Doctoral Thesis

Contribution to robust resin transfer molding

Author(s):
Arbter, Rene

Publication Date:
2008

Permanent Link:
https://doi.org/10.3929/ethz-a-005730612

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
DISS ETH NO. 18108

CONTRIBUTION TO
ROBUST RESIN TRANSFER MOLDING

A dissertation submitted to the
ETH ZURICH

for the degree of
Doctor of Sciences

presented by
RENE ARBTER

Dipl.-Ing. Maschinenbau, RWTH Aachen
born on July 11, 1977
citizen of the Federal Republic of Germany

accepted on recommendation of
Prof. Dr. P. Ermanni, examiner
Prof. Dr. F. Trochu, co-examiner

2008
Abstract

Resin transfer molding (RTM) is an established process to manufacture high quality composite parts with thermosetting matrices. In spite of the economically interesting properties of this process, there are critical issues concerning process reliability. For instance, the quality management and cycle time should be improved in order to make the process more economic. During mold design, injection gates and vents could be placed optimally using simulation software and optimization algorithms. The flow control is critical during the injection.

However, the permeability of the fabric preform is sensitive to handling and can differ in each injection. To achieve an optimal injection, those disturbances need to be considered in simulations. A finite element software for fill simulations (sLIP) has been developed which considers major effects such as anisotropy, gas entrapments, resin curing. A major enhancement of the simulation is a parametric simulation model to include stochastic disturbances. Particular attention is given to the initialization of the stochastic variations, and the evaluation of the resulting process variations.

To optimize process parameters, evolutionary algorithms are coupled with this simulation software. As free parameters, gate and vent locations are considered as well as injection pressures, volume flows, and their timing. Topics like model parametrization and the definition of the optimization objectives are discussed. Optimizations considering process reliability are compared to deterministic optimizations.

In a flow visualization approach, the real permeability distribution can be estimated for each single injection. By comparing simulation values with actual values of sensors, the differences between assumed and real permeability distributions can be determined iteratively. Thus, a model of the permeability distribution can be generated. This procedure is known in structure analysis as ‘model update’ and is applied if assumptions for unpredictable values have to be made. Only a few sensors are required to get precise monitoring of the flow in the closed mold.
Several benefits can come from this visualization technology. In this work, a continuous permeability measurement method has been derived, which allows to measure the saturated in plane permeability in a single experiment. A further possibility is to estimate the quality of a molded part by simulations based on the updated permeability model. Incomplete filling or intolerable joints of flow fronts can be recognized immediately after the injection. Also, it becomes possible to establish a feedback control system for better injection results.
Zusammenfassung


“All models are wrong, but some models are useful”
## Contents

1 Introduction .......................... 1
   1.1 Fiber reinforced composites ................. 1
   1.2 Manufacturing processes .................... 1
   1.3 Resin Transfer Molding ..................... 3
   1.4 Variants of the Resin Transfer Molding process . . . . . . . 4
   1.5 Need for qualified process parameters ............ 4
   1.6 Objective of this thesis ..................... 6
   1.7 Thesis outline .......................... 6

2 Literature review ........................ 9
   2.1 Numerical process simulation .................. 9
      2.1.1 Flow model based on Darcy’s Law ............. 9
      2.1.2 Finite element formulation .................... 10
      2.1.3 Definition of gates and vents ................. 12
      2.1.4 Iterative solution for moving flow front .......... 14
      2.1.5 Resin curing ................................ 17
      2.1.6 Robustness of RTM injections .................. 18
   2.2 Process optimization .......................... 19
   2.3 Permeability detection and flow monitoring ............. 22
      2.3.1 Permeability estimation in control approaches .... 22
      2.3.2 Flow sensing ................................ 25
      2.3.3 Permeability measurement ..................... 27
   2.4 Approach to robust RTM processes ................. 32

3 Concepts and tools for improved simulations .............. 35
   3.1 Node overfill algorithm ........................ 36
   3.2 Detection and tracking of gas entrapments .......... 38
      3.2.1 Extensions in the simulation ................. 39
      3.2.2 Detection of entrapments .................... 40
3.2.3 Tracking of entrapments ........................................ 40
3.2.4 Resin transport .................................................... 43
3.2.5 Simulation results .................................................. 44
3.2.6 Conclusion .......................................................... 47
3.3 Modeling the injection system ........................................ 49
  3.3.1 Modeling approach ................................................ 49
  3.3.2 Components of the injection system ............................ 50
  3.3.3 Connection algorithm ............................................ 51
  3.3.4 Simulation results ................................................ 52
  3.3.5 Conclusion ........................................................ 52
3.4 Flow marking particles ................................................ 52
  3.4.1 Marker movement ................................................ 54
  3.4.2 Simulation costs and marker precision .......................... 57
  3.4.3 Conclusions ....................................................... 59
3.5 Resin curing and viscosity ............................................ 59
  3.5.1 Curing model ...................................................... 61
  3.5.2 Viscosity model .................................................. 63
  3.5.3 Local resin viscosity from flow markers ....................... 64
  3.5.4 Conclusions ....................................................... 67
3.6 Prediction of matrix quality using void marker ..................... 70
  3.6.1 Void formation .................................................. 70
  3.6.2 Mobility of voids ................................................ 71
  3.6.3 Simplified simulation model for moving voids ................. 72
  3.6.4 Conclusions ....................................................... 73
3.7 Conclusions for simulation concepts and tools ....................... 73

4 Process variations and reliability ........................................ 77
  4.1 Parametric simulation model ....................................... 77
    4.1.1 Disturbance cell concept ..................................... 78
    4.1.2 General disturbances ......................................... 81
    4.1.3 Flow channels ................................................ 83
    4.1.4 Experimental investigation of flow channel geometry ....... 98
    4.1.5 Initialization of disturbance cells .......................... 101
    4.1.6 Conclusions ................................................... 102
  4.2 Monte Carlo estimation of process variations ....................... 102
    4.2.1 Initialization of the parametric simulation model .......... 103
    4.2.2 Variation of simulation output ................................ 103
    4.2.3 Estimation of the scrap rate ................................ 105
    4.2.4 Prediction of problem areas ................................ 112
    4.2.5 Conclusion .................................................... 113
  4.3 Conclusions for simulation of process variations .................. 113
## 5 Optimization of processing parameters

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Heuristic optimization</td>
<td>117</td>
</tr>
<tr>
<td>5.2 Numerical optimization algorithms</td>
<td>118</td>
</tr>
<tr>
<td>5.2.1 Formulation of optimization problem</td>
<td>118</td>
</tr>
<tr>
<td>5.2.2 Optimization algorithm overview</td>
<td>119</td>
</tr>
<tr>
<td>5.2.3 Response Surface method</td>
<td>121</td>
</tr>
<tr>
<td>5.2.4 Particle Swarm Optimization</td>
<td>122</td>
</tr>
<tr>
<td>5.2.5 Evolutionary Algorithms</td>
<td>123</td>
</tr>
<tr>
<td>5.3 Optimization parameters</td>
<td>125</td>
</tr>
<tr>
<td>5.3.1 Parametrization of the connector locations</td>
<td>125</td>
</tr>
<tr>
<td>5.3.2 Line connectors and predefined locations</td>
<td>126</td>
</tr>
<tr>
<td>5.3.3 Parametrization of the injection system</td>
<td>127</td>
</tr>
<tr>
<td>5.3.4 Conclusions</td>
<td>128</td>
</tr>
<tr>
<td>5.4 Definition of optimization objective</td>
<td>128</td>
</tr>
<tr>
<td>5.4.1 Topology of objective function</td>
<td>129</td>
</tr>
<tr>
<td>5.4.2 RTM specific optimization targets</td>
<td>131</td>
</tr>
<tr>
<td>5.4.3 Maximum pressure constraint</td>
<td>131</td>
</tr>
<tr>
<td>5.4.4 Robust processes with optimized scrap rate</td>
<td>132</td>
</tr>
<tr>
<td>5.4.5 Optimization of manufacturing costs</td>
<td>133</td>
</tr>
<tr>
<td>5.4.6 Conclusions</td>
<td>134</td>
</tr>
<tr>
<td>5.5 Number of simulation runs for scrap rate estimation</td>
<td>135</td>
</tr>
<tr>
<td>5.6 Local remeshing around gate and vent nodes</td>
<td>138</td>
</tr>
<tr>
<td>5.6.1 Single node connectors in process optimization</td>
<td>138</td>
</tr>
<tr>
<td>5.6.2 Modeling of mold connectors</td>
<td>141</td>
</tr>
<tr>
<td>5.6.3 Applying connector remesh in simulations</td>
<td>141</td>
</tr>
<tr>
<td>5.6.4 Conclusions</td>
<td>146</td>
</tr>
<tr>
<td>5.7 Evolutionary optimization of process parameters</td>
<td>147</td>
</tr>
<tr>
<td>5.7.1 Definition of the objective function</td>
<td>148</td>
</tr>
<tr>
<td>5.7.2 Optimization of connector position</td>
<td>148</td>
</tr>
<tr>
<td>5.7.3 Predefined connector positions</td>
<td>155</td>
</tr>
<tr>
<td>5.7.4 Optimization of configuration</td>
<td>159</td>
</tr>
<tr>
<td>5.7.5 Optimization of sort box</td>
<td>164</td>
</tr>
<tr>
<td>5.7.6 Demonstrator part of VANTEX project</td>
<td>174</td>
</tr>
<tr>
<td>5.7.7 Conclusions</td>
<td>185</td>
</tr>
<tr>
<td>5.8 Conclusions for process optimization</td>
<td>187</td>
</tr>
</tbody>
</table>

## 6 Estimation of the actual permeability distribution

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Update approach for the parametric permeability model</td>
<td>191</td>
</tr>
<tr>
<td>6.2 Implementation in the simulation environment</td>
<td>192</td>
</tr>
<tr>
<td>6.2.1 Levenberg Marquardt Method</td>
<td>193</td>
</tr>
<tr>
<td>6.2.2 Update algorithms for permeability estimation</td>
<td>195</td>
</tr>
</tbody>
</table>
6.3 Virtual experiments .......................... 195
6.4 Permeability estimation for rectangular panel ............ 197
6.5 Permeability estimation with balanced cell influence ...... 198
  6.5.1 Geometry and configuration ........................ 199
  6.5.2 Levenberg-Marquardt minimization ................. 200
  6.5.3 Topology of the objective function ................. 200
  6.5.4 Response Surface and Particle Swarm Optimization 203
6.6 Continuous in plane permeability measurement ............ 207
  6.6.1 Experimental setup ............................ 208
  6.6.2 Numerical methods ............................. 213
  6.6.3 Results of permeability measurements ............... 219
  6.6.4 Discussion of measurement results ................. 223
6.7 Conclusions for permeability estimation .................. 225

7 Conclusions and Outlook .......................... 227

A Flow marker details ............................ 231
  A.1 Detailed description of marker movement ............... 231
  A.2 Generation and storage of neighborship information .... 236

B Resin model ...................................... 241
  B.1 Experimental campaign ............................ 241
  B.2 Evaluation of experimental data ..................... 244
  B.3 Model accuracy for curing kinetics and rheology .... 250
  B.4 Exploration algorithm for resin marker assignment .... 255

C Realization of local connector remeshing ................ 257

Bibliography ........................................ 261
Chapter 1

Introduction

1.1 Fiber reinforced composites

In fiber reinforced composites the good properties of fibers in terms of stiffness and strength to weight ratio (figure 1.1) are combined with a matrix which gives the part stability in the transverse direction of the fibers and holds the fibers in their position relative to each other. Fibers can be of various materials, depending on the application. In the field of polymer matrix composites, structurally loaded parts are commonly made of glass fibers, carbon fibers or polymer fibers with a diameter in the range of 5 to 100 $\mu m$ and a length starting from a few millimeters. The strength of fibers results from the low number of voids, as in thin fibers there is simply no room for voids. Figure 1.2 shows an experimental result for the relation between fiber strength and fiber diameter. It clearly shows increasing fiber strength when the diameter is reduced.

Most parts are designed in layers of unidirectional or woven fabrics with a thickness of 50 up to a few 100 $\mu m$. Due to the combination of stiff fibers with a comparably soft matrix a strongly anisotropic material behavior results. Therefore composite specific design and calculation methods are required, including the consideration of anisotropic and location dependent properties during manufacturing.

1.2 Manufacturing processes

During manufacturing the fibers have to be brought into the desired position and orientation. In addition it is important that the matrix material
Figure 1.1: Specific mechanical properties of metals and fiber reinforced plastics [68]

Figure 1.2: Relation between fiber diameter and strength [42]
impregnates the fibers and that the matrix is consolidated properly. Several processing technologies have been developed

- Wet laminating
- Prepreg
- Resin Transfer Molding and variants
- Injection molding (short fibers, thermoplastic/duroplastic matrix)
- hot press forming
- SMC/BMC: reactive resin with chopped fibers is shaped and cured in hot press molding

For the selection of the best suited process for an economical production many part specific parameters have to be considered. Those are for instance the desired fiber volume content and the number of parts to be manufactured.

1.3 Resin Transfer Molding

In Resin Transfer Molding (RTM) processes, a reinforcing textile fabric is formed to the geometrical shape of the part to be produced. This preform is placed in a mold determining the final shape of the part. The mold can be stiff on both, the upper and the lower side of the part, or on only one sided, with a flexible bag on the other side, working with vacuum and air pressure. The dry preform is impregnated with a liquid matrix resin, injected at either one or several gates. After the curing of the resin, the part is de-molded (figure 1.3).

RTM processes allow the production of geometrical complex parts with a quality similar to the one achieved in autoclave processes. In a broad field of different industrial segments RTM processed parts can compete with state-of-the-art metallic products, from a cost as well as from a performance point of view. Resin Transfer Molding processes are capable to manufacture complex fiber reinforced parts with a high fiber volume content and well orientated fibers. Due to the comparably low cost of the raw materials and preforming technologies, it is applicable for medium size manufacturing series of structural parts.
1.4 Variants of the Resin Transfer Molding process

Particularly for Resin Transfer Molding (RTM) there are many process variants [11], which often combine properties of two or more manufacturing processes. Important variants are:

- Vacuum assisted resin infusion (VARI) with one-sided mold and vacuum bag
- Vacuum assisted RTM with evacuated mold
- Compression RTM with partially opened mold, closing during the injection
- VARI in autoclave for better compression

These injection processes need to be designed with respect to gate locations, where the resin is inserted and the vent locations, where vacuum is applied and where air can escape from the mold. Inappropriate selection of those locations may cause long cycle times, leading to higher production costs. In the worst case a poor laminate quality or even gas entrapments may occur and scrap parts may result.

1.5 Need for qualified process parameters

During the injection voids may occur, which lead to a reduced component quality, require manual finishing or – in the worst case – result in scrap.
Reduced mechanical properties may for instance be caused by confluence lines of flow fronts \([8, 97]\), or by macroscopic gas entrapments which are intolerable in most cases.

Although an injection process should avoid these flaws, in reality the resin flow is often disturbed. Typical disturbances of the resin flow may result from various causes. The most important ones are flow channels near the edge of the cavity or other locations where the geometry changes (i.e. ribs, cascaded cavity thickness, tapered regions, changes in preform layup, ply drop offs), variations in the fiber volume content, leaking cavities, unintentional foldings in the preform, incorrect permeability assumptions and stochastic changes of the fiber architecture (e.g. resulting from shearing, thick regions \([71]\) and fiber washing). Variations in the local resin temperature influence the viscosity and therefore result in a changed flow velocity.

Nowadays, the definition of appropriate process parameters for the resin injection in RTM processes mostly rely upon experience and trial-and-error procedures. This often leads to expensive and time consuming developments, when the realization of several configurations is required. Repeated trials, with all costs for raw materials and preparation of the injection are necessary, sometimes even modifications of the preform and the mold, or a re-manufacturing of the mold. Additionally, as process insight is limited and already slight changes in the process parameters can significantly influence the injection process, finding process parameters which achieve good part quality cannot be ensured.

A major issue for RTM processes is the reliability of the injection process. The permeability distribution within the mold may be different for each injection. Particularly, flow channels can influence the fill pattern significantly. Therefore, joint lines may occur in mechanically high stressed areas or the complete filling of the mold may be prevented. Cycle times may vary due to the permeability disturbances, leading to increased manufacturing costs as well as problems with premature curing of the resin.

For simple geometries, process design based on experience and process intuition can lead to excellent injection processes. However, for complex parts, such as e.g. shell parts with undercuts and confluences, process developments based on intuition often fail. When stochastic disturbances are considered in order to develop reliable processes, optima can no longer be found empirically.
1.6 Objective of this thesis

Resin transfer molding appears to be one of the mature manufacturing process for fiber reinforced composite materials. However, in industrial applications RTM is used far less than its potential might suggest. Major reasons are significant uncertainties concerning the required development costs and times, caused by the commonly used empirical design approach to determine process parameters. In addition, time and cost intense work is often necessary before the first part is manufactured. Without detailed knowledge of the injection process, the fine tuning of the processing parameters is difficult, as process insight is missing. This may result in a not satisfying part quality.

These obstacles prevent many industrial users from fully utilizing the potential of RTM processing technologies, as the processing risks of the RTM process out-weight its possibilities. As a consequence, other processing technologies are preferred and the specific possibilities of RTM, as for instance the integration of inserts, are not utilized.

The main objective of this thesis is therefore to establish tools for a systematic design approach to receive reliable injection parameters and insight of the process. Numerical optimization procedures shall be applied, which do not tend to follow prefabricated solutions as it easily happens with the straight forward intuitive approach. In addition to an optimized process design, online flow monitoring is aimed to improve the process insight. Deviations between simulation and the real process shall be detected, and it may serve a basis for feedback control mechanisms of the injection process.

These improvements can lead to controlled injection processes, as opposed to trail and error procedures. This can enhance the part quality and the calculability of the process development. Together with the optimized and therefore shorter processing times and the increased reliability of the process, lower manufacturing costs can be achieved. This work will hopefully contribute to the full utilization of the RTM process potential.

1.7 Thesis outline

Chapter 2 presents a survey of RTM fill simulations, process optimization, flow monitoring and permeability measurements. An approach to robust processing is presented in section 2.4, which has been derived from the results of this literature review.
Simulation of the injection process  A software library for the “simulation of Liquid Impregnation Processes” (sLIP) has been developed at the Center of Structure Technologies\(^1\) to simulate RTM injection processes. Basic properties are well documented in the literature, as for instance the finite element formulation, the mapping of the iterative fill process and the application of boundary conditions. Besides this, advanced features are discussed in chapter 3. These include the detection and tracking of gas entrapments, modeling of the injection system with improved vent control and simulation of the post pressure, and flow markers for the generation of local information, which depend on the resin history. The flow markers are applied for the simulation of the spatial resin viscosity of highly reactive resin systems, as well as to simulate the spatial void content with a simplified model.

Chapter 4 describes a parametric simulation model to account for variations of the permeability. Statistical data about the distribution of the permeability, the fiber volume content and flow channels can be used. Including stochastic variations in the simulation model allows one to predict the variations in the fill process with Monte Carlo simulations. Instead of fixed values for fill time and fill fraction, the range and the statistical distribution of these output parameters can be predicted. Suitable criteria can be used to quantify the process robustness in terms of the scrap rate.

Heuristic and numerical optimization  Methods to determine optimum processing parameters are presented and discussed in chapter 5, where process reliability is the major topic. In addition, relevant optimization parameters of the injection process are identified, and the definition of the optimization objective, including reliability, is discussed. As the process reliability is assessed in Monte Carlo simulations, the number of required simulation has to be minimized in order to keep the computational effort in a reasonable range. Optimized injection parameters for several geometries and test cases, determined with evolutionary optimization algorithms, are presented in section 5.7.

Visualization of the resin flow  Chapter 6 presents an approach to monitor the actual resin flow within a closed mold. Sensors, integrated in the mold, are used in combination with a parametric simulation model, which is updated according to the measured sensor data. Thus, the spatial permeability distribution can be obtained, allowing to simulate the actual resin flow. On one hand, an improved process insight can be achieved,

\(^1\)www.structures.ethz.ch
on the other hand online quality inspections or online feedback control systems for the injection could be established.

The permeability algorithm can also be applied to an accelerated permeability measurement technique called continuous in-plane permeability measurement (section 6.6). Experimental and numerical procedures are presented, and the reliability of the measurements is investigated. Finally, the results of permeability measurements of three different fabrics are presented.
Chapter 2

Literature review

2.1 Numerical process simulation

2.1.1 Flow model based on Darcy’s Law

Darcy’s Law (eq. 2.1) can be used to describe flows of incompressible liquids through porous media [24]. It states a direct proportionality between pressure gradients and flow velocities. The proportional factor \( \frac{K}{\eta} \) depends on the permeability of the porous medium and the fluid viscosity.

\[
v = -\frac{K}{\eta} (\nabla p + \rho g)
\] (2.1)

with
- \( v \): flow velocity [m/s]
- \( K \): permeability of the porous medium [m²]
- \( \eta \): fluid viscosity [Pa·s]
- \( \nabla p \): gradient due to applied pressure [Pa/m]
- \( \rho \): fluid density [kg/m³]
- \( g \): gravitational constant [m/s²]

If the pressure gradient \( \nabla p \) resulting from the injection pressures is high compared to the gravitational pressure gradient \( \rho g \), the gravity term can be neglected. This is true for most RTM processes, where high injection pressures up to 10 bar are used and the manufactured components have a small height difference between their highest and their lowest point during the injection. For VARI however, gravity may become important because the injection pressure difference is limited to the atmospheric pressure against vacuum. Additionally the manufactured parts may reach heights of 1 m or higher, e.g. boat hulls. Equation 2.2 shows the simplified
Darcy Law without the gravity term as it is used in sLIP to describe the resin flow through the textile preform

\[ v = -\frac{[K]}{\eta} \cdot \nabla p \]  

(2.2)

For modeling the following assumptions are made:

- Fully saturated flow within the wetted area behind the flow front
- Resin can be modeled as a Newtonian fluid
- Fiber preforms are modeled as homogeneous porous medium, the internal fiber structure and micro-scale flow channels are not resolved in the model
- Spatial permeability distribution is independent of time, fluid pressure and resin velocities

### 2.1.2 Finite element formulation

The injection of liquid resins into reinforcement textiles in RTM processes is generally described as pressure driven flow of viscous Newtonian liquids through porous media. As the resins are assumed to be incompressible, the equation of mass conservation for incompressible fluids

\[ \nabla v = 0 \]  

(2.3)

is satisfied in the saturated areas.

The correlation between pressure gradient and fluid velocity is described by Darcy’s Law (eq. 2.1 or eq. 2.2). Because the gravity term can be neglected for most RTM parts, combining the equation of continuity (eq. 2.3) and Darcy’s law (equation 2.2 for neglected gravity), leads to a second order partial differential equation that describes the pressure distribution within the fluid [33]

\[ \nabla(K \cdot \nabla p) = Q \]  

(2.4)

where Q are flow sources.

In general, equation 2.4 cannot be solved analytically. A finite element calculation can be used to solve the equation system. For a finite element calculation of the pressure distribution, the cavity is discretized by a mesh of finite elements. Discretization of the fluid domain with finite elements leads to the global conductivity matrix \( C \) and therefore to the set of equations
2.1 Numerical process simulation

where $p$ is the vector of the pressures of each node, $C$ the conductivity matrix, $\eta$ the resin viscosity and $q$ the volume flow from outside the control volume.

In sLIP conforming elements are used, which means that the continuity of the pressure field along the edges of adjacent elements is satisfied. Pressures are evaluated at each node and the control volumes are assigned to the nodes. This is in contrast to non-conforming elements, where pressures are evaluated at the mid-side points of the element boundaries and the elements themselves are control volumes [33].

For given geometry and permeability values, the boundary conditions have to be applied. Figure 2.1 shows a small model with one vent (node 1, $p=0$), one pressure gate (node 4) and one volume flow gate (node 7). On this example the application of boundary conditions onto the FE matrix system shall be illustrated. Boundary conditions have to be applied to obtain a solvable, non-singular equation system. In RTM fill simulations boundary conditions are prescribed pressures and volume flows at certain locations.

Depending on their type, boundary conditions are applied to the FE model as primary (pressures, left hand side) or secondary (flows, right hand side) boundary conditions. Beside gates and vents, where the pressures or fluxes are known, no fluid flow through the cavity walls can occur ($q = 0$). Equation 2.6 shows the resulting equation system.

$$Cp = \eta q$$

(2.5)
According to the basic matrix multiplication rules, column 1 can be multiplied with $p_1$ and the product vector can be brought to the right hand side. All other known pressure boundary conditions are applied analogously. For this example there is only one more pressure boundary condition, $p_4$. As only the unknown nodal pressures are of interest, the equations given in row 1 and 4 can be deleted. The resulting equation system is

\[
\begin{bmatrix}
    c_{11} & c_{12} & \cdots & \cdots & c_{17} \\
    c_{21} & c_{22} & \cdots & \cdots & \vdots \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    \vdots & \vdots & \cdots & \cdots & \vdots \\
    c_{71} & \cdots & \cdots & c_{77}
\end{bmatrix}
\cdot
\begin{bmatrix}
    p_1 = 0 \\
    p_2 \\
    p_3 \\
    p_4 = p_{\text{in}} \\
    p_5 \\
    p_6 \\
    p_7
\end{bmatrix}
= 
\begin{bmatrix}
    q_1 \\
    0 \\
    0 \\
    q_4 \\
    0 \\
    0 \\
    q_{\text{in}}
\end{bmatrix}
\] (2.6)

Solving this reduced equation system will provide all unknown pressures. In sLIP a fast sparse solver [114] is used for this purpose. The unknown volume flows ($q_1$ and $q_4$) can be calculated by multiplying the conductivity matrix $C$ with the pressure vector (eq. 2.6) in a subsequent step. Applying Darcy’s Law, the flow velocities can be computed on element level from the nodal pressures.

Figure 2.2 shows an example of the pressure distribution in a real part. In addition to the previous example, several elements are located at the flow front as this part is not completely filled. The pressure at the flow front is assumed to be the vacuum pressure, if applied, or the atmospheric pressure otherwise. Therefore boundary conditions are applied to the unfilled nodes at the flow front. Simulation models for the movement of advancing flow fronts will be presented in section 2.1.4.

### 2.1.3 Definition of gates and vents

Injection points (gates) and vents are modeled by applying boundary conditions on selected nodes of the finite element model. Because high resin
volume flows occur at vents and gates, their surrounding regions have a high influence on the injection process. As it is known from analytical models for the pressure field close to injection gates, the local pressure field strongly varies within short distances \([1, 2, 120]\). As the approximation quality of the real pressure field by the finite element simulation depends on the local size of the finite elements, the model must be meshed sufficiently fine to map the strong variations of the pressure filed in the area around the gates and vents (see section 5.6). Linking several nodes to each vent or gate helps to reduce those model induced errors and to obtain a good approximation to the real pressure field \([74]\).

**Vents**  Vents are resin outlets and connect the injection domain to the atmospheric pressure or to the vacuum pressure, if applied. In the simulation the pressure at the vents is used as reference pressure for eventual air entrapments to calculate the entrapped amount of air (section 3.2). For all other simulation purposes only the pressure difference between injection pressures and vent pressure is required. The major properties of vents are:

- Name for identification of the vent
- List of linked nodes
- Pressure boundary condition $p = 0$ for open vents, for closed vents flow rate $q = 0$

- Vents close after the flow front reaches them. Two closing condition modes are used in sLIP
  
  - close vent, if all nodes linked are filled by the fluid
  - close vent, after a specified amount of resin has escaped through the vent, e.g. 2 % of the total component’s resin volume.

**Gates** Gates are injection points and may force a certain pressure or volume flow. Advanced injection devices are able to apply pressure or flow as a function of time. The applied pressure values are pressure differences to vent pressure. The major properties of gates are:

- Name for identification of the gate
- List of linked nodes
- Flux or pressure boundary condition, eventually time dependent
- Type of the injected resin, required for modeling of resin viscosity and curing
- Volume of resin injected by this gate

In section 3.3 an advanced definition of gates and injection devices is introduced, as it is implemented in sLIP. There most properties of gates are transferred to injection devices and the gates only define the connection nodes.

### 2.1.4 Iterative solution for moving flow front

The fill process is transient and the boundary conditions on the finite element model change as the part gets filled. So, an iterative simulation algorithm is required to model the progression of the flow front. Hirt [47] proposed the Volume of Fluid algorithm, working with discrete volumes. These volumes are filled in subsequent time steps to model the flow front and its propagation. The flow front is modeled applying a pressure boundary condition of $p = 0$ onto all unfilled control volumes (CV). When a control volume gets filled the boundary condition is removed. Figure 2.3 shows the definition of a control volume of a node surrounded by triangular shell elements.
Starting with a calculated pressure distribution, the net inflows $q_{\text{in}}$ into each control volume can be determined. The required time $t_i$ to fill the control volume $V_i$ is calculated as

$$t_i = \frac{1 - f_{\text{fill}}}{q_{\text{in}}} \cdot V_i \quad q_{\text{in}} : \text{resin flow into the CV}$$

$$f_{\text{fill}} : \text{fill degree of CV}$$

(2.7)

Because all the boundary conditions remain unchanged until the next control volume is filled, the longest time step for the simulations without changing boundary conditions is

$$\Delta t = \min(t_i)$$

(2.8)

when at least one control volume is filled.

Figure 2.4 summarizes the main parts of the resulting simulation algorithm. From the element model the system of differential equations (eq. 2.5) describing the pressure distribution is built. Applying the current boundary conditions, and solving the reduced equation system will lead to the current pressure distribution. Thus, in each time step the pressure distribution, the nodal fluxes and, using Darcy’s law, the flow velocities can be determined. Knowing the flows into the control volumes, the optimum time step $\Delta t$ can be calculated. Multiplying this time step $\Delta t$ with the flows into the control volumes, the resin supplement added to each control volume. The iterative fill simulation advances until the part has been filled or until all vents are closed.

