INFORMATION TO USERS

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

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Simulation of SMC compression molding: Filling, curing, and volume changes

Hill, Ronald Raymond, Jr., Ph.D.

The Ohio State University, 1992
SIMULATION OF SMC COMPRESSION MOLDING -
FILLING, CURING, AND VOLUME CHANGES

DISSERTATION

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

By

Ronald R. Hill, Jr., B.Sc.Ch.E., M.S.

* * * * *

The Ohio State University
1992

Dissertation Committee:
L. J. Lee
J. L. Zakin
H. C. Hershey

Approved by
L. James Lee
Advisor, Department of
Chemical Engineering
To
Mom and Dad
ACKNOWLEDGMENTS

I would like to begin by thanking Dr. L. James Lee for the time and effort he has volunteered as my research advisor. I have learned much from Dr. Lee, and I consider the time I have spent working with him a valuable experience. I would also like to thank him for giving me the opportunity to speak at my first technical conference, which also resulted in my first journey outside of the United States. The financial support of the National Science Foundation funded Engineering Research Center for Net Shape Manufacturing and the Department of Chemical Engineering are very gratefully acknowledged. I also wish to thank Dr. J. L. Zakin and Dr. H. C. Hershey for their participation in my dissertation committee. Appreciation is also extended to Dr. Zakin for being a source of motivation to attend graduate school when I was an undergraduate several years ago.

Thanks also to Mark Kinkelaar and Jingyi Xu, who assisted me with some of the experimental work, to Wen Bin Young, who introduced me to the control volume method, and to the members of the Polymer Group for their friendship and graduate school memories. The material donations from Dr. Jose Castro, GenCorp, and Ashland Chemical are also appreciated. A grant from the Ohio Supercomputer Center for computing time on the CRAY Y-MP8/864 is also gratefully acknowledged.

Finally, I would like to thank my parents and family for their support and encouragement over the years.
VITA

March 7, 1963  Born, Columbus, Ohio
June, 1986  Bachelor of Science in Chemical Engineering
            The Ohio State University
September, 1988  Master of Science
            The Ohio State University

PUBLICATIONS


FIELD OF STUDY: Chemical Engineering
# TABLE OF CONTENTS

DEDICATION                                             ii  
ACKNOWLEDGMENTS                                       iii 
VITA                                                  iv  
LIST OF FIGURES                                        vii  
LIST OF TABLES                                         xii 

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>7</td>
</tr>
<tr>
<td>II.</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>46</td>
</tr>
<tr>
<td>III.</td>
<td>60</td>
</tr>
<tr>
<td>3.1</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>v</td>
</tr>
</tbody>
</table>
3.3 Kinetic Model for Unsaturated Polyester/Styrene System

IV. EXPERIMENTAL VERIFICATION OF MODEL SIMULATIONS
4.1 Mold Filling Experiments
4.2 Kinetic Characterization Using DSC
4.3 Temperature Measurements During Molding Experiments

V. ANALYSIS OF VOLUMETRIC CHANGES OF UNSATURATED POLYESTER RESINS DURING CURING
5.1 Introduction
5.2 Experimental Results
5.3 Modelling of Volumetric Changes
5.4 Determination of Parameters \( c \) and \( n \)
5.5 Estimation of Thermal Expansion Coefficients
5.6 Models for Conversion and Radical Concentration
5.7 Implementation of the Volume Change Model

VI. CONCLUSIONS AND RECOMMENDATIONS
6.1 Material Flow
6.2 Heat Transfer and Curing
6.3 Volume Change Model for UPE/styrene System

REFERENCES
APPENDIX A Fortran Code for Simulation Program
APPENDIX B Experimental Determination of \( K_p \)
APPENDIX C Modelling of 3-D Heat Transfer in Compression Molding
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Schematic representation of an SMC machine, Ref. [12].</td>
<td>8</td>
</tr>
<tr>
<td>1.2</td>
<td>Major aspects of the compression molding process.</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Simulated flow progression results for an isothermal Newtonian fluid, Ref. [26, 28].</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of simulation results using isothermal Newtonian and power law models, Ref. [28, 29].</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of simulation results using the GHS model with molding experiments which used one ply of SMC, Ref. [26, 28].</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Comparison of GHS model simulation results obtained from the FEM and BEM solutions to experimental results, Ref. [31].</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Comparison of simulation results using the GHS model with molding experiments which used three plies of SMC, Ref. [29].</td>
<td>28</td>
</tr>
<tr>
<td>2.6</td>
<td>Comparison of prediction using the thin charge model of Barone and Caulk to molding results, Ref. [35].</td>
<td>40</td>
</tr>
</tbody>
</table>
2.7 Comparison of simulation results obtained from the CV/FEM solution to experimental results from an automotive hood, Ref. [34].

2.8 Comparison of simulation results obtained from the CV/FEM solution to experimental results from a truck air deflector, Ref. [34].

2.9 Percent polymerization shrinkage versus temperature for five epoxy resins, Ref. [41].

2.10 Comparison of volume changes for an SMC resin (a) with LPA and (b) without LPA, Ref. [43, 44].

2.11 Elimination of all shrinkage prior to complete polymerization to reduce readout for an ideal material, Ref. [43].

2.12 Effect of MgO and glass fibers on mold shrinkage, Ref. [45].

2.13 Temperature, mold displacement, and pressure measurements during curing in a cylindrical reactor, Ref. [47].

3.1 The top view of a control volume in relation to the surrounding triangular elements.

3.2 An element and a portion of the control volume shown above in Figure 3.1.

3.3 Mold geometry, finite element mesh, and corresponding simulated flow front positions.
3.4 Simulated temperature profiles at various times during compression molding of an SMC paste. 80

3.5 Simulated conversion profiles at various times during compression molding of an SMC paste. 82

4.1 (a) Experimental mold geometry; (b) Comparison of simulation results to experimental flow progressions. 84

4.2 Method used to obtain activation energies and pre-exponential factors for the kinetic model. 86

4.3 Comparison of kinetic model simulations to isothermal DSC results. 89

4.4 Comparison of kinetic model simulations to scanning DSC results. 90

4.5 Comparison of simulated temperature profiles to actual temperature measurements during a molding experiment. 92

4.6 Comparison of simulated and experimental temperature profiles for a thermocouple located in a rib. 94

5.1 Combined dilatometer, DSC, and ESR results at 74°C. 99

5.2 Combined dilatometer, DSC, and ESR results at 85°C. 100

5.3 Overall shrinkage versus conversion master curve data. 102
5.4 Polymerization shrinkage versus radical concentration master curve data. 105

5.5 Research approach to developing a volume change model for the unsaturated polyester/styrene system. 107

5.6 Comparison of predicted polymerization shrinkage from equation (5.7) using average values of c and n to experimental master curve results. 110

5.7 Comparison of predicted polymerization shrinkage from equation (5.7) using average values of c and n to experimental results. 111

5.8 Experimental TMA results from a sample which had previously been cured in the dilatometer. 112

5.9 Comparison of simulated reaction rates to experimental DSC data. 117

5.10 Comparison of simulation results to radical concentration versus time data obtained from ESR experiments. 120

5.11 Comparison of simulation results from the volume change model to experimental volume change data. 122

5.12 Experimental mold displacement versus time showing the same qualitative trends given in Figure 5.11, Ref. [46, 47]. 123
B.1  Propagation rate constant computed from experimental data versus conversion.  

B.2  Propagation rate constant computed from experimental data versus time.  

C.1  New coordinate system and outward unit normal vectors, Ref. [71].
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Typical SMC Formulation, Ref. [11].</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Chemical Formulas of Materials Related to SMC</td>
<td>4, 5</td>
</tr>
<tr>
<td>1.3</td>
<td>Composition of E-Glass, Ref. [10].</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Properties of E-Glass, Ref. [10].</td>
<td>6</td>
</tr>
<tr>
<td>1.5</td>
<td>Representative Values of SMC Physical Properties</td>
<td>9, 10</td>
</tr>
<tr>
<td>4.1</td>
<td>Kinetic Parameter Values</td>
<td>87</td>
</tr>
<tr>
<td>5.1</td>
<td>Densities and Polymerization Shrinkages of Common Systems, Ref. [60].</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Resin Formulation Used in this Study</td>
<td>98</td>
</tr>
<tr>
<td>5.3</td>
<td>Values for $c$ and $n$</td>
<td>109</td>
</tr>
<tr>
<td>5.4</td>
<td>Kinetic Parameter Values</td>
<td>116</td>
</tr>
<tr>
<td>5.5</td>
<td>Polymerization Shrinkage Parameter Values</td>
<td>119</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 Applications of SMC

Sheet molding compound (SMC) is a thermosetting composite material consisting of a crosslinking unsaturated polyester/styrene matrix and short, randomly distributed glass fiber reinforcement. The most common usage of SMC is in the automotive industry, where compression molded SMC parts are often used as exterior body panels. For example, General Motors uses SMC for the body panels of its APV minivans, including the Pontiac Trans Sport, the Chevrolet Lumina, and the Oldsmobile Silhouette. Each van uses approximately 320 lbs. of SMC [1-3]. Plans have also been made to use SMC for the roofs, rear decks, and outer door panels of the 1993 Chevrolet Camaro and Pontiac Firebird [1]. SMC is a good candidate for the replacement of steel panels due to its relatively low cost, good stiffness and strength-to-weight ratio, corrosion resistance, design flexibility, and parts consolidation properties [4-8]. SMC has also been used in dishwashers, washing machines, clothes dryers, refrigerators, air conditioners, electrical housings, telephone switching cabinets, bathtubs, and shower enclosures [8, 9].

Various polyester molding compounds have been in use since the late 1940's [10], although usage of reinforced plastics in cars did not occur until the Chevrolet Corvette was introduced in 1953 [4]. In 1988, 270 million pounds of SMC were
shipped by suppliers, and this figure was projected to increase almost 6% per year through 1993 [4]. In 1985, an average of about 15 lbs. of SMC and BMC (bulk molding compound) was used per car. About 4.5 lbs., or 30%, of this was used for exterior body parts. It is predicted that by 1995, around 41 lbs. of SMC and BMC will be used per car, with exterior body parts comprising 70% of this usage [4].

1.2 Description of SMC and Its Constituent Materials

A typical SMC formulation for an automotive body panel is given in Table 1.1 [11]. Unsaturated polyester resin is usually used in SMC due to its low cost and fast cure ability, although phenolic and epoxy resins have been used to obtain altered properties [7, 10]. The most common unsaturated polyester used in SMC is obtained via the reaction between maleic anhydride and propylene glycol [10]. During reaction,

<table>
<thead>
<tr>
<th>Material</th>
<th>% by Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsaturated Polyester</td>
<td>10.50</td>
</tr>
<tr>
<td>Low Profile Additive</td>
<td>3.45</td>
</tr>
<tr>
<td>Styrene Monomer</td>
<td>13.40</td>
</tr>
<tr>
<td>Filler</td>
<td>40.70</td>
</tr>
<tr>
<td>Thickener</td>
<td>0.70</td>
</tr>
<tr>
<td>Initiator</td>
<td>0.25</td>
</tr>
<tr>
<td>Lubricant</td>
<td>1.00</td>
</tr>
<tr>
<td>Glass Fibers</td>
<td>30.00</td>
</tr>
</tbody>
</table>
both the maleate and fumarate isomers are formed, resulting in a polyester resin with about 94% fumarate double bonds [10]. The chemical structures of these isomers and other materials related to SMC are given in Table 1.2. Reaction speed with styrene is more rapid and composite physical properties are best when the fumarate content is high.

Low profile additives (LPA's) are thermoplastic materials which are used in SMC formulations to control part shrinkage and surface appearance. Typical LPA's include poly(vinyl acetate), poly(methyl methacrylate), saturated polyester urethanes, and polystyrene, among others. Use of these materials reduces or eliminates the shrinkage due to polymerization, which normally occurs during the polyester/styrene free radical chain-growth reaction. While the mechanism of low profile behavior is not totally understood, it has been suggested that the polymerization shrinkage is offset by expansion of the LPA with internal void formation, coupled with phase separation between the LPA and polyester resin [10]. LPA's have been used since their low shrink behavior was discovered in the early 1960's.

Other additives are also commonly used in SMC formulations. A filler, usually calcium carbonate (CaCO₃), is added to reduce cost and to improve the physical properties of molded parts [10]. Magnesium oxide, MgO, is added as a maturation or thickening agent in order to increase the viscosity of the molding compound to a puddy-like consistency for ease during handling. The increased viscosity, which may take 2-3 days to achieve, also results in more effective fiber transport during molding. Zinc stearate is added as a lubricant to allow for easier part removal after molding. The most common fiber reinforcement used in SMC is chopped E-glass (~1 in. long). The composition of E-glass and corresponding fiber properties are given in Tables 1.3 and 1.4, respectively [10].
Table 1.2 Chemical Formulas of Materials Related to SMC

<table>
<thead>
<tr>
<th>Structure Name</th>
<th>Chemical Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fumarate Polyester</td>
<td><img src="image" alt="Fumarate Polyester structure" /></td>
</tr>
<tr>
<td>Maleate Polyester</td>
<td><img src="image" alt="Maleate Polyester structure" /></td>
</tr>
<tr>
<td>Styrene</td>
<td><img src="image" alt="Styrene structure" /></td>
</tr>
<tr>
<td>Maleic Anhydride</td>
<td><img src="image" alt="Maleic Anhydride structure" /></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>---</td>
</tr>
<tr>
<td><strong>Propylene Glycol</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Propylene Glycol" /></td>
<td></td>
</tr>
<tr>
<td><strong>Benzoquinone</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Benzoquinone" /></td>
<td></td>
</tr>
<tr>
<td><strong>t-butyl perbenzoate</strong></td>
<td></td>
</tr>
<tr>
<td>(TBP)</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="t-butyl perbenzoate" /></td>
<td></td>
</tr>
<tr>
<td><strong>t-butyl peroxy-2-ethyl</strong></td>
<td></td>
</tr>
<tr>
<td>hexanoate (PDO)</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="t-butyl peroxy-2-ethyl" /></td>
<td></td>
</tr>
</tbody>
</table>
Table 1.3. Composition of E-Glass [10]

<table>
<thead>
<tr>
<th>Component</th>
<th>% by Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiO₂</td>
<td>55</td>
</tr>
<tr>
<td>Al₂O₃</td>
<td>15</td>
</tr>
<tr>
<td>CaO, MgO</td>
<td>22</td>
</tr>
<tr>
<td>B₂O₃</td>
<td>7</td>
</tr>
<tr>
<td>Na₂O, K₂O</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 1.4. Properties of E-Glass [10]

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Gravity</td>
<td>2.54</td>
</tr>
<tr>
<td>Young’s Modulus (22° C)</td>
<td>10.5 x 10⁶</td>
</tr>
<tr>
<td>Tensile Strength (22° C)</td>
<td>5 x 10⁵</td>
</tr>
<tr>
<td>Elastic Elongation (22° C)</td>
<td>4.8%</td>
</tr>
<tr>
<td>Coef. Therm. Exp. (in./in. °F x 10⁻⁶)</td>
<td>2.8</td>
</tr>
<tr>
<td>Refractive Index</td>
<td>1.547</td>
</tr>
</tbody>
</table>
An inhibitor such as benzoquinone is also often added to SMC formulations to prevent premature cure during processing. Sufficient time for material flow must be allowed during molding prior to material gelation in order to avoid problems such as incomplete mold filling. The copolymerization reaction can begin only after the inhibitor has been consumed by reaction with radicals which are formed originally from initiators or catalysts. Typical catalysts used to generate free radicals include t-butyl perbenzoate (TBP) and t-butyl peroxy-2-ethyl hexanoate (PDO), among others. The chemical structures of these materials are also given in Table 1.2.

When SMC is manufactured, the chopped fibers are sprinkled randomly between two moving layers of the paste molding compound. The outer surfaces of this sandwich are backed by a polyethylene film which is removed prior to molding. Rolls of this material then sit in storage until the viscosity increases to the desired level, and then used. A schematic representation of an SMC machine is given in Figure 1.1 [12]. Typical material properties for SMC which have appeared in the literature are given in Table 1.5 [8, 10, 13-18].

1.3 Scope of Study

Some of the important aspects of SMC compression molding are given in Figure 1.2. The flow which occurs during the mold filling stage may be described as a squeezing flow, with a moving mold boundary and a free surface at the flow front. Often, the heat transfer during this period causes a thin resin-rich layer to develop between the hot mold walls and the bulk SMC core. This can result in a lubricating effect which causes the flow regime to be approximately plug flow.
Figure 1.1. Schematic representation of an SMC machine [12].
Table 1.5. Representative Values of SMC Physical Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tensile Strength (psi)</td>
<td>8,000 - 25,000</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>4,500 - 20,000</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9,110</td>
<td>10</td>
</tr>
<tr>
<td>Tensile Modulus ($10^5$ psi)</td>
<td>14 - 25</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>10 - 25</td>
<td>8</td>
</tr>
<tr>
<td>Compressive Strength (psi)</td>
<td>15,000 - 30,000</td>
<td>8</td>
</tr>
<tr>
<td>Flexural Strength (psi)</td>
<td>9,000 - 35,000</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>26,760</td>
<td>10</td>
</tr>
<tr>
<td>Flexural Modulus ($10^3$ psi, 73°F)</td>
<td>1,000 - 2,500</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1,360</td>
<td>10</td>
</tr>
<tr>
<td>Impact Strength (ft-lb/in. of notch)</td>
<td>7 - 22</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.5 - 15</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>14.2</td>
<td>10</td>
</tr>
<tr>
<td>Impact Strength (ft-lb/in. unnotched)</td>
<td>19.1</td>
<td>10</td>
</tr>
<tr>
<td>Interlaminar Strength (26.1% glass, psi)</td>
<td>1,050</td>
<td>10</td>
</tr>
<tr>
<td>Elongation (%)</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3 - 5</td>
<td>8</td>
</tr>
<tr>
<td>Bending Elongation (%)</td>
<td>0.545</td>
<td>10</td>
</tr>
<tr>
<td>Dielectric Constant (1000 cycles)</td>
<td>4.15 - 5.17</td>
<td>13</td>
</tr>
<tr>
<td>Dielectric Loss (1000 cycles)</td>
<td>0.008 - 0.017</td>
<td>13</td>
</tr>
<tr>
<td>Dielectric Strength (0.125 in. thick, v./mil)</td>
<td>380 - 450</td>
<td>8</td>
</tr>
</tbody>
</table>
Table 1.5 (Continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Gravity (g/cc)</td>
<td>1.65 - 2.6</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>1.6 - 2.4</td>
<td>8</td>
</tr>
<tr>
<td>Heat Capacity (J/g-K)</td>
<td>1.0</td>
<td>14,15</td>
</tr>
<tr>
<td></td>
<td>1.004</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>1.079</td>
<td>17</td>
</tr>
<tr>
<td>Thermal Conductivity (W/m-K)</td>
<td>0.53</td>
<td>14,15</td>
</tr>
<tr>
<td></td>
<td>0.502</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>0.385</td>
<td>17</td>
</tr>
<tr>
<td>Mold (Linear) Shrinkage (in./in.)</td>
<td>0.0002 - 0.001</td>
<td>8</td>
</tr>
<tr>
<td>Coef. Linear Therm. Exp. 10^(-6) in./in./°C</td>
<td>6 - 30</td>
<td>8</td>
</tr>
<tr>
<td>Water Absorption (0.125 in. bar, 24 hr., %)</td>
<td>0.1 - 0.25</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>0.01 - 0.25</td>
<td>8</td>
</tr>
<tr>
<td>Burning Rate</td>
<td>None</td>
<td>13</td>
</tr>
<tr>
<td>Effect of Sunlight</td>
<td>Slight</td>
<td>13</td>
</tr>
<tr>
<td>Effect of Strong Acids or Bases</td>
<td>Attacked, acids</td>
<td>13</td>
</tr>
<tr>
<td>Effect of Organic Solvents</td>
<td>Resistant</td>
<td>13</td>
</tr>
<tr>
<td>Clarity</td>
<td>Opaque</td>
<td>13</td>
</tr>
<tr>
<td>Processing Temperature Range (°F)</td>
<td>270 - 330</td>
<td>8</td>
</tr>
<tr>
<td>Molding Pressure Range (psi)</td>
<td>500 - 2,000</td>
<td>8</td>
</tr>
<tr>
<td>Heat-deflection Temperature (°F, 264 psi)</td>
<td>375 - 500</td>
<td>8,13</td>
</tr>
<tr>
<td>Thermal Decomposition Temp. (°F)</td>
<td>1400</td>
<td>18</td>
</tr>
</tbody>
</table>
The Molding Process

Mold Filling → Curing → Cooling

- Filling before gelation
- Squeezing flow with moving boundary and free surface
- Near plug flow (heat transfer near surface provides a lubricating layer)

- Free radical crosslinking copolymerization
- Nonisothermal due to heat transfer from mold walls and reaction exotherm
- Expansion during heating
- Shrinkage due to polymerization

- Free surface shrinkage
- Surface defects
- Residual thermal stress

Figure 1.2. Major aspects of the compression molding process.
During the curing stage the free radical crosslinking copolymerization reaction occurs between styrene monomer and unsaturated polyester resin. The temperature distribution in the SMC can be very nonuniform during this period due to heat transfer from the hot mold walls and due to heat generation resulting from the exothermic reaction. Initially, the SMC may experience a thermal expansion as the material is heated by the mold. When the reaction begins to occur, however, polymerization shrinkage can result in an overall decrease in volume. Further shrinkage can occur during the cooling stage, when a thermal contraction takes place as the part cools to room temperature. The volume changes which occur during processing can result in surface defects and residual stresses.

The ability to simulate these aspects of the compression molding process numerically can reduce the amount of time and effort spent on costly experiments and trial and error optimization. While simulations for most of the major aspects of SMC processing have been developed individually, few attempts have been made to combine the models for mold filling, heat transfer, and curing into one simulation routine. Moreover, very little work has been done to model the volume changes which occur during SMC processing. The focus of the current research was to develop a computer package which can simulate the important stages of SMC compression molding (i.e., material flow and curing), to develop a model which can be used to predict volume changes during heating, curing, and cooling, and to characterize these features experimentally.

A control volume/finite element approach was used to obtain the pressure and velocity fields and to compute the flow progression during mold filling. The energy equation and a kinetic model were solved simultaneously for the temperature and conversion profiles, and a shrinkage model was developed to predict volumetric changes
of the part. A series of molding experiments was conducted to record the flow front location and material temperature; differential scanning calorimetry (DSC) was used to experimentally measure the polymerization reaction; and dilatometry, DSC, electron spin resonance spectroscopy (ESR), and thermomechanical analysis (TMA) results were used as a basis for the shrinkage model.
2.1 Models for the Flow of Sheet Molding Compound

The flow of sheet molding compound (SMC) during compression molding has been an active research area for many years. The compression molding process begins when plies of SMC, which have been stacked into an initial charge, are placed into a heated mold cavity. The top half of the mold is then lowered, compressing the material into the final part shape. Due to the presence of reinforcing fibers, SMC is at best isotropic only in the planar direction of the charge. As mentioned earlier, heat transfer from the mold walls results in the formation of a thin resin-rich lubricating layer between the mold and the central core region, causing the flow to resemble a membrane-like sheet which undergoes extensional deformation [19]. These observations, coupled with the fact that most SMC parts are thin, have resulted in a general trend among most researchers to transform the detailed 3-D flow problem into various 2-D representations of the flow by making assumptions based on the observed kinematics and process conditions. A number of experimental investigations have been conducted to study the flow kinematics during molding [19-23], while other studies have analyzed the process mathematically or simulated the process numerically [24-32].
Marker and Ford [20] used an instrumented mold to characterize the molding process by measuring the temperature and pressure at various locations in the mold cavity, and by recording the separation distance between the mold halves as a function of time. To delineate the nature of the flow pattern, plies of different color were used to form the initial charge, and the relative movement of each region during compression was noted. They observed a preferential flow of the hot, low viscosity layers near the mold surface, with an extensional plug-like deformation of the central core region. This flow pattern was later observed by Lee, Marker, and Griffith [21], who pointed out that the relative amounts of the hot lubricating layer and the central core region depended on viscosity, rate of heat transfer, and the mold closing speed.

Other results by Denton [22], and later experiments by Barone and Caulk [19] which used black and white charges of SMC, indicated that preferential flow of the outer layers did not occur under typical production conditions. Under these conditions, all plies of the charge were observed to extend uniformly with material slip occurring at the mold walls. Preferential flow only occurred when molding thick charges at a very slow mold closing speed. Kau [23] conducted experiments which used carbon black tracers to describe the flow, and proposed a partial-slip boundary condition at the mold surface based on the observed traces.

When the first models were developed for the flow of SMC, the anisotropic nature of the material was not taken into account. These models considered SMC to be an incompressible, isotropic, Newtonian or power law fluid. Moreover, the no-slip boundary condition was applied at the mold surface, which may not be a valid assumption in light of the experimental results described earlier. Silva-Neito, Fisher, and Birley [24] developed a model based on these assumptions, which resulted in Poisson's equation for the pressure distribution. The governing equation was solved
using the finite-difference method for flat geometries with the pressure boundary condition at the free surface being set at atmospheric. Therefore, the simulation could only be used when the charge was not in contact with any portion of the mold boundary, since the pressure at that location would no longer be atmospheric. Their model was basically used as a method to back-calculate an initial charge shape that would result in contact with all edges of the cavity simultaneously. Menges and Derek [25] also developed an early technique for predicting flow front progression to design charges.

Tucker and Folgar [26] developed a generalized Hele-Shaw (GHS) flow model for SMC which was an extension of Heiber and Shen's model for injection molding [27]. Some of the details of the GHS model are repeated here in order to explain the conditions under which it may be applicable, and to restate the final relevant expressions that have been used in numerical computations. This model applies when the thickness of the part is very thin compared to the lateral dimensions, and when the viscous forces are greater than inertial forces [28]. Under these conditions, the pressure is assumed not to vary in the thickness direction (Z-direction), and the equations of motion for the X-Y plane may be simplified to

\[
\frac{\partial P}{\partial X} = \frac{\partial \tau_{xz}}{\partial Z}, \quad \frac{\partial P}{\partial Y} = \frac{\partial \tau_{yz}}{\partial Z}.
\]

(2.1)

These equations were integrated with the boundary condition

\[
\tau_{xz} = 0, \quad \tau_{yz} = 0 \quad \text{at} \quad Z = \lambda,
\]
where [10]

\[
\lambda = \frac{\int_0^b \frac{Z \, dZ}{\eta}}{\int_0^b \frac{dZ}{\eta}},
\]

(2.2)

and the shear stress distributions were obtained:

\[
\tau_{xz} = \frac{\partial P}{\partial X} (Z - \lambda), \quad \tau_{yz} = \frac{\partial P}{\partial Y} (Z - \lambda).
\]

(2.3)

Here, \( \lambda \) is the value of \( Z \) at which the shear stresses are zero. Note that when the temperature profile is symmetric, the value of \( \lambda \) is \( h/2 \), where \( h = h(X,Y,t) \) is the cavity thickness.

Assuming generalized Newtonian fluid behavior, the constitutive relation for the molding material could be written as

\[
\tau_{xz} = \eta \frac{\partial V_x}{\partial Z}, \quad \tau_{yz} = \eta \frac{\partial V_y}{\partial Z}.
\]

(2.4)

Combining equations (2.3) and (2.4), and applying the no-slip boundary condition at the mold surface, the velocity distribution was obtained.

\[
V_x(Z) = \frac{\partial P}{\partial X} \int_0^Z \frac{(Z - \lambda)}{\eta} \, dZ, \quad V_y(Z) = \frac{\partial P}{\partial Y} \int_0^Z \frac{(Z - \lambda)}{\eta} \, dZ
\]

(2.5)
The gap-averaged velocity may be defined as

\[
\bar{V}_x = \frac{1}{h} \int_0^h V_x \, dZ, \quad \bar{V}_y = \frac{1}{h} \int_0^h V_y \, dZ.
\]  

(2.6)

Combining (2.5) and (2.6), and interchanging the order of integration, the average planar velocities resulted

\[
\bar{V}_x = -\frac{S}{h} \frac{\partial P}{\partial X}, \quad \bar{V}_y = -\frac{S}{h} \frac{\partial P}{\partial Y},
\]  

(2.7)

where

\[
S = \int_0^h \frac{(Z - \lambda)^2}{\eta} \, dZ.
\]  

(2.8)

The pressure field may be obtained by combining equations (2.7) and (2.8) with the continuity equation

\[
\frac{\partial V_x}{\partial X} + \frac{\partial V_y}{\partial Y} + \frac{\partial V_z}{\partial Z} = 0.
\]  

(2.9)

Assuming an approximate Z-velocity of

\[
V_z = \left( \frac{h}{\lambda} \right) Z,
\]  

(2.10)
which can be derived for a "stretching" type flow, one obtains

\[ \frac{\partial V_z}{\partial Z} = \frac{h}{h}, \quad (2.11) \]

where \( h = \frac{dh}{dt} \) is the rate of mold closure. Inserting (2.11) into (2.9), and integrating across the cavity thickness,

\[ \int_0^h \frac{\partial V_x}{\partial X} dZ + \int_0^h \frac{\partial V_y}{\partial Y} dZ + \dot{h} = 0 \quad (2.12) \]

Multiplying the average velocity given in equation (2.6) by \( h \), differentiating with respect to \( X \), and applying the Leibnitz formula, gives for the \( X \)-component

\[ \frac{\partial (h \bar{V}_x)}{\partial X} = \frac{\partial}{\partial X} \int_0^h V_x dZ = \int_0^h \frac{\partial V_x}{\partial X} dZ \quad (2.13) \]

Performing the same operations for the \( Y \)-component of velocity, and combining equations (2.12) and (2.13), a modified form of the continuity equation was obtained.

\[ \frac{\partial (h \bar{V}_x)}{\partial X} + \frac{\partial (h \bar{V}_y)}{\partial Y} + h = 0 \quad (2.14) \]

Combination of equations (2.7) and (2.14) results in an equation which may be solved for the pressure distribution:
With pressure distribution known, the average velocities may be found from equation (2.7).

Equations (2.7), (2.8), and (2.15) represent important starting equations in the generalized Hele-Shaw model from which various methods of solution have been developed. For example, Tucker, et. al. [26,29] implemented the finite element method (FEM) to solve the governing equations for the case of an isothermal Newtonian fluid. In this case, the viscosity was a constant, \( \mu \), and \( \lambda = h/2 \). Equation (2.8) could be solved analytically, giving

\[
S = \frac{h^3}{12 \mu} \tag{2.16}
\]

Combination of (2.15) and (2.16) and assuming uniform thickness resulted in Poisson's equation

\[
\frac{\partial^2 P}{\partial X^2} + \frac{\partial^2 P}{\partial Y^2} = \frac{12 \mu \dot{h}}{h^3} \tag{2.17}
\]

Equation (2.7) could then be written

\[
\dot{V}_x = -\frac{h^2}{12 \mu} \frac{\partial P}{\partial X} \quad \dot{V}_y = -\frac{h^2}{12 \mu} \frac{\partial P}{\partial Y} \tag{2.18}
\]
The boundary conditions on (2.17) were zero pressure at the free flow front, and zero normal pressure gradient at the edges of the mold. Six-node triangular elements were used in the simulation, and a single mesh specified for the initial charge location was expanded during mold filling to represent the flow progression, as shown in Figure 2.1 [26, 28]. The flow was advanced by moving the vertex nodes on the flow front distances $\vec{V}_X \cdot \Delta t$ in the X-direction and $\vec{V}_Y \cdot \Delta t$ in the Y-direction at each time step. A mesh equalization routine was used to prevent excessive element distortion during this procedure.

