INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
NOTE TO USERS

This reproduction is the best copy available.

UMI®
QUALITATIVE GEOMETRIC REASONING FOR THERMAL DESIGN EVALUATION OF DIE CASTING DIES

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
The Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

by

Yuming Ma, B.S., M.S.

*****

The Ohio State University
2000

Dissertation Committee:
Professor Miller, R. Allen, Advisor
Professor Brevick, Jerald R.
Professor Castro, Jose M.
Dr. Kabiri-Bamoradian, Khalil

Approved by
Advisor
Graduate Program of
Industrial and Systems Engineering
ABSTRACT

In order to assess manufacturability effectively at the conceptual design phase, a quick and easy-to-use tool is needed by the die-casting designer. This dissertation presents a qualitative geometric reasoning method and tool which approximately evaluates the steady-state thermal characteristics of die-casting dies at the conceptual design stage. The qualitative geometric reasoning approach is used because the geometric information is the only major information available at the conceptual design stage and the casting geometry dominates the physical characteristics of the casting and the die. The reasoning algorithms are built on a voxel model for its robustness and insensitivity to geometric complexity.

The main idea of the geometric reasoning approach is to apply the circuit analog of the heat flow on a voxel model and build the heat flow network for the die. An approximate temperature distribution of the die is then obtained through geometric reasoning, heat balance, and thermal resistance computation on the voxel model. The resulting temperature field of the die is visualized using volume visualization techniques, and heat converging regions (hot spots) can be identified from the pattern. Thermal control features such as cooling line can then be placed according to the location of the heat converging regions and part thickness.

The approximate temperature field computed by qualitative reasoning can also be plugged into the numerical simulation to jump start the die casting thermal simulation. To
achieve a steady-state temperature field, the conventional numerical-based thermal simulation of die-casting starts from room temperature and goes through many start-up cycles. The qualitative temperature is computed based on the assumption that the die-casting system is already at the steady state and is equivalent to the average temperature within a cycle at the steady state. Thus, this qualitative temperature is much more closer to the actual steady state temperature distribution and having it as the initial temperature of the die will reduce the number of start-up cycles and hence save a great amount of simulation time.
To my mother Liu Hui-Fang, and to my father Ma Ying-Zhi
ACKNOWLEDGMENTS

The work with this dissertation has been extensive and trying, but in the first place exciting, instructive and fun. Without help, support, and encouragement from several persons, I would never have been able to finish this work.

First of all, I would like to thank my adviser, Dr. R. Allen Miller, for his support and guidance throughout the years at The Ohio State University. I do not feel that I can adequately express how much I have learned from him and how much I admire and respect his knowledge, vision and insight on the subject. Over the past several years during many, many times whenever I came to him with questions and frustration of the research, I always could get direct and specific supervision from him, and I was always encouraged by his persistence and deep understanding of the research. Working with Dr. Miller will become one of the most precious memories in my life.

I am very grateful to Dr. Khalil Kabiri for his work on my committee and the valuable comments and suggestions during the work with this dissertation. I am also very grateful to Dr. Jerald R. Brevick and Dr. Jose M. Castro for their for their work on my committee.

I will give a special thanks to Mr. Adham Ragab, my friend and fellow colleague, who has spent a lot of time and effort to help me to build the mesh model and run the numerical simulation. I also must thank my friends and teammates, Dehua Cui, Li Cai and many
other members of CastView Project for our collaboration, discussion, argument, and encouragement with each other and for various things too innumerable to recount. The time we had together plus CastView software will become a cherished memory in my life.

Last but not least, I owe a special debt of gratitude to my dearest wife, Jinmei Liao, who witnessed the whole process of this dissertation work with infinite patience, support, and love.
VITA

September 13, 1968.................................Born - Shaanxi, China

1990....................................................B.S. Applied Mechanics,
Beijing University of Aeronautics
and Astronautics.

1995....................................................M.S. Manufacturing Engineering,
Beijing University of Aeronautics
and Astronautics.

1990-1992...........................................Aerodynamic Engineer,
Xi’an Aircraft Company, China

1995-1995...........................................Lecturer,
Beijing University of Aeronautics
and Astronautics.

1996-present......................................Graduate Research Associate,
The Ohio State University.

PUBLICATIONS

[1] Yuming Ma, Adham Ragab, Khalil Kabiri, and R. A. Miller, “Qualitative Geometric
Paper No. DETC2000/DFM-14007.

FIELDS OF STUDY

Major Field: Industrial and Systems Engineering
# TABLE OF CONTENTS

| ABSTRACT | ii |
| DEDICATION | iv |
| ACKNOWLEDGMENTS | v |
| VITA | vii |
| LIST OF FIGURES | xii |
| LIST OF TABLES | xvii |

## CHAPTERS:

1 INTRODUCTION 
   1.1 Problem Description 
   1.2 Research Objectives 
   1.3 Research Approach and Issues 
   1.4 Thesis Organization 

2 REVIEW OF RELATED RESEARCH 
   2.1 Introduction 
   2.2 Design for Manufacturing 
   2.3 Physical Analog Approach 
   2.4 Numerical Simulation Approach 
   2.5 Qualitative and Geometric Reasoning 

3 ANALOGICAL MODELING OF HEAT FLOW 
   3.1 Introduction 
   3.2 Heat Flow in Die-casting Systems 
   3.3 Geometric Representation 
   3.4 Heat Flow Modeling 
      3.4.1 Heat Flow from Casting to Die 

ix
5.6 Summary ................................................................. 123

6 JUMP START OF NUMERICAL SIMULATION ......................... 124
  6.1 Introduction ......................................................... 124
  6.2 Numerical Simulation Approach .............................. 126
    6.2.1 Heat Transfer in Die Casting ........................... 127
    6.2.2 Simulation Procedure ................................. 132
  6.3 Data Transfer From Voxel to Mesh Model .................. 135
    6.3.1 Orientation between Voxel and Mesh Model .......... 136
    6.3.2 Resolution between Voxel and Mesh Model ........ 139
  6.4 Simulation Process and Result Comparison ................ 148

7 CONCLUSION AND RECOMMENDATION ................................ 157
  7.1 Summary and Contribution .................................... 157
  7.2 Future Recommendations ..................................... 159

REFERENCES ................................................................. 161
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Illustration of section modulus calculation</td>
<td>18</td>
</tr>
<tr>
<td>2.2. STL model of the lock example input to CastView</td>
<td>21</td>
</tr>
<tr>
<td>2.3. Thickness pattern in the voxel model of the lock example</td>
<td>22</td>
</tr>
<tr>
<td>2.4. Flow pattern in the voxel model of the lock example</td>
<td>23</td>
</tr>
<tr>
<td>3.1. Die casting cycles and temperature variation</td>
<td>26</td>
</tr>
<tr>
<td>3.2. Total heat flow in a die-casting system</td>
<td>27</td>
</tr>
<tr>
<td>3.3. Illustration of transforming a CAD model to STL and to voxel model</td>
<td>29</td>
</tr>
<tr>
<td>3.4. Circuit analog modeling of heat flow from casting to die</td>
<td>31</td>
</tr>
<tr>
<td>3.5. Circuit analog modeling of heat flow between die and cooling line</td>
<td>33</td>
</tr>
<tr>
<td>3.6. Circuit analog modeling of heat flow between a die and the environment</td>
<td>35</td>
</tr>
<tr>
<td>3.7. Illustration of the heat sections used in the traditional thermal analysis method</td>
<td>36</td>
</tr>
<tr>
<td>3.8. Heat flow front surface and its discretization</td>
<td>37</td>
</tr>
<tr>
<td>3.9. Illustration of the voxel neighborhood and the rays from it</td>
<td>39</td>
</tr>
<tr>
<td>3.10. Total heat flow path network for a die voxel in 2D</td>
<td>41</td>
</tr>
<tr>
<td>3.11. Temperature gradients through the depth of the die (Booth, 1981)</td>
<td>46</td>
</tr>
<tr>
<td>3.12. Temperature profile on the interface during the entire cycle</td>
<td>47</td>
</tr>
</tbody>
</table>
6.3. Procedure of jump start the numerical simulation.................................136
6.4. Transformation between voxel model and mesh model...........................138
6.5. Element geometry and dual polyhedron of node.................................141
6.6. Imperial algorithm finding the voxels inside the Voronoi cell................144
6.7. Illustration of the main control algorithm.........................................147
6.8. Illustration of the flat plate casting and the die configuration...............151
6.9. Comparison of the simulation results with or without the jump start........152
6.10. Comparison of the simulation results with or without the jump start....153
6.11. Comparison of the simulation results with or without the jump start.....154
6.12. Comparison of the simulation results with or without the jump start....155
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Thermal conductivity, liquidus and solidus temperature of Zinc alloy.</td>
<td>61</td>
</tr>
<tr>
<td>4.2. Thermal conductivity, liquidus and solidus temperature of Magnesium alloy.</td>
<td>62</td>
</tr>
<tr>
<td>4.3. Thermal conductivity, liquidus and solidus temperature of Aluminum alloy.</td>
<td>62</td>
</tr>
<tr>
<td>4.4. Thermal conductivity of H-13.</td>
<td>62</td>
</tr>
<tr>
<td>4.5. Thermal conductivity of AlSi 4140.</td>
<td>63</td>
</tr>
<tr>
<td>4.6. Data for the 26 rays from a voxel.</td>
<td>76</td>
</tr>
<tr>
<td>5.1. Model size and running time for the four cases.</td>
<td>114</td>
</tr>
<tr>
<td>5.2. Quantitative comparison for case-2.</td>
<td>116</td>
</tr>
<tr>
<td>5.3. Cooling number and location vs. surface temperature variation.</td>
<td>121</td>
</tr>
<tr>
<td>6.1. Heat transfer coefficient between the die casting components (W/m² °C).</td>
<td>148</td>
</tr>
<tr>
<td>6.2. Time steps in each cycle.</td>
<td>149</td>
</tr>
<tr>
<td>6.3. Configuration of the computer and benchmark of the computation time.</td>
<td>156</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Problem Description

The die casting process is one type of net-shape manufacturing (NSM) process in which the liquid metal is injected into the die cavity under high pressure, solidifies, and is then ejected out of the die as a finished part. In addition to being a mold for the casting shape, a die-casting die is also a cooling media which extracts heat from liquid metal and cools the liquid, allowing it to solidify. To maintain an economical production rate and obtain desired casting quality, the die must rapidly extract heat from casting regions where late solidification could cause porosity, and also pump heat to casting regions where early solidification could obstruct complete filling of the cavity. Long cycle time operation runs successfully with uniform cavity surface temperature, but short cycle time operation may not produce such a condition. The ideal solution depends on the operation conditions and production rate (Herman 1974). Cooling lines and heating lines are used in the die to control the temperature and achieve the ideal condition. The design of cooling lines and heating lines involves thermal analysis of the die and casting, which plays a critical role in the design of the casting and die.

In die-casting production or any other net shape manufacturing (NSM) production, the part plays a critical role to the success of the final product. For a given production pro-
cess, a good or successful part design not only means the design perfectly satisfies the functional requirements, but also means the part design satisfies the life cycle requirements such as the manufacturability. For the die-casting process, the major impact of part design to the manufacturability is the compatibility of part design to the design of dies. The major features of the die are determined by the part geometry, such as parting lines and parting planes, cooling/heating channel locations, ejection pin location. Thus, the success of the part design largely determines the success of the overall competitiveness of the product.

It has been estimated that as much as 80% of the development and manufacturing cost is pre-determined at the conceptual design stage (Dorf and Kusiak, 1994). A good design with minimum turn back would save a significant amount of investment of reworks; however, bad design normally raises the final cost by both reworks and delay to market. To the die-casting industry, this issue is even more critical because the die design is completely driven by the geometry of the part, and the die design and manufacturing consume a substantial amount of investment (from $50,000 to $5,000,000 depending on the size of the dies). Any failure in the part design stage would result in a major investment loss. Thus, designing the part correctly from the start is essential in the die casting industry.

Design for Manufacturing (DFM) has emerged as a successful methodology that brings the manufacturing consideration to the design stage and helps the designer to design the product correctly from the beginning. DFM is especially important in the die-casting domain since one of the largest investments in the manufacturing of casting is the die, and the part design largely determines the die design, which in turn significantly affects the cost and quality of the resulting castings. In addition to the functional requirement, the part
geometry is the determining factor for all subsequent decisions in the die-casting production process.

Considering DFM in terms of thermal design, the designer should be able to evaluate the thermal characteristics of the design at the conceptual stage. The fundamental questions which have to be addressed by designers are: 1) what is the overall die temperature distribution? 2) where are the hot spots in the die through which the cooling lines should go? and 3) what are the reasonable locations of the cooling lines?. To answer these questions, the straightforward solution is to run a numerical simulation of the thermal process of die casting and determine the temperature distribution of the dies. However, the major shortcoming of using the numerical method at the conceptual design stage is that the numerical simulation of the thermal process for die casting is typically very time consuming. Furthermore, numerical simulation is sensitive to process parameters which are usually unclear at the conceptual design stage. Hence, there is a need for an alternative solution and tool to support the thermal evaluation. Such a tool should work interactively so that the designer can try different design configurations in a short period of time. This tool can only work in a qualitative fashion because the detailed process knowledge is incomplete at the conceptual design stage. The geometric reasoning approach is adopted because the part geometry is the only major information available at the conceptual design phase and also because the geometric reasoning algorithms are fast enough to support interactive evaluation.
1.2 Research Objectives

The primary objective of this study is to develop a qualitative methodology and tool to evaluate the thermal characteristics of the die-casting die during the conceptual design stage. Qualitative geometric reasoning has proved useful and successful in die-casting design evaluation by the CastView project and software. This research is an extension of CastView, which emphasizes thermal design evaluation, especially temperature distribution and cooling line location evaluation. The qualitative reasoning approach and tool are not expected to produce the same accurate results as numerical simulation. However, this approach is expected to provide a non-misleading ball-park approximation of the temperature field of the dies which have the major thermal features to support conceptual design evaluation. The running time is expected to be in the order of minutes so that the designers can evaluate the design in an interactive fashion.

This study does not try to replace regular numerical simulation (which is necessary for final design verification), but provides an alternative approach to numerical simulation, which can be used to assist numerical simulation. This sets another objective of investigating the possibility and approach to jump start the numerical thermal simulation of die casting by using the qualitative reasoning results. The die casting process is measured by cycles. Each cycle includes operations of die opening, metal injection, solidification, and part ejection. To obtain an accurate result, numerical simulation of the thermal process of the die casting process has to start from an initial temperature (normally room temperature) and go through many cycles to reach the equilibrium state, at which the total heat flow is balanced. Because the qualitative temperature approximates the average temperature of a
cycle at the equilibrium state, we are expecting the numerical simulation to take fewer cycles and the computation time to converge to the equilibrium state.

1.3 Research Approach and Issues

The part and die are represented using the voxel model which is a collection of uniform-sized cubes in 3D space. On the voxel model, each of the voxel can be viewed as a thermal resistor and the connectivity among the voxels defines a thermal network in the die-casting system. The main idea of this qualitative geometry reasoning approach is to apply the circuit analog of heat flow on a voxel model and build the heat flow network for each voxel in the die. An approximate temperature distribution throughout the die is then obtained mainly based on geometric computation. The result is visualized using volume rendering techniques, and the heating converging regions (hot spots) can be identified from the temperature pattern. Thermal control features can then be placed according to the location of the heat converging region and part thickness.

The qualitative reasoning is based on the assumption that the die-casting process is in the steady state mode at which the total amount of heat going into the die material is balanced with the total amount of heat going out of the die material within a cycle. The qualitative temperature of the die, which is equivalent to the average temperature of a cycle at the steady state, is computed and represented on the voxel model. To jump start the numerical simulation process, the temperature data on the voxel model is transferred to the mesh model as the initial condition of the dies. This data transferring and jump-start process is
investigated and tested on the mesh model created in I-DEAS and the numerical solver of ABAQUS.

The issues in this research include:

1. Applying the circuit analog method of heat flow on the voxel model and modeling the heat flow for the die-casting system.
2. Developing a method to build a heat flow network for each of the die voxel and solving the network to determine the die temperature distribution.
3. Developing a geometric reasoning approach for thermal resistance computation on the voxel model to compute die temperature.
4. Investigating an approach to jump start the numerical simulation with qualitative results.
5. Developing algorithms and programs to transfer temperature data from the voxel model to the mesh model to jump start the numerical thermal simulation.
6. Verifying the qualitative reasoning approach and results.