Bruschke [19] reduced the number of required iteration steps by flagging control volumes with a fill fraction of 99% as filled, accepting a small
Figure 2.4: Iterative simulation scheme of RTM fill simulation
fluid loss. Thus, the number of iterations can be reduced noticeably. Luoma [78] suggested an overfill approach for faster simulation of moving flow fronts. The basic idea is to introduce a pseudo compressibility, allowing an increased filling of control volumes. The excessive fluid expands in the next simulation step. According to the level of the pseudo compression $\frac{\rho}{\rho_0}$, the pressure distribution is influenced and the excessive resin is re-distributed in the following iteration steps.

2.1.5 Resin curing

RTM fill processes are often simulated under the assumption of constant mold temperature and constant resin viscosity. This is feasible for slow curing resin systems in isothermally heated molds. Advanced economical processes require shorter cycle times and therefore shorter curing times. Thus, fast curing resin systems, with a pot time in the range of the filling time become interesting.

For curing of reactive resin systems a high number of models are known in literature, mostly based on Arrhenius law (see e.g. [101]). To model the curing kinetics of the resin a model proposed by Henne is used [46].

$$\frac{d\alpha}{dt}(T, \alpha) = \left(\text{Scaling factor}\right) \cdot \left(\text{Temperature dependancy}\right) \cdot \left(\text{Number of reactive links}\right) \cdot \left(\text{Auto-catalytic term}\right)$$

(2.9)

with $\alpha_{\text{max}}(T) = 1 - e^{(-\beta \cdot (T - T_{\text{solid}}))}$

This model is based on a modified Arrhenius relation and is defined by the following characteristics:

- All reactions are related to the temperature $T' = T - T_{\text{solid}}$ above the solidification temperature of the resin
- Exponential dependency of the curing rate on the temperature
- Reduced reactivity of the resin with increasing degree of cure, due to decreasing number of reactive links by $m$-th order
- Auto catalytic reaction by $n$-th order

Henne [46] models the viscosity as
\[
\eta(T, \alpha) = \eta_\infty \cdot e^{\left(\frac{T_{rheo}}{T_{solid}} + B \cdot \alpha\right)}
\]  

(2.10)

with

- \(B\): Curing influence on viscosity
- \(T_{solid}\): Solidification temperature of the resin
- \(\eta_\infty\): Reference viscosity (uncured, high temperature)
- \(T_{rheo}\): Rheological reference temperature

All model parameters of the curing and rheological models are determined using least squares fit of experimentally determined calorimetry and viscosity data, respectively.

The resin viscosity depends on temperature and the curing degree. The curing degree can be integrated starting from an initial degree depending on temperature history. To include the local viscosity state in fill simulations, the curing history of resin at each point has to be known. Deleglise [27] predicts the local curing degree for an injection with constant injection rate, assuming constant resin age and history for each point. However, this assumption is only valid for simple fill behaviors, where the local flow velocities do not change during the fill process.

### 2.1.6 Robustness of RTM injections

Between fill simulation and carefully prepared injections a very good agreement in terms of the flow front propagation can be obtained [5]. However, in industrial applications there are unavoidable variations in the preparation of the injection, resulting in significant differences between fill simulations on the one hand and the real filling processes on the other hand. Each injection differs due to a slightly different permeability distribution. These differences typically result from various causes like permeability disturbances from handling, fiber washing, flow channels, mold deflections, and varying core dimensions. Variations of the processing parameters influence the filling process as well. For instance higher resin temperatures result in decreased resin viscosities, and therefore in higher flow velocities.

General preform disturbances are mainly induced by the handling of the preform. Local fiber shearing, foldings, changes in the area weight etc. cannot be completely avoided. Many publications focus on the variation of the preform permeability under ideal conditions, giving values in the range of ±5..30%, depending on the fabric type and the author [77, 83, 93, 92, 48]. Woven fabrics seem to have a higher variation than non woven fabrics.
Flow channels may occur at geometrically predestined locations like edges and other locations where the geometry changes, for example ribs, cascaded cavity thickness, tapered regions, change in preform layup or ply drop offs [71]. They may have a major influence on the resin flow and on the fill pattern. Li [73] investigated their occurrence for a specific configuration experimentally, and found that the ratio of flow channel permeability to the undisturbed permeability can be well approximated by a Weibull distribution.

To include a flow channel in Darcy based fill simulations, an equivalent permeability can be calculated [89, 17]. Therefore, a solution for the flow velocities of a Navier-Stokes flow can be calculated and the volume flow through the flow channel area can be calculated. Setting this volume flow equal to the Darcy flow through the same cross section with an equivalent permeability and solving the equation provides the equivalent permeability. Hammami [45] calculates the equivalent permeability making simplifications. The equivalent permeability is calculated for a tube with the same cross section as the real flow channel, accepting errors arising from cross sections differing from the circular shape. For a rectangular flow channel of height $h$ and width $d$ he obtains:

$$K_{eq} = \frac{R^2}{8} = \frac{hd}{8\pi}$$  \hspace{1cm} (2.11)

### 2.2 Process optimization

Today’s development of structural parts manufactured by Resin Transfer Molding is still based on time consuming try and error procedures. Numerical flow simulation software can contribute to design parts more efficiently and additionally reduce tooling costs. Usually, optimizations are performed heuristically by modifying the processing parameters like gate and vent locations as well as the injection pressures, injection fluxes and their timing. The more complex a part geometry gets, the more parameter combinations become possible and the less clear the process becomes on an intuitive level. Using automated optimization, time consuming simulations can be run without human interaction.

All numerical optimization procedures require an objective definition. Mostly, a single value function is defined, rating the quality of the simulation output for the investigated process parameter set. Jiang [53] reviews several possible process indices, those include the fill time, void fraction, inlet pressure, flow front angles. A process performance index is suggested, considering the fill time, and the variation of the distance from the flow front to an outlet.
Lin [74] investigated strengths and weaknesses of stochastic and gradient-based optimizations for RTM processes. For multi-parameter problems, gradient-based methods are recommended due to their fast convergence. To place gates and vents on existing nodes, a good resolution of the mesh is required so that the gradients can be calculated properly. Stochastic optimization algorithms like evolutionary optimizations are not recommended because of their slow convergence rate. Lin discusses the problem of applying gates on single nodes in the same paper. As the equivalent inlet radius depends on the size of the adjacent elements, errors in the simulation will occur.

Li [73, 72] included statistic variations due to flow channels in a process performance index. Thus, process robustness can be included in the process optimization. For process optimization a complex multi-stage algorithm has been used. The “graph-based two phases heuristic” optimization developed by [125] reduces the number of required simulation runs with the original model:

1. The geometry is replaced by graph representation of the geometry reducing the search domain.
2. Starting with an exhaustive optimization on the reduced graph geometry, an approximate optimum solution is found.
3. Starting at this solution, a local search algorithm is used, stepwise reducing the search radius.

Several researches have coupled evolutionary optimization strategies in order to automate RTM process optimization (figure 2.5). Optimization parameters are the gate and vent locations, aiming at minimum fill times [85, 59, 82].

Starting with an arbitrary or predefined set of process parameters (individuals), simulations are run. Those parameters may be gate locations, vent locations and injection flow rates. The simulation output is assessed by an objective function, giving a performance index. Barandun takes flow front confluences into account that influence the matrix porosity [7, 8].

To reduce the numerical effort, meta model methods can be used as described by Karakasis [56]. The initial population is evaluated by a meta model. The evaluated individuals are sorted by their fitness, and the best individuals are evaluated by the exact model. Thus, bad individuals do not have to be evaluated by the exact model, saving evaluation time. Evaluations of the exact model are stored in a database and can be used to improve the meta model. Meta models can e.g. be response surface models or neural networks [81] or coarse variants of the exact models.
2.2 Process optimization

Figure 2.5: Comparison of process optimization using an heuristic approach and evolutionary algorithms [7]
2.3 Permeability detection and flow monitoring

2.3.1 Permeability estimation in control approaches

For all feedback control mechanisms of the injection process, knowledge of the fill pattern is mandatory. This may be explicit monitoring of the propagation of the flow front, or implicit knowledge in more abstract control approaches. However, as the fill pattern directly depends on the permeability distribution within the mold, the permeability distribution is determined in all control approaches.

Feedback control of the injection process is intended to ensure an optimum filling of the part. Variations of the preform permeability shall be compensated by suitable control of the injection conditions at the various gates. The major objectives of injection feedback control are to avoid air entrapments, drive joint lines to uncritical areas, reduce porosity in the cured part, minimize the injection time and to achieve a reproducible part quality.

Several contributions to control the injection process can be found in the literature. Generally three different approaches can be identified:

- **Decision tree.** Advani et al. [71, 29, 50, 28] and Weyrauch [122] propose to build a database, called decision tree, of possible disturbance scenarios and suitable control actions that can be applied when the predefined disturbance scenario is detected. Prior to the injection, simulations are run to simulate the effect of all expected combinations of disturbances. Specific unique properties have to be defined for each scenario, thus, the present disturbance scenario can be identified with sensors that are integrated in the mold. Then a predefined specific control action is selected which has been optimized offline, prior to the injection. For each combination of events (in variable intensity) the appropriate reaction has to be predefined. As the number of scenarios grows exponentially with complexity of the part, this approach is obviously limited to simple geometries. In the current development state, it requires quite a lot of work to adapt it to the processing of new parts.

- **Abstract mathematical control.** Sun et al. [15, 9, 126] suggests a control approach based on a linear relation between gate pressures and sensor values.
\[ \vec{P}_i = [C_k] \cdot \vec{P}_j \]

\( \vec{P}_i \) : Sensor values

with \( \vec{P}_j \) : Gate values

\( C_k \) : System matrix

This relation at a given time is described by the system matrix \( C_k \). It can be calculated in two different ways. The first possibility is to run simulations online during the injection and determine the influence of a pressure change at each gate on the sensor signal. For each gate one simulation with slightly changed injection pressure has to be run. Thus, a system response matrix for the sensor signal in dependence on the gate pressures can be obtained. As simulations are based on an undisturbed model, the system response matrix \( C_k \) is an approximation for the true response of the actual system.

The second possibility to determine \( C_k \) uses the sensor history data instead of online simulations. A least squares minimization between measured and sensor data predicted with the momentary matrix is used to update the system matrix in each time step. During the first few control steps, the system matrix is not yet built and no control action can be taken. Therefore, the control steps should be chosen rather short. Moreover, a variation of the gate values is required to obtain a response from the injection and to determine the system matrix.

Having determined the system behavior, control actions can be taken, minimizing the difference between the actual and the desired sensor values, which have to be predefined.

No statement about the actual flow in the mold is made by this approach. So this control methods depends on the usage of a sufficiently high density of linear flow sensors, assuring a good mold filling if the predefined sensor values are met. I.e., this control approach cannot be used with pressure sensors, because driving the pressure sensor values to predefined values does not necessarily result in complete filling of the mold.

- **Permeability estimation.** For linear 1D injections, analytical solutions for the discrete permeability in predefined zones of a given length can be given [14]

Nielsen and Pitchumani [91] built up a feedback control system which consists mainly of three parts
1. Fuzzy logic based permeability estimation. The rules were programmed with fuzzy clustering technique [4], trained with numerical simulation data.

2. A fast flow simulator based on an artificial neural network (ANN). This has been trained with the same data like the fuzzy logic permeability estimation.

3. Online optimizer for injection parameters based on simulated annealing to achieve the desired flow scheme. The optimizer uses the ANN to generate improved injection parameters.

It was found that the permeability estimation could reveal the permeability variation in the rectangular mold investigated. The controller consisting of the ANN simulator and the simulated annealing optimizer could compensate various disturbances including flow channels, and drive the flow uniformly through the mold. The major difficulty of process feedback control, which is online sensing of the flow, has not been discussed in detail. Instead, full information about the flow front as it might be obtained from camera monitoring has been assumed (which is not feasible for closed mold processes).

In a further publication [90], Nielsen and Pitchumani present a control approach using a fixed flow rate schedule in combination with a finite differences flow simulation. For the considered three gate injection, the predefined control possibilities include 27 flow rate combinations. In each time step, the combination achieving the flow velocities closest to the predefined flow velocities is applied. Again, full flow front tracking is required.

Ding et al. [30] used a sensor array with 54 pressure sensors to determine the pressure distribution within a rectangular mold. For four test cases of intentionally and reproducibly disturbed preforms, the actual spatial permeability distribution was evaluated. An isotropic fabric has been assumed to simplify the problem. The permeability field could be calculated from the pressure using a finite difference calculation. In a similar approach of Kim et al [61] the local permeability distribution within the fabric is evaluated at five predefined zones.

Gagel et al. [37] suggests to measure the pressure distribution within
2.3 Permeability detection and flow monitoring

the mold with air flow measurements prior to the injection. The rectangular test geometry is equipped with 20 pressure sensors (0-20 mbar), from which a pressure field is derived. Thus, relative permeability values can be estimated. The target is to predict the fill success.

An iterative permeability update method has been used by Gokce et al. [40]. To characterize the in-plane permeability of a distribution medium and the through-thickness permeability of a fabric simultaneously, experiments have been coupled with numerical simulations. The model permeability values are modified iteratively, until simulated and experimentally determined flow front positions agree. A fixed relation is used for permeability update

\[ K_{i,j+1} = \left( \frac{t_{i,exp}}{t_{i,j}} \right)^p K_{i,j} \]

with

- \( i \): permeability region
- \( j \): update iteration step
- \( t_{i,exp} \): experimental arrival time
- \( t_{i,j} \): simulated arrival time in iteration step \( j \)
- \( K_{i,j} \): assumed permeability for region \( i \) in iteration step \( j \)
- \( p \): update control parameter

All these approaches have been realized at lab scale or as simulation model. The geometry of the cavity is either a flow channel with disturbed flow (1.5D) or a plane geometry (2D) [9], mostly with (line-) injection gates at one edge of the mold. Only the decision tree approach has been implemented for more complex geometries with corners, tapered regions, ribs and thick sections by Advani et al.

2.3.2 Flow sensing

Monitoring of the resin flow in closed mold injection processes requires appropriate sensing techniques. Various types of sensors can be used and were investigated in the past. Sensors for flow monitoring can be separated into sensors remaining in the part and sensors mounted in the mold. Sensors in the part may be used e.g. for health monitoring, but they might also affect the fill behavior and the mechanical properties. Here, only reusable sensors that are fixed in the mold are considered.
Optical tracking of the flow front with cameras may be used in vacuum bag processes. For the most processes, manufacturing complex structures or using stiff metal molds, cameras obviously cannot be used.

The SMARTweave [118] system uses a rectangular grid of wires embedded in the fabric to detect the flow front at discrete points. When the resin flow front reaches the intersection point between two wires, the electrical resistance between to separated wires drops due to the conductivity of the resin.

Luthy [79] has developed linear sensors to detect the flow front position. Two parallel wires are mounted on the surface of the mold. The further the wires are covered by resin, the lower drops the electrical resistance between the wires.

The application of all sensor principles using the electrical conductivity of resin are limited to non conductive fabrics.

Ultrasound sensors can detect the arrival of the flow front [80]. As an advantage compared to most other sensor types, ultrasound sensor can be mounted on the mold without getting in contact with the part. Additionally, Ultrasound sensors can be used for cure monitoring [111]. However, for fill monitoring ultrasound sensors provide very limited information, namely the arrival time of the flow front at the sensor location.

Tekscan\(^1\) pressure sensor membranes as used by [16] can be used to measure the pressure field. They are equipped with a grid of pressure sensors with a density of up to \(\approx 1\) sensor per square millimeter. As the membrane is in direct contact with the fabric, the total pressure of fluid pressure and preform compaction pressure is measured. So, it is difficult to differentiate between resin pressure on the one side and reduced compaction pressure of the preform due to lubrication after wetting and due to deflections of the mold on the other side. Limitating in serial production will probably be the durability of the sensor membrane in contact with the resin.

Pressure sensors can provide useful information about the pressure field after the flow front has passed the sensor [2]. The integration in the mold is difficult because the sensor should not be in direct contact with the preform (cmp. Tekscan sensors). A transmitter fluid like oil can be used on the lower part of the mold, connecting the sensor with the preform through a narrow drill hole.

Electric time-domain reflectometry (TDR) sensors can be used to track the flow front during the injection process and they can be used to monitor the resin curing. High frequency signals are send over electric wires, which are partly reflected where dielectric properties in the area around

\(^1\)www.tekscan.com
the sensor wires change. Because the dielectricity of resin is higher than the dielectricity of air, the flow front can be detected. The sensor is electrically insulated and can be used with glass and carbon fabrics. It can be embedded in the cavity, leaving minimal marks on the molded component. Its precision to detect the flow front position is within few millimeters [107, 31].

In this work, pressure sensors will be used exclusively, although the methods can be used with all other presented of sensor types as well.

2.3.3 Permeability measurement

The permeability of the used fabrics is a major input required for RTM simulations. As described in section 2.1.6, variations of the permeability are high. Therefore, many measurements are required to obtain reliable permeability data for the mean value and the variance. Besides the stochastic variations of the permeability itself, there are further errors in the measured values resulting from the experimental setup. These are for instance the limited accuracy of the used sensors, digitalization errors of the analog to digital converters and the precision of the cavity height. Additionally, errors may result from the processing method used to evaluate the permeability values from the measured data.

**Numerical calculation** There are several approaches to calculate and numerically predict the permeability tensor $K$ [12, 106, 124]. However, due to the complexity of the geometrical configuration of the fiber structures, these models are not accurate and reliable enough to replace experimental permeability measurements. Recent results from Verleye [117] are very promising, predicting the relative permeability under various influences, for instance the influence of shear on the permeability.

**Unidirectional measurement** Directional permeabilities can be measured in one-directional flow experiments [10]. The pressure distribution decreases linear from the inlet to the flow front. For both, pressure and volume flow controlled experiments, an analytic solution for the pressure and the velocity field can be given. In pressure controlled injections, the flow front position $x_f$ can be calculated by [34]:

$$x_f(t) = \sqrt{\frac{2K}{\phi \cdot \eta} \cdot P_i \cdot t} \sim \sqrt{t}$$

Evaluating the flow front positions at given times, the permeability can be determined.
Figure 2.6: Radial propagation of the liquid into the fiber structure with the principal permeabilities $K_1$ and $K_2$ and the orientation $\varphi$.

**Unsaturated in-plane permeability** Although the linear injection has some advantages because of the simple analytical description, many experiments have to be run to evaluate the full in plane permeability data $K_1$ and $K_2$ and the orientation $\varphi$, and achieve a good level of reliability. Adams et al. [1] introduced a radial injection method to measure the in-plane permeability in a single experiment. Resin or a test fluid is injected in the dry fabric through a circular gate. The flow front expands, forming a circular flow front for isotropic fabrics or an elliptical flow front for anisotropic fabrics (figure 2.6). Some effects are not considered in the model, including the following points:

- constant fluid viscosity is assumed, and the fluid is incompressible
- a homogeneous and inelastic preform of constant thickness is assumed
- capillary forces at the flow front are neglected
- no gravitation, the fluid pressure at the flow front is zero

For incompressible fluids the equation of continuity is

$$\nabla v = 0$$ (2.12)
2.3 Permeability detection and flow monitoring

Combining equation 2.12 with Darcy’s Law (eq. 2.2) leads to the Laplace equation, describing the pressure distribution within the fluid. The Laplace equation can be expressed in polar coordinates as

\[
\frac{1}{r} \cdot \frac{dp}{dr} + \frac{1}{r^2} \frac{d^2p}{dr^2} = 0 \quad p : \text{ pressure} \quad r : \text{ radius} \tag{2.13}
\]

In the case of homogeneous isotropic media the flow front is circular. Flow velocity and pressure distribution are equal in each radial direction. The Laplace Equation (2.13) is solved for radial injection by

\[
p(r) = a \cdot ln(r) + b \tag{2.14}
\]

Considering the pressure boundary conditions at the injection gate \( p(r_0) = p_0 \) and at the flow front \( p(r_F) = 0 \) the pressure distribution within the fluid can be determined (figure 2.7).

\[
p(r) = p_0 \cdot \frac{ln\left(\frac{r}{r_F}\right)}{ln\left(\frac{r_0}{r_F}\right)} \quad p_0 : \text{ injection pressure} \quad r_0 : \text{ radius of the injection gate} \quad r_F : \text{ flow front radius} \tag{2.15}
\]

For isotropic preforms, the pressure gradient at the flow front can be derived form eq. 2.15 and inserted in Darcy’s Law yielding the flow front radius

\[
\left(\frac{r_F}{r_0}\right)^2 \left(2 \cdot \ln\left(\frac{r_F}{r_0}\right) - 1\right) + 1 = 4 \frac{K \cdot P_0}{\eta r_0^2} \cdot t \tag{2.16}
\]

Plotting the left hand side over time results in a straight line going through the origin. Thus, the permeability \( K \) can be derived from its slope.
For anisotropic performs a set of governing equations is formulated, shifting the anisotropic problem to a isotropic problem with an elliptical injection gate. As the influence of the injection gate shape diminishes with advancing flow front, an analytical approximation solving the equation system could be found. Additionally, a finite elements approximation has been investigated showing good agreement with the analytical solution.

Chan and Hwang [22] proposed a simpler solution for an-isotropic permeabilities. A coordinate transformation mapping the elliptical (an-isotropic) system into an equivalent circular (quasi isotropic) coordinate system. The shape of the injection gate is neglected, and the permeabilities can be calculated using the equations of the circular isotropic injection. The permeability tensor $K$ can be derived from the orientation and the major and minor axes of the flow front ellipse. The corresponding equivalent permeability is the geometric average of the two principal permeabilities

$$K_e = \sqrt{K_1 \cdot K_2}$$

where $K_1, K_2$ : principle permeabilities

$K_e$ : equivalent permeability

A detailed evaluation procedure for the previous two contributions is presented by Weitzenböck [120, 121]. Additionally, the influence of the gate size on the evaluated permeability is investigated as a function of the flow front radius. From these results, an empirical equation for the minimal flow front radius required for reliable permeability data is derived.

Beside the high experimental effort for experimental permeability measurements, the evaluation is time consuming as well. So, experimental methods and sensing techniques allowing automated evaluation were investigated. The method used by Weitzenböck relies on optical evaluation of the flow front. Glass molds, however, commonly show significant deflections under the injection pressure and distort the results. To use stiff metallic molds appropriate sensors are required. In a work of Kissinger [63] a metallic mold equipped with six dielectric linear sensors has been developed. The experiments are evaluated by fitting an ellipse to the flow front position sensed by three selected sensors. Several further permeability measurement systems with automated evaluation were developed, e.g. using pressure sensors [2] or detecting the flow front arrival time using electrical sensors [48, 83].

**Continuous saturated permeability measurement** A step towards faster permeability determination are continuous methods, where the permeability can be measured at several fiber volume contents and injection
2.3 Permeability detection and flow monitoring

flow rates in a single experiment. Thus, a reduced amount of preform material is required and more permeability data can be obtained with the same experimental effort.

For one-dimensional permeability measurements, an analytical solution for the pressure and flow velocity field is available. So, the saturated permeability of a preform at a varied injection pressure and fiber volume content can be evaluated. A major difficulty in one-dimensional measurements is the sealing of the cavity and the preform edges to avoid flow channels in the flow direction. Stadtfeld [108] presented a corresponding experimental setup. The edge sealing was realized using closed cell rubber foam, which can be compressed during the measurement.

Another continuous measurement technique for saturated permeability measurement has been presented by Buntain [21]. It avoids the difficulties of sealing, but it is limited to isotropic fabrics. The preform is placed between circular plates, mounted in a mechanical test machine, which controls the cavity height. Prior to the measurement, the preform is wetted. Then the cavity is closed at a constant velocity $\dot{h}$. An analytical solution for the permeability $K$ has been derived

$$K = -\frac{\eta R_0^2}{4P_0} \cdot \frac{\dot{h}}{h} \quad \text{with} \quad h : \text{cavity height} \quad \text{R}_0 : \text{Radius of circular plates} \quad \text{P}_0 : \text{fluid pressure at center} \quad \eta : \text{fluid viscosity}$$

A very similar approach has been presented by Pomeroy [99] who used airflow for the measurements. He also used circular plates mounted in a testing machine, and the method is also limited to isotropic media. However, a controlled airflow is injected in the center of the plate during the measurement.

Kim et al. [62] presented a gas flow based in-plane permeability measurement. 20 cylindrical ports were drilled in a rectangular metallic mold, allowing the connection of gates, vents or pressure sensors. Thus, several test configurations are possible, generating different pressure distributions. The sensor data could be evaluated with respect to the in-plane permeability data using finite element simulations coupled with an optimization algorithm. Flow channels at the edges of the rectangular preform, which can have a significant influence on the evaluated permeabilities have been discussed, but cannot be avoided reliably. The principal direction of the permeability has not been evaluated in this work, but it was assumed to be parallel to the mold coordinates.
2.4 Approach to robust RTM processes

The difficult design of robust injection processes can benefit from a systematic design approach. The development time of stable processes becomes more predictable, thus reducing the planning and investment risks. A critical issue for economical success is the robustness of the individual RTM process, meaning in detail:

- Low scrap rate and thus minimum expenses for raw materials and labor
- Constantly high part quality minimizing the effort of finishing
- Reliability of production rates ensuring that production targets are met

A general trend to robust processes can be observed in computational design methods. Dynardo\textsuperscript{2} develops the optimization software “OptiSLang - The Optimizing Structural Language”, a general optimization tool, supporting various optimization methods. Its core feature is the statistical evaluation of influence parameters on the final product. Thus, the robustness of developed products and processes can be predicted and a specific quality management can be established for the relevant parameters.

Simulation of sheet metal forming is an important software tool in the automotive industry. AutoForm\textsuperscript{3}, one of the market leaders in the numerical simulation of sheet metal forming, has developed “AutoForm Robustness Solution” to predict the problem areas of formability, and to predict the scrap rate. Important properties of semi-finished products and processing parameters for successful metal forming can be identified.

Specifically for RTM, two possible ways are investigated to make the injection process more reliable compared to the current industrial standard. One is to design the injection process in a way that it can tolerate flow channels and other disturbances. The effect of stochastically occurring disturbances has to be included in the optimization. The other possibility is to actively control the flow during the injection, i.e. to detect disturbances and counteract by modifying the injection pressures and volume flows. Both ways rely on numerical process simulations and may be combined.

Thus, three fields of required improvements and developments have been identified:

\textsuperscript{2} DYNARDO Dynamic Software and Engineering GmbH, www.dynardo.de
\textsuperscript{3} AutoForm Engineering GmbH, www.autoform.com/products/solution_robustness.html
• **Fill simulation:** Deeper insight into the injection processes within the mold prior to the tool design can be obtained from fill simulations. Therefore, a finite element simulation of the injection processes should include

  – detection and tracking of gas entrapments
  – modeling of the injection system
  – flow markers to track resin particles of the flow front and to model resin curing
  – a parametric model including permeability disturbances to estimate process stability in Monte Carlo simulations

Chapter 3 provides an overview about modeling and simulation methods for RTM-processes. Process variations and Monte Carlo estimations of the scrap rate will be discussed in chapter 4.

• **Numerical process optimization:** Based on simulations coupled with numerical optimization algorithms, like for instance evolutionary algorithms, optimum process parameters can be identified. For optimizations the topics

  – parametrization of gates and vents in order to determine their optimum location in the mold
  – optimization targets, including process robustness in the objective function

will be discussed.

• **Permeability estimation:** Sensor information will be used in coupled simulations to compute a permeability model that is close to the current permeability distribution in the mold. Based on this knowledge, control actions can be taken during the injection to counteract undesired flow patterns, and the component quality can be determined already during the curing stage.

A variant of the approaches of Kim [61] and Ding [30] is developed in this work. Major topics are

  – the number of required flow sensors
  – suitable permeability update algorithms
  – and the extension to arbitrary geometries.
Chapter 3

Concepts and tools for improved simulations

The availability of appropriate process simulation models and tools is a major aspect of systematic and economical process developments of cost-efficient, reliable and fast RTM processes. Due to better process understanding it can help to considerably reduce the time, costs and risks for the development of process parameters for new parts. At the same time the quality of the manufactured parts can be improved.

Detailed filling information can be obtained from injection simulations of RTM processes. Basic simulation techniques for RTM fill processes were summarized in chapter 2.1. Basic RTM injection simulations use Darcy’s law to model resin flows through textile fabrics, providing process information about

- Approximate filling time
- Occurring pressures
- Flow front shape and propagation

All required methods and algorithms were implemented and tested in the in-house developed finite element software for the “simulation of Liquid Impregnation Processes” (sLIP). This finite element simulation has been derived from the “Finite Element Library Experiment” (FELyX [66]). Thus, full access to all flow data can be guaranteed, and additionally required features can be implemented.

The accuracy of a simulation result depends on the precision of the material data as well as on the physical effects considered and implemented.
in the simulation. Beside the basic simulation requirements, advanced features for more accurate simulation results are necessary:

- **Node overfill algorithm** (section 3.1)
  As larger models may be simulated within the same time if accelerated simulations are available, the precision of simulations is generally increased.

- **Detection and tracking of gas entrapments** (section 3.2)
  As gas entrapments are a major cause for scrap parts, their reliable detection is required for the prediction of the final fill degree.

- **Modeling of the injection system** (section 3.3)
  Flow volumes in the injection tubes may vary with time if several gates are used. Copying the real setup to the simulation simplifies the modeling.

- **Flow marking particles** (section 3.4)
  Spatial information depending on the history of the resin can be generated with flow markers. One application is the simulation of the resin curing.

- **Resin curing and viscosity** (section 3.5)
  Fast and therefore economic processes require fast curing resins. To ensure processibility, the curing kinetic and its influence on the viscosity has to be mapped in simulations.

- **Prediction of matrix quality** (section 3.6)
  Particularly for RTM processes, the laminate quality is critical issue. As void simulation models on a macroscopic level are not available, a simplified model to predict the laminate quality is necessary.