A finite element formulation was also developed for an isothermal power law fluid [29], where

$$\eta = m(\gamma)^{n-1}$$

(2.19)

For this case,

$$S = \frac{h^{s+2}}{(s + 2)m^s 2^{s+1}} \left[ \left( \frac{\partial P}{\partial X} \right)^2 + \left( \frac{\partial P}{\partial Y} \right)^2 \right]^{\frac{s-1}{2}}$$

(2.20)

where $s = 1/n$.

Simulation results for the isothermal power law model were compared to the isothermal Newtonian model, and very little difference was observed between the flow profiles, as seen in Figure 2.2 [28,29], although the calculated pressure distributions differed significantly. Both the Newtonian and the power-law models agreed well with molding experiments which used a single ply of SMC, as shown in Figure 2.3 [26,28]. These simulations were limited to flat parts with a single initial charge and no mold
Figure 2.1. Simulated flow progression results for an isothermal Newtonian fluid from references [26,28].
Figure 2.2. Comparison of simulation results using isothermal Newtonian and power law models from references [28,29].
Figure 2.3. Comparison of simulation results using the GHS model with molding experiments which used one ply of SMC [26,28].
inserts. A later version of the finite element simulation was developed for thin, flat parts, that could handle multiple initial charges and mold inserts [30]. Another feature of the later version was the ability to predict the appearance of knit lines in the molded part by monitoring the intersection of different portions of the flow front.

Osswald and Tucker also solved the GHS equations for an isothermal Newtonian fluid using the boundary element method (BEM) [31]. This simulation routine was also applied to flat parts of uniform thickness. For the boundary element method, the unknown variables, pressure and pressure gradient, were located only on the boundary of the charge. Therefore, the charge geometry could be represented solely by the flow front boundary, and a complete finite element mesh was not needed. This feature was attractive since mesh regeneration during program execution was not necessary. This simulation could also predict the appearance of knit lines in the molded part. The boundary element simulation results were compared to a molding experiment which used silicone oil, and good agreement was obtained between the predicted and experimental flow progressions. Simulation results using the finite element method were also given for comparison, as shown in Figure 2.4 [31]. Another experiment which used a plasticine charge also gave good agreement. It was noted, however, that for part geometries which have variable thickness, for non-Newtonian fluids, and for cases involving nonuniform heat transfer, the domain integrals which appear in the numerical formulation become more complex and a fine interior mesh would then be required. For these cases, the finite element method would be more appropriate.

As mentioned earlier, molding experiments which measured the flow progression of a single ply of SMC compared very well to simulations which were based on the isothermal Newtonian assumption. When three plies of SMC were used, however, the experimental results did not agree with the simulation, as seen in
Figure 2.4. Comparison of GHS model simulation results obtained from the FEM and BEM solutions to experimental results [31].
Figure 2.5 [29]. Since the GHS model neglects normal stress effects, the model is only applicable when both the part thickness is small, and when the extensional viscosity is small relative to the shear viscosity [29]. That is, when $\eta_e h^2/\eta_s L^2$ is much less than unity, where $\eta_e$ and $\eta_s$ are extensional and shear viscosities, respectively, and $L$ is a characteristic planar dimension. However, due to the fiber reinforcement of SMC, the extensional viscosity can be much larger than the shear viscosity [21], while for a Newtonian material the difference is not as large. Thus, for SMC the thin charge requirement is considerably more strict than for a truly Newtonian material. Castro and Griffith have also proposed a lumped parameter capable of indicating when a charge is rheologically thin [32,33].

Experimentally, then, it has been seen that the charge thickness does affect the shape of the advancing flow front. Yet it has been shown mathematically by writing the isothermal Newtonian GHS equations in dimensionless form that the predicted charge shape is independent of part thickness, mold closing speed, and viscosity [29,34]. Therefore, the GHS model cannot accurately predict the flow behavior for thick charges. Moreover, some of the basic assumptions upon which the GHS model is based may not be valid for anisotropic materials such as SMC. For example, the assumptions of negligible normal stress effects, isotropic material behavior, and no-slip at the mold surface are all questionable for SMC. In fact, even when the charge is thin and the GHS model correctly predicts the flow front progression, the predicted molding pressures are still incorrect when SMC is the molding material [32,34]. In passing, however, it is noted that during the flow of in-mold coating (IMC), which does not contain fibers, both the flow progression and the molding pressures should be correctly predicted by the GHS model.
Figure 2.5. Comparison of simulation results using the GHS model with molding experiments which used three plies of SMC [29].
A different approach to model the flow has been proposed which attempts to incorporate the actual SMC flow kinematics. As discussed earlier, Barone and Caulk observed experimentally that at typical mold closing speeds, the flow progression of SMC could be described as a uniform extension of material which slipped at the mold surface, resulting in a plug-like flow. Thus, to model the flow they assumed that the planar velocity components were independent of position in the thickness direction [35].

\[ \frac{\partial V_x}{\partial Z} = 0, \quad \frac{\partial V_y}{\partial Z} = 0 \]  
(2.21)

The momentum equation, or equilibrium condition, was written in an integrated form which neglects inertia

\[ \int_{-h/2}^{h/2} \tau_{ijj} \, dZ = 0 \]
(2.22)

Barone and Caulk used a notation where English subscripts took values 1, 2, 3, which corresponded to X, Y, Z, and Greek subscripts took values 1, 2. Partial differentiation was represented by a comma, and summation was implied over a repeated index [36]. Expanding equation (2.22), the three component equations may be written as
The components of the stress tensor were taken as

\[ \tau_{ij} = -p \delta_{ij} + \sigma_{ij} \]

or

\[
\tau = \begin{bmatrix}
-\sigma_{XX} + p & \sigma_{XY} & \sigma_{XZ} \\
\sigma_{YX} & -\sigma_{YY} + p & \sigma_{YZ} \\
\sigma_{ZX} & \sigma_{ZY} & -\sigma_{ZZ} + p
\end{bmatrix}
\]  

In these expressions, \( p \) is the hydrostatic pressure, \( \delta_{ij} \) is the Kronecker delta, and \( \sigma_{ij} \) is the extra stress term. Note that this formulation takes \( Z=X_3=0 \) at the midpoint of the charge thickness. Combining (2.23) and (2.24) for the X-component gives:
\[ \int_{-h/2}^{h/2} \left[ -\frac{\partial P}{\partial X} + \frac{\partial \sigma_{XX}}{\partial X} + \frac{\partial \sigma_{XY}}{\partial Y} + \frac{\partial \sigma_{XZ}}{\partial Z} \right] dZ = 0 \]

\[ \int_{-h/2}^{h/2} \left( -\frac{\partial P}{\partial X} \right) dZ + \int_{-h/2}^{h/2} \left( \frac{\partial \sigma_{XX}}{\partial X} \right) dZ + \int_{-h/2}^{h/2} \left( \frac{\partial \sigma_{XY}}{\partial Y} \right) dZ + \int_{-h/2}^{h/2} \left( \frac{\partial \sigma_{XZ}}{\partial Z} \right) dZ = 0 \]

\[ -\frac{\partial}{\partial X} \int_{-h/2}^{h/2} p dZ + \frac{\partial}{\partial X} \int_{-h/2}^{h/2} \sigma_{XX} dZ + \frac{\partial}{\partial Y} \int_{-h/2}^{h/2} \sigma_{XY} dZ + \frac{\partial}{\partial Z} \int_{-h/2}^{h/2} \sigma_{XZ} dZ = 0 \]

The last term on the left hand side is

\[ \left. \frac{\partial}{\partial Z} \int_{-h/2}^{h/2} \sigma_{XZ} dZ \right|_{-h/2}^{h/2} = \left. \sigma_{XZ} \right|_{h/2} - \left. \sigma_{XZ} \right|_{-h/2} \]

\[ = \left. \sigma_{XZ} \right|_{h/2} - \left( -\left. \sigma_{XZ} \right|_{h/2} \right) = 2\left. \sigma_{XZ} \right|_{Z=h/2} \]

Using the following definitions,

\[ P = \int_{-h/2}^{h/2} p dZ, \quad n_{\alpha\beta} = \int_{-h/2}^{h/2} \sigma_{\alpha\beta} dZ, \quad f_{\alpha} = \left. \sigma_{\alpha Z} \right|_{Z=h/2} \]

equation (2.25) becomes

\[ -\frac{\partial P}{\partial X} + \frac{\partial n_{XX}}{\partial X} + \frac{\partial n_{XY}}{\partial Y} + 2f_{X} = 0 \]
Similarly, the Y-component may be written

\[ -\frac{\partial P}{\partial Y} + \frac{\partial n_{YX}}{\partial X} + \frac{\partial n_{YY}}{\partial Y} + 2f_Y = 0 \]  

(2.29)

The Z-component equation resulting from combination of (2.23) and (2.24) is satisfied by symmetry and need not be considered. Barone and Caulk wrote equations (2.28) and (2.29) compactly as

\[ -P,\alpha + n_{\alpha\beta} + 2f_\alpha = 0 \]  

(2.30)

Here,

- \( P \) = pressure resultant through the cavity thickness
- \( n_{\alpha\beta} \) = planar stress resultant which describes the extensional response of the material
- \( f_\alpha \) = friction force per unit area between the SMC and the mold surface

If it is assumed that the pressure is independent of \( Z \), the normal pressure \( p \) and the pressure resultant \( P \) are related by \( p = P/h \). Note that \( p \) is the pressure of the material on the mold surface [36,37]. The planar stress resultant and the friction force must be obtained from constitutive relations.

Barone and Caulk suggested that when the material is assumed to be isotropic in the X-Y plane, neglecting any fiber orientation, the following constitutive relation for the planar stress resultant may be used.
\[ n_{\alpha\beta} = 2\mu h \left( d_{n\eta} \delta_{\alpha\beta} + d_{\alpha\beta} \right) \quad (2.31) \]

where

\[ d_{\alpha\beta} = \frac{1}{2} \left( V_{\alpha\beta} + V_{\beta\alpha} \right) \quad (2.32) \]

Here, \( \mu \) is the effective in-plane viscosity and \( d_{\alpha\beta} \) is the planar rate of deformation. The form of equation (2.31) is the same as that of a constitutive relation for an incompressible elastic membrane. For illustration, equations (2.31) and (2.32) may be manipulated for use in equation (2.28) for the X-component as follows. The needed rates of deformation are given by

\[
\begin{align*}
  d_{XX} &= \frac{1}{2} \left( \frac{\partial V_X}{\partial X} + \frac{\partial V_X}{\partial X} \right) = \frac{\partial V_X}{\partial X} \\
  d_{YY} &= \frac{1}{2} \left( \frac{\partial V_Y}{\partial Y} + \frac{\partial V_Y}{\partial Y} \right) = \frac{\partial V_Y}{\partial Y} \\
  d_{XY} &= \frac{1}{2} \left( \frac{\partial V_X}{\partial Y} + \frac{\partial V_Y}{\partial X} \right)
\end{align*}
\]

These may be used to obtain the planar stress resultants as

\[
\begin{align*}
  n_{XX} &= 2\mu h (d_{XX} + d_{YY} + d_{XX}) = 2\mu h (2d_{XX} + d_{YY}) = \mu h \left[ \frac{\partial V_X}{\partial X} + 2 \frac{\partial V_Y}{\partial Y} \right] \\
  (2.33)
\end{align*}
\]
\[ n_{XY} = 2\mu h d_{XY} = 2\mu h \left[ \frac{1}{2} \left( \frac{\partial V_X}{\partial Y} + \frac{\partial V_Y}{\partial X} \right) \right] = \mu h \left[ \frac{\partial V_X}{\partial Y} + \frac{\partial V_Y}{\partial X} \right]. \]  

(2.34)

It follows that

\[ \frac{\partial n_{XX}}{\partial X} = \mu h \left[ 4 \frac{\partial^2 V_X}{\partial X^2} + 2 \frac{\partial^2 V_Y}{\partial X \partial Y} \right]. \]  

(2.35)

\[ \frac{\partial n_{XY}}{\partial Y} = \mu h \left[ \frac{\partial^2 V_X}{\partial Y^2} + \frac{\partial^2 V_Y}{\partial X \partial Y} \right]. \]  

(2.36)

Then, for the X-component,

\[ n_{X \beta \beta} = \frac{\partial n_{XX}}{\partial X} + \frac{\partial n_{XY}}{\partial Y} = \mu h \left[ 4 \frac{\partial^2 V_X}{\partial X^2} + \frac{\partial^2 V_X}{\partial Y^2} + 3 \frac{\partial^2 V_Y}{\partial X \partial Y} \right]. \]  

(2.37)

Recall from continuity

\[ \frac{\partial V_X}{\partial X} + \frac{\partial V_Y}{\partial Y} + \frac{h}{h} = 0. \]  

(2.38)

Differentiating the continuity expression with respect to X gives

\[ \frac{\partial^2 V_X}{\partial X^2} + \frac{\partial^2 V_Y}{\partial X \partial Y} = 0. \]  

(2.39)
Combining (2.37) and (2.39),

$$n_{x\beta\gamma} = \mu h \left[ \frac{\partial^2 V_X}{\partial X^2} + \frac{\partial^2 V_X}{\partial Y^2} \right].$$

(2.40)

Equation (2.40) may be used in equation (2.28). However, a constitutive relation for the friction term must also be specified.

Three mechanisms for the frictional response were described, each based on different assumptions [35]. The form of the constitutive relation for the friction force depends on the assumed mechanism. If a constant friction response is used, the governing equation predicts that for a given mold closing speed any two charges which have the same initial shape will have identical flow progressions, regardless of their heights [35]. Since this contradicts observed experimental results, the constant friction mechanism would not be an appropriate choice. A hydrodynamic friction mechanism, on the other hand, applies if the assumption is made that a very thin resin-rich layer exists between the mold surface and the central core of the SMC. In this case, a lubricating effect may occur and the friction force is taken to be proportional to the velocity. This results in a solution which is not independent of gap height, and therefore represents a potentially realistic solution to the problem. A third mechanism, the Coulomb frictional response, would be appropriate when there is direct fiber-to-metal contact, with negligible resin lubrication. Then, the friction force is proportional to the normal stress at the cavity surface, and the solution of the governing equations is again dependent on the cavity thickness.

To gain insight into the relative importance of friction and extensional resistance, dimensionless variables were defined and the governing equations recast in terms of
these variables [35]. When the hydrodynamic and the Coulomb friction mechanisms were used, it was shown that the friction term becomes more important as the flow progresses (as the thickness decreases), although this behavior is more pronounced for the hydrodynamic friction assumption. The best agreement with experimental results was obtained using the hydrodynamic friction assumption, and therefore it was used as a basis for the constitutive relation for the friction force. The friction force $f_\alpha$ could then be written as

$$f_\alpha = -KV_\alpha,$$  \hspace{1cm} (2.41)

where $K$ may depend on the mold surface temperature.

When equations (2.40) and (2.41) are inserted in (2.28) for the $X$-component, the result is

$$-\frac{\partial P}{\partial X} + \mu h \left[ \frac{\partial^2 V_X}{\partial X^2} + \frac{\partial^2 V_X}{\partial Y^2} \right] - 2KV_X = 0.$$  \hspace{1cm} (2.42)

For the $Y$-component,

$$-\frac{\partial P}{\partial Y} + \mu h \left[ \frac{\partial^2 V_Y}{\partial X^2} + \frac{\partial^2 V_Y}{\partial Y^2} \right] - 2KV_Y = 0.$$  \hspace{1cm} (2.43)

Alternatively, both equations may be represented by

$$-P_\alpha + \mu h V_{\alpha,\beta\beta} - 2KV_\alpha = 0.$$  \hspace{1cm} (2.44)
To obtain the velocity field, equation (2.44) is solved with the modified continuity equation (2.38). Note that equation (2.38) is similar to equation (2.14) derived earlier for the GHS model, only now the velocities which appear in the equation are the planar "plug" velocities. Appropriate boundary conditions must also be specified. Note that this model consists of three equations: the modified continuity equation, and the two component equations given by equation (2.44). Three variables, $V_x$, $V_y$, and $P$ are all unknowns which must be obtained, and therefore the solution procedure is more involved than was the case for the GHS model, which had one variable $P$ in the Poisson equation.

The boundary element method was used by Barone and Osswald to simulate the flow of SMC using this model under isothermal conditions [38]. The simulated flow front progressions agreed well with experimental results for both thin and thick charges. In their numerical formulation the variables $\mu$, $K$, and $h$ were combined in a single parameter,

$$\psi = \sqrt{\frac{2K}{\mu h}} L,$$  \hspace{1cm} (2.45)

where $L$ was a characteristic planar dimension of the initial charge. The value of $\psi$ was chosen to match the experimental results, since actual values for the in-plane viscosity and friction coefficient were not available.

The two resistances to flow in equation (2.44), material extension and friction, are responsible for any changes in shape which the charge experiences during molding. Without these resistances, the flow progression would be equibiaxial and homogeneous. As mentioned earlier, the model predicts that friction becomes more important as the
thickness decreases. Therefore, for parts which are very thin, the friction effect dominates and the material resistance may be neglected as an approximation. The governing equation then reduces to

\[ P_{,\alpha} + 2KV_{,\alpha} = 0 \quad \text{or} \]

\[ \nabla P + 2KV = 0. \]  

The modified continuity equation remains as

\[ \nabla \cdot \mathbf{V} + \frac{h}{h} = 0. \]  

Combining the divergence of equation (2.46) with equation (2.47) gives the governing equation for pressure, which is the Poisson equation [35,36].

\[ \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} = \frac{2Kh}{h} \]  

The velocity field is written in terms of the pressure gradient from equation (2.46):

\[ V_x = -\frac{1}{2K} \frac{\partial P}{\partial x}, \quad V_y = -\frac{1}{2K} \frac{\partial P}{\partial y}. \]
It should be remembered that in Barone and Caulk's notation, $P = \rho p$ represents the pressure resultant, where $p$ is the hydrostatic pressure. The equations have also appeared in the literature in terms of the hydrostatic pressure. These equations, obtained in the limit of thin charge from Barone and Caulk's model, are analogous to equations (2.17) and (2.18), obtained from the isothermal GHS model for a Newtonian material. It is noted that if the friction coefficient $K$ is taken as $6\mu / h^2$, then the thin cavity approximation of Barone and Caulk becomes formally equivalent to the generalized Hele-Shaw model for an isotropic Newtonian fluid [35].

Short shot experiments were compared to predictions using the thin charge flow model of Barone and Caulk, and good agreement was obtained for a single layer charge using the hydrodynamic friction assumption, as shown in Figure 2.6 [35]. When a thicker charge using four layers was molded, the agreement was not as good.

The values of friction coefficient, gap height, and mold closing speed which appear in equations (2.48) and (2.49) affect the velocity field only as a shift in the time scale. The shape of the charge as the flow progresses does not depend on such variables for the thin cavity approximation of Barone and Caulk, as was the case for the GHS model [35]. The times at which the charge assumes these shapes do depend on these variables, as do the pressures.

The models described above which are applicable only to thin charges are useful since most exterior automotive parts are rheologically thin [32]. Recently Castro, Millward, and Straus demonstrated the applicability of thin charge models to design SMC charge locations to reduce knit line formation when molding an automotive hood panel [39]. Injection ports for in-mold coating flow may also be located using these models to avoid trapped air in the mold cavity.
Figure 2.6. Comparison of prediction using the thin charge model of Barone and Caulk to molding results [35].
Another numerical technique, the control volume/finite element method (CV/FEM), was employed recently by Osswald and Tucker to simulate the flow in more complex geometries [34]. Their simulation used both four-node quadrilateral elements and three-node triangular elements, and could use either the generalized Hele-Shaw formulation or the thin-charge limit of Barone and Caulk's model. The governing equations were solved using a Galerkin finite element method, and the movement of the charge was followed using a network of control volumes.

The control volume/finite element method used by Osswald and Tucker possessed several important features [34]. First, both a global coordinate system and a local element coordinate system were employed to accommodate simulations of non-planar geometry. Second, a fill factor was associated with each control volume, and was updated at each time step after computing the flow rate of material into the control volume. Third, variable time steps were used to allow only one control volume to become filled during each time step. Finally, the usual boundary conditions of zero pressure at the flow front and zero normal pressure gradient at the mold boundary were used.

In the CV/FEM simulation, the overall mold closing direction was defined by a vector in the global coordinate system [34]. For curved (non-planar) geometries, it was also necessary to compute a local mold closing speed for each element since the closing speed in curved regions was not equal to the overall closing speed unless the plane of the element was normal to the closing direction. The following equation was used to compute the local element closing speed:

\[ s_e = s_m \left| \vec{c} \cdot \vec{n}_e \right| \]  

(2.50)
where \( s_e \) = local element mold closing speed
\( s_m \) = overall mold closing speed
\( \vec{c} \) = overall mold closing direction vector
\( \vec{n}_e \) = unit vector normal to the plane of element e.

The desired final thicknesses for all elements were specified in an input file. Then, the upper mold was raised or "backed up" from this final position to the point at which contact was first made with the upper surface of the charge. The simulation began from this point, lowering the mold at each time step using the local element closing speeds given by equation (2.50). The input file was also used to define the location of the initial charge by specifying the nodes which made up the charge.

The simulation could handle multi-thickness charges by assigning fill factors of less than one to nodes lying in regions of the charge which were thin. Since the surface area of a control volume in the local X-Y plane was constant, the fill factor for a control volume in the thin charge region was simply the ratio of charge height to mold height. A value of zero pressure was also assigned to these nodes initially. No flow occurred in these control volumes until the top mold touched the upper surface of the charge, or until adjacent control volumes became filled.

An experiment which used a multi-thickness initial charge in a flat mold was conducted to test the ability to handle such charges. Some disagreement with simulation results occurred because movement of the thin part of the charge did occur prior to contact with the upper mold, while the program predicted material flow only in regions that were completely filled.

Good agreement was obtained between simulated and experimental flow front progressions for an automotive hood, as shown in Figure 2.7 [34]. Experiments were
Figure 2.7. Comparison of simulation results obtained from the CV/FEM solution to experimental results from an automotive hood geometry [34].
also conducted on a truck air deflector. The first experiment, shown in Figure 2.8(a) [34], used a relatively thick initial charge which was placed such that initially flow could occur mainly in only one direction. It is seen that the predicted flow progression agreed well with the experimental results in spite of the thick charge, because the simulation must obey conservation of mass. The second experiment, shown in Figure 2.8(b) [34], also used a relatively thick initial charge, but this time flow was not restricted to one direction. Therefore, the predicted and experimental results did not agree in the early stages of flow because the charge was too thick for a thin charge model to apply, and mass conservation did not help due to the multi-directional nature of the flow. During the later stages of the flow, when the charge had become thinner, the model did a reasonable job of predicting the flow.
Figure 2.8. Comparison of simulation results obtained from the CV/FEM solution to experimental results from a truck air deflector geometry [34].
2.2 Volumetric Changes During Polymer Processing

The fact that polymers may expand or contract during processing is a problem which has been widely recognized. For injection molded thermoplastic parts, for example, the material shrinkage which occurs during cooling can adversely effect the desired tolerances of the final product. To avoid this problem, the mold cavity dimensions must be specified to account for the thermal volume changes. A rough estimate of the dimension changes to be expected for very simple geometries may be obtained from the shrinkage rates given in material data sheets [40]. For design purposes, however, this simple method may be inadequate if the part geometry is complex and differs from the geometry used to measure the shrinkage rate.

Miller [40] discussed relevant factors which effect part shrinkage for thermoplastic injection molded parts, including degree of crystallinity, reinforcing fibers, part geometry, gate location and size, cooling system, and processing conditions, among others. Several computer packages, in various stages of development, were mentioned which have the capability to predict part shrinkage as the molten polymer cools to a solid. One package can automatically calculate the necessary mold cavity dimensions to account for the predicted volumetric changes during molding. However, the ability to model the effects of reinforcing fibers on the shrinkage is still an area under development, and the ability to predict shrinkage due to a polymerization reaction was not mentioned.

Salloum, et. al. [41], developed a model for predicting the shrinkage in a high density polyethylene resin of disk geometry. Of the eight molding parameters studied, five were found to have a significant influence on shrinkage during injection molding. These included holding pressure, cooling time, mold temperature, maximum injection
pressure, and melt temperature. For each of these parameters, a second order polynomial was used to capture the effect of the parameter on the mold shrinkage. Then, the total radial shrinkage of the disk was computed as a summation of the contributions from each parameter.

When thermosetting systems are processed, the shrinkage which occurs due to the polymerization reaction further complicates the situation. If the material contains fibers for reinforcement and the part is designed with a substructure such as a reinforcing rib, the net result of the volume changes which occur may be a poor surface appearance. This surface deformation, known as rib readout, occurs on the cosmetic side of the part opposite to the location of the rib. This has resulted in the practice of using two automotive body panels bonded together to manufacture a single part; one panel with ribs for strength, and a second panel without ribs for the cosmetic surface. Thus, a better understanding of the volume changes which occur during thermoset polymer processing may save the time and the expense associated with trial and error mold design and unnecessary manufacturing steps.

Some experimental studies have been conducted to gain insight into the volumetric changes which occur during thermoset processing. Epoxy resins, for example, are an important class of thermosetting materials used for the encapsulation of electronic circuits and for protective coatings [42]. For formulations which exhibit a large coefficient of linear thermal expansion, the materials can contract and the protective coatings may crack as a result. The effect of various fillers and additives on the thermal expansion and cure shrinkage of epoxy resins is therefore an important material design consideration. In a recent study in this area [42], the coefficients of linear thermal expansion of cured epoxy samples were obtained by measuring the elongation during heating to 150° C. The specific volumes versus temperature of the cured samples were
then calculated using the coefficients of linear thermal expansion. The specific volumes versus temperature of uncured liquid resins were measured using a quartz dilatometer. Some of the liquid samples did not contain hardeners, while others did not contain accelerators in order to prevent curing during heating. Then, the percent volume shrinkage due to polymerization versus temperature was calculated by subtracting the specific volume of cured resin from that of the liquid at each temperature, and dividing by the liquid specific volume. The result of this procedure for five resin samples is shown in Figure 2.9 [41]. It was found that cure shrinkage obtained in this manner decreased with decreasing cure temperature, and via the addition of Al₂O₃ filler to the resin.

Kia [43] studied many of the parameters which contribute to rib readout, or sink marks on SMC parts. For each of the parameters identified, the tendency to form a deformation in the form of a "hill" or a depressed "valley" was discussed. The effect of in-mold coating and paint application on surface quality was also examined. Parameters which were recognized to cause valleys on the surface included polymerization shrinkage of the resin and thermal contraction of the resin-rich region which is formed in the area above the rib. In contrast, compressibility of the resin-rich region and thermal contraction of the glass-rich region were identified as factors that result in hills on the surface. It was noted that even when the presence of an LPA causes an overall net increase in volume (in contrast to a formulation with no LPA, as shown in Figure 2.10 [43,44]), rib readout may still occur. This may be due to the thermal contraction which occurs as the material cools from the peak exotherm temperature back to the mold temperature prior to complete solidification. It was suggested that to further reduce readout due to polymerization, all volume shrinkage prior to solidification should be eliminated, as illustrated schematically for an "ideal" material in Figure 2.11 [43].
Figure 2.9. Percent polymerization shrinkage versus temperature for five epoxy resins [41].
Figure 2.10. Comparison of volume changes for an SMC resin (a) with LPA and (b) without LPA [43,44].
Figure 2.11. Elimination of all shrinkage prior to complete polymerization to reduce readout for an ideal material [43].
Mitani, et al. [45] combined data from molding experiments, dilatometer measurements, and morphological analysis to study the mechanisms of four different LPA's and the corresponding amounts of shrinkage. During the molding experiments, the percent shrinkage at the end of molding was measured for several paste molding compounds. The degree of polyester reactivity and the type of LPA were varied for each compound, and the effect on mold shrinkage was recorded. The results showed that mold shrinkage was lowest for compounds with the higher polyester reactivities, while greater mold shrinkage occurred for compounds of lower reactivity.

During the dilatometer experiments [45], volume changes were recorded for samples which had not been thickened with MgO using different LPA's. Each system showed an initial thermal expansion during heating to 140° C. Due to the presence of LPA, after thermal expansion some samples showed an additional expansion while others did not expand further. This behavior was attributed to differences in the polyester reactivities of the samples. Once the maximum volume of each sample was reached, a decrease in volume occurred which was attributed to polymerization shrinkage. The samples were then cooled to room temperature and a further thermal shrinkage occurred. Depending on the polyester reactivity and the type of LPA, the net volume change of the samples after cooling varied from more than 10% expansion, to a slight net shrinkage.

It was noted [45] that a linear relation resulted between final volume change using dilatometry and the level of final mold shrinkage, in spite of the fact that the former experiments were conducted at relatively low pressure (53 psi), while the latter experiments were performed under high pressures (967 psi). Samples with a large final volume expansion during dilatometry showed a smaller amount of mold shrinkage, and vice versa. However, in some cases the measured volume changes during molding
were the same regardless of LPA type, while the dilatometer measurements showed large differences. Thus, in some cases the high molding pressure was able to suppress the formation of microcracks and phase separation, which decreased the ability of the LPA to cause a large volume expansion. Nevertheless, the dilatometer results were able to provide a qualitative indication of the volume changes to be expected during molding.

The effect of MgO and glass fibers on the mold shrinkage was also reported, as shown in Figure 2.12 [45]. It was found that glass fibers always tended to reduce mold shrinkage, while magnesium oxide seemed to increase shrinkage with all but one LPA system studied. The combined effect was difficult to predict in general since the two factors have opposing effects.

As part of a larger study on monitoring the cure of SMC, Castro and Straus [46] used linear variable displacement transducers to follow the volumetric changes which occurred during SMC molding. After mold closure, the LVDT measurements indicated an initial expansion due to heat transfer from the hot mold walls to the SMC, which was initially at room temperature. Then, once the polymerization started, a shrinkage occurred due to the chemical reaction, followed by another contraction due to cooling from the maximum exotherm temperature to the mold wall temperature. It was found that if the mold was opened prior to the end of the polymerization shrinkage period, part blistering occurred. If the mold was opened just after the final dip in the LVDT motion, however, the part did not blister.

Kau [47] used a cylindrical cure reactor to monitor the stages of the curing process using SMC paste. This heated reactor was equipped with a plunger and was positioned between the upper and lower halves of a mold press. A thermocouple was placed along the centerline of the reactor, a pressure transducer was positioned at the tip of the plunger, and an LVDT recorded the displacement between the mold walls. Each
Figure 2.12. Effect of MgO and glass fibers on mold shrinkage [45].
of these measurements did a good job of showing the various stages which occurred during the curing experiment, as shown in Figure 2.13 [47]. In some cases, however, these measurements may have been out of step with each other with respect to defining the boundaries between the stages. This was due to the fact that the thermocouple measurement was essentially a point value, while the pressure and LVDT measurements were related to the sample as a whole. Furthermore, the temperature and conversion profiles in the sample were nonuniform and therefore location dependent. The measurements agreed fairly well when identifying the end of the reaction, however.

The thermocouple response [47] showed an initial gradual temperature rise due to conduction from the mold walls, followed by a more rapid increase due to chemical reaction. The temperature then slowly decreased from the peak exotherm temperature down to the mold temperature. It was found that plotting the derivative of the temperature curve, $dT/dt$, versus time made identification of the major stages easier.

