng.* **56**, 83–88.
- Greene, D.: 1983, The decomposition of polygons into convex parts, *Advances in computing reseaech, JAI press*, pp. 235–259.
- Habets, F. and Nyevelt, E. Z. V.: 1994, *Etude numérique de la convection turbulence en tirage de cristaux semiconducteurs: modélisation de la turbulence*, PhD thesis, Eng. Final Work, Université Catholique de Louvain.
- Hanjalic, K. and Launder, B.: 1972, Fully developed asymeric flow in a plan channel, *J. of Fluid Mechanics* **51**, 301.
- Held, M.: 2001, Fist: Fast industrial-strength triangulation of polygons, *Algorithmica* **30**, 563–596.
- Hertel, S. and Mehlhorn, K.: 1983, Fast triangulation of simple polygons, *Proc. 4th int. Conf. Found. Comput:Theory*, Springer-Verlag, Berlin, pp. 207–218.
- Hibiya, T. and Hoshikawa, K.: 2005, *Silicon, in Bulk Crystal Growth of Electronic. Optical. and Optoelectronic Materials*, John Wiley and Sons Ltd.
- Hintz, P., Schwabe, D. and Wilke, H.: 2001, Convection in a czochralski crucible ? part 1: non-rotating, *J. Crystal Growth* **222**, 343–355.
- Hirata, H. and Hoshikawa, K.: 1990, Oxygen solubility and its temperature dependence in a silicon melt in equilibrium with solid silica, *J. Crystal Growth* **106**, 657–664.
- Hoshikawa, K., Hirata, H., Nakanishi, H. and Ikuta, K.: 1981, Semiconductor silicon 1981, *The Electrochemical Society, Pennington, NJ*, p. 101.
- Hoshikawa, K. and Huang, X.: 2000, Oxygen transportation during czochralski silicon crystal growth, *Materials Science and Engineering* **72**, 73–79.
- Hu, S. and Pateick, W.: 1975, missed, *J. Appl. Phys.* **46**, 1869.
- Hurle, D.: 1993, *Crystal pulling from the melt*, Springer-Verlag, Berlin.

- Hurle, D.: 1994, *Bulk crystal growth (2B): growth mechanisms and dynamics*, Elsevier Science B.V.
- Jones, W. and Launder, B.: 1972, The prediction of laminarization with a two-equation model of turbulence, *Int. J. Heat Mass Transfer* **15**, 301–464.
- Kakimoto, K., Nicodeme, P., Lecomte, M., Dupret, F. and Crochet, M. J.: 1991, Numerical simulation of molten silicon flow; comparison with experiment, *J. Crystal Growth* **114**, 715–725.
- Kakimoto, K., Tashiro, A., Shinozaki, T., Ishii, H. and Hashimoto, Y.: 2002, Mechanisms of heat and oxygen transfer in silicon melt in an electromagnetic czochralski system, *J. Crystal Growth* **243**, 55–65.
- Kakimoto, K., Yi, K. and Eguchi, M.: 1996, Oxygen transfer during single silicon crystal growth in czochralski system with vertical magnetic fields, *J. Crystal Growth* **163**, 238–242.
- Kalaev, V. V., Evstratov, I. Y. and Makarov, Y. N.: 2003, Gas flow effect on global heat transport and melt convection in czochralski silicon growth, *J. Crystal Growth* **249**, 87–99.
- Kalaev, V., Zhmakin, A. and Smirnov, E.: 2002, Modeling of turbulent melt convection during czochralski bulk crystal growth, *J. Turbulence* **3**, 1–11.
- Keil, J.: 1985, Decomposing a polygon into simpler components, *SIAM J. Comput.* **14**, 799–817.
- Kim, K. and Langlois, W.: 1991, Computer simulation of oxygen segregation in cz/mcz silicon crystals and comparison with experimental results, *J. Electrochem. Soc.* **138**, 1850.
- Kinney, T. and Brown, R.: 1993, Application of turbulence modeling to the integrated hydrodynamic thermal-capillary model of czochralski crystal growth of silicon, *J. Crystal Growth* **132**, 551–574.
- Kirkpatrick, D., Klawe, M. and Tarjan, R.: 1990, Polygon triangulation in $o(n \log n)$ time with simple data structures, *In Proc. 6th ACM-SIAM Symp. on Comput. Geom.*, pp. 34–43.
- Kishida, Y. and Okazawa, K.: 1999, Geostrophic turbulence in czochralski silicon crucible, *J. Crystal Growth* **198**, 135–140.
- Kobayashi, N.: 1980, Computer simulation of heat, mass and fluid flows in a melt during czochralski crystal growth, *Comput. Methods Appl. Mech. Eng.* **23**, 21–33.