### 3.1 Node overfill algorithm

Short simulation times are desired for all applications of a simulation. As faster simulations allow to simulate larger models, the achieved precision of a simulation will increase. Likewise, the field of possible application is extended. For example, numerical optimizations require many simulation runs of the same problem with varying parameters. In general, optimization times may range between few hours and several days. Longer optimization times are generally not accepted by the users.
3.1 Node overfill algorithm

Extension of time step length  Usually the time step $\Delta t$ is calculated in a way, that the fill state of every control volume becomes 1 in maximum. Therefore, only very few control volumes (usually one) are filled in each time step, and the variation of the global pressure field and the flow velocities between two time steps is very small.

The basic idea of the overfill algorithm used in this work is to extend the time step $\Delta t$. As the effect of changed boundary conditions is very slight after one or in maximum a few additional control volumes are filled, it is possible to use the calculated pressure distribution and flow velocity field longer than the length of the time step calculated using equations 2.7 and 2.8. The effect is a significant reduction of the required number of iteration steps.

Sensibly extended time steps can be obtained by increasing the fill factor such that the fullest control volume reaches a fill value slightly above 1. Modifying equation 2.7, the required time to reach the fill degree $f_{\text{overfill}}$ in control volume $V_i$ can be determined by equation 3.1

$$ t_i = \frac{f_{\text{overfill}} - f_{\text{fill}}}{q_{\text{in}}} \cdot V_i $$

(3.1)

The extended time step $\Delta t$ can again be calculated using equation 2.8.

Good values for the overfill factor were found to be in the range of 1.1 to 1.4, a factor of 1.2 has been found to be adequate for most cases. Lower factors result in a lower effect of the overfill strategy, whereas higher overfill factors may result in significant resin loss due to excessive resin that cannot get distributed to neighbor nodes. Of course the optimum factor depends on the individual simulation problem.

Distribution algorithm for excessive fluid  As a consequence of overfilling, several control volumes reach a fill degree above 1. The excessive resin of the overfilled control volumes has to be distributed on the neighboring nodes in an appropriate way. In each iteration step excessive resin is distributed immediately after the update of the fill state has been done. An appropriate distribution strategy has been developed, considering two ideas:

1. The nearly filled neighboring control volumes should probably get filled next, prior to all other neighboring control volumes. So, partially filled control volumes are filled in favor, assuring a straight flow front. Therefore the control volume with the highest fill degree of all unfilled neighboring control volumes is chosen to be filled by the excessive resin first.
2. When the flow front advances steadily, the control volumes are filled uniformly. So, if the excessive resin could not be distributed to partially filled control volumes, the flow front is assumed to be smooth. Then the fill degree of all unfilled neighbor control volumes gets increased uniformly, conserving the overall resin volume.

Step 1 is repeated as long as excessive resin can be distributed successfully. Then, if excessive resin is left, it is distributed according to step 2.

In rare cases not all resin can get distributed to neighbor control volumes. For instance if straight flow fronts reach an edge of the cavity or if two flow fronts join in a low angle, all neighbor control volumes at the flow front are filled simultaneously. Than some excessive resin has to be discarded. This situation may result from too high overfill factors.

Results Using the presented node overfill algorithm was found to increase the simulation speed by a factor of up to \( \approx 10 \). The actual gain depends on the shape and width of the flow front. Distribution of excessive resin did not cause unphysical results if the overfill factor was chosen in the proposed range of 1.1 to 1.4.

3.2 Detection and tracking of gas entrapments

Dry spots resulting from gas (air) entrapments are a serious problem in RTM processes. Macroscopic gas entrapments may occur in unstable processes and have an unacceptably bad influence on the mechanical properties of the manufactured part. The concerned components require manual finishing or – in the worst case – result in scrap. Therefore it is important to detect gas entrapments in fill simulations used for process design.

Voids on micro and meso scale have been extensively investigated in the past in order to better understand their formation, transport mechanisms and their influence on the laminate quality. Among others, the formation of micro voids has been studied by Lundström et al. [76]. It was found that the distribution and size of voids could be explained by mechanical gas entrapment at the flow front and by the ideal gas law.

In this work, it is assumed that gas entrapments can be simulated using elementary physical laws. Mainly the conservation of the entrapped gas mass and the ideal gas law are applied. In addition to the flow front moving algorithm, an algorithm for tracking the evolution of the gas en-
3.2 Detection and tracking of gas entrapments

As the amount of required material data and simulation time increases, with the amount of physical details that are implemented in the simulation, a reasonable balance between these aspects is necessary. Capillary forces and fiber displacement behavior under resin pressure have therefore been neglected in this work.

For the verification of the developed simulation, non-moving as well as moving gas entrapments are considered.

3.2.1 Extensions in the simulation

In sLIP the flow front tracking is based on the volume of fluid algorithm [47]. To consider the formation of entrapments and their evolution during the fill process, methods to detect and track them were developed. The modeling and simulation of gas entrapments is based on following assumptions:

- Conservation of the entrapped gas mass; absorption and/or vaporization effects between the resin and the entrapped gas are not considered.

- The entrapped air behaves like an ideal gas and the ideal gas law is assumed to be valid.

- The viscosity of the entrapped gas is negligible compared to the viscosity of the resin. Thus the pressure within the gas entrapment is constant.

- Injection takes place at a constant temperature.

After an entrapment is detected as described in section 3.2.2, its volume can be calculated easily by summing up the control volume multiplied by the fill state of its member nodes.

\[
V = \sum_{k} v_k \cdot (1 - f_k)
\]

\( V \) : entrapment volume
\( k \) : member nodes of entrapment
\( v_k \) : control volume of node \( k \)
\( f_k \) : fill factor of node \( k \)

Each time a new entrapment appears, the enclosed gas mass can be calculated by the entrapment volume and the vent pressure which is 1 bar.
Concepts and tools for improved simulations

for RTM without evacuation of the mold. Because the entrapped gas mass is assumed to be constant, the pressure above vent pressure at all following time steps can be calculated as

\[ p = p_{\text{vent}} \cdot \frac{V_0}{V} - p_{\text{vent}} \]  

(3.3)

with

- \( p \): current pressure
- \( p_{\text{vent}} \): pressure at mould vent
- \( V_0 \): entrapment volume at appearance
- \( V \): current volume

This pressure is applied to all member nodes of the entrapment as a pressure boundary condition.

### 3.2.2 Detection of entrapments

In each time step the current entrapments have to be located. For this purpose a recursive detection algorithm for unfilled areas has been implemented (figure 3.1). The detection function starts at an unfilled node, marks it and marks all neighbor nodes which are not filled either. All marked nodes can thus be assigned to the current unfilled area. Unfilled areas which are directly connected to a vent are detected first, as they obviously must not be considered as entrapments. Therefore, the detection of the vent areas is started at each vent node. After the detection of the vent areas, all remaining unfilled areas can be identified as entrapments.

### 3.2.3 Tracking of entrapments

To calculate the current pressure within an entrapment, the mass of the entrapped gas needs to be known. The pressure has to be calculated in each time step as entrapments can move, join, split or dissolve. Therefore, the correlation between entrapment nodes during two consecutive time steps needs to be investigated. Because the location of the nodes is difficult to evaluate with respect to the membership to a certain entrapment, particularly if entrapments are close to each other, only the membership of nodes to entrapments in general is evaluated.

In the following the detection criteria for the five scenarios are explained (cmp. figure 3.2 and 3.3):

- Splitting of entrapment: two (or more) entrapments of the current time step consist of a high fraction (> 75%) of nodes that already belonged to an entrapment of the previous time step.
3.2 Detection and tracking of gas entrapments

Figure 3.1: Recursive detection of nodes in entrapment and at edge of entrapment

- Joining entrapments: two (or more) entrapments of the previous time step have a high fraction of nodes that belong to an entrapment of the current time step.

- Dissolving entrapment: none of the entrapments of the current time step has a high fraction of common nodes with this entrapment of the previous time step.

- New entrapment: none of the entrapments of the previous time step has a high fraction of common nodes with this entrapment of the current time step.

- Moving entrapment: this is the most frequently occurring scenario. An entrapment of the current time step has a high fraction of member nodes which have belonged to an entrapment of the previous time step. Unlike a splitting entrapment only one predecessor exists.

The described tracking strategy is illustrated by the following example. Five entrapments with a specified number of nodes are shown on the left side of figure 3.2. On the right side a set of five entrapments is shown for the next time step.

In order to track the evolution of the entrapments, the information is bundled in two matrices as shown in figure 3.3. The matrix RIN on
Figure 3.2: Entrapments and their member location at two successive time steps

![Figure 3.2](image)

Figure 3.3: Correlation matrices for entrapment tracking

<table>
<thead>
<tr>
<th>New Entrapments</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Entrapments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>45%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>50%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>75%</td>
<td>71%</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>60%</td>
</tr>
</tbody>
</table>

- RIN (reappear in new entrapment)
- AIO (appeared in old entrapment)

the left side ("reappear in new") describes how many nodes of the old entrapments reappear in the new entrapments, while the matrix AIO on the right side ("appeared in old") describes how many nodes of the new entrapments already belonged to an old entrapment.

Merging entrapments are characterized by columns with two or more high correlation values in the AIO-matrix, splitting entrapments by rows with two or more high correlation values in the RIN-matrix. Dissolving entrapments are characterized by rows without high correlation values in the AIO-matrix, new entrapments can be found by columns without high correlation values in the RIN- matrix. Finally, moving entrapments
3.2 Detection and tracking of gas entrapments

3.2.4 Resin transport

For correct simulation of the resin transport, the calculation of the net volume flow between the control volumes has to be modified, such that no resin can flow out of an empty control volume. This problem is exemplified in figure 3.4, considering an entrapment in a mesh of triangular elements (blue). Control volumes are highlighted in red. The green marked element is located at the edge of the entrapment in flow direction. On the nodes 1 and 2 a pressure boundary condition is applied because both nodes belong to the entrapment. Obviously, the pressure at node 3 will be lower than the pressure at nodes 1 and 2. The resulting pressure gradient would produce a flow from node 1 and 2 to node 3. If nodes 1 and 2 are actually empty, no flow into node 3 can occur.

Therefore, the calculation of the net flows into the control volumes have to consider the fill states. Otherwise, this artificial resin source would falsify the simulation. With respect to the simulation of the entrapment mobility, it would prevent node 3 from being emptied. Thus, the front edge of the entrapment would not move and entrapments would not expand when the surrounding pressure decreases.

Figure 3.4: Control volumes and element at the edge of an entrapment

belong to none of the previous cases and have a high correlation either in the RIN- or in the AIO-matrix.
In the last iteration steps, the remaining void covers a single node only, and is therefore not treated as an entrapment.

3.2.5 Simulation results

The simulation results shown in this section were obtained using 2D shell elements. Those were chosen for visualization reasons. However, the methods presented in the previous section are not restricted to 2D simulations and work for 3D models as well.

Moving entrapments Generally speaking, entrapments may expand or compress according to surrounding pressure changes and may move if they are located in an area with a pressure gradient.

Figure 3.5 shows a picture sequence of the simulated movement of a gas entrapment. The entrapment dissolves when it reaches the flow front. It can be seen that the entrapment is moving faster than the fluid. To understand this behavior, a 1D flow channel is considered. Without disturbances and without entrapments there is a uniform, linear pressure
distribution. This is illustrated by the resulting isobars in figure 3.6 (left). If an entrapment occurs as shown on the right side of figure 3.6, the pressure distribution is disturbed. High pressure gradients result on the left and on the right edge of the entrapment. (red marked areas). According to Darcy’s law these high pressure gradients lead to high flow velocities on the left and on the right hand side of the entrapment. The fast flowing resin in the front clears the way for the entrapment and the resin behind the entrapment fills the area behind it.

The setup used for the simulation shown in figure 3.5 has been reproduced in a glass-tool in order to validate the simulation results. The sequence of pictures shown in figure 3.7 shows the flow pattern observed through the glass lid. A 0°/90° woven fabric with 40 % fiber volume content was used.

Figure 3.7 shows that the entrapped air is separated into many small bubbles instead of being spilled out as a whole bubble. Figure 3.8 shows typical air bubbles traveling through the saturated areas of the laminate. These bubbles have a diameter up to 1 mm. The size may depend on the architecture of the used fabric, e.g. on roving diameter. This transport mechanism is well known in literature, e.g. [36].

**Non moving entrapments**  Entrapments that occur in corners or that cannot move due to other reasons are compressed or expand until the equilibrium pressure is reached. This mechanism can be shown by injecting a plate without vents as pictured in figure 3.9. The dashed lines show the equilibrium state for the two injection pressure values used for the simulation (\(\frac{1}{2}\) and 1). After the equilibrium pressure between injection pressure and entrapment pressure is reached the pressure at the injection
Figure 3.7: Entrapment evolution in a $0/90^\circ$ fabric
gate is reduced to $p_2 = \frac{1}{2}$. The entrapment expands again and after a long enough time, the new pressure equilibrium at $p = \frac{2}{3}$ is reached.

However, as shown in figure 3.9, the straight flow front is destroyed and the resin takes the way of the least resistance. The size of the fingers depends on the mesh density, as the minimal width of a finger is determined by the width of the control volumes. 16 elements in width were used in the simulation shown in figure 3.9 (upper picture), whereas 30 elements in width were used in the simulation shown in the lower picture of figure 3.9. Once a finger is formed the pressure gradient at its top towards the vent is higher then everywhere else. Thus, the finger will grow. In the presented simulations, the fingers were initialized by the overfill algorithm used, and did not occur when the overfill algorithm was turned off. It can be assumed that expanding entrapments in real injections would not form finger, but lose entrapped gas by releasing small bubbles.

### 3.2.6 Conclusion

Entrapments in RTM can be simulated based on elementary physical assumptions, mainly the conservation of the entrapped gas mass and the ideal gas law. The mobility of entrapments is implemented based on elementary rules. In this context the edge of an entrapment is considered to be a flow front and is simulated applying the volume of fluid algorithm [47].

Experimental results show that the developed algorithm simulates gas entrapment mechanism correctly with respect to the sizes and pressures. For practical applications many entrapments don’t move at all, but stay
Figure 3.9: Compressed and re-expanding gas entrapment. Upper pictures with 16 elements in width, lower pictures with 30 elements in width.)
at one location, e.g. they may be trapped in a corner. Those entrapments are detected and simulated correctly.

Nevertheless, experiments also showed, that the simplification, neglecting surface forces and the heterogeneous micro structure of the preform, lead to an incorrect simulation of moving gas entrapments. Moving gas entrapments in the simulation remain almost compact, large entrapments may split up, but the size of moving gas bubbles was found to depend on the mesh size. The smaller the mesh size, the smaller resin fingers are formed, and the smaller bubbles are observed.

In contrast, gas entrapments in experiments do not move as a compact volume. As shown in figure 3.7, entrapments vanish by releasing many small bubbles (≈0.1 mm diameter). The resulting small gas bubbles traveling through the resin are likely to create zones with increased matrix porosity along their flow path, causing reduced laminate quality.

### 3.3 Modeling the injection system

The injection system is an essential part of the simulation model. Most multi-gate injections are based on a resin distribution system, distributing resin from one or a few injection devices to several injection gates in the mold. To realize sequential injection strategies valves are introduced in the distribution system. These valves are opened when the flow front reaches certain locations.

In general, the resin volume flow in the branching tubes behind a flow controlled injection device will differ between the tubes and will not be half of the total flow. The flow may vary with time as well, as the flow resistance at the gate changes due to the progression of the flow front or due to curing of the injected resin.

In conventional simulations, vents and gates are modeled by boundary conditions by setting the node pressure to 0. Gates can be prescribed either as pressures or flows (section 2.1.3). Useful configurations can easily become complex and difficult to model in the conventional method. To simulate sequential injection strategies advanced simulation models, which introduce abstractions, are required. A simple and therefore more convenient approach is to copy the real setup with tubes, valves and injection devices to the simulation model.

#### 3.3.1 Modeling approach

Figure 3.10 illustrates the proposed connection algorithm. Each device of the injection system has a connector at its resin outlet. To this connec-
tor, an arbitrary number of tubes and valves can be connected. Active injection devices such as pumps and pressure pots do not need a superior device, and are therefore placed on the top row. Valves need a superior device for resin supply. This may be an active injection device or an already connected valve. The final connection step is the connection of gates to one of the available connectors at the resin outlets of the injection devices.

3.3.2 Components of the injection system

**Tubes**  Tubes are the simplest devices of the injection system. They connect two nodes which results in a finite element with a 2x2 conductivity matrix. Their actual conductivity can be calculated by Poiseuille’s law, assuming a circular cross section of the tube and a laminar flow.

\[
K_{\text{tube}} = \frac{d^2}{32}
\]  \hspace{1cm} (3.4)

Simulation results show that the flow resistance within the tubes is significantly lower than the other flow resistances.

**Valves**  Valves are similar to tubes, but in contrast to tubes valves can be opened and closed at arbitrary times. In sLIP, valves are implemented inheriting all properties from tubes, and additionally their conductivity
matrix can be set to 0 when they are closed. Valves are closed in the begin-
ning of a simulation and can open at a specified time, allowing sequential
injections.

**Active injection devices**  Active injection devices provide a resin flow
at a controlled pressure or volume flow. Four variants of active devices
are implemented in sLIP

- Constant pressure device
- Constant volume flow device
- Time dependent pressure device, following a linear ramp
- Time dependent volume flow device, following a linear ramp

**Post pressure device**  The post pressure device provides resin at a con-
stant pressure. It is similar to the device for constant injection pressure,
but it is turned on at the end of the injection phase, when all other devices
are turned off. Applying post pressure after the end of the injection in
the beginning of the curing cycle is an important factor for an improved
matrix quality. Eventual gas entrapments get compressed and the size of
voids can be reduced significantly. The matrix strength is improved [97].
Therefore, a specialized post pressure device has been implemented, pro-
viding resin pressure for the compaction of gas entrapments after all vent
are closed.

### 3.3.3 Connection algorithm

Each injection device (incl. valves) posses one connection node to which
valves and gates can be linked. Valves additionally have a source node
which has to be connected to a parent device.

1. Injection valves are connected first. The first valve can only get
connected to an active device (but not to a post pressure device).
The following valves can either be connected to one of the active
devices (but not to a post pressure pot) or to an already connected
valve. All valves get connected, but no circular connections are
possible.

2. Each gate on the component is connected to one connection node. It
can be a connection node of an active injection device or of a valve.
3. The post pressure device is connected to a single gate directly on the component.

### 3.3.4 Simulation results

Advanced injection strategies such as sequential injections can be simulated with low modeling effort. Optimization results can be directly implemented in the real processing, and a minimal number of free optimization parameters allow fast improvements in process optimizations.

Figure 3.11 shows the simulation model for the sequential injection of a hockey stick. One injection device providing a constant resin flow and one valve have been used. Additionally the locations of the gates have been defined, and two vents were located at the ends of the stick. Figure 3.12 shows the fill state of the stick and the flow velocities in the injection system at four times. In the first picture on the left, the valve is closed and all resin is injected in the middle gate. In pictures 2, the valve is open and the resin splits up, flowing through the upper and the lower gate. The resin flow through the middle gate has stopped. A similar situation can be seen in the third picture. In picture four, the upper vent has closed, and all resin flows through the lower gate.

### 3.3.5 Conclusion

A simple modeling of the injection system has been achieved. Five types of injection devices for pressure or flow controlled resin supply have been implemented in sLIP, as well as a valve which opens at a specified time. Devices with different capabilities can easily be added. Simulation results show the working simulation of resin flows within the injection system.

For the connection of the injection system, a “simple to use” algorithm has been developed. Circular connections of injection devices are avoided. With regard to process optimization, a parametrization with the lowest possible number of optimization parameters has been achieved, which can still describe all reasonable and useful configurations. Different available injection devices and their specific capabilities can be modeled and their configuration is left to the optimizations.

### 3.4 Flow marking particles

Although detailed flow information can be obtained from basic simulations of RTM injection processes (section 2.1), spatial information depending on the resin history is not available. The local flow velocities in the mold
Figure 3.11: Example for simulation model including the injection system.

Figure 3.12: Simulation results for model shown in figure 3.11. Hybrid illustration of flow velocity in the injection system and fill state in the part. Blue indicates low flow velocities in the injection system, green medium velocities and red indicates high velocities.
are known, but the actual moving of the resin particles through the cavity is not represented in standard simulations. Therefore, the tracking of resin particles becomes interesting in order to generate additional process information. Using flow markers, which move with the resin, allow to record the history of the resin particles. These markers may contain qualified information, as for instance the curing degree of the resin particle represented by this marker. In this way, local information that depend on the fill history and transportation can be generated. For the simulation of resin curing this may for example be the actual local resin viscosity and the degree of cure. Another possible application of flow markers is the monitoring of flow front zones, where reduced laminate quality has to be expected (section 3.6). Basics of marker simulations are described in this chapter.

### 3.4.1 Marker movement

**Basic movement principles** Markers are temporarily assigned to a host element and move within that host element. The marker’s flow velocity is given by the calculated resin velocity within the host element (figure 3.13). After a certain flight time the marker will reach an element boundary and must be forwarded to a neighboring element. In most cases there will be only one neighbor connected, which will be the next host element of the marker. Thus, in each step a clear assignment of the marker to an element is given and therefore the required flow velocity for the marker movement can be provided.

Due to the strict assignment to the element surface (2D) or volume (3D) the marker cannot leave the part in bended areas of the part. Inad-
3.4 Flow marking particles

Figure 3.14: Schematic illustration of marker leaving the element planes if not assigned to host element (left). Erroneous transition of marker to neighbor surface is possible (right)

Figure 3.15: Movement on a component edge

Possible transition of markers between neighboring surfaces that are close to each other, as illustrated in figure 3.14, is prevented.

Markers at component surfaces, component edges and branching points Several problematic cases can occur during the movement of markers, for which the basic principles need to be extended. Due to the domain discretization of the cavity with finite elements the calculated velocity field is non-continuous at the element edges. For instance the flow velocity within linear elements is constant. So the flow velocity in an edge element is not necessarily parallel to the edge but has a velocity component perpendicular to the edge in general (figure 3.15). If the marker cannot be given to a neighbor element\(^1\) and because the marker needs a host element, the marker is moved on the element interface (surface in 3D, edge in 2D) with the velocity component parallel to the edge (surface or edge).

\(^1\)Detailed description of the generation of neighborship information and the used objects for storing can be found in the appendix on page 236
A similar problem occurs if two neighbor elements have a contrarily directed flow velocity and the marker reaches the interface between these two elements (figure 3.16). If the marker would be given to the neighbor element it would reach the common interface after a flight time of zero and would be returned to the previous element. This could lead to an infinite loop which can be avoided for instance by either moving with an averaged flow velocity or by moving on the interface until the end of the interface is reached or the element’s flow velocities change in a following time step.

Slightly different are the branching problems illustrated in figures 3.17 and 3.18. Several neighboring elements may be connected to an edge, interface or node, being a potential new host element. This problem will often occur after a move on an interface when the interface edge is reached. If the flow within the possible new host elements spreads in two directions as illustrated in figure 3.17, the markers need to flow in both directions as well. Of course more than one marker is required for this. A possible solution might be a deterministic approach considering the number of markers moved to one direction and the number of markers moved to the other direction. But this approach becomes difficult if the flow pattern
changes over time, for instance if the flow in one direction is stopped after filling the corresponding area of the cavity. To avoid this complicated deterministic approach a stochastically controlled branching algorithm was chosen. The flow marker is handed over to a randomly chosen element connected to the branching point (node or element interface). Thus a good distribution of the markers is achieved.

If the marker is given to an element with a resin velocity towards the current interface, a new branching decision is made and a new host element is selected randomly. In some cases an infinite loop might occur when all flow velocities of the connected elements point into the direction of the current interface (variant of problem illustrated in figure 3.16). Then the flight time within all connected elements is calculated as zero, and the algorithm tries to move the marker on the element interface.

Details about marker movement procedures and generation of neighborhood information for elements can be found in the Appendix A.

3.4.2 Simulation costs and marker precision

Simulation of flow markers requires an additional effort for marker tracking. Required data structures containing neighborhood information must be generated, and the movement of the markers themselves require time. However, simulation costs do not exceed the simulation time of the overall calculation costs for the fill simulation. Depending on the number of markers, additional simulation time up to 30%...50% is required for the movement of the markers. The required number of markers depends on their purpose, for flow front tracking approx. one marker per node is used (figure 3.19), for the modeling the resin cure a lower number of markers will be sufficient.
Figure 3.19: Number of flow front marker ($\#marker \approx \#nodes$)

Figure 3.20 shows the effect of the simulation of flow front markers on the simulation time for different model sizes. The green line shows the simulation time for a simulation without flow front markers. The blue line shows the total simulation time for a simulation with flow front markers, and the red line represents the time fraction that is needed for the movement of the markers in that case. The difference between simulation times with and without markers is obviously higher than the time required for the moving of the markers. This can be explained by the need of additional information required for marker simulation. For instance in simulations without markers, the flow velocity has to be calculated at the flow front only. However markers can be everywhere within the filled area, so the local element velocity is needed in all filled areas, which is very time consuming.

The precision of the markers is assessed by testing flow front markers. Flow front markers are inserted at the current flow front when a control volume is filled. As the markers move with the resin, they always have to remain at the flow front. In case of flow front joints or if the flow front is stopped at a cavity wall, the markers have to remain at those locations. Figure 3.21 shows the locations of flow front markers for a test geometry. Resin is injected in a line gate at the left hand side in the thicker region. In the beginning the flow front moves to the right and is stopped by the right cavity wall. Most markers are located at the flow front, while the other markers have been stopped by the edges of the cut outs in the test geometry. Obviously, the markers move with the flow front as it was expected, proving their applicability for the tracking of resin flows.
3.4.3 Conclusions

The described markers provide a general method for generation and processing of information requiring the history of the resin flow. Examples for their application are given in the following sections 3.5 and 3.6.

The developed flow markers follow the flow very precisely as shown in section 3.4.2. The stochastic branching algorithm leads to a good marker distribution in the part. An important issue is to identify a suitable time and location to insert markers, depending on the purpose of the particular markers.

However due to the complex moving algorithm and the complex generation of the neighborhood information, the simulation time of the injection process is slightly longer ($\approx 30\%$) depending on the number of used markers. Unavoidable numerical errors in the calculation of intersections may lead to problems in the implementation. An additional effort is required to compensate these errors and distinguish between intersections with an interface on the one hand and markers being close to the concerned interface on the other hand.

3.5 Resin curing and viscosity

For fast processes with fast curing resin systems, the curing of the resin may have a significant influence on the fill behavior. Fresh uncured resin is injected at the gates, but it cures as it moves through the mold. Therefore, the resin inside the mold has a spatially varying degree of cure and a varying viscosity.
Figure 3.21: Flow front marker during simulation.
Injections may be accelerated using a sequential injection strategy, where some injection gates are opened when the flow front passes them. Thus, the flow length of the resin is reduced and uncured resin with a low viscosity is injected at the flow front. To model this strategy, it is necessary to include the effect of fresh resin at sequentially opened gates in the fill simulation.

To determine the distribution of the local viscosities, resin particles have to be tracked during the filling of the cavity. Flow Markers to track resin particles during the injection as introduced in section 3.4 are able to deliver information about the local resin properties of the resin they represent. A curing and viscosity model is required, and the curing state of the particle has to be updated in each time step of the simulation, depending on the current temperature and the previous degree of cure. Modified Stolin and Arrhenius models [46] are used to calculate the resin viscosity and the curing rate.

In order to obtain a good distribution of the resin markers within the filled area, resin markers are inserted in appropriately chosen time intervals at the injection devices or at the gates. A reasonably low number of markers can be used if they are well distributed in the filled domain. An additional algorithm is required to assign a reasonable surrounding influence area to each marker, in which the viscosity is defined according to the marker data. Thus, each finite element of the simulation model can be provided with updated information about the current viscosity of the resin in its area.

### 3.5.1 Curing model

To model the curing kinetics of the resin a model proposed by Henne [46] and described by equation 2.9 on page 17 is used.

Within the CTI supported project VANTEX (“Vanishing Textile binder structure for the stabilization of dry carbon fiber preforms in the Resin Transfer Molding process.” CTI-No. 8634.1) a thermoplastic phenoxy thread has been investigated, which can be solved in epoxy resin. On the one hand the fiber ondulation due to the sewing thread can be reduced and thus the mechanical strength can be improved. On the other hand the solved phenoxy modifies the matrix properties yielding a higher impact strength.

An extension of the curing model to include the influence of solved phenoxy additives has been made (“Phenoxy term”). There, a linear influence of phenoxy on the curing rate and on the viscosity is assumed, corresponding to a tailor expansion of first order [113].
Table 3.1: Parameters of the curing model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Scaling factor for reaction speed</td>
</tr>
<tr>
<td>$T_{\text{solid}}$</td>
<td>Solidification temperature of the resin</td>
</tr>
<tr>
<td>$T_{\text{kin}}$</td>
<td>Reference temperature for reaction speed</td>
</tr>
<tr>
<td>$m$</td>
<td>Exponent for reactive links</td>
</tr>
<tr>
<td>$n$</td>
<td>Exponent for auto catalytic reaction</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Factor for description of maximum degree of cure</td>
</tr>
<tr>
<td>$l_{\text{cure}}$</td>
<td>Phenoxy influence on curing kinetics</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>Pre-cure degree, when resin is injected into the mold</td>
</tr>
</tbody>
</table>

\[
\frac{d\alpha}{dt}(T, \alpha, c_{\text{ph}}) = \underbrace{A \cdot (1 + l_{\text{cure}} \cdot c_{\text{ph}})}_{\text{Phenoxy term}} \cdot \underbrace{e^{\left(-\frac{T_{\text{kin}}}{T_{\text{solid}}}\right)}}_{\text{Temperature dependancy}} \cdot \underbrace{(\alpha_{\text{max}}(T) - \alpha)^m}_{\text{Number of reactive links}} \cdot \underbrace{\alpha^n}_{\text{Auto-catalytic term}}.
\]  

$c_{\text{ph}}$ is the phenoxy concentration in the resin (weight ratio), and $\alpha$ is the resin’s current degree of cure. Table 3.1 summarizes all required model parameters.