The LVDT response [47] showed the expected initial thermal expansion due to heat conduction from the mold, followed first by an overall shrinkage due to the curing reaction, and then by a more gradual shrinkage due to thermal contraction of the part as it cooled. These results are in qualitative agreement with the study by Castro and Straus [46]. The pressure measurement was found to coincide with the volume changes, although the mold press dynamics and mechanical interactions between the press and the sample also influenced the pressure response.

An explanation was suggested by Kau [47] to account for the volume changes which occurred. Assuming the density $\rho$ was a function of radial position and time, the total volume change of the paste could be written as a summation of the volume changes of elements in the radial direction:
Figure 2.13. Temperature, mold displacement, and pressure measurements during curing in a cylindrical reactor [47].
\[ \Delta V = 2\pi f L \int_0^a \frac{(\rho_0 - \rho)}{\rho} \, r \, dr \]

(2.51)

where

- \( V \) = total volume
- \( f \) = multiplier related to material compressibility
- \( L \) = sample length
- \( a \) = radius of cure reactor.

This equation relates the total volume of the paste, \( V \), to the specific volume, \( \nu = 1/r \).

The overall volume change could also be written qualitatively as a combination of the stages observed during cure:

\[
dV = \left( \frac{\partial V}{\partial P} \right)_\text{P} \, dP + \left( \frac{\partial V}{\partial T} \right)_\text{T} \, dT + \left( \frac{\partial V}{\partial X} \right)_X \, dX + \left( \frac{\partial V}{\partial T} \right)_\text{T} \, dT \quad (2.52)
\]

The first term on the right hand side, the bulk modulus, was considered negligible in the analysis. The second and fourth terms are thermal expansion coefficients, and had different values depending on whether the sample was in the pre-cure or the post-cure stage. This difference was due to the phase change from liquid to solid and to the effect of low profile agents. An estimate for the third term, the volume change due to reaction, was made by assuming the reaction takes place in a compressible state with no correction made for the temperature effect. No attempt was made to actually implement this model to solve for the material volume changes.

Osswald, et al. [48] studied the effect of shrinkage during SMC processing on part warpage. When SMC parts are molded, the material properties may be nonuniform due to an assymetric temperature profile during curing and flow induced fiber orientation. It was noted that when the temperature and conversion profiles across the
part thickness are asymmetric, the amount of contraction or expansion of neighboring "layers" in the thickness direction may be different [48]. This can cause some layers to stretch while others are compressed in the planar direction. This situation can cause strain gradients and residual stresses in the material which in turn can cause warpage.

A model was proposed to predict the warpage of thin SMC parts by solving a stress analysis equation based on the principle of virtual displacements [48]. In this analysis, three-dimensional rectangular shaped elements were used which had twenty nodes each. The working equation was applied to each element and the resulting system of equations was solved. Since this model contained a material tensor which depended on the degree of cure, and since a term for the thermal strain was present, the stress analysis equation was coupled with the energy equation.

The energy equation was simplified [48] by including terms only for heat conduction in the thickness direction, a heat generation term due to chemical reaction, and the unsteady state term. The heat generation term was represented by an empirical kinetic model [49]. For heat transfer calculations, nodes were set up in the thickness direction and the finite difference method was used to solve the energy equation at each vertical node location. The temperatures of the upper and lower SMC boundaries were specified to be equal to the mold wall temperature.

Numerical simulations were performed which computed the deflections of three corners of an SMC plate, and the results agreed well with experimental deflections [48]. Simulations were then conducted to show the effect of various material and process variables on the warpage. For example, when the mold closing speed was reduced, a longer asymmetric heating period occurred and the predicted deflection increased. It was found that increasing the fiber content to 50% by volume decreased the predicted warpage due to the added stiffness caused by the fibers. When a charge preheated to
$50^\circ$ C was simulated, the level of warpage decreased due to smaller temperature gradients during heating. Finally, when the simulated mold temperature was increased, the warpage increased due to larger temperature gradients.

The author noted [48] that the three dimensional finite element scheme employed would be very difficult to apply when simulating complex geometries. For these cases "shell-type" elements, as used in many mold filling routines, were recommended.
3.1 Control Volume/Finite Element Method for Compression Molding

A control volume/finite element technique [50] was used to model the flow front progression of SMC during compression molding and to calculate the pressure and velocity fields in the charge. This approach has been used successfully by Tucker and Osswald [34]. The starting equations for the numerical formulation may be based on either the generalized Hele-Shaw model or the thin-charge model proposed by Barone and Caulk, described in the previous chapter. As discussed earlier, these models are applicable when the thickness of the part is small in comparison to the lateral dimensions. The GHS equations are

\[
\bar{V}_X = -\frac{S}{h} \frac{\partial P}{\partial X}, \quad \bar{V}_Y = -\frac{S}{h} \frac{\partial P}{\partial Y}
\]

where

\[
S = \int_0^h \frac{(Z - \lambda)^2}{\eta} dZ
\]

\[ (3.1) \]

\[ (3.2) \]
and the applicable equations proposed by Barone and Caulk are

\[ V_x = -\frac{h}{2K} \frac{\partial P}{\partial x}, \quad V_y = -\frac{h}{2K} \frac{\partial P}{\partial y}. \] (3.3)

To construct the control volumes which this simulation technique uses, the part geometry is divided into a mesh of three-node triangular elements. Each vertex of each triangle is called a node. A typical control volume is shown in relation to its constituent elements in Figure 3.1. The shaded region is the control volume which is associated with the node in the middle of the figure. The boundary of this region is formed by connecting the centroid of each triangular element with the midpoints of the sides of each element. A cut-away view of one of the elements is given in Figure 3.2. Each vertex or node in the mesh is associated with a control volume formed in this manner, although only one control volume is explicitly illustrated in Figure 3.1 for clarity. Note that the thicknesses of the control volumes will decrease with each time step, due to the closing upper mold. Mesh generation and post processing of results were accomplished using I-DEAS, developed by SDRC (Structural Dynamics Research Corp., Milford, Ohio).

The flow analysis for SMC begins with a mass balance on a control volume. The flux of material through the surface which bounds the control volume is equal to the induced flow due to mold closure. That is, the volumetric flow rate of material through the surface is equal to the rate at which the closing upper mold displaces volume previously occupied by SMC. The mass balance may therefore be written as

\[ \iiint_{\Omega} \vec{V} \cdot \vec{n} \, d\Omega = -\sum_{c=1}^{m} \frac{\dot{m}_c A_c}{3}, \] (3.4)
Figure 3.1 The top view of a control volume in relation to the surrounding triangular elements.

Figure 3.2 An element and a portion of the control volume shown above in Figure 3.1.
\[ \int \int_{\Omega} \left( V_x \hat{i} + V_y \hat{j} + V_z \hat{k} \right) \cdot \vec{n} \, d\Omega = -\sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3} \]  

(3.5)

where \( \Omega \) = control volume surface

\( \vec{n} \) = unit vector normal to the surface

\( \dot{h}_e \) = mold closing speed associated with element \( e \)

\( A_e \) = surface area of element \( e \) in the X-Y plane

\( m \) = number of elements in a control volume

By definition \([51]\), the left hand side of equation (3.5) is the flux of material through the surface, \( \Omega \). Since there is no flow through the upper and lower control volume surfaces which are in contact with the mold, the contribution of these surfaces to the integral in equation (3.5) is zero. Only surfaces of the type labeled "A" in Figure 3.2 need be considered. The right hand side of equation (3.5) is the rate of volume displacement by the closing mold.

The unit normal vector is computed for each area \( A \) on the control volume surface. For example, the unit normal vector associated with area \( A \) in Figure 3.2 is computed as

\[
\vec{n}_A = \frac{(y_b - y_c, x_c - x_b)}{\sqrt{(y_b - y_c)^2 + (x_c - x_b)^2}}
\]

(3.6)

where \((x_b, y_b)\) and \((x_c, y_c)\) are the coordinates at points \( b \) and \( c \), respectively. The orientations of the normal vectors used in this work are directed inward, towards the
inside of the control volume. With this convention, the integral in equation (3.5) is negative when the outward flux of material due to compression is calculated. To be consistent with the chosen convention, a negative sign is written on the right hand side of equation (3.5).

Taking the indicated scalar product, equation (3.5) may be written as

\[
\int \int \Omega \left[ n_x \ n_y \ n_z \right] \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \, d\Omega = - \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

(3.7)

The thin charge models given by equations (3.1) and (3.3) consider the pressure gradient and velocity in the thickness direction (the Z-direction) to be negligible relative to the pressure gradients and velocities in the X and Y-directions. For this case, \( V_Z \) is taken to be zero, and equation (3.7) may be written as

\[
\int \int \Omega \left[ n_x \ n_y \right] \begin{bmatrix} V_x \\ V_y \end{bmatrix} \, d\Omega = - \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

(3.8)

Inserting the GHS model, equation (3.1), into equation (3.8) gives

\[
\int \int \left( \frac{S}{h} \right) \left[ n_x \ n_y \right] \begin{bmatrix} \frac{\partial P}{\partial X} \\ \frac{\partial P}{\partial Y} \end{bmatrix} \, d\Omega = \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

(3.9)
Since the integrand in the above equation is independent of \( Z \), the surface integral over \( d\Omega \) may be written as the line integral along the boundary multiplied by the element thickness. For example, for area \( A \) in Figure 3.2 the line integral would be along segment \( cb \). This gives

\[
\int_C S \begin{bmatrix} n_X & n_Y \end{bmatrix} \begin{bmatrix} \frac{\partial P}{\partial X} \\ \frac{\partial P}{\partial Y} \end{bmatrix} dL = \sum_{e=1}^{m} \frac{h_e A_e}{3}
\]  

(3.10)

A linear interpolation function \( N_i \) for a three-node triangular element is used for the pressure [52].

\[
P = \sum_{i=1}^{3} P_i N_i
\]  

(3.11)

where

\[
N_i = \frac{(a_i + b_i X + c_i Y)}{2A_e}
\]  

(3.12)

\[
P_i = \text{pressure at nodes located at element vertices}
\]

\[
a_i = X_j Y_k - X_k Y_j
\]

\[
b_i = Y_j - Y_k
\]

\[
c_i = X_k - X_j
\]

\[
X = X \text{ coordinate at element vertex } i, j, \text{ or } k
\]

\[
Y = Y \text{ coordinate at element vertex } i, j, \text{ or } k
\]

\[
i \neq j \neq k
\]

Here, \( i, j, \text{ and } k \) permute in a natural order. Differentiating equation (3.11) with respect to \( X \) and \( Y \) results in the following expressions for the pressure gradients.
or

\[
\begin{bmatrix}
\frac{\partial P}{\partial X} \\
\frac{\partial P}{\partial Y}
\end{bmatrix}
= \frac{1}{2A_e}
\begin{bmatrix}
b_1 & b_2 & b_3 \\
c_1 & c_2 & c_3
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix}
\]

\hspace{1cm} (3.14)

Inserting equation (3.14) into (3.10) yields

\[
\int_C \frac{S}{2A_e} \begin{bmatrix}
n_X & n_Y
\end{bmatrix}
\begin{bmatrix}
b_1 & b_2 & b_3 \\
c_1 & c_2 & c_3
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix}
\, dL = \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

\hspace{1cm} (3.15)

To put this equation in a form which can be solved numerically, the left hand side may be written for a control volume as a summation of the contributions from each element associated with the control volume. For \( m \) elements,

\[
\sum_{e=1}^{m} \left\{ \int_{L_e} \frac{S}{2A_e} \begin{bmatrix}
n_X & n_Y
\end{bmatrix}
\begin{bmatrix}
b_1 & b_2 & b_3 \\
c_1 & c_2 & c_3
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix}
\, dL_e \right\} = \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

\hspace{1cm} (3.16)

It is seen in Figures 3.1 and 3.2 that the control volume boundary in each element consists of two portions. The components of the unit normal vector \( n_X \) and \( n_Y \) for segment ac are different than the components for segment bc. All other terms in
equation (3.16) do not depend on $L_e$. Therefore, the integral is split into two parts for each element:

\[
\sum_{e=1}^{m} \left[ \frac{S}{2A_e} \left( L_a \{n_X \ n_Y\}_{ac} + L_{bc} \{n_X \ n_Y\}_{bc} \right) \right] = \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

Expansion and rearrangement of equation (3.17) results in

\[
\sum_{e=1}^{m} \left\{ \left( b_1 F_X + c_1 F_Y \right) P_1 + \left( b_2 F_X + c_2 F_Y \right) P_2 + \left( b_3 F_X + c_3 F_Y \right) P_3 \right\} = \sum_{e=1}^{m} \frac{\dot{h}_e A_e}{3}
\]

where

\[
F_X = \frac{S}{2A_e} \left( L_a n_{Xac} + L_{bc} n_{Xbc} \right)
\]

\[
F_Y = \frac{S}{2A_e} \left( L_a n_{Yac} + L_{bc} n_{Ybc} \right)
\]

If the thin charge equations from Barone and Caulk's flow model had been used, the expressions for $F_X$ and $F_Y$ become
In the numerical scheme, the pressure field is solved by writing equations (3.18) and (3.19) for all control volumes which are full, resulting in a set of linear algebraic equations. The set of equations is solved for the nodal pressures by iteration using a relaxation factor. Iteration continues until the normalized difference between successive pressure iterations at each node converges to within an acceptable tolerance.

The boundary conditions used by the program are zero pressure gradient normal to the mold boundary, which implies no flow in this direction, and zero pressure at the flow front. Thus, control volumes located on the flow front are assigned zero pressure, and pressure calculations are performed only for control volumes which are completely filled.

A fill factor $f$ associated with each control volume indicates the volume fraction of material present. Thus, for full control volumes $f$ is equal to one, while empty control volumes are assigned fill factors equal to zero. For control volumes which are located on the flow front, the fill factor is equal to the volume fraction of SMC present. The fill factors are updated at each time step using the following relation

$$
f_{CV, NEW} = \frac{f_{CV, 0} V_{CV, 0} + \text{FLOW} \cdot \Delta t}{V_{CV, NEW}}.
$$

\[3.21\]
where

\[ f_{CV} = \text{fill factor of control volume (either new value, or original value)} \]
\[ V_{CV} = \text{volume of control volume (either new value, or original value)} \]
\[ \text{FLOW} = \text{computed flow rate of incoming material} \]

In the numerical scheme, incoming flow from neighboring control volumes will occur only if the neighboring control volumes are full \((f=1)\). If all surrounding control volumes happen to have \(f < 1\), then the value of \(\text{FLOW}\) in equation (3.21) would be zero. Nevertheless, the effect of compression will always be present and the value of \(f\) will be affected by mold closure even when there is no incoming material. That is, the original volume of a control volume prior to compression is different than the new volume of the control volume after compression.

For control volumes which are located on the flow front \((0 < f < 1)\), the incoming flow from any neighboring full nodes is computed from the left hand side of equation (3.18). As discussed earlier, the left hand side of equation (3.18) gives the outward flux of material from a full control volume due to compression. For each flow front control volume, the summation given in equation (3.18) is calculated. If a neighboring control volume is full, its nodal pressure will be nonzero and will contribute to the summation. If any neighboring control volumes are not full, their nodal pressures will be zero and will not contribute to the flux.

A variable time step is used which allows just one control volume to become newly filled with material at each time step. To determine this time step, the flow rate of material into each control volume located on the flow front is computed as described above. Based on these computed flow rates, and the known volumes and fill factors of the control volumes at the beginning of the time step, the amount of time needed to just fill each control volume (to bring \(f\) equal to one) is calculated. The smallest value of the
times determined in this manner is then chosen as the time step. The final equation used to compute values of $\Delta t$, which is derived based on mass balance considerations, is

$$
\Delta t = \frac{V_{CV,0} \left( 1 - f_{CV,0} \right)}{\text{FLOW} + \sum_{e=1}^{m} \frac{h_e A_e}{3}}
$$

(3.22)

As discussed earlier, this simulation method is applicable only to thin parts. When a mold geometry is thin and there are no sharp corners or ribs, the program should have no problem handling the simulation. However, when substructures are present which can introduce pressure drops not accounted for by the program, the accuracy of the simulation results may suffer. A numerical simulation was conducted to illustrate the behavior of the program when a rib and a boss are present in the mold geometry. A schematic diagram of the mold geometry, corresponding finite element mesh, and simulated flow front positions are shown in Figure 3.3. If the substructures are very shallow and the cavity height in different regions of the mold changes only gradually, the pressure field may still converge, but the results may be less accurate than if the mold had no substructures. The rib in Figure 3.3 was 3 cm deep, which caused the flow progression near the mold walls to catch up with and move somewhat ahead of the material near the center which passed directly over the rib. This behavior was due to strict adherence to conservation of mass. It was found that more iterations were required for convergence of the pressure field as the depth of the rib was increased. As the depth of the rib was increased to larger and larger values, a point was reached when the pressure field did not converge within 150,000 iterations and the program stopped automatically. More work could be done to determine the best method to accommodate
Figure 3.3. Mold geometry, finite element mesh, and corresponding simulated flow front positions.
ribs and sharp turns in the flow (such as a truck hood which has a bend of around 90°). One approach might be to include an artificial viscosity in these regions. A listing of the Fortran code for the compression molding simulation program is given in Appendix A.

3.2 Heat Transfer Formulation

Heat transfer occurs during the manufacture of SMC parts when the relatively cool initial charge of material is placed into a heated mold cavity. The period of time when the charge sits on the bottom half of the mold prior to contact with the closing upper mold is referred to as the "dwell" time. Most of the heat transfer to the SMC during this period is from the bottom mold half, since the top surface of the SMC is exposed to warm air. Often this period is relatively short since the mold closing speed is as rapid as possible. After contact, heat transfer from both mold halves continues during the filling stage and subsequent curing stage of the process.

For thin parts it may be reasonable to assume that heat conduction is dominant in the Z-direction, because this dimension is small relative to the planar dimensions [36, 48]. With this assumption, the equation of energy during mold filling may be written as

\[
\frac{\partial T}{\partial t} + V_z \frac{\partial T}{\partial Z} = \frac{k}{\rho C_p} \frac{\partial^2 T}{\partial Z^2} + \dot{Q}_R
\]

(3.23)

where the convective term was included to account for mold closure, and \( \dot{Q}_R \) is the source term due to chemical reaction. The source term will be discussed in the next section.
Equation (3.23) may be manipulated using a change of variables \[21\],

\[ \xi = \frac{Z}{h(t)} , \quad \tau = \frac{k}{\rho C_p} \int_0^1 \frac{1}{h(t')^2} dt' , \] (3.24)

and substituting

\[ V_z = \left( \frac{\partial \ln h}{\partial t} \right) Z , \] (3.25)

which can be derived from continuity and "stretching" flow analysis. The result is

\[ \frac{\partial T}{\partial \tau} = \frac{\partial^2 T}{\partial \xi^2} + \left( \frac{h^2 \rho C_p}{k} \right) \dot{Q}_R , \] (3.26)

The implicit finite difference method (FDM) was used to solve equation (3.26) for the temperature and conversion profiles in the thickness direction for flat parts. The finite difference form of equation (3.26) may be written as

\[ \frac{T_{i,j+1} - T_{i,j}}{\Delta \tau} = \frac{T_{i-1,j+1} - 2T_{i,j+1} + T_{i+1,j+1}}{(\Delta \xi)^2} + \left( \frac{h^2 \rho C_p}{k} \right) \dot{Q}_{R,i} , \] (3.27)

where
\[ \Delta \tau = \frac{k}{\rho C_p h^2} \Delta t \]  \hspace{1cm} (3.28)

\[ \Delta \xi = \frac{\Delta Z}{h} = \frac{1}{M-1} \]  \hspace{1cm} (3.29)

\[ \Delta t = \text{time step} \]
\[ i = \text{node number in the thickness direction} \]
\[ j = \text{time index} \]
\[ M = \text{number of nodes in thickness direction} \]

Equation (3.27) may be rearranged to a form suitable for use in a tridiagonal matrix:

\[
\begin{bmatrix}
-\frac{\Delta \tau}{(\Delta \xi)^2} & 1 + \frac{2\Delta \tau}{(\Delta \xi)^2} & 0 & \cdots & 0 \\
1 & -\frac{\Delta \tau}{(\Delta \xi)^2} & 1 & \cdots & 0 \\
0 & 1 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 1 \\
0 & \cdots & 1 & -\frac{\Delta \tau}{(\Delta \xi)^2} & 1
\end{bmatrix}
\begin{bmatrix}
T_{i-1,j+1} \\
T_{i,j+1} \\
T_{i,j} \\
T_{i,j-1} \\
T_{i,j-2}
\end{bmatrix}
= \begin{bmatrix}
\frac{h^2 \rho C_p \Delta \tau}{k} Q_{Ri} \\
T_{i,j} \\
T_{i+1,j} \\
T_{i+2,j} \\
T_{i+3,j}
\end{bmatrix}
\]  \hspace{1cm} (3.30)

A convective-type boundary condition was applied at the upper and lower mold surfaces.

\[-k \frac{\partial T}{\partial Z} = h_c (T - T_{\text{ref}}) \]  \hspace{1cm} (3.31)

where \( h_c \) = convective heat transfer coefficient = 497.9 J/m^2 sec K [17]
\( T_{\text{ref}} \) = \( T_w \) = mold wall temperature
This type of boundary condition is a general boundary condition which accounts for the fact that the SMC temperature just outside a small thermal boundary layer may be different than the mold wall temperature. For illustration, the finite difference representation of equation (3.31) may be written for the lower SMC boundary, corresponding to node number one, as

$$\frac{k(T_2 - T_1)}{\Delta Z} = h_c (T_1 - T_{ref})$$

(3.32)

Inserting equation (3.29) and rearranging,

$$\left[ 1 + \frac{h_c h \Delta \xi}{k} \right] T_1 - T_2 = \left[ \frac{h_c h \Delta \xi}{k} \right] T_{ref}$$

(3.33)

which corresponds to the first row of the tridiagonal matrix. In similar manner, the last row is given by

$$\left[ 1 + \frac{h_c h \Delta \xi}{k} \right] T_M - T_{M-1} = \left[ \frac{h_c h \Delta \xi}{k} \right] T_{ref}$$

(3.34)

The middle rows of the matrix are given by equation (3.30).

During the curing stage after mold filling is complete, equation (3.23) is used without the convective term to obtain the temperature and conversion profiles, again using a convective boundary condition.
The finite difference form of equation (3.35), after rearrangement is

\[
\frac{\partial T}{\partial t} = \frac{k}{\rho C_p} \frac{\partial^2 T}{\partial Z^2} + \dot{Q}_R
\]

(3.35)

The finite difference form of equation (3.35), after rearrangement is

\[
\left[ \frac{k \Delta t}{\rho C_p (\Delta Z)^2} \right] T_{i-1,j+1} + \left[ 1 + \frac{2k \Delta t}{\rho C_p (\Delta Z)^2} \right] T_{i,j+1} + \left[ - \frac{k \Delta t}{\rho C_p (\Delta Z)^2} \right] T_{i+1,j+1} = T_{i,j} + (\Delta t) \dot{Q}_R
\]

(3.36)

Equation (3.35) may also be used during the dwell time prior to mold closure. In this case \( T_{\text{ref}} \) in equation (3.31) becomes \( T_0 \), the warm air temperature for the upper SMC surface. As an approximation, during cooling after the part is ejected from the mold, equation (3.35) applies with \( T_{\text{ref}} = T_0 \) at both the upper and lower SMC surfaces.

3.3 Kinetic Model for Unsaturated Polyester/Styrene System

The chemical reaction which occurs during SMC processing is a free-radical crosslinking copolymerization between styrene monomers and unsaturated polyester resin. The exothermic nature of this reaction makes it necessary to include a source term in the energy equation as described in the previous section. The form of the reaction source term for use in equation (3.23) is

\[
\dot{Q}_R = \frac{\Delta H}{C_p} \frac{d\alpha}{dt}
\]

(3.37)
where \( \Delta H = \) heat of reaction, J/g
\( \alpha = \) fractional conversion
\( \frac{d\alpha}{dt} = \) rate of conversion, or reaction rate

The reaction rate \( \frac{d\alpha}{dt} \) may be calculated from any suitable kinetic model. The type of kinetic model chosen for use in numerical computations depends on the type of information needed by the user. Kamal and Sourour [49] proposed one of the first kinetic models applicable to unsaturated polyester/styrene systems. Their model was mostly empirical, but did a good job of correlating reaction rate data from DSC. Since then, other kinetic models have been advanced [53 - 56] to describe the curing reaction which are more phenomenological or mechanistic in nature, and therefore are more complex. The one chosen for use in the present work is a model proposed by Stevenson [56], which provides a mechanistic description of the reaction:

Initiation:

\[
\frac{d[I]}{dt} = -k_d[I] \tag{3.38}
\]

Inhibition:

\[
\frac{d[Z]}{dt} = -k_z[Z][R.] \tag{3.39}
\]

Radical Concentration:

\[
\frac{d[R.]}{dt} = 2f k_d[I] - q k_z[Z][R.] \tag{3.40}
\]
Propagation:

\[ \frac{d\alpha}{dt} = k_P (1 - \alpha)[R\cdot] \]  

(3.41)

where

\[ k_d = A_d \exp \left\{ \frac{-E_d}{RT} \right\} \]  

(3.42)

\[ k_z = A_z \exp \left\{ \frac{-E_z}{RT} \right\} \]  

(3.43)

\[ k_p = k_{p0} \left( 1 - \frac{\alpha}{\alpha_f} \right)^m \]  

(3.44)

\[ k_{p0} = A_{p0} \exp \left\{ \frac{-E_{p0}}{RT} \right\} \]  

(3.45)

and

\[ [I] \quad = \text{initiator concentration} \]

\[ [Z] \quad = \text{inhibitor concentration} \]

\[ [R\cdot] \quad = \text{radical concentration} \]

\( f, q \quad = \text{initiator and inhibitor efficiencies, respectively} \)

\( k_d \quad = \text{rate constant for initiator decomposition} \)
$k_Z = \text{rate constant for inhibitor consumption}$

$\text{k}_p = \text{propagation rate constant}$

$E_{P0}, E_d, E_Z = \text{activation energies}$

$A_{P0}, A_d, A_Z = \text{pre-exponential factors}$

The parameters in the model were found using a nonlinear search routine [57] to fit experimental data from differential scanning calorimetry (DSC) runs on an SMC paste. A description of the method used and values of the kinetic parameters are given in Chapter 4. With parameters known, the kinetic equations (3.38 - 3.45) were solved numerically at each vertical node position, and the resulting heat source terms from equation (3.37) were combined with the energy equation as written in equation (3.30) or (3.36).

With the source term due to reaction included, the energy equation was solved for the temperature and conversion profiles across the thickness of a flat part to illustrate the ability of the code to model the heat transfer during processing. The mold wall temperatures were specified to be 150°C. The initial height of the charge was arbitrarily chosen to be 7 cm, and the mold closing speed was set at 0.4 cm/sec. Figure 3.4 shows the resulting temperature profiles at various times. Initially, the material has a uniform temperature at 25°C as shown in Figure 3.4 (a). Once the material is placed on the bottom mold half prior to contact with the upper mold, Figure 3.4 (b) shows the simulated temperature profile during the dwell period. In part (c) of the figure, the upper mold has contacted the charge and a parabolic temperature profile is observed. At a later time in part (d), the reaction exotherm causes the middle region of the charge to become hotter than the material near the mold walls. This occurs because material close
Figure 3.4. Simulated temperature profiles at various times during compression molding of an SMC paste.
to the walls can dissipate heat to the mold more easily than material near the center.

Similarly, the conversion profiles at various times are shown in Figure 3.5.
Figure 3.5. Simulated conversion profiles at various times during compression molding of an SMC paste.
4.1 Mold Filling Experiments

To test the ability of the flow simulation routine described in section 3.1 to predict charge movement during molding, a set of isothermal short shot experiments using silly putty was conducted. Other studies have also been conducted to verify the accuracy of thin charge flow models [26, 28, 29, 34]. The experimental mold geometry is illustrated schematically in Figure 4.1 (a). A substructure with gradually varying depth was present on one side of the initial charge. The mold was mounted in an Instron Model 1137 press capable of closing at a constant speed. To approximate the lubricated mold surface condition commonly encountered due to heat transfer from the mold walls, grease was applied between the mold walls and the silly putty. The charge shape after each short shot was recorded by separating the mold halves without disturbing the charge, and tracing the charge shape onto a piece of graph paper. These results are compared to simulated flow progression results in Figure 4.1 (b) and reasonable agreement is observed.
Figure 4.1. (a) Experimental mold geometry; (b) Comparison of simulation results to experimental flow progressions.
4.2 Kinetic Characterization Using DSC

As discussed in section 3.3, a kinetic model proposed by Stevenson [56] was used to compute the fractional conversion and reaction rate of the free radical polymerization which occurs when SMC is processed. With reaction rate known, the heat release rate could be obtained. In this study, the parameters $k_{po}$, $k_z$, $k_d$, and $m$ were determined using a nonlinear search routine [57] for assumed values of initiator and inhibitor efficiencies, and initial inhibitor concentration. Then, $A_{po}$, $E_{po}$, $A_z$, $E_z$, $A_d$, and $E_d$ could be calculated from the rate constants and temperature data. Experimental data from DSC included one scanning run and three isothermal runs performed at temperatures of 90° C, 100° C, and 110° C. Initial guesses were made for the three rate constants and parameter $m$ for use in the nonlinear search, which was conducted for each isothermal run. The optimization began by reading in experimental reaction rate versus time data from DSC. The kinetic equations were then solved numerically, and the sum of the squared residuals between the experimental and calculated rates at each data point was determined. The routine then perturbed the initial guess values and the procedure was repeated until a minimum value for the sum of squared residuals was obtained.

The activation energies and pre-exponential factors were then obtained by plotting the natural logarithm of each rate constant versus ($1/T$). The slope of the best line through the points gave the activation energy, and the intercept provided the frequency factor. Plots of this type based on the optimum rate constants are given in Figure 4.2. A summary of the kinetic parameters is given in Table 4.1.

To verify that the kinetic parameters given in Table 4.1 were reasonable, simulation results which used these parameters were compared to the experimental DSC
Figure 4.2. Method used to obtain activation energies and pre-exponential factors for the kinetic model.
Table 4.1 Kinetic Parameter Values

<table>
<thead>
<tr>
<th>T (K)</th>
<th>k_{p0} (L/sec mol)</th>
<th>k_{z} (L/sec mol)</th>
<th>k_{d} (1/sec)</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>363</td>
<td>0.2342 E+03</td>
<td>0.4422 E+05</td>
<td>0.5070 E-03</td>
<td>1.344</td>
</tr>
<tr>
<td>373</td>
<td>0.4023 E+03</td>
<td>0.5857 E+05</td>
<td>0.1350 E-02</td>
<td>1.388</td>
</tr>
<tr>
<td>383</td>
<td>0.7558 E+03</td>
<td>0.1130 E+06</td>
<td>0.2929 E-02</td>
<td>1.034</td>
</tr>
</tbody>
</table>

\[ A_{p0} = 0.1254 \text{ E+13} \quad E_{p0} = 0.6769 \text{ E+05 (J/mol)} \]
\[ A_{z} = 0.2482 \text{ E+13} \quad E_{z} = 0.5407 \text{ E+05 (J/mol)} \]
\[ A_{d} = 0.2086 \text{ E+12} \quad E_{d} = 0.1015 \text{ E+06 (J/mol)} \]

m = 1.255
results. First, simulations were conducted at the three isothermal temperatures and compared to the isothermal DSC data. Note that for these simulations, the rate constants were calculated from the Arrhenius relations given in equations (3.42 - 3.45) of Chapter 3, rather than simply using the optimum rate constants found in the search routine. This provided a better test of the temperature dependence of the constants. The results of the isothermal comparisons are given in Figure 4.3, and good agreement is observed.

