COUPLING HEAT TRANSFER AND FLUID FLOW SOLVERS FOR MULTI-DISCIPLINARY SIMULATIONS

by

Qingyun Liu

A Dissertation
Submitted to the Faculty of Mississippi State University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computational Engineering in the Bagley College of Engineering

Mississippi State, Mississippi

December 2003
COUPLING HEAT TRANSFER AND FLUID FLOW SOLVERS FOR
MULTI-DISCIPLINARY SIMULATIONS

by

Qingyun Liu

Approved:

Pasquale Cinnella
Professor of
Aerospace Engineering
(Director of Dissertation)

Edward Luke
Assistant Professor of
Computer Science and Engineering
(Committee Member)

Seth Oppenheimer
Professor of
Mathematics and Statistics
(Committee Member)

Clarence Burg
Assistant Professor of
Computational Engineering
(Committee Member)

Boyd Gatlin
Associate Professor of
Aerospace Engineering
(Committee Member and
Graduate Coordinator)

A. Wayne Bennett
Dean of the Bagley College of Engineering
The purpose of this study is to build, test, validate, and implement two heat transfer models, and couple them to an existing fluid flow solver, which can then be used for simulating multi-disciplinary problems. The first model is for heat conduction computations, the other one is a quasi-one-dimensional cooling channel model for water-cooled jacket structural analysis. The first model employs the integral, conservative form of the thermal energy equation, which is discretized by means of a finite-volume numerical scheme. A special algorithm is developed at the interface between the solid and fluid regions, in order to keep the heat flux consistent. The properties of the solid region materials can be temperature dependent, and different materials can be used in different parts of the domains, thanks to a multi-block gridding strategy. The cooling channel flow model is developed by using quasi-one-dimensional conservation laws of mass, momentum, and energy, taking into account the effects of heat transfer and friction. It is possible to have phase changes in the channel, and a mixture model
is applied, which allows two phases to be present, as long as they move at the same bulk velocity and vapor quality does not exceed relatively small values. The coupling process of both models (with the fluid solver and with each other) is handled within the Loci system, and is detailed in this study. A hot-air nozzle wall problem is simulated, and the computed results are validated with available experimental data. Finally, a more complex case involving the water-cooled nozzle of a Rocket Based Combined Cycle (RBCC) gaseous oxygen/gaseous hydrogen thruster is simulated, which involves all three models, fully coupled. The calculated temperatures in the nozzle wall and at the cooling channel outlet compare favorably with experimental data.

Keywords: numerical heat conduction, numerical heat convection, cooling channel, model coupling, multi-disciplinary simulations
ACKNOWLEDGMENTS

First of all, the author would like to express his sincere gratitude to his major professor, Dr. Pasquale Cinnella for his patient guidance, encouragement and friendship throughout this entire study. Special thanks also goes to the members of the author’s doctoral committee, Drs. Edward Luke, Clarence Burg, Seth F. Oppenheimer, and Boyd Gatlin, for their valuable advice that has enhanced the quality of this thesis. The author also would like to give thanks to Drs. Lin Tang, Xiao-Ling Tong, Junxiao Wu, and Mr. Thomas George for their friendship and many valuable and invigorating technical discussions.

This research was supported by NASA Marshall Flight Research Center with Jeff West serving as a very enthusiastic, spirited, and encouraging technical monitor. This support is gratefully acknowledged. Thanks also goes to the Department of Aerospace Engineering at Mississippi State University for providing the author with initial support over the first one and half years.

The author would like to express his great thanks to his parents, and his sincere love to his wife and daughter. He could not have finished this program without their love, encouragement, patience, and support.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 A Brief Review of Heat Transfer Models</td>
<td>1</td>
</tr>
<tr>
<td>1.2 A Brief Review of Previous Work</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Goals of This Study</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Multi-Disciplinary Simulations Environment: Loci</td>
<td>6</td>
</tr>
<tr>
<td>1.5 Table of Contents</td>
<td>8</td>
</tr>
<tr>
<td>II. INTRODUCTION TO THE FLOW SOLVER (CHEM)</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Governing Equations</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Turbulent Models</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Temperature Gradient Computation</td>
<td>14</td>
</tr>
<tr>
<td>2.4 Thermal Boundary Condition</td>
<td>16</td>
</tr>
<tr>
<td>III. HEAT TRANSFER MODELS</td>
<td>17</td>
</tr>
<tr>
<td>3.1 Governing Equations for Solid Heat Conduction</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Governing Equations for Cooling Channel Flow</td>
<td>18</td>
</tr>
<tr>
<td>3.3 Water Steam Thermodynamic Model</td>
<td>21</td>
</tr>
<tr>
<td>3.4 Numerical Scheme for Solid Heat Conduction</td>
<td>23</td>
</tr>
<tr>
<td>3.5 Coupling CHEM with the Solid Heat Conduction Model</td>
<td>27</td>
</tr>
<tr>
<td>3.6 Numerical Scheme for Cooling Channel Flow</td>
<td>29</td>
</tr>
<tr>
<td>3.7 Coupling Solid Heat Conduction and Cooling Channel Flow</td>
<td>34</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>IV. DATA MODEL AND COMPUTATION RULES</td>
<td>36</td>
</tr>
<tr>
<td>4.1 Introduction to Loci</td>
<td>36</td>
</tr>
<tr>
<td>4.2 Solid Block Mesh Representation</td>
<td>39</td>
</tr>
<tr>
<td>4.3 Solid Heat Transfer Model Implementation</td>
<td>42</td>
</tr>
<tr>
<td>4.4 Cooling Channel Model</td>
<td>45</td>
</tr>
<tr>
<td>4.5 Coupling of the Models</td>
<td>46</td>
</tr>
<tr>
<td>V. TEST CASES</td>
<td>48</td>
</tr>
<tr>
<td>5.1 Verification of Heat Conduction Solver</td>
<td>48</td>
</tr>
<tr>
<td>5.1.1 Two Manufactured Solutions</td>
<td>50</td>
</tr>
<tr>
<td>5.1.2 Iteration Convergence</td>
<td>52</td>
</tr>
<tr>
<td>5.1.3 Spatial Convergence</td>
<td>52</td>
</tr>
<tr>
<td>5.1.4 Temporal Convergence</td>
<td>55</td>
</tr>
<tr>
<td>5.2 Composite Solid Block</td>
<td>58</td>
</tr>
<tr>
<td>5.3 Cooling Channel Test Cases</td>
<td>62</td>
</tr>
<tr>
<td>VI. VALIDATION RESULTS</td>
<td>73</td>
</tr>
<tr>
<td>6.1 Air in a Cooled Converging-Diverging Nozzle</td>
<td>73</td>
</tr>
<tr>
<td>6.2 RBCC Thruster Nozzle Simulation</td>
<td>82</td>
</tr>
<tr>
<td>VII. SUMMARY AND FUTURE WORK</td>
<td>100</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>102</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Thermodynamic Property Values in Two-Phase Region</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Water Viscosity and Conductivity</td>
<td>23</td>
</tr>
<tr>
<td>4.1 Solid Block Mesh Descriptions</td>
<td>41</td>
</tr>
<tr>
<td>4.2 Interface Maps</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Cooling Channel Grid Description</td>
<td>46</td>
</tr>
<tr>
<td>5.1 Gauss-Seidel Iteration Convergence History</td>
<td>52</td>
</tr>
<tr>
<td>5.2 Newton Iteration Convergence History</td>
<td>53</td>
</tr>
<tr>
<td>5.3 Second and Infinite Norm Errors for Spatial Convergence</td>
<td>54</td>
</tr>
<tr>
<td>5.4 Second and Infinite Norm Errors for Spatial Convergence</td>
<td>55</td>
</tr>
<tr>
<td>5.5 Second and Infinite Norm Errors for Temporal Convergence</td>
<td>57</td>
</tr>
<tr>
<td>5.6 Thermal Properties of Solid Block Materials</td>
<td>59</td>
</tr>
<tr>
<td>5.7 Cooling Channel Inlet Condition of the First Case</td>
<td>65</td>
</tr>
<tr>
<td>5.8 Operating Parameters for Solid Block of the First Case</td>
<td>66</td>
</tr>
<tr>
<td>5.9 Cooling Channel Inlet Condition of the Second Case</td>
<td>66</td>
</tr>
<tr>
<td>5.10 Operating Parameters for Solid Block of the Second Case</td>
<td>66</td>
</tr>
<tr>
<td>6.1 Operating Parameters of Nozzle Flow</td>
<td>74</td>
</tr>
<tr>
<td>6.2 Physical and Thermal Properties of AISI Stainless Steels at 400K</td>
<td>76</td>
</tr>
<tr>
<td>6.3 Inflow Conditions and Computational Model</td>
<td>88</td>
</tr>
<tr>
<td>6.4 Operating Parameters for Thruster Nozzle Wall</td>
<td>89</td>
</tr>
</tbody>
</table>
TABLE

<table>
<thead>
<tr>
<th></th>
<th>Cooling Channel Inlet Condition of the Thruster Nozzle</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>Coolng Channel Inlet Condition of the Thruster Nozzle</td>
<td>90</td>
</tr>
<tr>
<td>6.6</td>
<td>Cooling Channel Outflow Temperature Comparison</td>
<td>96</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Control Space for Cooling Channel</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Cooling Channel in Solid Block</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>Residual History for the First Function</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Error Norm for Spatial Convergence</td>
<td>55</td>
</tr>
<tr>
<td>5.3</td>
<td>Exact and Numerical Solution for Second Function</td>
<td>56</td>
</tr>
<tr>
<td>5.4</td>
<td>Error Norm for Spatial Convergence</td>
<td>56</td>
</tr>
<tr>
<td>5.5</td>
<td>Error Norm for Temporal Convergence</td>
<td>57</td>
</tr>
<tr>
<td>5.6</td>
<td>Temperature for Three Copper Blocks</td>
<td>59</td>
</tr>
<tr>
<td>5.7</td>
<td>Temperature for Copper-Aluminum Blocks</td>
<td>60</td>
</tr>
<tr>
<td>5.8</td>
<td>Temperature for Copper-Brass Blocks</td>
<td>60</td>
</tr>
<tr>
<td>5.9</td>
<td>Temperature for Copper-Bronze Blocks</td>
<td>61</td>
</tr>
<tr>
<td>5.10</td>
<td>Temperature for Copper-Brass-Bronze Blocks at $t = 100s$</td>
<td>62</td>
</tr>
<tr>
<td>5.11</td>
<td>Temperature for Copper-Brass-Bronze Blocks at $t = 200s$</td>
<td>63</td>
</tr>
<tr>
<td>5.12</td>
<td>Temperature for Copper-Brass-Bronze Blocks at $t = 300s$</td>
<td>63</td>
</tr>
<tr>
<td>5.13</td>
<td>Temperature for Copper-Brass-Bronze Blocks at $t = 400s$</td>
<td>64</td>
</tr>
<tr>
<td>5.14</td>
<td>Temperature for Copper-Brass-Bronze Blocks at $t = 500s$</td>
<td>64</td>
</tr>
<tr>
<td>5.15</td>
<td>Cooling Path for the First Case</td>
<td>67</td>
</tr>
<tr>
<td>5.16</td>
<td>Convergence History for Cooling Channel and Solid</td>
<td>67</td>
</tr>
<tr>
<td>5.17</td>
<td>Solid Block Temperature for the First Case</td>
<td>68</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>5.18 Cooling Channel Temperature for the First Case</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>5.19 Cooling Channel Pressure for the First Case</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>5.20 Cooling Channel Specific Enthalpy for the First Case</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>5.21 Cooling Channel Vapor Quality for the First Case</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5.22 Cooling Path for the Second Case</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5.23 Solid Block Temperature for the Second Case</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>5.24 Cooling Channel Temperature for the Second Case</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>5.25 Cooling Channel Pressure for the Second Case</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>5.26 Cooling Channel Specific Enthalpy for the Second Case</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>6.1 Hot Air Nozzle Flow and Solid Wall Mesh</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>6.2 Hot Air Nozzle Wall Outside Temperature</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>6.3 Convergence History for Solid and Flow Solvers</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>6.4 Nozzle Flow Density</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>6.5 Nozzle Flow Pressure</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>6.6 Nozzle Flow Temperature</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>6.7 Nozzle Flow Velocity</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>6.8 Nozzle Flow Mach Number</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>6.9 Nozzle Wall Temperature (AISI302)</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>6.10 Nozzle Wall Temperature (AISI316)</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>6.11 Nozzle Flow and Solid Wall Temperature</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>6.12 Comparison of Wall Temperature</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>6.13 Comparison of Wall Heat Flux</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>6.14 Comparison of Wall Heat Transfer Coefficient</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>6.15 Comparison of Wall Temperature</td>
<td>83</td>
<td></td>
</tr>
</tbody>
</table>
6.16 Measurement Thruster Nozzle Section and Thermocouple Location .................................................. 84
6.17 Cooling Channel Passages in the Top Nozzle Wall ...................... 85
6.18 Cooling Channel Passages in the side Nozzle Wall ...................... 86
6.19 Thruster Fluid and Solid Grids .................................................. 87
6.20 Cooling Lines Arrangement ...................................................... 87
6.21 Thruster Nozzle Side Wall Temperature specified ...................... 89
6.22 Residual History for Fluid, Solid and Cooling Flow ................... 91
6.23 Flow Mach Number Distribution .............................................. 91
6.24 Flow Temperature Distribution .............................................. 92
6.25 Flow Pressure Distribution ...................................................... 92
6.26 Flow Velocity Distribution ...................................................... 92
6.27 Flow Velocity Vector ............................................................. 93
6.28 Solid Temperature Slices ......................................................... 93
6.29 Solid Temperature Slices ......................................................... 94
6.30 Solid Temperature Contour Slices ............................................. 94
6.31 Cooled Solid Temperature Slice with Cooling Path ..................... 95
6.32 Thrust Nozzle and Cooled Solid Wall Temperature Slices ............. 95
6.33 Cooling Channel Temperature .................................................. 96
6.34 Cooling Channel Inflow and Outflow Temperature ..................... 97
6.35 Solid and Fluid Interface Heat Flux .......................................... 98
6.36 Solid and Fluid Interface Temperature ...................................... 99
6.37 Thermocouple Temperature ..................................................... 99
NOMENCLATURE

Variables:

$A$  
Area of a face, or specific Helmholtz free energy

$C_v$  
Specific heat at constant volume

$C_p$  
Specific heat at constant pressure

$C_f$  
Friction coefficient

$D$  
Diameter

$E$  
Total energy, or error norm

$e$  
Roughness of cooling channel

$e_0$  
Fluid total energy

$F_i$  
Inviscid flux vector

$F_v$  
Viscous flux vector

$f$  
Friction factor of cooling channel

$f_{wall}$  
Friction at cooling channel wall

$f_{body}$  
Body force

$g$  
Gravity constant, or thermal source

$h$  
Specific enthalpy

$h_c$  
Heat transfer coefficient

$h_s$  
Species enthalpy

$k$  
Thermal conductivity

$I$  
Identity tensor

$L$  
Length of cooling cell

$m$  
Mass flow rate
\( Nu \)  
Nusselt number

\( n \)  
Time step

\( \tilde{n} \)  
Unit vector normal to a surface

\( Pr \)  
Prandtl number

\( Pr_t \)  
Turbulent Prandtl number

\( p \)  
Pressure

\( Q \)  
Vector of conservative variables

\( \tilde{q} \)  
Heat flux vector

\( r \)  
Channel radius

\( \tilde{r} \)  
Position vector

\( Re \)  
Reynolds number

\( S \)  
Specific entropy

\( T \)  
Temperature

\( T_{ref} \)  
Reference temperature

\( t \)  
Time

\( U \)  
Internal energy

\( \tilde{u} \)  
Fluid velocity vector

\( v \)  
Specific volume

\( V \)  
Velocity, or cell volume

\( \tilde{V}_s \)  
Species mass diffusion velocity

\( W_p \)  
Cooling channel wet perimeter

\( \tilde{W} \)  
Chemistry source term

\( w \)  
Vapor quality

\( \dot{w}_s \)  
Species chemical production rate

\( Y_s \)  
Species mass fraction \( (\rho_s/\rho) \)
\( z \)  
Elevation

\( \alpha \)  
Thermal diffusivity coefficient

\( \lambda \)  
Thermal conductivity coefficient

\( \mu \)  
Dynamic viscosity coefficient

\( \mu_t \)  
Turbulent viscosity coefficient

\( \Omega_c \)  
Control volume of cell

\( \partial \Omega_c \)  
Boundary of cell

\( \rho \)  
Density

\( \rho_s \)  
Species density

\( \tilde{\tau} \)  
Stress tensor

Logical Operators:

\( \rightarrow \)  
Mapping operator

\( a \leftarrow b \)  
Rule operator, meaning \( b \) generates \( a \)

Subscripts:

1  
Inlet

2  
Outlet

c  
Cell

cool  
Cooling channel

fc  
Face

fluid  
Fluid side

interface  
Coupling interface

m  
Average value

solid  
Solid side

wall  
Cooling channel wall
CHAPTER I
INTRODUCTION

Heat transfer processes are very important in many engineering and technological applications which involve energy transport, ranging from industrial manufacturing processes to studies of the environment. The general goal of heat transfer studies is the accurate prediction of temperature and heat flux distributions in space and possibly time, in a material and on its boundaries. The temperature field is important when one has to consider thermal stresses and material properties, which are the key elements for the optimal design of thermal structures or the development of new composite materials, for example. Then the effect of heat transfer is predicted and evaluated to meet the need to develop new systems or optimize existing ones.