1.4 Thesis Organization

This section describes the organization of the remainder of this dissertation. Chapter 2 reviews the related research, including general design evaluation and design for manufacturing (DFM) methodology, physical analog and numerical simulation approach for thermal analysis of die casting, the knowledge-based approach, and the geometric reasoning approach. Chapter 3 presents a modeling method of heat flow between the die-casting components on the voxel model using circuit analogy. This chapter forms the basis of the
analogical and geometric reasoning method. Chapter 4 describes the geometric reasoning algorithms for thermal resistance computation on the voxel model, including the composition of each type of thermal resistance, distance computation from the die voxels to the thermal boundaries of casting surfaces and die surface and distance from die voxels to the cooling lines. Chapter 5 presents the verification of this qualitative reasoning method by comparing the results of various cases with the results from numerical simulation. Chapter 6 discusses the idea and approach to jump start the numerical simulation die-casting thermal process using the qualitative results as an initial condition. The contributions and recommendations of this study are summarized in Chapter 7.
CHAPTER 2

REVIEW OF RELATED RESEARCH

2.1 Introduction

There have been many studies on general design for manufacturing (DFM) and thermal design analysis for die-casting particularly. DFM forms the macro framework of this research. DFM has progressed into many directions and only the generic methodology and the branches closely related to Net Shape Manufacturing (NSM) are reviewed in this chapter. The physical analog approach and numerical simulation approach are the two common ways for perform die-casting thermal design evaluation. The circuit analog method inspired the initiative of this research to apply a similar method to a computer-based model rather than building physical devices. The generic idea and approaches of the simulation procedure are reviewed and the usefulness of such approaches is evaluated in terms of manufacturability assessment for conceptual design. Finally, geometric and qualitative reasoning, which forms the basis of this research, is reviewed in detail, especially the volume-base geometric reasoning method used in the CastView software. The existing algorithms and results in CastView will be directly used in this research.
2.2 Design for Manufacturing

Design for manufacturing is an implementation of concurrent engineering in the design stage. The emergence of concurrent engineering and DFM is meant to cut down the lead time and improve quality, more specifically, to have the design utilize the production and manufacturing facility correctly (Huang, 1996; Vliet and Luffervelt, 1999; Shah and Wright, 2000). A good design not only meets the functional specifications or optimizes the production, it should rather be “designed for everything”, i.e. to take account of manufacturing, assembly, quality, consumer satisfaction, distribution, destruction, the environment, and numerous other factors. This is not a trivial problem since it involves so many people and so many different interests. One comprehensive term has been adopted by the design community: DFX, design for X, where X is all the other activities directly or indirectly affected by the design as listed above (Huang, 1996).

In design for manufacturing, two fundamental questions arise concerning every type of manufacturing process: 1) Given an object, can it be fabricated using a particular process?; and 2) Given the particular process, what is the best way to construct the object to be compatible with the process? (Ishii and Miller, 1992). Answering these two questions require that the designers not only analyze the design itself but also the manufacturing processes so as to integrate the process specialties into the design.

Early studies on DFM were focused on assembly and machining processes. Boothroyd and Dewhurst describe a general design for manufacturing and assembly (DFMA) method which the designer could use to achieve the optimum product from the assembly viewpoint (Boothroyd and Dewhurst, 1990; Boothroyd, 1994, 1996). They developed a set
of design analysis tools which aim to help companies take full advantage of manufacturing processes that exist and keep the number of parts in an assembly to the minimum. Their method achieves this by enabling the analysis of design ideas, modulizing and integrating the part design as much as possible.

Due to the specialty of the net shape manufacturing process including die-casting, consideration of the manufacturability assessment of this type of process has been focused on moldability, including parting direction, parting line and surfaces, undercuts, draft surfaces, and mold orientation. The ultimate objective is to construct the mold or die automatically or at least semi-automatically with limited interaction from designers. Concerning the manufacturing features such as draw direction, undercuts and parting line in die-casting or injection molding processes, recognizing or identifying these features from the geometry of the part is the major challenge.

Theoretical analyses of moldability using geometric and computational-based reasoning were attempted by Bose and Toussaint (1994, 1995), and de Berg et al., (1997). These studies were based on either 2D polygon or 3D polyhedron models. However, the practical problems are far more complicated than the simplified situation used in such analysis. Researchers have investigated rules or decision criterion for parting direction and parting line selection, and have developed computer-based solutions for sand casting and die casting processes (Ravi and Srinivasan, 1989; Guleyyupoglu, et al., 1994). A rule-based heuristic search approach was used for parting direction and parting line selection (Ganter and Tuss, 1990; Hui and Tan, 1991). As is true for any heuristic search technique, an optimum solution can not be guaranteed by the such approaches. A deterministic method, on the other hand, was studied by researchers to find the optimum solutions for mold and die
design problems (Chen et al., 1993, 1995; Weinstein and Manoochehri, 1997). Different from the heuristic and deterministic approach, the feature-based approaches of DFM for net-shape manufacturing were developed to automate the manufacturing feature recognition process. Some techniques recognize features from a solid geometric model (Gadh and Prinz, 1992), while others covered components from a feature-based design representation to a manufacturing representation (Ishii et al., 1989; Shah et al., 1990; Rosen et al., 1992, 1994). A comprehensive feature-based manufacturability assessment approach particularly for die-casting or injection molding processes was reported by Yueh (1995) and Ren et al. (1995) which evaluates the die opening direction, parting line, and undercuts interactively in a CAD system. Liou and Miller (1990) proposed a knowledge-based approach to die casting design which analyzes the shape of the casting according to manufacturability and provides re-design suggestions.

As reviewed above, DFM for die-casting or for a general net-shaped manufacturing process mainly focuses on identifying die design components including draw direction, parting line, parting surfaces, and undercuts for the end to automatically construct the die or mold based on the part geometry. Geometric reasoning for the details of the die or more process related issues, such as cooling line, ejection pin, and hot spots are another category of design factors which have not been fully addressed by researches. This paper is aimed particularly at the thermal characteristics of the die and cooling line location assessment.
2.3 Physical Analog Approach

The Society of Die Casting Engineers (SDCE) published a Water Line Calculator which was used to calculate the distance of the cooling channels below the casting surface (Groenveld, et al., 1975, Tuten, et al., 1979, and Doyle, et al., 1981). The calculator solves the steady-state heat transfer equation on parallel heat flow paths with a single correction for non-parallel heat flow. Since such neat conditions do not actually exist in a die, the application of this calculator is limited to simple cavity shapes with planar surfaces.

The General Motors Company developed an electrical analog technique for cooling line design (Ruhlantd, 1967; and Szakacs, 1987). This method is based on the similarity of Ohm's Law for current and Fourier's Law for heat flow, as shown in Eq. (2.1) and Eq. (2.2):

\[
I(\text{current in Amperes}) = \frac{\Delta V}{R} = \frac{\text{Voltage difference in Volts}}{\text{Resistance } \left(\rho \frac{L}{A}\right) \text{ in Ohms}} \quad (2.1)
\]

\[
Q(\text{Heat flow in Watts}) = \frac{\Delta T}{R_t} = \frac{\text{Temperature difference in } ^\circ C}{\text{Thermal resistance } \left(\frac{L}{Ak}\right) \text{ in } ^\circ C/W} \quad (2.2)
\]

The casting and die are first sliced into many 2D sections according to the geometric configuration of the casting. The section view of the casting and the die are then drawn on electrically resistive Teledeltos paper. The cavity outline is painted with highly-conductive silver paint and charged with high voltage which is equivalent to the temperature of the
casting section. Water lines are simulated with steel contacts and charged with voltage equivalent to the water line temperature. The applied voltages cause the current to flow through the Teledeltors paper like heat would flow through the die. The desired voltage of the cooling lines is calculated so that such voltage could balance the heat flow in the die. The waterline contacts can be moved about on the paper until the measured voltages equal the calculated values, thus enabling the designer to know that the waterline is correctly placed.

Another type of thermal control design and analysis devices are reported in Booth (1981) and Edscher (1981). These studies focused on evaluating the cooling media type and flow rate of cooling media. Mini computer programs were also developed to assist the data acquisition and analysis via hardware devices.

All of such specially designed analog equipment make the utilization of this method very expensive. Furthermore, this method is not a computer-based tool, which limits the application in the era of information technology. However, this electrical analog method provided the motivation of this research: to build a computer-oriented software solution to the thermal design and analysis of die-casting dies.

2.4 Numerical Simulation Approach

The tasks of the thermal analysis of the die-casting design include heat flow analysis, fluid flow analyses, solidification kinetics and thermal-mechanical analyses. Among these subjects, heat flow and solidification are the two dominant phenomena for thermal design evaluation. Heat flow analysis aims to find the macroporosity (isolated hot spots in
the casting) and microporosity (empirical criteria, e.g. Niyama) and the dendrite-arm spacing of the casting metal. The solidification kinetics aims to find the grain size of the casting via nucleation and growth laws and the secondary dendrite-arm spacing and solidification pattern. The most commonly used method in thermal analysis is numerical simulations, including the Finite Element Method (FEM), Finite Difference Method (FDM), and Boundary Element Method (BEM). Many specialized software packages based on such methods are commercially available, such as the FEM package ProCast\(^1\), the FDM packages AFS-SOLID\(^2\) (Finite Solutions, 1997) and MAGMA\(^3\), and the BEM package DMT-Casttherm\(^4\). Generic FEM packages such as ABAQUS\(^5\) are also used for thermal and mechanical analysis of die casting.

Studies on the numerical simulation of die-casting thermal process including heat flow, and solidification and cooling are very numerous. The papers referenced here are examples meant to illustrate the breadth of the research, not to exclude other important works in the field.

The generic procedure of the numerical simulation of a casting thermal process is used to solve the heat transfer equation on a geometric mesh model with certain pre-defined boundary and initial conditions (Berry and Pehlke, 1982; Campbell, 1992). The die casting process is measured by cycles. Each cycle starts from closing a die, injecting liquid metal into the cavity, metal solidification, ejection, and spraying the contacting surfaces of the

---

dies, and repeats. To get a precise and useful result, the simulation must start from the initial temperature and go through many cycles to achieve the steady state.

Different software package or users make different assumptions according to their own application domain to simplify the comprehensive heat transfer equation and make it solvable efficiently. The early studies strived to solve the heat transfer on 2D shapes with steady state assumption (Thukkaram, 1970; Granchi, et al., 1983; Singh, 1986; Smiley, 1980, 1988). Powered by the advancement in geometric modeling and computer technology, more sophisticated 3D shapes with complete boundary and initial conditions can be solved now using commercial packages. Some studies concentrate on solidification analysis using finite difference method (Schmidt, 1989; Smiley, 1993; Schmidt and Smiley, 1993; Schmidt, 1997), or cooling line evaluation through boundary element method (Siauw and Nguyen, 1989; Siauw, et al., 1991), or overall thermal analysis using the finite element method (Barone and Caulk, 1993).

For casting with complex shape and complete boundary conditions, the time and effort required for the numerical simulation to reach the steady state is numerous. Methods have been developed to increase the efficiency of the simulation without sacrificing too much accuracy. Barone and Caulk (1993) proposed another solution which is capable of predicting the periodic die temperature at steady state without solving for the start-up transient. Rosidale and Davey (1998) presented a number of novel techniques that improve the efficiency and performance of a steady state thermal model used for die-casting thermal simulation using the boundary element method.

Though such techniques can improve the efficiency of the simulation, in order to successfully run the numerical simulation and get reasonable results from it, the simulation
can not avoid the start-up cycles, which still take a great deal of time. Moreover, the
designer also needs to have a significant amount of knowledge of the manufacturing pro-
cess, thermodynamics, fluid mechanics and simulation know-how. Designers also need
time and effort for preprocessing, simulation, and comprehension of the results. Furthermore, at the early design stage, many process parameters remain uncertain, which forces the designer to speculate about sensitive input values. Because of the complexity of the simulation and uncertainty of the process parameters, The CastView research group argues that the numerical simulation is still not a practical approach to help designers evaluate the design interactively, especially in the design. Numerical-based tools are better suited for post-design verification than for interactive design evaluation (Lu, et al., 1995, 1996, and 1997; Miller, 1996; Rebello, 1997a).

2.5 Qualitative and Geometric Reasoning

Geometric and qualitative reasoning approaches have been developed to facilitate the conceptual design evaluation and manufacturability assessment. The validity of the geometric and qualitative reasoning relies on two aspects: 1) the geometry is the only major information available at the conceptual design stage; 2) the manufacturing process knowledge is incomplete in the early design phase. Furthermore, the geometric and qualitative reasoning are efficient in terms of running time so that it can support interactive design evaluation. This is very important for conceptual design because the designers are expected to try the alternatives in a short period of time. Such issues of the geometric and qualitative reasoning also define the initiative of this research.
The section modulus approach was developed to predict the solidification macrostructures of casting alloy (Wlodawer, 1966). The application and extension of the section modulus approach are reported in Heine and Uicker (1984, 1985), Neises and Uicker (1987), DeKalb, et al. (1987), Ravi and Srinivasan (1989), Kotschi and Plutshack (1989, 1991), Upadhya and Wang (1992), and Upadhya, et al. (1994). The term “modulus” means the ratio of volume over surface area in 3D space or surface over perimeter in 2D space. The fundamental basis of this approach for the relationship between the solidification time and section modulus is given by Chvoinov’s Rule (Chvorinov, 1940):

\[ t = C \left( \frac{V}{A} \right)^2 \]  

(2.3)

Where \( t \) is the solidification time for a point in the casting section, \( C \) is a constant for a given metal-mold material and mold temperature (the details of this constant can be found in Upadhya, et al. (1994)), \( V \) is the volume of the casting section, and \( A \) is the surface area of the casting section. The ratio of volume over area defines section modulus for a point in the casting:

\[ M = \frac{V}{A} \]  

(2.4)

Thus, the solidification time is proportional to the square of the section modulus. Having the section modulus at every point in the casting, the solidification sequence of the casting alloy can be simulated accordingly. Several algorithms have been developed to cal-
culate the section modulus for 2D and 3D shapes (Neise, 1987; Upadhya, 1992). The calculation is based on the concept of distance from the mold surface, given by the following:

\[ M = \frac{2}{N} \sum \frac{1}{D_i} \]  

(2.5)

Where

\( N \): is the number of cooling directions. \( N \) should be chosen such that the heat transfer is simulated in three dimensions, e.g. the six principal directions along the orthogonal axes, or it can be chosen as 26 which are the neighboring directions of a three dimensional cube (Upadhya, et al., 1994).

\( D_i \): is the distance from the mold in direction \( i \).

The graphical illustration of the above Eq. (2.5) is shown in Figure 2.1.

![Figure 2.1. Illustration of section modulus calculation.](image-url)
The use of the section modulus for the prediction of solidification contours in arbitrary shape castings is an extremely difficult task. The calculation of the section modulus for a point requires the solidified volume at the same time that the solidification front reaches this point. But this is the information that is being sought from the analysis. This makes the section modulus calculation become a paradoxical problem (Lu, 1996). However, this method attracts researchers for the simple yet strong physics basis of Chvojinov's Rule.

The Volume-based geometric reasoning approach is used in the CastView project to develop a front-end tool for design evaluation and die-castability assessment in terms of flow and solidification related problems. The general procedure of the approach is to take the STL model of the casting generated from any CAD system and voxelize it into the voxel model (Ouyang, 1994), then apply the geometric reasoning algorithms on the voxel model to compute the geometry characteristics correlated with the casting problems (Lu and Miller, 1995; Lu, 1996; Lu, et al., 1997; Rebello, 1997; Rebello, et al., 1997). The result is presented to the designer through volume visualization (Kauffinan, 1990; Yagel and Kaufman, 1992; Yagel and Ciula, 1994). Though the results are qualitative, they are proved to be significant and not misleading to the conceptual design problems of die-casting (Miller, et al., 1997).

A 3D voxel model is used because it describes the volumetric information of casting, which is more suitable to reveal the die casting process characteristics. This approach highlights potential design flaws related to the die filling and part solidification as the part details are being defined. In contrast to the numerical method which solves the heat transfer equations, the volume-based approach addresses the thermal problems of the casting by
identifying the wall thickness of the part and the die by using a "peel-onion" concept on the voxel model. A distance transformation (DT) algorithm is used to compute the shortest distance from a voxel to the part boundary. Based on this, thick sections can be identified by examining voxels with a relatively large distance value. The thick section corresponds to portions of the part which may solidify later and may have a porosity problem. The ideal result is when the wall thickness of the casting is uniform. Lu compared the thickness information computed by his distance transformation method with the cooling curve obtained from the numerical simulation and found that they match very well with each other. This indicates that the continuous displaying of the thickness of the voxel can animate the solidification pattern or process of the casting (Lu, Miller, et al., 1997). This thick section is also important to the die designers for placing the cooling line. The thick sections contain more heat than the thin sections, which should be balanced by cooling devices. Cooling lines are normally placed near the thick sections to extract heat from them efficiently (Herman, 1974). The following figures show the STL model of the casting (Figure 2.2), thickness result (Figure 2.3), and filling pattern result (Figure 2.4) of a "lock" shape casting part.

A thinning algorithm (also called a skeleton algorithm) is used to identify the thin sections of the part and die. The thin regions of the part or die are extracted by identifying the medial surface voxels associated with small values of the distance to part-die interfaces. Together with the filling analysis (Rebello, 1997), the thin region identification in the part reveals potential filling problems in the cavity. Very thin regions of the die may indicate that the die is having a cooling or filling problem.

The volume-based geometric reasoning approach forms the basis of this research and the approach presented in this thesis is an extension of this method which intends to
compute the approximate temperature distribution of the die in the voxel model. This die temperature pattern, together with the thickness of the casting voxel, will provide information for designers to identify the thermal features for cooling line placement and furthermore to evaluate the design conceptually.

Figure 2.2. STL model of the lock example input to CastView.
Figure 2.3. Thickness pattern in the voxel model of the lock example.
Figure 2.4. Flow pattern in the voxel model of the lock example.
3.1 Introduction

The complete die casting system includes the die casting machine, furnace, die block, cooling system, spray system, and the operation environment. In this research, we are interested in the temperature distribution of the die block which is mainly affected by the casting section and cooling line, thus the die casting system is defined to only include the casting, die block and cooling line in this research. Henceforth, the die system refers to the combination of die block, casting section, and cooling line.

Because of the similarity of Ohm's Law for electrical current flow and Fourier's Law for heat flow, heat flow and current flow shares similar physics property and governing equation. Circuit analog has been a standard textbook-like method to model and analyze heat flow (Paschkis, 1973; Incropera, 1985). As reviewed in Chapter 2, the early application of the circuit analog of heat transfer in the die-casting system led to the hardware devices for cooling line design (Ruhlandt, 1967; Geza, 1987). On the other hand, Lu et al. proposed a method to use the voxel model for geometry reasoning of the die casting characteristics (Lu, 1996; Rebello, 1997; Miller, 1997), which has been implemented in CastView™. In this chapter, we apply the circuit analog method to the voxel model and
qualitatively model the heat flow in the die casting system and solve for the die temperature distribution.