Next, a nonisothermal simulation was conducted which started at room temperature and ramped the temperature linearly at 10° C/min. This simulation was then compared to the DSC scanning experiment, which also used a ramp of 10° C/min. This comparison is shown in Figure 4.4, and very good agreement is seen. Therefore, the kinetic parameters given in Table 4.1 were used in the heat transfer routine described in sections 3.2 and 3.3 for use during molding simulations.

4.3 Temperature Measurements During Molding Experiments

The heat transfer formulation was tested by comparing simulated temperatures with experimentally measured temperatures during compression molding. The press/mold set up described in section 4.1 was also used for these experiments, only this time cartridge heaters were placed into bored inserts located in the upper and lower mold halves. Before an experiment, thermocouples were embedded between the plies of the initial charge at known positions. The thermocouple wires were connected to a Dianachart data acquisition package [58] and the data were recorded by an IBM-XT computer and saved on the hard disk. The thermocouple wires were securely positioned
Figure 4.3. Comparison of kinetic model simulations to isothermal DSC results.
Figure 4.4. Comparison of kinetic model simulations to scanning DSC results.
beneath a removable side mold wall in order to prevent damage by the closing upper mold.

At time zero the initial charge was placed into the hot mold cavity, and the upper mold was lowered at constant speed. The time of contact between the upper charge surface and the top mold half was noted in order to determine the dwell time. Mold closure continued until the cavity became just filled, and then the apparatus remained in this configuration during the curing period. Throughout the experiment, thermocouple measurements were recorded by the computer. Figure 4.5 shows the results of an experiment which used two thermocouples, and the corresponding simulation results. The thermocouple locations in the figure correspond to their final positions. During the dwell period (~ 111 seconds), the temperature at T2 was less than that of T1 because the upper mold had not yet contacted the charge. After closure, both temperature responses increased together to around the mold wall temperature (112° C), and remained roughly in this range until the reaction exotherm occurred. At this point the temperatures increased to maximum values and then decreased gradually back to the mold wall temperature.

The simulation results of Figure 4.5 are in reasonable agreement with the experimental results. After the dwell period, the simulated temperatures continued to increase due to heat conduction from the mold, and then leveled off. The boundary condition used considered the temperature of the mold wall to be constant, and did not recognize that it actually varies somewhat. Thus, there is some minor disagreement with experimental results in the region prior to the exothermic peak. The temperature actually increased somewhat in this region due to a small heat release from the material in the early stages of reaction, and because the system was able to dissipate less heat once the upper mold was lowered. The increased temperature resulted in earlier initiation of
Figure 4.5. Comparison of simulated temperature profiles to actual temperature measurements during a molding experiment using SMC.
reaction, and therefore the exothermic peak occurred somewhat earlier than the simulated peak. A more refined prediction could be obtained by including the mold itself in the heat transfer calculations. The mold wall temperature could then vary and more closely match the observed behavior.

The results of a similar experiment are shown in Figure 4.6, only this time a thermocouple was positioned in the substructure region. The response prior to about 150 seconds corresponds to the time before the SMC contacted the thermocouple. Again, the experimental exotherm occurred sooner than the simulated prediction. A probable reason for the discrepancy in this case was the effect of the heat transfer from the vertical mold walls in the substructure region. Since the numerical formulation only considered one-dimensional heat transfer, this extra source of heat did not exist as far as the simulation was concerned. Therefore, the simulation predicted onset of the exothermic peak at a later time than was observed experimentally. Nevertheless, the general trend of the prediction is in reasonable qualitative agreement with the experimental measurements.
Figure 4.6. Comparison of simulated and experimental temperature profiles for a thermocouple located in a rib.
5.1 Introduction

As discussed in Chapter 2, the ability to predict part shrinkage could ultimately lead to savings in time and reduced manufacturing costs. For thermoplastic parts, any volume changes which occur are due to thermal expansion or contraction of the material. The mold cavity dimensions must be specified to account for these thermal volume changes. For thermosetting systems, however, the problem is more complicated because a polymerization shrinkage occurs in addition to the thermal volume changes. A predictive expression for the volumetric behavior of a thermosetting system must therefore include a means for modelling the polymerization shrinkage which occurs during processing.

For systems whose chemical reaction takes place via step polymerization, there is often little or no cure shrinkage because of the condensation by-products formed. For many systems which undergo chain polymerizations, however, the volume change due to polymerization shrinkage coincides with increasing fractional conversion [59]. In the past, the conversion versus time of such systems was measured using a dilatometer in a
constant temperature bath. Table 5.1 lists the percent volume changes due to chain polymerization of some common systems, along with the corresponding densities of the monomers and polymers [60].

While the chain growth systems given in Table 5.1 exhibit roughly simultaneous polymerization shrinkage and increasing conversion, it appears that for unsaturated polyester (UPE)/styrene reactions, material shrinkage may actually begin prior to any significant increase in conversion [61]. This chapter presents a new model to predict the volume changes which occur during curing of unsaturated polyester/styrene systems, including both thermal effects and polymerization effects. The model is able to accommodate the early onset of shrinkage and was developed by combining experimental results from differential scanning calorimetry (DSC), electron spin resonance spectroscopy (ESR), thermomechanical analysis (TMA), and dilatometry [62].

5.2 Experimental Results

The composition of the UPE/styrene sample used in the analysis is given in Table 5.2. After the mixture was prepared, the sample was first sealed in a plastic pouch to prevent evaporation of styrene, and then degassed to remove dissolved air using a vacuum pump. A dilatometer [61] was used to follow the volume change of the resin during cure at 74°C and 85°C. DSC experiments were conducted simultaneously at the same temperatures to measure the reaction rate and conversion of the resin, and ESR experiments were conducted to measure the radical concentration during the course of reaction. The results of these three sets of experiments at each temperature were
Table 5.1 Densities and Polymerization Shrinkages of Common Systems [60]

<table>
<thead>
<tr>
<th>Monomer</th>
<th>Density, g/ml, 25° C</th>
<th>Monomer</th>
<th>Polymer</th>
<th>% Volume Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vinyl Chloride</td>
<td>0.919</td>
<td>1.406</td>
<td>34.4</td>
<td></td>
</tr>
<tr>
<td>Acrylonitrile</td>
<td>0.800</td>
<td>1.17</td>
<td>31.0</td>
<td></td>
</tr>
<tr>
<td>Vinlylidene bromide</td>
<td>2.178</td>
<td>3.053</td>
<td>28.7</td>
<td></td>
</tr>
<tr>
<td>Vinlylidene chloride*</td>
<td>1.213</td>
<td>1.71</td>
<td>28.6</td>
<td></td>
</tr>
<tr>
<td>Vinyl bromide</td>
<td>1.512</td>
<td>2.075</td>
<td>27.3</td>
<td></td>
</tr>
<tr>
<td>Methacrylonitrile</td>
<td>0.800</td>
<td>1.10</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>Methyl acrylate</td>
<td>0.952</td>
<td>1.223</td>
<td>22.1</td>
<td></td>
</tr>
<tr>
<td>Vinyl acetate *</td>
<td>0.934</td>
<td>1.191</td>
<td>21.6</td>
<td></td>
</tr>
<tr>
<td>Methyl methacrylate</td>
<td>0.940</td>
<td>1.179</td>
<td>20.6</td>
<td></td>
</tr>
<tr>
<td>Diallyl succinate</td>
<td>1.056</td>
<td>1.30</td>
<td>18.8</td>
<td></td>
</tr>
<tr>
<td>Ethyl methacrylate</td>
<td>0.911</td>
<td>1.11</td>
<td>17.8</td>
<td></td>
</tr>
<tr>
<td>Diallyl maleate</td>
<td>1.077</td>
<td>1.30</td>
<td>17.2</td>
<td></td>
</tr>
<tr>
<td>Ethyl acrylate</td>
<td>0.919</td>
<td>1.095</td>
<td>16.1</td>
<td></td>
</tr>
<tr>
<td>n-Butyl acrylate</td>
<td>0.894</td>
<td>1.055</td>
<td>15.2</td>
<td></td>
</tr>
<tr>
<td>n-Propyl methacrylate</td>
<td>0.902</td>
<td>1.06</td>
<td>15.0</td>
<td></td>
</tr>
<tr>
<td>Styrene</td>
<td>0.905</td>
<td>1.062</td>
<td>14.5</td>
<td></td>
</tr>
<tr>
<td>n-Butyl methacrylate</td>
<td>0.889</td>
<td>1.055</td>
<td>14.3</td>
<td></td>
</tr>
</tbody>
</table>

* Density at 20° C
Table 5.2 Resin Formulation Used in this Study

<table>
<thead>
<tr>
<th>Ingredient</th>
<th>Mass (grams)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polyester resin (Ashland Chemical, Q6585)</td>
<td>56.0</td>
</tr>
<tr>
<td>Styrene monomer</td>
<td>6.5</td>
</tr>
<tr>
<td>Low profile additive in styrene (Union Carbide, LP-40A)</td>
<td>37.5</td>
</tr>
<tr>
<td>PDO initiator</td>
<td>1.1</td>
</tr>
</tbody>
</table>

combined and are shown in Figures 5.1 and 5.2. No runs were conducted at temperatures higher than 85°C because it was found that too much coupling occurred between the initial thermal expansion and the onset of polymerization shrinkage. The lower temperature bound, 74°C, allowed the volume changes to be complete within a reasonable amount of time. Lower than 74°C, the volume changes would continue for much longer time periods.

As shown in Figures 5.1 and 5.2, the volume change curves showed an expansion first due to the heating period of the dilatometer. After heating, the volume started to decrease as the samples began to shrink. Most of the shrinkage, however, occurred at the very beginning of reaction, where the total resin conversion was very low. Radical concentration measured by ESR showed a sharp increase which began at about the same time the material began to shrink, and continued to increase throughout
Figure 5.1. Combined dilatometer, DSC, and ESR results at 74°C.
Figure 5.2. Combined dilatometer, DSC, and ESR results at 85°C.
the reaction. Like the shrinkage, the radical concentration attained a substantial amount of its maximum value before any measurable increase in conversion could be detected. These experimental results suggest that to study the shrinkage mechanism of UPE/styrene resins, both the conversion and the radical concentration should be considered.

The volumetric polymerization shrinkage from each run was calculated from the volume change data beginning with the time at which shrinkage commenced using the relation

\[ S_t = \left( \frac{\Delta V}{V_0} \right)_{Max} - \left( \frac{\Delta V}{V_0} \right)_t \]  

(5.1)

where the subscript \( t \) denotes time. The shrinkage at various times was then cross-plotted versus the conversion at the same times using an interpolation program, and the result is shown in Figure 5.3. It is seen that the shrinkage is independent of temperature when plotted against the conversion and can be divided into two stages. In the first stage shrinkage increased sharply, while no change in conversion could be detected. The onset of reaction occurred in the second stage, where the shrinkage increased nearly linearly versus conversion as in conventional free radical polymerizations.
Figure 5.3. Overall shrinkage versus conversion master curve data.
5.3 Modelling of Volumetric Changes

The overall volumetric changes of a thermoset resin during curing can be considered to be a combination of thermal expansion/contraction and polymerization shrinkage. That is,

\[
\frac{\Delta V}{V_0} \text{ Overall} = \left( \frac{\Delta V}{V_0} \right)_{\text{Thermal Contribution}} - \left( \frac{\Delta V}{V_0} \right)_{\text{Polymerization Shrinkage}}
\]  \hfill (5.2)

where

\[
\left( \frac{\Delta V}{V_0} \right)_{\text{Thermal Contribution}} = \beta \Delta T
\]  \hfill (5.3)

\[
\beta = \beta_M (1 - \alpha) + \beta_P \alpha
\]  \hfill (5.4)

\[
\Delta T = T - T_0
\]  \hfill (5.5)

with

\[
\beta_M = \text{volumetric thermal expansion coefficient of monomer (or uncured resin)}
\]

\[
\beta_P = \text{volumetric thermal expansion coefficient of polymer (or cured resin)}
\]

T_0 = \text{ambient room temperature (25°C in this study)}

V_0 = \text{monomer volume at ambient room temperature}

\alpha = \text{fractional conversion}
The first term on the right hand side of equation (5.2) can be rewritten in the following form:

$$\beta \Delta T = \left[ \beta_M (1 - \alpha) + \beta_P \alpha \right] \Delta T = \beta_M \Delta T + \left[ (\beta_P - \beta_M) \Delta T \right] \alpha$$

(5.6)

This expression serves two purposes. First, it allows any thermal expansions or contractions experienced by the material to be predicted by the model. Second, when the material cures, the difference between $\beta_M$ and $\beta_P$ may result in a volume change. For a constant $\Delta T$ this effect is linear in conversion. This linearity reflects the experimental results shown earlier in Figure 5.3, where the shrinkage continued to increase linearly with conversion. Since, in general, $\beta_P$ is less than $\beta_M$, as the conversion increases the expression in equation (5.6) becomes smaller and contributes to the overall volume decrease or shrinkage.

The second term on the right hand side of equation (5.2) is the expression which models the early onset of shrinkage. When a simple linear relation exists between the shrinkage and the conversion, as for the systems given in Table 5.1, the second term would not be needed. However, as shown earlier, the polymerization shrinkage of unsaturated polyester resins begins prior to any increase in conversion. Therefore, the cure shrinkage cannot be solely a function of conversion for UPE resins.

The fact that the material began to shrink at about the same time the radical concentration began to increase motivates a cross-plot of the polymerization shrinkage term of equation (5.2) versus radical concentration. Figure 5.4 shows that a master curve is formed when shrinkage is plotted against radical concentration, indicating that the polymerization shrinkage mechanism is closely related to the radical formation. Note
Figure 5.4. Polymerization shrinkage versus radical concentration master curve data.
that the polymerization shrinkage plotted in Figure 5.4 has already subtracted out the
conversion contribution, and therefore reflects the early shrinkage which occurs during
the cure of UPE resins. The subtraction of the conversion effect was carried out by
solving equation (5.2) for the polymerization shrinkage term, and computing its value
using known values of conversion and overall volume change. The overall approach
taken to model the volumetric changes of unsaturated polyester resin during curing is
shown in Figure 5.5.

The experimental results indicate that the polymerization shrinkage is
independent of temperature when plotted against the radical concentration. This may be
used as a basis to model the polymerization shrinkage of the unsaturated
polyester/styrene system. From the shape of the master curve shown in Figure 5.4, the
cure shrinkage versus radical concentration behavior may be expressed as

\[
\left( \frac{\Delta V}{V_0} \right)_{\text{Polymerization Shrinkage}} = S_f \left\{ \frac{c\overline{R^n}}{1 + c\overline{R^n}} \right\}
\]

where

\[
\overline{R} = \frac{R}{R_f - R}
\]

R = radical concentration, (mol/L)
R_f = final radical concentration, (mol/L)
Modelling of Volumetric Changes of Unsaturated Polyester Resins

Figure 5.5. Research approach to developing a volume change model for the unsaturated polyester/styrene reactive system.
The form of equation (5.7) is similar to an early expression given by Kamal and Sourour [49] to model the "S-shaped" fractional conversion profile of an unsaturated polyester/styrene polymerizing system. An earlier version of the shrinkage model used an exponential-type relation to predict the shrinkage versus radical concentration behavior [63, 64]. The full volume change model for the unsaturated polyester/styrene reactive system is, from equations (5.3), (5.4), and (5.7),

$$\left(\frac{\Delta V}{V_0}\right)_{\text{Overall}} = \left[\beta_M (1 - \alpha) + \beta_P \alpha\right] \Delta T - \log\left\{\frac{c R^n}{1 + c R^n}\right\}$$

(5.9)

5.4 Determination of Parameters c and n

The parameters c and n were determined by fitting the polymerization shrinkage term in equation (5.7) to the master curve shown in Figure 5.4. If the data shown in Figure 5.4 are independent of temperature, either run may be used to obtain the best values of c and n. For conciseness, optimum parameter values at both 74°C and 85°C were obtained, and then the average values of c and n were taken as the best choices. Table 5.3 gives the resulting parameter values.
Table 5.3 Values for c and n

<table>
<thead>
<tr>
<th></th>
<th>c</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>74°C</td>
<td>2.775</td>
<td>1.6967</td>
</tr>
<tr>
<td>85°C</td>
<td>5.965</td>
<td>1.5837</td>
</tr>
<tr>
<td>Average</td>
<td>4.37</td>
<td>1.6402</td>
</tr>
</tbody>
</table>

Figure 5.6 compares the experimental data to the prediction obtained from equation (5.7) using average values of c and n. The values of R and R_f in equation (5.8) were obtained from the experimental radical concentration data at 85°C. It is seen that reasonable agreement with the data is obtained. Figure 5.7 compares the predicted polymerization shrinkage from equation (5.7) with the data as a function of the values of time which corresponded to the radical concentration values. In this figure, of course, the values of R and R_f in equation (5.8) were obtained from the data at each of the temperatures.

5.5 Estimation of Thermal Expansion Coefficients

To obtain values for the volumetric thermal expansion coefficients, $\beta_M$ and $\beta_P$, thermomechanical analysis (TMA) of a sample which had previously been cured in the dilatometer was conducted to obtain the coefficient of linear expansion for the cured polymer. The result is shown in Figure 5.8. The slope of the line in the region between...
Figure 5.6. Comparison of predicted polymerization shrinkage from equation (5.7) using average values of c and n to experimental master curve results.
Figure 5.7. Comparison of predicted polymerization shrinkage from equation (5.7) using average values of $c$ and $n$ to experimental results.
Figure 5.8. Experimental TMA results from a sample which had previously been cured in the dilatometer.
room temperature and 90°C gave an approximate value for the coefficient of linear expansion for the cured polymer in the temperature range at which the dilatometer was operated. An approximate value for the corresponding coefficient of volume expansion, $\beta_p$, is three times that of the linear coefficient [65 - 67]. For this sample, the linear coefficient is $8.93 \times 10^{-5}/^\circ C$, and therefore $\beta_p = 2.68 \times 10^{-4}/^\circ C$.

Next it was necessary to determine $\beta_M$, the coefficient of volume expansion of the liquid monomer. The initial thermal expansion periods in the volume change curves of Figures 5.1 and 5.2 could be used in the calculation of $\beta_M$ since ideally no reaction occurred in this period. The only volume changes during this period were due to thermal expansion from room temperature to the reaction temperature. It should be pointed out that during the dilatometer experiments, a first run on the sample was obtained, and then a second run on the cured sample was performed to serve as a baseline. The second curve was subtracted from the first, and this is the result that was plotted in Figures 5.1 and 5.2 for the two runs. Therefore, the initial thermal expansions in the figures actually provided approximate measures of $(\beta_M - \beta_p)$. Knowing $(\beta_M - \beta_p)$ and $\beta_p$, the value of $\beta_M$ may then be computed.

Examination of the data in Figures 5.1 and 5.2 indicates that the maximum thermal expansion achieved was about the same, 3%, at both 74°C and 85°C, in spite of a temperature difference of 11°C. An explanation for why a difference in thermal expansion was not observed is the possibility that the sample had not finished expanding prior to the onset of cure shrinkage at 85°C. Therefore, the data at 74°C were used to determine the thermal expansion coefficient of uncured resin. From the previously obtained value of $\beta_p = 2.68 \times 10^{-4}/^\circ C$, it follows that $\beta_M = 8.80 \times 10^{-4}/^\circ C$. For comparison, the volumetric thermal expansion coefficient of styrene monomer at 40°C is given in the literature as $9.98 \times 10^{-4}/^\circ C$ [68].
5.6 Models for Conversion and Radical Concentration

To apply the volume change model given by equation (5.9), it is necessary to be able to predict both the conversion and the radical concentration of the reactive system. The kinetic model proposed by Stevenson [56], described earlier in Chapter 3, was used here to predict the fractional conversion of the resin given in Table 5.2. The kinetic model equations are repeated here for ease of reference.

Initiation:

$$\frac{d[I]}{dt} = -k_d[I]$$

(5.10)

Inhibition:

$$\frac{d[Z]}{dt} = -k_z[Z][R\cdot]$$

(5.11)

Radical Concentration:

$$\frac{d[R\cdot]}{dt} = 2fk_d[I] - qk_z[Z][R\cdot]$$

(5.12)

Propagation:

$$\frac{d\alpha}{dt} = k_p(1 - \alpha)[R\cdot]$$

(5.13)
where

\[ k_p = k_{p0} \left(1 - \frac{\alpha}{\alpha_f}\right)^m \]  \hspace{1cm} (5.14)

\[ k_{p0} = A_{p0} \exp \left\{ \frac{-E_{p0}}{RT} \right\} \]  \hspace{1cm} (5.15)

\[ k_d = A_d \exp \left\{ \frac{-E_d}{RT} \right\} \]  \hspace{1cm} (5.16)

\[ k_z = A_z \exp \left\{ \frac{-E_z}{RT} \right\} \]  \hspace{1cm} (5.17)

To predict the conversion, it was necessary to fit the kinetic equations to the DSC runs conducted on the resin. The approach used in this study was to find optimum values of \( k_d, k_p, k_{p0}, \) and \( m \) using a nonlinear search routine [57]. After obtaining reasonable agreement between the kinetic equations and the DSC results, the activation energies and pre-exponential factors in the Arrhenius relations could be computed from the slopes and intercepts of plots of the logarithms of the rate constants versus the reciprocal absolute temperatures. This approach was described in section 4.2 and has also been used previously [53]. The kinetic parameter values obtained are given in Table 5.4. Figure 5.9 shows the DSC results at 74°C and 85°C in comparison to the predictions from the kinetic model, and good agreement is observed. In this figure, the predicted curves were generated using the rate constants as calculated from the Arrhenius expressions. With kinetic parameters known, the conversion of the resin may
Table 5.4. Kinetic Parameter Values

<table>
<thead>
<tr>
<th>T (K)</th>
<th>$k_{po}$ (L/sec mol)</th>
<th>$k_z$ (L/sec mol)</th>
<th>$k_d$ (1/sec)</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>347</td>
<td>386.92</td>
<td>44641.8</td>
<td>5.99 E-08</td>
<td>2.24</td>
</tr>
<tr>
<td>358</td>
<td>679.32</td>
<td>60635.3</td>
<td>3.64 E-07</td>
<td>1.96</td>
</tr>
</tbody>
</table>

$A_{po} = 3.1809 \times 10^{10}$  \hspace{1cm} $E_{po} = 52592.7$ (J/mol)

$A_z = 9.5409 \times 10^{08}$  \hspace{1cm} $E_z = 28775.0$ (J/mol)

$A_d = 1.4212 \times 10^{18}$  \hspace{1cm} $E_d = 168615.4$ (J/mol)

m = 2.1
Figure 5.9. Comparison of simulated reaction rates to experimental DSC data.
be predicted. An alternate method to obtain the kinetic parameters may be used when both DSC and ESR results are available, and is described in Appendix B.

It should be noted that according to equations (5.10) - (5.17), when the radical concentration begins to increase, the propagation reaction should also commence and the conversion should therefore increase as well. It appears, then, that the DSC and ESR results for the unsaturated polyester/styrene system do not exactly follow such a reaction mechanism in light of the early increase in radical concentration. A possible explanation for the lag between the increase in radical concentration and the onset of reaction is that initially the radicals may be involved in microstructure formation [69]. The radicals may not be effective in propagating the reaction until their role in the morphological changes of the system is completed. The fact that the polymerization shrinkage of the unsaturated polyester/styrene system begins at about the same time the radical concentration begins to increase, as shown in Figures 5.1 and 5.2, suggests that the shrinkage may depend largely on the morphological changes.

Given these results, the kinetic parameters obtained earlier to predict the conversion cannot be used to simultaneously predict the experimentally observed radical concentration. It appears that eventually the kinetic model cited earlier should be modified to accommodate the effect of morphological changes such that the radical concentration and the DSC reaction exotherm can be connected. At the present time, however, to predict the radical concentration a set of equations analogous to equations (5.10) - (5.12) were solved using a set of parameters specifically obtained for use in conjunction with the volume change model given by equation (5.9). These polymerization shrinkage parameters were also found using the nonlinear search routine, only now the ESR results were fitted. The resulting parameter values are given in Table 5.5. Figure 5.10 compares experimental radical concentration profiles at 74°C, 85°C,
Table 5.5 Polymerization Shrinkage Parameter Values

<table>
<thead>
<tr>
<th>T (K)</th>
<th>$k_z^*$ (L/sec mol)</th>
<th>$k_d^*$ (1/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>347</td>
<td>0.50 E+06</td>
<td>0.96 E-04</td>
</tr>
<tr>
<td>358</td>
<td>0.72 E+06</td>
<td>0.35 E-03</td>
</tr>
<tr>
<td>368</td>
<td>0.90 E+06</td>
<td>0.22 E-02</td>
</tr>
</tbody>
</table>

$A_z^* = 0.15522 \times 10^{11}$  \hspace{1cm} $E_z^* = 29815.0$ (J/mol)

$A_d^* = 0.43036 \times 10^{20}$  \hspace{1cm} $E_d^* = 157550.0$ (J/mol)

and 95°C to the predictions of the model, and reasonable agreement is observed. With polymerization shrinkage parameters known, the radical concentration of the resin may be computed and used in the volume change model. Comparing the activation energies given in Table 5.5 to the corresponding activation energies given in Table 5.4, it is seen that they are of similar magnitude. This would indicate that the temperature dependence of the initiation and inhibition steps for the two cases are similar, and that the influence of the morphological changes on the rate constants shows up mainly in the pre-exponential factors.
Figure 5.10. Comparison of simulation results to radical concentration versus time data obtained from ESR experiments.
5.7 Implementation of the Volume Change Model

The overall volume change model, equation (5.9), was tested in a separate experiment to predict both thermal volume changes and polymerization shrinkage. The resin was heated in the dilatometer from room temperature to 82°C, and the volume changes and resin temperature were recorded as a function of time. The experimental temperature profile, measured with a thermocouple, is given in Figure 5.11 along with the experimentally measured volume change data. The temperature profile experienced by the sample was used in the models mentioned earlier to predict the conversion and radical concentration of the reaction mixture. These simulated quantities were then used in equation (5.9) to predict the non-isothermal volume changes versus time. The simulated overall volume change curve is shown in Figure 5.11 in comparison to the experimental results. The predicted polymerization shrinkage begins after about 30 minutes of reaction time, which corresponds to an increase in the simulated radical concentration. This behavior agrees fairly well with the experimental data shown in Figure 5.11. To illustrate the interaction of the model with conversion, the predicted conversion profile and the first term on the right hand side of equation (5.9) are also shown in Figure 5.11. The first term on the right hand side of equation (5.9) is labeled the thermal term in the figure. As the conversion increases, the first term of the model shows a corresponding shrinkage which contributes to the overall volume change. The predicted shrinkage at the end of the run matches the experimental results reasonably well. The same qualitative behavior shown in Figure 5.11 has been observed experimentally by others during SMC cure, as shown in Figure 5.12 [46, 47].
Figure 5.11. Comparison of simulation results from the volume change model to experimental volume change data.
Figure 5.12. Experimental mold displacement versus time showing the same qualitative trends given in Figure 5.11 [46, 47].
This work has presented techniques to model three major aspects associated with SMC compression molding, including material flow, heat transfer and curing, and volumetric changes.

6.1 Material Flow

A control volume/finite element technique was developed to simulate the flow of SMC during compression molding. The underlying models used by the simulation require that the part thickness (Z-direction) be small in comparison to the lateral dimensions (X and Y-directions). Since many applications using SMC meet this requirement (i.e., an automotive hood), the simulation developed could be useful for problems such as designing initial charge locations and the elimination of knit lines.

For other applications, however, a simulation which is based on a thin-charge flow model may not be appropriate. The nature of certain part geometries, for example, may inhibit the use of a thin charge relative to the lateral dimensions (i.e., an automotive bumper). Moreover, for analyses which emphasize flow into deep substructure regions and the formation of sink marks, a simulation which accounts for flow in the Z-direction...
may be preferred. For these cases, material anisotropy and fiber orientation and
distribution become important considerations. A computer program applicable to flow
in the X-Z direction (plane-strain flow) has been developed elsewhere [70].

An extension of the existing flow simulations might be to develop a method to
link the simulation for X-Y flow with the simulation for X-Z flow, resulting in a
package capable of simulating 3-D flow in thick parts.

6.2 Heat Transfer and Curing

The heat transfer model developed in section 3.2 included terms to account for
the most significant contributions to the heat transfer during SMC compression molding;
i.e., the unsteady state term, heat conduction in the thickness direction, and a source
term due to chemical reaction. Mold closure and convection were implicitly accounted
for by including the convection term in the Z-direction and then using the warped time
approach to simplify the resulting equation to a form easily solved using the finite-
difference method. An extension of this approach to explicitly include convection and
conduction in three dimensions may be possible and is described in Appendix C [71].

The polymerization of the unsaturated polyester/styrene system was simulated
using the kinetic model proposed by Stevenson [56]. This model was used because it
includes the radical concentration in addition to the reaction rate term, and the radical
concentration in turn could be used to model the volumetric changes of the system. It
was found, however, that the curing reaction of the UPE/styrene system does not
exactly follow the traditional free radical chain polymerization mechanism. Instead, the
radical concentration measured by ESR attained a significant amount of its final or
maximum value before the conversion began to increase as measured by DSC. Moreover, as shown in Appendix B, the dependence of the propagation rate constant on the conversion is bell-shaped, in contrast to that predicted by equation (5.14). An extension of the kinetic model used would be to address these areas.

6.3 Volume Change Model for UPE/styrene System

A model was developed to predict the volume changes which occur during the cure of an unsaturated polyester/styrene system. The onset of polymerization shrinkage and the increase in radical concentration both began to occur prior to any significant increase in resin conversion, and then both continued throughout the reaction. The early stages of polymerization shrinkage were accommodated in the model using the radical concentration. The model is also able to predict thermal expansions and/or contractions experienced by the system. Predictions obtained from the model were compared to experimental results which incorporated a non-isothermal temperature profile, and reasonable agreement was obtained.

For SMC parts, an ultimate goal of a volume change model is the ability to predict local part shrinkage and sink marks on the cosmetic surface. To achieve this goal, future work in this area should involve inclusion of a term to account for the pressure effect, or compressibility, on the volume change. This could be done by adding the compressibility to the model [65]:

\[ \kappa = -\frac{1}{V_0} \frac{\partial V}{\partial P} \]  

(6.1)
The effect of fiber orientation and distribution on the volumetric behavior of the part should also be included. A complete treatment would also incorporate the effects of any thermal or residual stresses and strains. Additional model verification could then be done by comparing predictions to the results of non-isothermal press molding experiments conducted under high pressure using SMC.
REFERENCES


57. Hershey, H. C., Professor of Chemical Engineering, The Ohio State University, Columbus, OH, Non-linear Pattern Search Subroutine.