Calculating the degree of cure  Given the initial curing degree $\alpha_0$, the degree of cure at an arbitrary time $\alpha(t)$ can be calculated using a Runge-Kutta integration. Figure 3.22 shows some integration results. Very similar results for both time steps 50 s and 200 s have been calculated, proving a good approximation and a low sensitivity on the time step length. For comparison, results of Euler integrations are plotted as well, showing a strong influence of the time step. Both integration schemes, Runge-Kutta and Euler, converge to the same solution for integrations with short time steps.

A major influence is the appropriate assumption of the initial degree of cure at the beginning of the cure modeling. This pre-cure degree $\alpha_0$ causes
3.5 Resin curing and viscosity

Figure 3.22: Runge-Kutta integration of degree of cure compared to Euler integration for two different integration step lengths.

A significant shift of the calculated degree of cure $\alpha(t)$ on the time axis. It may vary for each injection because it depends on storing conditions and time, as well as on the treatment during the preparation of the injection. Thus, an averaged value has to be used. It can be determined by fitting calculated curing values $\alpha(t)$ to measured values of isothermal Differential Scanning Calorimetry (DSC) measurements.

3.5.2 Viscosity model

The viscosity model of [46] described in equation 2.10 has been slightly modified and extended by the parameter $f_{\text{stretch}}$ in order to receive a better adaption of the measured values to the viscosity model. Thus, the slope of viscosity with increasing degree of cure, $\eta(\alpha)$, on one side and the spreading of the viscosity due to the temperature influence on the other side can be adjusted separately. Analog to the curing model (eq. 3.5), a linear term to account for solved phenoxy additives is added.

$$\eta(T, \alpha, c_{\text{ph}}) = \eta_\infty \cdot \left(1 + l_{\text{rheo}} \cdot c_{\text{ph}}\right) \cdot e\left(\frac{T_{\text{rheo}}}{T_{\text{solid}}} + \alpha \cdot \left(\frac{T - T_{\text{solid}}}{T_{\text{solid}}}\right)^{f_{\text{stretch}}}ight)$$

(3.6)

Table 3.1 summarizes the model parameters of the viscosity model.

A principal limitation of the viscosity model arises as the maximum degree of cure is $\alpha = 1$, and therefore the maximum viscosity $\eta(\alpha = 1)$
Table 3.2: Parameters of the viscosity model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>Curing influence on viscosity</td>
</tr>
<tr>
<td>$T_{\text{solid}}$</td>
<td>Solidification temperature of the resin</td>
</tr>
<tr>
<td>$\eta_\infty$</td>
<td>Reference viscosity (uncured, high temperature)</td>
</tr>
<tr>
<td>$T_{\text{rheo}}$</td>
<td>Rheological reference temperature</td>
</tr>
<tr>
<td>$f_{\text{stretch}}$</td>
<td>Slope of viscosity with degree of cure</td>
</tr>
<tr>
<td>$l_{\text{rheo}}$</td>
<td>Phenolx influence on viscosity</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>Pre-cure degree when resin is injected into the mold</td>
</tr>
</tbody>
</table>

calculated with eq. 3.6 is finite. However, for practical applications this limitation does not start to influence the viscosity before the viscosity is far above the range of interest for the injection.

3.5.3 Local resin viscosity from flow markers

Assigning marker influence area A simple approach to define the influence area of each marker is to calculate the distance between markers and elements (resp. center of gravity). Thus, the closest marker to each element can be found. However, this algorithm is a very time consuming calculation and wrong markers will be chosen in some cases. As illustrated in figure 3.23, marker M4 would be assigned to the gray marked area on the lower part of the schematic component which should obviously be assigned to the markers M1 or M2.

A faster and more reliable algorithm has been developed. As it is illustrated in figure 3.24, the influence area is determined in an iterative neighborhood search. Starting at each marker, the marker is assigned to its host element. In the next iteration step, the marker influence is expanded all direct neighbor element which are not yet assigned to a resin marker. This is done for all markers, before the next iteration step is started. The assignment is repeated until all elements are assigned to a marker (cmp. algorithm 5, page 255 in the appendix).

Thus, all elements get assigned to a close located marker. If the local element sizes differ, the influence area of markers in an area with small elements may be small. However, in general the local element size will be of low influence.
Figure 3.23: Marker influence area avoiding inadequate assignment by distance

Figure 3.24: Influence area of resin markers by neighborhood search
Uniform marker distribution within the filled domain  In simulations modeling the injection system as described in section 3.3, the markers will be inserted into the mold at the nodes that belong to the device. Each injection device stores the iteration step or time at which the last markers have been inserted. As injection devices are modeled with one single node, a poor marker coverage of the part would result if only one marker is inserted at a node. Therefore, several markers are inserted at each node. At branching points within the injection system and at connectors on the part, the flow markers will flow in different directions due to the stochastic branching algorithm (section 3.4.1).

If a traditional modeling of injection points is used, these marker related functionalities are a subject of the gates. For a good spatial marker distribution, markers are inserted at each node that is linked to the gate.

The number of markers that are inserted into the cavity has to be reasonable low because their moving algorithm becomes costly with respect to calculation time for very high numbers of markers. On the other side, too few markers will give an imprecise modeling of the viscosity distribution and a coarse spatial discretization of the viscosity. Therefore two inserting strategies have been developed

1. Insert by iteration steps
   Inserting resin markers by iteration steps will lead to a rather uniform distribution of markers within the cavity and the total number of markers can be controlled very well. If the approximate filling time is unknown this strategy should be chosen.

2. Insert by filling time
   Resin curing is mainly controlled by time. So the markers should be inserted according to the gel time $t_{gel}$ of the resin they represent. The gel time is approximately the processible time of the resin. Thus an optimum discretization of the viscosity can be achieved. Approximately 20 to 50 markers should be inserted within the processible time of the resin. Problems may occur if the filling time is unknown, e.g. during early stages of the process development. For long filling times, too many markers might be inserted increasing simulation time significantly.

Simulation example  Two fill simulations for a paddle were carried out. Because RTM6 is a very slow curing resin the injection temperature has been increased to 160°C to accelerate the curing and show the curing effects in the simulation. Good spatial marker distribution and local vis-
3.5 Resin curing and viscosity

Curing and viscosity information were achieved. This is shown in figure 3.25 for single gate injection and in figure 3.26 for a sequential injection with two gates.

The first simulation was done with a single injection gate at the hand end of the paddle. The upper picture in figure 3.25 shows the final fill state, when the resin flow has stopped due to very high resin viscosity. The middle picture shows the distribution of the resin markers that are used to track the resin curing state, and the lower picture shows the corresponding viscosity distribution within the paddle.

The second simulation was done using a sequential injection with an additional gate at the junction of the tubular handling bar and the actual paddle. Using this injection strategy the component can be filled as shown in figure 3.26.

3.5.4 Conclusions

Flow markers can be used in numerical fill simulations to receive more realistic simulations of RTM injection processes. Slowing down of the resin flow and in the worst case the stopping of the resin flow due to cured resin can be modeled. Particularly sequential injection strategies that are used to refresh the resin at the flow front can benefit from the included curing models. On the other side simulation time will rise. The simulation will slow down as the movement of the resin markers has to be calculated and in each time step the curing state of each marker has to be updated as well as the local resin viscosities.

All model parameters can be calculated using a least squares fit of experimentally determined curing and viscosity data. A suggestion for the experimental campaign can be found in the appendix B.1. Additionally, a procedure to evaluate the measured data and evaluated parameters for the RTM6 resin system (B.2), as well as plots of the model accuracy for the “RTM6” resin system (B.3) can be found there.

For both, the rheological and the kinetic model, a good agreement between measured values on the one hand and values predicted by the model on the other hand has been achieved. An uncertainty in the used curing model is the pre-curing degree $\alpha_0$, which shifts the curing curve on the time scale. It may differ for each injection because it depends on the actual resin treatment prior to the injection.
Figure 3.25: Single gate injection of a paddle. Simulated with RTM6 resin under isothermal conditions at 160°C, the flow front stopped due to cured resin. The pictures show (from top to bottom):
- the final fill state at the end of the simulation
- the final distribution of resin markers
- and the final resin viscosity
3.5 Resin curing and viscosity

Figure 3.26: Simulation of successful sequential injection of the paddle, RTM6 resin under isothermal conditions at 160°C. The pictures show (from top to bottom):
- the final fill state at the end of the simulation
- the final distribution of resin markers
- and the final resin viscosity
3.6 Prediction of matrix quality using void marker

In RTM processes voids in the matrix occur with typical sizes in the range of several micrometers up to \( \approx 0.1 \) mm. Those voids cause matrix porosity and have a strong impact on mechanical transverse and shear properties. Beside appropriate processing conditions such as evacuating the resin and the cavity, rinsing the filled mold and applying post pressure, well suited gate locations and injection pressures contribute to a minimum void content.

Therefore, the simulation of laminate quality is relevant for process design with respect to optimum part quality. Although a lot of work has been done on the simulation of voids, a standard model is not yet established. Thus, a simplified simulation model shall be defined here. This model should account for void formation mechanisms as well as for void movement.

3.6.1 Void formation

A lot of research has been done on void behavior in the recent years with the objective to predict their spatial distribution within the part. The major difficulty is the dual-scale nature of the resin flow in RTM processes. Micro voids within the fiber bundles and macro voids between the fiber bundles are distinguished. There is a difference in scale of a factor of \( \approx 1000 \) between micro and macro scale, so micro scale simulations examine one or a few unit cells in maximum and simulations of whole parts cannot consider micro scale effects, respectively. While the flow between the fiber bundles is driven by the injection pressure, the flow within the bundles is additionally driven by capillary forces.

Void formation mechanisms received high attention in the reviewed literature. There is a large agreement about mechanical gas entrapments at the flow front as the major mechanism for void formation [43, 54, 76]. The behavior of entrapped gas with respect to its compressibility can be modeled using the ideal gas law [76]. Other sources for entrapments are subject to chemical processes as for example degassing. The entrapped void volume can be approximated using appropriate models, e.g. depending on the flow velocity at the flow front \( v_{FF} \) [95, 96]. As low flow front velocities favor the formation of macro voids between fiber bundles and high flow front velocities favor the formation of micro voids inside the fiber bundles, the existence of an optimum flow front velocity can be assumed [112].
3.6 Prediction of matrix quality using void marker

3.6.2 Mobility of voids

A further topic of high importance is the movement of voids, including conditions for release or trapping of gas bubbles. Experimental investigations show voids remaining at their initial location as well as voids traveling with a velocity several times higher than the velocity of the resin.

Again, simulation is difficult due to the dual scale nature of the problem and a lot of material data is required. Capillary numbers, surface tension and contact angles between resin and fibers, geometric information about the fabric bundle and gap structure might be required, all information in its spatial distribution.

Once gas is entrapped it may remain at the same location or it may escape from the resin again, if the mobility of the voids is sufficiently high. Their mobility depends on surface forces between fibers, resin and air. Also, void mobility depends on pressure gradients and void size, sizes of the flow channels and therefore on fiber volume content, fiber orientation relative to the flow direction and the location of the initial void. Multilayer preforms with varying fiber orientations make the problem even more complex, because the optimum flow front velocity depends on the flow direction relative to the fiber structure.

In [43] a critical release pressure $p_c$ is defined, which controls the release of voids. Unfortunately this release pressure $p_c$ is theoretically not well-founded and might depend on several parameters.

On a micro scale an interesting investigation on void deformation, movement and breakup is presented in [55]. Bubble movement was observed in a macroscopic cylinder field representing a fiber bundle. It was shown that voids of critical size can break up into several smaller voids while traveling through the fiber bundles. In [105] void transport is understood to be subject to retardation due to narrowing flow channels where fiber structure changes (bottleneck). A pressure difference is required to split up or deform the void against surface tensions, to let it move through the fabric structure. Frishfields [36] follows a similar idea of trapped bubbles in narrowing channels. Using flow channel networks and probabilistic models in combination with criteria on bubble deformation, movement of voids could be simulated, achieving qualitative agreement with experimental data.

While large voids tend to remain at their initial location, medium and small voids migrate to the exit, leading to an increased void content there [44]. Due to the accumulation of entrapped gas at the flow front, avoiding flow front confluences in highly stressed areas contributes to bet-
ter part quality as well. As joining flow fronts stop at confluence lines, areas of high void content are caused. Experimental investigations clearly show the effect of flow front confluences on the spatial distribution of the void content [8, 97].

3.6.3 Simplified simulation model for moving voids

However, in the literature reviewed no comprehensive and feasible solution to simulate and predict the void content during RTM fill simulations on component scale could be found [98]. Except form [36], simulation models for migrating and vanishing voids, washed out with resin flow on component scale could not be found. For all models detailed information about fabric architecture is required.

To get a rough approximation of moving voids, particularly of voids accumulating at flow front confluences, the simplified approach of void markers has been developed. Void markers are used to map the movement of voids and to track the movement of flow front resin. Therefore markers are inserted each time a nodal control volume gets filled. The gas volume entrapped at this node is stored in the marker. Thus moving voids can be simulated. At each time step the laminate quality can be calculated summing up all void volumes of the void markers. For the global quality criterion void markers at the edges of the part are not considered in the quality criterion, because those markers do not indicate confluence ares.

The entrapped volume can be calculated using appropriate models, e.g. depending on the flow velocity at the flow front $v_{FF}$ [95, 96]. Instead of inserting one single marker at each filled control volume, several markers can be used to represent micro- and macro-voids. Markers representing different void bubble sizes and different traveling velocities can also get different mobility properties.

However these models require additional data like surface angles and flow channel dimensions. To simplify the simulation and the model setup all simulations in this work were done assuming a constant fraction of the nodes control volume to be void volume. Furthermore all void markers move with the local resin velocity, although voids may move with different velocity. Thus, approximate information about relative void content is generated, allowing to compare the quality of different processing parameters.

Figure 3.27 shows an example of the simulation results for void markers inserted and flowing with the flow front. Resin is injected in a line gate at the left side in the thicker region. The graphics show the propagation of void markers with time. The final state shows remaining void markers
at the top-right corner of the thicker region, indicating reduced matrix quality that has to be expected in that area.

3.6.4 Conclusions

A strongly simplified simulation model has been developed to simulate the void content of the matrix. The model assumes mechanical void trapping at the flow front as the only formation mechanism, and existing void travel with the resin viscosity. Both assumptions could be extended, for instance the size of newly formed voids can be depending on the flow front velocity. However, additional material data would become necessary for realistic results.

3.7 Conclusions for simulation concepts and tools

The presented methods to automatically detect gas entrapments and to track their movement can be used to realistically determine the final fill degree of a part. Entrapped gas volumes are compressed according to the surrounding resin pressures and therefore show realistic sizes of dry spots. To further improve the simulation of gas entrapments, a refined physical model should be included. The physical model should also account for the heterogeneous fiber structure of the fabric, and thus will realistically simulate moving and dissolving entrapments. Eventually, flow markers could be used to represent traveling small gas volumes, resulting in voids.

A further development was to include the injection system, consisting of injection devices, valves and tubes, in the simulation model. Thus, applying the boundary conditions could be simplified, as compared to directly setting boundary conditions on the actual model. If volume flow controlled injection devices are used, the volume flows in branching tubes may particularly differ with time, which can be easily modeled in the suggested way.

Realistic prediction of the final laminate quality of a part is still an open issue, although much research has been done on this topic. Macroscopic void prediction models usable on a part scale level are not yet available. The void marker concept suggested in this work is a simplified quality prediction approach tracking the flow front resin, where an increased void ratio has to be expected. Thus, the required amount of material data is drastically reduced. To obtain an improved void prediction model, the void marker concept can be combined with existing
Figure 3.27: Void markers during fill simulation. The upper pictures show the flow front markers, the lower pictures show the corresponding fill states.
void formation models, and be extended by appropriate rules for the void traveling velocity.
Chapter 4

Process variations and reliability

Variations within the injection process are an immanent property of the resin transfer molding. A parametric simulation model to account for these variations is presented in this chapter. Derived from mold geometry and the preform handling, locations with a high probability permeability of variations can be identified, and the range of these variations can be physically founded.

This parametric model is used in Monte Carlo simulations to estimate the process robustness against variations in preform layup and processing parameters. To quantify the process robustness, a determination method for the scrap rate with respect to the minimum required part quality has been developed.

4.1 Parametric simulation model

In this section, the disturbance cell concept is presented, which allows to map the permeability distribution of the real injection situation to a simulation model. Using this concept, a parametric model is obtained, which has a sufficient variability of the permeability to get adapted to the actual permeability distributions within the mold. The required input is limited to only a few parameters.

Several sources of disturbances reported in literature were summarized in section 2.1.6. Figure 4.1 shows an example of locations and areas with high probabilities of disturbances for a typical RTM part. Probable flow
channel locations (red) are edges and joint lines. Flow channels may also occur around the inserts. The plane surfaces may be disturbed with respect to the permeability and the fiber volume content. The blue lines bound possible sections of the component with uniform disturbances.

Including these disturbances in the simulation model will allow more realistic simulation results.

### 4.1.1 Disturbance cell concept

Sections are defined where a uniform influence of a disturbance can be expected, influencing all finite elements inside this area. Each disturbance cell is linked to several finite elements of the simulation model (figure 4.2). All member elements of a disturbance cell are changed uniformly, according to the disturbance type and values of the cell. The disturbances include variations of the fiber volume content and the permeability values within predefined ranges as well as their statistical distribution.

Two or more disturbance cells may overlap, having common member elements in the simulation model. Therefore the disturbance of the concerned elements can increase or get compensated. An influence factor $\eta$ is introduced to achieve a smooth cell influence on the model (fig. 4.3). Within the core zone the cell’s influence is $\eta = 1$. In the transition zone from the cell’s core to its outer edge, the influence factor decreases to 0.

Disturbance cells can be adapted to different characteristics. Therefore location, size and influence shape must be selected suitably. For distur-
4.1 Parametric simulation model

Figure 4.2: Finite element mesh and disturbance cells

Example for $\Delta K +40\%$
$\varphi +30^\circ$

Core zone: $(\eta = 1)$

Transition zone: $(\eta = 0.5)$

Influence on permeability tensor

Figure 4.3: Influence of disturbance cells on permeability tensor
Figure 4.4: Initialization of disturbance cell with normal distributed permeability

Disturbances such as flow channels near cavity edges, stretched disturbance cells can be applied along the locations with high risk of flow channels. Depending on the modeling method for flow channels, the local fiber volume content is influenced or a permeability supplement $K_{\text{plus}}$ is applied onto the finite elements linked to the flow channel (section 4.1.3). Errors like handling induced shearing of the fabrics require disturbance cells that are spread over a larger part of the injection domain. Those will influence the permeability values and the principal directions as well as the fiber volume content.

For realistic simulation models, information about the statistical distribution and the ranges of disturbances needs to be given to the disturbance cells. Distribution functions can differ for different types of disturbances. E.g. the normal distribution can be used for the fiber volume content and the log-normal distribution can be used for the principal permeability values. An initialization number $s_{\text{init}} \in [0, 1]$ with rectangular probability distribution is used to initialize the disturbance values according to the range and distribution function of the disturbance (Figure 4.4). Thus, the cell initialization is independent from distribution types and value ranges and therefore the initialization of the disturbance cells is simplified.
4.1.2 General disturbances

Disturbances of the permeability can be caused during several processing steps of the RTM process.

- Preforming
  Depending on the individual preforming process different sources of disturbance can be identified. There are for instance
  - imperfections of semi finished products
  - deformation of fabrics during cutting of layers
  - displaced layers or missing layers during stacking of the preform
  - variations of the thread tension during sewing, stitching or knitting
  - deformations due to handling and transport of the (semi-) finished preform
  - variations in permeabilities and fiber volume content due to draping [33]

- Preform positioning within the mold

- Mold closing
  Closing forces can drape the fabric or move the preform slightly within the mold, leading to flow channels at edges and joint lines

- Injection
  - fiber washing due to hydro-mechanical forces, particularly in unidirectional layers
  - mold deflections leading to changed fiber volume content
  - varying viscosity due to changed resin and mold temperature and due to curing of the resin
  - dependence of permeability on the resin’s flow velocity [60]

Variations of the fiber volume content highly depend on the tool’s precision and stiffness. Particularly for large thin walled parts this can become important. Laminates of 1 mm thickness can easily be affected by a variation of ±0.1 mm, resulting in a ±10% relative variation and a 2.5-fold permeability variation (according to the equation of Kozeny-Carman, eq. 4.12). Further variation can result from cores. The machining precision of cores is typically not as good as the precision of the mold. Often, the cores are
Table 4.1: Guide values for general disturbances cells

<table>
<thead>
<tr>
<th>Disturbed Parameter</th>
<th>Typical Value</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$ FVC</td>
<td>±2...10%</td>
<td>depending on the laminate thickness and mold stiffness</td>
</tr>
<tr>
<td>$K_1, K_2, K_3$</td>
<td>±10...20%</td>
<td>variation of permeability measurements</td>
</tr>
<tr>
<td>$\varphi, \theta, \psi$</td>
<td>±3°</td>
<td>variation of permeability measurements</td>
</tr>
</tbody>
</table>

not fixed in the tool but they are positioned by the encapsulating fabric. This positioning may differ for each injection and the core can move during the injection due to resin pressure.

To account for these influences general disturbance cells can influence

- the fiber volume content
- the principal permeabilities $K_1, K_2$ and $K_3$
- their directions described by $\varphi, \theta$ and $\psi$

As it is well known from literature there is a noticeable variation in permeabilities even under lab conditions. Depending on the fabric the variation is in the range of 5...20% [92]. The orthotropic ratio $K_1/K_2$ can be effected as well as the orientation of the principal directions. Usually these disturbances depend on the individual processing steps, the mold geometry and the material used. Rough guide values for the magnitude of these disturbances are given in table 4.1.

It requires a high effort to model all physical effects in flow simulations. A high number of material data might be necessary for modeling, as well as additional effort in data acquisition and simulation time. However, the influences of many disturbances mentioned above on the global filling are rather low. If the approximate range of variations in the filling process should be determined, the influence of the smaller effects on the permeability can be estimated and added to the global disturbance cells. Obviously, this approach does not provide more precise flow simulations but it can help to estimate the robustness of the simulated process.
4.1.3 Flow channels

Beside general disturbances, flow channels can occur. Flow channels, which are also referred to as race tracking flow channels, are areas without fiber material or with a very low fiber volume fraction. In these flow channels the flow resistance is very low compared to fiber filled areas with fiber volume fractions around 50%, and they cause high disturbances to the resin flow.

Flow channels occur with a high probability along lines at geometrically exposed locations like component edges, steps in the number of layers, stitching lines, joint areas or in corners. Their dimensions and statistical distribution may depend on the particular situation, but basic reoccurring types can be identified. For instance bending layers at a corner will generate certain flow channels, and different flow channels may be found when two cut layers join at a corner (figure 4.5). In filling thick parts flow channels can also occur as extended two dimensional areas instead of flow channel lines, if layers are displaced due to compression, injection at low fiber volume content or at high injection pressures.

For this disturbance type two different approaches to model these disturbances within the flow simulation are discussed. The first one (“permeability supplement”, page 87) calculates an equivalent permeability supplement \( K_{\text{plus}} \) for the flow channel area. In the second approach (“reduced fiber volume content”) the fiber volume content is lowered locally, directly leading to increased permeabilities in the concerned areas. The disturbance values of both approaches – permeability supplement and reduction of the fiber volume content – can be calculated according to the equivalent permeability of a Navier Stokes flow [89]. Thus, realistic values can be obtained for given flow channel dimensions.
Influence shape of disturbance cells on flow channel areas The flow channel width in the model can be adapted to the dimensions of the part. To model flow channels, an arbitrary permeability distribution over the flow channel width could be used. In order to obtain a smooth modeling in the finite element model, a triangular influence on the local permeability is desired.

Figure 4.6 shows the disturbance influence on flow channel elements schematically. As the effect of flow channels on the effective permeability is very high, high differences between neighboring elements can occur. For good simulation results, an effective flow channel width which is independent of the mesh size is desired. Using flow channel representation with rectangular shape, elements are influenced by a flow channel with full weight or not at all. As illustrated in figure 4.6 on the left side, the mesh size has a strong influence on the effective flow channel width for rectangular influence shape. A flow channel influence decreasing linearly with the distance from the flow channel line (triangular influence) is obviously less mesh dependent. Figure 4.7 shows an example of a flat plate with elements on the lower edge influenced by a flow channel with triangular influence shape.
4.1 Parametric simulation model

Figure 4.7: Modeling a flow channel on the lower side of the plate with a permeability supplement $K_{plus}$ applied over several elements in width. Dark red colors indicate high flow channel influence on the elements, blue indicates no influence.

Figure 4.8: Integration of resin flow over the flow channel area

For both, triangular and rectangular influence, a higher mesh resolution results in smoother flow channel representation. Therefore the model should be refined along flow channel lines, to ensure that two or more elements are within the influence width of the flow channel.

**Averaged permeability of 1D flow** To estimate the effect of a given permeability distribution within a modeled flow channel, the averaged permeability of a rectangular cross section with an exclusively one dimensional flow can be calculated. Thus, the impact of different flow channel modelings can be compared.

According to Darcy’s Law, the resin flow $q$ through the infinitesimal area $dA = t \cdot dy$ (figure 4.8) can be calculated as given by equation 4.1, where $\eta$ is the resin viscosity, $\frac{dp}{dx}$ the pressure gradient in flow direction, $K$ is the permeability in the flow direction and $\phi$ is the local porosity.

$$v_x = -\frac{K}{\eta} \cdot \frac{dp}{dx} \implies dq = K \cdot t \cdot \frac{dp}{\eta} \cdot \phi \cdot dy \quad (4.1)$$
The total volume flow $Q$ through the flow channel cross section with the modeling width $w$ can be calculated as

$$Q = \int_{y=0}^{w} dq = a \cdot \int_{0}^{w} K(\varphi(y)) \cdot \phi(y) \cdot dy \quad (4.2)$$

The averaged permeability $K_{avg}$ is defined in eq. 4.3 as the permeability that leads to the same volume flow at a porosity of $\phi = 1$

$$Q = K_{avg} \cdot w \cdot t \cdot \frac{dp}{\eta} \quad (4.3)$$

Finally, by combining the equations 4.2 and 4.3 the averaged permeability for a rectangular area can be calculated as

$$\Rightarrow K_{avg} = \frac{\int_{0}^{w} K(\varphi(y)) \cdot \phi(y) \cdot dy}{w} \quad (4.4)$$

The averaged permeability is related to a porosity of $\phi = 1$ because the porosity within the flow channel area is not constant in general.

**Equivalent permeability of flow channels with Navier-Stokes flow**

An equivalent permeability of flow channels can be calculated at which the volume flow of both, the Darcy channel and the laminar Navier-Stokes channel, are equivalent. Hagen-Poiseuille provides the solution for the volume flow through a tube with circular cross section.

$$Q = \frac{\pi r^4}{8 \eta} \frac{\partial p}{\partial x} \quad (4.5)$$

On the other side, Darcy’s Law can be used to calculate the volume flow through a porous channel

$$Q = A \cdot \phi \cdot v = \pi r^2 \cdot \frac{K_{eff}}{\eta} \cdot \phi \cdot \frac{\partial p}{\partial x} \quad (4.6)$$

Solving eq. 4.5 and 4.6 for $K_{eff}$ assuming a porosity of $\phi = 1$, the equivalent permeability for circular cross sections is

$$K_{eff} = \frac{r^2}{8} \quad (4.7)$$

Solutions for various channel cross sections are presented in [45] and [89]. Figure 4.9 shows two rectangular cross sections for which equivalent permeabilities are given. The left channel has one permeable wall as it is the
In the paper of Ni et al. [89] the denominator is given as  
\[ \beta_n \sinh^2(\beta_n W) \] 
which seems to be wrong and should not be squared, and instead be  
\[ \beta_n \sinh(\beta_n W) \] as given above in eq. 4.8. The subtraction of  
\[ \sinh^2(\beta_n W) - 1 \] in the numerator can lead to numerical erasure problems if the value of  
\[ \cosh(\beta_n W) \] becomes high, causing the calculation of the equivalent permeability to fail. This problem can be reduced by swapping \( H \) and \( W \) if \( W > H \) because thus the values of  
\[ \cosh(\beta_n W) \] are smaller. Using a series expansion of the numerator to a high order (approx. 200th order) led to a better numerical stability.

**Flow channel modeling with permeability supplement** Using equation 4.8 the equivalent permeability of a flow channel can be calculated. This equivalent permeability is applied as permeability supplement \( K_{PLUS} \), distributed over the runner model width \( w \) (figure 4.10). The total flow through a flow channel modeled in this approach is the sum of the flow through the porous area and the flow through the free flow area whose equivalent permeability is calculated as given by eq. 4.8.
Figure 4.10: Permeability supplement $K_{\text{Plus}}$ with triangular influence to model flow channel

\[ Q_{\text{Runner}} = Q_{\text{FreeFlow}} + Q_{\text{Darcy}} = -\frac{dp}{\eta} \cdot (HW \cdot K_{\text{eq}} + (A - HW) \cdot K_{\text{eff}} \cdot \phi_{\infty}) \]  \hspace{1cm} (4.9)

with
- $H$: Height of free flow channel
- $W$: Width of free flow channel
- $A$: Area of modeled channel
- $\infty$: Undisturbed values

To obtain a smooth influence of the flow channel, and therefore an acceptable independency of element sizes, a permeability supplement decreasing linearly with the distance from the flow channel line is chosen. The values of $K_{\text{Plus}}$ depending on the distance $r$ from the flow channel line can be calculated as

\[ K_{\text{Plus}}(r) = (1 - \frac{r}{w}) \cdot 2 \cdot K_{\text{eq}} \cdot \frac{A_{\text{channel}}}{A_{\text{model}}} \]  \hspace{1cm} (4.10)

where $A_{\text{channel}}$ is the cross section of the free flow channel in the mold, $A_{\text{model}}$ is the total area of the model that is influenced by the flow channel disturbance and $K_{\text{eq}}$ the equivalent permeability of the flow channel according to equation 4.8.