- Kong, X., Everett, H. and Toussaint, G.: 1990, The graham scan triangulates simple polygons, *Pattern Recognition Letters* **11**, 713–716.
- Langlois, W.: 1977, Digital simulation of czochralski bulk flow in a parameter range appropriate for liquid semiconductors, *J. Crystal Growth* **42**, 386–399.
- Langlois, W.: 1982, A parameter sensitivity study for czochralski bulk flow of silicon, *J. Crystal Growth* **56**, 15–19.
- Langlois, W.: 1983, Czochralski bulk flow of silicon at large melt aspect ratio, *J. Crystal Growth* **63**, 67–69.

- Langlois, W.: 1985, Buoyancy-driven flows in crystal-growth melts, *Ann. Rev. Fluid Mech.* **17**, 191–215.
- Lawson, C.: 1977, *Software for C^1 surface interpolation*, Academic Press.
- Lecomte, M., Olivari, E., Van den Bogaert, N. and Dupret, F.: 1992, Fluid flow phenomena in crystal growth, *Eurochem. Colloquium No. 284*.
- Lee, M. J. and Reynolds, W.: 1985, Numerical experiments on the structure of homogeneous turbulence, *Technical Report TF-24*, Stanford University.
- Lennes, N.: 1911, Theorems on the simple finite polygon and polyhedron, *Amer. J. of Mathematics* **33**, 36–72.
- Li, Y., Imaishi, N., Akiyama, Y., Peng, L., Wu, S. and Tsukada, T.: 2004, Effects of temperature coefficient of surface tension on oxygen transport in a small silicon cz furnace, *J. Crystal Growth* **266**, 48–53.
- Li, Y., Li, M., Imaishi, N., Akiyama, Y. and Tsukada, T.: 2004, Oxygen-transport phenomena in a small silicon czochralski furnace, *J. Crystal Growth* **267**, 466–474.
- Lipchin, A. and Brown, R.: 2000, Hybrid finite-volume/finite-element simulation of heat transfer and melt turbulence in czochralski crystal growth of silicon, *J. Crystal Growth* **216**, 192–203.
- Lipchin, A. and Brown, R. A.: 1999, Comparison of three turbulence models for simulation of melt convection in czochralski crystal growth of silicon, *J. Crystal Growth* **205**, 71–91.
- L.J. Guibas, J. H., Leven, D. and Tarjan, R.: 1987, Linear time algorithm for visibility and shortest path problems inside triangulated simple polygons, *Algorithmica* **2**, 209–233.
- Machida, N., Hoshikawa, K. and Shimizu, Y.: 2000, The effects of argon gas flow rate and furnace pressure on oxygen concentration in czochralski silicon single crystals grown in a transverse magnetic field, *J. Crystal Growth* **210**, 532–540.
- Machida, N., Suzuki, Y., Abe, K., Ono, N., Kida, M. and Shimizu, Y.: 1998, The effects of argon gas flow rate and furnace pressure on oxygen concentration in czochralski-grown silicon crystals, *J. Crystal Growth* **186**, 362–368.
- Mansour, N., Kim, J. and Moin, P.: 1988, Reynolds stress and dissipation rate budgets in turbulent channel flow, *J. of Fluid Mechanics* **194**, 15–44.
- Medić, G. and Mohammadi, B.: 1999, Nsike—an incompressible navier-stokes solver for unstructured meshes, *Technical Report RR 3644*, INRIA–Rocquencourt. Info at <http://www.inria.fr/RRRT/RR-3644.html>.
- Meisters, G.: 1975, Polygons have ears, *Amer. Math. Monthly* **82**, 648–651.
- Menter, F. R.: 1992, Improved two-equation $k-\omega$ turbulence models for aerodynamic flows, *Technical Report 103975*, NASA.
- Moerschel, K., Pearce, C. and Reusser, R.: 1977, *The Electrochem. Soc.*, Pennington, NJ.

- Mucke, E. P., Saias, I. and Zhu, B.: 1996, Fast randomized point location without pre-processing in two- and three-dimensional delaunay triangulations, *In Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pp. 274–283.
- Mühe, A., Backofen, R., Fainberg, J., Müller, G., Dornberger, E., Tomzig, E. and von Ammon, W.: 1999, Oxygen distribution in silicon melt during a standard czochralski process studied by sensor measurements and comparison to numerical simulation, *J. Crystal Growth* **198**, 409–413.
- Narkhede, A. and Manocha, D.: 1995, *Fast Polygon Triangulation based on Seidel's Algorithm*, <http://www.cs.unc.edu/dm/CODE/GEM/chapter.html>.