58. DianaChart Data Acquisition Package, DianaChart, Inc., 129 Hibernia Avenue, Rockaway, NJ, 07866.


APPENDIX A

Fortran Code for Compression Molding
Simulation Program
Program name: SMC.FOR
Last update: January 8, 1992
This program is used to simulate the SMC compression molding process for thin parts.
Input the initial charge on elements and assume the charge will not flow before the f value of the first control volume (based on nodes) inside this region reaches 1.0.
The final thickness of the product is uniform.

common/sa/x(1800),y(1800),z(1800)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/se/kf(1800),fht(1800),f(1800)
common/sh/xs1(1800,10),ys1(1800,10),cx1(1800,10),cy1(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/sj/ncv(1800)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
common/boud/ibc(1000),n
common/monitr/dt,adt
common/far/areat(1800)
common/dim/jz
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/two/dlo(100),diag(100),dup(100),ans(100),soln(100)
common/three/istart,iz
common/four/treed,tadt
common/five/height(10),nh0,ielem(2000),ihe(10),ih(10,1000),time
common/itcrt/accu,taccu,erri,tha,thetac
common/ini/vol,achrg,dti,delt
common/flow/frate(2000),mnf,mdf(1000)
common/che/ndcoe(1800,10),coef(1800,10),factor
common/curing/dtc,mature
common/rh/cr,kro,cp,alfa,tvlow,tvup,haug(2000),ccp
common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/cppty/rh,chk1,chk2,chk1,chk2,g(1800,45)
common/edmi/edgmi(2000)
dimension fx(1800,10),fy(1800,10),pcl(1800,10),pc2(1800,10)
dimension sumc(1800),spc(1800),pold(1800),hmust(2000)
dimension sl(10),slx(2),sly(2),vold(2000)
dimension mdr(1000),flow(1000),hpc(100)
dimension ot(1800,3),ocv(1800,33),ocb3(33),rcov(33),ocom(33)
character fgeo*15,fchrg*15,fana*15

write(6,801)
801 format('please enter name of geometry file: ','/)
read(5,802)fgeo
write(6,803)
803 format('please enter name of charge configuration: ','/)
read(5,802)fchrg
802 format(a)

open(unit=4,file=fgeo,status='old')
open(unit=2,file=fchrg,status='old')

c-------------------------------------------------------------
c read coordinates of nodes, total number of nodes (nde) and
c elements (nel), and the connectivity of elements

call data
c-------------------------------------------------------------
c read the information about the upper mold
c-----------------------------------------------
call upper
c-------------------------------------------------------------
c initialize the pressures and velocities to be zero
c-----------------------------------------------

do 1 i=1,nde
   p(i)=0.0
1 continue

   do 2 i=1,nel
      u(i)=0.0
      v(i)=0.0
      w(i)=0.0
   2 continue

c  imfull = 0 means mold is not full yet
c  = 1 means mold is full

   imfull = 0
   itab = 0
   nstep = 0
   time = 0.0
   isweep = 0
   icount = 0
   istop = 0
   tprint = 10.
   tinic = 10.
   total = 180.
material properties and mold wall temperatures...
ck is the thermal conductivity of smc, assumed constant...

ck = 0.3682
rho = 1850000.
cp = 1.1171
ccp = rho*cp
alfa = ck/(rho*cp)
rh = 412.7
tlow = 25. + 273.15
twup = 25. + 273.15
accu = 1.e-04
taccu = accu/nde
thetac = 1.0
thetae = 1.0
jz = 41

crit is the convergence criteria for the pressure iteration
crit = 0.1e-5

x, y are the global coordinates, and their indices
c which i is the element number and j is from 1 to 3.
c xnode and ynode are the local coordinates, and their
c indices are i (element number) and j (from 1 to 3)
c ktype=1 : when the whole domain is a flat plate, then the
c global coordinates (x,y) can be used directly
c as the local coordinates (xnode,ynode)
c ktype=2 : the domain "surface" is not on the x-y plane, and
c therefore a transformation has to be performed
c from (x,y,z) to (xnode,ynode)
ktype=2

if(ktype.eq.1)
do 11 i=1,nel
do 11 j=1,3
 xnode(i,j)=x(nep(i,j))
 ynode(i,j)=y(nep(i,j))
11 continue
else
call redu3
endif

c check the areas of the elements and make sure the order
c of the indices is in the counterclockwise direction;
c construct the control volumes.
call elegeo
call setup1
obtain the information about the initial charge,
the list of occupied elements, height, etc...
call bounc

initialize the viscosities and temperatures

do 738 i = 1, nel
    do 738 j = 1, 60
        visc(i,j) = 1000000.
    continue

do 416 i = 1, nde
    t(i,1) = twlow
    t(i,jz) = twup
    do 416 j = 2, jz-1
        t(i,j) = 25. + 273.15
    continue

write(6,805)
write(6,*)'total elements= ', nel
write(6,*)'total nodes= ', nde
write(6,*)'nh0 = ', nh0
write(6,*)'n = ', n

do 12 i = 1, nh0-1
    write(6,804)i, ihn(i)
804 format(' ', 'ihn(', iii, ') = ', i4)
12 continue

write(6,805)
805 format(' ', '//')

cinicize the viscosities and temperatures

--start the calculations of mold filling

500 loop=0

calculate mobility(s). it changes with h(height). since h is a
function of time, s therefore has to be recalculated when the
upper mold advances. consequently, fx, fy, pc1, pc2, spc also
have to be recalculated when the upper mold's position changes
(large loop).
however, these calculations do not have to be inside the
pressure iteration (small loop).
do 13 i = 1, nel
   call mobil(i)
13   continue

c do 19 i = 1, nde
    spc(i) = 0.
 do 14 j = 1, ncv(i)
    k = num(i,j)
    k0 = ncel(i,j,7)
    l1 = ncel(i,j,1)
    l2 = ncel(i,j,2)
    fx(i,j) = s(k)*(xs1(i,j)+xs2(i,j))/(2.*area(k))
    fy(i,j) = s(k)*(ys1(i,j)+ys2(i,j))/(2.*area(k))
    pc1(i,j) = fx(i,j)*b(k,l1) + fy(i,j)*c(k,l1)
    pc2(i,j) = fx(i,j)*b(k,l2) + fy(i,j)*c(k,l2)
    spc(i) = spc(i) + fx(i,j)*b(k,k0) + fy(i,j)*c(k,k0)
14   continue
19   continue

/*---------------------------------------------*/
c start the iteration for each time step
write(6,*) 'mn= ', mn
100 continue
   loop=loop+1
write(6,*) 'loop= ', loop

if(loop.ge.150000) then
   write(6,*) 'loop=150000, stop calculation'
   stop
endif

/*---------------------------------------------*/
do 15 ii = 1, mn
   i=ncal(ii)
\texttt{sumc(i) = 0.}

\texttt{do 15 j = 1, ncv(i) }
\texttt{
\hspace{1cm} k = num(i, j) \\
\hspace{1cm} l1 = ncel(i, j, 1) \\
\hspace{1cm} l2 = ncel(i, j, 2) \\
\hspace{1cm} sumc(i) = sumc(i) + pc1(i, j)*p(nep(num(i, j), l1)) \\
\hspace{1cm} & + pc2(i, j)*p(nep(num(i, j), l2))
}
\texttt{15 continue}

\texttt{--------------------------------------------------------------------------------------------------------------------}
\texttt{under-relaxation factor = 'factor' }
\texttt{input in subroutine 'bounc' }

\texttt{do 21 i = 1, mn}
\texttt{\hspace{1cm} ip = ncal(i) \\
\hspace{1.5cm} pold(ip) = p(ip) \\
\hspace{1.5cm} p(ip) = (frate(ip) - sumc(ip))/spc(ip) \\
\hspace{1cm} c write(25,*)'mn= ',mn \\
\hspace{1cm} c write(25,806)ip,frate(ip),sumc(ip),p(ip) \\
\hspace{1cm} 806 format('frate,sumc,p(ip)=',i4,3e16.6) \\
\hspace{1cm} p(ip) = pold(ip)+(p(ip)-pold(ip))*factor \\
\hspace{1cm} if(p(ip).lt.0.)then \\
\hspace{1.5cm} write(6,*)'frate,sumc,p(ip)=' ,ip,frate(ip),sumc(ip),p(ip) \\
\hspace{1.5cm} write(6,*)'negative pressure' \\
\hspace{1.5cm} do 22 jj=1,mn \\
\hspace{2cm} iip=ncal(jj) \\
\hspace{2cm} write(6,*)'ncal= ',iip \\
\hspace{2cm} 22 continue \\
\hspace{1cm} call check1 \\
\hspace{1cm} call output1 }
\texttt{stop}
\texttt{end if}
\texttt{21 continue}

\texttt{--------------------------------------------------------------------------------------------------------------------}

\texttt{do 31 i=1,mn}
\texttt{\hspace{1cm} error=(p(ncal(i))-pold(ncal(i)))/p(ncal(i)) \\
\hspace{1cm} c write(6,*)'error of pressure(',error,ncal(i),p(ncal(i)) \\
\hspace{1cm} if(abs(error).ge.crit) goto 100 \\
\hspace{1cm} continue \\
\hspace{1cm} write(6,*)'mn= ',mn}
write(6,*)'iterations for pressure= ', loop, ' loop'
write(6,*)'error of pressure(%) ', error, ncal(i), p(ncal(i))

end of iteration for each time step

---------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------
calculate velocity

---------------------------------------------------------------------------------------------------------------
call velocity

---------------------------------------------------------------------------------------------------------------
output the pressure and velocity fields.
delt=(tadt-dti)/18.0
the whole process is divided into 18 parts.
output the results when: 1. time >= dti+1.0*delta,
and 2. time >= dti+(n-0.75)*delta where n=2,3,4, ....

if(time.gt.trecd) then
  itab = itab + 1
  nstep=nstep+1
  call output1(nstep, itab)
call output4(nstep, itab)
tch=3.0*delt/4.0
32  trecd=trecd+delt
  if((trecd-tch).lt.time) go to 32
endif

determine the control volumes in the new flow front
and the next time step

tmin=1.e20
asum = 0.0

do 41 i = 1, mn
  ip = ncal(i)
do 41 j=1,ncv(ip)
  k=num(ip,j)
  asum = asum + hdotlo(k)*area(k)/3.0
41  continue

do 42 i = 1, mnf
  mdl = mdf(i)
142

    volcv(mdl) = 0.0
    do 42 j = 1, ncv(mdl)
        k = num(mdl,j)
        volcv(mdl) = volcv(mdl) + area(k)*h(k)/3.
    continue

42    fsum = 0.0
    ineg = 0
    write(6,*)'mnf= ',mnf
    write(20,*)'mnf= ',mnf
    do 51 i=1,mnf
        if( f(mdf(i)) .eq. 1. ) go to 51
        dvolr=0.0
        flow(mdf(i))=0.
        mdl=mdf(i)
        do 52 j=1,ncv(mdl)
            len=num(mdl,j)
            k0=ncel(mdl,j,7)
            k1=ncel(mdl,j,1)
            k2=ncel(mdl,j,2)
            k3=nep(len,k1)
            k4=nep(len,k2)

            flow(mdl)=flow(mdl)
            & +fx(mdl,j)*b(len,k0)+fy(mdl,j)*c(len,k0)*p(mdl)
            & +(fx(mdl,j)*b(len,k1)+fy(mdl,j)*c(len,k1))*p(k3)
            & +(fx(mdl,j)*b(len,k2)+fy(mdl,j)*c(len,k2))*p(k4)

        dvolr=dvolr+hdotlo(len)*area(len)/3.0
    continue

52    write(6,*)'flow(mdl)= ',mdl,flow(mdl)
    flow(mdl) = -flow(mdl)
    if( flow(mdl) .lt. 0.0 )then
        write(6,*)'node: ',mdl,' flow: ',flow(mdl),', f= ',f(mdl)
        ineg = ineg + 1
    end if

    there are two kinds of mnf nodes (partially filled
    control volumes); one type with a certain amount of "flow"
    because of the pressure difference with any neighboring
    completely filled control volumes, and one with zero "flow".
    the f value of the latter will still change because of
    the closing of upper mold (even though there is no "flow" into
    this control volumes).
    therefore both of these two kinds of partially filled control
    volumes have a predicted dt.
    the first kind of node has two "sources" to increase the
f value, while the second kind can only depend on the closing upper mold speed. However, if originally there were more charge in the second kind of control volume, with a single "source", it could reach $f=1.0$ faster than the first kind.

```fortran
if(flow(mdl).le.1.e-25) then
    flow(mdl)=.0.
dt=1.e20
goto 51
endif

fsum = fsum + flow(mdl)
dt=(1.-f(mdl))*volcv(mdl)/(flow(mdl)+dvolr)
write(20,*)'dt ',mdl,dt
if(dt.le.0.) then
    write(6,*)'*** dt < 0 ',i,j,f(mdl),flow(mdl)
endif
if(abs(dt).lt.tmin) tmin = dt
51 continue

cor = asum/fsum
```

```fortran
favg = fsum/(mnf - ineg)
if( ineg .ne. 0 ) write(6,*)'favg=',favg
dt = tmin
tmin = 1.e20
time = time + dt
```

```fortran
call shapvel(courant)
call enrcnvr(courant)
```

```fortran
do 53 i = 1, nel
   h(i) = h0(i) - hdotlo(i)*time
53 continue
```

```fortran
do 54 i = 1, mnf
   mdl = mdf(i)
   vold(mdl) = volcv(mdl)
   volcv(mdl) = 0.0
```

do 54 j = 1, ncv(mdl)
  k = num(mdl,j)
  volcv(mdl) = volcv(mdl) + h(k)*area(k)/3.
54 continue

---------------------------------------------------------------------
c advance the flow front

mnp=0
kr=0

do 61 i=1,mnf
  mdl=mdf(i)
  if(f(mdl).lt.0.49999999999999) then
    fht(mdl)= ( 0.5-f(mdl) )*vold(mdl)/(flow(mdl)+frate(mdl))
    fht(mdl)=fht(mdl)+time-dt
  end if
  if(fht(mdl).lt.0.0) then
    write(6,*),'fht(',mdl,') less than zero '
    write(6,*),'fht=',fht(mdl),'
    write(6,*),'vold=',vold(mdl),'
    write(6,*),'flow=',flow(mdl)
    write(6,*),'time=',time,'
    write(6,*),'dt=',dt
  endif
  f(mdl) = ( f(mdl)*vold(mdl) + flow(mdl)*dt )/volcv(mdl)
if(f(mdl).gt.1.000001) write(6,*),'----> f > 1 ',mdl,f(mdl)

---------------------------------------------------------------------
c ncp(mnp) is the list of node numbers of the "new" filled nodes (CV)
c mnp is the total number of new filled nodes; it is changed after
c every flow advancement and it is recounted each time

if(f(mdl).gt.0.9999) then
  kf(mdl)=2
  p(mdl)=0.01
  mnp=mnp+1
  ncp(mnp)=mdl

  do 62 j=1,ncv(mdl)
    do 63 k=1,3
      if(kf(nep(num(mdl,j),k)).eq.0) then
        kf(nep(num(mdl,j),k))=1
        kr=kr+1
        mdr(kr)=nep(num(mdl,j),k)
      endif
  63 continue
  62 continue

  continue
c mv is an index used to indicate the elements which are filled.
c mv(kj) is the list of element numbers, and kj is the total number.
c mdr(kr) is the list of node numbers of the "new" moving front.
c kr is the new total number; it is changed after every
c flow advancement and it is recounted each time.

                do 64 lv = l, ncv(md1)
                if(kelem(num(md1,lv)) .eq. 0) then
                    kj = kj + 1
                    mv(kj) = num(md1,lv)
                    kelem(num(md1,lv)) = 1
                endif
                64 continue

                else

                    kr = kr + 1
                    mdr(kr) = md1

                endif

    c --mass balance verification--

                call mass(mnp,npc,kr,mdr,cor)

    if( mn .ge. (nde-2) ) imfull = 1

    c the mold is full -- write the flow front positions
    c to the data file and exit to "temperature" routine...

                if( imfull .eq. 1 ) then
                    call check1
                    call output1(nstep, itab)
                    call output2
                    call output4(nstep, itab)
                    go to 200
                endif

                go to 500

    c --begin curing routine--
200  time = time + dtc
      if ( time .gt. total ) stop

      call cure

      if ( time .ge. tprint ) then
        call output4(nstep, itab)
        tprint = tprint + tinc
      end if

      go to 200

end

c==================================
c==================================
c

subroutine data

common/sa/x(1800),y(1800),z(1800)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
991  continue

read(4,'(i6)',end=993,err=991) nc

if(nc.eq. -1) then
  read(4,'(i6)') nk

  if(nk.eq.15) then
    nde=0
  10    read(4,200,end=993) n,k1,k2,k3,cx,cy,cz
    if(n.eq. -1) then
      go to 991
    else
      nde=nde+1
      x(n)=cx
      y(n)=cy
      z(n)=cz
    end if
  go to 10
end if

else if(nk.eq.71) then
  nel=0
  20    read(4,400,end=993) n,k1,k2,k3,k4,k5,k6
    if(n.eq. -1) go to 991
    read(4,401,end=993) k7,k8,k9
  400  format(7i10)
  401  format(3i10)

  nel=nel+1
nep(n,1)=k7
nep(n,2)=k8
nep(n,3)=k9

go to 20
else
   continue
   read(4,'(i6)',err=992) ne
   if (ne.eq.-1) then
      go to 991
   endif
else
   go to 991
endif

993 continue

c= close(4)
return
end

==============================================================================

subroutine upper

common/one/visc(2000,60),imfull,hdot,h0(2000),hmax


c input the upper mold advancing speed and its advancing direction
c vector (it will be a unit vector after calculation)
c hdot : mold closing speed (meters/sec)

read(2,*),hdot
read(2,*),cosx,cosy,cosz

c= sqrt(cosx**2+cosy**2+cosz**2)
cosx=cosx/c=0

cosy=cosy/c=0

cosz=cosz/c=0

return
end

==============================================================================
subroutine redu3

common/as/a(1800),y(1800),z(1800)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/tran/trmax(2000,6)
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax

cmin=1.0e9
cmax=-1.0e9
c1=0.0

do 10 i=1,nel
  xl=x(nep(i,1))
x2=x(nep(i,2))
x3=x(nep(i,3))
y1=y(nep(i,1))
y2=y(nep(i,2))
y3=y(nep(i,3))
z1=z(nep(i,1))
z2=z(nep(i,2))
z3=z(nep(i,3))
x1=(y2-y1)*(z3-z1)-(z2-z1)*(y3-y1)
ym=(z2-z1)*(x3-x1)-(x2-x1)*(z3-z1)
zn=(x2-x1)*(y3-y1)-(y2-y1)*(x3-x1)
tol=sqrt(xl*xl+ym*ym+zn*zn)
  if(abs(tol).lt.1.e-15) then
    write(6,*)'tol=0',tol,nel,x1,x2,x3,y1,y2,y3,z1,z2,z3
  endif
  now calculate unit normal vector (xl,ym,zn) for each element
  and the direction cosine (cosine(i)) of each element with the
  unit vector (cosx,cosy,cosz) of the upper mold advancing
direction.
hdotlo(i) is the upper mold advancing speed of each element
since we already change the "sign" of the direction cosine
hdotlo ( local h dot ) should be positive

  x1=x1/tol
  ym=ym/tol
  zn=zn/tol
  rew=x1*xl+ym*ym+zn*zn
  cosine(i)=x1*cosx+ym*cosy+zn*cosz
  c1=c1+cosine(i)
write(21,201)i,xl,ym,zn,cosine(i)
format('*(xl,ym,zn)= ',i5,3f13.6,' cosine= ',f13.6)

if(cosine(i).lt.cmin) then
  cmin=cosine(i)
  imin=i
endif

if(cosine(i).gt.cmax) then
  cmax=cosine(i)
  imax=i
endif

hdotlo(i)=hdot*cosine(i)

if( hdotlo(i) .lt. 0.0 ) then
  write(6,'*')'hdotlo(\i,\i)=' ,hdotlo(i)
  hdotlo(i)=-hdotlo(i)
  cosine(i)=-cosine(i)
endif

c

if(xl.eq.0..and.ym.eq.0.) then
  wl=1.
  wm=0.
  wn=1.
else
  wl=(ym*ym+zn*xl*xl)/(ym*ym+xl*xl)
  wm=(ym*xl*(zn-1.))/(ym*ym+xl*xl)
  wn=(xl*xl+zn*ym*ym)/(ym*ym+xl*xl)
endif

trmax(i,1)=xl
trmax(i,2)=ym
trmax(i,3)=zn
trmax(i,4)=wl
trmax(i,5)=wm
trmax(i,6)=wn

do 20 k=1,3
   xnode(i,k)=wl*x(nep(i,k))+wm*y(nep(i,k))-xl*z(nep(i,k))
   ynode(i,k)=wm*x(nep(i,k))+wn*y(nep(i,k))-ym*z(nep(i,k))
20 continue

continue

write(21,*)'the min. of cosine is element ',imin,cmin
write(21,*)'the max. of cosine is element ',imax,cmax
write(21,*)'the ave. of cosine is ',cl/nel

return
end
subroutine setup1

common/boud2/nih(10,1000),nn,hh(1000)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/se/kf(1800),fht(1800),f(1800)
common/sh1/xs1(1800,10),ys1(1800,10),cx1(1800,10),cy1(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/sj/ncv(1800)
common/const/n,mcold,nnew(1800)
common/monitrdt,adt
common/area(1800)
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/three/istart,iz
common/four/tred,tadt
common/five/height(10),n10,ielem(2000),ihn(10),ih(10,1000),time
common/ini/vol,achrg,dti,delt
common/flow/frate(2000),mnf,mdf(1000)
common/edmi/edgmi(2000)
dimension fx(1800,10),fy(1800,10),pc1(1800,10),pc2(1800,10)
dimension sl(10),sli(2),sly(2),vold(2000)
dimension volcv(2000)

do 1 i=1,nel

c----- calculating the coordinate of the center of each element

cx(i)=xnode(i,1)+xnode(i,2)+xnode(i,3)
cx(i)=cx(i)/3.

cy(i)=ynode(i,1)+ynode(i,2)+ynode(i,3)
cy(i)=cy(i)/3.

compute 2* element area -------

a2=xnode(i,2)*ynode(i,3)+xnode(i,3)*ynode(i,1)
& +xnode(i,1)*ynode(i,2)
& -xnode(i,2)*ynode(i,1)-xnode(i,3)*ynode(i,2)
& -xnode(i,1)*ynode(i,3)
area(i)=a2/2.0
write(22,*)'area ',i,area(i)

c---- check the sequence of nodes in an element. if it is c.w.,
c---- change the sequence to c.c.w.

if(area(i).le.0.) then
  ilter=nep(i,2)
  nep(i,2)=nep(i,3)
  nep(i,3)=ilter
  xalter=xnode(i,2)
  yalter=ynode(i,2)
  xnode(i,2)=xnode(i,3)
  ynode(i,2)=ynode(i,3)
  xnode(i,3)=xalter
  ynode(i,3)=yalter
  area(i)=-area(i)
  cosine(i)=-cosine(i)
  hdotlo(i)=-hdotlo(i)
  icount=icount+1
  write(6,*),'the order of the node have been changed
    in the',i,'element'
endif
write(23,801)i,area(i),cosine(i),hdotlo(i)
801  format('element ',i4,' area= ',f13.6,' cosine= ',f13.6,
     & ' hdotlo= ',f13.6)
1    continue

c ---------------------------------------------------------------

if( icount.ne. 0 ) write(6,802) icount
802  format('(),' the order of the nodes has been changed ',
     & 'for ',i4,' elements...//')

do 2 i=1,nel
  b(i,1)=ynode(i,2)-ynode(i,3)
  b(i,2)=ynode(i,3)-ynode(i,1)
  b(i,3)=ynode(i,1)-ynode(i,2)
  c(i,1)=xnode(i,3)-xnode(i,2)
  c(i,2)=xnode(i,1)-xnode(i,3)
  c(i,3)=xnode(i,2)-xnode(i,1)
2    continue

c ---------------------------------------------------------------

c construct the control volume

c in do loop 10,
c (xs1,ys1) and (xs2,ys2) are the unit normal

c vectors times the corresponding lengths for

c each subdomain of the specified control

c volume (represented by node number i)

do 10 i=1,nde
  ncv(i)=0
do 12 j=1,nel
   do 14 k=1,3
      km=0
      if(nep(j,k).eq.i) then
         ncv(i)=ncv(i)+1
         num(i,ncv(i))=j
         do 15 kk=1,3
            if(kk.eq.k) then
               bx=xnode(j,k)
               by=ynode(j,k)
               bb(i,ncv(i))=b(j,kk)
               cc(i,ncv(i))=c(j,kk)
            else
               km=km+1
               ncel(i,ncv(i),km)=kk
               ax(km)=xnode(j,kk)
               ay(km)=ynode(j,kk)
            endif
         15 continue
      xmid=(ax(1)+ax(2))/2.
      ymid=(ay(1)+ay(2))/2.
      dirx=bx-xmid
      diry=by-ymid
      sdx=cx(j)-(bx+ax(1))/2.
      sdy=cy(j)-(by+ay(1))/2.
      xy=sdx*dirx-sdy*diry
      cx1(i,ncv(i))=(bx+ax(1))/2.
      cy1(i,ncv(i))=(by+ay(1))/2.
      if(xy.lt.0.) then
         xs1(i,ncv(i))=-sdy
         ys1(i,ncv(i))=sdx
      else
         xs1(i,ncv(i))=sdy
         ys1(i,ncv(i))=-sdx
      endif
      sdx=cx(j)-(bx+ax(2))/2.
      sdy=cy(j)-(by+ay(2))/2.
      xy=sdx*dirx-sdy*diry
      cx2(i,ncv(i))=(bx+ax(2))/2.
      cy2(i,ncv(i))=(by+ay(2))/2.
      if(xy.lt.0.) then
         xs2(i,ncv(i))=-sdy
         ys2(i,ncv(i))=sdx
      else
         xs2(i,ncv(i))=sdy
         ys2(i,ncv(i))=-sdx
      endif
      go to 12
end if
14 continue
12 continue
10 continue

return
end

subroutine bounc

common/boud/ibc(1000),n
common/boud2/nih(10,1000),nn,hh(1000)
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/five/height(10),nih,ielem(2000),ihn(10),ih(10,1000),time
common/sb/xnode(2000,3),ynode(2000,3),nnde,nel
common/se/kf(1800),fht(1800),f(1800)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/sj/nov(1800)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
common/const/mn,mmold,nnew(1800)
common/six/narea,nmrt(10),href,iref
common/ini/vol,achrg,dti,delt
common/flow/frate(2000),mfn,mdf(1000)
common/che/ncoe(1800,10),coef(1800,10),factor
common/dim/j2
common/rad1/rad(1800,45),prr(1800,45),prh(1800,45)
common/rad2/ri(1800,45),pri(1800,45),rih(1800,45)

dimension hica(2000)
dimension volcv(2000)
dimension volcv(2000)

c initial charge can have nn regions of different initial height
c hh. hic is the real height of Initial Charge for different
c elements (there could be a number of nodes in different regions).
c there are nhc elements in each region, and the list of elements
c is represented by nih(i,j).
c when we "back up" the upper mold in a specified direction, the
c height of the gap (the "perpendicular" distance between upper
c and lower molds) for "each element" will not be the same if the
c mold surface is not a plane.
c now consider each "element" separately;
c for a specified element, if we put an initial charge (the
height is hic) on it, then we have to move the mold a distance of "hic" in the direction we specified, since we can put initial charge of different heights on the different elements and each element has a different direction cosine. therefore the distance which the upper mold should be moved should be the maximum hic. then initially the upper mold "theoretically" will just "touch" one (or more than one simultaneously) element with occupied fraction factor "f" to be 1.0 without "squeezing" the charge.

read under-relaxation factor used for pressure iteration
in main program
read(2,801) factor
801 format(f4.1)

initially there is no charge in the mold
and no "element" is filled, kelem=0

do 1 i=1,nell
   hic(i)=0.0
   kelem(i)=0
1 continue

read(2,10) nn

do 2 k = 1, nn
   read(2,40) hh(k)
2 continue

hmmax=1.0e9

mv is an index used to indicate the elements filled.
mv(kj) is the list of element numbers and kj is the total number.
kj=0

do 3 i=1,nn
   read(2,20) nic
   do 3 j = 1, nic
      read(2,20) nih(i,j)
      ip=nih(i,j)

      kj=kj+1
      mv(kj)=ip
      kelem(ip)=1

      hic(ip)=hh(i)
hica(ip)=hic(ip)/cosine(ip)
if(hica(ip).gt.hmmax) then
   hmmax=hica(ip)
   index=ip
endif
3 continue

write(6,*)'the element used as hmax is ', index

do 4 i=1,nell
   h0(i)=hmmax*cosine(i)
nn is the total number of nodes in the initial charge region(s)

ip = nih(i,j) is each element number

achrg (charge area) is the area occupied by the initial charge

vol (charge volume) is the volume occupied by the initial charge

vol = 0.0
achrg = 0.0
dvol = 0.0
dvolr = 0.0

do 11 i = 1, nn
   do 11 j = 1, nic
      ip = nih(i,j)
      achrg = achrg + area(ip)
      vol = vol + area(ip) * hic(ip)

11 continue

in the following do loop, we only want to find the node(s)
(control volumes) which have already been filled (f=1.0).
we assign kelem=1 to all the elements occupied by the
initial charge. We also assume that before "the first"
control volume is filled, the charge will not flow out of
this region (the elements occupied by the initial charge),
so initially set the control volume with fmax to be f=1.
notice that initially we set all the kf=0, f=0.0.
the f in do loop 12 is the initial f before the upper mold
has any motion at all (that is the upper mold only contacts
the initial charge at "one point", without any compression).

fmax = -1.0e10

fm a x = -1.0e10

12 continue
   f(i) = volcv(i)/volcv(i)
13 continue
if( f(i).gt.fmax ) then
   fmax = f(i)
   index = i
endif

f(index) = 1.0
k f(index) = 2
mn = index
ncal(1) = index
write(6,*),'mn = ', mn

nn is the total number of nodes in the initial charge region(s)
ip = nih(i,j) is each element number
cachrg (charge area) is the area occupied by the initial charge
cvol (charge volume) is the volume occupied by the initial charge
dvol = 0.0
dvolr = 0.0
do 14 i = 1, mn
    ip = ncal(i)
do 14 j = 1, ncv(ip)
    mel = num(ip,j)
    dvol = dvol + (h0(mel) - hic(mel)) * area(mel) / 3.0
    dvolr = dvolr + hdotlo(mel) * area(mel) / 3.0
14 continue
dti = dvol / dvolr
c write(6,*) 'dti = ', dti
do 19 i = 1, nde
c write(6,*) 'f = ', i, f(i)
c19 continue
time = dti
c when the upper mold starts in the initial location (just touching
c the initial charge before "squeezing"), calculate the initial
c height of each element
do 15 i = 1, nel
    h(i) = h0(i) - hdotlo(i) * dti
15 continue
c right now we have already advanced the upper mold to a position
c where there is only one control volume (node) filled.
c so we have to recalculate all the f values
c and at the same time, we assign all the partially filled
c control volumes to be kf=1 and count them into mnf.
c there are two kinds of control volumes counted into
c mnf. one is the definition of moving flow front which consists of
c the control volumes just next to the completely filled
c ones. the second kind is also partially filled, however
c they are not in the immediate location of the completely
c filled ones. therefore, the pressures on these nodes are
c zero, and their neighboring pressures are also zero. then,
c their f values will only be changed because of closing
c the upper mold. however, since there are no pressure gradients
c there will not be any "flow" flow into these control volumes.
c but notice that their f values do change.
c 1. pressure is only calculated for mn nodes, that is only
c completely filled control volumes (f=1.0)
c 2. "flow" depends on pressure. in do loop 51, it goes
c through two kinds of partially filled control volumes.
c only the real moving flow front will get non-zero "flow".
c for the other kind the "flow=0.0", because their pressure
c and their neighbors' are all zero.
c next, assign the initial initiator and inhibitor
concentrations...