This chapter is divided into four sections. The first section describes the total heat flow in the die casting system. The geometric representation, i.e. the STL and voxel model are explained briefly in the second section. The third section discusses the analogical modeling and solving approach in detail. The assumptions and justifications of the approach are discussed in the last section.

3.2 Heat Flow in Die-casting Systems

In the die casting system, the major heat source is the casting metal. The other heat source is the heating media if the die-casting system has heating lines. The heat sinks are the cooling line and environment.

An important concept in the die casting process is “cycle”. At the beginning of the operation, the die is closed, the liquid metal is then injected into the die cavity, it solidifies in the cavity and is ejected out of the cavity, the die is then sprayed and ready for the next shot. One repeat of these operations is called a “cycle”, which indicates a complete operation of the die casting process.

In the die casting process, the input of heat to the die is intermittent. During each cycle, heat is pumped into the die from the casting metal and heating lines, and extracted out by die material, cooling lines and environment. When the heat flow in the system reaches a balanced state, the die no longer stores heat in its material and the amount of heat input to the die from the heat sources equals the amount of heat output from the die to the
heat sinks. However, the temperature profile still varies at different steps of a cycle due to the cyclic heat input from the casting, as shown in Figure 3.1. Thus, this balanced state is also called cyclic steady state or quasi-equilibrium state.

![Figure 3.1. Die casting cycles and temperature variation.](image)

Heat flow is analogous to electric current flow, and can be analyzed by the circuit model. An electric analog of the heat flow in a die casting system is shown in Figure 3.2.
Though convection and radiation occurs between casting and die, they are small enough to be neglected in approximate computations (Thukkaram, 1970). Furthermore, to simplify the model, die accessories, such as inserts and holders are also assumed to be one metal monolithic block and the heat transfer coefficient between them is not considered. Hence, this qualitative approach considers only the conductive heat flow between casting and die, convection between die and cooling/heating media, and the heat transfer between a die and the environment.

This qualitative approach assumes that the process is already in the equilibrium state and the heat transfer is in the steady-state mode. The approximate temperature corre-
sponds to the average temperature of each cycle once the process reaches quasi-equilibrium.

3.3 Geometric Representation

This thermal design evaluation tool is part of the CastView software, thus it follows similar working process as other modules of CastView. The system takes a stereolithography (SLA or STL) model of the part, and transforms it into a voxel model. The STL model comes from the CAD system on which the designer designs the basic geometric layout of the part. The strategic model transformation process is illustrated in Figure 3.3.

The main reason for using the STL model as the input of the system is its neutrality and simplicity. The STL model is a collection of triangles which forms a facet approximation of the original geometry of the part. Each triangle is represented by three vertices and a normal vector pointing outside of the object volume. The model is purely geometric and does not contain any topological information. The model is vendor and system independent, which makes it very robust and neutral for using across platforms.

The voxel model is a collection of uniform cubic cells on which the evaluation is based. Many algorithms have been developed for transforming the facet model (such as STL model) into the voxel model (Kaufman and Shimony, 1986; Foley, et al., 1990; Cohen and Kaufman, 1991). The voxelization algorithm used in CastView was developed by Ouyang (1994). Because the thermal evaluation needs the volume of the die block, the voxel model is extended to include the die volume according to the die sketched on the STL mode. The die sketch and die volume definition are discussed in Chapter 4.
The voxel model has many advantages especially for this type of application. Many of the phenomena of die casting are volumetric, such as solidification, filling, and temperature distribution, which can be well addressed by a voxel model. The uniformity of the voxel representation enables the reasoning algorithms not to be sensitive to the shape complexity. Furthermore, the computational and visualization techniques for the voxel model are rich and available to use, such as distance transformation and volume rendering (Lu, et. al. 1995, and Lu, 1996).

**Figure 3.3.** Illustration of transforming a CAD model to STL and to voxel model.
3.4 Heat Flow Modeling

As stated in the introduction of this chapter, there is a similarity between Ohm's Law for current flow and Fourier’s Law for heat flow, as the following Eq. (3.1) and Eq. (3.2) shows:

\[ I = \frac{\Delta V}{R} \]  
\[ Q = \frac{\Delta T}{R_t} \]

Where:

- \( I \): is the current, in Amperes.
- \( \Delta V \): is the voltage difference, in Volts.
- \( R \): is the electric resistance, in Ohms.

- \( Q \): is the heat flow rate, in Watts.
- \( \Delta T \): is the temperature difference, in K.
- \( R_t \): is the thermal resistance, in °C/W.

Current flow is equivalent to the heat flow, voltage is equivalent to the temperature, and electric resistance is equivalent to the thermal resistance. The analysis and modeling of
heat flow in the die casting system is based on this circuit analog. The heat flow modeling between die and casting, die and cooling line, and die and environment are discussed respectively.

3.4.1 Heat Flow from Casting to Die

To analyze and model the heat flow from casting to die, the casting can be divided into several small uniform sections. The heat flow from the casting section to a die voxel can be modeled using circuit analog, as illustrated in Figure 3.4.

![Diagram](a)

**Figure 3.4.** Circuit analog modeling of heat flow from casting to die.
Assuming the heat transfer is in steady state, heat transfer between the die and a casting section can be modeled by applying a constant heat flux to the casting surface, where the heat flow is calculated as:

\[
Q_c = \frac{T_c - T_d}{R_c} = \frac{T_c - T_d}{D_c/(kA_c)} = V \cdot P_r[(T_{in} - T_{ej})C_p + \lambda]
\]  

(3.3)

where:

- \(Q_c\): is the heat flow to the die, W.
- \(T_c\): temperature of the casting section, °C.
- \(T_d\): temperature at the die point, °C.
- \(R_c\): thermal resistance, °C/W.
- \(D_c\): distance from the die point to the casting section, m.
- \(k\): thermal conductivity of the material, W/m°C.
- \(V\): volume of the casting section, m³.
- \(A_c\): contacting area of the casting and die, m².
- \(P_r\): production rate, cycle/second.
- \(T_{in}\): injection temperature of casting, °C.
- \(T_{ej}\): ejection temperature of the casting, °C.
- \(C_p\): specific heat, J/m³°C.
- \(\lambda\): latent heat, J/m³.
3.4.2 Heat Flow from Die to Cooling Line

The cooling line extracts heat from the die, and the heat line inputs heat to the die section when early solidification may obstruct continues filling of the cavity. The mechanism of cooling or heating is to run the liquid cooling or heating media through the die material so as to extract heat from or input heat to the die. Heat flow between the die and the cooling and heating line are essentially the same. Thus, only the heat flow between the die and cooling line are discussed in this section to illustrate the concept.

Similar to the modeling of heat flow between die and casting sections, the heat flow links can be built from a die voxel to the cooling line. The heat flow depends on the temperature difference between the die voxel and the cooling line. The thermal resistance between them is illustrated in Figure 3.5.

![Figure 3.5. Circuit analog modeling of heat flow between die and cooling line.](image)
The equation for heat flow between the die and cooling line is as follows Eq. (3.4):

\[ Q_w = \frac{T_w - T_d}{R_w} \]  

(3.4)

where:

- \( Q_w \): is the heat flow rate from the die to cooling line, W.
- \( T_w \): temperature of the cooling media, °C.
- \( T_d \): temperature at the die voxel, °C.
- \( R_w \): thermal resistance between the die voxel and the cooling media, °C/W.

3.4.3 Heat Flow from Die to Environment

The heat flow between the die material and the environment can also be modeled similar to the modeling of heat flow between the die and casting sections, as shown in Figure 3.6.
The heat flow depends on the temperature difference between the die and the environment, as shown in Eq. (3.5):

\[ Q_{\infty} = \frac{T_{\infty} - T_d}{R_{\infty}} \]  

where:

- \( Q_{\infty} \): is the heat flow rate between die and environment, W.
- \( T_{\infty} \): temperature of the cooling media, °C.
- \( T_d \): temperature at the die voxel, °C.
- \( R_{\infty} \): thermal resistance between the die voxel and the cooling media, °C/W.
3.4.4 Total Heat Flow Network and Solving

Traditionally, the casting is divided into certain small thermal sections, as shown in Figure 3.7. The amount of heat contained in each of the thermal sections is proportional to its volume and is estimated as the basic heat input from this thermal section to the die. The designer then can compute the approximate temperature at every area in the die with a given distance to the thermal section (Herman, 1974).

Figure 3.7. Illustration of the heat sections used in the traditional thermal analysis method (Herman, 1974).

For a given point in the die, it is not only the nearest thermal section but all the surrounding thermal sections that contribute heat to it. Thus, the key factor of this method is
how to divide the thermal sections and compute the overall effects of all sections to a given point in the die. This method is efficient for castings with simple shapes but ill suited for castings with complex shapes.

However, based on this method and the circuit analog model, a similar yet computer-oriented approach is used in this research to model and solve the total heat flow network in the die, which will simultaneously take into consideration of the effects of the casting, cooling line, and environment to the thermal characteristics of the die.

Considering a point in the die, the heat flow front is a surface surrounding the point, and the heat flow is a 3D phenomena, as shown in Figure 3.8-a. The heat flow can be discretized and approximated with 1D heat flow paths, as shown in Figure 3.8-b.

![Figure 3.8. Heat flow front surface and its discretization: a) heat front surface; b) discretization.](image-url)
Theoretically, for die point, there exists an infinite number of heat flow paths through which the heat enters and exits the die point. In one dimension approximation, the more heat flow paths the discretization model has, the closer the approximation will be to the accurate result.

On the voxel model, each voxel is surrounded by its 26 neighbors, as shown in Figure 3.9-a. The first level of neighbors surrounding a voxel can be divided into three groups: face-connected neighbor, edge-connected neighbor, and vertex-connected neighbor. The face-connected neighbor shares a common face with the designated voxel. Similarly, the edge-connected neighbor shares a common edge with the designated voxel, and the vertex-connected neighbor shares a common vertex with the designated voxel. Total 26 neighbors surround a voxel. Thus, 26 directions can be defined from the center of the voxel to its surrounding neighbors, as shown in Figure 3.9-b.

Heat flow paths can be formed along such neighbor directions. Of course, the more the directions are, the better the approximate will be. However, the 26 heat flow paths from the first level of a voxel neighbor are sufficient in this qualitative analysis. These first level neighbor directions also have computational advantage in the voxel model, which is very important to support the interactive design evaluation in the conceptual design phase.
Figure 3.9. Illustration of the voxel neighborhood and the rays from it: a) the 26 neighbors of a voxel; b) rays from face-connected, edge-connected, and vertex connected directions.
Some of the heat flow paths will intersect with the casting sections, and others will intersect with the environment. Each intersection with casting forms a heat flow path with a thermal resistance due to the thermal conductivity of the casting and die material and the thermal resistance due to the heat transfer coefficient between the casting and die interface. Each intersection with the die surface forms a heat flow path with thermal resistance due to the die material and heat transfer coefficient between the die and environment.

Heat flow paths between a die voxel and the cooling/heating lines can also be formed using a similar concept.

Each of the heat flow paths surrounding a die voxel can be modeled using the circuit analog as discussed in previous sections (Section 3.4.1, Section 3.4.2, and Section 3.4.3). The casting sections and the heating lines serve as high voltage sources, and the exposed surfaces and the cooling lines serve as low voltage sinks. Thus, a total heat flow network can be modeled for each of the die voxels. A 2D example of the total heat flow network of a die voxel is illustrated in Figure 3.10.

In the network, we assume that the temperature of the casting section, cooling/heating line and environment are known. The temperature of the die voxel, what we want to solve for, is unknown.
As explained in the introductory part of this chapter, we assume that the die casting system is already in steady state, i.e. the total amount of heat flows out of that die voxel equals the total amount of heat flows into that die voxel. In other words, the heat flow network is balanced, as the following Eq. (3.6) shows:

\[
\sum Q = \sum_{i=1}^{\mu} \left( \frac{T_{c,i} - T_d}{R_{c,i}} \right) + \sum_{i=1}^{\nu} \left( \frac{T_{\infty} - T_d}{R_{\infty,i}} \right) + \sum_{k=1}^{\omega} \left( \frac{T_{w,k} - T_d}{R_{w,k}} \right) = 0
\]  (3.6)
Where:

$Q$: the heat flow to the die along a heat path, W.

$T_d$: temperature at the die point, °C.

$T_c$: temperature of the casting section, °C.

$T_\infty$: temperature of the environment, °C.

$T_w$: temperature of the cooling/heating media, °C.

$R_c$: thermal resistance between casting section and die voxel, °C/W.

$R_\infty$: thermal resistance between die surfaces and die voxel, °C/W.

$R_w$: thermal resistance between die voxel and cooling/heating line, °C/W.

$n$: the number of rays (from the die voxel) intersecting the casting surfaces.

$m$: the number of heat control features connected to the die voxel.

Similar to solving a balanced circuit network, Kirchoff’s Law is applied to this heat flow network to solve for the temperature at the die voxel. Solving the above equation for temperature $T_d$ at die voxel gives:

$$T_d = \frac{\sum_{i=1}^{n} \frac{T_{c,i}}{R_{c,i}} + \sum_{j=1}^{26-n} \frac{T_\infty}{R_\infty,j} + \sum_{k=1}^{s} \frac{T_w,k}{R_w,k}}{\sum_{i=1}^{n} \frac{1}{R_{c,i}} + \sum_{j=1}^{26-n} \frac{1}{R_\infty,j} + \sum_{k=1}^{s} \frac{1}{R_w,k}}$$

(3.7)

From the above equation, it can be seen that the die temperature depends on the temperature of the casting section, cooling/heating line, and environment, and depends on the
thermal resistances between the die and the heat sources and sinks including casting sections, cooling/heating line, and the environment.

The thermal resistance includes the resistance within casting and die material which depends on the distances and thermal conductivity of the material; and the resistance between the die and casting interface, which depends on the heat transfer coefficient between the boundaries. The thermal resistance computation will be discussed in detail in Chapter 4.

### 3.4.5 Assumption of Casting Temperature

To solve Eq. (3.7), the casting temperature, environment temperature and cooling/heating media temperature have to be known. The environment and cooling/heating media temperature can be well assumed based on field data or legacy data.

The casting temperature depends on part geometry and process parameters such as cycle time and injection temperature. For a given process, the part temperature distribution is mainly determined by the geometry of the part. One of the available function in Castview is to compute the distance transformation for part voxels, which in fact is the minimum distance from a part voxel to the die-casting interface. Lu (1995) and Lu et al. (1997) studied and found that this minimum distance pattern is a good approximation of the solidification pattern. In other words, the value of the distance estimates the solidification sequence of that voxel.

The solidification pattern reveals the temperature change of the casting. When the casting temperature drops below the solidus temperature, the liquid metal starts to solidify.
The highest temperature region will solidify last, which is also correspondent to the heavy mass section found by the thickness analysis module. Thus, the solidification pattern of the casting can be a first order approximation of the casting temperature distribution.

After thickness analysis of casting, the value of each casting voxel is the minimum distance from this voxel to the casting-die surface, i.e. the solidification sequence. This value is then mapped into the range of the casting temperature. The qualitative temperature of the die voxels is equivalent to the average temperature of the die in a cycle (see Section 3.5 below). The casting temperature is also assumed to be the average temperature of a casting in a cycle. The range of the casting temperature is from the average maximum temperature (which is normally 500-550 °C for aluminum) to the surface temperature of the casting (which is about 100-200 °C for aluminum). The high temperature occurs at the center of the thickest section which solidifies the last, while the low temperature is on the casting surface which solidifies first.

The other factor affecting the die temperature is the production rate or the cycle time. The total heat load rate of the die-casting system is proportional to the production rate, i.e. the faster production rate (shorter cycle time) is, the more heat is pumped into the system in a constant period of time, and the higher the die temperature will be. Since the production rate or the cycle time factor can not considered directly in this qualitative analysis, the effect of such factor is considered through the casting temperature. In the qualitative reasoning approach, the impact of increasing the production rate to the die temperature is equivalent to increasing the casting temperature. The casting temperature is considered to be proportional to the production rate and inverse proportional to the cycle time.
3.5 Justification of Die Temperature Distribution

Generally, the temperature distribution computed by this qualitative approach will not give the actual temperature at each time step during the die-casting cycle. In the calculation, it is assumed that the cavity is already filled and solidification is occurring, thus the calculated temperature distribution would correspond to the average temperature distribution during the solidification.

The calculation reveals the approximate characteristics of the temperature distribution pattern of the die. This temperature distribution pattern will identify some thermal-related features, such as heat converging/diverging areas and heat sinks. At the early design stage, it is these manufacturing features that help the designers to evaluate the design, not detailed temperature values at each step of the cycle.

3.5.1 Temperature Gradients in the Die

Because of the intermittent heat flow into the die, the temperature gradient through the thickness of the die may be divided into two zones, a) the fluctuation zone where the cyclic fluctuations are present; and b) the steady state zone where these variations are damped out and a steady gradient exists (Booth, et al., 1981). Figure 3.11 shows the temperature gradient present in a typical die.

The extension of the first zone depends on the amplitude and frequency of the cycle. The larger the amplitude (i.e. heavy section) is and the lower the frequency (i.e. slow casting rate) is, the deeper penetration of the cycle variations will be. On the other hand, thinner sections and faster casting rates reduce the depth of the penetration of the temperature fluc-
tuation. The gradient in the steady state zone is determined by the rate of heat input. Increasing the heat input (i.e. casting rate) will produce a steeper gradient. Of course the gradient is also effected by the thermal diffusibility of the die material.