For the implementation in finite elements it is important to consider the different porosities. While the porous flow requires the local porosity information of the element, the supplementary permeability $K_{\text{Plus}}$ is related to a porosity of $\phi = 1$. 

Figure 4.11: Permeability of a flow channel with Navier Stokes flow. Values are averaged over a stripe of 20 mm width, assuming a permeability of $4 \cdot 10^{-11} m^2$ in the porous area and 50% fiber volume content.

**Example for averaged permeability** Using equation 4.9, the averaged permeability can be described as

$$K_{\text{stripe}} = \frac{1}{A} \cdot (HW \cdot K_{\text{eq}} + (A - HW) \cdot K_{\text{eff}} \cdot \phi)$$  \hspace{1cm} (4.11)

Figure 4.11 shows an exemplary calculation result of the permeability averaged over a 20 mm wide stripe. Therefore the equivalent permeability of a solid walled, rectangular flow channel has been calculated as described by [89], and a porous material with a permeability of $4 \cdot 10^{-11} m^2$ had been assumed in the remaining area. The different lines represent the results for different flow channel heights or laminate thickness, respectively. On the abscissa the width of the flow channel is given. It can be seen that the influence of flow channels with a width below 1 mm is already significant.

The minimum permeability for a channel width of zero reaches $2 \cdot 10^{-11} m^2$ although the undisturbed permeability of the preform is $4 \cdot 10^{-11} m^2$. This is due to the definition of the averaged permeability in equation 4.4, where $K$ is multiplied by the porosity.

**Flow channel modeling with reduced fiber volume content** In this approach the fiber volume content is reduced in the influence area of
a flow channel. Thus the corresponding permeability is increased automatically, according to the Kozeny-Carman equation

\[ K_{kc}(\varphi) = \frac{c \cdot (1 - \varphi)^3}{\varphi^2} \]  (4.12)

with \( K_{kc} \) calculated permeability \([m^2]\), \( c \) Kozeny constant \([m^2]\), \( \varphi \) fiber volume content [-]

The Kozeny constant \( c \) has to be determined from permeability measurements, but conversion between two fiber volume contents \( \overline{\varphi} \) and \( \varphi \) follow the equation

\[ K_{eff}(\varphi) = \frac{K_{eff} \cdot (1 - \varphi)^3 \cdot \overline{\varphi}^2}{(1 - \overline{\varphi})^3} \]  (4.13)

with \( K_{eff} \) effective permeability \([m^2]\), \( \overline{K}_{eff} \) reference permeability \([m^2]\), \( \varphi \) fiber volume content [-], \( \overline{\varphi} \) reference fiber volume content [-]

The effect of reducing the FVC is rather high, according to eq. 4.13 a reduction of the FVC from 60% to 50% causes an increase of the permeability by a factor of approx. 2.

**Line areas of increased fiber volume content**  Beside flow channel with increased flow velocities at cavity edges and other geometrically predestined locations, the opposite effect may occur as well. A locally increased FVC can be expected where the fabric is forced into the cavity when it is closed. So, the effect of an increased FVC can occur at similar geometrically predestined locations like flow channels. It can be modeled using the same influence factor \( f_{FVC} \) as for reduced fiber volume contents. In these cases \( f_{FVC} \) can be set to values above 1.0. It seems to be useful to vary the factor \( f_{FVC} \) in a range of [0.2, 1.4] at certain edges.

However, compared to flow channels the effect of a locally increased fiber volume content on the global flow is limited. This can be seen in equation 4.4 where the averaged permeability is not changed noticeably, even if the permeability in the small flow channel area drops to zero.

**Parameters for flow channels with reduced fiber volume content**  First the effect of the distribution function for reduction of the fiber volume content will be discussed. For smooth modeling of flow channels,
4.1 Parametric simulation model

the permeability should decrease roughly linear to the undisturbed values with increasing distance from the runner line.

Because the influence of the FVC on the preform permeability is strongly non-linear the distribution of the effective permeability within the flow channel area has to be considered. The effective permeability at a certain location can be calculated using equation 4.13. The linear and the quadratic influence shapes shown in figure 4.12 are considered.

As figure 4.13 (left) shows, the linear $f_{FVC}$ distribution leads to a narrow peak near the flow channel line. As the center of the element will never be within this high influence area, a very small mesh size is required to map the permeability distribution properly onto the elements. For quadratic influence, the peak is considerably broader and shows a more linear characteristic (figure 4.13, right). So, the flow channel can be mapped better and the permeability distribution gets smoother.
Averaged permeability of flow channel modeled with reduced FVC

![Graph showing averaged permeability vs reduction factor for different FVC values](image)

Figure 4.14: Averaged permeability of a flow channel modeled with reduced fiber volumed content, assuming a permeability of $4 \cdot 10^{-11} \text{m}^2$ in the porous area at 50% fiber volume content. The averaged permeability is independent of modeled width $w$.

Using equations 4.4 and 4.13, the averaged permeability can be calculated. Figure 4.14 shows the calculated permeability for various parameter combinations averaged over the modeled channel width. The influence of $f_{\text{FVC}}$ as well as the averaging area depend on the modeling width $w$. So, the averaged permeability is independent of the model width.

The diagram in figure 4.15 shows the correlation between the reduction factor $f_{\text{FVC}}$ for the fiber volume content and the free flow channel width. An undisturbed FVC of 50% and a modeling width $w = 0.02 \text{ m}$ have been assumed for the calculation of the correlation.

Geometric influence on resulting flow channel strength In the disturbance cell concept the influence on elements depends on the distance from the defining flow channel line. Figure 4.16 shows three different flow channel locations, at an edge, at a corner and at the joint line of three laminates. Depending on the position of a flow channel within the part, its effect on the simulated flow will be different. Thus, flow channels at joint lines or at corners have an increased influence on the flow due to the modeling. So, the disturbance parameters need to be adapted to the number of laminates joining at the flow channel line.
Figure 4.15: Correlation between FVC reduction factor $f_{\text{FVC}}$ and free flow channel width $W$. The FVC in the undisturbed area is 50% and the model channel width $w = 0.02$

Figure 4.16: Single sided flow channel at an edge, two sided flow channel at a corner and three sided flow channel at joint line
Figure 4.17: Linear probability distribution used in stochastic determination of free flow channel width $w$

For modeling flow channels with permeability supplement, $K_{\text{plus}}$ can be divided by the number of connected laminate sections. If flow channels are modeled with reduced fiber volume content, the influence of $f_{\text{FVC}}$ is non-linear, and slightly higher values for $f_{\text{FVC}}$ should be chosen.

**Statistical distribution of flow channel strength**  Few statistical data about frequency and strength of flow channels is available. Li et al. [73] carried out some experiments in a plane cavity and evaluated a multiplication factor for the local runner permeability compared to the undisturbed permeability. For this work a Weibull distribution fitted the found multiplier best. However, generalization to different geometries and fabrics does not appear to be sensible. The permeability multipliers depend on the undisturbed permeability, and their relation to the flow channel dimensions is highly non-linear.

Therefore, a triangular probability distribution is used for $K_{\text{plus}}$ (figure 4.17) and a uniform distribution is used for $f_{\text{FVC}}$. This corresponds to a small influence of the flow channel for most simulation runs. It seems highly possible that each component might have different probabilities of the effective flow channel width, even each flow channel location might have a different probability distribution of the flow channel width.

For the initialization of disturbance cells to model flow channels with permeability supplements, a linear decreasing probability distribution is used (figure 4.17). The highest probability is for a flow channel with a width of $w = 0$, and a flow channel with a width of $w_{\text{max}}$ has a probability of 0. The integral of the probability distribution is

$$F = \int_{\hat{W}=0}^{W} p \cdot d\hat{W} = \int_{0}^{W} \left(-\frac{\hat{W}}{W} + 1\right) \cdot d\hat{W}$$
4.1 Parametric simulation model

Figure 4.18: Inverse of cumulated probability function of the linear function given in figure 4.17, used to initialize a random flow channel width according to the given probability

This can be inverted in order to get the inverse distribution function that can be used to initialize the channel width $w$ according to the desired linear probability distribution (figure 4.18).

$$W = W_{\text{max}} \cdot (1 - \sqrt{1 - s_{\text{init}}}) \quad (4.14)$$

Simulation examples for influence of modeling parameters  Figures 4.19 and 4.20 show simulation results for different models. All of them are based on the equivalence to a $1 \times 1$ mm flow channel. There is a clear difference between each modeling, particularly the channel width $w$ has a dilating influence on the advancement of the resin in the flow channel.

For the derivation of the averaged permeability a pure 1D flow has been assumed. However, because the real flow has a transverse flow component as well, there will be slight differences between the modelings, although the same averaged permeability was calculated. The major influence is the modeling width $w$, where small widths cause narrow flow channels in the simulation.

Conservation of resin  Numerical solutions of equation systems are generally affected by numerical inaccuracy. These errors depend on the used solution algorithm, as well as on the condition of the equation system [23]. In case of strong differences between the permeabilities of adjacent elements, which may occur if flow channels are modeled, the condition
Figure 4.19: Fill pattern for line injection at constant pressure on the left edge. The flow channel values are based on a 1x1 mm free flow channel and a cavity size 300 x 100 x 1 mm with an undisturbed FVC of 50%.
Figure 4.20: Fill pattern for point injection with constant volume flow at the lower left corner. The flow channel values are based on a 1x1 mm free flow channel and a cavity size 300 x 100 x 1 mm with an undisturbed FVC of 50%.
number of the matrix is negatively affected. This problem increases for larger matrices.

Thus, the conservation of the resin volume has been verified by running fill simulations with a constant volume flow at the gate. The total net flow over all control volumes in the simulated injection domain must be zero, and the total resin flux at the flow front equal to the injected resin flux. A two dimensional model with linear triangular elements and a three dimensional model with linear tetrahedral elements have been generated with a flow channel on the upper edge. The graphics in table 4.2 show the element permeabilities, the arrival times of the flow front and the relative error $\varepsilon$ in the total resin flux

$$\varepsilon (t) = \frac{1}{q_{in}} \cdot \sum_{i=1}^{k} (q_i (t) - q_{in})$$

with  
- $k$ : Number of nodes at the flow front  
- $q_{in}$ : injected volume flow  
- $q_i$ : nodal volume flow

In this test a rectangular influence shape of the flow channel has been used. On the dark red elements, a permeability supplement of $1.28 \cdot 10^{-7} \text{ m}^2$ has been applied, while the undisturbed element permeability is $1.4 \cdot 10^{-11} \text{ m}^2$. Thus, an extreme permeability difference between two adjacent elements at the border of the flow channel of nearly 4 orders of magnitude was obtained. A constant volume flow has been injected at the lower left corner and a vent has been placed on the lower right corner. For the 2D simulation, a relative error below $\|\varepsilon\| < 10^{-9}$ has been determined, which is within the precision of the used variable type (double). The 3D simulation showed relative errors below $\|\varepsilon\| < 0.25\%$. In this test case, the relative error has been at a constant level throughout the simulation, even in the beginning of the fill process before the high permeable elements are reached. So, even for extreme permeability differences between adjacent elements a good conservation of the resin volume has been observed.

### 4.1.4 Experimental investigation of flow channel geometry

Appropriate initialization of the flow channel dimensions and their statistical distribution remains a critical issue. To investigate the variation of flow channels experiments were performed on a complex shaped shell geometry [64]. The produced part is 180 mm long, 120 mm wide and 30 mm
### 4.1 Parametric simulation model

#### Mesh size and permeability distribution:

<table>
<thead>
<tr>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Elements: 17128</td>
<td>Number of Elements: 69560</td>
</tr>
<tr>
<td>Number of Nodes: 8765</td>
<td>Number of Nodes: 13728</td>
</tr>
</tbody>
</table>

#### Fill pattern (flow front arrival times):

#### Relative Error of flow front net flows:

Table 4.2: Test of resin conservation in models with strong permeability discontinuities
Figure 4.21: Microscopy of edge in an area of two layers ±45 glass fabric, samples taken at the same position of three different parts.

Figure 4.22: Microscopy of edge in an area of three layers ±45 glass fabric, samples taken at the same position of three different parts.

high. From these parts cut outs for microscopy were prepared, coming from different positions on the part. The pictures in the figures 4.21, 4.22 and 4.23 show the same location for three different parts.

Figure 4.21 shows the results for a position with two layers at an edge, where rather varying results were obtained. The left picture shows a constant fiber filling that is similar to the undisturbed area. The next picture shows a strong flow channel with an approximate cross section of 0.5 mm$^2$. On the right picture the opposite effect of an increased fiber volume content can be seen.

The results found for a similar edge situation, but with three layers seems to be a bit more constant (figure 4.22). All cut outs are filled with fibers, without significant variation in the fiber volume content. Only one sample shows a large flow channel with a cross section of about 0.3 mm$^2$ (figure 4.22, left) It can be seen that one layer reaches the edge, the second layer is slightly separated from the edge, and the third layer is one step more retracted.

A different situation is shown in figure 4.23, where two layers are bent in a corner. Due to the stiffness of the fibers, the outer point of the corner is never filled with fibers, and therefore a flow channel of rather reproducible dimensions occurs. In addition to the outer flow channel, two flow channels on the inside occur. Those are narrow, but wide extended along the inner walls. It seem that the dimension of these flow channels depend on the stiffness of the fibers and therefore appear to be well reproducible.
4.1 Parametric simulation model

Figure 4.23: Microscopy of corner in an area with 2 layers ±45 glass fabric, samples taken at the same position of three different parts

4.1.5 Initialization of disturbance cells

To initialize the parametric model in a reproducible and controlled way, initialization with a predefined list of input numbers shall be possible. Thus, the model can be initialized with a Latin-Hypercube scheme, and it can be simulated with the same disturbances but different process condition.

The disturbance cells need to be initialized within the given disturbance range. Input numbers with a uniform probability density in the range $[0, 1]$ should be used for initialization. Thus, stochastic disturbance can be applied to the disturbance cells, based on initialization numbers and the predefined ranges.

There are various methods to generate normal distributed random numbers. For most of them, random numbers with a uniform probability distribution in a range $[0, 1]$ or similar are required. Several generation methods need a varying number of input numbers because they need to rerun the calculation if intermediary results do not meet certain criteria. Examples are the Polar-Method and the Rejection-Method. Those methods are not suitable for the initialization of the parametric model.

The Box-Muller-Method [123] (algorithm 1) can be used to generate pairs of normal distributed random numbers with a mean value of $\mu = 0$ and a standard deviation $s = 1$. Two uniformly distributed input numbers on the interval $(0, 1]$ are required. Thus a reproducible sequence of random numbers can generated. It can be used for general disturbance cells, where several normal distributed random numbers are required.

For flow channel cells, where only one normal distributed number is required a numerical approximation of the inverse of the cumulated density function of the normal distribution is made. The boost libraries [18] provide useful methods for such calculations.
Algorithm 1 Box-Muller-Method

1. generate uniformly distributed \( u_1, u_2 \sim U(0, 1) \)

2. \( r = \sqrt{-2 \cdot ln(1 - u_1)} \) and \( \varphi = 2\pi \cdot u_2 \)

3. \( z_1 = r \cdot cos(\varphi) \)
   \( z_2 = r \cdot sin(\varphi) \)

4.1.6 Conclusions

The disturbance cell concept allows to include expert knowledge about variations in the simulation model. Statistical distributions of the model data, e.g. of the permeabilities, of the fiber volume content or the occurrences of flow channels can be included. The influence of flow channels on the permeability could be derived from their physical dimensions. Typical flow channel dimensions were obtained from an exemplary investigation. However, conclusions about the statistical distribution of the flow channels cannot be drawn due to the low number of available samples. Different modelings of flow channels were compared and a similar effect was found. Investigations of very strong flow channels showed numerical difficulties with large models. However, these are irrelevant for common problem sizes, particularly if the suggested smooth modeling of flow channels is applied.

4.2 Monte Carlo estimation of process variations

As it has been discussed in the previous section, disturbances may have a strong influence on the injection process. Therefore, process robustness is a critical issue for the development of RTM processes. By including stochastic disturbances in the permeability model, the process variations can be estimated in a simulation environment using Monte Carlo simulations [38, 75]. Within the Monte Carlo simulation, the parametric simulation model described in section 4.1 is initialized stochastically and several simulations with the same process parameters are run. The resulting output variations show the range of variations that has to be expected for the real process. If an appropriate criterion is defined, the scrap rate can be estimated. In this work, a combination of fill degree and matrix quality is used as scrap criterion.
Because Monte Carlo methods are sampling methods, the estimation results show variances depending on sample size and random data. Hence, the estimate should be accompanied by an estimate of its variance [38]. Bootstrap and sub-sample techniques can be used to estimate the variance of Monte Carlo estimations. In [32] several estimates are compared with respect to their variance. Confidence Intervals for distributions similar to the normal distribution can be determined using the Student-t distribution. Particularly if large sample sizes are available, the confidence intervals can be calculated for distributions differing from the normal distribution as well [102].

4.2.1 Initialization of the parametric simulation model

Besides setting the process parameters, the disturbances of the parametric simulation model need to be initialized before simulations can be run. As described in section 4.1.5 the disturbance cells have to be initialized by random numbers between 0 and 1 with a uniform probability distribution. The disturbances of the finite elements that are linked to the disturbance cells will therefore be initialized within the variation ranges of the cells.

Assuming independency of the disturbances, the initialization can be done using a plane Monte Carlo initialization (figure 4.24, left). However, the coverage of the parameter space is not optimal for plane initialization. Some parameter combinations are emphasized, others are not investigated. Correlations in the input values of the simulation model will cause artificial correlations in the simulation output. This problem can be reduced by using a Latin Hypercube initialization [84, 20, 119]. Thus the parameter space is well covered and irregular correlations in the disturbances are minimized. Theoretical and empirical evidence on the superior performance of Latin Hypercube sampling compared to simple random sampling has been provided by [109, 110].

4.2.2 Variation of simulation output

Figure 4.25 shows a part with a complex shell geometry for which a Monte Carlo simulation has been run. In parts with undercuts such as this sort box, flow channels often lead to the formation of gas entrapments. As it could be seen in section 4.1.3, flow channels may have a very high influence on the injection. So, their parameters have to be adjusted carefully to the individual situation for realistic results. Here a maximum channel width $W_{max} = 0.4 \text{ mm}$ has been used. The permeability supplements $K_{\text{Plus}}$ to model the flow channels for one specific initialization are illustrated in
Figure 4.24: Plain Monte Carlo initialization (left) and Latin Hypercube initialization (right) of input parameters.

Figure 4.25: Demonstrator geometry with an overall part size of 180 x 120 x 30 mm, all walls have a thickness of 1 mm.
4.2 Monte Carlo estimation of process variations

Figure 4.26: Model disturbed by flow channels, stochastically initialized permeability supplement $K_{\text{plus}}$ on base of 0.4 mm wide channels in maximum (left). Intermediate fill pattern, showing race tracking effects on several part edges (right).

Figure 4.26 (left). In figure 4.26 (right) an intermediate fill state for this injection model is presented, showing the effect of flow channels on the fill pattern. As it can be seen, it has a strong impact on the flow front.

In a Monte Carlo Simulation several simulations are run to evaluate the range of the output variables, e.g. fill time and fill degree. Compared to commonly used simulations with a deterministic model these output variables are not a fixed number, but a range of expected values and their distribution.

The diagrams in figure 4.27 show the distribution of the output variables fill time, final fill degree and laminate quality. The laminate quality criterion is introduced in section 3.6.3 and is a measure for the void content averaged over the part. Obviously there is a high effect of the disturbances on the simulation output. Particularly the fill time is strongly reduced from 82 s without disturbances to an average of 52 s, corresponding to a reduction to 63%.

On the one side, even for small (narrow) flow channels below 0.4 mm a strong reduction of the filling time and a strong influence on the fill pattern could be observed. On the other side, general disturbances such as disturbances of the fiber volume content of ±5% showed a comparatively low influence on the filling pattern and the fill time (figure 4.28).

4.2.3 Estimation of the scrap rate

$\mathcal{R}$-criterion A low scrap rate is an important target for process development. To define a suitable criterion to estimate the scrap rate and to assess the quality of a manufactured part various parameters may be
Figure 4.27: Variation of the simulation output parameters fill time, final fill degree and laminate quality due to flow channels
Figure 4.28: Variation of the simulation output parameters fill time, final fill degree and laminate quality due to variations of the fiber volume content of ±5%
of interest. In this work, the definition of a scrap part is based on two parameters, the fill degree and the laminate quality criterion, where low values indicate good quality. An elliptical interpolation is used to couple both parameters to the $R$-criterion

$$ R = \sqrt{\left(\frac{\text{Fill}}{\text{Fill}_{\text{min}}}\right)^2 + \left(\frac{\text{LQ}}{\text{LQ}_{\text{max}}}\right)^2} \quad (4.15) $$

By this definition, the $R$-value corresponds to the distance from the point of optimum part quality, which is the top left corner in figure 4.29. It can be used to assess the part quality with respect to the defined limits for the minimal required fill degree $\text{Fill}_{\text{min}}$ and the maximum laminate quality value $\text{LQ}_{\text{max}}$. While values of $R = 0$ represent a perfect part quality, values of $0 < R \leq 1$ indicate an acceptable part quality. Simulation results with higher values ($R > 1$) mark scrap parts. This is illustrated in figure 4.29, where the gray area represents the scrap region, and the dark blue area marks the region of good parts.

**Monte Carlo estimation** Each fill simulation with a specific initialization of the disturbance cells provides a single $R$ value. The simulation returns the evaluation of $f(\bar{P}, \bar{s})$ for given geometry, injection parameters $\bar{P}$, and the current disturbance values $\bar{s}$. For typical RTM parts, the vector of injection parameters $\bar{P}$ has 1 to 10 entries, the vector of disturbances $\bar{s}$ about 10 to 100.
4.2 Monte Carlo estimation of process variations

The two quality relevant output parameters, the final fill degree and the laminate quality criterion, are presented in an anthill plot in figure 4.30. It shows the results of a Monte Carlo simulation with 500 runs. The scrap rate can be determined by evaluating the number of points inside and outside the good part area.

Replacing the fix value vector \( \bar{s} \) in equation 4.16 by its distribution \( \bar{S} \), a Monte Carlo simulation provides an estimation of the distribution \( p_R \) of the \( R \)-criterion.

\[
\mathcal{R} = f(\bar{P}, \bar{s}) \quad \text{with} \quad \bar{P} : \text{Injection parameters} \quad \bar{s} : \text{Vector of disturbances} \\
\]

\[
p_R = F(\bar{P}, \bar{S}) \quad \text{with} \quad \bar{P} : \text{Injection parameters} \quad \bar{S} : \text{Vector of disturbance distributions} \\
\]

In figure 4.31 the distribution of the \( R \) values is shown for the two simulations of the earlier presented sort box. The upper distribution results from the simulations with flow channel disturbances, the lower from...
Figure 4.31: Variation of the calculated $R$ values due to flow channels (upper, cmp. figure 4.27) and due to variations of the fiber volume content of $\pm 5\%$ (lower, cmp. figure 4.28). The used limit value were $\text{Fill}_{\text{min}} = 0.995$ and $\text{LQ}_{\text{max}} = 0.25$. 
disturbances of the fiber volume content. The quality limits had been set to Fill_{\text{min}} = 0.995 and LQ_{\text{max}} = 0.4.

**Reduced number of required simulation runs** However, for many applications the high number of 500 simulation runs is not acceptable. To derive useful information from small numbers of Monte Carlo runs, suitable probability distributions can be fitted onto the simulation output data. Thus, an analytical approximation of $p_R$ is obtained. Integrating $p_R$ over the value range for good parts $R = [0, 1]$ provides the rate of good parts (figure 4.32), and the scrap rate can be expressed by the complement

$$\text{Scrap rate} = 1 - \int_0^1 p_R \cdot dR \quad (4.18)$$

The distribution function of the $R$-criterion is not known a priori. To investigate this point, Monte Carlo simulations for several problems were run. Depending on part geometry and processing parameters, in particular the gate and vent locations, different distributions of the simulation output have been observed.

To adapt the distribution models to sample data a Maximum Likelihood Estimation has been used [88, 100]. Beside the optimum parameters for the distribution, the Maximum Likelihood Estimation returns the total sample probability, which can be used to assess the approximation quality. Thus, the best fitting distributions among the adapted distributions can be selected for each case. In figure 4.31, for example, log-normal distributions can be used to approximate the $R$-distribution. Distributions found to generally match the simulation results are

- Normal (Gauss) distribution
4.2.4 Prediction of problem areas

Figure 4.33 shows the final fill degree averaged over 500 simulations. This illustration shows at a glance the areas where high probability of unfilled spots are to be expected. Because the fill degree is 0 in the unfilled areas and 1 in all filled areas, and because the flow front separates those areas rather sharp, the averaged fill value can be interpreted directly as the probability of filling of this location.

However, some information is lost in this graphic. For instance, it is not possible to distinguish between rare but large dry spots on the one hand, and small but frequently occurring spots at similar locations on the other hand. Both can lead to the same averaged results. As detailed investigations of several runs showed, small but more frequent dry spots cause the problem areas in figure 4.33.
4.3 Conclusions for simulation of process variations

4.2.5 Conclusion

Including process variations in the simulation model allows to estimate process variations with Monte Carlo simulations. Noticeable variations have been found for the final fill degree and the fill time. While variations of the fiber volume content cause variations of these output values, flow channels cause variations plus an additional shift to shorter fill times. Problematic areas, as for example dry spots, may be identified in illustrations of the final fill state, averaged over many simulation runs.

A criterion to distinguish between scrap and good parts has been developed. It requires the desired minimum fill degree and the minimum required laminate quality as input parameters and may be applied in automated numerical process optimizations to determine the scrap rate of a specific process.

4.3 Conclusions for simulation of process variations

The influence of disturbances like flow channels and variations in the local fiber volume content on the fill pattern and the fill times is significant. They mainly result from preform manufacturing and handling. A parametric permeability model has been developed to allow the simulation of stochastic variances and disturbances in the resin flow. In this model, instead of one deterministic average value, statistical distributions of the material data can be used. Thus, the effect of stochastic variations on the injection process can be simulated. Flow channels can be applied by directly defining their geometrical dimensions. This is advantageous compared to multiplication factors for the permeability, which is also suggested in the literature, because the permeability of a flow channel depends on its geometry. A given flow channel has a stronger influence in a low permeability surrounding than in a high permeable area, which is ignored if multiplication rules are used.

Depending on the individual geometrical situation, the size and the variation of disturbances as well as their statistical probability distribution can be estimated. Due to the large quantity of experimental data required to determine statistically reliable data, further investigations of flow channel sizes and general disturbances of the permeability and the fiber volume content are necessary.

Monte Carlo simulations have been used with the parametric simulation model to evaluate the resulting process variations. Instead of fixed
output values, an expected range may be obtained. By applying a suitable criterion, the scrap rate of a particular process can be predicted. The scrap rate is a critical issue for RTM injection processes and should in general be minimized. To obtain realistic results from Monte Carlo simulations, it is important to apply realistic assumptions about disturbances, including the sizes of flow channels, deviations of the fiber volume content and the permeability as well as their frequency of occurrence. Therefore, further investigations are necessary. Relations between disturbances may occur due to geometric coupling, and should be considered in future contributions.
Chapter 5

Optimization of processing parameters

Compared to heuristic optimization numerical optimization techniques provide several advantages. Due to non linear interference of parameters useful solutions can be found for problems with high complexity, that exceed the intuition and process understanding of a process engineer. For instance the influence of stochastic variations in the input parameters easily exceed intuition. In these cases intuition or experience may mislead optimizations. Therefore, numerical optimizations enhance the possibility of obtaining improved, nontraditional designs. Optimizations virtually always achieve improved parameters, e.g. Giger found a CFRP-design for a highly optimized magnesium motorcycle racing rim, which reduced the total weight by 20% [39].

In addition, shifting the process design of RTM processes from an experimental environment to a virtual simulation environment reduces the development time and cost. Besides, if a structured approach is followed, the process design can be faster, less expensive and better planned. During the design phase simulation and optimization tools should be used to determine optimum processing parameters. Numerical fill simulations can be used to predict the mold filling. Thus, for instance the gate and the vent locations can be optimized, or the preform may be modified. Numerical process optimizations can automize this process, minimizing the required amount of manual process assessment and the input of the process designer. Using these simulation based methods, the experimental effort to determine the optimal parameters is minimized.
As discussed in section 4.1, the permeability distribution is not perfectly reproducible and differs with each injection. This causes major stability problems for the process as the changed permeability significantly changes the fill pattern. Therefore, the complete filling of the mold may be prevented or joint lines may occur in mechanically high stressed areas. In the case of increased cycle times, premature curing of the resin may avoid a successful filling of the mold. Hence, it is necessary to consider process robustness in numerical optimizations. Thus, the scrap rates can be reduced while the additional effort is low compared to a conventional deterministic process design. Robust optimizations require higher computational power and disturbances have to be quantified in the model. However, compared to feedback controlled system, no expensive computer controlled injection devices or flow sensors with coupled online calculations are necessary.

In this chapter an approach to improve the process development with numerical optimizations is presented. In this approach, the flow simulation software sLIP has been coupled with an evolutionary optimization library [67, 65, 57]. Discussed aspects are:

- **Heuristic optimization** (section 5.1)
  Example of an optimized sequential injection.

- **Numerical optimization algorithms** (section 5.2)
  Review of relevant optimization algorithms.

- **Optimization parameters** (section 5.4)
  Parameters, which should be optimized during process design in order to obtain good processes.

- **Optimization objective** (section 5.4)
  Definition of the optimization objective which may be composed of several sub objectives. Adaption of the objectives for a good convergence of the numerical optimization procedures.