do 77 i = 1, nde
   if ( f(i) .gt. 0.0 ) then
      do 78 k = 1, jz
         if ( f(i,k) .gt. 0.0 ) then
            rii(i,k) = 0.05576
            rih(i,k) = 2.0803755e-04
            rad(i,k) = 0.0
            continue
         endif
      enddo 78
      continue 77
   endif
   mnf=0
   do 16 i=1,nnde
      volcv(i) = 0.0
      volci(i) = 0.0
   enddo 17
   do 17 j=1,ncv(i)
      mel=nuin(i,j)
      volcv(i) = volcv(i) + area(mel)*h(mel)/3.
      volci(i) = volci(i) + area(mel)*hic(mel)/3.
      continue 17
   f(i)=volcv(i)/volci(i)
   if ( f(i).gt.0.0 .and. f(i).lt.1.0 ) then
      kf(i)=1
      mnf=mnf+1
      mdf(mnf)=i
   endif
   continue 16

write(6,')'dtn= ',dtn
write(6,')'fil= ',fil

assign the first completely filled control volume
   f(index)=1.0
   kf(index)=2
   mn=1
   ncall(1)=index
   write(6,')'mn= ',mn

number of profiles (nodes) to be printed, and node numbers
read(2,10) nparea
   do 460 i = 1, nparea
      read(2,20) nprint(i)
   continue 460

number and value of different initial height regions
make sure that height(1) represents the main region
read(2,10) nh0
   do 490 i = 1, nh0

read(2,40) height(i)
c 490  continue

assign height values to all the elements ( initially
c they all have the same height, height(1).
c
do 502  i = 1, nel
  h0(i) = height(1)
  h(i) = height(1)
c 502  continue

c use the first node to be printed as the reference node
( npnt(1) is its node number and its height, height(1)
c is the reference height )

href = height(1)
iref = npnt(1)

c control volumes in the initial charge
n is the total nodes in the initial charge region
and ibc(j) is the node number

c read(2,20) n
do 500  j = 1, n
  read(2,20) ibc(j)
c 500  continue

nh0 = 1 means the whole domain has a uniform height

if( nh0 .eq. 1 ) return

c elements in the "substructures".
c if there is a substructure in the domain, that is if nh0
c is greater than 1, then we have to assign the node
c numbers and their corresponding heights.
c there are nh0-1 substructures.
c each substructure has ihn(i) nodes.
c each node number in this specified substructure is
c represented by ih(i,j), and their corresponding heights
c are h0(ih(i,j)) ( or h(ih(i,j)) ).

do 501  i = 1, nh0-1
  read(2,20) ihn(i)
do 501  j = 1, ihn(i)
  read(2,30) ih(i,j)
  h0(ih(i,j)) = height(i+1)
  h(ih(i,j)) = height(i+1)
c 501  continue

t  format(i1)
20  format(i3)
30  format(i4)
40  format(f11.10)

c 600  close(2)
return
end

subroutine setup2

common/boud2/nih(10,1000),nn, hh(1000)
common/sb/xnode(2000,3), ynode(2000,3), nde, nel
common/sc/nep(2000,3)
common/se/b(2000,3), c(2000,3), p(1800)
common/shl/xs1(1800,10), ysi(1800,10), cx1(1800,10), cy1(1800,10)
common/sh/xs2(1800,10), ys2(1800,10), cx2(1800,10), cy2(1800,10)
common/si/ncal(1800), bb(1800,10), cc(1800,10)
common/sj/ncv(1800)
common/sk/num(1800,10), ncel(1800,10,10), ax(2), ay(2)
common/boud/ibc(1000), n
common/const/mn, mnold, nnew(1800)
common/monitr/dt, adt
common/far/area(1800)
common/one/visc(2000,60), imfull, hdot, h0(2000), hmax
common/three/istart, iz
common/four/trecd, tad
common/five/height(10), nh0, ielem(2000), ihn(10), ih(10,1000), time
common/ini/vol, achr, d, dhi, delt
common/flow/frate(2000), mnf, mdf(1000)
common/edmi/edgmi(2000)
dimension fx(1800,10), fy(1800,10), pcl(1800,10), pc2(1800,10)
dimension sl(10), slix(2), sly(2), vold(2000)
dimension volcvi(2000)

c--------calculate the area of a control volume

do 1 i=1, nde
   areacv(i) = 0.
do 1 j=1, ncv(i)
   k=num(i,j)
   areacv(i) = areacv(i) + area(k)/3.
1 continue
do 2 i=1,n
d  
if(ncv(i).ge.10) write(6,*)'***** ncv= ',ncv(i)
continue

afill is the total area of the whole domain
afill = 0.0

do 3 i = 1, n
  afill = afill + area(i)
continue

area(k) : area of element k
h0(k) : the initial height of element k
sumel : total "domain" area
sumelh : total "cavity" volume before beginning to close
         the mold
hdtsum : volume decreasing rate due to closing mold

sumel = 0.0
hdtsum = 0.0
sumelh = 0.0

do 4 k = 1, n
  sumel = sumel + area(k)
  hdtsum = hdtsum + hdotlo(k)*area(k)
  sumelh = sumelh + area(k)*h0(k)
continue

n is the total number of nodes in the initial charge region(s)
ip=ibc(i) is each node number
achrg (charge area) is the area occupied by the initial charge
vol (charge volume) is the volume occupied by the initial charge
hdotlo : local mold closing speed (for each element)
sumel : total domain area
hdtsum : mold closing rate (volume rate)
sumelh : "initial" total cavity volume
vol : total charge volume (constant)
sumelh - vol : the "net" volume to be closed
tadt : the expected time needed to close the mold


tadt = ( sumelh - vol )/hdtsum

delt=(tadt-dti)/18.
trecd=dti+delt

ncv(1) = 10

call output3(afill)
c----determine the initial flow front location

nstep=0

c check, for the control volumes (nodes) of the initial charge
all their f=1.0 and kf=2
do 31 i = 1, m
  j = ncal(i)
  if ( f(j) .ne. 1.0 ) write(6,*) j,'mn-------> ERROR'
  if ( kf(j) .ne. 2 ) write(6,*) j,'mn-------> ERROR'
31 continue

check, for the control volumes (nodes) of the moving flow front
resulting from the initial charge (before closing the mold)
all their f=0.0 and kf=1.

do 32 i = 1, m
  j = mdf(i)
  if ( f(j) .ge. 0.999999) write(6,*)'mnf f(j) » ERROR', j , f(j)
  if ( kf(j) .ne. 1) write(6,*)'mnf ------------> kf(j) » ERROR', j,kf(j)
32 continue

ielem(i) = 1 -----> element i is not in a substructure
ielem(i) = 2 -----> element i is in substructure region A
ielem(i) = 3 -----> element i is in substructure region B

nh0=1

do 33 i = 1, nel
  ielem(i) = 1
33 continue

compute the induced flow rate due to mold closure

do 34 i = 1, nde
  frate(i)=0.0
  do 34 k = 1, ncv(i)
    mel=num(i,k)
    frate(i)=frate(i)+hdotlo(mel)*area(mel)/3.0
    write(7,*)'frate = ',i,frate(i)
 34 continue

return

end

subroutine velocity

common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/tran/trmax(2000,6)
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/five/height(10),nh0,iem(2000),ihn(10),ih(10,1000),time

do 10 i=1,kj
   j=mv(i)
   dpdx=(p(nep(j,1))*b(j,1)
          +p(nep(j,2))*b(j,2)
          +p(nep(j,3))*b(j,3))/(2.*area(j))
   dpdy=(p(nep(j,1))*c(j,1)
          +p(nep(j,2))*c(j,2)
          +p(nep(j,3))*c(j,3))/(2.*area(j))

c compute the local velocities and store them in ut, vt.
   u(j) = -(s(j)/h(j))*dpdx
   v(j) = -(s(j)/h(j))*dpdy
   ut(j) = u(j)
   vt(j) = v(j)
10 continue

c transform the velocities to global coordinates
goto (401,403,402),ktype
402 do 20 i = 1, nel
   uu=u(i)
   vv=v(i)
   u(i)=trmax(i,2)/trmax(i,4)*uu
          +trmax(i,1)*trmax(i,3)/trmax(i,5)*vv
   v(i)=-trmax(i,1)/trmax(i,4)*uu
          +trmax(i,2)*trmax(i,3)/trmax(i,5)*vv
   w(i)=(-trmax(i,1)*trmax(i,1)-trmax(i,2)*trmax(i,2))/trmax(i,5)
20 continue

goto 401
403 do 30 i = 1, nel
   uu=u(i)
   vv=v(i)
   u(i)=trmax(i,4)*uu+trmax(i,5)*vv
   v(i)=trmax(i,5)*uu+trmax(i,6)*vv
   w(i)=-trmax(i,1)*uu-trmax(i,2)*vv
30 continue
401 continue

return
end

-c================================================================================================
subroutine mobil(igo)

common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sj/ncv(1800)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time

dimension z(41), arg(41), volcv(2000), hcalc(2000)

*** this subroutine uses simpson's rule to calculate "s" ***

itr ap = 41

hm = h0(igo) - hdotlo(igo)*time

if ( hm .le. 0.0 )then
    write(6,*), 'hm less than zero!  time = ',time
    stop
end if

write(6,*), 'igo= ', igo, ' h= ', h(igo)

delz = hm/( itr ap - 1 )

z(1) = 0.0

do 100 i = 1, itr ap - 1
    z(i+1) = z(i) + delz
100 continue

write(6,*), 'h/2= ',hm/2.
call lamda(hm,delz,z,itr ap,xlamda,igo)
write(6,*), 'lamda= ',xlamda

arg(1) = (( z(1) - xlamda )**2)/visc(igo,1)
do 377 i = 1, itr ap - 1
   arg(i+1) = (( z(i+1) - xlamda )**2)/visc(igo,i+1)
377 continue

sumevn = 0.0

do 378 j = 2, itr ap - 1, 2
    sumevn = sumevn + arg(j)
378 continue
    sumodd = 0.0
    do 379 j = 3, itrap - 2, 2
    sumodd = sumodd + arg(j)
379 continue

    s(igo) = (delz/3.)* (arg(1) + arg(itrap) + & 4.*sumevn + 2.*sumodd)
    if (s(igo) .le. 0.0 )then
      write(6,* ) '------> s less than zero'
      write(6,* ) 'corresponding time: t= ',time
      write(6,* ) 'corresponding hm= ',hm
    end if
    write(6,* ) 'igo= ', igo, ' s= ',  s(igo)
    return
end

====================================================================
====================================================================

subroutine lamda(hm,delz,z,itrap,xd,jj)

common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
dimension z(41), argtop(41), argbot(41)

  this subroutine computes values of lambda using
  simpson's rule

do 10 i = 1, itrap
  argtop(i) = z(i)/visc(jj,i)
  argbot(i) = 1./visc(jj,i)
10 continue

evetop = 0.0
evebot = 0.0
    do 20 i = 2, itrap - 1, 2
      evetop = evetop + argtop(i)
      evebot = evebot + argbot(i)
20 continue

    oddtop = 0.0
oddbot = 0.0

do 30 i = 3, itrap - 2, 2
    oddtop = oddtop + argtop(i)
    oddbot = oddbot + argbot(i)
    continue
    top = (delz/3.)*(argtop(1) + argtop(itrap) + &
            4.*evetop + 2.*oddtop )
    bot = (delz/3.)*(argbot(1) + argbot(itrap) + &
            4.*evebot + 2.*oddbot )
    xlambda = top/bot
    return
    end

subroutine temp

common/abc/sint2(1000,60),qs(1000,60),radical(1000,60)
common/abc2/qz(1000,60)
common/one/visc(2000),imfull,hdot,h0(2000),hmax
common/two/dlo(100), diag(100), dup(100), ans(100), soln(100)
common/three/ istart, iz
common/four/trecc,tadt
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
common/six/narea,nprnt(10),href,iref
common/si/nca(1800),bb(1800,10),cc(1800,10)
common/sj/nv(1800)
common/sk/npm(1800,10),ncl(1800,10,10),ax(2),ay(2)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/const/mn,mnold,nnew(1800)
dimension t(1000,60), tg(1000,60), told(1000,60), grdot(100)
dimension x(1000,60), xg(1000,60), xold(1000,60), z(100)
dimension volcv(2000), hcalc(2000), gi(1000,60), g3(1000,60)
c
units of viscosity [ = ] grams/m*sec

if( isweep .eq. 0 ) then
    ileft = 1
i = start = 1
do 5 i = 1, mn
  do 5 k = 1, iz
    t(j,k) = t0
    told(j,k) = t(j,k)
    x(j,k) = 0.0
    xold(j,k) = x(j,k)
  5 continue

do 12 i = 1, n de
  do 12 j = 1, ncv(i)
    k = num(i,j)
    do 12 ig = 1, iz
      visc(k,igo) = v0
    12 continue

do 7637 i = 1, n de
  do 7637 ig = 1, iz
    sint2(i,igo) = 0.0
    radical(i,igo) = 0.0
    gz0(i,igo) = 6.033e-5*1.35
    gz(i,igo) = gz0(i,igo)
  7637 continue

zero = 0.0
ione = 1
write(8,2300) zero, iref, href

2300 format(' ',e16.9,3x,i4,3x,e16.9)
set up parameters for finite difference formulation

compute the average height of each control volume

do 25 ii = 1, mn
   jj = ncal(ii)
   volcv(jj) = 0.0
   do 15 k = 1, ncv(jj)
      index = num(jj,k)
      hcalc(index) = h0(index) - hdotlo(index)*tt
      volcv(jj) = volcv(jj) + area(index)*hcalc(index)/3.
   continue

ht = 0.0
do 16 k = 1, ncv(jj)
   index = num(jj,k)
   ht = ht + area(index)*hcalc(index)*hcalc(index)/(3.*volcv(jj))
continue

delz = href/(iz - 1)
delz = ht/(iz-1)
c1 = h1*delz/xk
cu = hu*delz/xk
g2 = delta*alfa/(delz**2)

z(1) = 0.0
do 73 ik = 1, iz - 1
   z(ik+1) = z(ik) + delz
continue

write(8,2301) z(i), t(iref,i)-273.15, x(iref,i), i
2301 format(3(lx,e16.9),2x,i4)

***********************************************************************
computation of heat transfer due to dwell effect
***********************************************************************

jtot = tdwell/delta
tstep = tdwell/10.
td = 0.0

the next loop steps through time, computing the temperature
and conversion profiles in the thickness direction for the
reference node located in the initial charge for the dwell period

do 783 j = 1, jtot
   td = delta*j
   do 779 my = 1, iz
      g3(iref,my) = delta*gl(iref,my)
   continue

first row of tridiagonal matrix
\begin{verbatim}

diag(1) = 1.0 + cl
dup(1) = -1.0
ans(1) = cl*tl

c middle rows of tridiagonal matrix

do 796 i = 2, iz-1
dlo(i) = -g2
diag(i) = 2.*g2 + 1.0
dup(i) = -g2
ans(i) = t(iref,i) + g3(iref,i)
796 continue

c last row of tridiagonal matrix

dlo(iz) = -1.0
diag(iz) = 1.0 + cu
ans(iz) = cu*tu

c call tridag to solve the tridiagonal matrix formed above

call tridag

do 780 my = 1, iz
t(iref,my) = soln(my)
780 continue

call cdhan(delta,t,qrdot,alfa,x,iref)
do 781 my = 1, iz
gl(iref,my) = qrdot(my)
781 continue

c call viscos to compute the viscosity

call viscos(t,iref)

if( td .ge. tstep )then
  write(8,2300) td, iref, href
  do 782 i = 1, iz
  write(8,2301) z(i), t(iref,i)-273.15, x(iref,i), i
782 continue
  tstep = tstep + tstep
end if

783 continue

c assign the initial temperature condition to all control
  c volumes which are full, using the profile of the reference
  c node as computed during the dwell period

c do 25 i = 1, mn
c  j = ncal(i)
do 25 k = 1, iz
t(jj,k) = t(iref,k)
\end{verbatim}
\[ x(jj,k) = x(iref,k) \]
\[ gl(jj,k) = gl(iref,k) \]
\[ sint2(jj,k) = sint2(iref,k) \]
\[ qz(jj,k) = qz(iref,k) \]
\[ radical(jj,k) = radical(iref,k) \]

continue

\[ tu = tw \]
\[ hu = huhot \]

return

endif

c---------------------------------------------------------------
c---------------------------------------------------------------
c the next loop does a temperature calculation for each
c control volume which is full
do 10 i = 1, mn

j = ncal(i)
if( imfull .eq. 1 ) tprint = tadt + 2.
c for control volumes which do not yet have calculated
c temperature profiles, compute the average profile of
c surrounding control volumes and use as initial condition

if( t(j,1) .lt. 5. )then
count = 0.0
do 52 k = 1, ncv(j)
  len = num(j,k)
  k1 = ncel(j,k,1)
  k2 = ncel(j,k,2)
  k3 = nep(len,k1)
  k4 = nep(len,k2)
if( t(k3,1) .gt. 5. )then
  count = count + 1.
do 71 igo = 1, iz
    t(j,igo) = t(j,igo) + t(k3,igo)
    x(j,igo) = x(j,igo) + x(k3,igo)
    gl(j,igo) = gl(j,igo) + gl(k3,igo)
    sint2(j,igo) = sint2(j,igo) + sint2(k3,igo)
    radical(j,igo) = radical(j,igo) + radical(k3,igo)
    qz(j,igo) = qz(j,igo) + qz(k3,igo)
  continue
end if
if( t(k4,1) .gt. 5. )then
  count = count + 1.
do 72 igo = 1, iz
    t(j,igo) = t(j,igo) + t(k4,igo)
    x(j,igo) = x(j,igo) + x(k4,igo)
    gl(j,igo) = gl(j,igo) + gl(k4,igo)
\[ \text{sint2}(j,igo) = \text{sint2}(k4,igo) + \text{sint2}(k4,igo) \]
\[ \text{radical}(j,igo) = \text{radical}(j,igo) + \text{radical}(k4,igo) \]
\[ \text{qz}(j,igo) = \text{qz}(j,igo) + \text{qz}(k4,igo) \]

52 continue
end if

55 continue
do 75 igo = 1, iz
\[ t(j,igo) = t(j,igo)/\text{count} \]
\[ x(j,igo) = x(j,igo)/\text{count} \]
\[ gl(j,igo) = gl(j,igo)/\text{count} \]
\[ \text{sint2}(j,igo) = \text{sint2}(j,igo)/\text{count} \]
\[ \text{radical}(j,igo) = \text{radical}(j,igo)/\text{count} \]
\[ \text{qz}(j,igo) = \text{qz}(j,igo)/\text{count} \]
end if

170 the next loop steps through time for the CV of interest
to compute the temperature and conversion profiles in
the thickness direction (Z-direction)
do 1000 j t = ileft, jcount
\[ \text{tt} = \text{delta} \times \text{jt} \]

if( imfull .eq. 0 )then
compute the average height of each control volume
\[ \text{volcv}(j) = 0.0 \]
do 20 k = 1, ncv(j)
index = num(j,k)
\[ \text{hcalc}(\text{index}) = \text{h0}(\text{index}) - \text{hdot} \times \text{tt} \]
\[ \text{volcv}(j) = \text{volcv}(j) + \text{area}(\text{index}) \times \text{hcalc}(\text{index})/3. \]
continue

20 \[ \text{ht} = 0.0 \]
do 30 k = 1, ncv(j)
index = num(j,k)
\[ \text{ht} = \text{ht} + \text{area}(\text{index}) \times \text{hcalc}(\text{index}) \times \text{hcalc}(\text{index})/(3. \times \text{volcv}(j)) \]
continue

30 \[ \text{z}(1) = 0.0 \]
do 45 k = 2, iz
\[ \text{z}(k) = \text{z}(k-1) + \text{ht} \times \text{dele} \]
continue

45 if( \text{ht} .le. 0. )then
write(6,*) 'ht less than zero >> ht= ',ht,' time= ',tt
stop
end if
\[ \text{cl} = \text{hl} \times \text{ht} \times \text{dele}/\text{xk} \]
\[ \text{cu} = \text{hu} \times \text{ht} \times \text{xk} \]
\[ \text{deltau} = \text{alfa} \times \text{delta} / (\text{ht} \times \text{ht}) \]
\[ g_2 = \text{deltau} / (\text{dele} \times \text{dele}) \]

\[
\text{do 39 my} = 1, \text{iz} \\
g_3(j,my) = \text{deltau} \times g_1(j,my) \\
\text{continue}
\]

else

\[
\text{if(itell.eq.0) then} \\
\actext{write(6,*)} 'the mold is full -- control given to the' \\
\actext{write(6,*)} 'temperature routine ...' \\
itell = 5 \\
\text{end if}
\]

\[
\text{if(tt .lt. time) then}
\text{if(itell2.eq.0) then} \\
\actext{write(6,*)} 'temp routine catching up ...' \\
\actext{write(6,*)} 'tt = ',tt,' tadt = ',tadt \\
itell2 = 5 \\
\text{end if}
\]

\[
c \quad \text{compute the average height of each control volume}
\]
\[
\text{volcv}(j) = 0.0 \\
\text{do 21 k} = 1, \text{ncv}(j) \\
\text{index} = \text{num}(j,k) \\
\text{hcalc(index)} = \text{ho(index)} - \text{hdot*tt} \\
\text{volcv}(j) = \text{volcv}(j) + \text{area(index)} \times \text{hcalc(index)} / 3. \\
\text{continue}
\]

\[
\text{ht} = 0.0 \\
\text{do 31 k} = 1, \text{ncv}(j) \\
\text{index} = \text{num}(j,k) \\
\text{ht} = \text{ht} + \text{area(index)} \times \text{hcalc(index)} \times \text{hcalc(index)} / (3. \times \text{volcv}(j)) \\
\text{continue}
\]

\[
\text{z(1)} = 0.0 \\
\text{do 46 k} = 2, \text{iz} \\
\text{z}(k) = \text{z}(k-1) + \text{ht} \times \text{dele} \\
\text{continue}
\]

\[
\text{cl} = \text{hl} \times \text{ht} \times \text{xk} \\
\text{cu} = \text{hu} \times \text{ht} \times \text{xk} \\
\text{deltau} = \text{alfa} \times \text{delta} / (\text{ht} \times \text{ht}) \\
\text{g2} = \text{deltau} / (\text{dele} \times \text{dele}) \\
\text{do 41 my} = 1, \text{iz} \\
g_3(j,my) = \text{deltau} \times g_1(j,my) \\
\text{continue}
\]

else

\[
c \quad \text{non-warped time parameters go here}
\]
if(itell3.eq.0)then
write(6,*), 'beginning use of non-warped time parameters...'
write(6,*), 'tt = ',tt, ' tadt = ',tadt
itell3 = 5
end if

compute the average height of each control volume

iwarp = 0
volcv(j) = 0.0
do 22 k = 1, ncv(j)
   index = num(j,k)
   hcalc(index) = h0(index) - hdot*time
   volcv(j) = volcv(j) + area(index)*hcalc(index)/3.
22 continue

ht = 0.0
do 33 k = 1, ncv(j)
   index = num(j,k)
   ht = ht + area(index)*hcalc(index)*hcalc(index)/(3.*volcv(j))
33 continue

delz = ht/(iz - 1)
z(1) = 0.0
do 48 k = 1, iz-1
   z(k+1) = z(k) + delz
48 continue

c1 = hl*delz/xk
cu = hu*delz/xk
g2 = delta*alfa/(delz**2)
do 773 my = 1, iz
   g3(j,my) = delta*gl(j,my)
773 continue
end if
end if

c first row of tridiagonal matrix

diag(1) = 1.0 + c1
dup(1) = -1.0
ans(1) = c1*tl

middle rows of tridiagonal matrix

do 7000 k = 2, iz-1
dlo(k) = -g2
diag(k) = 2.*g2 + 1.0

dup(k) = -g2
ans(k) = t(j,k) + g3(j,k)
7000 continue

c last row of tridiagonal matrix
dlo(iz) = -1.0
diag(iz) = 1.0 + cu
ans(iz) = cu*tu

c call tridag to solve the tridiagonal matrix formed above
call tridag
do 7001 k = 1, iz
told(j,k) = t(j,k)
xold(j,k) = x(j,k)
t(j,k) = soln(k)
7001 continue

c call cdhan to compute heat generation due to reaction, etc.
call cdhan(delta,t,qrdot,alfa,x,j)
do 774 k = 1, iz
if( iwp .eq. 1 ) then
gl(j,k) = qrdot(k)*(ht**2)/alfa
else
gl(j,k) = qrdot(k)
end if
774 continue

c call viscos to compute the viscosity of each element
call viscos(t,j)

C---
if( imfull .eq. 1 ) then

c this is the output section for after the mold is full
if( tt .ge. tprint ) then
do 301 jj = 1, narea
  kk = nprt(jj)
  if( kk .eq. j ) then
    write(8,2300) tt + tdwell, kk, ht
  do 307 k = 1, iz
    write(8,2301) z(k), t(kk,k)-273.15, x(kk,k), k
  307 continue
  tprint = tprint + 10.
go to 1000
end if
continue
end if
go to 1000
end if

if( tt .ge. time ) then
  if( i .eq. mn ) ileft = jt + 1
  go to 10
end if

1000 continue

if( imfull .eq. 1 ) and. (i .eq. mn) ) stop

10 continue

if( time .gt. trecd ) then
  itab2 = itab2 + 1
  write(6,1010) itab2, time + tdwell
  1010 format( ' ' , i4 , ' ... writing temperature profile ' ,
  & ' to data file ... time = ', f10.4 )
  do 7005 jj = 1, nparea
  kk = nprn(jj)
  write(8,2305) time+tdwell, kk, ht, tt+tdwell
  2305 format( ' ' , e16.9, 3x, i4, 3x, e16.9, 3x, e16.9 )
  do 7005 k = 1, iz
    if( tt .eq. time ) then
      tg(kk,k) = t(kk,k)
      xg(kk,k) = x(kk,k)
      go to 1002
    end if
    tg(kk,k)=((t(kk,k) - told(kk,k))/(delta*jt - delta*(jt-1)))*
    & (time - delta*(jt-1)) + told(kk,k)
    xg(kk,k)=((x(kk,k) - xold(kk,k))/(delta*jt - delta*(jt-1)))*
    & (time - delta*(jt-1)) + xold(kk,k)
  end if
  7005 continue
1002 write(8,2301) z(k), tg(kk,k)-273.15, xg(kk,k), k

7005 continue
    end if

if( imfull .eq. 0 ) return
    end

*****************************************************************************

c*****************************************************************************

common/two/ dlo(lO0), diag(lO0), dup(lO0), ans(lO0), soln(lO0)
common/three/ istart, iz

dimension beta(lO0), gamma(lO0)

beta(istart) = diag(istart)
gamma(istart) = ans(istart)/beta(istart)
isp1 = istart + 1

500 do 500 i = ispl, iz
    beta(i) = diag(i) - dlo(i)*dup(i-1)/beta(i-1)
    gamma(i) = (ans(i) - dlo(i)*gamma(i-1))/beta(i)
    continue

500 soln(iz) = gamma(iz)
last = iz - istart

600 do 600 k = 1, last
    i = iz - k
    soln(i) = gamma(i) - dup(i)*soln(i+1)/beta(i)
    continue

600 return
subroutine cdhan(delta,t,qrdot,alfa,ep,jj)

** Subroutine cdhan uses a kinetic model to compute the **
** heat generation due to chemical reaction, the reaction **
** rate, and the fractional conversion **

common/abc/sint2(1000,60),qz(1000,60),radical(1000,60)
common/abc2/qz0(1000,60)
common/three/istart,iz

dimension t(1000,60), ep(1000,60), aep(1000,60), qrdot(100)

if( isw .eq. 0 )then

** Experimentally determined kinetic parameters for SMC paste **

az = 0.2482484e13
ad = 0.2086285e12
ap0 = 0.1254301e13
ez = 0.5406955e05
ed = 0.1015223e05
ep0 = 0.6769401e05
am = 1.25543
delh = 116.35
rho = 1850000.
cp = 1.1171
xk = 0.3682
alfa = xk/(rho*cp)
epf = 1.0
fi20 = 1.73e-4
r = 8.314
isw = 5

end if

do 10 i = 1, iz

akz = az*exp( -ez/(r*t(jj,i)) )
akd = ad*exp( -ed/(r*t(jj,i)) )
akp0 = ap0*exp( -ep0/(r*t(jj,i)) )
c  initiator decomposition
sint2(jj,i) = sint2(jj,i) + akd*delta
ss = fi20*(1.0 - exp(-sint2(jj,i)))

-- inhibitory decomposition

delz = akz*qz(jj,i)*radical(jj,i)*delta
qz(jj,i) = qz(jj,i) - delz
ss = qz0(jj,i) - qz(jj,i)

-- radical generation

radical(jj,i) = ss - sss
if (ss .le. sss) radical(jj,i) = 0.0

-- propagation

if (ep(jj,i) .lt. epf) then
  q1 = akp0*(1.0 - ep(jj,i)/epf)**am
else
  ep(jj,i) = epf
  q1 = 0.0
end if

aep(jj,i) = q1*(1.0 - ep(jj,i))*radical(jj,i)
if (ss .le. sss) aep(jj,i) = 0.0

ep(jj,i) = ep(jj,i) + aep(jj,i)*delta
qrdot(i) = aep(jj,i)*delh/cp

10 continue

return
end

C=====================================================================
C=====================================================================

subroutine viscos(t,jj)

common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/three/istart,iz
common/sj/ncv(1800)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)

dimension t(1000,60)
do 10 i = 1, ncv(jj)
    k = num(jj,i)
do 10 j = 1, iz
    visc(k,j) = 1000000.
continue
return
end

subroutine outputl(nstep,tab)

common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/se/kf(1800),fht(1800),f(1800)
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
c-------- output the pressure and velocity field  ----------
c
write(6,801) itab
801 format(' /i4,' ... writing pressure and velocity ')
write(6,802) time
802 format(' / distributions to data file ... time = /

n0=0
n1=1
n2=2
n3=3
n4=4
n5=5
n11=11
n15=15
a0=0.0

write(14,803) time,n3,n4,n1,n15,n2,
803 format(4x,'-1',/4x,'55',/,&
      1x,'pressure distribution',
      1x,'at time=',f10.3,' SEC',/,,
      6i10,/,4i10,/,e13.5)
do 11 i=1,nde
    write(14,804) i,p(i)
format(i10,/,e13.5)
continue
write(14,805)
format(4x,'-1')

write(14,806) time,n3,n4,n2,n11,n2, &
& time,n3,n2,n1,n1,nstep,time
format(4x,'-1',/,'56',/,
& 'velocity distribution',
& ' at time = ',f10.3,' SEC',/,'/',
& 6i10,/,4i10,/,e13.5)
do 12 i=1,nel
che=abs(u(i))+abs(v(i))+abs(w(i))
if(che.ne.0.0) then
write(14,807) i,n3,u(i),v(i),w(i)
endif
continue
format(2i10,/,3e13.5)
write(14,805)

the mold is full -- write the flow front positions
to the data file and exit to temperature routine...
write(14,808) n3,n1,n1,n0,n2,nl,n1,n1,n1,a0
format(4x,'-1',/,'55',/,
& 'simulation flow front',/,
& 'flow front',/,'/',
& 6i10,/,3i10,/,e13.5)
do 13 i=1,nde
write(14,804) i,fht(i)
continue
write(14,805)
return
end

subroutine output4(nstep,tab)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/se/kf(1800),fht(1800),f(1800)
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/dim/jz