![Graph showing temperature gradients through the depth of the die.](image)

**Figure 3.11.** Temperature gradients through the depth of the die (Booth, 1981).

### 3.5.2 Temperature Distribution in the Fluctuation Zone

The temperature profiles at the casting-die interface vary greatly during each cycle. Figure 3.12 shows how the temperature profile at the fluctuation zone varies with time when making a shot with an average thickness of 1.5mm magnesium alloy (Brevick, 1997).
Analyzing the graph, we notice that the injection stage is an extremely brief duration, normally in the band of 0.01 to 0.2 seconds: (Granchi, et al., 1983). At this stage, the temperatures at each point on the casting surfaces are almost the same as the temperature of molten metal. The temperature of the further regions from the gate may be slightly lower than the temperature near the gate area because of the heat loss through the surface where the metal passes by. But the difference is very subtle due to the extremely short injection time.
After the molten metal is injected into the cavity, its temperature decreases exponentially with time. The metal starts to solidify due to the tremendous temperature drop and heat loss from the die. The duration of solidification process depends on the thickness of the casting section (Granchi, et al., 1983). The temperature at each point on the casting surface therefore is dominated by the thickness of the sections. The temperatures around the thick section areas are higher than those around the thin section areas. The difference of the temperature between the thick section and the thin section could be very considerable if their section thickness differs significantly. When the metal is solidified, the surface temperature drops linearly due to minor flow of heat crossing the interface. The period from the beginning of solidification to the end of solidification (point A to B in Figure 3.12) is the time on which the casting surface temperature is dominated by the section thickness of the casting.

In this qualitative reasoning approach, the temperature of the casting section is approximated by the solidification pattern. The thicker the casting section is, the more heat content it possesses, therefore, the higher temperature the section surface has. This temperature corresponds to the temperature in the solidification period. Since this approach is a qualitative geometric-reasoning-based approach, the temperature calculated is not the actual temperature. It would be more reasonable to consider the calculated temperature as the average temperature during the solidification rather than the temperature at a specific point in this period.
3.5.3 Temperature Distribution in the Steady State Zone

From Figure 3.11, it can be seen that the temperature in the steady state zone in the die varies little with time. However, the temperature may vary significantly with the distance from the die and casting interface. At different locations in the steady state zone, the temperatures could be different. The temperature at a given location in the die is determined by the amount of heat transfer through that point, which is determined by the section thickness, the distance to the casting surface and the distance to cooling lines. Therefore, the calculated temperature distribution in the steady state zone is already the average temperature within a cycle and will not change with time, which reveals the characteristics of the die temperature distribution pattern.
CHAPTER 4

GEOMETRIC REASONING ALGORITHMS

4.1 Introduction

In the previous chapter we discussed the circuit analog modeling approach of heat flow between the die-casting components in the voxel model and the overall heat flow network of a die voxel and the method to solve for the steady state temperature at the die voxel. In the temperature equation given by Kirchoff’s Law, the major computational task is determining the thermal resistance. Neglecting the heat convection and radiation between casting and die surfaces, the thermal resistance is a function of material, distance and interface characteristics between the contacting materials. Among such factors, the essential computation burden is the distance measurement in the voxel model, for which time efficiency is the primary concern is of the interactive design evaluation tool. Each of the die voxels has 26 rays going out, and we need to know the intersection of each ray at either the casting section or the die surfaces and compute the distance within the die material and casting material respectively. We also need to know the distance from each of the die voxels to the cooling line or oil lines. The total value of the thermal resistance depends on the distances and the heat transfer coefficient at the boundary between materials.

In this chapter we will first discuss the composition of thermal resistances between the die and the thermal features including casting section, cooling line and environment.
We then discuss the algorithms for distance measurement in the voxel model, which is essential for the resistance computation. Distance measurement to the cooling lines are discussed briefly because it has already been implemented in CastView.

### 4.2 Thermal Resistances

The computation method for thermal resistances between die and the casting, cooling line, and environment are discussed respectively. For each type of the resistance, the assumptions corresponding thermal coefficients are explained.

#### 4.2.1 Thermal Resistance between Casting and Die

The majority heat flow in the die-casting system occurs between the casting alloy and the die. The heat flow from casting to die consists of several parts: transient heat flow during injection; conductive heat flow during solidification, and convection and radiation heat loss through the boundary. The convection and radiation are very subtle compared to the total heat loss from casting to die, thus these can be neglected in the qualitative analysis. The injection time is normally in the order of fractional seconds and the pressure is in the order of 10,000 psi. Experimental studies (Szaakacs, 1987; Herman, 1992) have found that about 20% of the heat moves from the casting to die during the transit period, and about 80% of the heat moves from the casting to die during the solidification period. This ratio varies according to the specific casting geometry and operation conditions such as pressure and cycle time. However, the transient phenomenon during injection is not the major concern in the qualitative study, and the injection can be assumed to be completed instantly.
The die-casting dies are normally coated to prevent wearing. During operation, the die cavity is also lubricated for better ejectability and surface finish. The coating material and the lubricant material also produces resistance to the heat flow from casting to die. However, if the die is coated properly and lubricated thoroughly, and the materials are distributed evenly throughout the die surfaces, then the thermal resistance would be evenly distributed throughout the die surfaces as well. The thermal resistance due to the coating and lubricant will affect the heat transfer during the start-up cycles, but will not affect the overall temperature distribution pattern of the die at the steady state. Thus the thermal resistance due to these two materials can be neglected. But their effect can be considered by choosing a proper heat transfer coefficient between the die and casting materials.

During the solidification, the casting alloy will shrink inside the cavity and there will be gaps between the casting and the die. These gaps will then cause thermal resistance to the heat flow from casting to the die. Modeling of these gaps, such as the distribution and size of the gap during or after solidification, is one of the major challenges in the detailed thermal-mechanical analysis of the die casting. The direct effect of the gaps on the heat flow, which can be approximated using the heat transfer coefficient, is that they will change the heat transfer coefficient between the casting and die.

Based on the above analysis, we assume that the cavity is filled instantly and thermal resistance due to conductive heat flow from casting to die during solidification are considered. Such thermal resistance essentially consists of the conductive resistance in the casting material, resistance at the casting and die interface, and the conductive resistance in the die material, as Figure 4.1 shows.
The total resistance \( R_t \) is a serial connection of the resistance in casting material \( R_c \), between casting and die interface \( R_h \), and in the die material \( R_d \), as in the following Eq. (4.1):

\[
R_t = R_c + R_h + R_d
\]  

The conductive resistance \( R_c \) is defined as Eq. (4.2).
The thermal resistance in the die material has a similar equation as that of the casting thermal resistance, as in Eq. (4.3).

\[
R_d = \frac{D_d}{k_d A} \tag{4.3}
\]

Where:

\( R_d \): thermal resistance in the die material, °C/W.
\( D_d \): distance in die material, m.
\( k_d \): thermal conductivity of the die material, W/m°C.
\( A \): contacting area between the casting and die, m².

The thermal resistance between the casting and die interface is a function of the heat transfer coefficient, as in Eq. (4.4).

\[
R_h = \frac{1}{hA} \tag{4.4}
\]
Where:

\[ R_h : \text{thermal resistance between casting and die interface, } ^\circ\text{C}/W. \]
\[ h : \text{heat transfer coefficient between casting and die, } W/m^2{^\circ}\text{C}. \]
\[ A : \text{contacting area between the casting and die, } m^2. \]

4.2.2 Thermal Resistance between Die and Cooling Line

In the voxel model, the cooling line or heating line is modeled as a line with unit thickness, i.e. one voxel size on the diameter. This is because the diameter or size of the cooling/heating channel is relatively small compared to the dimension of the dies.

The heat transfer between the die and cooling/heating channel involves heat conduction in the die material, radiation and convection through the wall of the cooling/heating channel. The governing factor in the heat transfer through the wall is the heat transfer coefficient between die material and the cooling/heating media. Thus the thermal resistance between the die and cooling/heating channel consists of conductive resistance in the die material and resistance due to the interface between the die and cooling/heating channels, as modeled in Figure 4.2.
Figure 4.2. Composition of thermal resistance between die material and cooling channel.

The total resistance ($R_w$) is a serial connection of the resistance in die material ($R_d$) and the resistance between die cooling channel interface ($R_h$), as shown in Eq. (4.5):

$$R_w = R_h + R_d$$  \hspace{1cm} \text{(4.5)}

The conductive resistance ($R_d$) is defined as Eq. (4.6):

$$R_d = \frac{D_d}{k_d A}$$  \hspace{1cm} \text{(4.6)}

Where:
R_d: thermal resistance in the die material, °C/W.

D_d: distance in die material, m.

k_d: thermal conductivity of the die material, W/m°C.

A: contacting area between die and cooling/heating channel, m².

The thermal resistance between the die and cooling channel interface is a function of the heat transfer coefficient:

\[ R_h = \frac{1}{hA} \]  \hspace{1cm} (4.7)

Where:

R_h: thermal resistance between casting and die interface, °C/W.

h: heat transfer coefficient between casting and die, W/m²°C.

A: contacting area between the casting and die, m².

### 4.2.3 Thermal Resistance between Die and Environment

Because the qualitative reasoning is based on the assumption that the die is closed and liquid metal is injected into the cavity with zero time, the die surfaces contacting the environment only refer to the surfaces between the inserts and the holder block, not including the shut-off surface, which is also exposed to the air when the die is opened.
The die model in the voxel space at least will include the volume of inserts. It also may include some volume of the holder block depending on the die sketched on the STL space (see Section 4.3). The die surface hitherto refers to the outside surface of the die volume sketched on the STL space. Thus, heat transfer between a die voxel to the die surface includes the heat conductivity within the die material, heat transfer across the die-holder interface, conduction with the holder block, and heat transfer between the outer surfaces of the holder block. The correspondent thermal resistance is illustrated in Figure 4.3.

![Figure 4.3. Composition of thermal resistance between die and environment.](image)

The total thermal resistance is:
\[ R = R_d + R_h + R_b + R_{\infty} \]  \hspace{1cm} (4.8)

\( R_d \) is the conductive resistance within the die material, \( R_h \) is the resistance across the surface between die and holder block, \( R_b \) is the resistance within the holder block, and \( R_{\infty} \) is the resistance between the interface of the holder block and the air. These are defined by the following set of equations:

\[ R_d = \frac{D_d}{k_d A} \]  \hspace{1cm} (4.9)

Where:

\( R_d \) : thermal resistance in the die material, \( ^\circ C/W \).

\( D_d \) : distance in die material, \( m \).

\( k_d \) : thermal conductivity of the die material, \( W/m^\circ C \).

\( A \) : contacting area, \( m^2 \).

\[ R_h = \frac{1}{hA} \]  \hspace{1cm} (4.10)

Where:

\( R_h \) : thermal resistance across die-holder interface, \( ^\circ C/W \).

\( h \) : heat transfer coefficient on the interface, \( W/m^2^\circ C \).

\( A \) : contacting area, \( m^2 \).
\[ R_b = \frac{D_b}{k_dA} \]  
\hspace{1cm} (4.11)

Where:

- \( R_b \): thermal resistance within the holder block, \( ^\circ C/W \).
- \( D_b \): extended distance of the holder block, \( m \).
- \( k_d \): thermal conductivity of the holder material, \( W/m^2\circ C \).
- \( A \): contacting area, \( m^2 \).

\[ R_\infty = \frac{1}{h_\infty A} \]  
\hspace{1cm} (4.12)

Where:

- \( R_\infty \): thermal resistance across holder-air interface, \( ^\circ C/W \).
- \( h_\infty \): heat transfer coefficient on the interface, \( W/m^2\circ C \).
- \( A \): contacting area, \( m^2 \).

The actual die model in the voxel space does not include the whole volume of the holder block. However, the thickness of the holder block is estimated and an extended distance within the holder is used to compute the thermal resistance.

Based on the assumption, i.e. using a 1-D heat transfer model to discretely approximate 3-D heat flow in the voxel space, the area involved in the thermal resistance can be assumed to be constant along every heat flow path and thus can be eliminated from the computation.
Having the distances in casting and die, thermal conductivity, and heat transfer coefficient between casting and die, and the heat transfer coefficient between the die and cooling channel, the thermal resistances discussed above can be evaluated on the voxel model.

4.2.4 Thermal Conductivities and Heat Transfer Coefficients

The thermal conductivity is an intrinsic material property and the value of it depends on the particular material. The thermal conductivity also depends on the temperature of material. The following table show the thermal conductivity and correspondent temperature of typical casting materials (Zinc, Table 4.1; Magnesium, Table 4.2; and Aluminum, Table 4.3) and die material (H-13, Table 4.4; AiSi 4140, Table 4.5).

<table>
<thead>
<tr>
<th>Alloy</th>
<th>Thermal conductivity (W/mK)</th>
<th>Liquidus temperature (\degree C)</th>
<th>Solidus temperature (\degree C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zamac 2</td>
<td>105</td>
<td>390</td>
<td>379</td>
</tr>
<tr>
<td>Zamac 3</td>
<td>113</td>
<td>387</td>
<td>381</td>
</tr>
<tr>
<td>Zamac 5</td>
<td>109</td>
<td>386</td>
<td>380</td>
</tr>
<tr>
<td>Zamac 7</td>
<td>113</td>
<td>387</td>
<td>380</td>
</tr>
<tr>
<td>ZA - 8</td>
<td>115</td>
<td>404</td>
<td>375</td>
</tr>
<tr>
<td>ZA - 12</td>
<td>116</td>
<td>432</td>
<td>450</td>
</tr>
<tr>
<td>ZA - 27</td>
<td>125.5</td>
<td>484</td>
<td>375</td>
</tr>
</tbody>
</table>

**Table 4.1.** Thermal conductivity, liquidus and solidus temperature of Zinc alloy.
<table>
<thead>
<tr>
<th>Alloy</th>
<th>Thermal conductivity $\text{W/mK}$</th>
<th>Liquidus temperature $^\circ\text{C}$</th>
<th>Solidus temperature $^\circ\text{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM 60</td>
<td>62</td>
<td>615</td>
<td>540</td>
</tr>
<tr>
<td>AM 91</td>
<td>72</td>
<td>595</td>
<td>470</td>
</tr>
<tr>
<td>AS 41A, 41XB</td>
<td>68</td>
<td>620</td>
<td>565</td>
</tr>
</tbody>
</table>

Table 4.2. Thermal conductivity, liquidus and solidus temperature of Magnesium alloy.

<table>
<thead>
<tr>
<th>Alloy</th>
<th>Thermal conductivity $\text{W/mK}$</th>
<th>Liquidus temperature $^\circ\text{C}$</th>
<th>Solidus temperature $^\circ\text{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 356</td>
<td>159</td>
<td>615</td>
<td>555</td>
</tr>
<tr>
<td>A 357</td>
<td>152</td>
<td>615</td>
<td>555</td>
</tr>
<tr>
<td>A 360</td>
<td>113</td>
<td>595</td>
<td>555</td>
</tr>
<tr>
<td>A 380</td>
<td>96.2</td>
<td>595</td>
<td>540</td>
</tr>
<tr>
<td>A 383</td>
<td>96.2</td>
<td>580</td>
<td>515</td>
</tr>
<tr>
<td>A 384</td>
<td>92</td>
<td>580</td>
<td>515</td>
</tr>
<tr>
<td>A 390</td>
<td>134</td>
<td>650</td>
<td>505</td>
</tr>
</tbody>
</table>

Table 4.3. Thermal conductivity, liquidus and solidus temperature of Aluminum alloy.

<table>
<thead>
<tr>
<th>Temperature, ($^\circ\text{C}$)</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>600</th>
<th>700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annealed, ($\text{W/mK}$)</td>
<td>27.21</td>
<td>28.09</td>
<td>29.22</td>
<td>30.14</td>
<td>31.74</td>
<td>34.08</td>
<td>36.51</td>
</tr>
<tr>
<td>Heat-treated, ($\text{W/mK}$)</td>
<td>25.47</td>
<td>27.10</td>
<td>28.86</td>
<td>29.89</td>
<td>30.99</td>
<td>32.45</td>
<td>34.25</td>
</tr>
</tbody>
</table>

Table 4.4. Thermal conductivity of H-13.
As can be seen from the above tables, the thermal conductivity not only changes with material types, it also varies with the temperature of the material. For the die material, the paradox is that the conductivity is required to compute the die temperature on which the thermal conductivity also depends. The theoretical solution would be to solve the conduction equation by iterations. However, the qualitative computation tries to avoid any iterations and means to solve for an approximate result directly.

Since the qualitative temperature is equivalent to the average temperature of a cycle at the steady state. The thermal conductivity used in the computation does not necessarily have to be very accurate. For the casting material, the conductivity can be directly chosen from the above tables according to the material types. Only an approximate integer value is sufficient. For the die material, the average value within the temperature range is chosen in the computation, e.g. 30 W/mK is chosen for H-13 die steel.

The heat transfer coefficient between the contacting materials can be constant or time or temperature dependent. Figure 4.4 shows the heat transfer coefficient curve of the A380 and H-13 interface. The average value is chosen in the computation for the nature of the qualitative reasoning as explained above.
Figure 4.4. Heat transfer coefficient curve between A380 and H-13 (extracted from the material database of MagmaSoft).

4.3 Die and Cooling Line Sketch

The die and cooling line geometry are sketched on the STL model and the data are stored in a text file. Their geometry will then be transferred to the voxel space and imported to the procedure for thermal evaluation. This section briefly describes the geometric sketching methods only. The design and evaluation of the cooling line will be discussed in Section 4.5 in detail.
4.3.1 Die Sketch

The temperature distribution is computed on the die voxels. Because of the memory required for the voxel model and the time required for the temperature computation, the large voxel model is not time and space efficient. So, the holder blocks are not included in the die voxel model, only the inserts are simulated. The other practical reason is that the temperature on the holder block is normally uniform and less important for placing cooling lines. Hereby, the term “die” refers only to the insert and does not include the holder block.