1.1 A Brief Review of Heat Transfer Models

Heat transfer is generally defined as thermal energy transit due to a temperature difference\[1\]. Heat conduction and heat radiation are two basic processes by which heat transfer occurs: heat conduction occurs if a temperature gradient exists in a stationary material with no bulk motion, which may be a solid or a fluid, and the transport of energy is due to the random thermal motion of the microscopic particles that the material is composed of, which may be atoms or molecules; while heat radiation occurs between materials at different temperatures in the form of electro-magnetic waves, and the energy is emitted and transmitted between two
bodies either through vacuum, or through a medium, which could be participating in the energy exchanges.

When heat transfer occurs in a fluid that is in motion, the conductive (and, if not negligible, radiative) heat transfer process is affected by the relative motion within the fluid, and energy is also transferred by bulk or macroscopic motion of the fluid. The process of energy transport by the combined effect of heat conduction (and radiation) and the movement of fluid is referred to as convection heat transfer\[2\]. It is actually conduction (and radiation) in moving medium. Heat convection analysis is more complicated, because the motion of the fluid must be studied simultaneously with the energy transfer process.

In most practical applications, heat transfer modes are coupled either in the material or at the boundaries. For low temperature bodies, heat radiation may be negligible. But, if an absorbing and emitting gas is present, such as moisture or carbon dioxide, the heat radiation to and from the fluid must be considered. A general form of the energy differential equation can be written as follows:

\[
\frac{\partial E}{\partial t} + \text{div} \left\{ (E + p)\ddot{u} - [\ddot{u} \cdot \tilde{\tau} - \tilde{q}_{\text{effective}}] \right\} = 0, \quad (1.1)
\]

where \(\tilde{q}_{\text{effective}}\) is the effective heat flux, which can include the contribution of thermal molecular activity, thermal radiation, and flow fluid fluctuations in the turbulent case.

Similarly, in material with no bulk motion, the energy equation can be set up as:

\[
\frac{\partial E}{\partial t} + \text{div} \left\{ -\tilde{q}_{\text{effective}} \right\} = g, \quad (1.2)
\]

where \(g\) is an internal thermal source.
Particularly interesting is focused on the heat transfer that occurs between a fluid in motion and a solid at the boundary, when they are at different temperatures. The interface can be stationary, such as cool fluid flow over a heated solid surface, or possibly moving, such as ice melting caused by natural or forced convection. Then the heat conduction problem must be solved, possibly with radiation coupled in order to find the temperature in the solid part, and the convection problem must be solved, possibly with radiation coupled in order to find the temperature in the fluid part. Both solid and fluid values are dependent on the boundary conditions imposed at the boundary surface. In particular, the boundary conditions to be imposed at the surface in order to determine the heat transfer between solid and fluid may not be easily specifiable in this situation. One case in point is the thermally convective wall (heating or cooling): this implies the heat transfer coefficient or heat flux should be obtained from a solution of the coupled solid-fluid problem. Therefore, models have to be developed considering both solid and fluid regions, and the solution of the coupled problem can then be recovered.

The coupled problem is non-linear because the dependence on the governing variable (temperature or energy) is different for different models, and becomes more complicated when material properties are temperature-dependent or an irregular geometry is found. So a coupled problem is often approached numerically. Numerical modeling, because of its flexibility, can deal with irregular boundaries and complicated physical circumstances, and has received considerable attention as a tool for finding solutions to practical engineering problems.
1.2 A Brief Review of Previous Work

Several studies have been done in the area of coupled heat transfer modes. Shope[3] dealt with the cooling of a nozzle wall for a supersonic wind tunnel. DeLise and Naraghi[4] solved the compressible boundary layer equations and evaluated the convective heat transfer rates from high temperature combustion gases to the converging-diverging nozzle of a liquid-fueled rocket engine. Janus and Newman[5] coupled aerodynamic and thermal effects for an optimization study of turbine airfoil design. Sondak and Dorney[6] investigated coupled unsteady flow and heat conduction for a turbine stage. Tiwari and Pidugu[7] coupled convection and radiation heat transfer models to investigate radiation-chemistry interactions in expanding nozzle flows. Webster[8] developed a heat-conduction solver and coupled it with an existing flow solver, and he also discussed some issues related to ensuring thermal communication between solid grid blocks. Naraghi[9] developed a Rocket Thermal Evaluation code for regeneratively cooled rocket thrust chambers and nozzles: in this approach, either the CET code developed by Gordon and McBride[10, 11] (Chemical Equilibrium with Transport Properties) can be used for the evaluation of hot gas properties, or a shell program TDK[12] (Two Dimensional Kinetics Nozzle Performance Computer Program) can be incorporated (sharing I/O files) to calculate the hot-gas-side wall heat flux; then the GASP[13]/WASP[14] codes can be used to calculate the coolant flow properties. To account for high speed and high temperature difference situations, reference and adiabatic wall properties are used to determine both the coolant and hot-gas thermodynamic and transport properties, which are then used to calculate the wall friction factor and wall Nusselt number. Sibtosh and Kevin[15] investigated the heat transfer characteristics of a gaseous oxygen(GO$_2$)/gaseous hydrogen(GH$_2$) two-dimensional compact rocket thruster, especially in the nozzle region. Thermocouple were
buried deep inside the nozzle wall to measure the axial wall temperature profile. The nozzle was heavily cooled with water during the thruster firing. They also gave numerical results, whereby the FDNS CFD code[16] was used for the fluid simulation, and a one-dimensional heat transfer model based on Bartz’s correction[17, 18] was incorporated to calculate the nozzle wall heat flux and temperature using the measured temperatures inside the nozzle wall.

1.3 Goals of This Study

As seen, in the brief review above, a lot of effort have been focused on the development of individual flow fluid models or individual heat transfer models, and some attention has been paid to model coupling, but the coupling was conducted through explicit boundary conditions in a loose manner. The resulting models may have stability problems, especially for transient calculation. Thus, there is a strong need for more studies to address the issues of model coupling in various application cases.

The goal of this study is to build, test, validate, and implement two heat transfer models, and couple them to an existing fluid flow solver. The first model is for heat conduction computations, the other one is a quasi-one-dimensional cooling channel model. This study proposes to enhance the stability of the coupled model by developing special algorithm at the interface. This study proposes to couple the models in an implicit and tight approach into an integral model. Therefore, the resulting model can handle linear and non-linear, steady and unsteady problems, and can be applied to solid-fluid, and solid-fluid-cooling channel fully coupled truly multi-disciplinary problems, which this is never done before.
1.4 Multi-Disciplinary Simulations Environment: Loci

Concurrent engineering analysis typically requires a multi-disciplinary approach: fluid flow, heat transfer, elasticity, electro-magnetism, and design optimization have to be considered concurrently. Therefore, it becomes necessary to build an application framework to assemble these individual physical components into an integrated application. This assembling can be divided into two general approaches: loose coupling of a variety of applications that are specialized to single disciplines, hopefully iterating to convergence; or tight coupling in one integrated multidisciplinary application. The former approach has the advantage that the coupled codes employ appropriate numerical models that have been validated for each individual discipline, and relatively straightforward to assemble. However, the coupling can have uncertain stability properties, and may be problematic for transient problems, particularly when the characteristic time scales of the various disciplines are similar. The latter approach typically places all disciplines under one numerical method (e.g. finite-element or finite-volume). This approach has the advantages that coupling is seamless, easily incorporating transient and non-linear solvers. However, one may have conditioning problems if disciplines have widely different time-scales, unless care is taken in formulation. Also, this one-size-fits-all approach removes the possibility of using the most appropriate numerical method for each individual discipline.

The current work is based on the Loci system[19, 20], which was developed at Mississippi State University. With Loci, a third approach is possible. Each discipline can use the numerical method (finite-difference, finite-volume, or finite-element) that is best suited to its accurate simulation (as in the loosely coupled approach), while the interface between disciplines can take advantage of knowledge of the specific numerical methods (e.g., space and time integration) to develop a
coupling that remains true to the physics and numerics. The Loci framework solves the coupling problem by automatically generating the control and data movement operations of an application from component specifications, while keeping data consistent between components. A full range of possible interface treatments can be implemented: from loose coupling techniques, to domain-decomposition methods, all the way to tight non-linear coupling. The coupling or data movement can be operated at the outside level of a time iteration step, or at the inner level of a Newton iteration step. Moreover, the final code developed based on Loci is automatically parallelized, a model developer need not know the details of computer architecture (such as parallelization directives or messages passing calls). The parallel implementation can support scalability for clusters consisting of hundreds of processors.

The Loci system connects user applications through a fact database and computation rules. The fact database contains data elements such as: parameter, store, map, and constraint. Basic computation rules include: singleton rules, point-wise rules, reduction rules, and iteration rules. Some important data structures used in this study will be discussed in this work. For more details, please look at [19].

CHEM[21, 19, 22, 23], is a flow solver developed within Loci that includes complex thermodynamic, chemistry, transport, and turbulent models. A finite-volume method for three-dimensional generalized grids, Roe flux difference techniques, and explicit or implicit time integration schemes are employed. The CHEM code is also a library of reusable rules that can be dynamically reconfigured to solve a variety of problems by changing the given fact database, adding rules, or changing the query.
The successful development of CHEM has demonstrated the flexibility of Loci to deal with multi-physical models. It is now possible to couple heat transfer models to the flow solver, and obtain accurate simulation of heat transfer and temperature fields in both solid and fluid phases, and this is the focus of the present study. In particular, two heat transfer models have been coupled with CHEM using the Loci system: the first one is a solid heat conduction model, the latter is a quasi one-dimensional water cooling channel model. Also efforts were made on developing algorithms for interface entities, in order to ensure the consistency of the data at the phase boundaries. The coupling processes based on the Loci data structure and computation rules will be presented, in order to demonstrate the flexibility of this system for the seamless integration of multi-physical components.

1.5 Table of Contents

In Chapter II, CHEM flow solver is introduced, and the issues that arise from the coupling with solid heat conduction model are discussed. Governing equations for two heat transfer models and numerical schemes are presented in chapter III. Also a water steam thermodynamic model is introduced in chapter III. Chapter IV presents how these models are implemented based on the Loci framework and model coupling issue is discussed. In chapter V, several preliminary test cases of both solid heat conduction and cooling channel flow are briefly discussed. Solid heat conduction solver is verified, and solid block to block communications are discussed. Two real engineering problems are simulated in chapter VI. The first case involves a fully coupled solid-fluid problem, the second case involves fluid-solid-cooling channel three models. The calculated results for both cases are compared with available experimental data. Finally, chapter VII provides a conclusion of this study.
CHAPTER II
INTRODUCTION TO THE FLOW SOLVER (CHEM)

In this chapter, CHEM flow solver is introduced, and the issues that arise from the coupling with solid heat conduction model are discussed. For an inviscid fluid model with adiabatic assumption, there is no heat transfer term in the energy equation, so there is no base for coupling solid heat transfer model, while for viscous fluid model, heat transfer term appears in the energy equation, the fluid model incorporates the viscous models and heat transfer model. The flow solver governing equations, viscous models equations, temperature gradient, thermal boundary condition issues are discussed below.

2.1 Governing Equations

A finite volume method is applied to discretize the flow equations. After integration over a computational cell or control volume $\Omega_c$, closed by a boundary $\partial\Omega_c$, the governing equations in vector form can be written as follows:

$$
\frac{d}{dt} \int_{\Omega_c(t)} QdV + \int_{\partial\Omega_c(t)} (F_i - F_\nu) dS = \int_{\Omega_c(t)} \dot{W}dV, \quad (2.1)
$$

where the vector of conservative variables $Q$, the inviscid flux $F_i$, the viscous flux $F_\nu$, and the chemistry source term $\dot{W}$, are given by:
As noticed in the energy term of viscous flux $F_v$, the $\tilde{q}$ is denoted as the effective heat flux, in this case, radiation heat transfer is neglected, so Fourier’s Law is employed to calculate the heat conduction flux related heat conductivity and temperature gradients. The heat flux vector can be written as:

$$\tilde{q} = - (\lambda + \mu_t c_p / Pr_t) \nabla T, \quad (2.2)$$
where \((\lambda + \mu_t c_p/Pr_t)\) is the effective thermal conductivity, in which \(\lambda\) associates with the contribution of thermal molecular activity, and \(\mu_t c_p/Pr_t\) with the contribution of turbulent fluctuating motion, and the eddy viscosity \(\mu_t\) is calculated by incorporating a turbulent model (given later).

For laminar flow, it is well ordered and well layered and the fluctuations are small in magnitude compared to the mean flow. The heat transfer is largely dominated by molecular activity, and determined in terms of fluid transport property, such as viscosity \(\mu\), and thermal conductivity \(\lambda\). For turbulent flow, an eddy viscosity for momentum and an eddy diffusivity for heat transfer are introduced by analogy with laminar flows, also an hypothetical analogy between momentum and heat transfers is considered by defining the so called turbulent Prandtl number, which is generally considered as a universal constant. So, heat transfer is enhanced by changing the thermal conductivity to an effective thermal conductivity \((\lambda + \mu_t c_p/Pr_t)\).

The transport properties of the mixture are usually evaluated in two steps: first, transport properties for each species is determined; then a mixing rule is invoked in order to obtain mixture values.

Two models are applied to compute species transport properties. At temperature lower than 1000 K, Sutherland’s law is used as:

\[
t_i = T^{3/2} \frac{F_{t,i}}{T + G_{t,i}},
\]

(2.3)

where \(t_i\) stands for either \(\mu_i\) or \(\lambda_i\), and \(F_{t,i}, G_{t,i}\) are constants determined empirically. At temperature higher than 1000 K, a more accurate model based
on curve fit tabulation proposed by Gupta[24] is utilized:

\[ \mu_i = \exp(C_{\mu,i})T^{A_{\mu,i}\ln T + B_{\mu,i}}, \] (2.4)

\[ \lambda_i = \exp(E_{f,i})T^{(A_{f,i}(\ln T)^3 + B_{f,i}(\ln T)^2 + C_{f,i}\ln T + D_{f,i})}, \] (2.5)

where \( A_{\mu,i}, B_{\mu,i}, C_{\mu,i}, E_{f,i}, A_{f,i}, B_{f,i}, C_{f,i} \) and \( D_{f,i} \) are tabulated curve fit coefficients. Alternatively, 4th degree polynomial curve fit formula can be specified in place of equations (2.4) and (2.5), and represented as:

\[ t_i = A_i + B_i T + C_i T^2 + D_i T^3 + E_i T^4, \] (2.6)

where the coefficients of these curve fits can be obtained using the CHEMKIN transport library[25].