- **Reliability of Monte Carlo estimation** (section 5.5)
  Number of required simulation runs to estimate the process reliability.

- **Connector remeshing** (section 5.6)
  Placing additional gates and vents with a defined effectiveness.

- **Optimization results** (section 5.7)
  Optimization examples with evolutionary algorithms to test and verify the developed procedures are presented. Results for heuristically optimized processes, numerical optimized processes including
5.1 Heuristic optimization

As stated earlier, for geometrically simple parts an intuitive process optimization can result in significant improvements. The example of a hockey stick manufactured in an RTM process illustrates this. In the original process (figure 5.1) a single gate is used and a constant volume flow is applied. After 360 s the component is filled and a high maximum pressure of 18 bar is reached.

For this part an empirical process optimization has been performed aiming at lower injection times and lower injection pressures, simultaneously. Therefore a sequential injection strategy has been chosen, shortening the effective flow path of the resin. After 40 s gate 1 is closed and the gates 2 and 3 are opened. Thus an injection with a maximum pressure of 8 bar and a filling time of 120 s are achieved (figure 5.2).
5.2 Numerical optimization algorithms

To apply numerical procedures to an optimization problem, the objective and parametrization should be described in a mathematical formulation. An algorithm suitable to the problem has to be selected. Section 5.2.2 provides a general overview of established optimization methods. In sections 5.2.3 to 5.2.5 selected methods that appear to be suitable for the processes optimization are described in detail.

5.2.1 Formulation of optimization problem

The nonlinear optimization problem with constrains can be written as follows [115]:

![Figure 5.2: Heuristically optimized injection process](image)

- Process history
  - Max. Pressure: 8 bar
  - Filling time: 120s

- Injection parameters:
  1. 1.2 cm³/s 0 to 40s
  2. 1.2 cm³/s 40s to 105s
  3. 1.2 cm³/s 40s to 120s
5.2 Numerical optimization algorithms

Minimize: \( f(X) \)  
Subject to:  
\[ g_j(X) \leq 0 \]  
\[ h_k(X) = 0 \]  
\[ x_i^l \leq x_i \leq x_i^u \]  

with \( X = \{ x_1, x_2, x_3, \ldots, x_n \} \)  

objective function  
inequality constraints  
equality constraints  
side constraints  

(5.1)

where the vector \( X \) contains the optimization (design) variables. The objective function as well as the constraining functions may be linear or nonlinear functions of \( X \). They may be explicit or implicit in \( X \) and may be evaluated by any analytical or numerical technique.

If mathematical programming is used, it is important that these functions are continuous and have continuous first derivatives in \( X \). If these conditions are not satisfied, for instance when discrete valued variables appear, one must either invent homogenization techniques such as [13] or resort to other methods such as genetic algorithms [70].

### 5.2.2 Optimization algorithm overview

There is a large number of optimization methods and derived algorithms. A concise overview over numerical optimization algorithms is given by Müller [86]. Figure 5.3 summarizes her classification of optimization methods.

Direct methods require only evaluations of the objective function. They can be divided into stochastic and deterministic methods. Deterministic methods, such as Simplex optimization or the Response Surface method, evaluate the objective function, but do not require gradients of the objective function. In the Simplex method, new search points are calculated following a fixed systematic procedure depending only on the evaluated values of the objective function.

In contrast to deterministic methods, where one single start point is used, in stochastic optimization methods a whole set of start points (population) is used. New search points in the iterative solution procedure are determined by using random numbers. Typically, new search points tend towards the direction of previously found good solutions. To facilitate
explorative behavior, some spreading elements for new search directions are incorporated as well. Stochastic optimization algorithms are robust against noise and do not need gradient information. Popular methods are

- Particle Swarm optimization
- Genetic algorithms
- Simulated Annealing

Indirect methods are deterministic and use gradient information of first and second order. They often have the advantage of a fast convergence rate, particularly near the optimum. Important examples for gradient based methods are:

- Cauchy’s method of steepest gradient
- Fletcher Reeves method of non-linear conjugated gradients [104]
- Response surface method
- Newton method

For most practical applications in the engineering field, the objective function is not accessible analytically. So, gradients have to be approximated by difference quotients. Beside problems with numerical noise, the computational effort to determine gradients is often rather high.

All deterministic methods including direct and indirect methods, which are often referred to as mathematical programming methods, work locally. Starting at one solution, a suitable criterion based on local functions and
5.2 Numerical optimization algorithms

derivative information is used to calculate a new solution. This new solution is accepted if a better function value is obtained, or rejected otherwise. Because of the limitation to local information, the iterative optimization can easily be trapped at local optima. So, depending on the start point of the optimization, different solutions may be found, if there are local optima. Therefore, the optimization cannot guarantee to find the global optimum. Difficulties in finding the optimum may also result from the numerical scattering of the calculated gradients. Noise can occur due to the domain discretization used in numerical simulation. Derivatives are usually approximated by difference quotients causing further difficulties, as difference quotients depend on step width as well as on the magnitude of the neglected derivatives of higher order. A possible improvement is to use a multi start strategy. Optimizations are started from various initial solutions usually chosen in a stochastic procedure, improving the probability to find a global optimum, but accepting a significant reduction of the search efficiency.

5.2.3 Response Surface method

The most time consuming part of most practical optimization problems is the evaluation of the objective function. In the Response Surface method, the number of function evaluations is reduced. Therefore, the original objective function as given in equation 5.1 is locally approximated by an analytical function. Thus, gradients can be calculated analytically without the need of numerically determined derivatives.

Minimize: \( \tilde{f}(X) \) response surface function
Subject to:
\[
\begin{align*}
\tilde{g}_j(X) & \leq 0 \quad \text{inequality constraints} \\
\tilde{h}_k(X) & = 0 \quad \text{equality constraints} \\
x_l^i & \leq x_i \leq x_u^i \quad \text{side constraints}
\end{align*}
\] (5.2)

The analytical response surface function is fitted to objective values, evaluated from the original objective function at suitably defined points. Thus, an approximate optimum point can be obtained from \( \tilde{f}' = 0 \), which now can be determined analytically. If the response surface function \( \tilde{f} \) is similar to the original function, the optimum of the response surface is close to the local optimum of the original function.

Knowledge about the shape of the objective function can be used to define a problem specific response surface. Another possibility is to use quadratic functions, which are the simplest functions with a minimum point and the minimum point can be determined easily. For practical
problems, quadratic functions provide good approximations, particularly near the optimum point.

Supporting points have to be determined to establish the response function. The minimum number of support points equals the number of parameters in the response function. It can be shown, that the Response Surface method is equivalent to the Newton method for a specific scheme for selecting support points [69]. Choosing more support points than required can help to reduce numerical noise, which is inherent to most numerical analysis procedures. In this case of an overdetermined system, the response surface can be defined by a best fit procedure. [70, 116]

### 5.2.4 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy [58], inspired by social behavior of birds or fish schooling. Particles with moving procedure using local information from neighbor particles can show hardly predictable behavior. This can be beneficially used to search the optimum of an objective function.

Each particle represents a potential solution of the optimization problem. The system is initialized with a population of random solutions and searches for optima by updating iteratively. The particles fly through the problem space by following the current optimum points (figure 5.4). The investigated solutions are stored by the particles, where each particle remembers its best solution $X_{\text{particle best}}$. The best solution found so far by the swarm is stored as $X_{\text{global best}}$. The iterative update of the particle solutions is calculated by equation 5.3

\[
\begin{align*}
    v_{i+1} &= c_0 v^i + c_1 r_1 (X_{\text{particle best}} - X^i) + c_2 r_2 (X_{\text{global best}} - X^i) \\
    X_{i+1} &= X^i + v_{i+1}
\end{align*}
\]

(5.3)

with $r_1, r_2$: random number $(0, 1)$
$c_0$: inertia factor $\lesssim 1$
$c_1, c_2$: learning factor $\approx 2$
$X^i$: solution of iteration step $i$
$v^i$: particle velocity in step $i$

In particle swarm optimizations, the velocity of each particle is changed at each iteration step. The flight direction is changed towards the particles best $X_{\text{particle best}}$ and the global best $X_{\text{global best}}$, weighted by separate
5.2 Numerical optimization algorithms

5.2.5 Evolutionary Algorithms

Its basic principles are adopted from biological evolution. Although evolutionary algorithms attracted attention not before the 1960s, several independent paradigms were developed and a lot of research has been done in this field. The paradigms differ in their reproduction mechanisms and the selection strategies, and were mainly Evolutionary Programming [35], Evolution Strategies [103] and Genetic Algorithms [49, 25]. However, the interaction between these groups was low until the 1990s, when the various approaches were unified under the term Evolutionary Computation [26].

There are several advantages of evolutionary optimization:

random numbers $r_1$ and $r_2$. Inertia keeps the particles flying in a similar direction in subsequent iteration steps.

An advantage of the particle swarm optimization is the low number of parameters to adjust. Most settings work well for a variety of problems. A typical range for the number of particles is 20 to 40. As the dimension of the design space directly corresponds to the number of optimization parameters, the particles fly in this n-dimensional design space. Side constraints defined by the general problem formulation (equation 5.1) can be specified for each parameter $x_i$. [51]
Evolutionary optimizations can find global optimum, and they have a certain robustness against local optima. Although the global optimum can be found, appropriate definition of the objective function and an appropriate parametrization help improve the convergence rate. Thus, good processing parameters can be found within reasonable (limited) time.

Evolutionary optimizations are robust against scattering which may for instance result from finite element discretization or random variation of Monte Carlo simulations.

Heterogeneous optimization parameters can be treated. For instance, discrete parameters such as a selected node for a gate location and continuous parameters such as the injection flow rate can be optimized in one go.

Implementation of evolutionary algorithms is rather simple, compared to mathematical programming.

Figure 5.5 illustrates the iterative optimization procedure of evolutionary algorithms. Starting with an arbitrary or predefined set of process parameters (individuals), simulations are run. Those parameters may be gate locations, vent locations and injection flow rates. The output of each simulation run is assessed by an objective function, giving a performance index, and rating the quality of the process parameter set on
which the simulation was based [53]. In Evolutionary Optimization, the performance index is commonly named fitness. According to the evaluated fitness values of the individuals (parameters), the new individuals of the next generation are defined, using recombination and mutation of parameters. Those new individuals are evaluated again, assessed and improved, until a sufficiently good solution is obtained.

For the representation of heterogeneous parameter types the “universal genotype” concept can be used [65]. Various parameter types (e.g. string lists, floating point variables, integer and boolean variables) can be packed into one set of genes.

5.3 Optimization parameters

The parametrization of the optimization problem may have a strong influence on the convergence rate of the optimization algorithm. Therefore, an appropriate parametrization has to be developed, contributing to successful optimizations. In general, small changes in the parameters should cause small changes in the configuration of the simulation [41]. This is particularly important for the fine tuning of the parameters [26].

The general optimization parameters for RTM fill simulations are:

- Gate and vent locations
- Injection pressures and flow rates of injection devices
- Timing of the injection devices
- Configuration of the injection system
- Mold temperature, if fast curing resin systems are used

Appropriate parametrizations of gates and vents have to be developed for connectors, which may be placed freely on the part, as well as for connectors at predefined locations. Additionally, the devices of the injection system have to be initialized with optimum values, and simultaneously the connections within the injection system have to be defined.

5.3.1 Parametrization of the connector locations

Locations of additionally placed gates and vents are defined by selecting single nodes. Problems resulting from this single node selection approach will be discussed and addressed in section 5.6. Each of the selected gate locations is linked to an injection device, and vents are applied on the
selected vent locations. Two different parametrizations to select a node were considered:

- A 3D point is used as optimization parameter, and the closest node to this point is used as a connector location. Finding the closest node to an arbitrary 3D point is numerically costly, because the distance between each node and this point has to be calculated. Additionally, slight changes in the parameters may flip the selected nodes from one surface to another, if the parametrization point is in between the two surfaces. Therefore, small parameter changes may cause big changes in the configuration. Points far from the part should be avoided, for instance by setting appropriate limits for each coordinate, as remote points could make it very unlikely to escape from a solution that requires to find other connection points.

- The node number can be used as the parameter. A limited number of integer values (i.e. the number of nodes in the model) can be chosen by the optimization algorithm. The small parameter space allows good efficiency. However, the node number is an unsorted parameter, so small changes in node number can cause relocation of mold connectors at points far from the previous location.

To avoid the disadvantage of the second parametrization resulting from the unsorted numbering, an additional stochastic operator can be introduced. Neighborship information, which is available via the finite element model, can be used to move the connector to a node located nearby. Starting from the original connector node, either one or several random steps to neighbor nodes can be made. Thus, local repositioning can be achieved. If desired, additional repositioning steps can be made by directly modifying the node number. This parametrization using node numbers and a local repositioning operator as well as direct modifications of the node number has been used throughout this work.

5.3.2 Line connectors and predefined locations

In the previous section, the gate and vent locations were defined by single points. Line connectors are often used due to their large effective flow cross section, but applying line connectors in simulations requires geometrical information about the part. Therefore, they are more difficult to apply in an automated optimization. For some applications, it may additionally be necessary to predetermine allowed connector locations to avoid connectors at functional or visible surfaces.
Possible connector lines and locations can be predefined during the modeling of the part. Those predefined connectors may be connected to injection devices or vents by the optimization algorithm, or they can be left unconnected. The number of used connectors can be limited to a maximum allowed number of additional connectors. Thus, line connectors can be placed in the mold, and the user can influence the locations of the connectors.

The number of the allowed number of connectors is defined by the software user as an input to the optimization. For each allowed connector, a set of variables is defined, which controls the configuration:

- Selection of the considered connector, selected by the identification numbers of the predefined connectors
- Use the connector as gate, as vent or not at all
- Number of injection device, to which the gate should be connected

In the finite element model, the connectors are modeled with tube elements. To allow an arbitrary number of line connectors without disturbing the simulation, the cross sections of the unused connectors have to be set to zero by the optimization.

5.3.3 Parametrization of the injection system

Improved economic processes take advantage of the optimum usage of the available injection devices. Modeling the injection system, as described in section 3.3, is particularly interesting for numerical process optimizations. In this approach, the connection among the injection devices and their connection to the mold can be included in the optimization as well as the parameters of the injection devices like switching times and flow rates or pressures.

The number and capabilities of available injection devices can be given as a fix side condition, e.g. two flow controlled pumps. Additionally, the number of injection points on the part can be defined by the process developer. In the optimization, the available devices can be connected in an optimized configuration in order to achieve the best possible process.

Because the capabilities of the available injection devices are defined before the optimization is started, the output can be directly implemented in the real process. Requirements, like individual control of each gate, are unrealistic for most industrial applications, and are thus avoided. Several gates can be controlled with a minimum number of injection devices.
The parametrization of the injection system developed and used in this work has been designed with the objective to avoid useless device configurations. The major issues were:

- Circular connections have to be avoided.
- Not all injection devices have to be used, but all may be used.
- Connection of additional gates, according to the given number
- Connection of additional vents, according to the given number

In the developed connection algorithm, the first step is to connect the valves to the injection devices and eventually to previously connected valves if several valves are used. Then, the gates are placed at the selected nodes and connected to one of the active injection devices, or to one of the valves. Finally, the vents are placed in the mold. Thus, circular connections cannot occur, not all injection devices have to be used, and the desired number of additional gates and vents is applied.

5.3.4 Conclusions

For free placed gate and vent locations, a parametrization based on the node numbers has been developed. Therefore, the design space can be limited to a minimum. To ensure the near repositioning of the connector locations in mutations, neighborship information can be included. In addition, a parametrization for the predefined connectors has been found, requiring only three parameters are per connector. Hence, they can now also be included in optimizations.

As simple parametrization was an objective already during the development of the injection system modeling, a straightforward definition of the connectivity within the injection system could be implemented.

The mold temperature can directly be implemented as a floating point value. Although the resin temperature may be important for highly reactive resin systems, its influence has not been investigated in this work.

5.4 Definition of optimization objective

Independent of the used optimization method, a suitable definition of the optimization target is essential. In general, it may be composed of weighted portions for several design objectives and several design constraints. According to the general definition of the optimization objective
in equation 5.1, design objectives can be included in the objective function $f(x) \rightarrow \min$. To account for violations of equality constraints $h_k(x) = 0$, and inequality constraints $g_i(x) \leq 0$, penalties can be added to $f(x)$ if the constraints are violated.

To apply numerical optimizations, the RTM specific optimization targets have to be identified. The thus defined sub-objectives have to be weighted because these sub-objectives are often competing or even contradicting. A smooth topology of the optimization function, ideally with a single optimum, facilitates fast convergence rates of the optimization.

### 5.4.1 Topology of objective function

Each considered simulation output value can be scaled by an appropriate function in order to make various different output values comparable. For instance, the functions shown in figure 5.6 can be used to scale design objectives (left) and parameter constraints (right). Additionally, an appropriate weight can be given to each partial objective to vary their relative importance.

The appropriate definition and weighting of the partial objective functions is the Achilles tendon of process optimization. For instance, the final fill degree of an injection should always be close to 100%. Emphasizing low fill times, which is often another important design objective, will on one hand lead to fast filling processes, but on the other hand, over-emphasizing short fill times may lead to only partially filled molds. Furthermore, short fill times can be achieved by processes with high flow front velocities, or configurations with large confluence zones. Both will result in rather poor matrix quality and might contradict objectives concerning the part
quality. Reducing the filling time is an important target, yet to enhance the quality of parts manufactured in RTM processes, it is important to properly balance process relevant aspects and quality aspects.

To allow fast convergence toward the desired optimum, the topology of the objective function should be smooth and, if possible, it should not contain local optima. An ideal definition of the fitness function would show a slight gradient towards the global optimum in the whole design space. This can be influenced by an appropriate definition of the objective function [26]. The final objective used for the optimization of the RTM fill process can be composed of several parameters, however, there may be only a single output parameter of actual interest, e.g manufacturing costs. Considering the manufacturing costs as the only output parameter in the fitness function will lead to a small but steep peaked optimum (figure 5.7, upper). In most areas of the design space, the fitness value is constantly high, and the optimization algorithm will not receive any information about directions of possible process improvements. Information about process improvements can only be obtained close to the optimum. An improved definition of the fitness function can be obtained by including partial objectives as it is shown in figure 5.7.
5.4 Definition of optimization objective

5.4.2 RTM specific optimization targets

All relevant simulation results that should be optimized have to be integrated in the objective function. For RTM injection processes, these are:

- process related e.g.
  - minimum fill time
  - limited injection pressure (stiffness of the mold, strength of the mold)

- quality related e.g.
  - low matrix porosity
  - complete filling of the cavity

- and cost related e.g.
  - minimum scrap rate
  - resin loss
  - total manufacturing costs

Disturbances of the injection process are unavoidable, and the quality of the part is easily affected by these disturbances. In the worst case, a poor laminate quality or even gas entrapments may occur, and scrap parts may result. Material costs and preparation effort for injections are major matters of expense for complex parts.

5.4.3 Maximum pressure constraint

Injections utilizing pressure controlled injection devices, the maximum pressure can be directly limited when setting the the injection devices. For volume flow controlled injections, it is more difficult to control the injection pressures. The injection pressure may rise above the allowed maximum pressure which is determined by the mechanical properties of the mold.

In optimizations with volume controlled injection devices, a penalty for pressures above the maximum allowed limit could be added to the objective function. Thus, processing parameters leading to excessive high pressures are rated poorly, and are therefore avoided. However, this approach is not optimal for the optimization efficiency and requires a well
balanced definition of the objective function. For instance, an improvement in the fill degree may be achieved due to improved placement of the gates. As the flow path may become longer, the injection pressures might rise above the allowed pressure. Then, the new process could be rated worse than the previous process, although the fill has been improved, but the injection flow rates were not adjusted yet.

It is quite obvious that processes parameters reaching pressures above the allowed level can be improved if the injection pressures and volume flows are reduced. For incompressible resins and stiff molds, reduced injection pressures will lead to longer fill times, but the flow pattern will remain similar. The maximum pressure can be reduced to the allowed pressure limit by scaling the injection flow rates and pressures with the factor $\frac{P_{\text{allowed}}}{P_{\text{simulated}}}$. Because the fill process with reduced injection rates takes longer, switching times of the injection devices have to be increased accordingly.

In the case of non-linearities between input parameters and simulation output, a second simulation run with adjusted injection parameters may become necessary. For instance, the formation of voids depends on the flow velocity, and an optimum injection velocity should be achieved for good matrix quality [36, 94]. Further examples for non-linearities are the time dependent curing of resin, which influences the resin viscosity and gas entrapments, and may be different for adjusted injection pressures due to the compressibility of the enclosed gas.

5.4.4 Robust processes with optimized scrap rate

To support the development of robust manufacturing processes, the reliability of injections with new processing parameters can be estimated using Monte Carlo simulations. In general, the scrap rate of RTM injection processes should be reduced. Based on Monte Carlo simulations using the parametric simulation model described in section 4.1, variations of the process can be considered within the process optimization. Relevant disturbances are variations in:

- Fiber volume content
- Principal permeabilities and their orientation
- Flow channels with varying cross sections. These may occur frequently and often lead to strong disturbances (cmp. section 4.1.3.

In section 4.2.3, a single valued quality criterion has been presented, which was developed to assess the part quality, summarizing all quality relevant
5.4 Definition of optimization objective

Figure 5.8: Schematic of Pareto optimization for fill time and scrap rate. The blue lines of constant costs are defined by the fixed costs per injection and the fill time dependent costs (eq. 5.4).

parameters. Thus, the scrap rate can estimated from the distribution of the $R$-criterion applying a quality limit value.

5.4.5 Optimization of manufacturing costs

The major target of process optimization is to developed a process which is capable to manufacture parts of the required quality at minimal costs. For robust processes, the scrap rate should be considered in the objective function. However, as a further and often contradicting objective, the fill time should be reduced. Therefore, a reasonable ratio between fast injections on the one hand and reliable injections on the other hand is required, which may be difficult to define.

Figure 5.8 shows a schematic example of possible process results (red). The Pareto front represents the solutions that can be achieved, and the decision between fast versus reliable processes is left to the process developer after the numerical optimization has finished.

However, Pareto optimizations require a much higher computational effort compared to single objective optimizations. To include both objectives in a single value objective, optimization of the manufacturing costs is advised. Thus, weighting between fast and reliable processes can be made on a comprehensible base.

A simplified cost calculation was developed, including the scrap rate and the fill time. The development costs for the process and tooling costs for the mold are assumed to be constant, independent of the quality of the
processing parameters, and are not considered here. The costs influenced by the process can be separated as follows below.

- Injection related costs depending on number of injections. They include raw material costs (fiber material costs, preforming costs, resin costs, core materials, consumables such as tubes or binders), preparation costs (application of release agent, lay-up of preform, connecting the devices), demolding costs (demolding time, mold cleaning) and energy costs.

Scrap parts cause the same costs as good parts, so the contribution of the injection related costs to the final costs are:

\[
\text{costs}_{\text{fix}} = \frac{\text{row material} + \text{preparation} + \text{demolding} + \text{energy}}{1 - \text{scrap rate}}
\]

- Fill time dependent costs.

Several costs are time dependent, for which a total time rate can be calculated. Those include the hourly rates for the injection devices, the tool carrier, the employees running the injection, rent for working place, etc. Fast curing resins with gel times that match the injection time can be used. Thus, the total cycle time can be calculated as a fixed multiple of the injection time:

\[
\text{costs}_{\text{time}} = \frac{\text{rate} \cdot \text{injection time} \cdot \text{curing time multiplier}}{1 - \text{scrap rate}}
\]

Thus, an equation for the processing costs of each produced good part can be formulated, which can be directly included in the objective function:

\[
\text{costs per part} = \frac{\text{costs}_{\text{fix}} + \text{costs}_{\text{time}}}{1 - \text{scrap rate}}
\] (5.4)

The lines of constant costs per good part are plotted in figure 5.8 for given time rates and fix costs per injection. The area of optimum solution depends on these two cost rates, and will move if different values are used.

### 5.4.6 Conclusions

RTM specific objectives have been summarized, and the scrap rate and the manufacturing costs have been introduced. A specific solution to limit the maximum injection pressure has been developed. Here, the maximum pressure is not included with a penalty factor, but instead the injection parameters are modified in an appropriate way.
In addition, the topology of the objective function has been optimized for fast convergence. It shows a slight slope towards (local) optima within the whole design space. Due to the straightforward implementation of manufacturing costs, a convenient weighting between various objectives, particularly between fast and reliable processes, becomes possible.

5.5 Number of simulation runs for scrap rate estimation

As discussed in section 4.2, Monte Carlo simulations can be used to estimate the scrap rate. However, as simulations of geometrical complex parts are time consuming, Monte Carlo simulations with a high number of simulation runs will cause a very high computational effort for the evaluation of each parameter set. Therefore, a good balance between sufficient reliability of the Monte Carlo estimations on the one hand and acceptable evaluation effort on the other hand has to be found.

Hierarchical optimization strategies using coarse simulation models for screening [56] or combined search strategies (e.g. involving neural networks and response surfaces [81]) become interesting. In order to keep the complexity level of the optimization manageable, none of those techniques has been used in this work. Instead, the number of Monte Carlo simulation runs is kept as low as possible by checking the level of reliability that is reached after each run within the Monte Carlo simulation. If a sufficiently high level of reliability is reached, the Monte Carlo simulation is stopped.

To assess the reliability of Monte Carlo estimations, bootstrap and sub-sample methods can be used [32]. However, the number of simulations required for one parameter set increases noticeably. Another possibility to assess the reliability of Monte Carlo results is the calculate the confidence intervals for the mean value and the variance. The confidence intervals give the range of values for a certain level of reliability (e.g. 95%), within which the true values lie.

For any distribution type, the arithmetic mean value \( \mu \) can be estimated by

\[
\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i
\]

and the variance \( \sigma^2 \) can be estimated by
\[ s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \] (5.6)

While it is difficult to give reliability levels and confidence intervals for general distributions, the confidence intervals can be given if the sample values follow the well investigated normal (Gauss) distribution. The confidence interval for the mean value is symmetric and can be calculated using the standard deviation \( s \) and the student distribution \( t \)

\[ \bar{x} - \frac{s \cdot t_{n-1,\alpha}}{\sqrt{n}} \leq \mu \leq \bar{x} + \frac{s \cdot t_{n-1,\alpha}}{\sqrt{n}} \] (5.7)

The confidence interval of the average value gets smaller with a lower standard deviation and with a higher number of samples. For the variance, the confidence can be calculated using the \( \chi^2 \)-distribution

\[ \frac{s^2 (n-1)}{\chi^2_{n-1,\alpha/2}} \leq \sigma^2 \leq \frac{s^2 (n-1)}{\chi^2_{n-1,1-\alpha/2}} \] (5.8)

Again, a higher number of samples reduces the width of the confidence interval. A variance \( \hat{\sigma}^2 \) can be calculated, which will be above the true variance with a 50% probability, and below the true variance for the other 50%. As the \( \chi^2 \)-distribution is asymmetric, the variance parameter \( \hat{\sigma}^2 \) is slightly higher than \( s^2 \) and can be calculated as

\[ \hat{\sigma}^2 \leq \frac{s^2 (n-1)}{\chi^2_{n-1,1/2}} \] (5.9)

Many practical distributions, particularly if they are bounded by zero, can be well approximated by a normal distribution after applying the logarithm function \[102\]. The distribution of the \( R \)-criterion (section 4.2.3, figure 4.31) can be well approximated by a log-normal distribution. Thus, the confidence intervals for estimated parameters of the distribution can be given.

Considering the logarithmic \( R \)-criterion, the separation line between good parts and scrap parts is at \( \ln(R) = 0 \). The \( R \)-distribution can be shifted to the confidence interval limits (figure 5.9), calculated according to the equations 5.5 to 5.8 for a given confidence level. This results in four distributions for the limit values, plus the distribution defined by the most probable values of \( \bar{x} \) and \( \sigma^2 \).

The scrap rate can thus be calculated for all five combinations of mean and variance values, resulting in five different values. For increasing number of samples, the confidence intervals gets narrower and the maximum
5.5 Number of simulation runs for scrap rate estimation

Figure 5.9: Distribution of \( \mathcal{R} \)-criterion shifted to the limits of the confidence intervals for mean value and variance

Figure 5.10: Distribution of \( \mathcal{R} \)-criterion shifted to the limits of the confidence intervals for mean value and variance

difference between the five calculated scrap rates becomes smaller, and the scrap rate tends asymptotically to the true value (figure 5.10).

In the implementation used for this work, the Monte Carlo simulation within the optimization loop is run until a predefined difference of 20\% between minimal and maximal calculated scrap rate is reached.

\[
\text{stop} = \left[ \max (\text{cdf}(p_R)) - \min (\text{cdf}(p_R)) < 20\% \right] \tag{5.10}
\]

The scrap rate is then calculated using the most probable \( R \)-distribution, which is defined by \( \bar{x} \) (eq. 5.5) and \( \hat{\sigma}^2 \) (eq. 5.9).

For injection parameters which are evaluated with a quality of \( \mathcal{R} > 10 \) in the first simulation run, the Monte Carlo simulation is stopped. In these cases, the quality is far outside the requested quality range, and a scrap
rate of 100% is assumed without further investigation of the statistical reliability. To avoid very long simulations of a single individual, the Monte Carlo simulations will be stopped additionally, if a predefined maximum number of simulation runs is reached.

The proposed method to estimate the reliability level of the calculated scrap rate allows statistically founded statements. It may be used to reduce the computational effort during an optimization. However, as a high statistical reliability requires large samples, the scrap rate calculated within an optimization may differ from the value calculated in a Monte Carlo simulation with several hundred simulation runs.

5.6 Local remeshing around gate and vent nodes

During the optimization of injection processes, additional gates and vents (mold connectors) may be placed. As described in section 5.3.1, gate and vent locations are defined by single nodes.