The die geometry is sketched on the STL model of the part. For most of the part, the die orientation is aligned with the part bounding box, and the die geometry is just a parallel extension of the bounding box. However, some parts may need to be oriented differently in the die and the die is no longer aligned with the bounding box of the part. Figure 4.5 shows a die geometry extended from the part bounding box.

The die geometry and dimension is later used by the voxelization procedure to specify the die voxels in the voxel model. The geometry serves as the base for cooling line placement on the STL model.
Figure 4.5. Die geometry extended from the part bounding box.
4.3.2 Cooling Line Sketch

The candidate locations of the cooling lines are drawn from the thermal feature of the die and part, i.e. hot spots and heavy mass sections. These thermal features are presented on the voxel model and perceived visually by the designer. Because of the nature of the qualitative reasoning, the sketching is not based on the exact location of the thermal features on the voxel model, but on the visual perception of these features on the STL model.

Each of the cooling lines consists a group of consecutive line segments. Since the die volume defined on the STL model may only include the die insert and may not include the holder block, and the actual cooling line may run through the holder block volume also, the cooling lines sketched on the STL model are not constrained to be within the die block. A cooling line starts from one face on the die block and exits to another face, but its interior may run out of the die block. However, when transferring the cooling line layout from STL space to the voxel space, the portions outside of the die volume could not be mapped onto voxel mode and thus will not be considered in the thermal evaluation. An example of a cooling line sketch is presented in Figure 4.6.

Once cooling lines are sketched on the STL model, they can be transferred to the voxel mode directly. The cooling line locations can also be saved in a text file for off-line transformation. The file format is as shown in Figure 4.7. The comments starts with "#". The header contains the file name, design date and other notation. The first entry is the total number of cooling lines in the file. Each of following section is for a separate cooling line including the number of the line segment and the coordinates of the each line segments in floating-point format.
Figure 4.6. Example of cooling line sketch on the STL model.
#Headers: (STL file name, date of design, and designer etc.)

#(number of cooling lines, integer)

n

#(number of line segments in the first cooling line)

m

# the coordinates of each line segments

(x₁, y₁, z₁), (x₁', y₁', z₁')
(x₂, y₂, z₂), (x₂', y₂', z₂')
...
(xₘ, yₘ, zₘ), (xₘ', yₘ', zₘ')

#end of cooling line segments

...

Figure 4.7. Cooling line file format.
4.4 Distance Computation

The major computation burden in this research is to efficiently compute the distance of each die voxel to the casting sections and to the exposed surfaces along each of the neighbor directions from the voxel. Given the objective of this approach, i.e. to provide a quick and easy-to-use tool to support interactive design evaluation at the conceptual design stage, the time efficiency is the critical requirement for the algorithm design.

4.4.1 Introduction and Intuitive Algorithm

An intuitive algorithm starts from the center of the die voxel and accumulates the distance step by step along each direction, as illustrated in Figure 4.8. For each voxel in the die region, rays are formed according to its first layer of neighbors, i.e. 26 rays from 26 neighbors. Each ray starts from the designated die voxel and proceeds voxel by voxel until it intersects with either the center of the casting section or an exposed surface. Meanwhile, distances within the die and the casting are computed. Thermal resistances are later computed according to the distances and material as well as interface properties, as discussed in the previous section.
void Compute_Distance(voxeI_t * voxel_buffer, int buffer_size)
{
    for(int i=0; i<buffer_size; i++)
    {
        // initialize the current voxel
        voxel_t curr = voxel_buffer[i];

        if( curr.position == IN_DIE)
        {
            // shoting 26 rays from each voxel
            for( int ray=0; ray<26; ray++)
            {
                // initialize distance to zero
                int dist = 0;
                // find the next voxel along the ray
                voxel_t next = curr.next(ray);
                if ( next.position == IN_DIE)
                    // if the voxel is in the die, accumulate the distance
                    dist++;

                if ( next.position == IN_CAST || next.position == ON_SURF)
                {
                    // if the voxel is in the casting section
                    // or the exposed surfaces, compute the thermal resistance
                    // along this ray
                    float Rt= Thermal_Resistance(curr, ray, dist);
                }
            }
        }
    }
}

Figure 4.8. Illustration of the intuitive algorithm for distance computation.
The inefficiency of this intuitive algorithm is that it does not utilize the coherence between the voxels on the same ray. For example, in the following Figure 4.9, the voxel on the end of the ray will be traversed \( N \) times, where \( N \) is the total number of voxels along the ray.

Suppose voxelization resolution is \( N \), i.e. the number of slices along the maximum dimension of the die voxel model, for each ray, a voxel has to be traversed by \( N \) passes. Totally, each voxel has to be traversed by \( 26*N \) passes. The overall time complexity of this algorithm is \( O(26 \cdot N \cdot N^3) = O(N^4) \). The actual running time of this algorithm is in the order of hours for a part with a moderate number of slices (e.g. 200) on a PII-500Mhz-128Mb_NT4.0 workstation. This is not acceptable for an interactive conceptual design tool.

![Illustration of the traversing procedure of a ray](image)

**Figure 4.9.** Illustration of the traversing procedure of a ray.
4.4.2 Parallel Ray Casting Algorithm

The intuitive algorithm described in the previous Section 4.4.1 shoots out rays from the center of each voxel such that distances are computed from the center to the designated features along the ray. The coherence information between the voxels along the ray is not fully exploited in this algorithm and the computational time is high. A solution faster than the intuitive algorithm is needed to bring the running time range down from the order of hours to the order of minutes in order to support an interactive try-out of design alternatives. A ray casting algorithm which fully exploits the coherence between voxels along a ray was developed whose time complexity is $O(N^3)$, where $N$ is the voxelization resolution.

A detailed description of the algorithm follows the introduction of the data structure supporting the algorithm.

4.4.2.1 Data Structure

The first data structure is the coding of the faces of the voxel model. The voxel model has a rectangular volume with six faces. The coding of the faces and the reference coordinate system is illustrated in Figure 4.10. Each of the faces is indexed by an integer value.

The coding of the ray from each of the neighbor directions of a voxel is shown in Figure 4.10. The structure of the ray is defined as shown in Figure 4.12-a. The direction of the ray is defined using the relative increment of the coordinates of the center voxel along $X$, $Y$, and $Z$ axes, as shown in Figure 4.12-b. The value of the increment is either "-1", "0", or "1" depending on the relative position of the neighbor voxel to the center voxel. This
way, when the index or coordinate of a voxel, it is easy to get the index or coordinate for its neighbor voxel along a particular ray. The attribute “dist” is the factor of the cutting distance of the ray in the voxel. The actual distance a ray travels in a voxel is the actual size of the voxel multiplied by this distance factor. The value of the factor is “1.0”, “1.414” ($\sqrt{2}$), or “1.732” ($\sqrt{3}$) depending on the direction of the ray.

Each ray can be projected to certain faces of the volume, i.e. the faces which can be “seen” along the direction of the ray. For example, ray-0 can see face F1; ray-1 can see face F1, F2, and F3. The face which can be seen from the ray is called the “projection face” and is denoted as “face” in the ray data structure where the value in the array is the index of the faces. If a ray has less than three projection faces, then “-1” is assigned to the index to indicate a void face index.

Table 4.6 shows data of the 26 rays from a voxel.
Figure 4.10. Voxel Model and code for the faces.

Figure 4.11. Code of voxel neighbor directions.
Class Ray
{
    int id;
    int dx,dy,dz; // incremental at X,Y,Z axes
    float dist; // factor cutting distance in voxel
    int face[3]; // indices of the projected faces
};

Figure 4.12. a) Data structure of a ray; b) Example of a ray and its attributes.

<table>
<thead>
<tr>
<th>Ray ID</th>
<th>dx</th>
<th>dy</th>
<th>dz</th>
<th>dist</th>
<th>face</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td>0, -1, -1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\sqrt{3}$</td>
<td>0, 1, 2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$\sqrt{3}$</td>
<td>0, 2, -1</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>$\sqrt{3}$</td>
<td>0, 2, 3</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>$\sqrt{2}$</td>
<td>0, 3, -1</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>$\sqrt{3}$</td>
<td>0, 3, 4</td>
</tr>
</tbody>
</table>

Table 4.6. Data for the 26 rays from a voxel (to be continued).
Table 4.6. Data for the 26 rays from a voxel (continued).

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>$\sqrt{2}$</td>
<td>0, 4, -1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>$\sqrt{3}$</td>
<td>0, 1, 4</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$\sqrt{2}$</td>
<td>0, 1, -1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td>1, 2, -1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1.0</td>
<td>2, -1, -1</td>
</tr>
<tr>
<td>11</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td>2, 3, -1</td>
</tr>
<tr>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>3, -1, -1</td>
</tr>
<tr>
<td>13</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td>3, 4, -1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1.0</td>
<td>4, -1, -1</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td>4, 1, -1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>1, -1, -1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>$\sqrt{3}$</td>
<td>1, 2, 5</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>$\sqrt{2}$</td>
<td>2, 5, -1</td>
</tr>
<tr>
<td>19</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>$\sqrt{3}$</td>
<td>2, 3, 5</td>
</tr>
<tr>
<td>20</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>$\sqrt{2}$</td>
<td>3, 5, -1</td>
</tr>
<tr>
<td>21</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>$\sqrt{3}$</td>
<td>3, 4, 5</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>$\sqrt{2}$</td>
<td>4, 5, -1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>$\sqrt{3}$</td>
<td>1, 4, 5</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>$\sqrt{2}$</td>
<td>1, 5, -1</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1.0</td>
<td>5, -1, -1</td>
</tr>
</tbody>
</table>

4.4.2.2 Ray Casting Algorithm

The basic idea of the ray casting algorithm is to exploit the coherence information between each neighbor voxel along a ray and traverse each voxel only one time for each ray. Each voxel in the model has a homogeneous structure, the ray from the same voxel neighbor has the same spatial direction. The coherence between each two consecutive voxels along a ray gives that the distance of one voxel to a featured voxel (casting or bound-
ary etc.) is the distance of its neighbor voxel plus or minus one voxel size along the ray. Therefore, parallel rays can be projected on the projection surfaces simultaneously and distances can be computed consecutively along each instance of the ray.

A graphical illustration of the algorithm is shown in Figure 4.13. The indices of the projection faces are pre-defined for each of the rays. The voxel on the projection face is called the “entry voxel”. Figure 4.14 provides an example of a ray instance and the traversing process.

The ray starts from the entry voxel and moves forward step by step. Before the ray moves into the die region, the thermal resistance due to the heat transfer coefficient on the boundary surface is evaluated (see Section 4.2.2). At the first step, the distance of the entry voxel to the boundary surface can be computed, which is in fact one voxel size multiplied by the distance factor of the ray. When the ray is moving forward to the next voxel, the distance of this voxel to the boundary surface is the previous distance plus the unit voxel size times the distance factor, as shown in the following equation. This process is repeated for each voxel along the ray in the die until it reaches the casting section. Thermal resistance in the die region is evaluated according to the distance and the thermal conductivity of the die material.

\[ D_{i+1} = D_i + (\text{distance factor}) \times (\text{voxel size}) \]

When the ray enters the casting section, the pre-computed temperature of the casting voxels are accumulated for all the casting voxels traversed by the ray and the average
is assigned as the temperature of this casting section. The distance the ray travels in the casting section is used to compute the thermal resistance in the casting section.

When the ray leaves the casting section, it enters into the die region again. The thermal resistance across the casting-die interfaces is evaluated before the ray marches into the die region. When the ray travels into the die region, similar as accumulating the distance from a die voxel to the boundary surfaces, the distance from a die voxel to the casting surface can be accumulated progressively until the ray enters into another casting section. Thermal resistance in the die region can be evaluated according to the distance and the thermal conductivity of the die material.

This process repeats until the ray reaches the other side of the boundary surface. The skeleton of the algorithm for one ray from the entry voxel on the projection face is illustrated in Figure 4.15.

![Figure 4.13. Graphical illustration of the ray casting algorithm.](image)
Figure 4.14. Example of an instance of a ray.
void Ray_Casting(voxel_t entry, ray_t ray)
{
  voxel_t curr = entry; // the current voxel
  voxel_t prev=curr.previous(ray); // the previous voxel along the ray
  float Dd = 0.0, Dc = 0.0; // distance in the die and voxel
  float casting_temp = 0.0; // casting temperature
  do{
    if(prev.posistion == ON_SURF){
      // initialize the distance in the die
      Dd = ray.dist*voxel_size;
      // export the thermal resistance across the boundary
      float Rh = Thermal_Resistance(HT_AIR);
      break;
    }
    if(prev.position == IN_DIE){
      switch(curr.position){
      // the ray is marching in the die region
        case IN_DIE:
          // accumulate the distance in the die
          Dd += ray.dist*voxel_size;
          // export the thermal resistance in the die
          float Rd = Thermal_Resistance(curr, Dd, Kd);
          break;
        // ray is entering the casting section
        case IN_CAST:
          // initialize the distance in the casting region
          Dc = ray.dist*voxel_size;
          // initialize the casting temperature
          casting_temp = curr.temperature;
          break;
        // the ray is leaving the volume
        case ON_SURF:
          break;
      }
    }
  }
}

Figure 4.15. Illustration of the ray casting algorithm for thermal distance and thermal resistance computation (to be continued).
// accumulate the distance in the die
Dd += ray.dist*voxel_size;

// export the thermal resistance in the die
float Rd = Thermal_Resistance(current, Dd, Kd);
break;
}
if(prev.position == IN_CAST){
    switch(curr.position){
    // the ray is marching in the casting section
    case IN_CAST:
        // update the distance in the casting section
        Dc += ray.dist*voxel_size;
        // update the casting temperature
        casting_temp = curr.temperature;
        break;
    // the ray is leaving the casting section
    case IN_DIE:
        // the average temperature of the casting section
        casting_temp = average(casting_temp, thickness);
        // the thermal resistance across the casting boundary
        // and the thermal resistance in the casting region
        float Rc = Thermal_Resistance(HT_CAST, Kc);
        // initialize the distance in the die
        Dd = ray.dist*voxel_size;
        // export the thermal resistance across the boundary
        float Rh = Thermal_Resistance(HT_AIR);
        break;
    }
    // marching the voxel a step ahead
    prev = next;
    next = next.next_voxel(prev);
}
while(next.position != OUT_OF_VOLUME);
The input for the algorithm is the entry voxel on the project face and the ray. It traverses the voxels along the ray just once. A wrapper procedure repeats the above algorithm for all the entry voxels on the projection faces along the input ray, and a higher level wrapper procedure repeats the wrapper for all the 26 rays. Suppose the voxelization resolution is $N$, the size the voxel buffer would be $N^3$. The number of the entry voxels on the project face is $N^2$, and the traversing depth along a ray is in the order of $N$. The time complexity for computing all thermal resistance for every voxel along all the 26 rays is $\Omega(26 \times N^2 \times N) = O(N^3)$. The efficiency is improved by one order from the intuitive algorithm, which is in the order of $O(N^4)$.

4.5 Cooling Line Design and Evaluation

The cooling line design discussed in this section refers to the selection of cooling line location. The sketching of cooling line layout is described in Section 4.3.2. Other aspects such as the detailed geometry, and flow control, will not be included in this section.

The cooling line evaluation refers to the impact of the cooling line on the die temperature distribution, i.e. how the cooling line will affect the overall distribution of the die temperature. Similar to the other qualitative evaluations, the result of the cooling line effect will also be presented visually.

4.5.1 Cooling Line Location

From a conceptual design perspective, there are two criterion which guide the selection of cooling line location: hot spots in the die and heavy mass section of the casting (Her-
man, 1992). Cooling lines could be placed near the hot spots and heavy sections. The heavy mass sections can be identified by the thickness analysis module available in CastView.

The hot spot is also addressed as the heat converging area in the die. The heat flow pattern in the die is affected by the geometric shape of the cavity. For a flat cavity, the heat flows through the die in a reasonably parallel pattern, as illustrated in Figure 4.16-a. In some cases, the heat flows away from the cavity in an ever-widening pattern. Such a pattern is called diverging heat flow, while the area associated with it is called diverging heat flow area. In the other cases, the heat flows away from the cavity in an ever-narrowing pattern, known as converging heat flow (Herman, 1992). Figure 4.16-b illustrates the converging and diverging heat flow patterns.

![Diagram](image)

**Figure 4.16.** Thermal features in the die: a) parallel heat flow; b) diverging and converging heat flow.
The heat converging area is normally associated with high temperature. The converging heat flow can produce severe heat congestion, which delays the solidification of casting and causes heat checks on the die. Therefore, cooling lines must be placed in such areas to compensate the heat congestion (Herman, 1974). By calculating the temperature distribution pattern of the die, certain thermal-related features can be identified. Having the hot spots in the die and heavy mass section of the casting, the cooling line location can be determined conceptually. Figure 4.17 illustrates the schematic view of the thermal patterns and the placement of cooling lines.

Figure 4.17. Thermal features and cooling line locations in the die (Herman, 1992).
4.5.2 Cooling Line Evaluation

As discussed in Chapter 3, the effect of the cooling line on the die temperature distribution is evaluated through the thermal resistance between the cooling line and the die voxels. The thermal resistance depends primarily on the distance of each die voxel to the cooling line. Hence, the measurement of the distance from die voxels to the cooling line is the major hurdle of geometric reasoning.

Fortunately, the distance measurement problem has been solved by Lu and Rebello in CastView (Lu, 1996; Rebello, 1997). The algorithm used in CastView for distance measurement is called “Distance Transformation” (DT), which approximates the Euclidean distance in a discrete space. The DT algorithm was originally developed by the image processing community to measure the minimum distance from each pixel to the designated features on a 2D pixel model (Rosenfel and Pfaltz, 1968; Borgefors, 1986; Vossepoel, 1988; Borgefors, 1991), and was expanded to a 3D voxel model (Verwer and Verbeek, 1989; Rosenfel and Pfaltz, 1996) which has applications in manufacturing feature extraction (Chu and Lee, 1993; Lu, et al., 1997). An accurate measuring algorithm of Euclidean distance in discrete space has been developed (Huang and Mitchell, 1994; Chang and Yan, 1996; Cuisenaire and Macq, 1999).