**** output the temperature field ********

! n0=0
! n1=1
! n2=2
! n3=3
! n4=4
! n5=5
! n11=11
! n15=15
! a0=0.0
! mid = jz/2 + 1

write(14,803) time,n3,n4,n1,n15,n2,
! n1,n2,n1,n1,nstep,time
803 format(4x,'-1',/,'55',/,
& 1x,'temperature distribution',
& 1x,'at time= ','f10.3,' SEC',////,  
& 6i10,/,4i10,/,e13.5)

do 11 i=1,nde
write(14,804) i,t(i,mid) - 273.15
804 format(i10,/,e13.5)
11 continue

write(14,805)
805 format(4x,'-1')

return
end

=======================================

subroutine output2

common/ab/xnode(2000,3),ynode(2000,3),nde,nel
common/se/kf(1800),fht(1800),f(1800)
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
common/four/tred,tad
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/balance/vf1,vf2
c-------- output the pressure and velocity field ------------------------
c
n0=0
n1=1
n2=2
n3=3
n4=4
n5=5
n11=11
n15=15
a0=0.0

write(13,130) n3,n1,n1,n0,n2,n1,n1,n1,n1,a0
130 format(4x, '-1',/,'55',/,'simulation flow front',/,'flow front',///,'6i10',/,'e13.5')

do 197 i=1,nde
   write(13,210) i,fht(i)
210 format(i10,/,e13.5)
197 continue

write(13,310)
310 format(4x, '-1')

write(6,842)
write(9,842)
842 format(' ',/)
write(9,850)
850 format( '9x,'---- Post-simulation calculated variables:' ,/)

write(6,851) time
write(9,851) time
851 format( '9x,'Simulated mold filling time (sec) ',& 'f13.9',/)

write(6,852) tadt
852 format( '9x,'Mass balance mold filling time (sec) ',& 'f16.9',/)

do 870 i = 1, nh0
   write(6,853) i, height(i) - hdot*time
   write(9,853) i, height(i) - hdot*time
853 format( '9x,'Final thickness, region ',il,' (m) ',7x,& 'f13.9',/)
870 continue

write(6,855) vfl
write(9,855) vfl
182
855 format(' ',9x,'vl
& f13.9,/
write(6,856) vf2
write(9,856) vf2
856 format(' ',9x,'vf2
& f13.9,/
return
end

subroutine output3(afill)

common/boud2/nih(10,1000),nn,hh(1000)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/four/trecd,tadt
common/ini/vol,achrg,dti,delt

write(9,842)
write(9,842)
write(9,841)
841 format(' ',9x,'******************************')
write(9,800)
800 format(' ',9x,'** Run Diagnostics Summary **)'
write(9,841)
write(9,842)
842 format(' ',//)
write(9,3450) nde
3450 format(' ',9x,'Nodes: ',i6)
write(9,3451) nel
3451 format(' ',9x,'Elements: ',i6,//)
write(9,843)
843 format(' ',9x,'----- Pre-simulation calculated variables:/',)
do 977 i = 1, nn
write(9,801) i, hh(i)
801 format(' ',5x,'Height of initial charge, region ',il,' (m)
',8x,
& f13.9,/
977 continue
write(9,802) hdot,cosx,cosy,cosz
802 format(' ',9x,'Mold closing speed (m/sec)
& f13.9,//', 'In the direction of: ',3f13.9)
write(9,803) afill
subroutine mass(mnp, npc, kr, mdr, cor)

    c npc(mnp) is the list of node numbers of "new" filled nodes (CV)
    c mnp is the total number of new filled nodes. it is changed after
    c every flow advancement, and it is recounted each time.
    c mdr(kr) is the list of node numbers of the "new" moving front.
    c kr is the new total number. it is changed after every
    c flow advancement and it is recounted each time.

    common/sb/xnode(2000,3),ynode(2000,3),nde,nel
    common/se/kf(1800),fht(1800),f(1800)
    common/const/mn,mnold,nnew(1800)
    common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
    common/si/ncal(1800),bb(1800,10),cc(1800,10)
    common/mj/ncv(1800)
    common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
    common/flow/frate(2000),mnf,mdf(1000)
    common/ini/vol,achrg,dti,delt
    common/monitor/dt,adt
    common/four/trec,tadt
    common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
common/balance/vf1,vf2

dimension hmust(2000)
dimension mdr(1000),npc(100)

c---- mass balance verification

   s1 = 0.0
   s4 = 0.0
   do 6100 i = 1, mn
         ip = ncal(i)
      do 6100 j = 1, ncv(ip)
          k = num(ip,j)
          s1 = s1 + area(k)*h0(k)/3.
          s4 = s4 + hdotlo(k)*area(k)/3.
6100 continue

   s2 = 0.0
   s3 = 0.0
   do 6101 ip = 1, mnf
         i = mdf(ip)
      do 6101 j = 1, ncv(i)
          k = num(i,j)
          s2 = s2 + f(i)*area(k)*h0(k)/3.
          s3 = s3 + f(i)*hdotlo(k)*area(k)/3.
6101 continue

   time2 = ( s1 + s2 - vol )/( s4 + s3 )

   do 7319 j = 1, nel
      hmust(j) = h0(j) - hdotlo(j)*time2
    7319 continue

   vcalc1 = 0.0
   vcalc2 = 0.0
   do 7220 i = 1, mn
         ip = ncal(i)
      do 7220 j = 1, ncv(ip)
          k = num(ip,j)
          vcalc1 = vcalc1 + area(k)*hmust(k)/3.
          vcalc2 = vcalc2 + area(k)*h(k)/3.
7220 continue

   do 1795 i = 1, mnf
      mdl = mdf(i)
      volcv1(mdl) = 0.0
      volcv2(mdl) = 0.0
   do 1795 j = 1, ncv(mdl)
      k = num(mdl,j)
      volcv1(mdl) = volcv1(mdl) + area(k)*hmust(k)/3.
      volcv2(mdl) = volcv2(mdl) + area(k)*h(k)/3.
1795 continue
1795 continue

    do 7221 i = 1, mnf
        ip = mdf(i)
        vcalc1 = vcalc1 + f(ip)*volcv1(ip)
        vcalc2 = vcalc2 + f(ip)*volcv2(ip)
    7221 continue

    vf1 = vcalc1/vol
    vf2 = vcalc2/vol

do 57 i = mn+1, mn+mnp
    ncal(i) = npc(i-mn)
57 continue

mn is the total number of filled (f=1.0) nodes.
mnp is the number of "new" filled nodes after this advancement.
so it should be added to the old number.
kr is the recounted number of the nodes on the moving front.
the list of moving front nodes should also be assigned to
mdf(i) from mdr(i).

mn = mn+mnp
mnf = kr

do 77 i = 1, mnf
    mdf(i) = mdr(i)
77 continue

write(6,9679)
9679 format( ' ',/)
write(6,*), 'dt= ', dt, ' time= ', time, ' trecd= ', trecd
write(6,78) nde, mn
78 format(1x,(total nodes = ', i5,
         ' nodes filled = ', i5, '))
write(6,*), 'vf1= ',vf1,' vf2= ',vf2,' cor= ',cor
write(6,9679)

return
end

===============================================

subroutine check1
call check2

do 19 i = 1, nde
    test=0.0
    spc(i) = 0.
    do 101 jj=1,ncv(i)
        coef(i,jj)=0.0
    101 continue

do 14 j = 1, ncv(i)
    k = num(i,j)
    l1 = ncel(i,j,1)
    l2 = ncel(i,j,2)
    ml=nep(k,l1)
    m2=nep(k,l2)
    fx(i,j) = s(k)*( xl1(i,j)+xl2(i,j) )/(2.*area(k))
    fy(i,j) = s(k)*( yl1(i,j)+yl2(i,j) )/(2.*area(k))
    pcl(i,j) = fx(i,j)*b(k,l1) + fy(i,j)*c(k,l1)
    pc2(i,j) = fx(i,j)*b(k,l2) + fy(i,j)*c(k,l2)
    spc(i)=spc(i)+fx(i,j)*bb(i,j)+fy(i,j)*cc(i,j)
    do 102 jj=1,ncv(i)
        if( ml.eq.ndcoe(i,jj) )then
            coef(i,jj)=coef(i,jj)+pcl(i,j)
        endif
        if( m2.eq.ndcoe(i,jj) )then
            coef(i,jj)=coef(i,jj)+pc2(i,j)
        endif
    102 continue
14 continue

do 103 jj=1,ncv(i)
test = test + abs(coef(i, jj))
continue

diff = abs(spc(i)) - abs(test)
if(abs(diff).lt.1.0e-22) diff = 0.0
if(diff.lt.0.0) then
  write(25,898)i, spc(i), test, diff
else
  write(25,899)i, spc(i), test, diff
endif

continue

format(i5,' spc, SON, SON*factor = ',3e16.8,* *****)
format(i5,' spc, SON, SON*factor = ',3e16.8)

return
end

subroutine check2

common/sb/xnode(2000,3), ynode(2000,3), nde, nel
common/sc/nep(2000,3)
common/sj/ncv(1800)
common/sk/num(1800,10), ncel(1800,10,10), ax(2), ay(2)
common/che/ndcoe(1800,10), coef(1800,10), factor

dimension fx(1800,10), fy(1800,10), pcl(1800,10), pc2(1800,10)
dimension sumc(1800), spc(1800), pold(1800), hmust(2000)

do 1 i = 1, nde
  jj = 0
  do 2 ii = 1, nde
    ifact = 0
    do 2 jj = 1, ncv(i)
      ip = num(i, jj)
      ml = ncel(i, jj, 1)
      m2 = ncel(i, jj, 2)
      if( ii.eq.nep(ip, ml) ) ifact = 1
      if( ii.eq.nep(ip, m2) ) ifact = 1

      if( jj.ne. 0 ) then
        do 3 k = 1, jj
          if( ii.eq.ndcoe(i, jj) ) ifact = 0
        3 continue
      endif
  2 continue

  c
if (ifact.eq.1) then
  jj=jj+1
  ndcoe(i,jj)=ii
  write(24,801)i,jj,ndcoe(i,jj)
endif

2  continue
1  continue

801  format(3i5)

return
end

c====
c====

subroutine enrcnvrt(courant)

common/radl/rad(1800,45),prr(1800,45),prh(1800,45)
common/rad2/rii(1800,45),pri(1800,45),rih(1800,45)
common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/itort/accu,taccu,errvis,thetae,thetaac
common/dim/jz
common/one/visc(2000,60),imfull,hdot,h0(2000),hmax
common/five/height(10),nh0,ielem(2000),ihn(10),ih(10,1000),time
common/const/mn,mnold,nnew(1800)
common/monitr/dt,adt
common/flow/frate(2000),mnf,mdf(1000)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sh/xs1(1800,10),ys1(1800,10),cx1(1800,10),cy1(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/se/kf(1800),fht(1800),f(1800)
common/sk/num(1800,10),nce(1800,10,10),ax(2),ay(2)
common/sj/ncv(1800)
common/rnh2/ncn(1800),ndcv(1800,20),nnchn(1800,20,2)
common/epgpt/timin,timax,cp0(2000),tf(2000),ekz(1800,45)
common/sc/nep(2000,3)
common/epgpt/rh,chk1,chk2,cm1,cm2,g(1800,45)
  l  epeb(2000,3),epse(2000),cndt
common/coemtl/tmntl,cm1l
common/far/area(1800)
common/rnh/cr,rho,cp,alfa,tlow,tup,havg(2000),cct
common/jan8/hs ESC SUB
ns = int((courant+0.7)/0.7)
dts = dt/float(ns)
write(6,*), 'ns=
write(6,*), 'dts=
d write(6,*), 'dt=
if( dts .gt. dt )then
write(6,*), 'dts > dt !!! '
stop
endif

if( dts .gt. 0.1 )then
ns = 10*ns
go to 10
endif

jz = 41
dele = 1./(float(jz) - 1.)
dele2 = dele**2

do 3010 ins = 1, ns

if( ins .eq. 1 )then
tsub = time
do 14 isub = 1, nel
hsub(isub) = h(isub)
continue
else
tsub = tsub + dts
do 15 isub = 1, nel
hsub(isub) = h0(isub) - hdotlo(isub)*tsub
continue
endif

do 20 isub = 1, nde
havg(isub) = 0.0
do 21 jsub = 1, ncv(isub)
ksub = num(isub,jsub)
hsavg(isub) = havg(isub) + hsub(ksub)
continue
havg(isub) = havg(isub)/float(ncv(isub))
continue

do 1110 imm=1,mn
i=ncal(imm)
do 1110 k=1,jz
p( i,k ) = t(i,k)
ps( i,k ) = cv(i,k)
tri( i,k ) = rii(i,k)
phr(i,k) = rih(i,k)
prr(i,k) = rad(i,k)
1110 continue

heat conduction in thickness direction using
c the finite difference form

    do 45 imn = 1, mn
        i = ncal(imn)
        do 45 k = 1, jz
            ekz(i,k) = 0.0
        45 continue

    do 40 imn = 1, mn
        i = ncal(imn)
        dtau = dts*alfa/havg(i)**2
        do 40 k = 2, jz-1
            ekz(i,k) = ( prt(i,k+1) - 2.*prt(i,k) + prt(i,k-1) )*1
            dtau*areat(i)*ck/( dele2*havg(i)**2 )
        40 continue

    call cnvrad(dts)
    call energy(dts,ns)

3010 continue

return
end

subroutine energy(dts,ns)

common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/itrcrt/accu,taccu,errvis,thetas,thetas
common/dim/jz
common/ecootr/xm6(2000,6),ym6(2000,6),xrc(2000,10),yrc(2000,10)
common/empty/timin,timax,cp0(2000),tf(2000),ekz(1800,45)
common/cppty/rh,CHK1,CHK2,CHML,CHM2,g(1800,45)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/se/kf(1800),fht(1800),f(1800)
common/sh/xs1(1800,10),ys1(1800,10),cx1(1800,10),cy1(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
common/sj/ncv(1800)
common/srh2/ncn(1800),ndc(1800,20),nnnc(1800,20,2)
common/boud/lbc(1000),n
common/boud2/md(5,15),bc(5),io(100),nout
common/const/mn,mnold,nnew(1800)
common/monitor/dt,adt
common/flow/frate(2000),mnf,mdf(1000)
constant wall temperature boundary conditions...

do 1 imn = 1, mn
   i = ncal(imn)
   t(i,1) = twlow
   t(i,jz) = twup
 1 continue

do 6 k = 2, jz - 1

do 11 imn = 1, mn
   i = ncal(imn)
   iai = ncn(i) + 1
   fe(i) = 0.0
   dtau = dts*alfa/havg(i)**2
   te(i) = 0.0

   do 12 j = 1, iai
       a(i,j) = 0.
       a2(i,j) = 0.
   12 continue

   do 42 ii=1,ncv(i)
       iee=num(i,ii)
       iel=ncel(i,ii,1)
       ie2=ncel(i,ii,2)
       ie3=ncel(i,ii,3)
       ie4=ncel(i,ii,4)
       ie5=ncel(i,ii,5)
       ie6=ncel(i,ii,6)
       ie7=ncel(i,ii,7)
       iil=nep(iee,iel)
       ii2=nep(iee,ie2)
       i1=nncn(i,ii,1)
       i2=nncn(i,ii,2)

   internal energy

   fe(i) = fe(i) + (area(iee)/3.)*prt(i,k)*ck/havg(i)**2
   a(i,iai) = a(i,iai) + (area(iee)/3.)*ck/havg(i)**2
   te(i) = te(i) + (area(iee)/3.)*ck/havg(i)**2

   source term due to chemical reaction

   fe(i) = fe(i) + dtau*(area(iee)/3.)*rh*g(i,k)*rho
c heat convection & conduction

\[ a(i,iai) = a(i,iai) + dtau*ccp*uav(iee)* \\
\] 
\[ (ynew(iee,ie3)*(epea(iee,ie7)-eeb(iee,ie7)) \\
\] 
\[ *(4.*ynew(iee,ie5)+ynew(iee,ie3))/6.-ee(iee,ie7)) \\
\] 
\[ xnew(iee,ie3)*epeb(iee,ie7)+ \\
\] 
\[ ynew(iee,ie4)*(-epea(iee,ie7)+eeb(iee,ie7)) \\
\] 
\[ *(4.*ynew(iee,ie6)+ynew(iee,ie4))/6.+ee(iee,ie7)) \\
\] 
\[ +xnew(iee,ie4)*epeb(iee,ie7)) \\
\]

\[ a(i,i1) = a(i,i1) + dtau*ccp*uav(iee)* \\
\] 
\[ (ynew(iee,ie3)*(epea(iee,iel)-eeb(iee,iel)) \\
\] 
\[ *(4.*ynew(iee,ie5)+ynew(iee,ie3))/6.-ee(iee,iel)) \\
\] 
\[ xnew(iee,ie3)*epeb(iee,iel)+ \\
\] 
\[ ynew(iee,ie4)*(-epea(iee,iel)+eeb(iee,iel)) \\
\] 
\[ *(4.*ynew(iee,ie6)+ynew(iee,ie4))/6.+ee(iee,iel)) \\
\] 
\[ +xnew(iee,ie4)*epeb(iee,iel)) \\
\]

\[ a(i,i2) = a(i,i2) + dtau*ccp*uav(iee)* \\
\] 
\[ (ynew(iee,ie3)*(epea(iee,iel2)-eeb(iee,iel2)) \\
\] 
\[ *(4.*ynew(iee,ie5)+ynew(iee,ie3))/6.-ee(iee,iel2)) \\
\] 
\[ xnew(iee,ie3)*epeb(iee,iel2)+ \\
\] 
\[ ynew(iee,ie4)*(-epea(iee,iel2)+eeb(iee,iel2)) \\
\] 
\[ *(4.*ynew(iee,ie6)+ynew(iee,ie4))/6.+ee(iee,iel2)) \\
\] 
\[ +xnew(iee,ie4)*epeb(iee,iel2)) \\
\]

42 continue

c heat conduction in gapwise direction

\[ fe(i) = fe(i) + ekz(i,k) \]

11 continue

do 200 imm = 1, mn
i = ncal(imm)
iai = ncn(i) + 1
ccc = te(i) - a(i,iai)
sum = 0.0
do 201 j = 1, ncn(i)
sum = sum + a(i,j)
201 continue

c write(6,*), 'node=',i
\[ c write(6,*), 'cc=',ccc, 'sum=',sum, 'te=',te(i) \]
\[ c write(6,*), 'fe=',fe(i),' a(iai)=',a(i,iai) \]
c read(5,*), pause
200 continue

c solve \{a\}*{T} = {fe} by iteration

100 err=0.

do 15 imm = 1, mn
i = ncal(imm)
iai = ncn(i)+1
if (f(i) .eq. 0.) go to 15
if( a(i,iai) .eq. 0. ) go to 15
ffe(i) = fe(i)
do 20 j = 1, ncn(i)
    ffe(i) = ffe(i) - t(ndcv(i,j),k)*a(i,j)
20 continue
tod = t(i,k)
t(i,k) = thetæ*ffe(i)/a(i,iai) + (1.-thetæ)*t(i,k)
err = err + abs(t(i,k) - tod)/t(i,k)
15 continue

c write (6,*) 't(228,2),err',t(228,2),err
if (err .gt. 10.*accu) go to 100
6 continue
return
end

subroutine elim
common/sc/nep(2000,3)
common/sb/xnode(2000,3),ynode(2000,3),nde,ncl
common/far/areat(1800)
common/eppty/timin,timax,cp0(2000),tf(2000),ekz(1800,45)
common/cppty/rh,chk1,chk2,chk1,chk2,g(1800,45)
common/covis/aet,eet,cvg,aet,bet,cet,ret,eal,ea2,ea3,cmva,visma
common/coemtl/tmtl,cmtl
common/curing/dtc,mature
common/se/kf(1800),fht(1800),f(1800)
common/sknum(1800,10),nccl(1800,10,10),ax(2),ay(2)
common/sj/ncv(1800)
common/rh2/ncn(1800),ndcv(1800,20),nncl(1800,20,2)
common/rh2/ck,rho,cp,alfa,twlw,tfup,havg(2000),ccp

ami=1.ell
ama=0.
cpma=0.
pi=2.*acos(0.)
do 10 i=1,nde
    if (ami .gt. areat(i)) ami=areat(i)
10 continue
if (ama .lt. areat(i)) ama = areat(i)
10 continue

amid = (ami + ama) / 2.
rmid = (amid / pi) ** 0.5

for curing

rmi = (ami / pi)
dtc = 0.1 * rmi / alfa
write (6, *) 'dt for curing dtc=', dtc

return
end

-----------------------------------------------

subroutine shapvel(courant)

common/sb/xnode(2000,3), ynode(2000,3), nde, nel
common/sc/nep(2000,3)
common/sk/num(1800,10), ncel(1800,10,10), ax(2), ay(2)
common/sj/nov(1800)
common/rrh2/ncn(1800), ndcv(1800,20), nnnc(1800,20,2)
common/const/mn, mnold, nnew(1800)
common/monitr/dt, adt
common/dim/jz
common/ecootr/xm6(2000,6), ym6(2000,6), xrc(2000,10), yrc(2000,10)
common/empty/timin, timax, cp0(2000), tf(2000), ekz(1800,45)
   , epeb(2000,3), epse(2000), cndt
   , epcb(2000,3), epsc(2000), coart
common/edm/edgmi(2000)
common/rrh/cp, rhol, ccp, twlow, twup, havg(2000), ccp

c coart: if small then approach upwind scheme

c used for mass convection

c cndl: 1 for actually conduction; small for neglect the heat

c conduction

courant = 0.
coart = 1.e-5
cndt = 1.

do 10 ikj = 1, kj
\[ i = \text{mv}(ik) \]

\[ \text{uav}(i) = (\text{ut}(i)^2 + \text{vt}(i)^2)^{0.5} \]

if (\text{uav}(i) \gt 0.0) then
    \[ \text{eps}(i) = \text{cndt*ck}/(\text{ccp*uav}(i)) \]
    \[ \text{epsc}(i) = \text{coart}/(\text{uav}(i)) \]
    \[ \text{ut}1 = \text{ut}(i)/(\text{uav}(i)) \]
    \[ \text{ut}2 = \text{vt}(i)/(\text{uav}(i)) \]

\[
\begin{align*}
\text{write (6,*) 'i,eps}(i),epsc}(i)',i,\text{eps}(i),\text{epsc}(i) \\
\text{do 15 j=1,10} \\
\text{xnew}(i,j) = \text{xrc}(i,j)\text{ut}1 + \text{yrc}(i,j)\text{ut}2 \\
\text{ynew}(i,j) = -\text{xrc}(i,j)\text{ut}2 + \text{yrc}(i,j)\text{ut}1 \\
\text{continue} \\
\text{xma} = 0. \\
\text{do 20 j=1,3} \\
\text{if (xnew}(i,j) \gt \text{xma}) then \\
\text{xma} = \text{xnew}(i,j) \\
\text{jj} = j \\
\text{end if} \\
\text{continue} \\
\text{if (eps}(i) \lt 1.e-10) then \\
\text{if (jj .eq. 1) then} \\
\text{e1=1.} \\
\text{e2=0.} \\
\text{e3=0.} \\
\text{end if} \\
\text{if (jj .eq. 2) then} \\
\text{e1=0.} \\
\text{e2=1.} \\
\text{e3=0.} \\
\text{end if} \\
\text{if (jj .eq. 3) then} \\
\text{e1=0.} \\
\text{e2=0.} \\
\text{e3=1.} \\
\text{end if} \\
\text{go to 50} \\
\text{end if} \\
\text{e1 = exp( (xnew}(i,1)-\text{xnew}(i,jj))/\text{eps}(i) )} \\
\text{e2 = exp( (xnew}(i,2)-\text{xnew}(i,jj))/\text{eps}(i) )} \\
\text{e3 = exp( (xnew}(i,3)-\text{xnew}(i,jj))/\text{eps}(i) )} \\
\text{if (epsc}(i) \lt 1.e-10) then \\
\text{if (jj .eq. 1) then} \\
\text{c1=1.} \\
\text{c2=0.} \]
\begin{verbatim}
196
   c3=0.
   end if
   if ( jj .eq. 2) then
      c1=0.
      c2=1.
      c3=0.
   end if
   if ( jj .eq. 3) then
      c1=0.
      c2=0.
      c3=1.
   end if
   go to 55
end if

   c1 = exp( (xnew(i,1)-xnew(i,jj))/epsc(i) )
   c2 = exp( (xnew(i,2)-xnew(i,jj))/epsc(i) )
   c3 = exp( (xnew(i,3)-xnew(i,jj))/epsc(i) )

   55 denoe = (ynew(i,2) - ynew(i,3))*el - (ynew(i,1) - ynew(i,3))
              *e2 + (ynew(i,1) - ynew(i,2))*e3
   epea(i,1) = (ynew(i,2) - ynew(i,3))/denoe
   epea(i,2) = (ynew(i,3) - ynew(i,1))/denoe
   epea(i,3) = (ynew(i,1) - ynew(i,2))/denoe
   eeb(i,1) = (e3 - e2)/denoe
   eeb(i,2) = (e1 - e3)/denoe
   eeb(i,3) = (e2 - e1)/denoe
   epeb(i,1) = eeb(i,1)*epsc(i)
   epeb(i,2) = eeb(i,2)*epsc(i)
   epeb(i,3) = eeb(i,3)*epsc(i)
   eec(i,1) = (ynew(i,3)*e2-ynew(i,2)*e3-ynew(i,3)+ynew(i,2))
              /denoe
   eec(i,2) = (ynew(i,1)*e3-ynew(i,3)*el-ynew(i,1)+ynew(i,3))
              /denoe
   eec(i,3) = (ynew(i,2)*el-ynew(i,1)*e2-ynew(i,2)+ynew(i,1))
              /denoe
   denoc = (ynew(i,2)-ynew(i,3))*c1-(ynew(i,1)-ynew(i,3))
           *c2+(ynew(i,1)-ynew(i,2))*c3
   write (6,* ) 'denoe,denoc',denoe,denoc

   epca(i,1) = (ynew(i,2)-ynew(i,3))/denoc
   epca(i,2) = (ynew(i,3)-ynew(i,1))/denoc
   epca(i,3) = (ynew(i,1)-ynew(i,2))/denoc
   ecb(i,1) = (c3 - c2)/denoc
   ecb(i,2) = (c1 - c3)/denoc
   ecb(i,3) = (c2 - c1)/denoc
   epcb(i,1) = ecb(i,1)*epsc(i)
   epcb(i,2) = ecb(i,2)*epsc(i)
   epcb(i,3) = ecb(i,3)*epsc(i)
\end{verbatim}
\[
\begin{align*}
\text{ecc}(i,1) &= (\text{ynew}(i,3)*c2-\text{ynew}(i,2)*c3-\text{ynew}(i,3)+\text{ynew}(i,2)) \\
&= \text{denoc} \\
\text{ecc}(i,2) &= (\text{ynew}(i,1)*c3-\text{ynew}(i,3)*c1-\text{ynew}(i,1)+\text{ynew}(i,3)) \\
&= \text{denoc} \\
\text{ecc}(i,3) &= (\text{ynew}(i,2)*c1-\text{ynew}(i,1)*c2-\text{ynew}(i,2)+\text{ynew}(i,1)) \\
&= \text{denoc}
\end{align*}
\]

end if

*compute the courant number*

\[
\begin{align*}
courl &= \text{abs}(\text{ut}(i)*\text{dt}/\text{edgm}(i)) \\
cour2 &= \text{abs}(\text{vt}(i)*\text{dt}/\text{edgm}(i))
\end{align*}
\]

if (courant \text{lt} courl) then
  courant = courl
end if

if (courant \text{lt} cour2) then
  courant = cour2
end if

continue

return
end

\[
\begin{align*}
c==&\text{===================================================================================================}
\end{align*}
\]

\[
\begin{align*}
c==&\text{===================================================================================================}
\end{align*}
\]

\[
\begin{align*}
\text{subroutine cnvrad(dts)}
\end{align*}
\]

\[
\begin{align*}
\text{common/tcv/t}(1800,45),\text{prc}(1800,45),\text{prt}(1800,45),\text{cv}(1800,45) \\
\text{common/rad1/rad}(1800,45),\text{prr}(1800,45),\text{prh}(1800,45) \\
\text{common/rad2/rii}(1800,45),\text{pri}(1800,45),\text{rih}(1800,45) \\
\text{common/itcrit/accu},\text{taccu},\text{errvis},\text{thetae},\text{thetae} \\
\text{common/dim/j2} \\
\text{common/eco6tr/xm6}(2000,6),\text{ym6}(2000,6),\text{xrc}(2000,10),\text{yrc}(2000,10) \\
\end{align*}
\]

\[
\begin{align*}
\text{common/spppt/rah},\text{chk1},\text{chk2},\text{chp1},\text{chm2},\text{g}(1800,45) \\
\text{common/sb/xnode}(2000,3),\text{ynode}(2000,3),\text{nde},\text{nel} \\
\text{common/sc/nep}(2000,3) \\
\text{common/sd/b}(2000,3),\text{c}(2000,3),\text{p}(1800) \\
\text{common/se/kf}(1800),\text{fht}(1800),\text{f}(1800) \\
\text{common/sf/cx}(2000),\text{cy}(2000),\text{area}(2000) \\
\text{common/shl/xs1}(1800,10),\text{ys1}(1800,10),\text{cx1}(1800,10),\text{cy1}(1800,10) \\
\text{common/shb/xs2}(1800,10),\text{ys2}(1800,10),\text{cx2}(1800,10),\text{cy2}(1800,10) \\
\text{common/sk/num}(1800,10),\text{ncl}(1800,10,10),\text{ax}(2),\text{ay}(2)
\end{align*}
\]
Kinetic Parameters used by Ron Hill for DSC results...

eff = 2.*0.2
efq = 0.1
az = 827363.98
ez = 8127.18
ad = 1.4212e+18
ed = 168615.4
ap0 = 3.1809e+10
ep0 = 52592.7
am = 2.1005
af = 1.0
ret = 8.314

c since iterative scheme is used, the coefficient matrix and vectors
c need to be zeroed out before calculation

do 6 k=1,jz

do 11 imm=1,mn
i=ncal(imm)

akd = ad*exp( -ed/( ret*prr(i,k) ) )
akz = az*exp( -ez/( ret*prr(i,k) ) )
akp = ap0*exp(-ep0/(ret*prr(i,k) ) )
akp = akp*( 1. - prc(i,k)/af )**am

iai=ncn(i)+1
fe(i)=0.0
feh(i)=0.0
fer(i)=0.0

do 12 j=1,iai
a(i,j)=0.
12 continue

do 41 ii=1,ncv(i)
\[ i_{ee} = \text{num}(i, ii) \]
\[ i_{el} = \text{ncel}(i, ii, 1) \]
\[ i_{e2} = \text{ncel}(i, ii, 2) \]
\[ i_{e3} = \text{ncel}(i, ii, 3) \]
\[ i_{e4} = \text{ncel}(i, ii, 4) \]
\[ i_{e5} = \text{ncel}(i, ii, 5) \]
\[ i_{e6} = \text{ncel}(i, ii, 6) \]
\[ i_{e7} = \text{ncel}(i, ii, 7) \]
\[ i_{il} = \text{nep}(i_{ee}, i_{el}) \]
\[ i_{i2} = \text{nep}(i_{ee}, i_{e2}) \]
\[ i_{l} = \text{nncn}(i, ii, 1) \]
\[ i_{2} = \text{nncn}(i, ii, 2) \]