The effective size of the connector depends on the size of the elements and the control volumes connected to the selected node (for control volume definition see figure 2.3). Calculated fill times and injection pressures are affected by the effective size of the connector. Particularly for multi-gate-injections, confluences of the flow fronts may be wrongly predicted.

Local remeshing of the additional gates and vents ensures a similar effectiveness of all additionally placed gates. Thus, the element sizes and control volumes are independent of the original element sizes.

5.6.1 Single node connectors in process optimization

For reliable optimization results, it is important to model the additionally placed mold connectors with identical properties, independent of the mesh size. Otherwise optimization of parameters such as valve timing or gate pressures could provide incorrect values, causing the flow front to converge at different locations than predicted. Dry spots or poor laminate quality might be caused in consequence.

The influence of the gate definition is illustrated in figure 5.11. A rectangular plate (size $1 \times 1$) has been meshed with constant element size (0.03) and elements varying in size between upper and lower side by a factor 10. Two gates with constant pressure are applied in the middle of the upper and the lower edge, a vent is placed at the middle of the right edge.
### Table 5.11: Influence of gate modeling and local element sizes at gates on flow pattern and fill time

<table>
<thead>
<tr>
<th>Element Size</th>
<th>Single node gates</th>
<th>Multi node gates with 7 to 27 nodes per gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>0.01...0.1</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>0.03</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>0.01...0.1</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 5.11: Influence of gate modeling and local element sizes at gates on flow pattern and fill time. The rectangular panel (size $1 \times 1$) is meshed with either a constant typical element length of 0.03, or varying from 0.01 to 0.1. One vent is applied at the middle of the right edge. For the multi node gates in the lower two rows, the gate diameter is $D = 0.1$. 
Figure 5.12: Influence of gate modeling and local element sizes at gates on flow pattern and fill time. The rectangular plate (size $1 \times 1$) is meshed with either constant a typical element length of 0.03, or varying from 0.01 to 0.1. For the multi node gates in the lower two rows, the gate diameter is $D = 0.1$.

In the upper two rows it can be seen that the flow pattern and the fill time for the single node gates depend on the element size. The fill time differs by about 10%. Although the flow should be symmetric, the simulation with varying element sizes shows an asymmetrically advancing flow front, and a dry spot may appear near the vent (figure 5.12). In contrast, the effect on the fill pattern and fill time is negligible if the gates are meshed with several nodes.

Often meshes are refined at edges or geometric joint lines in order to better resolve the local flow. Thus, connectors get a lower effective size and therefore the pressure/flux ratio decreases compared to a node in a region with large elements. The problem can be reduced by selecting several connected nodes, but still the local element size affects the calculated pressure distribution [74, 87].

Figure 5.13 shows an example of the different pressure distribution of a radial injection, for large and small elements. The flow velocity, which is proportional to the pressure gradient, differs between both modelings.

Analog to gates, vents need to be remeshed as well. Low effective vents sizes, e.g., lead to high pressure concentrations around the vent, which is unrealistic and can influence the filling time significantly. If vents are closed according to the resin volume that has spilled out, the time until a defined resin volume has escaped through the vent is influenced as well.
5.6 Local remeshing around gate and vent nodes

5.6.2 Modeling of mold connectors

Whenever another node is chosen to be used as gate (or vent), apart from the location, its effective size may be altered, too, depending on the local mesh size. In a simulation based evaluation of a selected set of process parameters the maximum pressure and the fill time are affected by the problem described above. In order to eliminate this error, local remeshing is suggested. Thus, similar effective connector sizes, independent of the local element sizes, are obtained for all additional connectors placed by the optimization algorithm.

As the finite element modeling is a (in this case linear) piecewise approximation of the real pressure distribution within the resin filled area, the length of the used elements is also important. Very long elements cannot map the actual pressure distribution, for instance. So, an additional layer of elements of defined size is created around the mold connector.

5.6.3 Applying connector remesh in simulations

Example of a locally remeshed grid Figure 5.14 shows a close view to a remeshed connector. As the local element size at the remeshing location was small compared to the remeshing diameter, many original elements were cut. This resulted in a lot of new nodes and therefore,
Optimization of processing parameters

Figure 5.14: Detail view of remeshing example of a surface in a originally rather fine meshed area

Figure 5.15: Remeshing example of a simple part with 2 gates and one vent

in a good approximation, to a circular connector. Radial tube elements connect the center node to the sphere elements.

Figure 5.15 shows a further remeshing example. Here, the local element size is similar to the remeshing diameter. The number of elements cut when the connector sphere are remeshed is considerably lower compared to the example shown in figure 5.14. Nevertheless, the connector shape is still close to a circular connector.

Radial injection and analytical solution  Injections of a circular plate with an outer diameter of $D = 0.4$ have been simulated. A line vent is placed all around the outer edge, and an injection gate with a diameter of $d = 0.01$ has been defined in the middle of the part. For isotropic
preform materials an analytical solution of the fill process is available, including pressures, flow velocities and flow front positions. This can be compared to results of finite element simulations.

Two different fill processes are discussed, one at constant injection flow rate \( (q = 10^{-7}) \) and the other at constant injection pressure \( (p = 10^5) \). For injections at constant flow rate, the injection pressure at the end of the simulation can be taken for the comparison of the analytical and the numerical solution. Injections at constant pressure are compared with respect to the fill time.

Finite element simulations with two different series of remeshing diameters were tested. The first series investigates the influence of the element length of the inner element row. Therefore the inner radius is kept constant \( d_i = d_{\text{nominal}} \) and the outer diameter scaling factor varies between \([1.2, 2]\). In the second series the middle diameter \( d_m = \frac{d_i + d_o}{2} \) is constantly \( d_m = d_{\text{nominal}} \), keeping the effective interface area between the control volume of gate nodes and non gate nodes constant.

Figure 5.16 shows the resulting injection pressures. The analytical solution is represented by the red line, the numerical simulation results are shown as single points. For a constant inner diameter \( (d_i = d_{\text{nominal}}) \) and varying outer diameters, the calculated injection pressure is slightly \((\approx 3\%)\) too low, compared to the analytical solution. The reduced pressure may result from the increased surface area of the control volume, allowing the same flux at lower pressure gradients.

Additionally, figure 5.16 shows the calculated injection pressures for different inner and outer diameters with constant middle diameter \( d_m = d_{\text{nominal}} \). Modeling the injection gate with small (short) elements and approximating the control volume surface to the nominal connector diameter \( (d_m = d_{\text{nominal}}) \), small differences between analytical and numerical pressures result \((\approx 0.7\%)\). However, increasing the outer diameter and therefore decreasing the inner diameter in this investigation series, results in significant errors for the simulated values for \( d_o/d_i \gtrsim 1.2 \).

Similar results were obtained for injections at constant pressure. Figure 5.17 shows the calculated fill times for this case. Again, good approximation \((\approx 2\%)\) can be obtained for inner diameters \( d_i = d_{\text{nominal}} \) and short (small) elements at the connector. Fill times for modeled connectors with \( d_m = d_{\text{nominal}} \) also show good accordance for short elements in the inner row \((l_{\text{ele}} \approx 0.1 \cdot d_{\text{nominal}})\). Longer (bigger) elements lead to modeling errors of significantly increased filling times if \( d_o/d_i > 1.2 \).

Two simulations without remeshing the connector, but different mesh size were compared. Figure 5.18 shows the results for two different models with element sizes of 0.002 and 0.015, resp. The necessity of local remesh-
Figure 5.16: Influence of ratio between outer and inner remeshing diameter on maximum pressure of a radial injection with constant injection flow rate. Two results for constant inner diameter $d_i = d_{\text{nominal}}$ and for constant middle diameter $d_m = d_{\text{nominal}}$ are presented.

Figure 5.17: Influence of inner and outer remeshing diameter on fill time of a radial injection with constant injection pressure. Inner diameter is kept constant $d_i = d_{\text{nominal}}$
5.6 Local remeshing around gate and vent nodes

Figure 5.18: Influence of element size on fill time (upper graph) and on maximum injection pressure (lower graph) without local remeshing of the connector.

This investigation shows the influence of the diameter of the connector sphere as well as the influence of the element size of the inner layer of elements at a circular connector. As the results show, small (short in radial direction) elements compared to the connector diameter should be used for the inner row. The inner diameter $d_i$ can be chosen as $d_{nominal}$ resulting in small errors of approx. 2%. If elements of a radial length up
to $0.1 \cdot d_{\text{nominal}}$ are used in the inner row, results may be slightly improved by selecting the middle diameter $d_{M} = d_{\text{nominal}}$.

**Example for influence on simulation times** In this example, simulated filling times of a flat panel with a cut-out are compared. Its outer dimensions are $100 \times 100$ mm, the element size is 3.5 mm and the nominal connector diameter is 5 mm. Two connector tubes are used to connect the part to a pressure pot and a line vent is placed on the right side of the part.

Figure 5.19 shows the final pressure distribution in sample part at the end of the simulation. Differences can be identified in the area close to the gates. The simulation with locally remeshed connectors results in a filling time of 35.4 s while the fill time of the not remeshed part is 39.6 s. This is a $+12\%$ difference for the used dimensions. Simulations with different element sizes also show the importance of local connector remeshing.

### 5.6.4 Conclusions

A method to locally remesh a finite element model has been developed. No further information about the geometry is required. In the proposed method, the inner element layer is re-build to achieve better simulation results.

The influence of the inner and the outer remeshing diameter on the accuracy of the simulation has been investigated using flat parts. Comparing the simulation to an analytical solution for a radial injection, it
has been found that the inner diameter should equal the theoretical diameter in order to minimize the influence of the outer diameter on the simulation. Then, the outer diameter may be chosen as twice the size of the inner diameter.

5.7 Evolutionary optimization of process parameters

Several optimization problems have been investigate to test and demonstrate the developed methods. Evolutionary optimization algorithms have been chosen, as these algorithms are robust against local optima. Furthermore, they are able to treat heterogeneous sets of optimization parameters, consisting of integer and continuous variables, simultaneously. Numerical noise, resulting form the discretization of the finite element models, does not have a significant influence on the convergence of evolutionary algorithms. Hence, noise resulting form the evaluation with stochastically initialized disturbances should not cause any problems as well.

Starting with two simple geometries, a symmetric bar with a locally increased fiber volume content (figure 5.20) and a rectangular panel with a fixed vent (figure 5.23), the optimization of injection points and the parameters of the injection devices have been verified. The used geometries were chosen due to their intuitive accessibility with the intention to find a good solution in a heuristic approach, and to enable an easy interpretation of the optimization results. Thus, the parametrization and the definition of the fitness function can be tested and improved.

Besides the simple geometries to develop and test the optimization algorithms, more complex parts were investigated as well. To develop and verify the positioning of gates and vents on predefined connector locations, the filling of a rectangular plate and a boat hull were investigated (section 5.7.3). An optimized injection strategy for the symmetric bar has been sought in section 5.7.4. Furthermore, the optimized injection strategy is compared to a sequential injection, similar to the heuristically optimized injection strategy suggested for the hockey stick (section 5.1). In section 5.7.5 the optimized process for a complex shell component is be presented. Finally the process for the demonstrator part of the VANTEX project (“Vanishing Textile binder structure for the stabilization of dry carbon fiber preforms in the Resin Transfer Molding process.” CTI-No. 8634.1) has been optimized. Results are shown in section 5.7.6.

Two different types of optimizations were considered:
- Deterministic optimizations, evaluating each parameter set in a single simulation. Eventually applied disturbance zones in the model are initialized with 0.5. This corresponds to an average disturbance (section 4.1).

- Robust optimizations, using a Monte Carlo simulation with several simulation runs to estimate the process results and variations for each parameter set. The disturbance zones are stochastically initialized using a Latin Hypercube scheme (section 4.2).

The results of robust optimizations will be compared to the results of deterministic optimizations.

### 5.7.1 Definition of the objective function

The main objective is the reduction of the manufacturing costs. As described in section 5.4.5, the manufacturing costs of a part can be calculated from the scrap rate and the injection time. Then, an objective value calculated from a linear relation with 0 for zero costs and 1 for the estimated initial cost is used. This estimate is a required input to the optimization and may for example be the manufacturing costs assuming a high scrap rate and a realistic fill time.

To accelerate finding the areas of interesting process parameters within the design space, a twofold definition of the objective function has been used. As illustrated in figure 5.7, the costs are constant over a large area of the design space, specifically these are constant for all parameters leading to a scrap rate of 1. Therefore, besides the manufacturing costs the $R$-criterion is included with a linear influence

$$ f_R = w_R \cdot \frac{R}{R_{\text{max}}} \quad \text{with} \quad R_{\text{max}} = \frac{1}{1 - \text{Fill}_{\text{min}}} \quad (5.11) $$

As defined in section 4.2.3, $\text{Fill}_{\text{min}}$ is the required minimum fill degree for good parts. The weight factor $w_R$ should usually be low compared to the weight of the cost influence. These sub-objectives result in the final objective function with weighted sub-objectives

$$ f = w_R \cdot \frac{R}{R_{\text{max}}} + w_{\text{Cost}} \cdot \frac{\text{Cost}}{\text{Cost}_{\text{init}}} \quad (5.12) $$

### 5.7.2 Optimization of connector position

From two simple test geometries, a symmetric bar and a rectangular panel, three different cases have been derived to test the positioning of gates.
5.7 Evolutionary optimization of process parameters

Figure 5.20: Geometry and fiber volume content of symmetric bar. The bar has a dimension of 1 x 0.1 m

All three test problems were optimized with deterministic models, where permeability variations were not included.

**Symmetric bar** The first test presented is the optimization of a symmetric bar (figure 5.20). The bar has a fiber volume content of 40%. In the middle a section is included, which has an increased fiber volume content of 50%. A line vent is applied on the right edge, which closes after 2% of the total resin volume of the part has been spilled out. No fixed gate has been placed on the part, but one gate can be placed by the optimization anywhere on the part. This gate will be connected to a injection device which provides a constant pressure.

In figure 5.21, the resulting optimization history is plotted. A population of 20 individuals has been used. The best found position of the connector is shown in figure 5.22. The found connector position is not on the left edge as it could be expected. Instead, the connector is placed slightly towards the vent and not on the center line. As the resin is injected with constant pressure, a slightly faster fill process is achieved, compared to an injection on the left edge.

A small unfilled area remains in the lower left corner after the injection is stopped. Due to the used definition of the scrap limit of 99.9% for the fill fraction, the optimization considered the part to be of sufficiently good quality.
Figure 5.21: Optimization history of fill degree, fill time and part costs for the connector position on the symmetric bar.
Figure 5.22: Connector position and resulting process for the symmetric bar. Pictured are an intermediate fill state, the final pressure distribution and the arrival times of the flow front.
Rectangular panel with fixed vent  In this test case, the positions of two connectors on a rectangular panel with constant thickness and constant fiber volume content are optimized (figure 5.23). The plate has a size of 0.3x0.3 m. In the lower right corner, a vent is applied on an arc with a radius of 0.01 m.

As in the previous optimization, no permeability variations are considered, but two additional gates can be placed by the optimization. Two optimizations were made, differing in the used injection devices. In the first optimization an injection device with constant injection pressure was used, in the second an injection device with variable flow rate.

Figure 5.24 shows the optimization history for the optimization with the pressure controlled injection device. After 500 generations, the differences between the five optimization runs is still noticeable, although the improvement rate is low. It can be seen that the fill degree rises above the required limit value of 99.9% in the beginning of the optimization. Then, the fill degree lowers to values close to the defined scrap limit value of 99.9%, allowing faster fill processes and therefore reduced manufacturing costs. Both optimizations resulted in a similar configuration of the gate, leading to an asymmetric fill pattern in both cases (figure 5.25).
Figure 5.24: Optimization history of fill degree, fill time and part costs for free gate positions on rectangular panel.
Figure 5.25: Optimized process resulting for two gates, freely positioned on the part. One injection device with constant pressure (left column) and one volume flow controlled (right column) have been used. Pictured are an intermediate fill state, the final pressure field and the fill times.
5.7 Evolutionary optimization of process parameters

Predefined connector positions

Predefined connectors only affect the simulation model if the connectors are actually used. Unused connectors are reduced to a cross section of zero, and therefore neither influence the volume of the component nor the resin flow.

**Rectangular panel** Figure 5.26 shows the geometry, the mesh and the predefined connectors of this optimization problem. The vent is fixed at the lower left corner of the panel. One injection device providing a constant volume flow may be used, as well as two out of the twelve predefined connectors. The optimization has to decide for each predefined connector whether to use it as a vent or to connect it to an injection device.

Figure 5.27 shows the optimized injection process. According to the geometry and position, different predefined connectors may have a different effectiveness on the resin flow. Apparently, the effectiveness of the predefined connectors is higher than the effectiveness of the fixed vent. Therefore, one connector is used as an additional vent which allows faster processes.

**Boat like shell structure** The possibility of predefined connector locations has been used to determine an optimized injection system for a boat like shell structure. Because of the symmetrical geometry (figure 5.28), a half model has been used in this optimization.
Figure 5.27: Optimized process using two of the twelve predefined connectors shown in figure 5.26. Here, the optimization linked one connector to the injection device, while the second connector is used as vent.
A fixed flow channel was defined on the symmetry line of the structure. On the half model, 46 additional flow channels were pre-defined, which could be activated by the optimization. A limit of five additional flow channels was set as an input parameter.

As the predefined flow channels are parallel aligned, all possible configurations of the injection system will result in completely filled parts. Thus, the costs can only be optimized by reducing the fill time of the boat hull. Figure 5.29 shows the optimization history for the fill time. The fill time of the best solution found is plotted for five optimization runs. Although 30 individuals have been used per generation, the five optimization runs did not reach the same result within the 200 generations. Nevertheless, similar configurations have been found. The optimized injection system for the best found solution and the resulting fill pattern are shown in figure 5.30. Yellow lines represent possible connectors, green are the actually used connector lines. The used flow channels form a nearly symmetric distribution system (perspective distorted in the illustration).
Figure 5.29: Optimization history of the fill time for a molding process of a boat hull.

Figure 5.30: Molding of a boat hull. Injection system obtained form the optimization (upper picture) and resulting fill pattern illustrated by flow front arrival times (lower picture).
5.7 Evolutionary optimization of process parameters

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 5.31: Definition of 10 disturbance zones for variations of the fiber volume content (±5%).

5.7.4 Optimization of configuration

An optimization of the configuration of the injection system has been done for the symmetric bar, that has been presented previously. The geometry of the bar is shown in figure 5.20, together with the spatial distribution of the local fiber volume content. To account for process variations, ten zones with variations of the fiber volume content were defined with a standard deviation of ±5% (figure 5.31). Flow channels have not been assumed.

In contrast to the previous process optimization for this part, where the position of a single gate had to be optimized, the focus is now on the configuration of the injection system. The injection system consists of:

- 2 freely placed vents
- 3 freely placed gates
- 1 Injection device with constant pressure
- 1 injection valve, which opens at a certain time, defined by the optimization

Three optimizations have been made and their results are compared here.

**Heuristic optimization** A sequential injection was developed in the heuristic process optimization. The injection is started at the center gate (figure 5.32, 1). Both lateral gates (figure 5.32, 2) are connected to the valve, which is opened when the resin front reaches the lateral gates.

Figure 5.32 shows the averaged final fill state. Only very small and rare gas entrapments occur on the left and the right edge of the part. As the anthill plot in figure 5.32 shows, the process is very reliable, and a very high part quality can be expected. These variations of the fill degree and the laminate quality have been estimated by a Monte Carlo simulation with 500 simulation runs.
Figure 5.32: Averaged final fill state and anthill plot of quality criteria resulting from sequential injection process. A scrap rate of 0% will result, and the average fill time will be 1886 s.
Figure 5.33: Averaged final fill state and anthill plot of quality criteria for deterministically optimized process. As the Monte Carlo simulation shows, the injection will be very fast (337 s), but a very high scrap rate has to be expected (81%).

**Deterministic optimization** Figure 5.33 shows the results of a deterministic optimization. Two gates (1) are directly connected to the injection device, and the third gate is connected to the valve (2). However, the valve does not open during the injection, and due to the comparably large volume of the connection tube, the third gate is used to buffer entrapped air. Thus, it is used more as a vent than as a gate. The two vents, which are available for the optimization, are located on the left edge and close to the right edge. This is similar to the vent locations used in the heuristic optimized process.

In the deterministic simulation, the flow fronts of the resin injected on the left and on the right gate join at a vertical line, which crosses the third gate. Thus, the entrapped air may be buffered in the connection tube, and no air remains in the middle section of the part. Obviously, the suggested process is fast compared to the heuristic process, but is not robust. An estimated probability of 70% for dry spots on the center line was found by Monte Carlo simulation.
Figure 5.34: Averaged final fill state and anthill plot of quality criteria for optimized process considering robustness. A scrap rate of 1% was calculated, as well as an average fill time of 600 s.

**Robust optimization**  The third process optimization presented includes variations of the fiber volume content. In the best found process, all three gates are located in the center region of the part (figure 5.34). The injection valve is not used, and therefore the resin is injected through all three gates directly at the beginning of the injection. The two vents are located close to the part’s left and right edge, respectively. The gates are placed on slightly asymmetric locations, resulting in a slightly asymmetric fill process.

The Monte Carlo estimation for the process reliability shows a rather stable process with a scrap rate close to zero (figure 5.34). Small gas entrapments have to be expected at the lower edge in center area of the part, as well as at the parts edge close to the vents. Compared to the heuristically optimized process, the fill process could be accelerated by a factor of three.

**Comparison of fill times**  Figure 5.35 summarizes the average fill time of the three optimized processes. The deterministically optimized process is much faster than the heuristic process and still twice as fast as the robust
5.7 Evolutionary optimization of process parameters

Figure 5.35: Averaged fill time for the three process variants of the symmetric bar

Optimized process. However, the standard deviation of the fill time is very high, and, as it is shown above, the process leads to a very high scrap rate.

Heuristic optimizations may benefit from the symmetry, as the effective problem size is reduced. Therefore, good results can be expected from heuristic optimization. Nevertheless the optimum result (with respect to the defined fitness function) is not necessarily a symmetric process, particularly if robustness is considered.
5.7.5 Optimization of sort box

For the sort box presented in section 4.2, optimized injection processes are discussed here. Compared to the previous investigation, the model has been modified. According to experimental results of Klauser [64], the fiber volume content, variations of the fiber volume content and the shell thickness have been adapted. Figure 5.36 shows the updated values.

Klauser [64] also investigated flow channel, as they have a strong influence on the resin flow. Figure 5.37 shows the dimensions and the ranges of the flow channels derived from his experimental results. The results of the simulation for the configuration used in the experimental investigation are summarized in figure 5.38. Note, that due to the different modeling, the simulation results presented here differ from the results presented in section 4.2.

Optimizations In the optimization of the sort box, no fixed vents or gates were used. The optimization could freely place one gate and one vent. Resin was injected by an injection device for constant pressure. Four optimization variants have been investigated:

- Optimization of the fill degree
  - deterministic optimization
  - robust optimization

- Optimization of the fill degree and the laminate quality
  - deterministic optimization
  - robust optimization

All optimizations used a population of 30 individuals, evolving over 500 generations.

Optimization of the fill degree The injection process was optimized to achieve a fill degree of 99.5% or higher, while the Laminate Quality criterion has not been considered within the target function of the optimization.

Figure 5.39 shows the resulting process for the deterministic optimization. Although the fill simulations of the optimized process shows a high fill degree of 99.8% if the disturbances are initialized with 0.5, the Monte Carlo simulation predicts several small spots with a high probability of remaining gas entrapments. As the anthill plot and the class plot of the
Figure 5.36: Geometry of the generic part “sort box”. Shown are the shell thickness, the fiber volume content and the considered variation of the fiber volume content.
Figure 5.37: Flow channel dimensions, derived from experimental investigations [64]

$\mathcal{R}$-values show, the process variations due to disturbances are high. A relatively unstable process with a scrap rate of 26% has been found.

With robust optimization, the scrap rate could be reduced to 7%. The injection gate and the vent location are similar to the locations of the deterministically optimized process, but slightly shifted (figure 5.40). In the Monte Carlo estimation of the process variations, beside the lower scrap rate, a lower variation of the $\mathcal{R}$-values has been found.

Optimization of the fill degree and laminate quality

Aimed was a process with a laminate quality shifted to better (lower) values. Therefore, the limit for the accepted laminate quality was set to $LQ = 18\%$. The resulting point clouds in the anthill plots of the figures 5.39 and 5.40 should therefore be shifted above the dashed line.

Again, similar to the previous optimization without consideration of the Laminate Quality criterion, the variation of the $\mathcal{R}$-values is rather high (figure 5.41). Even though the simulation of the obtained process with the model disturbances initialized with 0.5 leads to the desired fill and laminate quality values above the dashed line in the anthill plot, the process is not reliable and leads to a scrap rate of 70%.

The robust optimized process reduced the process variations, but the resulting process was similar to the robust optimized process without consideration of the laminate quality criterion (figure 5.42). This process is slightly better than the previous robust optimized process, but the resulting scrap rate of 31% is rather high. Compared to the other four investigated processes, the variation of the $\mathcal{R}$-values is rather low.
Figure 5.38: Results of Monte Carlo simulation for realized injection process. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $\mathcal{R}$-criterion.
Figure 5.39: Results of Monte Carlo simulation for deterministically optimized process. Shown are the averaged final fill state, the distribution of fill and laminate quality, and a class diagram of the $R$-values.
Figure 5.40: Results of Monte Carlo simulation for optimized process considering process variations. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $R$-values.
Figure 5.41: Results of Monte Carlo simulation for deterministic optimized process. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $\mathcal{R}$-values.
Figure 5.42: Results of Monte Carlo simulation for optimized process considering process variations. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $R$-values.
Optimization of processing parameters

Figure 5.43: Summary of all four optimized processes and the heuristically designed process.

Figure 5.44: Averaged fill time for the five process variants of the sort box

Summary and discussion of the optimized processes  All five optimized process results are summarized in figure 5.43. As plotting all points in the diagram would lead to a cluttered illustration, the average values and an ellipse representing the standard deviations are shown. One result of the numerical optimizations is the significant improvement of fill degree and laminate quality compared to the heuristic process. In addition, figure 5.44 shows that all numerically optimized processes are considerably faster than the heuristic one.

A second result is the clear difference between deterministically and robust optimized processes. The process variations of processes optimized
considering robustness are significantly lower and the average fill degree is higher. On the other side, robust optimizations lead to longer fill times.

Concerning the optimizations considering a laminate quality, the process objective could not be achieved. All four numerically optimized processes are similar, and it may be assumed that there is no substantially better process. To achieve the desired part quality, the input parameters of the optimization should be modified, for instance two vents could be used in order to obtain better results.
Figure 5.45: Geometry of demonstrator part. Shown are the gate and vent locations, as they were defined in a heuristic process optimization.

5.7.6 Demonstrator part of VANTEX project

Within the VANTEX project (CTI-No. 8634.1), a helicopter force introduction flange, which is currently machined of aluminum, has been redesigned to prove the applicability of the materials and processing parameters developed in the VANTEX project. The demonstrator is manufactured of carbon fabric with a quasi-isotropic layup. Figure 5.45 shows the geometry of the demonstrator part, with the approximate dimension of the base plate of 180 x 240 mm with a rib of 70 mm height. The shell thicknesses and the local fiber volume contents are shown in figure 5.46.

Variations of the permeability model used in the Monte Carlo simulations are illustrated in figure 5.47. Those variations include flow channels and variations of the fiber volume content.

Heuristically optimized process In a heuristic optimization, two gates have been placed on the ribs, and one central vent on the edge of the base plate (figure 5.45). The variation of the injection process has been assessed by a Monte Carlo Simulation with 500 runs. The resulting anthill plot for the fill degree and the laminate quality is shown in fig-
Figure 5.46: Fiber volume content and shell thickness of the VANTEK demonstrator part during the injection.
Figure 5.47: Considered permeability variations, including flow channels and variations of the fiber volume content.
Figure 5.48: Anthill plot of laminate quality vs. final fill degree for heuristically optimized process.

Figure 5.49 shows the fill degree plotted as a class diagram, clarifying the high variation of the fill degree.

Figure 5.50 shows the average fill degree. Problem areas can be identified, as for instance the area between the two ribs, where gas is entrapped with a high probability. The lateral problem areas result from the strong flow channels along the top of the two ribs. Dry spots in this area mainly result from the single central vent, which is reached by the resin before the lateral areas of the base plate are filled.

**Improved ventilation** To account for lateral problem areas, possibilities for an improved ventilation are investigated. Restriction due to the tool design limits the possibilities to reposition the vent. New vents may only be placed on the outer edge of the components base plate. 15 line segments shown in figure 5.51 were defined as possible vent locations.

In the following, two optimizations, one deterministic and one robust optimization, are presented and discussed. The fill degree has been considered as the only quality relevant optimization target which should reach values above 99.9%, the laminate quality has not been considered. The results of the deterministic optimization are summarized in figure 5.52. A significantly better fill degree can be achieved by the optimized process,
178 Optimization of processing parameters

Figure 5.49: Distribution of the final fill degree for the heuristically optimized process, obtained from 500 simulation runs.

the lateral problem areas of the ground plate can be filled. However, a non-optimum filling of the lower edge has been found.

At this point, the process found by robust optimization is slightly superior (figure 5.53). The vent on the lower edge is moved closer to this problem area, leading to a similar process, but with slightly higher fill degrees.

Asymmetric injection process In the previously optimized process, a gas entrapment may occur in the area between the two ribs. This entrapment is formed when the flow front reaches the ground plate simultaneously. Therefore, as the gate locations should not be moved, an asymmetric injection was investigated. Because both connectors of the gates are separately accessible, asymmetric injections can be realized without any modifications of the tool. Again, deterministic and robust optimization were made and the results are compared.

Figure 5.54 shows the variations of the resulting process. The injection starts at the upper gate in figure 5.54. After 5.5 s, which corresponds to 25% of the average fill time, the injection valve is opened, and resin is also injected at the lower gate.
Figure 5.50: Averaged fill degree of 500 simulation runs.
Figure 5.51: Allowed connector positions which may be used by the optimization.