Once the transport properties for individual species are obtained, Wilke’s rule is applied to determine mixture values, as follows:

\[ t = \sum_{i=1}^{NS} W_i t_i, \] (2.7)

where \( t \) denotes transport properties for the mixture (either \( \mu \) or \( \lambda \)) and the weighting function \( W_i \) is given by

\[ W_i = \frac{X_i}{\sum_{j=1}^{NS} X_j \phi_{ij}}, \] (2.8)

where the coefficient \( \phi_{ij} \) is given by

\[ \phi_{ij} = \frac{1}{\sqrt{8}} \left( 1 + \frac{M_i}{M_j} \right)^{-1/2} \left[ 1 + \sqrt{\frac{\mu_i}{\mu_j}} \left( \frac{M_i}{M_j} \right)^{1/4} \right]^2. \] (2.9)
Thermodynamic model, chemistry model and numerical formulation, including time and spatial integration, for CHEM, are detailed in [26].

### 2.2 Turbulent Models

Several turbulence models are implemented in CHEM, including the algebraic Baldwin-Lomax[27], the one-equation Spalart-Allmaras[28], and a family two-equation models including Menter’s SST[29] model.

The Baldwin-Lomax turbulence model is an algebraic eddy viscosity, zero-equation model. The advantages of this model are its computational efficiency and robustness. This model works best in wall bounded flows with favorable pressure gradients. As the flow physics and geometry become more complicated, the performance of this turbulence model greatly decreases. The model is not reliable for separated flows.

The Spalart-Allmaras turbulence model is a one-equation model assembled using empiricism and arguments of dimensional analysis, Galilean invariance, and selective dependence on the molecular viscosity[28]. A damping function is used into this model in order to properly attenuate the turbulent viscosity in the viscous sublayer. The model is applicable to wall bounded flows, as well as free shear flows. For best results with the Spalart-Allmaras model, a very fine near-wall mesh spacing (on the order of \( y^+ \approx 1 \)) is required, where \( y^+ (y^+ = u^+_\tau y) \) is the dimensionless distance of the nearest cell to the wall, \( u_\tau \) is the friction velocity \( (u_\tau = \sqrt{\tau_{wall}/\rho}) \), \( y \) is the normal distance of the first grid point to wall.

It is well known that two-equation eddy-viscosity “low-Reynolds-number” turbulence models are among the most widely used models for engineering applications today, and the \( k \) – \( \epsilon \) model with damping functions near the wall is the most popular. However, the \( k \) – \( \epsilon \) model often suffers from numerical stability
problems due to disparate turbulent time scales. Another well-known two-equation turbulence model is the $k-\omega$ model, developed by Wilcox [30], tends to be more accurate for boundary layers with adverse pressure gradients. The $k-\omega$ model does not require damping functions in viscous sublayer and that the equations are less stiff near the wall, so it is superior to the $k-\epsilon$ model with regard to numerical stability. However, when applied to the free shear layers, it is found that there is a strong dependency of the results on the free-stream value of $\omega$[31, 29]. Menter[32] created a new model, called baseline (BSL) model, by blending the $k-\epsilon$ and Wilcox’s ’88 $k-\omega$ model. This model tries to apply the Wilcox’s ’88 model to the inner wall regions of a boundary layer and a transformed $k-\epsilon$ model for the outer boundary layer regions. In order to accurately predict adverse pressure gradient flows, especially in the wake region, Menter [32] modified the BSL model by including the transport of the principal turbulent shear stress [33] in the eddy-viscosity formulations, which leads to the shear-stress transport (SST) model. SST Model was evaluated for heat transfer applications by Tong[34].

2.3 Temperature Gradient Computation

Heat flux is a part of viscous flux, which is computed by associating with face temperature gradient. It was concerned that the computation of face gradient makes sure the coefficients of final integral stencil are positive. It is known that gradient operator associates with Laplacian operator by Gauss’ theorem. In general, a simple average of neighboring cell gradients incorrectly filters high frequency modes from the resulting Laplacian and possibly introduce non-physical negative coefficient. To correct this problem, face gradients are computed using a simple centered difference in the direction of the vector connecting cell centroid on either side of the face, while cell averaged gradients are used in orthogonal
\[ \nabla T_f = \nabla T_{avg} - (\nabla T_{avg} \cdot \hat{n}) \hat{n} + \frac{T(\tilde{r}_c) - T(\tilde{r}_f)}{\tilde{r}_c - \tilde{r}_f} \cdot \hat{n}, \quad (2.10) \]

where \( \nabla T_f \) is the face center temperature gradient, \( \nabla T_{avg} \) is the averaged cell center temperature gradient, \( \tilde{r}_c \) and \( \tilde{r}_f \) are cell center and face center positions, respectively.

A piecewise linear function associating with cell value and gradient is written as a second-order Taylor-series expansion:

\[ T(\tilde{r}_j) = T(\tilde{r}_c) + \nabla T(\tilde{r}_c) \cdot (\tilde{r}_j - \tilde{r}_c), \quad (2.11) \]

where \( \nabla T(\tilde{r}_c) \) is the cell center temperature gradient, \( \tilde{r}_c \) is the position of computed cell center, and \( \tilde{r}_j \) is the position of neighboring cell center.

Then cell-centered gradient is evaluated by utilizing an area-weighted least-squares approach from a stencil of neighboring cells. Area weighting of error allows for thin-layer stencil for viscous grid.

\[
\begin{bmatrix}
\Delta x_1 A_{f1} & \Delta y_1 A_{f1} & \Delta z_1 A_{f1} \\
\Delta x_2 A_{f2} & \Delta y_2 A_{f2} & \Delta z_2 A_{f2} \\
\vdots & \vdots & \vdots \\
\Delta x_N A_{fN} & \Delta y_N A_{fN} & \Delta z_N A_{fN}
\end{bmatrix}
\begin{bmatrix}
\frac{dT}{dx} \\
\frac{dT}{dy} \\
\frac{dT}{dz}
\end{bmatrix}
= \begin{bmatrix}
(T(\tilde{r}_1) - T(\tilde{r}_c)) A_{f1} \\
(T(\tilde{r}_2) - T(\tilde{r}_c)) A_{f2} \\
\vdots \\
(T(\tilde{r}_N) - T(\tilde{r}_c)) A_{fN}
\end{bmatrix},
\]

(2.12)

where \( A_{fj} \) is the area of the common face shared by the adjoining cell \( j \). The over-determined system (2.12) is solved using the modified Gram-Schmidt QR factorization method, which is stable on highly skewed viscous meshes.
2.4 Thermal Boundary Condition

For non-slip thermally adiabatic solid wall, heat flux is set to zero, wall temperature is approximated as the close cell center temperature. For a temperature specified solid wall, a molecular characterized heat flux is computed, because the eddy viscosity is defined as zero there. Other variables used are detailed in [26]

\[ \bar{q} = -\lambda \nabla T, \quad (2.13) \]

where \( \lambda \) is fluid thermal conductivity evaluated using equations introduced in section 2.1, based on the temperature specified. For coupling case, interface parameters are calculated using parameters of both fluid and solid sides, this will be detailed in section 3.5.
CHAPTER III
HEAT TRANSFER MODELS

Governing equations for two heat transfer models and numerical schemes are discussed in this chapter. Both solid-phase and cooling channel model developments are presented.

The actual goal of heat transfer study is to find the temperature field and heat fluxes in a material, given a set of Partial Differential Equation (PDE), and boundary conditions, initial conditions and distribution of thermal sources. The set of mathematical equations (governing equation) is an abstraction of reality, which retains only key features of physical process.

3.1 Governing Equations for Solid Heat Conduction

The basic governing equation for the solid heat conduction model is obtained by applying the principle of conservation of energy to a control volume: in integral form it is given as:

\[
\frac{d}{dt} \int_{\Omega_c(t)} EdV + \int_{\partial\Omega_c(t)} qdS = \int_{\Omega_c(t)} gdV, \tag{3.1}
\]

where \( g \) is the thermal source per unit volume per unit time. The left side is the rate of energy increase in the control volume, the first term of the right side is the total heat conducted through the control surface, the second term is the heat generated or consumed within the control volume. It is also assumed that the energy associated with volume change is negligible.
A relationship between temperature and total energy is stated as:

\[ E = \rho \int_{T_{ref}}^{T} C_p(t) dt, \]  

(3.2)

The thermal properties, such as heat capacity can be temperature dependent.

Fourier’s law of conduction is applied to compute local heat flux. For isotropic material, in which thermal conductivity is the same in all directions, the conductive heat flux is written as:

\[ q = -k \nabla T, \]  

(3.3)

The thermal conductivity can also depend on the temperature and on the location. For example, the thermal conductivity and heat capacity of oxygen-free high-conductivity (OFHC) copper change from 482 (W/m.K), 252 (J/kg.K) at 100 K to 339 (W/m.K), 480 (J/kg.K) at 1200 K, respectively. In this study, a 4th degree polynomial curve fit formula is employed as:

\[ k = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4, \]  

(3.4)

where the coefficients of the curve are obtained by fitting available data in [35, 36].

3.2 Governing Equations for Cooling Channel Flow

The cooling channel flow governing equations are developed by using conservation laws for mass, momentum, and energy. The main effects of heat transfer and channel viscous friction are taken into account, and, assuming that flow parameters change in the flow direction (stream-wise), transverse variations are ignored. It is possible to have phase change in the water: a mixture model is applied, resulting in a single-fluid approach. The assumption is that the two
phases are inter-penetrating and moving at the same velocity, and vapor quality is limited to a small value. As shown in figure 3.1, taking a general one-dimensional control space with volume $dV$, inlet area $A_1$, outlet area $A_2$, and wall area $A_{wall}$, taking $Q$ as the vector of conservative variables, $F$ as the fluxes through the flow-area $A_1$ and $A_2$, $S$ as the vector of source-terms, i.e., additional fluxes of properties coming through the walls, also assuming no mass diffusion, no reaction, then the governing equation is written as:

$$ \frac{d}{dt} \int Q dV + F_2 A_2 - F_1 A_1 = S $$

Figure 3.1: Control Space for Cooling Channel

where the variables $Q$, $F$ and $S$ are listed as:

$$ Q = \begin{bmatrix} \rho \\ \rho V \\ \rho e_0 \end{bmatrix} , \quad F = \begin{bmatrix} \rho V \\ \rho V^2 + p \\ (\rho e_0 + p)V \end{bmatrix} , \quad S = \begin{bmatrix} 0 \\ f_{wall} + f_{body} \\ \Delta Q \end{bmatrix} , \quad (3.6) $$
and \( V \) is velocity, \( e_0 = U + \frac{1}{2}V^2 + gz, \) \( f_{body} = -\rho Agdz, \) \( g \) is gravity constant, \( \Delta Q \) is the net heat transferred through wall. Coolant thermodynamic properties are computed in next section,

The viscous forces is assumed in the direction of the flow and given by

\[
f_{wall} = -C_f \frac{1}{2} \rho V^2 A_{wall} \tag{3.7}
\]

where the friction coefficient \( C_f \) associates with friction factor \( f \) as

\[
C_f = \frac{f}{4} \tag{3.8}
\]

And the wall area \( A_{wall} \) is computed as:

\[
A_{wall} = W_p ds \tag{3.9}
\]

where wet perimeter \( W_p \) is computed relating to channel diameter \( D \) or the hydraulic diameter (for non-circular cross-section tubes) as:

\[
\frac{D}{4} = \frac{A}{W_p} \tag{3.10}
\]

The steady quasi-one-dimensional cooling channel flow governing equations in integral form can be derived as follows:

\[
\begin{bmatrix}
\rho_1 V_1 A_1 = \rho_2 V_2 A_2, \\
\frac{\rho_1 V_1^2}{\rho_1} + \frac{V_1^2}{2} + g z_1 = \frac{\rho_2 V_2^2}{\rho_2} + \frac{V_2^2}{2} + g z_2 + \int_1^2 \frac{V^2}{2} f_D ds, \\
\Delta Q + h_1 + \frac{V_1^2}{2} + g z_1 = h_2 + \frac{V_2^2}{2} + g z_2,
\end{bmatrix} \tag{3.11}
\]
3.3 Water Steam Thermodynamic Model

For both liquid and gas phases, the thermodynamic parameters are calculated based on the standard released by The International Association for Properties of Water and Steam (IAPWS) [37]. The formulation is based on the Helmholtz function [14] and its partial derivatives. The Helmholtz function is defined as:

\[ A = U - TS, \]  
\[ (3.12) \]

\[ dA = dU - TdS - SdT, \]  
\[ (3.13) \]

Combining the first and the second laws of thermodynamics, one can write:

\[ dU = TdS - pdv, \]  
\[ (3.14) \]

Then equation (3.13) can be combined with equation (3.14), yielding

\[ dA = -SdT - pdv, \]  
\[ (3.15) \]

Introducing the differential form of the function \( A = A(T, v) \), one can write

\[ dA = \left( \frac{\partial A}{\partial T} \right)_v dT + \left( \frac{\partial A}{\partial v} \right)_T dv, \]  
\[ (3.16) \]

Comparing equation (3.15) and equation (3.16) yields

\[ S = - \left( \frac{\partial A}{\partial T} \right)_v, \quad p = - \left( \frac{\partial A}{\partial v} \right)_T. \]  
\[ (3.17) \]
If density rather than specific volume is used then the above result becomes

\[ p = \rho^2 \left( \frac{\partial A}{\partial \rho} \right)_T, \quad S = \left( \frac{\partial A}{\partial T} \right)_\rho. \quad (3.18) \]

If pressure and temperature are known, then density can be found from the equation \( p = p(\rho, T) \).

Specific internal energy and specific enthalpy can be recovered from the above relationships, and read:

\[ U = A + T S, \quad h = U + \frac{p}{\rho}. \quad (3.19) \]

Moreover, specific heats can be defined at this point, as follows:

\[ C_v = \left( \frac{\partial U}{\partial T} \right)_\rho, \quad C_p = \left( \frac{\partial h}{\partial T} \right)_p. \quad (3.20) \]

Additionally, interpolating equations[38, 39] are recommended by IAPWS to calculate viscosity and thermal conductivity for both liquid and gas phases.

For phase changes or saturation conditions, the average values of liquid and gas phases are calculated as the mixture properties:

\[ (\bullet)_{\text{mixture}} = w \times (\bullet)_{\text{vapor}} + (1 - w) \times (\bullet)_{\text{liquid}}, \quad (3.21) \]

Where the vapor quality \( w \) is defined as

\[ w = \frac{\text{mass of vapor}}{\text{mass of liquid} + \text{mass of vapor}}. \quad (3.22) \]
Some thermodynamic and transport properties of water are calculated and listed in table 3.2 and 3.1. It shows good agreement with exist IAPWS database.