Measurement of distance of the die voxels to cooling line is based on the DT algorithm already available in CastView. A detailed description of the algorithm can be found in Lu (1996) and Rebello (1997). The cooling line layout is defined in the STL model and transferred to the voxel space. Because the thickness of the cooling line is neglected in qualitative reasoning, the corresponding voxels on the cooling line is just one layer thick. The
designated features are the cooling line voxels to which the distance from each of the die voxel is measured.

Figure 4.18. Distance Transformation from cooling line to die voxels: a) cooling line in the voxel model; b) sliced view of distance transformation on the ejector half of the die; c) distance transformation on the cover half of the die.
CHAPTER 5

VERIFICATION OF THE QUALITATIVE RESULTS

5.1 Introduction

As a conceptual design tool which aims to support interactive thermal design evaluation, this qualitative reasoning approach is not expected to give high accuracy with a long run time. To the contrary, it is expected to be able to provide non-misleading and ball-park result which has sufficient information for conceptual design evaluation within a short period of time. For this end, the subject of the verification includes two aspects: correctness of the die temperature pattern and running time. This involves measuring the running time and comparing the qualitative results with the numerical simulation results by finite element package and finite difference package.

Several cases are studied for verification. For each case, we run the numerical simulation with FEM package (ABAQUS) or FDM package (MAGMASoft). The verification is conducted in two ways: qualitatively and quantitatively.

In the qualitative verification, we compare the thermal features identified from both the numerical simulation and qualitative reasoning result on the die, such as hot spots, low temperature region. The thermal features are important to the designer and they represent the overall temperature distribution of the die. By comparing the thermal features, we will
know whether the qualitative result is misleading or promising. The comparison is based on the visualization of the both numerical results and the reasoning result.

In the quantitative verification, we compute the percentage of difference between the numerical simulation results and the reasoning results. The percentage of difference will give us ball park perception of accuracy of the qualitative reasoning. The quantitative comparison is conducted for the cases whose numerical simulation solver is ABAQUS. This is because that the ABAQUS database is open and we can extract the temperature value from it. However, the MAGMASoft database is not open therefore the quantitative comparison is not conducted for these cases.

The run time of these cases are also collected. But it is not compared with the running time of correspondent numerical simulation. This is because that this qualitative reasoning tool does not aim to compete with the numerical simulation in terms of running time. The running time of qualitative reasoning is evaluated in terms of whether or not it could support interactive design evaluation. The measurement is taken on regular computers which are commonly used by die-casting designers.

A design study case is presented in the last to illustrate the design evaluation process of cooling line effects. We also studied the sensitivity of temperature distribution vs. the voxelization resolution to show that the reasoning result is stable and does not change when the voxelization resolution reaches certain level.
5.2 Case Studies

Four cases are presented in this section. The first case is compared with finite element result from ABAQUS. The second case uses the same part as the first case but has cooling lines to verify the cooling line effect to the die temperature distribution. The comparison is also to an ABAQUS result. The last two cases are compared with the result from finite difference analysis using MAGMASoft.

5.2.1 Case 1

The configuration of the part used in this case is shown in Figure 5.1. The part itself has four walls on a thin plate. Its configuration in the casting includes biscuit, runner and overflows. These components and the part are considered to be a single piece of the casting the numerical simulation and qualitative reasoning.

The qualitative reasoning result are visualized using the volume render module available in CastView. The temperature pattern on the cavity side of the ejector die and cover die are shown in Figure 5.2 and Figure 5.3 respectively. This qualitative temperature is equivalent to the average temperature of a steady state cycle.

The numerical simulation is done with ABAQUS. The simulation starts from the room temperature and runs through about 100 cycles to reach the steady state. To compare the qualitative result, the average temperature of each node at the last cycle are calculated and displayed in ABAQUS CAE, as shown in Figure 5.4 and Figure 5.5.

It can seen from the numerical simulation result, the hot region on the ejector cavity is between the ribs, especially between the middle two ribs, and also around the biscuit area.
This is because the ribs forms heat converging area and traps heat in between. Roughly the similar thermal features can be identified from the qualitative reasoning result. We can also see the hot region between ribs and around biscuit areas. These features plus the thickness information of casting will help the designers to decide where to locate the cooling lines.

Figure 5.1. Geometry of the part for case-1.
Figure 5.2. Qualitative reasoning result: temperature pattern on ejector die.
Figure 5.3. Qualitative reasoning result: temperature pattern on the cover die.
Figure 5.4. Numerical simulation: temperature pattern on the ejector die.
Figure 5.5. Numerical simulation: temperature pattern on the cover die.
5.2.2 Case 2

This case intends to verify the effects of the cooling line on the die temperature distribution. The same casting as case-1 is used. From case-1, we know that the hot region is in between the ribs. Considering the dimension between the ribs, a cooling line is placed between the middle two ribs, as shown in Figure 5.6.

The qualitative reasoning temperature pattern on the ejector and cover are shown in Figure 5.7 and Figure 5.8 respectively. In order to compare with the numerical simulation result in the same scale, the qualitative results are transformed from the voxel model to the mesh model displayed with the ABAQUS CAE as shown in Figure 5.9 and Figure 5.10. The numerical simulation is setup with the same configuration as case-1 except that a cooling line is included in the model. The results are shown in Figure 5.11 and Figure 5.12. These temperature results are the average of the last cycle at the steady state.

From both the numerical simulation results and qualitative reasoning results, it can be seen that the heat trap region without cooling line is cooled after placing cooling though the middle two ribs. Now, the other heat traps are between the top two and bottom tow ribs. However, because the overall temperature at this hot region are very low compared to the temperature at the heat trap without the cooling line, the cooling line is proven to be effective at this location.
Figure 5.6. Part and cooling line configuration in case-2.
Figure 5.7. Qualitative reasoning result: temperature pattern on the ejector cavity with cooling line.
Figure 5.8. Qualitative reasoning result: temperature pattern on the cover cavity with cooling line.
Figure 5.9. Qualitative reasoning result displayed in ABAQUS CAE: temperature pattern on the ejector cavity with cooling line.
Figure 5.10. Qualitative reasoning result displayed in ABAQUS CAE: temperature pattern on the cover cavity with cooling line.
Figure 5.11. Numerical simulation result: temperature pattern on the ejector cavity with cooling line.
Figure 5.12. Numerical simulation result: temperature pattern on the cover die with cooling line.
5.2.3 Case 3

A thin wall box type of part is studied in this case. The part geometry is shown in Figure 5.13. The qualitative temperature results of the dies are shown in Figure 5.14.

The numerical simulation used in this case is a finite difference package, MAGMASoft. The temperature pattern from the numerical simulation are displayed in Figure 5.15 and Figure 5.16. The temperature is taken at the time step when about 90% of the casting is solidified. This temperature is not the average temperature within the cycle at the steady state because the database of MAGMASoft is not readable by third party program and the average temperature can not be computed for this case. However, by checking the temperature pattern through out the solidification process, we found the location major thermal pattern do not change too much except the temperature value, e.g. the hot spot in this cases around the center of the bottom wall and around the biscuit area. The similar pattern can be found on the qualitative reasoning result.
Figure 5.13. Geometry of the box part studied in case-3.
Figure 5.14. Qualitative reasoning result: temperature pattern on the ejector (right) and cover (left) cavity surfaces of case-3.
Figure 5.15. Numerical simulation result (MagmaSoft): temperature pattern on the ejector cavity surface of case-3.
Figure 5.16. Numerical simulation result (MagmaSoft): temperature pattern on the cover cavity surface of case-3.
5.2.4 Case 4

This case studies a casting with relatively thick wall. The geometry of the part is shown in Figure 5.17. The model includes runner and overflows.

The qualitative reasoning result are shown in Figure 5.18. The numerical simulation in this case is also done with MAGMASoft. Figure 5.19 shows the temperature pattern on the die surface when 50% of the casting solidifies. Figure 5.20 shows the temperature pattern on the die surface when 90% of the casting solidifies. From both of the patterns, it can be seen that the hot spots in this case is around the two heavy mass bosses. The qualitative reasoning result displays hot spots roughly at the same location as that on the numerical simulation result.
Figure 5.17. Geometry of the part of case-4.
Figure 5.18. Qualitative reasoning result: temperature pattern on the cavity surface of case-4.
Figure 5.19. Numerical simulation result: temperature pattern on the die surface when 50% of the casting solidifies.
Figure 5.20. Numerical simulation result: temperature pattern on the die surface when 90% of the casting solidifies.
5.2.5 Run Time

All the four cases were tested on Windows-NT workstation with PIII-400Mhz CPU and 128Mb RAM, which is the typical configuration used by the designers. Given the fast pace of the computer industry, the latest front end workstation is much powerful, in terms of CPU speed and RAM size, than the platform used in this test (ZDNet, 2000).

The run time measured for the cases is the user time of the temperature evaluation process only. The model preparation time including reading the STL model, voxelization etc. is not measured in the cases because this time is relatively very small and is in the order or seconds. The size of the voxel model and the detail running time for each of these cases are listed in the following Table 5.1.

The running time for all the four cases are in the order or minute. We can conclude that this qualitative reasoning approach can support the thermal design evaluation interactively.

<table>
<thead>
<tr>
<th>OS: Windows NT 4.0</th>
<th>size of voxel mode (dimX<em>dimY</em>dimZ)</th>
<th>running time (user time) (minute'second&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU: Intel P-III 400Mhz.</td>
<td>174<em>308</em>71</td>
<td>4'56&quot;</td>
</tr>
<tr>
<td>RAM: 128Mb.</td>
<td>174<em>308</em>71</td>
<td>6'17&quot;</td>
</tr>
<tr>
<td></td>
<td>246<em>65</em>280</td>
<td>2'10&quot;</td>
</tr>
<tr>
<td></td>
<td>196<em>310</em>50</td>
<td>5'38&quot;</td>
</tr>
</tbody>
</table>

Table 5.1. Model size and running time for the four cases.

5.3 Quantitative Verification

In the quantitative verification, we compute the percentage of difference between the numerical simulation results and the qualitative reasoning result. The percentage of dif-
ference will give a ballpark perception of accuracy of the qualitative reasoning. We also plot the histogram of the temperature difference and the regions which have the maximum difference to illustrate the distribution of the difference.

To compare the numerical result and the qualitative result, we mapped the qualitative result to the same FEM mesh used in the numerical simulation, and compute the average percentage of error at each node. Suppose the mesh has \( n \) nodes. The temperature value at the nodes from the numerical simulation is:

\[
a = \{\alpha_i \mid \alpha_i \text{ is temperature at node } i, i = 1, 2, \ldots n\}
\]

The temperature value from qualitative reasoning is:

\[
b = \{\beta_i \mid \beta_i \text{ is temperature at node } i, i = 1, 2, \ldots n\}
\]

The difference between \( A \) and \( B \) is:

\[
d = \{\delta_i \mid \delta_i = \alpha_i - \beta_i, i = 1, 2, \ldots n\}
\]

The percentage of error at each node is the absolute error between the numerical simulation temperature value and the qualitative temperature value at a given node divided by the numerical simulation temperature value, as shown in Eq. (5.1):
Thus, the average percentage of error among the mesh is as shown in Eq. (5.2):

$$
\bar{E} = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}
$$

We apply this analysis procedure to the case-2. The comparison result is shown in Table 5.2:

<table>
<thead>
<tr>
<th></th>
<th>min (°C)</th>
<th>max (°C)</th>
<th>mean (°C)</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>numerical simulation</td>
<td>36</td>
<td>358</td>
<td>179</td>
<td>27.4%</td>
</tr>
<tr>
<td>qualitative reasoning</td>
<td>41</td>
<td>404</td>
<td>180</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.2.** Quantitative comparison for case-2.

The average percentage of error for this case is 27.4%, which means that the accuracy of qualitative reasoning compared against numerical simulation is about 70%. For conceptual design evaluation, this accuracy is sufficient to help the designers to figure out the major design characteristics and potential problems.

In order to know the distribution of the error, the histogram of the difference between the numerical and the qualitative temperature at each node is plotted and shown in Figure 5.21. The distribution is a near normal distribution around the mean value 0°C. The average absolute difference is 50°C. The maximum absolute difference is 179°C.
Figure 5.21. Histogram of the temperature difference $D$ for case-2.

shows the distribution of the temperature difference over the nodes at which the difference is more than 100°C. From this plot, we can see the numerical simulation value is higher than the qualitative reasoning result on the cover die, but lower on the ejector die.
5.4 Design Study

The objective of the design study is to illustrate the evaluation process and result of the cooling line number and location. The achieve quality casting and production rate, the die-casting dies have to be cooled properly so that the casting alloy solidifies roughly by the same rate within the given cycle time. Herman (1974) states that one criterion to achieve the same solidification rate of casting alloy is to maintain uniform temperature across the die cavity surface. Uniform surface temperature may result uniform dendrite growth rate of
casting alloy if the wall thickness of casting is uniform. If the wall thickness of the casting
is not uniform, extra cooling lines are needed near the thick region.

The uniformity of the cavity surface temperature is evaluated using the standard
deviation of the surface temperature, as shown in Eq. (5.3).

\[ \sigma = \sqrt{\frac{(T_i - \bar{T})^2}{n-1}} \]  

(5.3)

where:

\( T_i \) : is the temperature at voxel i.
\( \bar{T} \) : is the average temperature on the cavity surface.
\( n \) : is the total number of surface voxels.

The adaptor part used in case-4 is also used here to illustrate the design study pro­cess. The dimension of the die block and cooling line location is shown in Figure 5.23. The study results are listed in Table 5.3. The temperature patterns are shown in The design variables are 1) the number of cooling lines; 2) the location of the cooling lines. The first case in the table has no cooling line, we can see the temperature deviation is as high as 1098, and high temperature flame in the temperature pattern. When two cooling lines were added near the biscuit area, the surface temperature deviation drops to 1033, and high temperature flame near the biscuit area is cooled down. When another two cooling lines were added at 0.75" to the cavity surface near the shoulder region of the part when high temperature were noticed (case-4 in Section 5.2.4), the surface temperature deviation drops to 1000. Then
another pare of cooling lines were added at various locations near the bottom region of the part, and we can see that the surface temperature deviation increases as the cooling lines were placed further away from the cavity surface. Comparing these cases, it seems that the six cooling lines at 0.75” location produces the minimum surface temperature deviation. The actual cooling lines are place at the 0.80” location.

By analyzing the surface temperature deviation, the designer can evaluate the compare the different cooling line design. However, the optimal location can not be achieved by this process.

**Figure 5.23.** Sketch of the die dimension and cooling line location for the adaptor part.
<table>
<thead>
<tr>
<th>Cooling line number and location</th>
<th>min (°C)</th>
<th>max (°C)</th>
<th>mean (°C)</th>
<th>deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No cooling line</td>
<td>59</td>
<td>444</td>
<td>358</td>
<td>1098</td>
</tr>
<tr>
<td>2 cooling lines at biscuit</td>
<td>53</td>
<td>427</td>
<td>330</td>
<td>1033</td>
</tr>
<tr>
<td>4 cooling lines (0.75&quot;)</td>
<td>43</td>
<td>417</td>
<td>318</td>
<td>1000</td>
</tr>
<tr>
<td>6 cooling lines (0.75&quot;)</td>
<td>43</td>
<td>415</td>
<td>311</td>
<td>978</td>
</tr>
<tr>
<td>6 cooling lines (1.00&quot;)</td>
<td>45</td>
<td>417</td>
<td>316</td>
<td>990</td>
</tr>
<tr>
<td>6 cooling lines (1.25&quot;)</td>
<td>46</td>
<td>418</td>
<td>318</td>
<td>997</td>
</tr>
</tbody>
</table>

Table 5.3. Cooling number and location vs. surface temperature variation.

Figure 5.24. Temperature patterns of the design case.
5.5 Sensitivity Analysis

The sensitivity of temperature pattern with respect to the voxel resolution is analyzed in this section. The "flat plate" part with a cooling line used in case-2 is also used here to illustrate the procedure and result of the analysis. The part is voxelized with 100, 200, and 300 sliced along the maximum dimension. The temperature sample data is taken along a line in the middle of the die, as shown in Figure 5.25.

The temperature data on this sample line for all the three cases with 100, 200 and 300 voxel resolution is listed in Figure 5.26. From the figure, we can see that the temperature distribution is not sensitive with respect to the voxel resolution, in other words, the qualitative temperature pattern is stable. However, because the part thickness pattern is used as the approximation of the casting temperature pattern, the voxel resolution should be high enough to represent the casting wall thickness.

Figure 5.25. Die block and sample line for the flat plate part.
5.6 Summary

Four cases studies have been presented in this chapter. Two aspects of this qualitative reasoning approach are evaluated through these case studies: the correctness of the qualitative thermal result and the running time to support interactive design. From these cases, we can conclude that the this qualitative reasoning approach gives non-misleading ball-park thermal result for the die-casting dies, and it is also efficient enough to support interactive conceptual design evaluation. The qualitative temperature result is also not sensitive with respect to voxel resolution.
CHAPTER 6

JUMP START OF NUMERICAL SIMULATION

6.1 Introduction

Besides the thermal design evaluation, the other application of this qualitative geometric reasoning method is to jump start the numerical simulation of die-casting thermal processes.