- Nicodème, P.: 1990, *Simulation numérique des échanges thermiques dans un four et calcul du taux de dislocation dans un cristal d'arséniure de gallium*, PhD thesis, Université Catholique de Louvain.
- Nicodème, P., Dupret, F., Crochet, M., Farges, J. and Nagel, G.: 1988, In: Semi-insulating iii-v materials, *Proc. 5th Conf. Semi-Insulating III-V Materials*, pp. 471–476.
- O'Rourke, J.: 2000, *Computational Geometry in C, 2nd Edition*, Cambridge University Press.
- Owen, J. M. and Rogers, R. H.: 1989, Flow and heat transfer in rotating-disk systems, *Technical Report Vol. 1*, Research Studies, Taunton, Somerset, England.
- Owen, S. J.: 1998, A survey of unstructured mesh generation technology, *Proceedings 7th International Meshing Roundtable, Dearborn, MI, October 1998*, pp. 83–92.
- Pimpputkar, S. and Ostrach, S.: 1981, Convective effects in crystals grown from melt, *J. Crystal Growth* **55**, 614–646.
- Polezhaev, V.: 1984, *Hydrodynamics of heat and mass transfer during crystal growth*, in *Crystals: Growth, Properties and Applications* Eds. H.C. Freyhardt, A.A. Chernov and H. M. Krumbhaar, Springer-Verlag, Berlin.
- Pope, S.: 2000, *Turbulent Flows*, Cambridge University Press.
- Pulicani, J., del Arco, E. C., Randriamampianina, A., Bontoux, P. and Peyret, R.: 1990, Spectral simulations of oscillatory convection at low prandtl number, *Int. J. for Numer. Methods in Fluids* **10**, 481.
- R. Boux, E.: 1990, *Notes on Numerical Fluid Mechanics, Vol. 27. Numerical simulation of Oscillatory Convection in Low-Pr Fluids, A GAMM Workshop*.
- Rodi, W.: 1985, *Turbulence models for environmental problems, in prediction for turbulent flows*, Van Kármán Institute Books, Kolman Edition.
- Ruppert, J.: 1995, A delaunay refinement algorithm for quality 2-dimensional mesh generation, *Journal of Algorithms* **18**, 548–585.
- Ryckmans, Y.: 1989, *Simulation numérique de la croissance des cristaux semi-conducteurs*, PhD thesis, Université Catholique de Louvain.
- Saad, Y.: 2000, *Iterative Methods for Sparse Linear Systems, Second Edition*, Society for Industrial and Applied Mathematics.

- Sackinger, P., Brown, R. and Derby, J.: 1989, A finite element method for analysis of fluid flow, heat transfer and free interfaces in czochralski crystal growth, *Intern. J. Numer. Meth. Fluids* **9**, 453–492.
- Scheel, H.: 1998, Preface of the lecture notes, *ISCGT1*.
- Schewchuk, J. R.: 1997, *Delaunay refinement mesh generation*, PhD thesis, School of Computer Science, Carnegie Mellon University.
- Seidel, R.: 1991, A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons, *Comput. Geom. Theory Appl.* **1**, 51–64.
- Seidl, A.: 1995, *Über den Einflub der Tiegelrotation auf den Wärme und Stofftransport in Halbleiterschmelzen bei Czochralski - Anordnungen*, PhD thesis, Universität Erlangen.
- Shewchuk, J. R.: 1996a, Robust Adaptive Floating-Point Geometric Predicates, *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, Association for Computing Machinery, pp. 141–150.
- Shewchuk, J. R.: 1996b, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in M. C. Lin and D. Manocha (eds), *Applied Computational Geometry: Towards Geometric Engineering*, Vol. 1148 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 203–222.
From the First ACM Workshop on Applied Computational Geometry.
- Shiraishi, Y., Takano, K., Matsubara, J., Iida, T., Takase, N., N.Machida, Kuramotoi, M. and Yamagishi, H.: 2001, Growth of silicon crystal with a diameter of 400 mm and weight of 400kg, *J. Crystal Growth* **229**, 17–21.
- Sim, B., andi K.H. Kim, I. L. and Lee, H.: 2005, Oxygen concentration in the czochralski-grown crystals with cusp-magnetic field, *J. Crystal Growth* **275**, 455–459.
- Smagorinsky, J.: 1963, Generation circulation experiments with the primitive equations:i. the basic equations, *Mon. Weather Rev.* **91**, 99–164.
- Smith, A. and Cebeci, T.: 1967, Numerical solution of the trubulent boundary-layer equation, *Technical Report DAC 33735*, Douglas Aircraft Division Report.