<table>
<thead>
<tr>
<th>Table 3.1: Thermodynamic Property Values in Two-Phase Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T(K)</strong></td>
</tr>
<tr>
<td><strong>P(Kpa)</strong></td>
</tr>
<tr>
<td><strong>IAPWS</strong></td>
</tr>
<tr>
<td><strong>ρ_l (kg / m^3)</strong></td>
</tr>
<tr>
<td><strong>ρ_v (kg / m^3)</strong></td>
</tr>
<tr>
<td><strong>h_l (kJ / kg)</strong></td>
</tr>
<tr>
<td><strong>h_v (kJ / kg)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3.2: Water Viscosity and Conductivity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T(K)</strong></td>
</tr>
<tr>
<td><strong>P(KPa)</strong></td>
</tr>
<tr>
<td><strong>IAPWS</strong></td>
</tr>
<tr>
<td><strong>µ (µPa.s)</strong></td>
</tr>
<tr>
<td><strong>k (mWK^{-1}m^{-1})</strong></td>
</tr>
</tbody>
</table>

### 3.4 Numerical Scheme for Solid Heat Conduction

There are three basic numerical approaches for heat transfer problem: the finite-difference, finite-volume and finite-element methods. In finite-difference method, the computational domain is discretized in a regular mesh, and derivatives in the PDE are approximated by finite differences in terms of Taylor series based on nodal points values and the distances with each other. This procedure results into a set of algebraic equations, one for each node. Nodes at boundaries are considered for given boundary conditions, and special equations are obtained. Then the whole
system equation is solved numerically to obtain the variables at various nodes in
the computational domain, as an initial condition is given.

In finite-volume method, the computational domain is divided into a finite
number of regular or irregular cell volumes, an integral operation based on the
conservation law of physical concept is applied to each cell. Face gradient or
flux is computed from left and right cell variables. Similarly, this procedure results
into a set of algebraic equations about volume-averaged variables, one for each cell.
Boundary conditions are applied noninvasively. It does not require a structured
mesh (although a structured mesh can also be used) and can handle irregular
boundaries and complicated conditions.

In finite-element method, the computational domain is divided into a finite
number of regular or irregular cell volumes. The integral statements of the
governing conservation postulates result into the integral equations that apply for
each cell. Minimization of the integrals is carried out to satisfy the conservation
principles. Matrix equations for an individual cell are formulated, then matrix
equations for the overall system are assembled and solved. This method is suitable
for irregular boundaries.

As already mentioned, the two heat transfer models introduced in the above
are implemented within the Loci system, and coupled with the CHEM flow solver.
Loci allows each physical model to be simulated by a numerical method that is best
suited for accuracy and robustness of the overall procedure. In this study, a finite-
volume method is employed for the solid heat conduction model, and a steady-state
integral method is employed for the cooling channel model. Incidentally, the flow
solver CHEM is an application template built on Loci, and some of its parts can be
reused to develop new models. On the other hand, new rules are required for the
interface between different physical models. Loci is designed to generate control
and data movement operations automatically, in order to facilitate the coupling of multi-disciplinary models. In the following, numerical schemes for the solid heat conduction and cooling channel models are given, and the coupling process with the flow solver is presented.

Recalling equation 3.1, written for a small control volume, the volume integrals can be approximated as:

\[ \int_{\Omega_c(t)} E dV = E_c(t) \int_{\Omega_c(t)} dV = E_c(t) V_c(t), \]  

(3.23)

where \( E_c(t) \) is the value of \( E \) at the cell centroid, \( V_c(t) \) is the control volume, and subscript \( c \) represents a generic cell. A similar result applies to the thermal source integral.

The numerical surface integral is discretized by summing the heat flux of each face of the cell. Here a generalized grid can be used (generalized grids are discretizations composed of arbitrary polyhedra, including tetrahedra, prisms, pyramids, and hexahedra). The face temperature gradient is mapped from the cell center temperature gradient, which is constructed by a linear least-squares fit method. In summary, the numerical surface integral is approximated as:

\[ \int_{\partial \Omega_c(t)} q dS = \sum_{f_c=1}^{m} \int_{\partial \Omega_{c,f_c(t)}} q_{f_c} dS \approx \sum_{f_c=1}^{m} A_{f_c(t)} q_{f_c}, \]  

(3.24)

where \( q_{f_c} \) is the face heat flux, \( A_{f_c} \) is the area of the face, subscripts \( f_c \) stands for face, and \( m \) is the number of faces for the given cell.

At this point, equation (3.1) can be numerically approximated by the equation

\[ \frac{d}{dt} [V_c(t) E_c(t)] + \sum_{f_c=1}^{m} A_{f_c(t)} q_{f_c} = V_c(t) g_c(t). \]  

(3.25)
If the mesh is fixed, then an ordinary differential equation is derivated, which reads:

\[
\frac{dE_c(t)}{dt} = R[E_c(t), t] = g_c(t) - \frac{1}{V_c(t)} \sum_{fc=1}^{m} A_{fc} q_{fc},
\]

(3.26)

Equation (3.26) is satisfied simultaneously for all cells as the time changes. Therefore, a global system of ordinary differential equation results (the subscript \(c\) will be removed at this juncture), given by:

\[
\frac{dE(t)}{dt} = R[E(t), t],
\]

(3.27)

Where the vector \(E(t)\) represents the values of all cells at time level \(t\).

The implicit time integration scheme employees a two-parameter family of algorithms for equation (3.27), and is given as follows

\[
\frac{(1 + \psi) \Delta E^n - \psi \Delta E^{n-1}}{\Delta t} = (1 - \theta) R^n(E^n) + \theta R^{n+1}(E^{n+1}),
\]

(3.28)

where \(R\) is the residual term, \(n\) stands for the current time level, and \(\Delta E^n = E^{n+1} - E^n\). In this scheme, \(\varphi\) and \(\theta\) form a two-parameter family of algorithms. For example, setting \(\theta = 1, \varphi = 0\) gives the implicit backward Euler scheme typically used for steady state simulations, while a second order three point backward scheme \((\theta = 1, \varphi = 1/2)\) is used for time-accurate simulations.

Equation (3.28) is a non-linear system of equations for variable \(E^{n+1}\), and can be solved by Newton iterative methods, as follows:

\[
L'(E^{n+1,p})(E^{n+1,p+1} - E^{n+1,p}) = -L(E^{n+1,p}),
\]

(3.29)
where
\[ L(E^{n+1}) = E^{n+1} - E^n - \frac{\Delta t}{(1 + \varphi)}[(1 - \theta)R^n + \theta R^{n+1}(E^{n+1})] - \frac{\varphi}{1 + \varphi}(E^n - E^{n-1}), \]
(3.30)

In the above, the Newton iteration is initialized using the previous time step value \((E^{n+1,p}=0 = E^n)\), and the Jacobian is given as
\[ L'(E^{n+1,p}) = I - \frac{\Delta t \theta}{1 + \varphi} \frac{\partial}{\partial E} [R^{n+1}(E)] = I - \frac{\Delta t \theta}{1 + \varphi} \frac{\partial q}{\partial E} - \frac{1}{V} \sum_{f_c=1}^{m} A_{f_c}^{n+1} \frac{\partial q_{f_c}(E)}{\partial E}, \]
(3.31)

Equation (3.29) is solved using a Gauss-Seidel iteration method.

If the solid solver is used independently, either a temperature or heat density boundary conditions can be specified. For coupled case, the interface conditions are determined by both solid and fluid part. This will be discussed in the next section.

### 3.5 Coupling CHEM with the Solid Heat Conduction Model

The flow solver, CHEM, also employs a finite-volume numerical scheme, and has the same three iteration levels (time step iteration, Newton iteration, and Gauss-Seidel iteration). Consequently, the solid heat conduction model is fully coupled with the CHEM flow solver into a seamless application, within the Loci framework. Here fully coupled means that the two models are coupled at the Newton iteration level of the time integrator, as opposed to loosely coupled stand-alone codes. Due to this tight coupling, this new model is appropriate for time-accurate problems. For unsteady, time-accurate problems, the same time step is used for flow and solid parts, while for steady state problems, different time steps can be used for each model.
Solid and flow solvers are coupled by keeping the heat flux term consistent at the interface between fluid and solid components. The fluid part gets the interface temperature from the solid part and calculates the heat flux term, while the solid part gets the interface heat flux and calculates the temperature in return. The interface values are computed as satisfying the relationship:

\[ q_{\text{solid}} \rightleftharpoons q_{\text{interface}} \rightleftharpoons q_{\text{fluid}} \rightleftharpoons T_{\text{interface}}, \quad (3.32) \]

Because eddy viscosity is defined as zero on no-slip viscous wall, only a molecular characterized heat flux is computed. For example, Sutherland law can be applied to calculate the air thermal conductivity.

\[ \lambda = 2.495 \times 10^{-3} \left( \frac{T_{\text{interface}}^{1.5}}{T_{\text{interface}} + 194.0} \right), \quad (3.33) \]

more accurate values can be obtained using the CHEMKIN transport library[25], as presented in chapter II, then the fluid heat flux is given as

\[ q_{\text{fluid}} = -\lambda \nabla T. \quad (3.34) \]

At the solid side, the interface heat flux from fluid is directly used for solid for keeping the heat flux consistency. The interface temperature is calculated from the following relationship:

\[ q_{\text{solid}} = -q_{\text{fluid}}, \quad (3.35) \]

\[ q_{\text{solid}} = -k_{\text{solid}} \frac{T_{\text{solid}} - T_{\text{interface}}}{|\vec{r}_{\text{solid}} - \vec{r}_{\text{interface}}|} \hat{n}_{\text{interface}}, \quad (3.36) \]

\[ q_{\text{fluid}} = -\lambda \frac{T_{\text{interface}} - T_{\text{fluid}}}{|\vec{r}_{\text{interface}} - \vec{r}_{\text{fluid}}|} \hat{n}_{\text{interface}}, \quad (3.37) \]
It is reminded that this is not an explicit boundary. These parameters are evaluated not in the outer time iteration step but in a deeper Newton iteration step, to make the fluid and solid solvers tightly coupled and working in an integral mode.

### 3.6 Numerical Scheme for Cooling Channel Flow

The integral form of the system equations is applied for the quasi-one-dimensional cooling channel model, and the steady state values at the outlet of cooling channel segments (cells) are computed. Pressure and specific enthalpy are selected as independent variables. Pressure loss from inlet to outlet of each cooling cell results from viscous friction, velocity changes, and body force (gravity), and is given as follows:

\[
dp^{n+1,i} = \frac{Lf}{2} \left[ \frac{\rho_{1}^{n} + \rho_{2}^{n+1,i}}{2} \right] \left[ \frac{V_{1}^{n} + V_{2}^{n+1,i}}{2} \right]^{2} \left( \frac{2}{D_{1} + D_{2}} \right) + \left[ \frac{2\dot{m}}{A_{1} + A_{2}} \right] \left[ \frac{\dot{m}}{(\rho^{n+1,i}A)^{2}} - \frac{\dot{m}}{(\rho^{n}A)^{2}} \right] + \left[ \frac{\rho_{1}^{n} + \rho_{2}^{n+1,i}}{2} \right] g dz, \tag{3.38}
\]

\[
\dot{m} = \left[ (\rho VA)_{2} \right]^{n+1,i}, \tag{3.39}
\]

Consequently, the outlet pressure is given as:

\[
p_{2}^{n+1,i} = p_{1}^{n} - dp^{n+1,i} \tag{3.40}
\]

The outlet enthalpy is computed as follows:

\[
h_{2}^{n+1,i} = \Delta Q^{n+1,i} + h_{1}^{n} + \frac{1}{2} [V_{1}^{n}]^{2} - \frac{1}{2} [V_{2}^{n+1,i}]^{2} + g z_{1} - g z_{2}, \tag{3.41}
\]
and the heat gain from solid part is computed as follows:

\[ \Delta Q = \frac{q_{wall}A_{wall}}{m} \]  

(3.42)

where, wall heat flux \( q_{wall} \) is calculate as:

\[ q_{wall} = h_c n_c (T_{wall} - T_{n+1,i}^m) \]  

(3.43)

where \( T_m \) is mean flow temperature, defined as the average of inlet and outlet values:

\[ T_{n+1,i}^m = \frac{T_1^n + T_2^{n+1,i}}{2} \]  

(3.44)

and the local heat transfer coefficient \( h_c \) is calculated as:

\[ h_c^n = \frac{N u^n k^n}{D} \]  

(3.45)

Where, \( Nu \) is Nusselt number, \( k \) is water thermal conductivity, and \( T_{wall} \) is the cooling channel wall temperature. These values are computed iteratively, \( n \) and \( n + 1 \) mean time step, \((n + 1, i)\) means \( i \)th iteration in time step \( n + 1 \).

Equation (3.40) and (3.41) can be solved iteratively, after an initial value \((i = 0)\) are given for pressure and specific enthalpy,

\[ p_{2i}^{n+1,i=0} = p_{2i}^n, \]

\[ h_{2i}^{n+1,i=0} = h_{2i}^n, \]  

(3.46)

Because the solid phase temperatures change at each Newton iterative level, the computations for the cooling channel should be repeated at each Newton iteration. However, only a steady state cooling channel model was implemented,
therefore time accuracy was not a consideration. In this case, the cooling channel computations were scheduled once per (pseudo)-time step.

These formula are developed in a general sense, in practice, a wide variety of correlations are developed for particular cases to calculate the friction factor and Nusselt number.

When a viscous fluid, such as water, flow in a pipe, a boundary layer will form along the pipe. Gradually, the boundary layer fills the entire pipe and the flow is then fully-developed. If the pipe wall is heated or cooled, then a thermal boundary layer will also develop along the pipe. At a certain point downstream, a fully-developed thermal boundary layer is built. If the heating or cooling starts from the inlet of the pipe, then both velocity and temperature boundary layers are developing simultaneously. The rate of development of velocity and temperature in the entrance region depends on the fluid Prandtl number.

Convection problem in pipe can be approached analytically, in some special cases, such as, constant wall heat flux, constant wall temperature. But, it is not always possible to approach forced convection problems analytically, especially when the flow is turbulent. It is more complicated for the cases in which the difference between the fluid mean and wall temperature is high, together with the fluid properties variation is important to be accounted for. So, in this study, empirical correlation methods are used to calculate the friction factor and Nusselt number, which the former associates with pressure drop, while the later with heat transfer. Also, both velocity and thermal boundary layer are assumed being fully-developed.

For fully developed laminar circular pipe flow, the friction factor is independent of relative roughness, but Reynolds number, also a parabolic velocity profile is
assumed, then the friction factor and Nusselt number are calculated [40] by Deissler:

\[ f = \frac{64}{Re} \left( \frac{\mu_m}{\mu_{wall}} \right)^{-0.58}, \]  

(3.47)

\[ Nu = 4.364 \left( \frac{\mu_m}{\mu_{wall}} \right)^{0.14}, \]  

(3.48)

for \( Re \leq 2300 \)

where \( \mu_m \) is dynamic viscosity at the mean temperature, \( \mu_{wall} \) is dynamic viscosity at wall temperature, this factor accounts for the water properties variation with temperature. Reynolds number is calculated as:

\[ Re = \frac{\rho V D}{\mu_m}, \]  

(3.49)

Deissler [41] gave a relationship used to calculate \( \mu \) ratio from \( T \) ratio:

\[ \frac{\mu_m}{\mu_{wall}} = \left( \frac{T_m}{T_{wall}} \right)^{-1.6}, \]  

(3.50)

For fully developed turbulent pipe flow, the friction factor is a function of Reynolds number and the relative roughness. Blasius [2] gave a correlation between the friction factor and the Reynolds number for turbulent flow in smooth pipe for values of Reynolds number up to \( 10^5 \). For high values of Reynolds number, beyond \( 10^5 \), von Karman [42] gave a relationship. Also, a simple expression between the Reynolds number and the friction factor is given by Sadik [2] as:

\[ f = 0.184 Re^{-0.2}, \]  

(3.51)
for $3 \times 10^4 \leq Re \leq 10^6$, and $\frac{\epsilon}{D} \leq 10^{-6}$

But commercial pipes are quite rough, Moody[43] chart gave a relationship between Reynolds number, relative roughness and friction factor. In this study, Colebrook equation[44] is used to calculate the friction factor. Chen[45] approximated it by an explicit formula:

$$\frac{1}{\sqrt{f}} = -2.0 \log \left\{ \frac{e}{3.7065D} - \frac{5.0452}{Re} \log \left[ \frac{1}{2.8257} \left( \frac{e}{D} \right)^{1.1098} + \frac{5.8506}{Re^{0.8981}} \right] \right\}, \quad (3.52)$$

for $2300 \leq Re \leq 10^8$, and $\frac{\epsilon}{D} \leq 0.05$

If relative roughness is given smaller than $10^{-6}$, a smooth wall friction factor is obtained. If properties variation with temperature is considered, equation(3.51) and (3.52) are multiplied with a factor of $\frac{1}{6}(7 - \frac{\mu_m}{\mu_{wall}})$ [46].