The die casting process is a periodic operation which is measured by “cycles”. To produce quality castings, the process must be at steady state, i.e. the total amount of heat going to the die is balanced with the total amount of heat going out of the die. In the actual production process, the dies are either in room temperature or a pre-heated temperature. The dies then start from this initial temperature and go through many “start-up” cycles and finally reach the steady state at which the production of quality casting begins. Correspondingly, the numerical simulation of die casting process follows the same physical route: starting from either room temperature or pre-heated temperature, going through the start-up cycles and reaching the steady state.

This simulation schema can, of course, reveal the transient thermal characteristics during the start-up period. But, from the design point of view, many the design decisions regarding to thermal characteristics of the part and die are made upon the steady state phase, such as cooling line. Thus, to the designers, the steady state is of more interests. One or a
few cycles in the steady state are needed for design decision making. Now, the problem is
that the simulation has to go through the start-up cycles to get to the steady state phase,
which normally takes a great amount of computational time, relatively or absolutely. For
example, the following Figure 6.1 shows a temperature profile of a node in the die simu-
lated using ABAQUS. The curve shows that it takes about 100 cycles to get to the steady
state condition. This example was run on a Window-NT workstation with P-III 400Mhz
CPU and 128Mb RAM, each cycle takes about 3 hours, and the total time required for 100
cycles to reach the steady state is about 12.5 day, roughly two weeks. This time is quite long
even if the objective of the simulation is only for the final design check.

Now, the problem for the numerical simulation is how to make the simulation
quickly get to the steady state.

There has been efforts made by researchers to simplify the numerical simulation
process based on rudimentary assumptions. The following section reviews the general sim-
ulation procedure and the acceleration approaches. However, many of these acceleration
approaches focused on simplifying the mathematical solving methods or simulation condi-
tions.

For the qualitative temperature computed by this geometric reasoning approach, we
assume that the die casting is already at the steady state, and this is why we can use the Kir-
chhoff's Law to solve the temperature network for temperature of a die voxel. Thus, the qual-
itative temperature, by assumption, is at the steady state. The straightforward utilization of
this result is to plug it into the numerical simulation as the initial condition to jump-start the
simulation. The expectation is to reduce the start-up cycles and hence save the computation
time to reach the periodic steady state.
6.2 Numerical Simulation Approach

Computer-based numerical simulation has been a long time favorite approach for die casting thermal analysis. Some of the works have been reviewed in Chapter 2. This sec-

Figure 6.1. Illustration of the temperature profile of a node in die using numerical package (ABAQUS).
tion will discuss the mathematical modeling and generic simulation procedure. Some acceleration methods will be reviewed as well.

### 6.2.1 Heat Transfer in Die Casting

The physical phenomena involved in the die casting process include fluid flow, solidification, and heat transfer. Mathematical models capable of simulating various aspects of the die casting process have been developed as aids to die design and process optimization and there have been a great amount of literatures on the modeling method (see Section 2.4). The generic modeling method described herein is summarized from the literatures such as (Stefanescu, 1990; Frayce, and Loong, 1991; Hu, et al., 1992; Barone, and Caulk, 1993) and is used in our simulation practice.

A schematic illustration of the components and interfaces in a die casting system is given in Figure 6.2. The components and their denotations are: B for holder blocks, D for inserts, S for slides, C for casting cavity, W for cooling lines, and A for the environment. The components consisting of a die include holder blocks, inserts, cores and slides, which are normally of the same material. For the convenience of notation, these components are all called “die” herein. The interface regarding to the heat conduction in the systems can be divided into: the cavity surface ($\delta C$), where the die is in contact with the casting; the die surface ($\delta D$), where the die components, such as inserts, hold blocks, and slides, are in contact with each other; cooling surface ($\delta W$), which are just the internal surfaces of the cooling lines; and the outer surface ($\delta A$), which are contact with either the air of the die casting machine.
The thermal behavior involves the process of filling, solidification and heat conduction between the components. It is commonly assumed that the die casting operation is in steady-state cycle. The metal is also assumed to be injected in the cavity at zero time duration and has the same initial temperature. These assumptions are valid if the objective of the simulation is only for the thermal analysis and thermal design aid.

The more comprehensive analysis modeling technique is to solve the fluid flow and heat transfer equations simultaneously to get the overall filling pattern and solidification pattern. But this will take tremendous efforts of preparation and computation. This approach is beyond the scope of this thesis.

Depending on the size of the casting, the liquid metal can take from 10 to 500ms to fill the cavity (Barone and Caulk, 1993). This period of time is small compared with the total cycle time, and the heat transfer during flow is normally neglected, which is called "instant fill" assumption. The cavity is considered to be filled instantly with no time. Thus, the aspects involved in the thermal analysis are mainly solidification of liquid casting metal and heat conduction across the surfaces described above. The cycle time is divided into two parts: residence time (which is also called dwell time in literatures), when the cavity is closed and the casting is in the cavity; opening time (which is also called leading time), when the die is opened. The total cycle time is denoted as "$t_c$", the residence time is denoted as "$t_r$" and open time is denoted as "$t_o$". The relationship between them is: $t_c = t_r + t_o$. 

128
Figure 6.2. Schematic illustration of die casting components and interfaces.

The equation governing the thermal history inside the casting is obtained by applying the law of energy conservation to the casting material:

$$\frac{\partial T_c}{\partial t} = \alpha_c \nabla^2 T_c + Q, \quad \text{in } C \text{ for } 0 \leq t \leq t_r$$  \hspace{1cm} (6.1)
The definition of the terms are:

\( \alpha_c \): thermal diffusivity of the casting material, \( = \frac{K}{(\rho C_p)} \).

\( C_p \): specific heat of casting material.

\( \rho \): density of the casting material.

\( K \): thermal conductivity of the casting material.

\( Q \): rate of latent heat generation.

\( t_r \): residence time, when die is closed and casting is in the cavity.

The left term indicates the overall temperature change of casting as a function of time. The first term in the right is the heat conduction caused by temperature gradients and heat flux through the cavity surface. The source term, i.e. the second on the right, describes the latent heat evolution during the liquid-solid phase transformation, which is defined as the following Eq. (6.2):

\[
Q = L \frac{df_s}{dt}
\]  \hspace{1cm} (6.2)

where:

\( L \): is the latent heat of the solidifying phase.

\( \frac{df_s}{dt} \): is the rate of evolution of the fraction of solid.

Similar as the casting, the heat conduction inside the die material can be modeled by:
\[ \frac{\partial T_d}{\partial t} = \alpha_d \nabla^2 T_d \quad \text{in } D \text{ for } 0 \leq t \leq t_c \quad (6.3) \]

Where:
\[ \alpha_d : \text{thermal diffusibility of the die material, } (\equiv \frac{K}{\rho C_p}). \]
\[ t_c : \text{total time of a cycle.} \]

Neglecting the radiation, and assuming constant thermal and physical properties of the die material, the boundary condition on the surfaces during the residence time are given by:

\[ -k_d \frac{\partial T_d}{\partial n} = \begin{cases} 
  h_c (T_d - T_c) & \text{on } dC \\
  h_d \Delta T_d & \text{on } dD \\
  h_a (T_d - T_\infty) & \text{on } dA \\
  h_w (T_d - T_w) & \text{on } dW 
\end{cases}, \text{ for } 0 \leq t \leq t_r \quad (6.4) \]

where:
\[ k_d : \text{thermal conductivity of the die material.} \]
\[ n : \text{normal vector of the surfaces.} \]
\[ h_c : \text{heat transfer coefficient on the cavity surface.} \]
\[ h_d : \text{heat transfer coefficient on the die surface.} \]
\[ h_a : \text{heat transfer coefficient on the outer surface.} \]
\[ h_w : \text{heat transfer coefficient on the cooling surface.} \]
\[ T_c : \text{temperature of casting material.} \]
$T_d$: temperature of die material.

$T_a$: temperature of environment.

$T_w$: temperature of cooling media.

The boundary condition during the open time period are given as:

$$-k_d \frac{\partial T_d}{\partial n} = \left( \begin{array}{l} h_d(T_d - T_w) \text{ on } dC, dD, dA \\ h_w(T_d - T_w) \text{ on } dW \end{array} \right), \text{ for } t_r \leq t \leq t_c \quad (6.5)$$

Neglecting the heat transfer during filling and assuming a “instant fill”, the initial conduction of the casting temperature is as Eq. (6.6). $T^o$ is the average temperature of the liquid metal during injection.

$$T_c(t=0) = T^o \quad (6.6)$$

### 6.2.2 Simulation Procedure

The primary characteristic of the die casting process is its periodic cycle. This means the thermal disturbance caused by the previous castings remains in the die to affect the solidification and temperature distribution of the next casting. The die temperature keeps to increase as more castings are made, until the process reaches steady state and the die temperature becomes periodic. For thermal design aid, the solution at the steady state are the most important. However, at steady state, the heat conduction problem of the casting and the die are quite different. The casting temperature explicitly known at the moment of
injection, whereas the initial die temperature are determined implicitly by the periodic condition.

The numerical simulation is to solve the casting temperature Eq. (6.1) and the die temperature Eq. (6.3) which are coupled by the boundary condition on the surfaces. Since the casting temperature subjects to the initial condition, solving Eq. (6.1) is essentially an initial value problem. The die temperature changes periodically at each cycle, and solving Eq. (6.3) is a steady periodic problem (Barone and Caulk, 1993). Though solving each of the problem separately is not difficult, but the combination of these two problems are difficult to solve directly.

The classical method to solve the steady state periodic problem is to represent the boundary conduction with a series of harmonics and use Fourier Transformation (Baker and Overman II, 1996). But since the boundary condition of the cavity surfaces is linked with the initial condition problem of the casting, the periodic boundary condition remains unknown before solving the initial problem of the casting. This makes the harmonic presentation of the periodic boundary condition difficult to be applied in this context. An alternative solution is to treat both the casting and the die as initial value problems and integrate both solutions through enough consecutive cycles for the die temperature to become periodic (Kearl, 1986). The problem now becomes how to choose the initial temperature of the die to make the numerical simulation process reaches steady periodic state as fast as possible.

The solution to solve for the steady state die temperature distribution lies in one of the two ways:
1. Some distribution of the die temperature is arbitrarily assigned at the beginning of a steady-state cycle (Riegger, 1981; Ohtsuka et al., 1982; and Granchi, 1983);

2. The steady periodic solution is achieved asymptotically by starting the die from the room temperature and solving for the entire start-up transient process (Grand, 1981; Kearns, 1986).

The first approach relies on an ad-hoc assumption. Since the deepest layers of the die never feel the effects of the thermal shock from casting, the temperatures in these layers are assumed to be linearly decreasing with depth. Temperature at the cavity surfaces are assumed to be known and the values are derived from the experience of similar cases or are imposed by the need of correct working of the die. Thus the temperature in the linear zone can be solve easily. This approach depends too much on the historic data and experience, it could be effective to some cases with simple casting geometry, but it could lead to unsolvable problems if the casting geometries are too complex or the historical data are lacking.

The second approach is physically rigorous and thorough, and is used by most of the commercial numerical packages. The initial temperature of the die can be either the room temperature or a preheated temperature. The steady periodic state can be reached after the simulation process going through many start-up cycles. Though the approach is close to the physical condition of the actual process, but numerically it can lead to unacceptably long computation time for the start-up cycles.

Barone and Caulk proposed another solution to the thermal analysis of die casting, which is capable of predicting the periodic die temperature at steady state without solving
for the start-up transient (Barone and Caulk, 1993). The validity of this approach relies on the fact that the all the transient heat conduction is directed perpendicular to the casting and the periodic temperature in the die penetrate only a short distance below the cavity surface, and this fluctuation layer normally very short compared to the overall die thickness (Booth, et al, 1981). Thus the transient heat flow in the thin layer can be represented as a finite polynomial with coefficients depending on time; and the rest bulk region are assumed to be independent of time and are linear to the distance away from the cavity surface. Barone and Caulk developed a whole set of tools and method make this approach practical in use.

6.3 Data Transfer From Voxel to Mesh Model

All methods including the acceleration method by Barone and Caulk described in the previous section are numerical-based solution. Qualitative reasoning provides another approximate assessment to the problem. As for this research, according to the equilibrium state assumption, the result of the qualitative temperature of the die are already at the steady state. The qualitative temperature is equivalent to the average temperature of the die of a cycle in the periodic steady state. In stead of using room temperature or the pre-heated temperature as the initial condition to the die, the qualitative temperature is used in the simulation to jump state the simulation.

The procedure to jump start the numerical simulation is as shown in Figure 6.3. The qualitative temperature is computed based on the STL model of the part. The result temperature of the insert is stored in a voxel model. The mesh was created with I-DEAS for numerical simulation. An input file to the numerical package was also create from the mesh model.
which contains the coordinates of all the node, the structure of every element, boundary condition and the initial conditions as the designated nodes. The simulation was done with ABAQUS. The critical step is to translate the qualitative temperature from the voxel model to the mesh model described in the input file. Two issues should be considered for the translation: orientation and resolution.

![Diagram](image)

**Figure 6.3.** Procedure of jump start the numerical simulation.

### 6.3.1 Orientation between Voxel and Mesh Model

Ideally, the orientation of the mesh model are expected to be the same as that of the voxel model. But this expectation does not hold all the time. The coordinate of the mesh is in a continues space with floating-point values. The voxel coordinate is a discrete integer grid on which the coordinates of voxels are the indices. This voxel model is transferred from the STL model which ideally should share the same coordinate with the mesh model. However, for most of the cases, due to the different coordinate systems used during the tri-
angulation, the STL mode normally has a different coordinate system from the mesh model. This makes the translation of voxel data to the mesh indirect and error-prone if correct measurement are not taken.

The coordinate transformation are divided into two steps. The process is shown in Figure 6.4.

First, the reversed translation was used to transfer the voxel space back to STL space. The integer indices of the voxel in the model are converted into floating-point coordinates in the STL space. This transformation is scaling and translation only because the voxel model is indeed produced from the STL model and they are well aligned. The transformation matrix is easy to form because the translation and scaling factors are all known during the voxelization procedure.

Next, the coordinate in the STL space must be mapped onto the mesh space. The operation in this step is to snap the two models into the same coordinate system. For most of the voxel model and mesh model generated from the same CAD system, their coordinate system are normally well aligned with the part bounding box. Thus the bounding box can be sued as the sources for the snapping operation. The geometric features, such as length and orientation of each edge of the bounding boxes are compared to find out the corresponding axes among the two coordinate system. For example, if the order of the edge length of the STL bounding box is $X > Y > Z$, while the order of the edge length of the bounding box of the mesh model is $Y' > Z' > X'$, then the correspondent axes between these two model is: $X \leftrightarrow Y'$, $Y \leftrightarrow Z'$, $Z \leftrightarrow X'$. If this inequality exists among the edges along each axis, this process can be automatic. If the inequality among the edge length of the bounding box does not exists, i.e. two or more length of the edge of each axis equal to
Figure 6.4. Transformation between voxel model and mesh model.

Each other, then there exists ambiguity and the process can not be automatic without further comparison of other features between the model.
Take the voxel model in STL space as example, suppose the new the coordinate in the current coordinate system is \( O(x_0, y_0, z_0) \), and the normalized vector of the three axes in the current system are: \( X(\xi_x, \psi_x, \xi_x) \), \( Y(\xi_y, \psi_y, \xi_y) \), and \( Z(\xi_z, \psi_z, \xi_z) \), then the matrix which transforms a point \( P \begin{bmatrix} x \ y \ z \ 1 \end{bmatrix}^T \) in the current system in the current system to the point \( P \begin{bmatrix} x' \ y' \ z' \ 1 \end{bmatrix}^T \) in the new system is:

\[
\begin{bmatrix}
    x' \\
    y' \\
    z' \\
    1
\end{bmatrix} = \begin{bmatrix}
    \xi_x & \psi_x & \xi_x & 1 \\
    \xi_x & \psi_x & \xi_x & 1 \\
    \xi_x & \psi_x & \xi_x & 1 \\
    x_0 & y_0 & z_0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]

The transformation of the mesh model can be derived in a similar manner. Once the two models are snapped to the same orientation in the same coordinate system, one of the two models, either the Voxel model or the mesh model, is scaled to the same size with the other one. Data translation can now be performed between the two models.

### 6.3.2 Resolution between Voxel and Mesh Model

The next issue in the data transformation is the sampling resolution of the two models. The resolution of the mesh model is measured by the distance between the consecutive nodes, and the resolution of the voxel model is measured by the voxel size. The ideal case is that each on the node in the mesh is exactly correspondent to one voxel in the voxel model. But this is hardly the reality, and resolution of the two model does not match.
If the mesh resolution is higher than or equal to the voxel resolution, i.e. the distance between two nodes is less or equal to the voxel size, then the solution is very obvious: for each of the node, find the nearest correspondent voxel in the voxel model and take the temperature value of the that voxel.

If the resolution of mesh model is lower than the voxel model, i.e. the distance between the two nodes on the mesh model is bigger than the voxel size, one node is correspondent to a collection of voxels in a region surrounding the node. The average temperature of the voxels in this region is taken as the temperature at the node. The problem now is how to define the “effect region” of a node and find the voxels in this region.

The element in the mesh model and normally polyhedron with many nodes. In the current mesh model used in this experiment, the element is tetrahedron, as shown in Figure 6.5-a, which has four nodes. The mesh model can also be viewed as a set of nodes \( P := \{ p_1, p_2, \ldots, p_n \} \). For a given node \( p_i \) in the mesh model, there exist a collection edges, \( E := \{ e_{ij}, e_{i2}, \ldots, e_{ik} \} \), connecting this node to its neighbor nodes \( N := \{ p_{i1}, p_{i2}, \ldots, p_{ik} \} \), as Figure 6.5-b shows.