- Speziale, C.: 1995, A review of reynolds stress models for turbulent shear flows, *Technical Report No 95-15*, NASA.
Info at <http://www.icas.edu/library/reports/rdp/95.html>.
- Stewart, M. and Weinberg, F.: 1972, Fluid flow in liquid metals : Experimental observations, *J. Crystal Growth* **12**, 228–238.
- Su, P. and Drysdale, R.: 1995, A comparsion of sequential delaunay triangulation algorithms, *Proceedings of the 11th Annual Symposium in Computational Geometry*, pp. 61–70.
- Tarjan, R. and Wyk, C. V.: 1988, An $o(n \log n)$ -time algorithm for triangualting a simple polygon, *SIAM J. Comput.* **17**, 143–178.
- Togawa, S., Izunome, K., Kawanishi, S., Chung, S., Terashima, K. and Kimura, S.: 1996, Oxygen transport from a silica crucible in czochralski silicon growth, *J. Crystal Growth* **165**, 362–371.

- Toussaint, G.: 1991, Efficient triangulation of simple polygons, *Visual Comput.* **7**, 280–295.
- Tucker, H.: 1970, The distortion of turbulence by irrotational strain, *Technical Report 70-7*, McGill University.
- U. Ghia, K. N. G. and Shin, C. T.: 1982, High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method, *J. Comput. Phys.* **48**, 387–411.
- Uberoi, M.: 56, Effect of wind-tunnel contraction on free-stream turbulence, *J. Aerospace Sci.* **23**, 754–764.
- Van den Bogaert, N.: 1993, *Simulation dynamique de la croissance dans un four Czochralski: Modèle global de calcul des échanges thermiques et de la forme du cristal sur une géométrie variable*, PhD thesis, Université Catholique de Louvain.
- Van den Bogaert, N. and Dupret, F.: 1997a, Dynamic global simulation of the czochralski process i: Principles of the method, *J. Crystal Growth* **171**, 65–76.
- Van den Bogaert, N. and Dupret, F.: 1997b, Dynamic global simulation of the czochralski process ii: Analysis of the growth of a germanium crystal, *J. Crystal Growth* **171**, 77–93.
- Vizman, D., Grabner, O. and Müller, G.: 2000, Three-dimensional numerical simulation of thermal convection in an industrial czochralski melt: comparison to experimental results, *J. Crystal Growth* **233**, 225–231.
- von Ammon, W.: 2004, *Silicon crystal growth*, in: *Crystal Growth – from Fundamentals to Technology*, Eds. G. Müller, J.J. Métois and P. Rudolph, Elsevier B.V.
- Waston, D.: 1981, Computing the n-dimensional delaunay tessellation with applications to voronoi polytopes, *The computer Journal* **24**, 167–172.
- Watanabe, M., Eguchi, M., Wang, W., Hibiya, T. and Kuragaki, S.: 2002, Controlling oxygen concentration and distribution in 200 mm diameter si crystals using the electromagnetic czochralski (emcz) method, *J. Crystal Growth* **237**, 1657–1662.
- Wilcox, D.: 1998, *Turbulence modeling for CFD, 2th Edition*, D C W Industries.
- Wouters, P.: 1985, *Simulation numérique des échanges thermiques et application à la croissance des cristaux semi-conducteurs*, PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve.
- Wu, L., Rolinsky, R., Van den Bogaert, N., Regnier, V. and Dupret, F.: 2003, Development of new meshing techniques for the dynamic simulation of bulk crystal growth processes, *Rare Metals* **22**, 27–33.
- Xu, L.: 1999, A statistical thermodynamic model for oxygen segregation during czochralski growth of silicon single crystals, *J. Crystal Growth* **200**, 414–420.
- Yeckel, A. and Derby, J.: 2005, *Computer modeling of bulk crystal growth*, in *Bulk Crystal Growth of Electronic, Optical, and Optoelectronic Materials*, John Wiley and Sons Ltd.

- Yen, C. T. and Tiller, W. A.: 1991, Oxygen partitioning analysis during czochralski silicon crystal growth via a dopant marker and a simple transfer function modeling technique i. rotation rate transients, *J. Crystal Growth* **109**, 120–126.
- Yen, C. T. and Tiller, W. A.: 1992, Oxygen partitioning analysis during czochralski silicon crystal growth via a dopant marker and a simple transfer function modeling technique ii. growth velocity and applied magnetic field transients, *J. Crystal Growth* **118**, 85–92.
- Zhang, T., Ladeinde, F., Zhang, H. and Prasad, V.: 1996, *Proceedings of the 1996 31st ASME National Heat transfer conference, Part 1.*, p. 17.