Dittus-Boultet[47] formula is employed to calculate the Nusselt number for fully-developed turbulent flow:

$$Nu = 0.023 Re^{0.8} Pr^{0.4}, \quad (3.53)$$

$$Pr = \frac{C_p \mu}{k}. \quad (3.54)$$

In the above, Pr is Prandtl number. If properties variation with temperature is considered, equation(3.53) is multiplied with a factor of $(\frac{\mu_m}{\mu_{wall}})^{0.11}$ [46].

In practice, viscous friction and heat transfer in channel are affected by channel cross-section geometry, curvature, edge effect, twisted effect, entrance effect, roughness, and coolant type used. So many correlation or empirical formula are proposed for different situations. A comparison between calculated results and experiments should be taken for actual design.
3.7 Coupling Solid Heat Conduction and Cooling Channel Flow

In particular case, cooling channel transverse through a heated solid block with an non-uniform temperature distribution, maybe time-dependently, illustrated in figure 3.2.

![Figure 3.2: Cooling Channel in Solid Block](image)

For a circular cooling channel, a relationship between cooling channel and solid part is developed based on the energy conservation law, and is given as:

\[
2\pi r_{\text{solid}} q_{\text{solid}} = 2\pi r_{\text{cool}} q_{\text{wall}},
\]

\[
r_{\text{solid}} k_{\text{solid}} \frac{dT}{dr} = r_{\text{cool}} h_c (T_{\text{wall}} - T_m),
\]

\[
dT = \frac{r_{\text{cool}} h_c (T_{\text{wall}} - T_m)}{r_{\text{solid}} k_{\text{solid}}} dr,
\]

where \( q_{\text{solid}} \) is the heat flux in solid part, \( q_{\text{wall}} \) is the heat flux at cooling channel wall, \( r_{\text{solid}} \) is the distance between the center of a solid cell and the cooling cell, \( r_{\text{cool}} \) is the cooling channel radius, \( T_m \) is the mean temperature of the cooling cell, and \( k_{\text{solid}} \) is solid thermal conductivity.
Integrating equation (3.57) to get the cooling channel temperature results in
the following:

\[ T_{wall} = \frac{r_{cool} h_c T_m (\ln r_{solid} - \ln r_{cool}) + T_{solid}}{r_{cool} h_c (\ln r_{solid} - \ln r_{cool}) + 1}, \]  

(3.58)

where \( T_{solid} \) is the solid cell temperature.

The average cooling channel wall temperature is computed as:

\[ \bar{T}_{wall} = \frac{\int T_{wall} dl}{\int dl} \]  

(3.59)

For the purpose of evaluating equation (3.59), a stencil of solid cells is constructed
around the cooling cell, excluding the solid cells that contain the cooling cell.
In chapter III, two heat transfer models are developed. This chapter will demonstrate how these models are constructed based on Loci framework and how the interfaces between models are specified. The geometry mesh, boundary, and initial conditions are described using the Loci data types and statements. The numerical methods are represented as a set of computation rules, which are a set of classes to implement the algorithms.

4.1 Introduction to Loci

Loci framework was written in C++ language. Besides inheriting general data type from C++ language, Loci defines some more powerful data types for numerical computation. They are entity, entitySet, constraint, sequence, store, storeVec, storeMat, parameter, map, multiMap.

The most fundamental concept in the Loci framework is the concept of an entity. Entities are conceptually places to store values. In Loci, these entities are given integer identifiers. An entity can be considered to be an interval and an entitySet, which is a collection of intervals. The entitySet are used for control and allocation. For example, grid related elements of nodes, faces, cells are defined as entitySet. The constraint specifies a set of entities. Consider to specify a boundary condition to a set of boundary faces, but not the interior faces. The constraint is to constrain a rule so that it can only apply to a subset of entities. For example, constraint("interior_faces") defines the rule operation just taking place for interior
faces. The sequence gives a particular ordering for the entitySet. Suppose there is a function called ‘calculate’ defined over the Entity ‘cc’, in the class ‘compute cell volume’ described as following:

```cpp
class compute cell volume : public pointwise rule {
    void calculate(Entity cc) ;
    virtual void compute(const sequence &seq) {
        do_loop(seq, this, &compute cell volume::calculate) }
}
```

This will calculates the volume for all entities in this sequence ‘seq’. And this sequence can be constrained by constraint(“cells”).

The store is a container that associates values to entities given in an entitySet. The store provides an injective mapping from entities to values one by one. For example, face centers are defined by a store over the entitySet faces. StoreVec and storeMat are defined to allocate vectors and square matrix. For example, in fluid model, primitive vector of species density, x, y, z velocities, total energy is defined as storeVec, and diagonal, lower and upper matrix of Gauss-Seidel algorithm are defined as storeMat. The parameter maps many entities to a single value, for example, initial temperature is defined as a parameter for all solid cells.

The map and mutiMap are used to specified relationships between entities. Maps provide a one to one relationship, for example, a map is defined for a face to map the right cell, while multiMap provides a one to multiple relationship. For example, a multiMap is defined for a face to map to nodes that construct the given face.

Computation rules are a set of classes to implement the numerical algorithm. They are divided into point-wise rules, singleton rules, unit-apply rules, and iteration rules. The point-wise rule provides a set of values for every entity in
the rule context and produces store type variables, while singleton rule computes a single value for every entity and produces parameter type variables. For example, a rule to compute store type value of face center is defined as point-wise rule, but a parameter type value of initial temperature can be specified in a singleton rule.

A variable is defined either by point-wise rule or unit-apply rule, but not both. For example, the maximum error can be found through a series of comparison computations, source terms can be reduced from the integral of cell based thermal source and face based heat flux source (possibly, a kind of map from cell to face will be used). Unit-apply rule provides such abilities. In unit rule, an initial value is specified for a set of entities, then each apply rules are applied to the entities, and the final value of the entity is uniquely specified.

The most important rule type is iteration rule. According to the time integration scheme employed in equation (3.28), variables in two or three time levels of (n-1, n, n+1) are maintained, which dependent on the two parameters of $\varphi$ and $\theta$. As time step promoting, these variables are automatically updated to new values. If implicit time scheme is used, Newton iteration level is the second sub-level iteration, and the Gauss-Seidel iteration is the inner iteration. The promotion processes are described as following:

- Initialization of Newton iteration: $(n, it = 0) \leftarrow (n)$
- Initialization of Gauss-Seidel iteration: $(n, it, igs = 0) \leftarrow (n, it)$
- Update in Gauss-Seidel Iteration: $(n, it, igs) \leftarrow (n, it, igs + 1)$
- From GS promotion to Newton: $(n, it + 1) \leftarrow (n, it, igs)$
- Update in Newton Iteration: $(n, it) \leftarrow (n, it + 1)$
- $(n + 1) \leftarrow (n, it)$ from Newton promotion to n+1
- $(n) \leftarrow (n + 1)$ update to new time level
where, \( n, it, igs \) are stand for time level, Newton iteration level, and Gauss-Seidel iteration level, respectively. \((+1)\) stands for new level of its iteration. Iteration rule is implemented by beginning with building rule to initialize the values, and advance rule to update the values, and collapse rule to stop the iteration and prompt to higher level of iteration.

### 4.2 Solid Block Mesh Representation

The mesh is a discretization of space, which consists of a collections of points or nodes that have spatial positions distributed over the region of interest. An unstructured generalized grid is employed in this study, which the face and cell can has arbitrary shapes. These nodes, edges, faces and cells are called entities. These entities are defined with respect to their relationship to one other. An edge is defined by two points in the mesh, while a face is defined by a set of edges, and a cell is defined by a set of faces. The links between these entities are constructed by maps, which provide access from a face to the nodes that constructed the face, from a cell to the faces and nodes belong to the cell, from a face to the left and right cells.

The numerical algorithm developed in chapter III reduces to a set of volume and surface integrals, where volume integrals are computed for each cell based on cell values, and surface integrals are computed based on extrapolated values for the left and right cells. The area and face center computations of a face require access to nodal positions, so a map \( h_{-}face2node \) (from face to node) is defined, and the nodes are ordered in the map such that the face normal vector points in the direction of the right side of the face, or points outside of a boundary face. The prefix \( h_{-} \) represents for solid heat transfer contrast to fluid model. Face values are extrapolated from the left and right cell’s values, so the maps to a face’s left
and right cells are defined by map $h_{cl}, h_{cr}$, for a boundary face, the map $h_{ci}$ is defined to access the boundary cells. Cells based computations require access to the faces of a given cell, so the map $h_{cell2face}$ is defined, specially, the map $h_{lower}$ accesses to the faces, whose $h_{cr}$ map corresponds to the given cell, while the map $h_{upper}$ accesses to the faces, whose $h_{cl}$ map corresponds to the given cell, and the map $h_{boundary\_map}$ accesses to the faces, whose $h_{ci}$ map corresponds to the given cell, and they work as the inverse maps. Table 4.2 shows all these maps descriptions. In addition to these maps, rules for the computation of face areas, face center, cell volumes, cell center are required. These grid-related computations were listed in equations from (4.1) to (4.4), where the rule operator $a \leftarrow b$ meaning $b$ generates $a$, and the mapping operator $\rightarrow$ defining the sources.

\[ h_{facecenter} \leftarrow [h_{face2node} \rightarrow h_{pos}], \quad (4.1) \]

\[ h_{area} \leftarrow [(h_{face2node} \rightarrow h_{pos}), h_{facecenter}], \quad (4.2) \]

\[ h_{cellcenter} \leftarrow \begin{pmatrix} (h_{upper}, h_{lower}) \rightarrow (h_{facecenter}, h_{area}) \\ h_{boundary\_map} \rightarrow (h_{facecenter}, h_{area}) \end{pmatrix}, \quad (4.3) \]

\[ h_{vol} \leftarrow \begin{pmatrix} (h_{upper}, h_{lower}) \rightarrow (h_{facecenter}, h_{area}) \\ h_{boundary\_map} \rightarrow (h_{facecenter}, h_{area}) \\ h_{cellcenter} \end{pmatrix}, \quad (4.4) \]
Table 4.1: Solid Block Mesh Descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Value Type</th>
<th>Function</th>
<th>EntitySet</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{pos}$</td>
<td>vector 3-D</td>
<td>node position</td>
<td>$h_{nodes}$</td>
</tr>
<tr>
<td>$h_{face2node}$</td>
<td>multi-map</td>
<td>map to the nodes of the given face</td>
<td>$h_{faces}$</td>
</tr>
<tr>
<td>$h_{cr}$</td>
<td>map</td>
<td>map to the right cell</td>
<td>$h_{interior_faces}$</td>
</tr>
<tr>
<td>$h_{cl}$</td>
<td>map</td>
<td>map to the left cell</td>
<td>$h_{interior_faces}$</td>
</tr>
<tr>
<td>$h_{ci}$</td>
<td>map</td>
<td>map to the boundary cell</td>
<td>$h_{boundary_faces}$</td>
</tr>
<tr>
<td>$h_{cell2face}$</td>
<td>multi-map</td>
<td>map to the faces of the given cell</td>
<td>$h_{cells}$</td>
</tr>
<tr>
<td>$h_{lower}$</td>
<td>multi-map</td>
<td>map to the lower faces</td>
<td>$h_{cells}$</td>
</tr>
<tr>
<td>$h_{upper}$</td>
<td>multi-map</td>
<td>map to the upper faces</td>
<td>$h_{cells}$</td>
</tr>
<tr>
<td>$h_{boundary_map}$</td>
<td>multi-map</td>
<td>map to the boundary faces</td>
<td>$h_{cells}$</td>
</tr>
</tbody>
</table>
4.3 Solid Heat Transfer Model Implementation

Equation (3.27) is numerically equal to equation (3.29), which is in the form of $AX = B$, where the coefficient matrix $A$ is calculated using equation (3.31), and vector $B$ is calculated using equation (3.30).

If Euler implicit time scheme is employed, the right side vector will be simplified as following:

$$L(E^{n+1,p}) = E^{n+1,p} - E^n - \Delta t R^{n+1}(E^{n+1,p}), \quad (4.5)$$

where $E^{n+1,p}$ is the value of local Newton iteration step, and $E^n$ is the old value. The residual term $R$ includes volumetrical thermal source and face heat flux integrals, and might include a coupling source in coupled case. So it was calculated with unit-apply rule, and new source terms can be added by adding new apply rules. The volumetrical thermal source is integrated of the thermal generate rate over cell volume, while the face heat flux is integrated over all faces of the given cell. The heat flux is calculated using equation (3.3), and the temperature gradient is calculated using method described in section (2.3).

The Jacobian matrix used in the Newton method, described by equation (3.31), consists of a diagonal block matrix combined with off-diagonal matrices formed from the differentiation of the heat flux function, which are calculated numerically using second order central differential scheme and given as following:

$$f_{jm} = A_{fc}^{n+1} q_{fc}(h_{cr} \rightarrow E + \Delta E) - q_{fc}(h_{cr} \rightarrow E - \Delta E) \over 2\Delta E \quad (4.6)$$

$$f_{jp} = A_{fc}^{n+1} q_{fc}(h_{cl} \rightarrow E + \Delta E) - q_{fc}(h_{cl} \rightarrow E - \Delta E) \over 2\Delta E \quad (4.7)$$

where $f_{jm}$, and $f_{jp}$ are Jacobins of the heat flux functions located at faces and $f_{jm}$ reflects the effect of right cell value, while $f_{jp}$ reflects the effect of left cell value.
value. So $f_{jm}$ corresponding to right cell of the upper faces of a given cell goes to upper off-diagonal matrix, while $f_{jp}$ corresponding to left cell of the lower faces of a given cell goes to lower off-diagonal matrix. The volumetrical thermal source is treated explicitly, so no contribution to diagonal block matrix. So the identity matrix with $f_{jm}$, and $f_{jp}$ together go to the diagonal matrix. If the boundary is treated implicitly, the boundary face heat flux Jacobian is calculated:

$$f_{ci} = A_{fc}^{n+1} q_{fe} (h_{ci} \rightarrow E + \Delta E) - q_{fe} (h_{ci} \rightarrow E - \Delta E) \quad (4.8)$$

This term will go into the diagonal matrix and given as following:

$$D = I - \frac{\Delta t}{\Delta V} \left[ - \sum (h_{\text{upper}} \rightarrow f_{jp}) + \sum h_{\text{lower}} \rightarrow f_{jm} - \sum f_{ci} \right] \quad (4.9)$$

Boundary faces work as upper faces, so $f_{ci}$ has the same sign of $f_{jp}$.

Finally, an iterative linear solver is called to get the solution of thermal energy. If temperature dependent properties are used, the temperature can be solved from equation (3.2).

For fluid model and solid heat transfer model coupling case, fluid and solid solvers use separated grids, the same physical face has different index in fluid face entity and solid face entity, so a reference entity set and two maps of solid_face and fluid_face were created, which listed in table 4.3. The reference entity set is allocated for the interface number, and solid_face and fluid_face maps are allocated for the reference entity set. In this way, the interface values are computed at reference entity set, but can be accessed at the same manner of other boundary faces do. The interface values are evaluated based on this reference entity set by extrapolating fluid cell and solid cells values through maps of solid_face,
fluid_face, ci, and h_ci. For example, interface temperature is calculated using equation (4.10).