**c------- mass convection using special shape function**

\[
A(i, i_{ai}) = A(i, i_{ai}) + \text{dts} \times u_{av}(i_{ee}) \times (y_{new}(i_{ee}, i_{e3}) \times (\text{epca}(i_{ee}, i_{e7}) - \text{ecb}(i_{ee}, i_{e7}))
\]
\[
\text{epca}(i_{ee}, i_{e7}) - \text{ecb}(i_{ee}, i_{e7}) -
\]
\[
* (4 \times y_{new}(i_{ee}, i_{e5}) + y_{new}(i_{ee}, i_{e3})) / 6 - \text{ecc}(i_{ee}, i_{e7}) -
\]
\[
x_{new}(i_{ee}, i_{e3}) \times \text{epcb}(i_{ee}, i_{e7}) +
\]
\[
y_{new}(i_{ee}, i_{e4}) \times (-\text{epca}(i_{ee}, i_{e7}) + \text{ecb}(i_{ee}, i_{e7})) \times (4.
\]
\[
\text{ecb}(i_{ee}, i_{e7}) \times (4. -
\]
\[
\text{ecb}(i_{ee}, i_{e7}) -
\]
\[
* y_{new}(i_{ee}, i_{e6}) + y_{new}(i_{ee}, i_{e4})) / 6 + \text{ecc}(i_{ee}, i_{e7}) -
\]
\[
+ x_{new}(i_{ee}, i_{e4}) \times \text{epcb}(i_{ee}, i_{e7}))
\]

\[
A(i, i_{il}) = A(i, i_{il}) + \text{dts} \times u_{av}(i_{ee}) \times (y_{new}(i_{ee}, i_{e3}) \times (\text{epca}(i_{ee}, i_{el}) - \text{ecb}(i_{ee}, i_{el}))
\]
\[
\text{epca}(i_{ee}, i_{el}) - \text{ecb}(i_{ee}, i_{el}) -
\]
\[
* (4 \times y_{new}(i_{ee}, i_{e5}) + y_{new}(i_{ee}, i_{e3})) / 5 - \text{ecc}(i_{ee}, i_{el}) -
\]
\[
x_{new}(i_{ee}, i_{e3}) \times \text{epcb}(i_{ee}, i_{el}) +
\]
\[
y_{new}(i_{ee}, i_{e4}) \times (-\text{epca}(i_{ee}, i_{el}) + \text{ecb}(i_{ee}, i_{el})) \times (4.
\]
\[
\text{ecb}(i_{ee}, i_{el}) \times (4. -
\]
\[
\text{ecb}(i_{ee}, i_{el}) -
\]
\[
* y_{new}(i_{ee}, i_{e6}) + y_{new}(i_{ee}, i_{e4})) / 5 + \text{ecc}(i_{ee}, i_{el}) -
\]
\[
+ x_{new}(i_{ee}, i_{e4}) \times \text{epcb}(i_{ee}, i_{el}))
\]

\[
A(i, i_{i2}) = A(i, i_{i2}) + \text{dts} \times u_{av}(i_{ee}) \times (y_{new}(i_{ee}, i_{e3}) \times (\text{epca}(i_{ee}, i_{e2}) - \text{ecb}(i_{ee}, i_{e2}))
\]
\[
\text{epca}(i_{ee}, i_{e2}) - \text{ecb}(i_{ee}, i_{e2}) -
\]
\[
* (4 \times y_{new}(i_{ee}, i_{e5}) + y_{new}(i_{ee}, i_{e3})) / 6 - \text{ecc}(i_{ee}, i_{e2}) -
\]
\[
x_{new}(i_{ee}, i_{e3}) \times \text{epcb}(i_{ee}, i_{e2}) +
\]
\[
y_{new}(i_{ee}, i_{e4}) \times (-\text{epca}(i_{ee}, i_{e2}) + \text{ecb}(i_{ee}, i_{e2})) \times (4.
\]
\[
\text{ecb}(i_{ee}, i_{e2}) \times (4. -
\]
\[
\text{ecb}(i_{ee}, i_{e2}) -
\]
\[
* y_{new}(i_{ee}, i_{e6}) + y_{new}(i_{ee}, i_{e4})) / 6 + \text{ecc}(i_{ee}, i_{e2}) -
\]
\[
+ x_{new}(i_{ee}, i_{e4}) \times \text{epcb}(i_{ee}, i_{e2}))
\]

**c------- chemical reaction and internal mass rate term**

\[
fee(i) = fee(i) + \text{area}(i_{ee}) / 3 \times \text{pri}(i, k) \times (1 - \text{dts} \times \text{akd})
\]

\[
feh(i) = feh(i) + \text{area}(i_{ee}) / 3 \times \text{prh}(i, k) \times (1 - \text{dts} \times \text{prr}(i, k) \times \text{akz})
\]

\[
fer(i) = fer(i) + \text{area}(i_{ee}) / 3 \times (\text{prr}(i, k) +
\]
\[
\text{dts} \times \text{eff} \times \text{akd} \times \text{pri}(i, k) - \text{efq} \times \text{akz} \times \text{prh}(i, k) \times \text{prr}(i, k))
\]

\[
fe(i) = fe(i) + \text{area}(i_{ee}) / 3 \times (\text{prc}(i, k) +
\]
\[
\text{dts} \times \text{akp} \times (1 - \text{prc}(i, k) \times \text{prr}(i, k))
\]

41 continue
\[ a(i,iai) = a(i,iai) + \text{areat}(i) \]
\[ g(i,k) = \text{akp}(1. - \text{prc}(i,k)) \times \text{prr}(i,k) \]

11 continue

100 err=0.
C------- solve \([a]\text{cv}=(fe)\) by iteration
   do 15 imm=1,mn
      i=ncal(imm)
      iai=ncn(i)+1
      if (a(i,iai) .eq. 0.) go to 15
      ffe(i)=fei(i)
      do 20 j=1,ncn(i)
         ffe(i)=ffe(i)-rri(ndcv(i,j),k)*a(i,j)
      20 continue
      cvod=rri(i,k)
      rri(i,k)=thetac*ffe(i)/a(i,iai)+(1.-thetac)*cvod
      err=err+abs(rri(i,k)-cvod)
   15 continue
   if (err .gt. accu) go to 100

200 err=0.
   do 25 imm=1,mn
      i=ncal(imm)
      iai=ncn(i)+1
      if (a(i,iai) .eq. 0.) go to 25
      ffe(i)=feh(i)
      do 30 j=1,ncn(i)
         ffe(i)=ffe(i)-rih(ndcv(i,j),k)*a(i,j)
      30 continue
      cvod=rih(i,k)
      rih(i,k)=thetac*ffe(i)/a(i,iai)+(1.-thetac)*cvod
      err=err+abs(rih(i,k)-cvod)
   25 continue
   if (err .gt. accu) go to 200

300 err=0.
   do 35 imm=1,mn
      i=ncal(imm)
      iai=ncn(i)+1
      if (a(i,iai) .eq. 0.) go to 35
      ffe(i)=fer(i)
      do 40 j=1,ncn(i)
         ffe(i)=ffe(i)-rad(ndcv(i,j),k)*a(i,j)
      40 continue
      cvod=rad(i,k)
      rad(i,k)=thetac*ffe(i)/a(i,iai)+(1.-thetac)*cvod
      err=err+abs(rad(i,k)-cvod)
   35 continue
   if (err .gt. accu) go to 300
err=0.
do 45 imn=1,nn
   i=ncal(imn)
   iai=ncn(i)+1
   if (a(i,iai) .eq. 0.) go to 45
   ffe(i)=fe(i)
do 50 j=1,ncn(i)
   ffe(i)=ffe(i)-cv(ndcv(i,j),k)*a(i,j)
50 continue
   cvod=cv(i,k)
cv(i,k)=thetac*ffe(i)/a(i,iai)+(1.-thetac)*cvod
   if (rii(i,k) .lt. 0.) rii(i,k)=0.
   if (rih(i,k) .lt. 0.) rih(i,k)=0.
   if (cv(i,k) .gt. af) cv(i,k)=af
   err=err+abs(cv(i,k)-cvod)
45 continue
   if (err .gt. accu) go to 400
continue
return
end

subroutine cure
common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/itrcrt/accu,taccu,errvis,thetas,thetac
common/dim/jz
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/se/kf(1800),fht(1800),f(1800)
common/sk/num(1800,10),ncel(1800,10,10),ax(2),ay(2)
common/sj/nov(1800)
common/rzh2/ncn(1800),ndcv(1800,20),nnccn(1800,20,2)
common/eppty/timin,timax,cp0(2000),tf(2000),ekz(1800,45)
common/sc/nep(2000,3)
common/cppty/zh,chk1,chk2,chm1,chm2,g(1800,45)
common/curing/dtc,mature
common/rad1/rad(1800,45),prr(1800,45),prh(1800,45)
common/rad2/rii(1800,45),pri(1800,45),rih(1800,45)
common/far/areat(1800)
common/rrh/ck,rho,cp,alfa,twlow,twup,havg(2000),ccp

update the temperature and conversion

   do 1110 i=1,nde
   do 1110 k=1,jz
Subroutine end

end

call end

call end

call end

continue

decrement (r) = (r - 1)

continue

decrement (r) = (r - 1)

continue

r10
common/sj/ncv(1800)
common/rrh2/ncv(1800),ndcv(1800,20),nncn(1800,20,2)
common/far/areat(1800)
common/curing/ dtc,mature
common/rrh/cp,alfa,twlow,twup,havg(2000),ccp


c------ zero out vector and matrix before calculation

do 1 i = 1, nde
  t(i,1) = twlow
  t(i,jz) = twup
  continue

do 6 k = 2, jz-1

do 11 i = 1, nde

  fe(i) = 0.0
  a(i) = 0.

do 41 ii = 1, ncv(i)

    iee=num(i,ii)
    iel=ncel(i,ii,1)
    ie2=ncel(i,ii,2)
    ie7=ncel(i,ii,7)
    iil=nep(iee,ncel(i,ii,1))
    iii2=nep(iee,ncel(i,ii,2))

c------ internal energy

    fe(i) = fe(i) + (area(iee)/3.)*ccp*prt(i,k)
    a(i) = a(i) + (area(iee)/3.)*ccp

c------ source term due to chemical reaction

    fe(i) = fe(i) + dtc*(area(iee)/3.)*rh*g(i,k)*rho

c------ heat condution at control boundary by linear shape function

    fe(i) = fe(i) - ck*dtc*(b(iee,ie7)*(xsl(i,ii) + xs2(i,ii)) + c(iee,ie7)*(ysl(i,ii) + ys2(i,ii)))/
               (2.*area(iee))*prt(i,k)

    fe(i) = fe(i) - ck*dtc*(b(iee,iel)*
               (xsl(i,ii) + xs2(i,ii)) + c(iee,iel)*
               (ysl(i,ii) + ys2(i,ii)))/(2.*area(iee))*prt(iil,k)

    fe(i) = fe(i) - ck*dtc*(b(iee,ie2)*
               (xsl(i,ii) + xs2(i,ii)) + c(iee,ie2)*
               (ysl(i,ii) + ys2(i,ii)))/(2.*area(iee))*prt(iii2,k)
continue

c------ heat conduction in gapwise direction

\[ fe(i) = fe(i) + ekz(i,k) \]

continue

c------ explicit method

do 15 i = 1, nde
   \[ t(i,k) = fe(i)/a(i) \]
continue

return
end

subroutine curcr

common/tcv/t(1800,45),prc(1800,45),prt(1800,45),cv(1800,45)
common/itcrt/accu,taccu,errvis,thetae,thetas
common/dim/jz
common/cpty/rh,chk1,chk2,chn1,chn2,g(1800,45)
common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/lep(2000,3)
common/sh/xs1(1800,10),ys1(1800,10),cx1(1800,10),cy1(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/si/nca1(1800),bb(1800,10),cc(1800,10)
common/se/kf(1800),fht(1800),f(1800)
common/sk/num(1800,10),ncel(1800,10,2),ax(2),ay(2)
common/sj/ncv(1800)
common/rh2/ncn(1800),ndcv(1800,20),nncn(1800,20,2)
common/far/areat(1800)
common/curing/dtc,mature
common/rac1/rad(1800,45),prr(1800,45),p(rh(1800,45)
common/rad2/rfi(1800,45),pri(1800,45),rhi(1800,45)


Kinetic Parameters used by Ron Hill for DSC results...

\[ \text{eff} = 2.70.2 \]
\[ \text{efq} = 0.1 \]
az = 827363.98
ez = 8127.18
ad = 1.4212e+18
ed = 168615.4
ap0 = 3.1809e+10
ep0 = 52592.7
am = 2.1005
af = 1.0
ret = 8.314

c------ zeroed out vector and matrix before calculation

do 6 k=1,jz

do 11 i=1,nde

    akd = ad*exp( -ed/( ret*pri(i,k) ) )
    akz = az*exp( -ez/( ret*pri(i,k) ) )
    akp = ap0*exp(-ep0/(ret*pri(i,k) ) )
    akp = akp*( 1. - pri(i,k)/af)**am

    iai=ncn(i)+1
    fei(i)=0.0
    feh(i)=0.0
    fer(i)=0.0
    fe(i)=0.0
    a(i)=0.

do 42 ii=1,ncv(i)

    iee=num(i,ii)

    c------ chemical reaction and internal mass rate term

    fei(i) = fei(i) + area(ieee)/3.*pri(i,k)*(1.-dtc*akd)
    feh(i) = feh(i) + area(ieee)/3.*prh(i,k)*(1.-dtc*prr(i,k)*akz)
    fer(i) = fer(i) + area(ieee)/3.*(prr(i,k) +
    1
    dtc*( eff*akd*pri(i,k) - efq*akz*prh(i,k)*prr(i,k)))
    fe(i) = fe(i) + area(ieee)/3.*(prr(i,k) +
    1
    dtc*akp*(1.-prc(i,k)))*prr(i,k))
    a(i) = a(i) + area(ieee)/3.

    continue

    rii(i,k) = fei(i)/a(i)
    rih(i,k) = feh(i)/a(i)
    rad(i,k) = fer(i)/a(i)
    cv(i,k) = fe(i)/a(i)

    if (rii(i,k) .lt. 0.) rii(i,k) = 0.
    if (rih(i,k) .lt. 0.) rih(i,k) = 0.
if (cv(i,k) .gt. af)  cv(i,k) = af

        g(i,k) = akp*(1.-prc(i,k))**prr(i,k)

11  continue

       continue

        return

        end

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

subroutine elegeo

common/sb/xnode(2000,3),ynode(2000,3),nde,nel
common/sc/nep(2000,3)
common/shl/xsl(1800,10),ysl(1800,10),cxl(1800,10),cyl(1800,10)
common/sh/xs2(1800,10),ys2(1800,10),cx2(1800,10),cy2(1800,10)
common/rrh2/ncn(1800),ndcv(1800,20),nnncn(1800,20,2)
common/sj/ncv(1800)
common/sk/num(1800,10),ncel(1800,10,2),ax(2),ay(2)
common/eco/xm6(2000,6),ym6(2000,6),xrc(2000,10),yrc(2000,10)
1 ,xnew(2000,10),ynew(2000,10),uav(2000)
common/si/ncal(1800),bb(1800,10),cc(1800,10)
common/far/areat(1800)
common/edmi/edgmi(2000)

do 5 i=1,nel

c----- calculating the coordinate of the center of each element

        cx(i)=xnode(i,1)+xnode(i,2)+xnode(i,3)
        cx(i)=cx(i)/3.
        cy(i)=ynode(i,1)+ynode(i,2)+ynode(i,3)
        cy(i)=cy(i)/3.

c-------------------------- compute 2* element area ----

        a2=xnode(i,2)*ynode(i,3)+xnode(i,3)*ynode(i,1)
         +xnode(i,1)*ynode(i,2)
         -xnode(i,1)*ynode(i,2)+xnode(i,3)*ynode(i,2)
         -xnode(i,1)*ynode(i,3)
        area(i)=a2/2.0

c----- check the sequence of nodes in an element. if it is c.w.,
c----- change the sequence to c.c.w.
if ( area(i) .le. 0. ) then
  ilter  = nep(i,2)
  nep(i,2) = nep(i,3)
  nep(i,3) = ilter
  xalter  = xnode(i,2)
  yalter  = ynode(i,2)
  xnode(i,2) = xnode(i,3)
  ynode(i,2) = ynode(i,3)
  xnode(i,3) = xalter
  ynode(i,3) = yalter
  area(i)  = -area(i)
  cosine(i) = -cosine(i)
  hdotlo(i) = -hdotlo(i)
 .icount = icount + 1
  write(6,*),'the order of the node have been changed
* in the',i,' element'
endif

5 continue

c-------- calculate b and c for each element
  see formulation for meaning of b and c
  and mid points for each side

  do 7 i=1,nel
    b(i,1)=ynode(i,2)-ynode(i,3)
    b(i,2)=ynode(i,3)-ynode(i,1)
    b(i,3)=ynode(i,1)-ynode(i,2)
    c(i,1)=xnode(i,3)-xnode(i,2)
    c(i,2)=xnode(i,1)-xnode(i,3)
    c(i,3)=xnode(i,2)-xnode(i,1)
  7 continue

  c ***** calculate the mid-points inside the elements and coordinate
  c ***** relative to the centroids inside elements
  do 1099 i=1,nel
    xm6(i,1)=(xnode(i,1)+xnode(i,2))/2.
    xm6(i,2)=(xnode(i,2)+xnode(i,3))/2.
    xm6(i,3)=(xnode(i,3)+xnode(i,1))/2.
    xm6(i,4)=(cx(i)+xm6(i,1))/2.
    xm6(i,5)=(cx(i)+xm6(i,2))/2.
    xm6(i,6)=(cx(i)+xm6(i,3))/2.
    ym6(i,1)=(ynode(i,1)+ynode(i,2))/2.
    ym6(i,2)=(ynode(i,2)+ynode(i,3))/2.
    ym6(i,3)=(ynode(i,3)+ynode(i,1))/2.
    ym6(i,4)=(cy(i)+ym6(i,1))/2.
    ym6(i,5)=(cy(i)+ym6(i,2))/2.
    ym6(i,6)=(cy(i)+ym6(i,3))/2.
    xrc(i,1)=xnode(i,1)-cx(i)
    xrc(i,2)=xnode(i,2)-cx(i)
    xrc(i,3)=xnode(i,3)-cx(i)
    xrc(i,4)=0.
    xrc(i,5)=xm6(i,1)-cx(i)
    xrc(i,6)=xm6(i,2)-cx(i)
\[ \begin{align*}
\text{xrc}(i,7) &= \text{xm6}(i,3) - \text{cx}(i) \\
\text{xrc}(i,8) &= \text{xm6}(i,4) - \text{cx}(i) \\
\text{xrc}(i,9) &= \text{xm6}(i,5) - \text{cx}(i) \\
\text{xrc}(i,10) &= \text{xm6}(i,6) - \text{cx}(i) \\
\text{yrc}(i,1) &= \text{ynode}(i,1) - \text{cy}(i) \\
\text{yrc}(i,2) &= \text{ynode}(i,2) - \text{cy}(i) \\
\text{yrc}(i,3) &= \text{ynode}(i,3) - \text{cy}(i) \\
\text{yrc}(i,4) &= 0. \\
\text{yrc}(i,5) &= \text{ym6}(i,1) - \text{cy}(i) \\
\text{yrc}(i,6) &= \text{ym6}(i,2) - \text{cy}(i) \\
\text{yrc}(i,7) &= \text{ym6}(i,3) - \text{cy}(i) \\
\text{yrc}(i,8) &= \text{ym6}(i,4) - \text{cy}(i) \\
\text{yrc}(i,9) &= \text{ym6}(i,5) - \text{cy}(i) \\
\text{yrc}(i,10) &= \text{ym6}(i,6) - \text{cy}(i)
\end{align*} \]

1099 continue

c-------- construct the control volume.

c for a node, go through the domain to identify the
c elements which include this node.
c use ncv(i) to count the elements involved in node i
c use num(i,j) to list the corresponding element number
c use ncel(i,j,k) to represent the local node number

c for c.v. i and jth element. the local node number
c for an element is 1, 2, and 3. the jth element belonging
c to c.v. i is retrieved through num(i,j) to get the
c corresponding element number

do 10 i=1,nnd
ncv(i)=0

do 12 j=1,nel

do 14 k=1,3

if(nep(j,k).eq.i) then
  ncv(i)=ncv(i)+1
  num(i,ncv(i))=j
  if (k.eq.1) then
    bx=xnode(j,1)
    by=ynode(j,1)
    ncel(i,ncv(i),1)=2
    ncel(i,ncv(i),2)=3
    ncel(i,ncv(i),3)=5
    ncel(i,ncv(i),4)=7
    ncel(i,ncv(i),5)=8
    ncel(i,ncv(i),6)=10
    ncel(i,ncv(i),7)=1
    ax1=xnode(j,2)
    ax2=xnode(j,3)
    ay1=ynode(j,2)
    ay2=ynode(j,3)
    goto 17
  end if
  if (k.eq.2) then
    bx=xnode(j,2)
    by=ynode(j,2)
    ncel(i,ncv(i),1)=3
end if

end if
ncel(i,ncv(i),2)=1
ncel(i,ncv(i),3)=6
ncel(i,ncv(i),4)=5
ncel(i,ncv(i),5)=9
ncel(i,ncv(i),6)=8
ncel(i,ncv(i),7)=2
ax1=xnode(j,3)
ax2=xnode(j,1)
ay1=ynode(j,3)
ay2=ynode(j,1)
goto 17
endif
endif
14 continue
goto 12
17 continue
xmid=(ax1+ax2)/2.
ymid=(ay1+ay2)/2.
dirx=bx-xmid
diry=by-ymid
sdx=cx(j)-(bx+ax1)/2.
sdy=cy(j)-(by+ay1)/2.
xy=sdy*dirx-sdx*diry
if(xy.lt.0.) then
xsl(i,ncv(i))=-sdy
ysl(i,ncv(i))=sdx
goto 16
endif
xsl(i,ncv(i))=sdy
ysl(i,ncv(i))=-sdx
16 continue
sdx=cx(j)-(bx+ax2)/2.
sdy=cy(j)-(by+ay2)/2.
xy=sdy*dirx-sdx*diry
if(xy.lt.0.) then
  xs2(i,ncv(i))=-sdy
  ys2(i,ncv(i))=sdx
  goto 18
endif
xs2(i,ncv(i))=sdy
ys2(i,ncv(i))=-sdx
18 continue
12 continue
10 continue
do 11 i=1,nde
  ncn(i)=0
  do 11 j=1,nde
    do 13 ii=1,ncv(i)
      do 13 k=1,2
        ik=nep(num(i,ii),ncel(i,ii,k))
        if (j.eq.i) then
          ncn(i)=ncn(i)+1
          ndcv(i,ncn(i))=j
          goto 11
        endif
      end
    end
  end
11 continue
do 15 i=1,nde
  do 15 ii=1,ncv(i)
    do 15 k=1,2
      ik=nep(num(i,ii),ncel(i,ii,k))
      do 15 j=1,ncn(i)
        if (ndcv(i,j).eq.ik) then
          nncn(i,ii,k)=j
        endif
      end
    end
  end
15 continue
do 3060 i=1,nde
  areat(i)=0.
  do 3060 ii=1,ncv(i)
    iee=num(i,ii)
    areat(i)=areat(i)+area(iee)/3.
  end
3060 continue
do 3010 i=1,nel
  edgmi(i)=(xnode(i,1)-xnode(i,2))**2+(ynode(i,1)-ynode(i,2))**2
  edgl=(xnode(i,1)-xnode(i,3))**2+(ynode(i,1)-ynode(i,3))**2
  if (edgmi(i).gt.edgl) then
    edgmi(i)=edgl
  endif
  edgmi(i)=edgmi(i)*.5
3010 continue
return
end
APPENDIX B

Experimental Determination of
Propagation Rate Constant
It should be possible to actually compute values for $k_p$ from experimental DSC and ESR data. From equation (5.13), the propagation rate constant is

$$k_p = \frac{1}{[R]} \frac{1}{(1 - \alpha)} \frac{d\alpha}{dt}$$

(B.1)

From DSC results, $d\alpha/dt$ and $\alpha$ are known. The radical concentration is obtained from ESR results. Therefore, experimental values of $k_p$ may be calculated from equation (B.1). For illustration, the experimental DSC and ESR data at 85°C were used to compute $k_p$ as a function of conversion and as a function of time. The results of this procedure are shown in Figures B.1 and B.2.
Figure B.1. Propagation rate constant computed from experimental data versus conversion.
Figure B.2. Propagation rate constant computed from experimental data versus time.
APPENDIX C

MODELLING OF 3-D HEAT TRANSFER IN
COMPRESSION MOLDING DURING FLOW
The energy equation given by equation (3.23) did not include convection and conduction in the X and Y-directions. It may be possible to include these terms by using an exponential shape function in the direction parallel to the average velocity, and a linear shape function in the direction normal to the average velocity [71]. If the local element coordinate system is transformed so that the origin is at the centroid of the element, one axis is parallel to the average velocity, and one axis is normal to the average velocity, then the convective term in the normal direction becomes zero because the average velocity is zero in that direction. The convection/diffusion problem may then be written

\[ \rho U_{av} \frac{\partial T}{\partial \zeta} = \frac{\partial}{\partial \zeta} \left( \frac{k}{C_p} \frac{\partial T}{\partial \zeta} \right) + \frac{\partial}{\partial \eta} \left( \frac{k}{C_p} \frac{\partial T}{\partial \eta} \right) \]  

(C.1)

where \( \zeta \) is in the direction parallel to the flow and \( \eta \) is normal to that direction. The shape function is written in the following form [50, 71]:

\[ T = A \varepsilon \left( \frac{e^{(\zeta-\zeta_{m})/\varepsilon}}{e} - 1 \right) + B \eta + C \]  

(C.2)

where \( \zeta_{m} \) is the maximum value of \( \zeta \) in the element, A, B, and C are parameters to be found, and

\[ \varepsilon = \frac{\alpha_{th}}{U_{av}} \]  

(C.3)

where \( \alpha_{th} \) is the thermal diffusivity.
Differentiating,

\[ \frac{\partial T}{\partial \xi} = Ae^{(\xi - \xi_m) / \varepsilon}, \quad \frac{\partial T}{\partial \eta} = B \]  

(C.4)

Manipulating equation (C.1),

\[
\frac{\partial}{\partial \xi} \left[ \rho U_{av} T - \frac{k}{C_p} \frac{\partial T}{\partial \xi} \right] - \frac{\partial}{\partial \eta} \left[ \rho \frac{C_p}{C_p} \frac{\partial T}{\partial \eta} \right] = 0
\]  

(C.5)

The expression in brackets in the first term of this equation may be written

\[
J_\xi = \rho U_{av} T - \frac{k}{C_p} \frac{\partial T}{\partial \xi} = \rho U_{av} \left[ T - \varepsilon \frac{\partial T}{\partial \xi} \right]
\]  

(C.6)

Inserting equation (C.2) into (C.6),

\[
J_\xi = \rho U_{av} (-A\varepsilon + B\eta + C)
\]  

(C.7)

The second expression in brackets in equation (C.5) may be written

\[
J_\eta = -\frac{k}{C_p} \frac{\partial T}{\partial \eta} = -\rho U_{av} \varepsilon B
\]  

(C.8)
To apply the control volume method, equation (C.1) must be integrated over the control volume surface in the local X-Y plane, at each vertical node position (in the Z-direction).

\[
\int_{A} \left[ \rho U_{av} \frac{\partial T}{\partial \xi} - \frac{\partial}{\partial \xi} \left( \frac{k}{C_p} \frac{\partial T}{\partial \xi} \right) - \frac{\partial}{\partial \eta} \left( \frac{k}{C_p} \frac{\partial T}{\partial \eta} \right) \right] dA = 0
\]

(C.9)

Equation (C.9) can be rewritten using the divergence for the conduction terms as

\[
\int_{A} \left[ \rho U_{av} \frac{\partial T}{\partial \xi} - \nabla \cdot \left\{ \frac{k}{C_p} \frac{\partial T}{\partial \xi} \hat{e}_i + \frac{k}{C_p} \frac{\partial T}{\partial \eta} \hat{e}_j \right\} \right] dA = 0
\]

(C.10)

A useful form of Green's theorem may be written [72]

\[
\int_{A} \nabla \cdot \vec{F} dA = \int_{C} \vec{F} \cdot \vec{n} dL
\]

(C.11)

where \( \vec{n} \) is the outward unit normal vector to the control volume boundary. Applying Green's theorem to equation (C.10),

\[
\int_{C} \left[ \rho U_{av} T \hat{e}_i - \left\{ \frac{k}{C_p} \frac{\partial T}{\partial \xi} \hat{e}_i + \frac{k}{C_p} \frac{\partial T}{\partial \eta} \hat{e}_j \right\} \right] \cdot \vec{n} dL = 0
\]

(C.12)
Inserting equations (C.6) and (C.8),

\[ \int_{C} \left( J_{\xi} \hat{e}_{i} + J_{\eta} \hat{e}_{j} \right) \cdot \vec{n} \, dL = 0 \]  
(C.13)

Inserting (C.7) and (C.8),

\[ \rho U_{av} \int_{C} \left[ (-A \varepsilon + B \eta + C) \hat{e}_{i} - \varepsilon B \hat{e}_{j} \right] \cdot \vec{n} \, dL = 0 \]  
(C.14)

\[ \rho U_{av} \int_{C} \left[ (-A \varepsilon + B \eta + C) n_1 - \varepsilon B n_2 \right] \, dL = 0 \]  
(C.15)

Here, \( n_1 \) and \( n_2 \) are the components of the outward normal vector. Referring to Figure C.1 [71],

\[ \vec{n}_A = (-\eta_3, \zeta_3), \quad \vec{n}_B = (\eta_4, -\zeta_4) \]

Inserting these relations and using Simpson's rule, equation (C.13) becomes [71]

\[ \int_{3-5-0} \left( J_{\xi} \hat{e}_{i} + J_{\eta} \hat{e}_{j} \right) \cdot \vec{n} \, dL = \rho U_{av} \left\{ \eta_3 \left[ A \varepsilon - \frac{4\eta_5 + \eta_3}{6} B - C \right] - \varepsilon B \zeta_3 \right\} \]

\[ \int_{0-6-4} \left( J_{\xi} \hat{e}_{i} + J_{\eta} \hat{e}_{j} \right) \cdot \vec{n} \, dL = \rho U_{av} \left\{ \eta_4 \left[ -A \varepsilon + \frac{4\eta_6 + \eta_4}{6} B + C \right] + \varepsilon B \zeta_4 \right\} \]  
(C.16)
Figure C.1. New coordinate system and outward unit normal vectors [71].
The expressions for A, B, and C have been given elsewhere [71]. The full energy equation for compression molding after transformation using warped-time variables is

\[
\frac{k}{h^2} \frac{\partial T}{\partial \tau} + \rho C_p \left[ \nabla_X \frac{\partial T}{\partial X} + \nabla_Y \frac{\partial T}{\partial Y} \right] = k \left[ \frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} \right] + \frac{k}{h^2} \frac{\partial^2 T}{\partial \xi^2} + Q_R
\]

(C.17)

The expressions given by equation (C.16) may be used to handle the convection/conduction terms in the planar direction after coordinate transformation. The first term in equation (C.17) is discretized as

\[
\frac{k}{h^2} \frac{T^{n+1} - T^n}{\Delta \tau}
\]

(C.18)

and the Z-conduction term is discretized using the finite-difference form as

\[
\frac{T^n_{i-1} - 2T^n_i + T^n_{i+1}}{(\Delta \xi)^2}
\]

(C.19)

Equation (C.17) is solved by bringing all terms related to the temperature at the previous time step, \(T^n\), to the right hand side of the equation, while the temperatures at the next time step, \(T^{n+1}\), remain on the left hand side. The right hand side is known, and the new temperature field is solved by iteration.