The “effect region” of a node can be represented by the Voronoi cell in 3D space. The Voronoi cell defines a region where every point in this region has the nearest distance to the given node among the node set in 3D space (Berg, et al, 1997). The mathematical property of the Voronoi cell \( V(p_i) \) of a point \( p_i \in P = \{ p_1, p_2, \ldots, p_n \} \) is that for every point \( q \in \Omega_i \), there exists:

\[
\text{dist}(q, p_i) < \text{dist}(q, p_j), \text{ for } j = 1, 2, \ldots, n, \text{ and } i \neq j. \tag{6.7}
\]
where: $\text{dist}(q, p) = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2 + (q_z - p_z)^2}$, is the distance between point $q$ and $p$.

For two nodes which are neighbors in the mesh (i.e. they are connected by an edge of element), $p_i$ and $p_j$, there exists a plane $L( a \cdot x + b \cdot y + c \cdot z + D = 0 )$ which is a perpendicular bisector of line segment $\overline{p_ip_j}$ such that
\[ \text{dist}(p, L) = \text{dist}(p_j, L) \] (6.8)

where: \[ \text{dist}(p, L) = \frac{a \cdot x_p + b \cdot y_p + c \cdot z_p + D}{\sqrt{a^2 + b^2 + c^2}} \] is the distance from point \( p \) to plane \( L \).

This plane splits the space into two half space. We denote the open half space that contains \( p_i \) by \( h(p_i, p_j) \), and then the open half space that contains \( p_j \) by \( h(p_j, p_i) \), notice that \( q \in h(p_i, p_j) \) if and only if \( \text{dist}(q, p_i) < \text{dist}(q, p_j) \). Then the Voronoi cell of node \( p_i \) can be defined as:

\[ V(p_i) = \bigcap_{j \in \{p_i, p_2, \ldots, p_k\}} h(p_i, p_j) \] (6.9)

where: \( p_j \in \{p_1, p_2, \ldots, p_k\} \) are the neighbor nodes of \( p_i \).

Thus \( V(p_i) \) is the intersection of \( k \) half spaces and, hence a closed convex polyhedron region bounded by \( k \) planes. If the node \( p_i \) is on the boundary, then its Voronoi cell defined by the half spaces and the boundary surface of the mesh model.

For a given pair of nodes \( p_i(x_i, y_i, z_i) \) and \( p_j(x_j, y_j, z_j) \), the coefficients of the bisector plane \( L(a,b,c,d) \) which defines the half space \( h(p_i, p_j) \) is derived as

\[
\begin{align*}
    a &= x_i - x_j \\
    b &= y_i - y_j \\
    c &= z_i - z_j \\
    d &= \frac{[a(x_i + x_j) + b(y_i + y_j) + c(z_i + z_j)]}{2}
\end{align*}
\]

and \( a \cdot x_i + b \cdot y_i + c \cdot z_i + d > 0 \).
To get the explicit representation of the polyhedron of a Voronoi cell, it needs to compute the intersection of the $k$ bisector planes, which is not numerically efficient. The primary objective to find the Voronoi cell of the node is to find the voxels lying inside this Voronoi cell. The property of the half space can be used to find the voxels in the Voronoi cell but avoid computing the vertices of the Voronoi cell.

Suppose the $k$ planes of the Voronoi cell are:

$$\begin{align*}
    a_1x + b_1y + c_1z + d_1 &= 0 \\
    a_2x + b_2y + c_2z + d_2 &= 0 \\
    \vdots \\
    a_kx + b_ky + c_kz + d_k &= 0
\end{align*}$$

(6.10)

Where the normal vector of each plane is pointing into the inner portion of the Voronoi cell. Thus, for a given point $p(x_p, y_p, z_p)$, it lies in the Voronoi cell if and only if

$$\begin{align*}
    a_1x + b_1y + c_1z + d_1 &< 0 \\
    a_2x + b_2y + c_2z + d_2 &< 0 \\
    \vdots \\
    a_kx + b_ky + c_kz + d_k &< 0
\end{align*}$$

(6.11)

Each row in Eq. (6.11) defines a half space, and the Voronoi cell is defined as the intersection all these half spaces.

Based on this property, an algorithm to find retrieve the voxels are inside the Voronoi cell can be constructed as following Figure 6.6 shows.
void get_voxels_in_vcell
(
    int buf_size,  // size of the voxel buffer
    voxel_t *vox_buf, // pointer to the voxel buffer
    int k,  // the length of list pl
    plane_t *pl,  // list of planes defining the cell
    int& vn,  // the length of list vx
    voxel_t& *vx  // the list of voxels found inside the cell
)
{
    for(int i=0; i<buf_size; i++)
    {
        // current voxel
        voxel_var v = vox_buf[i];

        // flag to indicate whether the current voxel v is
        // inside of out side of the cell
        int inside = 1;

        for(int j=0; j<k; j++)
        {
            if(signof(pl[j].a*v.x + pl[j].b*v.y + pl[j].c*v.z + pl[j].d)<0))
                // if the the current voxel is the negetive side
                // of one plane, then it is not in the cell
                inside = 0;
        } // end of for(j)

        if(inside)
            // if the voxel is in the positive side of every plane
            // of the cell, then it is inside the cell. and the
            // current voxel is put in the return voxel list
            vx[vn++] = v;
    } // end of for(i)
}

Figure 6.6. Imperial algorithm finding the voxels inside the Voronoi cell.
The algorithm searches the entire buffer of the voxel model, which is not efficient. If the bounding box of the Voronoi cell is known, then the searching space could be limited to the tangent bounding box, thus improves efficiency. The bounding box is the rectangular polyhedron whose faces touch the minimum and maximum vertex of the Voronoi cell as shown in Figure 6.5-c.

The Voronoi cell is defined by the Voronoi half spaces, as Eq. (6.11) shows. In order to get the bounding box, the vertices of the Voronoi cell need to be computed first. Each of the vertex is an intersection of three planes. Given $n$ Voronoi planes, the maximum number of intersection points are $C_n^3$. Some of these points are vertices of Voronoi cell and some are not. The determine rule is that if an intersection point is the vertex of Voronoi cell, it must lies on the same side of half space of all the Voronoi planes. For example, suppose point $P(x, y, z)$ is the intersection of plane $L_1(a_1, b_1, c_1, d_1)$, $L_2(a_2, b_2, c_2, d_2)$, $L_3(a_3, b_3, c_3, d_3)$, then the coordinates of P is computed solving the standard linear equation Eq. (6.12):

$$\begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \\ a_3 x + b_3 y + c_3 z + d_3 = 0 \end{cases} \quad (6.12)$$

The condition for $P$ to be a vertex of Voronoi cell is as:

$$\begin{cases} a_4 x + b_4 y + c_4 z + d_4 < 0 \\ a_5 x + b_5 y + c_5 z + d_5 < 0 \\ \vdots \\ a_k x + b_k y + c_k z + d_k < 0 \end{cases} \quad (6.13)$$
This algorithm thoroughly checks all the intersection points and finds the one on the Voronoi cell, yet is not time efficient. The time complexity of $O(n^4)$ where $n$ is the number of half spaces. The number of half space in this case is also the number of connected edges of node in the mesh. However, given the particular situation of the Voronoi cell for a node in the FEM mesh, the number of connected edges is normally not very many and this through searching algorithm is satisfactory.

Now, having the filter bounding box, the searching can be limited into the bounding box. The main algorithm to transfer the data from voxel model to mesh model is illustrated in the following Figure 6.7.

This algorithm takes a node from the mesh and computes its Voronoi cell and find the bounding box of the cell, it then get all the voxels in the cell and compute the average temperature of those voxels in the Voronoi cell as the temperature of the input node. This process is repeated for every node in the mesh model and thus the temperature at every node can be obtained from the voxel model.
void voxel_to_mesh
(
    int buf_size, // size of the voxel buffer
    voxel_t *vox_buf, // pointer to the voxel buffer
    int n // the number of the nodes
    node_t *nodes, // list of nodes in the mesh model
)
{
    for(int i= 0; i<n; i++)
    {
        // get the current node
        node_t cur_node = nodes[i];

        // get the Voronoi cell of the current node
        int cur_cell = get_voronoi_cell(cur_node);

        // get the filter bounding box of the voronoi cell
        box_t cur_box = get_bounding_box(cur_node, cut_cell);

        // get the voxel lists within the bounding box
        int cur_num; // number of voxels
        voxel_t *cur_vox; // voxel list
        get_voxels_int_box(buf_size, vox_buf, cur_box,
            cur_num, cur_vox);

        // get the voxels in the Voronoi cell
        int nvox;
        voxel_t *vox_in_cell;
        get_voxels_in_vcell(cur_in_cell, cur_vox,
            cur_node.nplane, cur_node.planes,
            nvox, vox_in_cell);

        // compute the average temperature for the node
        cur_node.temp=0;
        for(int j=0; j<nv; j++)
            cur_node.temp = cur_node.temp + vox_in_cell[j].temp
        cur_node.temp = cur_node.temp/nvox;
    }// end offor(i)
}

Figure 6.7. Illustration of the main control algorithm to transfer data from voxel model to mesh model.
6.4 Simulation Process and Result Comparison

The simulation process and verification of the expectation is illustrated through an example.

The first example is a flat plate part which has four thin walls. The configuration of the model used in the simulation is shown in Figure 6.8. The model includes part, ejector and cover inserts, ejector and cover holder blocks, plunger, and cooling line. The model is meshed in I-DEAS and then exported to ABAQUS for simulation.

The boundary conditions are defined between: part and insert, ejector insert and cover insert, ejector insert and ejector holder block, covert insert and cover holder block, cooling line and inserts, cooling line and holder blocks, plunger and inserts, plunger and holder blocks, holder blocks and the environment. The heat transfer coefficients between the boundaries are listed in Table 6.1.

The total cycle time is 90 seconds. Each cycle is divided into 6 steps: die-closed, filling-solidification, die-open-with-part-on, die-open-with-part-ejected, spray, and idling. The time and explanation of each step is listed in the following Table 6.2.

<table>
<thead>
<tr>
<th></th>
<th>part</th>
<th>ejector die</th>
<th>cover die</th>
<th>ejector holder</th>
<th>cover holder</th>
<th>plunger</th>
<th>cooling line</th>
<th>environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>part</td>
<td>5,000</td>
<td>5,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ejector</td>
<td>7,000</td>
<td>7,000</td>
<td>7,000</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cover</td>
<td>7,000</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1. Heat transfer coefficient between the die casting components (W/m²°C).
Table 6.1. Heat transfer coefficient between the die casting components (W/m²°C).

<table>
<thead>
<tr>
<th>Component</th>
<th>7,000</th>
<th>10,000</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>ejector holder</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cover holder</td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>plunger</td>
<td>5,000</td>
<td>5,000</td>
<td>50</td>
</tr>
<tr>
<td>cooling line</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>environment</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2. Time steps in each cycle.

<table>
<thead>
<tr>
<th>Step</th>
<th>Time(sec)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>die-closed</td>
<td>1.0</td>
<td>die is closed, waiting for injection.</td>
</tr>
<tr>
<td>filling-solidification</td>
<td>9.0</td>
<td>liquid metal is injected and solidifies.</td>
</tr>
<tr>
<td>die-open-part-on</td>
<td>7.0</td>
<td>die is opened, the part in still in the cavity.</td>
</tr>
<tr>
<td>die-open-part-ejected</td>
<td>10.0</td>
<td>die is opened and part is ejected.</td>
</tr>
<tr>
<td>spray</td>
<td>3.0</td>
<td>spraying</td>
</tr>
<tr>
<td>die-idling</td>
<td>60.0</td>
<td>die is opened, and idling for next cycle.</td>
</tr>
</tbody>
</table>

The numerical simulation and jump-start simulation are all set to the same boundary conductions and cycle steps except the initial temperature for the inserts. The numerical simulation takes the pre-heated temperature (80°C) as the initial condition for the two inserts. The jump-start simulation takes the qualitative temperature from the voxel model as the initial condition of the two inserts, and 60°C for the initial temperature of the holder blocks.
The simulations are done on Intel-based Windows NT workstation. The system configuration is shown in Table 6.3. Figure 6.9 to Figure 6.12 shows the results of the numerical simulation and the jump-start simulation. It takes about 100 cycles for the numerical simulation to get to the steady state, while it takes about 6 cycles for the jump-start simulation to get to the same steady state. The computational time required for the regular simulation and simulation with jump-start condition is compared as shown in Table 6.3. The time required for each step for are the same because the simulation configurations for the regular simulation and jump-start simulation are essentially the same. The total amount of time to reach the steady periodic state for the regular simulation is about two weeks, however the time required for the jump-start is just about one day. The computational resource is save remarkably.
Figure 6.8. Illustration of the flat plate casting and the die configuration.
Figure 6.9. Comparison of the simulation results with or without the jump start: temperature profile of node between top two walls.
Figure 6.10. Comparison of the simulation results with or without the jump start: temperature profile of node at the middle of the bottom two walls.
Figure 6.11. Comparison of the simulation results with or without the jump start: temperature profile of node beneath the runner.
Figure 6.12. Comparison of the simulation results with or without the jump start: average temperature of the ejector insert.
**Table 6.3.** Configuration of the computer and benchmark of the computation time.

<table>
<thead>
<tr>
<th></th>
<th>Regular simulation</th>
<th>Jump-start simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time for each cycle (hour)</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Number of cycles to reach steady state</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>Total time to reach steady state (hour)</td>
<td>18.0</td>
<td>300.0</td>
</tr>
</tbody>
</table>
CHAPTER 7

CONCLUSION AND RECOMMENDATION

7.1 Summary and Contribution

One of the challenges of implementing concurrent engineering and design for manufacturing in conceptual phase is to provide the suitable tools to the designer to evaluated the manufacturability efficiently. In this dissertation, we have developed the methodology and tools to support interactive thermal design evaluation for die-casting dies. Circuit analog method is used to model the heat flow in the die-casting system. Graphical solution of heat flow path and heat flow network is formed for each of the die voxel. Geometric reasoning approach to efficiently compute the thermal resistances along each of the heat flow path in the voxel mode. The die-casting system is assumed to at the quasi-steady state, which means the total heat input to the system within a cycle is balanced with the total heat output from the system within a cycle, while the heat redistributes between the thermal components in a cycle. The qualitative temperature distribution of the die is computed by solving the heat flow network and computing the thermal resistances based this steady-state assumption.

The qualitative temperature is equivalent to the average temperature of a cycle at the steady state. It is then plugged into the numerical simulation to jump start the thermal simulation of die casting.
This qualitative geometric reasoning approach provides the die-casting designer a fast and ballpark type of evaluation of the thermal characteristics of the die. It is proved to be efficient and not misleading in terms of the thermal design evaluation.

The contributions of this research are briefly listed as follows:

1. Applied the circuit analog of heat flow on the voxel model and modeling the heat flow for the die-casting system. The circuit analog has been the standard textbook-like method for heat transfer analysis. It has been used broadly especially in the thermal analysis and cooling line design for die-casting systems, and physical devices have been build based this method. We combined the circuit analog method with the 3D voxel model and proposed a software-oriented solution.

2. Developed the method to build the heat flow network for each die voxels and the solution to solve the network to determine the die temperature distribution. Steady-state heat flow assumption is made for the die-casting system so as to solve the network. Certain simplification and assumptions are also made for the casting temperature, cooling and environment temperature in order to solve for the qualitative die temperature distribution efficiently but not losing the primary thermal characteristics of the distribution.

3. Developed the geometric reasoning approach for thermal resistance computation on the voxel model. The major computational task in this study is the computation of thermal resistances between the die voxel and the casting section, cooling
line and environment. The ray casting algorithm is developed to compute the thermal resistance efficiently.

4. Developed the approach to jump start the numerical simulation of the die-casting thermal process using qualitative temperature result obtained from the geometric reasoning. Certain algorithms and programs to transfer temperature data from the voxel model to the mesh model are developed as well to. The approach is proved to be able to save the start-up cycles required for the conventional numerical simulation to reach the steady state, hence reduce the total computational time significantly.

7.2 Future Recommendations

In this study, we use the discretized heat flow path along the first level of the neighboring directions of a voxel to approximate the actual heat flow in the die-casting system. This discretization of heat flow causes the die temperature pattern to be wedged and discontinues in some region of the voxel model. Furthermore, the discretization produces a systematic error of the temperature distribution. Thus, one future work would be to develop more accurate approximation of heat flow in the die-casting system. A feasible solution would be to consider that the heat concentrated on the skeleton of the casting and propagate throughout the casting and die volume continuously. The challenge would be how to determine the interaction when heat flow fronts meet each other.

Because of the wedged pattern of the die temperature in the voxel model, it is difficult to apply optimization techniques to search for the optimal location of the cooling
line. For example, the gradient search might result the cooling line to be in a misleading area. However, with the continues temperature pattern, it is possible to optimize the cooling location in the die.

The solidification pattern of the casting is used in this study to approximate the average temperature pattern of the casting in the steady state. Because the heat flow in the die casting system is intermittent and the majority of the heat load to the system is from the casting, the casting temperature dominates the heat flow and temperature distribution of the die. A better approximation of the casting temperature would significantly increase the correctness and accuracy of the die temperature.

The voxel model used in this study has a uniform-sized voxel throughout the casting and die region disregarding the geometric details. This might have too much problem for the casting such as thickness and flow pattern analysis because the size of the volume involved in such analysis is not too big and the model can be voxelized with high resolution. However, for the die temperature related analysis, the entire die region is needed, which increase the volume of the model significantly. Fine Voxelization in all the die areas will produce more accurate result but increasing the computation time because the temperature at the regions far away from the casting surface does not vary too much compared with that close to the casting surface. Thus, a non-uniform voxel model, which has finer resolution near the casting surface and rougher resolution away from the casting surface would satisfaction and accuracy yet reduce the computation time.
REFERENCES


168