<table>
<thead>
<tr>
<th>Name</th>
<th>Value Type</th>
<th>Function</th>
<th>EntitySet</th>
</tr>
</thead>
<tbody>
<tr>
<td>solid_face</td>
<td>map</td>
<td>access to the face of solid side</td>
<td>interface</td>
</tr>
<tr>
<td>fluid_face</td>
<td>map</td>
<td>access to the face of fluid side</td>
<td>interface</td>
</tr>
</tbody>
</table>

\[
T_{\text{interface}} \leftarrow \begin{pmatrix}
    \text{solid}_\text{face} \to h_{\text{ci}} \to h_{\text{cellcenter}} \\
    \text{solid}_\text{face} \to h_{\text{ci}} \to h_{\text{Tcell}} \\
    \text{solid}_\text{face} \to h_{\text{ci}} \to h_{k_{\text{conduct}}} \\
    \text{solid}_\text{face} \to (h_{\text{facecenter}}, h_{\text{area}}) \\
    \text{fluid}_\text{face} \to ci \to (\text{cellcenter}, T_{\text{cell}}) \\
    \text{fluid}_\text{face} \to ci \to k_{\text{conduct}}
\end{pmatrix}
\]

(4.10)

New rules for this couple interface are required to added to the fact database, but main model needn’t anything change. The Loci scheduler will automatically check the data consistency. If the fact database provided inconsistent information (e.g., heat fluxes were impossible to compute with the given facts, or no unique solution was found), then the system would automatically generate error messages warning the user of incomplete or inconsistent formulation. In couple case, the interface conditions will go into source, and Jacobian matrices as needed. For these computations are finished at each Newton iteration, so this couple is a kind of implicit and tight couple.
4.4 Cooling Channel Model

The quasi-1D cooling channel can be curve or straight line and discretized as a set of straight segments, which is called the cooling cell, with one inflow face and one outflow face. The cooling channel has a virtual circular or rectangular section. The section geometry parameters and face center position will be read from an individual grid file. The face area is calculated using the channel section parameters. Cooling cell center values, such as position, temperature, will be calculated by averaging the inflow face center and outflow face center positions and temperatures. So two maps of $c_{\text{lower}}$ and $c_{\text{upper}}$ are created to access to inflow and outflow face. The prefix $c_{\text{c}}$ represents for cooling channel contrast to solid and fluid models. Also the face values are extrapolated from the right and left side cell’s values, so two maps of $c_{\text{cr}}$ and $c_{\text{cl}}$ are created to access to the right and left cell. These maps are listed in table 4.4. The cell volume is calculated using inflow and outflow face center positions and their section parameters.

The cooling channel was designed such that each cooling cell goes through one solid cell, with the inflow face and outflow face of the cooling channel cell be located in two faces of the solid cell. A special entity set of cooled\_solidcc was created for these cooled solid cells. Also a map of solid2cool was created to access the cooling channel cell for solid cell. For the purpose of evaluating equation (3.59), a stencil of solid cells is constructed around the cooling cell, excluding the solid cells that contain the cooling cell, so a multi-map of cool\_stencil was created for cooling cell to access to the solid stencil cells. The heat source calculated in equation (3.42) will be explicitly added to the cooled solid cell, which the rule specification was shown in equation (4.11), and this thermal source is assumed to be diffused to the neighbor solid cells. The cooling\_src calculated in cooling channel side was transferred to cooled\_src of cooled solid cell, and reduced to $h_{\text{src}}$ together with
Table 4.3: Cooling Channel Grid Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Value Type</th>
<th>Function</th>
<th>EntitySet</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_lower</td>
<td>map</td>
<td>access to the inflow face</td>
<td>c_cells</td>
</tr>
<tr>
<td>c_upper</td>
<td>map</td>
<td>access to the outflow face</td>
<td>c_cells</td>
</tr>
<tr>
<td>c_cr</td>
<td>map</td>
<td>access to the right cell</td>
<td>c_faces</td>
</tr>
<tr>
<td>c_cl</td>
<td>map</td>
<td>access to the left cell</td>
<td>c_faces</td>
</tr>
<tr>
<td>solid2cool</td>
<td>map</td>
<td>access to the cool cell</td>
<td>cooled_solidcc</td>
</tr>
<tr>
<td>coolStencil</td>
<td>multiMap</td>
<td>access to the solid cell</td>
<td>c_cells</td>
</tr>
</tbody>
</table>

other source terms.

\[
h_{\text{src}} \leftarrow \text{cooled}_{\text{src}} \leftarrow [\text{cool2solid} \rightarrow \text{cooling}_{\text{src}}] \quad (4.11)
\]

The cooling channel \( T_{\text{wall}} \) was implemented as equation (4.12) and averaged over all stencil computations. This computation is scheduled in each Newton iteration, for the solid cell temperature is solved at Newton iterative level, but a steady state cooling channel model is implemented, so there is no senses of time accurate issues. Equation (3.40) and (3.41) are solved iteratively in local senses, same as Gauss-Seidel iteration.

\[
T_{\text{wall}} \leftarrow \begin{cases} 
\text{cool}_\text{stencil} \rightarrow (h_{\text{cellcenter}}, h_{T_{\text{cell}}}) \\
\text{cool}_\text{stencil} \rightarrow h_{k_\text{conduct}} \\
c_{T_{\text{cell}}}, c_{k_\text{conduct}}, c_{\text{diameter}}
\end{cases} \quad (4.12)
\]

4.5 Coupling of the Models

For finite volume scheme is employed, the interface source flux integrals is computed and will go into source terms. If implicit method, such as Newton iterative method, is used, the source Jacobian is also computed. The interface
jacobian contribution will go into diagonal Jacobian matrix. It was demonstrated as following:

\[
q_{\text{interface}} = -A_{\text{interface}} K_{\text{interface}} \frac{T_{\text{interface}}^{n+1,p} - T_{\text{solid}}^{n+1,p}}{d} \hat{n} \tag{4.13}
\]

where \( p \) stands for the \( p \)th step of Newton iteration, and \( T_{\text{interface}} \) is computed using equation (4.10), that means \( T_{\text{interface}} \) is an implicit function of \( T_{\text{solid}} \), and \( T_{\text{fluid}} \), read as:

\[
T_{\text{interface}} = F(T_{\text{fluid}}^{n+1,p}, T_{\text{solid}}^{n+1,p}) \tag{4.14}
\]

So the derivative term from \( T_{\text{interface}} \) is a contribution to jacobian matrix. This term is numerically calculated, so no need a explicit form.

Both fluid and solid models are solved using Newton method, so this is easily scheduled. And the coupling is tight and seamless.

In the case of cooling channel coupling with solid models, the heat source transferred is treated as an explicit term, It will be reduced to source term in solid side by creating a new apply rule. In cooling channel side, it will go into equation (3.41). And no jacobian of this source is calculated. So it’s a loosely and explicitly couple.
As discussed in chapter I, Loci and CHEM system requires each new built model must be tested before coupling to flow solver. In this chapter, several preliminary test cases of both solid conduction and cooling channel flow will be introduced and briefly discussed. Solid heat conduction solver is verified to present the code works properly, and block to block communications are discussed.

5.1 Verification of Heat Conduction Solver

The main object of verification is to estimate the levels of uncertainty and error in engineering simulations. Verification assessment determines whether the implementation of the mathematical physical models, including solution algorithm are accurate through comparison to exact analytical results, it examines the mathematics in the models. Verification assessment process includes examining the iterative convergence and the order of grid and temporal convergence.

The verification process compares a computer program’s numerical results to an exact solution obtained independently. The analytical solution to the linear heat conduction equation can be obtained using Separation of Variables(SOV) or Fourier Integral Transform(FIT)[48] methods. The general solution for a three-dimensional cube in a form of series theoretically has infinite terms, which looks like:

\[ T(x, y, z, t) = A \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} e^{-\alpha t} \sin(a_i x) \sin(a_j y) \sin(a_k z), \]  

(5.1)
where $A, a_i, a_j, a_k$ and $\alpha$ are constant, $x, y, z$ are spatial coordinates, $t$ is time. The number of the summed terms should be larger enough to include all effective eigenvalues and kernel functions to get an accurate analytical solution. For one-dimensional problem, the number can be 10000 or more to obtain an exact solution. So, for three-dimensional problem this at least will be $10^{12}$, the computation is non-affordable even using today’s fast computer.

Since verification is an exercise in mathematics to examine whether the mathematical partial differential equation is solved accurately, a real physical solution is not necessarily required. Roache[49] advocates the method of manufactured solutions in which an analytical function is arbitrarily selected as the solution. So, a one-dimensional manufactured solution[50] and a three-dimensional function easily evaluated are employed in this study. Passing this manufactured function to the mathematical partial differential equation will produce a time-space-dependent volumetric source term. The source term and initial/boundary conditions from the manufactured function are imposed to the code to get a numerical solution. Of course, the program must have a volumetric source term input option. One important aspect of verification is to check the dependence of the numerical solution on the discretization in spatial and temporal. Provided that the numerical solution is accurate, as the discretization is refined, the accuracy improvement will be achieved at a rate consistent with the theoretical convergence rate for the numerical method. Also, the temporal residual, Newton iterative and Gauss-Seidel iterative convergence are checked.
5.1.1 Two Manufactured Solutions

First three-dimensional function selected as the exact solution is read as:

\[ T(x, y, z, t) = A(2.0 - e^{-\alpha t}) \sin(x) \sin(y) \sin(z) \] (5.2)

The exponential relation of time variable makes the solution convergent to a steady values as the time marching forward, this also helps to examine the numerical method how to follow the exact solution at some time, where the exponential part reaches to a small value. The volumetric source term after passing this function to the PDE (1.2) is obtained as:

\[ \frac{d}{dt}(\rho C_p T) + \text{div}(q_{xi}) = g \] (5.3)

\[ q_{xi} = -k \frac{dT}{dx_i} \] (5.4)

is obtained as:

\[ g(x, y, z, t) = Ak(6.0 - 2.0e^{-\alpha t}) \sin(x) \sin(y) \sin(z) \] (5.5)

This function, the initial and boundary conditions can be easily evaluated. Obviously, function (5.2) is an exact solution of PDE (5.3). For the convenience of computation, material thermal properties are selected as constant, density \( \rho = 1.0 \), specific heat \( C_p = 1.0 \), thermal conductivity \( k = 1.0 \). The constant \( A \) is adjusted to make the solution value not too big or too small.

Infinite (\( E_{\text{inf}} \)) error norm or second (\( E_2 \)) error norm are employed which defined as:

\[ E_{\infty} = \max \| T_N(x, t) - T_A(x, t) \| \] (5.6)
\[ E_2 = \sqrt{\sum [T_N(x,t) - T_A(x,t)]^2 dv} \]  (5.7)

where \( T_N \) and \( T_A \) are numerical and analytical solutions, respectively, \( dv \) is the cell volume.

In the second situation, thermal conductivity \( k \) is assumed as a linear function of temperature, expressed as:

\[ k = k_0 (1 + k_1 T) \]  (5.8)

where \( k_0 \) and \( k_1 \) are constant. If density \( \rho = 1.0 \), specific heat \( C_p = 1.0 \), \( g = 0 \), \( k_0 = 1 \) and \( k_1 = 1 \) are used in equation 5.3, in one-dimensional case, the differential equation is derived as:

\[ \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left[ (1 + T) \frac{\partial T}{\partial x} \right] \]  (5.9)

Equation (5.10) can be verified as an exact solution for equation (5.9)

\[ T(x,t) = \frac{x^2}{6(\gamma - t)} - 1 \]  (5.10)

where \( \gamma \) is a constant to determine when the temperature to reach infinity. This singularity may help to verify the ability of a numerical method to follow the exact solution at the theoretical convergence rate.

In numerical simulation, as discussed in section 3.4 an implicit backward Euler time integration scheme is used, Newton iterative method is employed to solve the time integral equation, and Gauss-Seidel iterative method is employed to solve the numerical linear algebra system equation.
5.1.2 Iteration Convergence

The Gauss-Seidel iterative convergence is checked for the first function. In this case, grid size and time step are not concerned much, at some time step the Gauss-Seidel iterative history is shown in table 4.1. It can be seen that Gauss-Seidel iteration is convergent. At this step to make sure that the numerical linear algebra system equation is solved correctly.

For a non-linear problem, time integration can be finished in several steps. Residual of each Newton iteration is checked to make sure Newton iteration convergent in each time step. Newton iterative history of the first function is shown in table 4.2. It can be seen that Newton iteration is convergent.

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>Infinite Norm</th>
<th>Second Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.005434</td>
<td>0.0398463</td>
</tr>
<tr>
<td>2</td>
<td>0.00173734</td>
<td>0.012948</td>
</tr>
<tr>
<td>3</td>
<td>0.000586649</td>
<td>0.00527441</td>
</tr>
<tr>
<td>4</td>
<td>0.000250736</td>
<td>0.00231041</td>
</tr>
<tr>
<td>5</td>
<td>0.000103598</td>
<td>0.00104663</td>
</tr>
<tr>
<td>6</td>
<td>4.83339e-05</td>
<td>0.000483214</td>
</tr>
<tr>
<td>7</td>
<td>2.24167e-05</td>
<td>0.000225788</td>
</tr>
<tr>
<td>8</td>
<td>1.03188e-05</td>
<td>0.000106365</td>
</tr>
<tr>
<td>9</td>
<td>4.7394e-06</td>
<td>5.0396e-05</td>
</tr>
<tr>
<td>10</td>
<td>2.1917e-06</td>
<td>2.39786e-05</td>
</tr>
</tbody>
</table>

5.1.3 Spatial Convergence

Any comparison is based on the assumption that numerical solution is accurate. So the residual history for the first function is shown in figure 5.1 to make sure the numerical solution is comparable.
Table 5.2: Newton Iteration Convergence History

<table>
<thead>
<tr>
<th>Iteration No.</th>
<th>Second Norm</th>
<th>Infinite Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.079e-01</td>
<td>2.524e-02</td>
</tr>
<tr>
<td>2</td>
<td>6.200e-05</td>
<td>5.671e-06</td>
</tr>
<tr>
<td>3</td>
<td>3.361e-08</td>
<td>2.997e-09</td>
</tr>
</tbody>
</table>

Figure 5.1: Residual History for the First Function
Table 5.3: Second and Infinite Norm Errors for Spatial Convergence

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Time Step</th>
<th>Second Norm</th>
<th>Infinite Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.04</td>
<td>6.5497025571e-3</td>
<td>2.275176900e-02</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>1.611224686e-3</td>
<td>6.392674027e-03</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0025</td>
<td>3.6634107963e-4</td>
<td>1.685446179e-03</td>
</tr>
<tr>
<td>0.025</td>
<td>0.000625</td>
<td>7.1327600148e-5</td>
<td>4.321265455e-04</td>
</tr>
</tbody>
</table>

The computational domain is a 1X1X1 $m^3$ unit cube, and constant $A = 5$ is used. The initial and boundary conditions are derived from the analytical function. The numerical values at time of 5s are extracted. Grid sizes, corresponding time steps, the infinite and second error norms are calculated and listed in table 5.3, and the log plot of the error norms as the grid size is shown in figure 5.2. Time step is decreased such the error distributed from temporal term is negligible compared with spatial term. The time step is selected such that the mesh Fourier ($Fo = \frac{At}{\Delta x^2}$) remains a constant value of 1.0[50] as the mesh is refined. It can be seen the infinite error norm to decrease with a slope of 2 as the grid size decreases. So the spatial discretization scheme is second order in infinite error norm.

Also, comparison for the second function is performed. The computational domain is also a 1X1X1 $m^3$ unit cube. The initial and boundary conditions are computed from the analytical function. $\gamma = 0.44$ is used and numerical and exact results at $t = 0.4$ are compared. Exact solution and numerical solutions with different cell number are shown in figure 5.3 to provide a reference for comparison. The error norms computed with the same methods for the first function, related grid size and time steps are listed in table 4.4, and the log plot of the error norms as the grid size is shown in figure 5.4. It can be seen the infinite error norm to
Table 5.4: Second and Infinite Norm Errors for Spatial Convergence

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Time Step</th>
<th>Second Norm</th>
<th>Infinite Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.04</td>
<td>0.84753</td>
<td>1.63100</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>0.32845</td>
<td>0.73249</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0025</td>
<td>0.09486</td>
<td>0.23024</td>
</tr>
<tr>
<td>0.025</td>
<td>0.000625</td>
<td>0.02468</td>
<td>0.06242</td>
</tr>
<tr>
<td>0.0125</td>
<td>0.00015625</td>
<td>0.00624</td>
<td>0.01608</td>
</tr>
<tr>
<td>0.00625</td>
<td>0.0000390625</td>
<td>0.00156</td>
<td>0.00407</td>
</tr>
</tbody>
</table>

decrease with a slope of 2 as the grid size decreases. So the spatial discretization scheme is second order in infinite error norm for the second function.

5.1.4 Temporal Convergence

Compare to spatial convergence study, temporal convergence requires time step decreasing while maintaining a constant grid size. The grid size is selected to make a negligible spatial error contribution compared to temporal error. Here the first function is employed, and the grid size is selected to be 0.001, and time step are 0.2,
Figure 5.3: Exact and Numerical Solution for Second Function

Figure 5.4: Error Norm for Spatial Convergence
Table 5.5: Second and Infinite Norm Errors for Temporal Convergence

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Time Step</th>
<th>Second Norm</th>
<th>Infinite Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.2</td>
<td>7.80787e-06</td>
<td>3.491473792e-02</td>
</tr>
<tr>
<td>0.001</td>
<td>0.1</td>
<td>3.70889e-06</td>
<td>1.658518110e-02</td>
</tr>
<tr>
<td>0.001</td>
<td>0.05</td>
<td>1.80808e-06</td>
<td>8.085274110e-03</td>
</tr>
<tr>
<td>0.001</td>
<td>0.025</td>
<td>8.92732e-07</td>
<td>3.992062441e-03</td>
</tr>
<tr>
<td>0.001</td>
<td>0.0125</td>
<td>4.43566e-07</td>
<td>1.983512185e-03</td>
</tr>
<tr>
<td>0.001</td>
<td>0.00625</td>
<td>2.21080e-07</td>
<td>9.886132107e-04</td>
</tr>
</tbody>
</table>

0.1, 0.05, 0.025, 0.0125. The computation domain also change to 0.01X0.01X0.01 m³ in order not to increase the computation work. The infinite and second error norms are listed in table 4.5, and a log plot of error norm as time step is shown in figure 5.5. So the final temporal integration is first order in infinite and second error norms.

Figure 5.5: Error Norm for Temporal Convergence
5.2 Composite Solid Block

In practice, often more than one kind of material are used for the heat transfer structure, for example, a rocket chamber and nozzle wall usually is constructed by two layers of different property materials, the inner layer is composed of low thermal conductivity material to prevent heat from transferring to the outer metal structure concerning on the thermal stress, deformation even break. Especially, maybe more complicated structure is designed for the nozzle throat part, where heat flux is measured to be the maximum. The purpose here is to show the physical behavior if different materials used, and check the consistency of parameters, also present the solver’s ability to handle such problems.

In this study, a multi-block grid technology is employed, and each block grid can be specified with different or same properties. A single entity-set of geometry cell is created to include all solid cells, and a single algebraic equations system is assembled and solved for all these cells. Actually, this is the tight coupling and no explicit boundaries between solid blocks, of course no parameters exchanged between solid blocks. But, to average the left and right cell’s thermal conductivity and temperature to get the face values is not enough in the case of multi-materials used. Here the face values are calculated using both left and right sides cells’ properties based on the requirement of face heat flux consistency.

The geometry used is a 0.3m by 0.9m rectangular block, equally divided into three sub-blocks. Their thermal properties are listed in table 5.2. A heating boundary with heating density of 9000(W/m²) is specified at the bottom of the rectangular, and all other boundaries are specified with 400K temperature, and an initial temperature of 300K is specified for the whole block. The temperature contours with time step of 0.01s marching upto time of 80s are shown in figures 5.6, 5.7, 5.8, 5.9.
Table 5.6: Thermal Properties of Solid Block Materials

<table>
<thead>
<tr>
<th>Material</th>
<th>Copper</th>
<th>Aluminum</th>
<th>Brass</th>
<th>Bronze</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_p$</td>
<td>384.91</td>
<td>895</td>
<td>380</td>
<td>340</td>
</tr>
<tr>
<td>$k$</td>
<td>401</td>
<td>204</td>
<td>104</td>
<td>26</td>
</tr>
<tr>
<td>$\rho$</td>
<td>8920</td>
<td>2720</td>
<td>8520</td>
<td>8670</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1.1676e-4</td>
<td>8.38e-5</td>
<td>3.212e-5</td>
<td>8.82e-6</td>
</tr>
</tbody>
</table>

Figure 5.6: Temperature for Three Copper Blocks
Figure 5.7: Temperature for Copper-Aluminum Blocks

Figure 5.8: Temperature for Copper-Brass Blocks
Figure 5.9: Temperature for Copper-Bronze Blocks

The block is heated at all boundaries at earlier time and heat flows from boundaries into the composite block, because initial temperature is specified lower than boundary conditions. It is found that the heating density from the 400K boundaries is stronger than the heating density specified at the bottom at earlier time, but it is expected the 400K boundary temperature will be the lowest value in the final state, and then heat is flow out from these boundaries. From these figures, it can be seen that heat transfer behavior is largely affected by the thermal properties of the materials, as the thermal diffusion coefficient decreases from copper, aluminum, brass to bronze as shown in table 5.2, the change of of contour lines at the vicinity of material interface becomes faster. As the thermal diffusivity is a measure of how rapidly heat diffuses through a material.

More interesting computation is performed, in which copper, brass and bronze three materials are used, and a larger heating density (90000W/m\(^2\)) is specified at bottom and 300K temperature is specified at top, left and right sides are adiabatic, a same initial temperature of 300K and same time step of 0.01s are used as before.
In this case, the top boundary is initially no heat transferred, and heat can only flows into from the bottom side. The temperature distributions at different time are shown in figures from 5.10 to 5.14. It is observed that heat transfers in copper is faster than it does in brass, and the slowest in bronze, corresponding to their thermal diffusivity. Also more heat is transferred from copper to brass than it does from brass to bronze. It also can be found the parameters consistency are achieved in either cases.

![Temperature for Copper-Brass-Bronze Blocks at \( t = 100\)s](image)

**Figure 5.10: Temperature for Copper-Brass-Bronze Blocks at \( t = 100\)s**

### 5.3 Cooling Channel Test Cases

Two simple cooling channel cases are presented in this paper. The first case features a channel composed of two straight segments of pipe in a cubic solid block, as shown in figure 5.15. The inlet conditions for the cooling channel are listed in table 5.7, and the operating parameters for the solid block are listed in table 5.8. The convergence history is shown in figure 5.16. The solid temperature is shown in figure 5.17, whereas cooling channel temperature, pressure, specific enthalpy, and
Figure 5.11: Temperature for Copper-Brass-Bronze Blocks at $t = 200\,s$

Figure 5.12: Temperature for Copper-Brass-Bronze Blocks at $t = 300\,s$
Figure 5.13: Temperature for Copper-Brass-Bronze Blocks at $t = 400\,s$

Figure 5.14: Temperature for Copper-Brass-Bronze Blocks at $t = 500\,s$
vapor quality are shown in figure 5.18, 5.19, 5.20 and 5.21, respectively. It can be seen that as the specific enthalpy increases, the temperature increases before any phase changes, and as the vapor quality increases, the temperature decreases while the specific enthalpy continues to increase. The preliminary computational results are physically reasonable.

The second cooling channel configuration discussed in this study includes two independent channels in a cubic solid block, as shown in figure 5.22. At the present time, the cooling channel model allows for multiple independent channels to be present in the same solid matrix. Moreover, the geometry and inlet conditions can be different for each channel. The inlet conditions for this test case are listed in table 5.9, and the operating parameters of the solid block are listed in table 5.10. Figure 5.23 shows the solid block temperature distribution. In the case, no vapor appears. Cooling channel temperature, pressure, and specific enthalpy are shown in figures 5.24, 5.25, and 5.26, respectively. It can be seen that these computational results are also physically reasonable. In the final version of this paper, a real thruster nozzle case will be computed, and computational results will be compared with available experimental data.

Table 5.7: Cooling Channel Inlet Condition of the First Case

<table>
<thead>
<tr>
<th>$T$ (K)</th>
<th>350</th>
<th>$p$ (kPa)</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$</td>
<td>0</td>
<td>$\dot{m}$ (kg/s)</td>
<td>12.56</td>
</tr>
<tr>
<td>$e$ (m)</td>
<td>4.57e-5</td>
<td>$r$ (m)</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Table 5.8: Operating Parameters for Solid Block of the First Case

<table>
<thead>
<tr>
<th></th>
<th>south</th>
<th>adiabatic</th>
<th>north</th>
<th>adiabatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom (K)</td>
<td>1900</td>
<td></td>
<td>1800</td>
<td></td>
</tr>
<tr>
<td>left</td>
<td>adiabatic</td>
<td></td>
<td>right</td>
<td>adiabatic</td>
</tr>
<tr>
<td>dtmax (s)</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C_p(J/kg.K))</td>
<td>384.91</td>
<td></td>
<td>(k(W/m/K))</td>
<td>401</td>
</tr>
</tbody>
</table>

Table 5.9: Cooling Channel Inlet Condition of the Second Case

<table>
<thead>
<tr>
<th></th>
<th>T (K)</th>
<th>p (kPa)</th>
<th>w</th>
<th>(\dot{m} \ (kg/s))</th>
<th>e (m)</th>
<th>r (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>350</td>
<td>1</td>
<td>0</td>
<td>1.256</td>
<td>4.57e-5</td>
<td>0.003</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>2</td>
<td>0</td>
<td>0.240</td>
<td>4.57e-5</td>
<td>0.004</td>
</tr>
</tbody>
</table>

Table 5.10: Operating Parameters for Solid Block of the Second Case

<table>
<thead>
<tr>
<th></th>
<th>south</th>
<th>adiabatic</th>
<th>north</th>
<th>adiabatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom ( (W/m^2) )</td>
<td>-66400</td>
<td></td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>left</td>
<td>adiabatic</td>
<td></td>
<td>right</td>
<td>adiabatic</td>
</tr>
<tr>
<td>dtmax (s)</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C_p(J/kg.K))</td>
<td>384.91</td>
<td></td>
<td>(k(W/m/K))</td>
<td>401</td>
</tr>
</tbody>
</table>
Figure 5.15: Cooling Path for the First Case

Figure 5.16: Convergence History for Cooling Channel and Solid
Figure 5.17: Solid Block Temperature for the First Case

Figure 5.18: Cooling Channel Temperature for the First Case
Figure 5.19: Cooling Channel Pressure for the First Case

Figure 5.20: Cooling Channel Specific Enthalpy for the First Case
Figure 5.21: Cooling Channel Vapor Quality for the First Case

Figure 5.22: Cooling Path for the Second Case
Figure 5.23: Solid Block Temperature for the Second Case

Figure 5.24: Cooling Channel Temperature for the Second Case
Figure 5.25: Cooling Channel Pressure for the Second Case

Figure 5.26: Cooling Channel Specific Enthalpy for the Second Case
CHAPTER VI
VALIDATION RESULTS

In this chapter, two real engineering problems are simulated. The first case involves a fully coupled solid-fluid problem, the second case involves fluid-solid-cooling channel three models. The calculated results for both cases are compared with available experimental data.

6.1 Air in a Cooled Converging-Diverging Nozzle

The first test case is a fully coupled fluid-solid problem, involving the flow of heated air in a cooled converging-diverging nozzle. The analysis of this case is based on the data reported by Back et al.[51], they investigated the convective heat transfer from turbulent boundary layers accelerated under the influence of large pressure gradients in a cooled converging-diverging nozzle. The test nozzle has a throat diameter of 0.0458 m, a contraction-area ratio 7.75 to 1, an expansion-area ratio of 2.68 to 1, a convergent half-angle of 30°, and a divergent half-angle of 15°.

The nozzle mesh is shown in figure 6.1. The computed domain matches the experimental one. The operating parameters for the fluid part are listed in table 6.1. All physical values are from the experimental apparatus in [51]. The interface is processed as a traditional no-slip boundary condition for air flow, and the interface temperature is computed by considering the grid spacing and thermodynamic properties (conductivity and heat capacity) of the materials on both sides of the interface.

73
Table 6.1: Operating Parameters of Nozzle Flow

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nozzle inlet pressure (N/m²)</td>
<td>$5.171 \times 10^5$</td>
</tr>
<tr>
<td>nozzle inlet temperature (K)</td>
<td>843.33</td>
</tr>
<tr>
<td>nozzle inlet density (kg/m³)</td>
<td>2.1306</td>
</tr>
<tr>
<td>transport model</td>
<td>sutherland</td>
</tr>
<tr>
<td>turbulence model</td>
<td>Spalart-Allmaras</td>
</tr>
<tr>
<td>limiter</td>
<td>Venkatakrishnan</td>
</tr>
<tr>
<td>cflmax</td>
<td>500.0</td>
</tr>
<tr>
<td>dtmax (s)</td>
<td>1.0e-5</td>
</tr>
<tr>
<td>iteration number</td>
<td>3000</td>
</tr>
</tbody>
</table>
For the solid phase, the temperature at the external side of the nozzle wall was specified from experimental values, as illustrated in figure 6.2. The interface heat flux was extracted from flow side, as already discussed. The temperatures at the solid sides that correspond with the nozzle inlet and outlet are specified as 299 and 283 Kelvin, respectively, as dictated by the experimental data. Computations are performed for three different AISI stainless steels. The physical and thermal properties of stainless steel used are listed in table 6.2 (obtained from [36]). Since only the steady-state solution to this problem is of interest, independent time-steps are chosen for the fluid mechanics and solid mechanics algorithms, in order to accelerate convergence. A time-step on the order of seconds is used for the solid part, while the fluid part time-step is on the order of milliseconds.

![Figure 6.2: Hot Air Nozzle Wall Outside Temperature](image)

The coupled convergence history is shown in figure 6.3, where the label "fd302" stands for fluid coupled with AISI302 solid, "sd302" stand for AISI302 solid, and so on. As it can be seen, a four-order of magnitude reduction in residuals for both the solid and gas phases is achieved.
Table 6.2: Physical and Thermal Properties of AISI Stainless Steels at 400K

<table>
<thead>
<tr>
<th>Composition</th>
<th>Density ($kg/m^3$)</th>
<th>Heat Capacity ($J/kg.K$)</th>
<th>Thermal Conductivity ($W/m/K$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AISI302</td>
<td>8055</td>
<td>512</td>
<td>17.3</td>
</tr>
<tr>
<td>AISI304</td>
<td>7900</td>
<td>515</td>
<td>16.6</td>
</tr>
<tr>
<td>AISI316</td>
<td>8238</td>
<td>504</td>
<td>15.2</td>
</tr>
</tbody>
</table>

Figure 6.3: Convergence History for Solid and Flow Solvers
The nozzle flow calculated results of density, pressure, temperature, velocity and Mach number are shown in figure 6.4, 6.5, 6.6, 6.7, 6.8, respectively. The solid nozzle wall temperature contour are shown in figure 6.9 and 6.10, and the coupled temperature contours are shown in figure 6.11.

Figure 6.4: Nozzle Flow Density

A comparison of the computed wall temperature and heat flux using adiabatic and constant temperature wall boundary conditions with the coupled results shows the improvement of prediction by the coupled model, which is shown in figure 6.12 and 6.13. If an adiabatic boundary condition is specified, the wall heat flux is zero, but the wall temperature is higher. While, the constant wall temperature boundary condition can not account the change at the throat part, so it makes the error of heat flux computation bigger.

A comparison of the computed heat transfer coefficient with experimental results provided by Back is shown in figure 6.14. The comparison is satisfactory. However, results that are more promising are the computed wall temperatures,
Figure 6.5: Nozzle Flow Pressure

Figure 6.6: Nozzle Flow Temperature
Figure 6.7: Nozzle Flow Velocity

Figure 6.8: Nozzle Flow Mach Number
Figure 6.9: Nozzle Wall Temperature (AISI302)

Figure 6.10: Nozzle Wall Temperature (AISI316)
Figure 6.11: Nozzle Flow and Solid Wall Temperature

Figure 6.12: Comparison of Wall Temperature
Figure 6.13: Comparison of Wall Heat Flux

shown in figure 6.15. This figure shows the predicted wall temperatures from an earlier uncoupled case and from the current coupled case. The experimental results are also included under the label “test.” As it can be seen, the temperature predictions are significantly improved by coupling of fluid and solid models. Moreover, the predicted wall temperatures are very close to the experimental values.

6.2 RBCC Thruster Nozzle Simulation

A more complex computation is carried out for Rocket Based Combined Cycle (RBCC) thruster nozzle. This analysis is based on the experimental data reported by Pal et al. [15]. Heat transfer characteristics are investigated, particularly in the throat part, where the peak heat flux occurs. This thruster has a two-dimensional section and the geometry is very compact as required by packaging considerations. A schematic of the nozzle is shown in figure 6.16. The nozzle has a length of 50.8 mm, a height of 44.45 mm, and a width of 76.2 mm. The nozzle throat
Figure 6.14: Comparison of Wall Heat Transfer Coefficient

Figure 6.15: Comparison of Wall Temperature
height is small as 2.5 mm and the outlet height is 15.24 mm. Gaseous oxygen (GO₂) and gaseous hydrogen (GH₂) are used as the propellant. The design includes two oxygen-free high-conductivity (OFHC) copper sections welded together to make up the nozzle flow path with stainless steel plates welded on top and bottom to strength the structure. The number ranging from 1 to 10 shown in the figure 6.16 denote each thermocouple hole. The Type-K thermocouples were silver-brazed to the bottom of the holes with approximately 1.27 mm of material separating them from the hot gases. Thermocouple #3 and #10 were not used, and remaining eight thermocouples were used to obtain the experimental data[15].

![Figure 6.16: Measurement Thruster Nozzle Section and Thermocouple Location](image)

Both chamber and nozzle of the thruster are fully water-cooled. The nozzle is cooled on all four sides. The water cooling passages with diameter of 1.5875 mm, and run parallel inside nozzle walls. Twenty-four channels are positioned in each of the top and bottom walls, and four are in each side wall. The cooling channel passages in the top nozzle wall are shown in figure 6.17, and the passages in the side wall are shown in figure 6.18. Water enters from the center manifolds, provides
impingement cooling to the throat region, bifurcates to cool the converging and diverging sections and then exits through two manifolds. There are four identical independent water circuits (two each on top and bottom) that are separated by a 2.54 mm "middle land region" on the top and bottom walls, where the Type-K thermocouples are located. The thermocouples temperature and water inlet and outlet temperature were recorded during the experiment.

Figure 6.17: Cooling Channel Passages in the Top Nozzle Wall

The cooling channel passages are formed by drilling and sealing processes, so the actual design of the nozzle wall is very complex. Therefore, a simplified geometric model was generated, in order to reduce the grid generation difficulties and computation cost of the simulation. The computation meshes of nozzle flow and wall are shown in figure 6.19, which is one fourth of the whole nozzle due to symmetry (the nozzle side is not shown). The cooling channel arrangement is shown in figure 6.20. There are twelve channels in the computation domain cooling each of the converging and diverging portions of the nozzle wall, plus three main vertical channels, for a total of 27 channels, as indicated in the figure. Water enters
Figure 6.18: Cooling Channel Passages in the side Nozzle Wall

from the second(#2) central channel, bifurcates to channel #4, #5, #6, #7, ..., #26, #27, then emerged to exit from channel #1, #3.

The thruster nozzle inlet flow boundary was modelled with the mass flow-rate fixed at the level specified by the experiments. The fuel and oxidizer were assumed to be completely mixed. Equilibrium properties to specify the inlet were obtained using CEA[52] at the experimental chamber pressure and mixture ratio and are summarized in table 6.3. Case #12 of the experimental sequence was employed[15].

The downstream boundary was treated as a supersonic outflow, whereby all variables are extrapolated from the interior of the domain. The interface is processed as a traditional no-slip boundary condition for thruster nozzle flow, and the interface temperature is computed by considering the grid spacing and thermodynamic properties (conductivity and heat capacity) of the materials on both sides of the interface. The side wall boundary temperature was set to be equal to the top wall temperature, shown in figure 6.21[15].
Figure 6.19: Thruster Fluid and Solid Grids

Figure 6.20: Cooling Lines Arrangement
Table 6.3: Inflow Conditions and Computational Model

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>O/F</td>
<td>8.0</td>
</tr>
<tr>
<td>$P_e (N/M^2)$</td>
<td>3.4476E6</td>
</tr>
<tr>
<td>$T_e (K)$</td>
<td>3576</td>
</tr>
<tr>
<td>$\rho_e (kg/m^3)$</td>
<td>1.8158</td>
</tr>
<tr>
<td>$\dot{m}_{H_2} (kg/s)$</td>
<td>0.034473</td>
</tr>
<tr>
<td>$\dot{m}_{O_2} (kg/s)$</td>
<td>0.2758</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>1.1282</td>
</tr>
<tr>
<td>Molecular Weight</td>
<td>15.642</td>
</tr>
<tr>
<td>$a_e (m/s)$</td>
<td>1464.4</td>
</tr>
<tr>
<td>Mass Fraction</td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>0.002744</td>
</tr>
<tr>
<td>$H_2$</td>
<td>0.01657</td>
</tr>
<tr>
<td>$H_2O$</td>
<td>0.75654</td>
</tr>
<tr>
<td>$O$</td>
<td>0.02115</td>
</tr>
<tr>
<td>$OH$</td>
<td>0.11824</td>
</tr>
<tr>
<td>$O_2$</td>
<td>0.08477</td>
</tr>
<tr>
<td>Transport Model</td>
<td>Chemkin</td>
</tr>
<tr>
<td>Diffusion Model</td>
<td>Chemkin</td>
</tr>
<tr>
<td>Chemistry Model</td>
<td>$H_2 - O_2 - 6s28r$</td>
</tr>
<tr>
<td>Turbulence Model</td>
<td>Menter’s BSL models</td>
</tr>
</tbody>
</table>
For thruster nozzle wall, the operating parameters are listed in table 6.4. The interface heat flux is extracted from nozzle flow side, as already discussed. The constant values of temperature for south, left and right walls were specified as an approximate of the experimental data. Temperature-dependent properties are used, with coefficients obtained by fitting available data[36].

<table>
<thead>
<tr>
<th>south (K)</th>
<th>300</th>
<th>north</th>
<th>adiabatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom (K)</td>
<td>interface</td>
<td>top (K)</td>
<td>350</td>
</tr>
<tr>
<td>left (K)</td>
<td>400</td>
<td>right (K)</td>
<td>350</td>
</tr>
<tr>
<td>$C_p (J/kg.K)$</td>
<td>$355.6(1 + 2.8 \times 10^{-4}T)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k (W/m/K)$</td>
<td>$419.8(1 - 1.6 \times 10^{-4}T)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho (kg/m^3)$</td>
<td>8933</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The cooling channels inlet conditions are listed in table 6.5. Currently, the cooling channel model can allow multiple, independent lines, but line crossing is not modeled. Therefore, each channel needs an individual inlet condition. All channels
are assumed to be smooth. The inlet temperature and mass flow-rate set for channel #2 are taken from experiment, the pressure is temporarily set as 5E5 Pa; inlet temperature for channel from #4 through #27 are specified to be the same as channel #2, assuming changes along channel #2 are small; conditions specified for channel #1, #3 are obtained from an early two-dimensional calculation. Specifically, in the two-dimensional model, just two channels are present, from the throat to the converging and diverging section, respectively. Their inlet conditions are specified to be the same as channel #4, all other conditions are kept unchanged. The outlet data obtained is now used as the inlet conditions for channels #1 and #3 in the current three-dimensional analysis.

Table 6.5: Cooling Channel Inlet Condition of the Thruster Nozzle

<table>
<thead>
<tr>
<th>T (K)</th>
<th>1</th>
<th>304</th>
<th>p (kPa)</th>
<th>1</th>
<th>476.95</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>287</td>
<td></td>
<td>2</td>
<td>5.0E2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>302</td>
<td></td>
<td>3</td>
<td>480.85</td>
</tr>
<tr>
<td></td>
<td>4-27</td>
<td>287</td>
<td></td>
<td>4-27</td>
<td>5.0E2</td>
</tr>
<tr>
<td>w</td>
<td>1</td>
<td>0</td>
<td>w (kg/s)</td>
<td>1</td>
<td>0.2518</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td></td>
<td>2</td>
<td>0.5036</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td></td>
<td>3</td>
<td>0.2518</td>
</tr>
<tr>
<td></td>
<td>4-27</td>
<td>0</td>
<td></td>
<td>4-27</td>
<td>0.02098</td>
</tr>
<tr>
<td>e (m)</td>
<td>1 smooth</td>
<td>1</td>
<td>r (mm)</td>
<td>1</td>
<td>2.286</td>
</tr>
<tr>
<td></td>
<td>2 Smooth</td>
<td>2</td>
<td></td>
<td>2</td>
<td>3.556</td>
</tr>
<tr>
<td></td>
<td>3 Smooth</td>
<td>3</td>
<td></td>
<td>3</td>
<td>2.286</td>
</tr>
<tr>
<td></td>
<td>4-27 Smooth</td>
<td>4-27</td>
<td></td>
<td>4-27</td>
<td>0.7937</td>
</tr>
</tbody>
</table>

The residual history for this calculation is shown in figure 6.22. As seen, a four-order of magnitude reduction in residuals for nozzle flow, solid heat transfer and cooling channel calculations is achieved. The nozzle flow Mach number, temperature, pressure, and velocity are shown in figure 6.23, 6.24, 6.25, 6.26. The results indicate that the flow is well behaved. Plots of the velocity vector, shown
in figure 6.27, indicate that no recirculation around the sharp corner was found. The Mach number at the exit plane is 2.808, and temperature drops from 3579 Kelvin to 2174 Kelvin.

![Residual History for Fluid, Solid and Cooling Flow](image)

Figure 6.22: Residual History for Fluid, Solid and Cooling Flow

![Flow Mach Number Distribution](image)

Figure 6.23: Flow Mach Number Distribution

A sample of temperature contours in the solid phase is shown in figure 6.28, 6.29, 6.30. The cooling channel path is explicitly shown in figure 6.31. Both the nozzle flow and solid wall temperatures are shown in figure 6.32. The cooling effects are evident in the local temperature contours: the cooling channel are working well.
Figure 6.24: Flow Temperature Distribution

Figure 6.25: Flow Pressure Distribution

Figure 6.26: Flow Velocity Distribution
Figure 6.27: Flow Velocity Vector

Figure 6.28: Solid Temperature Slices
Figure 6.29: Solid Temperature Slices

Figure 6.30: Solid Temperature Contour Slices
Figure 6.31: Cooled Solid Temperature Slice with Cooling Path

Figure 6.32: Thrust Nozzle and Cooled Solid Wall Temperature Slices
The cooling channel temperature along channel flow direction is shown in figure 6.33; the temperature difference between inlet and outlet is shown in figure 6.34. The temperature changes in the channel lines for the converging and diverging sections of the nozzle are higher than those of vertical lines. Moreover, using the outlet values of channel #26 and #25 as the inlet conditions of channel #1 and #3 is shown to be acceptable. The outlet temperatures are compared with the experimental data in table 6.6. The maximum error is approximately 2.8%. Overall, the channel outlet temperature values look reasonable: the temperature increase is not very high, because the nozzle dimension is very compact.

Figure 6.33: Cooling Channel Temperature

Table 6.6: Cooling Channel Outflow Temperature Comparison

<table>
<thead>
<tr>
<th>Channel Index</th>
<th>Calculated Outflow T(K)</th>
<th>Measured Outflow T(K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>307.7</td>
<td>316.33</td>
</tr>
<tr>
<td>3</td>
<td>303.939</td>
<td>308.63</td>
</tr>
</tbody>
</table>
Figure 6.34: Cooling Channel Inflow and Outflow Temperature

The computed nozzle wall heat flux is given in figure 6.35. The heat flux peaks at the nozzle throat due to the sharp corners in the nozzle profile and flow acceleration. The computational results are higher than previously reported values [15], which were obtained from a combination of a one-dimensional analytical model and two-dimensional numerical predictions. The computed wall temperature is shown in figure 6.36, and the thermocouple temperature at the thermocouple positions are shown in figure 6.37, together with the measured results. The computed temperature at the thermocouple locations are higher than the experimental date. The discrepancy is probably due to the differences between real and modeled geometry of the nozzle. As already mentioned, the cooling passages were formed by first drilling, then sealing at appropriate places in order to match the dimension requirements. Holes for burying thermocouples are drilled on the top and bottom nozzle walls. The resulting nozzle structure is fully three-dimensional. The byproducts of machining and the channels in side wall are not considered here, due to the difficulties of generating a grid for such a
geometry. It is very likely that the instrumental nozzle wall was cooled much than the computational model. A more complete simulation of the instrumental nozzle requires more works on grid generation, and probably a generalized grid.

Overall, the simulation provided reasonable results, that validate the models developed and implemented. Moreover, the ability to tackle realistic engineering applications was demonstrated.

Figure 6.35: Solid and Fluid Interface Heat Flux
Figure 6.36: Solid and Fluid Interface Temperature

Figure 6.37: Thermocouple Temperature
CHAPTER VII
SUMMARY AND FUTURE WORK

A finite-volume, time-accurate, non-linear heat conduction model has been built, verified and validated. A comparison of computational results with available experimental data shows good agreement. Different materials and temperature dependent properties can be employed in the same simulation, using a multi-block strategy.

A quasi-one-dimensional cooling channel model has been implemented: the model simplifications make it reasonably cheap for numerical computations, but realistic heat transfer behavior is included (e.g., coolant vaporization, frictional effects), and multiple cooling channels can be present in the same domain, which is frequently the case in engineering applications. A water/steam thermodynamic model is implemented for water cooling channel computations, which could be easily extended to other coolants. Preliminary results of coupling the quasi-one-dimensional cooling channel model with a three-dimensional solid heat transfer model are reasonable. Moreover, the full coupling of all three models (flow, solid, cooling channel), was presented and the results are reasonable when compared with available experimental data.

In addition to the models development, interface issues are also addressed, and the coupling strategy is introduced. The resulting models are fully coupled within Loci. The data structure and computation rules employed by Loci allow for the seamless integration of multi-physical components. One of the advantages of this kind of tight coupling is that the stability of the overall computation is enhanced,
when compared with loosely coupled approaches. Moreover, the coupled models can be applied to time-accurate, non-linear problems.

Coupling cooling channel flow model with solid-phase heat transfer and fluid dynamics provides a good starting point for further extensions to more complex physical models, such as thermal stress analysis, or composite material research. While the results of this study look quite promising, there is much left to address. A real cooling channel network might be very complex, and the channels might cross each other: in such case the solid grid generation becomes more difficult. Therefore, fully coupling fluid, solid and cooling channel models will require enhancements in grid generation technology.
REFERENCES