The obtained process can reduce the problem of entrapped air between the two ribs, but it does completely avoid entrapments. Additionally, the location of the lower vent is not suitably selected. Compared to the symmetric processes, this asymmetric, deterministically optimized process leads to worse fill degrees.

The robust process was optimized with the same input parameters. The resulting process is shown in figure 5.55, together with the process variations. Similar to the deterministic optimization, the injection starts at the upper gate and the lower gate is activated delayed. However, the opening time of the injection valve is 22.4707 s (\(\approx 67\%\) of the average fill time), which is late compared to the deterministically optimized process. The resulting fill degree is high, and the variations are much lower. The determined scrap rate for parts with a fill degree below 99.9\% is 16\%.

**Summary**  
Figure 5.56 summarizes the variations of fill degree and laminate quality for the investigated processes. The fill degree can be increased significantly, when lateral vents are used. The process with asynchronous injection, which was optimized considering robustness, provides significant improvement to process quality and reliability.
Figure 5.52: Variation of the resulting process for deterministic optimization with symmetric injection conditions. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $\mathcal{R}$-values.
Figure 5.53: Variation of the resulting process for robust optimization with symmetric injection conditions. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $R$-values.
Figure 5.54: Variation of the resulting process for deterministic optimization with asymmetric injection conditions. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $R$-values.
Figure 5.55: Variation of the resulting process for robust optimization with asymmetric injection conditions. Shown are the averaged final fill state, the distribution of fill and laminate quality, and the distribution of the $\mathcal{R}$-values.
Fill time differences between the investigated processes are small, which may be because of the few possibilities to modify the process (figure 5.57). A longer fill time is obtained for the asymmetric, robust optimized process. This is similar to the results of the sort box, where robust processes required longer fill times than the deterministic optimized processes.

Although no significantly faster injection processes could be found, better fill degrees may be expected from the optimized processes, particularly if asymmetric injections are used.

### 5.7.7 Conclusions

Basic functioning of the developed methods could be shown in the optimizations of the connector positions. A fast convergence has been achieved, and the connector positions were reasonably selected. For the symmetric bar, the gate has not been placed on the left edge, as it might be expected. A faster injection was achieved when the gate is placed beside the symmetry line and slightly towards the vent. The optimizations of the rectangular panel also resulted in asymmetric injection processes.
Optimizations with predefined connectors were able to verify the developed concept. In the optimization of the injection system for the boat hull, a nearly symmetric distribution system has been obtained as it was expected. The optimization of the rectangular panel with predefined connectors also resulted in a reasonable process.

All three parts, symmetric bar, sort box and the VANTEX demonstrator, were optimized with all three optimization methods (heuristic, numerically deterministic and numerically robust). For all three parts numerical methods were found to be superior. The heuristic process of the symmetric part was very robust, but the fill time could be significantly reduced by a factor of $\frac{1}{3}$ with a numerical robust optimization. For this problem, the deterministic optimization led to a very unstable process. Similar results were obtained for the sort box, where both numerically optimized processes are better than the heuristic process. Although the fill time of the robust process rose by a factor of $\approx 1.5$ compared to the deterministic optimized process, the scrap rate could be reduced from 26% to 7%.

Besides the similar results for the symmetric bar and the sort box, the results of the VANTEX demonstrator part also showed the influence of the optimization possibilities. The robust optimization of this part with a symmetric injection condition did not reach the desired quality limit of 99.9% fill degree. When asymmetric injections were facilitated, a process could be found, which fills the base plate and achieves the desired part
quality. Therefore, it is important to recognize if the design space is to small to reach the desired objective. Then additional design freedom has to be given to the optimization.

Using the manufacturing costs as the main objective provides convenient results. Similar definitions of the objective function could be used for all presented problems.

Deterministic and robust optimization results were compared in Monte Carlo simulations with 500 runs. The Monte Carlo simulations showed clear advantages of processes that were optimized considering robustness. In general, the robust processes are significantly slower, but their lower scrap rate compensates the longer injection times.

5.8 Conclusions for process optimization

Numerical process optimizations significantly contribute to the development of improved RTM process parameters. In this chapter, the main focus has been on methods for the numerical optimization of robust injection processes. Gate and vent locations, as well as the pressures and flow rates of injection devices were considered as the major injection parameters.

Gates and vents can either be placed freely on the part or at predefined connector locations. If additional gates and vents may be placed freely on the part, a similar gate effectiveness has to be guaranteed, independent of the local mesh size. Therefore, a method to locally remesh the area around the additional mold connector has been developed. Simulations with the resulting remeshed models show a very similar influence of additionally placed gates on the flow, independent of the original mesh size.

The second option of using predefined connector locations may be applied if there are restrictions to possible locations of mold connectors, such as restrictions due to optical reasons or tooling. Connectors are defined during the preprocessing and the optimization algorithm decides whether to use the connector or not. If the connector is used, it can be connected to an injection device or it can be defined as a vent. This possibility of predefined connector locations is particularly interesting for vacuum assisted resin infusion (VARI), where a major issue of the process design is to find a suitable resin distribution system. An arbitrary number of possible connector lines may be defined before the optimization is started. Then, the optimization may use a fixed number of these connectors. Thus, an optimized injection system can be determined.
To improve the robustness of new designed RTM injection processes, the scrap rate obtained from a Monte Carlo simulation has been included in the process optimization. The number of required simulation runs can be kept at an acceptable level, if a statistical estimations of the reliability level of the Monte Carlo simulation used. This estimation has been included in the process optimization.

For almost all optimizations of real processes there are several objectives, which may be contradicting. For instance, two contradicting objectives are a fast fill time versus a high fill degree at limited injection pressures. Complex formulations of the objective function, which contain contradicting objectives have to be avoided. Otherwise, the optimization may easily get trapped in undesired optima. Including process robustness in the optimization, a low scrap rate becomes an additional objective. A convincing simple definition of the objective function has been developed, which focuses directly on the manufacturing costs. This allows the clear definition of the optimization targets, weighting various objectives against each other. Eventual penalty factors are not necessary.

The robust optimization approach has been a major development objective of this work. Besides the ability of numerical optimizations to treat several objectives simultaneously, the process variations can now be reliably assessed. As it may be expected, the various numerically optimized processes differ from heuristically optimized processes. Obviously, asymmetric processes are suggested by the numerical optimizations, while intuition will usually lead to symmetric solutions.

Helpful in developing robust processes is the identification of critical disturbances and combinations of disturbances, which might prevent successful injections. After identifying these critical disturbances, the possible problems may be addressed by an appropriate mold and preform design. However, this aspect has not been treated here and may be subject to future work.
Chapter 6

Estimation of the actual permeability distribution

In the previous chapter, the determination of optimum processing parameters with numerical optimizations has been discussed. However, incorrect assumptions about model variations and physical effects not implemented in the simulation, can lead to differences between the simulated and the real processes.

Here, deviations between real processes and simulations are assumed to dominantly result from incorrect permeability assumptions. If the permeability model mirrors the situation inside the mold, the model can be used to simulate the actual flow. Therefore, an approach to estimate the actual permeability distribution is developed. Based on the estimated permeability distribution, a fill simulation can visualize the true resin flow.

Visualization of the injection process, respectively the permeability distribution, may help to better understand the injection process of a specific part. Thus, the reasons for problems in the injection process can be identified. Typical sources of poor part quality can be identified and the monitoring of the resin flow within the closed mold can provide enhanced process insights. For instance, runners often appear on the same location because the preforms are cut or handled suboptimal. If runners are detected at a certain location in almost every injection, it is possible to alter and improve the preform. Alternatively, if preform or mold modifications are not possible, at least the process parameters can be optimized with the improved model.

In a direct quality management approach, the resulting laminate quality can be predicted based on the obtained fill pattern. Incomplete fillings
or intolerable joints of flow fronts can be recognized already during the injection.

Another possible application of flow visualization is the utilization within a feedback controlled injection system. Online feedback control requires detailed information about the flow within the closed mold. If this information is available, feedback control systems may compensate differences between simulated and the real process and thus make the injection processes more reliable. For illustration, the impregnation of a sandwich is examined where feedback control can improve the process reliability. Figure 6.1 shows a sandwich panel with a slightly reduced permeability in a small area. Due to the reduced permeability on the lower side, the resin flow went further on the upper side than on the lower side. The spatial pressure distribution in the upper and the lower layer is plotted in figure 6.1 as well. The total force on the sandwich core resulting from the fluid pressure on the upper layer is higher than the total fluid force from the lower layer. Thus, the force equilibrium will compress the lower layer stronger, while the upper layer will be released. Due to the dependence between permeability and fiber volume content, the effect will be cumulative. The resin flow in the lower layer experiences an additional reduction of the permeability. Here, feedback control provides a possible solution to reduce process variations.

An outline of the main topics discussed and the main results is listed below:

- **Update approach for the parametric permeability model** (section 6.1)
A basic idea to couple sensors and simulations to estimate the true permeability distribution is developed.

- **Implementation in the simulation environment** (section 6.2)
  Mathematical target definition required for the iterative update algorithm and summary of considered optimization algorithms.

- **Virtual experiments** (section 6.3)
  Shifting experiments to the virtual simulation environment makes the permeability estimations repeatable. Thus, estimation algorithms can be developed and tested.

- **Permeability estimation for a rectangular panel** (section 6.4)
  Plane geometry, where the permeability distribution is estimated using a Levenberg-Marquardt optimization.

- **Permeability estimation with balanced cell influence** (section 6.5)
  Investigation of the topology of the objective function. Applied optimization algorithms are a Levenberg-Marquardt algorithm and a combined Response Surface / Particle Swarm optimization.

- **Continuous in plane permeability measurement** (section 6.6)
  Continuous in plane permeability measurement application of developed estimation algorithms on permeability measurement. Besides the numerical methods the mechanical setup of the measurement device and measurement results are presented.

## 6.1 Update approach for the parametric permeability model

An sufficient variability of the parametric permeability model is the basic requirement to map the real injection situation. It must have the flexibility to approximate the real permeability distribution sufficiently well. Thus, the permeability distribution of the simulation model can be adapted to the real situation in the mold and the required model update can be based on few free parameters.

For this purpose, the parametric simulation model introduced in chapter 4 can be used. In contrast to the initialization using by a Latin-Hypercube scheme as it is done for Monte Carlo simulation, the disturbance values of the current injection are estimated.
Sensors have to be placed in the mold, and virtual sensors have to be implement in the simulation. Thus, the permeability values of the model can be updated according to information from the flow sensors, using an optimization algorithm, until the simulated sensor values match the measured values. Minimizing these differences, an approximative model of the real permeability distribution is obtained.

The estimation approach is illustrated in figure 6.2. Parallel to the real injection, a flow simulation is run to visualize the injection process. Thus, full information about the flow, including pressure distribution and flow velocities, can be calculated. Comparing simulated and measured values leads to modifications of the parametric permeability model. This model update is run iteratively until the injection is stopped.

**6.2 Implementation in the simulation environment**

The update of the permeability model can be formulated as an optimization (minimization) problem. The objective is to minimize the difference between measured sensor values on the one side and sensor values simulated with the current approximative permeability model on the other side. In this work pressure sensors are used to monitor the injection, resulting in the following least squares minimization
\[ F(X) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{p_{s,i}(X) - p_{m,i}}{p_{m,i}} \right)^2 \rightarrow \min \]  

(6.1)

- \( p_m \): measured sensor values
- \( p_s \): simulated values based on current (and updated) model
- \( X \): optimization parameter

The parameter vector \( X \) contains the disturbance values of the permeability model. In contrast to the process optimization discussed in section 5, the injection parameters such as vent locations or gate pressures are fixed. For better treating the pressure values of one injection from all sensors and all times are arranged in a single vector \( p \) (eq. 6.2).

\[
p = \begin{pmatrix}
P_1(t_1) \\
P_1(t_2) \\
\vdots \\
P_1(t_n) \\
P_2(t_1) \\
P_2(t_2) \\
\vdots \\
P_2(t_n) \\
\vdots \\
P_k(t_1) \\
\vdots \\
P_k(t_n)
\end{pmatrix}
\]  

(6.2)

with \( p_j(t_i) \): value of sensor \( j \) at time \( t_i \)
- \( k \): number of sensors
- \( n \): number of measured values per sensor

### 6.2.1 Levenberg Marquardt Method

Beside the optimization methods mentioned in section 5.2, the Levenberg-Marquardt method is a gradient based optimization algorithm for least squares minimization. Among others, this popular method is described in [23]. Close to the local optimum it achieves the fast convergence of second order methods. In contrast to general optimization methods as for example the Newton method, it does not require the second order derivatives of the objective function, because information from the special form
of the least squares problem (equation 6.1) is used. In an optimization problem with local minimum it cannot be guaranteed that the global optimum will be found. To get a reasonable probability of obtaining the global minimum, repeated minimizations should be run with different start points.

The method can be formulated as

\[ \left\| \begin{pmatrix} F'(x_i) \\ \mu I \end{pmatrix} s_i + \begin{pmatrix} F(x_i) \\ 0 \end{pmatrix} \right\| = \min \] (6.3)

\[ x_{i+1} = x_i + s_i \] (6.4)

Compared to the Gauss-Newton method, this formulation has the advantage that a unique solution \( s_i \) can always be found because the matrix \( \begin{pmatrix} F'(x_i) \\ \mu I \end{pmatrix} \) always has a full rank.

The correction step \( s_i \) is damped for sufficiently high parameter \( \mu \). Thus, too large iteration steps can be avoided, and convergence of \( x_{i+1} = x_i + s_i \) towards a local minimum can be proven. Then again, high \( \mu \) cause small steps and therefore slow convergence. In practical applications heuristic criteria are used to control the parameter \( \mu \).

The required Jacobi matrix is assembled as shown in equation 6.5.

\[ F' = \begin{pmatrix} \frac{\Delta p_1(t_1)}{\Delta K_1} & \frac{\Delta p_1(t_1)}{\Delta K_2} & \ldots & \frac{\Delta p_1(t_1)}{\Delta K_j} \\
\frac{\Delta p_1(t_2)}{\Delta K_1} & \frac{\Delta p_1(t_2)}{\Delta K_2} & & \\
\vdots & \vdots & & \\
\frac{\Delta p_2(t_1)}{\Delta K_1} & \frac{\Delta p_2(t_1)}{\Delta K_2} & & \\
\vdots & \vdots & & \\
\frac{\Delta p_k(t_n)}{\Delta K_1} & \frac{\Delta p_k(t_n)}{\Delta K_2} & \ldots & \frac{\Delta p_k(t_n)}{\Delta K_j} \end{pmatrix} \] (6.5)

\[ p_j(t_i) : \text{value of sensor } j \text{ at time } t = t_i \]

with \( K_j \) parameters to be optimized

\[ k \] number of sensors

\[ n \] number of measured values per sensor

As it is not composed of analytically determined gradients but approximated with difference quotients, it is subject to numerical scattering of the simulated values and the influence of the step width used to calculate the difference quotient.
6.2.2 Update algorithms for permeability estimation

Various optimization algorithms can be used to solve the minimization problem described by equation 6.1. In this work the following methods have been chosen:

- Levenberg-Marquardt: This gradient based algorithm with fast convergence near the optimum is particularly suitable for the solution of least squares problems. Although it shows convergence rates like the Gauss-Newton method, it does not require the second derivation (Hessian matrix), the first derivative of the objective is sufficient.

- Response Surface: Similar to Levenberg-Marquardt, it can show fast convergence near the optimum [69]. As it uses derivatives calculated from the response function \( f(X) \), no numerically determined gradients form the original function are required. It can smoothen scattering of the sample data if the number of sample points used to determine the Response Surface is higher than the number of parameters.

- Particle Swarm Optimization: No derivatives are required in the optimization (0-th order method). It is simple to implement and to tune. Because it is a stochastic method there is a chance to find the global optimum, and in particular to leave small optima resulting from numerical noise.

6.3 Virtual experiments

For the reproducibility the investigations were carried out as “virtual experiments” in a simulation environment. The work flow of the virtual experiments is summarized in figure 6.3

1. Fix locations of the sensors
2. Definition of disturbance cells
3. Application of permeability disturbances (e.g. \( \Delta K_4 = +20\% \))
4. Simulation of “real sensor values”
5. Clearing the applied permeability disturbances (\( \Delta K_1 \ldots \Delta K_6 = 0 \))
6. Start of permeability disturbance determination
Figure 6.3: Setup of virtual experiments
Thus, the present permeability disturbances during the calculation of the measured sensor data are exactly known. The estimated permeability distribution can be compared with the original one, which is an important issue to assess the estimation quality.

6.4 Permeability estimation for rectangular panel

Figure 6.4 shows a test geometry for permeability estimation. The gate is located at the lower left corner of the flat plate, a line vent is applied at the top. Two sensors have been used to estimate the disturbances of six zones, K1 to K6. The original disturbance for the generation of the “real sensor values” was an isotropically increased permeability of \( K_1 = 0.5 \). All other zones were undisturbed with \( K_2 \ldots K_6 = 0 \). Deviating from the initialization of disturbance zones described in section 4.1, the disturbance values correspond directly to the permeability influence. In this section, the disturbed permeabilities are calculated by \( K_{dist} = K_{\infty} \cdot (1 + K_i) \), e.g. \( K_1 = 0.5 \) means a permeability increase of 50% for zone 1.

The Levenberg-Marquardt method has been used for this estimation example because of its particular applicability to least square minimization. Although six disturbance zones are used covering only half of the part, sensor information from full simulation till complete filling of the mold has been used. Figure 6.5 shows the estimation history for the dis-
turbance zones. Already after 9 iteration steps steady disturbance values have been reached. A permeability disturbance of +50 % has been found for the cell $K_1$, the permeability of all other cells has been determined to be 0. These are the values that were originally used to generate the “real sensor values”.

### 6.5 Permeability estimation with balanced cell influence

In the previous example the disturbance zone 1 has an over-proportionally high influence on the flow as all resin flows through this zone. So, transferability to other configurations may be questioned. A more representative scenario will be discussed in this section.

After the flow front has passed a disturbance zone, its disturbance values can be fixed. Thus, disturbances only have to be estimated in a limited area behind the current flow front and the number of parameters that have to be estimated at once can be reduced. The estimation of zones closer to the gate is assumed to be done in a previous estimation, the remaining disturbance values will be evaluated in following time steps. However, to account for this iterative permeability estimation scheme,
only sensor information obtained up to the time when the flow front has passed these zones should be used. Deviating from the definition given in section 4.1, the disturbance values are directly related to the permeability as it has been used in the previous example in section 6.4.

### 6.5.1 Geometry and configuration

Unlike the problem presented in the previous section 6.4, now the disturbance zones are arranged in an area close to the flow front at a certain time during the injection. Due to the geometrical symmetry, the effect of the six disturbance zones on the flow is almost equal, i.e. no zone has an superior influence as it had the lower left zone K1 in figure 6.4. The pseudo measured values required in the virtual experiments were generated using the isotropic disturbances given in figure 6.6.

Figure 6.7 shows three different experimental configurations with 3, 7 and 13 sensors. At first, the configuration with 13 sensors is used. In section 6.5.4 the sensor configuration will be varied to test the influence of different numbers of sensors on the estimation. Beside their number, the sensor locations with respect to the disturbance zone may have an influence on the estimation as well. Here, the sensors are preferably located closer to the gate because the usable information gets less the later the sensors are reached by the resin. In the configurations with a low number of sensors, they are placed on the interface between the zones.
6.5.2 Levenberg-Marquardt minimization

As it showed fast convergence in the previous investigation, the first used optimization method is the Levenberg-Marquardt algorithm.

Figure 6.8 shows the optimization history, including the residuum and the differences between estimated values and the correct values. After 10 iteration steps no further improvement could be found, the original disturbance values used for the generation of the “measured sensor values” (figure 6.6) could not be determined. The final residuum of the minimization is rather high ($\approx 5 \cdot 10^{-4}$), so it can be concluded that a local minimum has been found.

6.5.3 Topology of the objective function

For a better understanding of the behavior of the Levenberg Marquardt algorithm in this problem, the topology of optimization function has been investigated. For visualization reasons, only results from varying the permeabilities of the surfaces 1 and 5 are shown (figure 6.9).

Figure 6.10 shows two views on its topology for varying disturbances of the surfaces 1 and 5. On the left picture a valley of low values can be seen, which is oriented along a line defined by parameter combinations $K1 + K5 \approx 0$. Apart from this valley where the global optimum is located, at least one more local minimum can be identified in the right picture. Local waviness, that can be seen in the right picture as well, makes the optimization even more difficult. These problems may become worse when the other four defined disturbance zones are varied as well.
6.5 Permeability estimation with balanced cell influence

Figure 6.8: Permeability estimation result for problem given in figure 6.6 using Levenberg-Marquardt optimization. The original disturbance could not be found although 13 sensors and a fine mesh with 4077 nodes in total have been used.

Figure 6.9: Location of the surfaces 1 and 5, where the permeability is varied.
Figure 6.10: Far (left) and medium (right) look on the topology of objective function. Illustrated for permeability variations of the surfaces 1 and 5.

Figure 6.11: Close look on the optimum region of the topology shown in figure 6.10.

A closer look on the area near the global minimum is given in figure 6.11, where the valley with low function values can be seen more detailed. Particularly the figure on the right shows interesting details. The dents in the illustration result from the graphic processing and should therefore be ignored. The red line interpolates the calculated minimum function values. On the one hand, low values can be found in a wide range of parameters which can be derived from the very low slope of the red line. On the other hand parameters close to the optimum can lead to significantly higher function values, if the parameters are outside the valley. The residuum of $\approx 5 \cdot 10^{-4}$ in the optimization results shown in figure 6.8 is obviously too high to guarantee good permeability estimation results. Comparing this residuum with the topology of figure 6.11, one can explain the difficulties to find the original values with the Levenberg-Marquardt
algorithm. In figure 6.8, the sum of the remaining permeability disturbances $\sum K_i$ is approximately 0 after 10 iteration steps. This indicates, that a solution point within the valley has been found, but the global minimum could not be reached.

6.5.4 Response Surface and Particle Swarm Optimization

To overcome the problems encountered with the Levenberg Marquardt algorithm, Particle Swarm Optimization was used. Thus, the evaluation of gradients is avoided and solutions can escape local minima. An improved convergence behavior towards the optimum was found when a Particle Swarm optimization was combined with the Response Surface method. Therefore, a response surface was calculated using the data from the particle history as support points. The selection of the support points was done by their fitness values. A selection by the distance from the global best point found so far showed a slower rate of improvement. The parameters of the response surface were calculated from a slightly overdetermined system to reduce the numerical noise. The objective values evaluated at the suggested optimum locations can be shared with the swarm by adding them to the global list of evaluated points, which is used by the stochastic particles.

In figure 6.12 estimation results obtained with this algorithm are presented. Within the first 50 iteration steps the residuum rapidly drops, and reaches values significantly below the previous results of Levenberg-Marquardt minimization. The remaining disturbance between the originally applied disturbances and the estimated values reaches a negligible level.

In further investigations the influence of the number of sensors was considered. Therefore, the estimation has been repeated including pressure values from seven (figure 6.13) and three sensors (figure 6.14) in the objective function.

Within 60 to 70 iterations good estimations results were achieved. The differences between correct and estimated values could be reduced to values below 5% for both objective functions. Analogously, the residuum has been reduced to the same level that had been reached with 13 sensors. So, it can be assumed that the estimation of the permeability distribution with a reasonable number of sensors is possible.
Figure 6.12: Permeability estimation result for the problem given in figure 6.6. Particle Swarm optimization combined with Response Surface optimization has been used, including 13 sensors in the objective function. The simulations were run on a fine model with 4077 nodes.
Figure 6.13: Permeability estimation result using 7 sensors.
Figure 6.14: Permeability estimation result using 3 sensors.
6.6 Continuous in plane permeability measurement

The permeability is an important input to RTM fill simulations. As the permeability shows significant scatter, a high number of measurements is required to obtain statistically reliable values. So, fast experiments with automated data evaluation are needed to minimize the practical effort for experimental determination. The presented work focuses on the automated experimental characterization of the permeability of textile fabrics.

To evaluate the sensor data from the permeability measurements, a model describing the injection is required. For one-directional flow experiments an analytical description of the pressure field is available. Even if the flow front reached the end of the preform, the permeability can still be evaluated and the saturated permeability can be determined. However, for the state of the art 2D in plane permeability measurement, an analytical description is possible only as long as the expanding flow front is of elliptical shape. Limitations resulting from the analytical flow model are

- Limitations of the preform shape, in particular a small injection gate diameter compared to the resin flow length is required
- Flow front has to expand without reaching the edge of the fiber preform
- A permeability averaged between the partially saturated permeability in the flow front region and the fully saturated permeability far behind the flow front is determined

A practicable solution to overcome these limitations is to replace the analytical description by numerical calculations. Corresponding to the analytical model, the difference between simulated and measured sensor values is minimized in an iterative adaption of the permeability tensor. By coupling simulations with experiments, virtually any preform shape can be used. This may include large inlet sizes and non circular shapes or small preform sizes, e.g. may be required for braided preforms that are often limited in their size. Thus, the restriction to small diameters of the injection gate [120] is overcome.

The continuous measurement reduces effort for measurements. One preform can be used in one experimental setup to measure all fiber volume contents in the range of interest and at several injection flow rates. Here, saturated permeabilities are considered exclusively. Pressure sensors were
chosen for flow monitoring as they can provide continuous information about the flow in closed metallic molds.

6.6.1 Experimental setup

A fully computer controlled experimental setup has been developed, in which the measurement cycle for the continuous permeability measurement is automated (figure 6.15). A test fluid is injected into the fabric at a controlled pressure, excessive resin is collected and can be reused in following measurements. Pneumatic cylinders are used to compact the fabric to the desired fiber volume contents, and the data required for permeability evaluation is recorded.

Measuring procedure  Figure 6.16 shows the measurement procedure for continuous permeability measurements. First, the preform is positioned and the cavity is closed. Then, fluid is injected until the preform is completely wetted out. After applying the initial compaction pressure the fiber volume content stabilizes at a specific level, depending on the applied compaction pressure and the investigated fabric.

The actual measuring cycle at the current fiber volume content is started with injecting at the desired injection pressure. A sufficiently long settling time has to pass until the pressure distribution is stabilized, then the pressure distribution can be recorded. Thereafter, the next injection pressure can be applied.

When all desired injection pressures have been tested, the next higher fiber volume content can be applied. After the permeability has been measured under the maximum desired compaction pressure, the measurement stops.
6.6 Continuous in plane permeability measurement

Mechanical realization of preform compaction  
Pneumatic cylinders are used to apply the compaction pressure on the preform (figure 6.17). The lower cavity plate is equipped with six pressure sensors to record the pressure distribution within the fabric during the injection. Setting a fiber volume content requires a force equilibrium between cylinder forces on the one side, fabric compression force and fluid pressure on the other side.

When the fluid is injected to measure the permeability, this equilibrium of the forces changes. As the cylinders would retract due to the increased counter pressure when the test fluid is injected, wedges are used at the sides of the load frame, preventing lift of the cavity and therefore of the fiber volume content [3]. Figure 6.18 shows the final continuous permeability measurement setup.

Measured data  
Discrete pressure sensors punctually measure the pressure within the flow field. This sensor data is used to evaluate the principal orientation of the permeability tensor and the anisotropy ratio \( \frac{K_1}{K_2} \). Additionally, the injection pressure at the inlet and the compaction pressure in the cylinders is recorded.
Figure 6.17: Mechanical setup: Cylinders apply compaction pressure onto the preform, wedges inhibit decompression when fluid is injected. The lower bolts are removable to make rigid frame mountable. Wedges are moved by springs (not in the scheme) when the upper plate lowers due to compression.

Figure 6.18: Realized experimental setup with load frame, pneumatic cylinders and laser displacement meter [3].
The flow rate of the injected fluid is required to calculate the absolute values of the preform permeabilities. An electronic balance records the weight of the pressure pot over time, from which the injected flow rate can be determined. Although it is possible to measure the volume flow directly for non-curing fluids such as silicon oil, indirect measurement of mass flow has been preferred as it leads to precise values for the calculated volume flow.

A laser displacement meter is used to measure the cavity height. From the cavity height, the current fiber volume content can be calculated, using the preform weight, size and the fiber density.

**Sensor integration in the measurement setup**  Six sensors are integrated in the measurement setup. Their location can be changed depending on the tested preform, and their expected anisotropy. For almost isotropic preforms the sensors are distributed evenly over an angle of $180^\circ$. Due to the symmetry of the permeability tensor only one half of the preform has to be observed. For strongly anisotropic preforms the sensor positions can be changed to better match elliptical isobars. Those locations are covered by blind plugs in figure 6.19.

Pressure measurements provide flow data after the flow front has passed the sensor which is a prerequisite for continuous measurements. However they require appropriate integration in the setup to provide reliable pressure data. The sensor surface must not be in direct contact with the preform. As figure 6.19 shows the fluid pressure is measured via a transmitter oil, decoupling fluid pressure and preform compaction pressure. Otherwise unavoidable slight deflections of the cavity plates would distort the sensor values.

**Preform shape**  Figure 6.20 shows the developed preform shape that has been used for the measurements. A large gate diameter has been chosen to minimize the occurring pressure gradients, allowing low injection pressures at relatively high flow rates. The rounded outer edges reduce the risk of fringed edges.

**Test fluid**  All measurements have been done using the silicon oil Bay-silone M100 from Bayer, with a viscosity of $\eta = 0.105 \text{Pa} \cdot \text{s}$ as test fluid. Obviously, reactive resins can only be used in continuous measurements, if the curing time is significantly longer than the total time required for a measurement cycle.
Figure 6.19: Integration of pressure sensors in the lower cavity plate. Fluid pressure is measured via a transmitter oil, decoupling fluid pressure and preform compaction pressure.

Figure 6.20: Shape of preform used in measurements
6.6 Continuous in plane permeability measurement

6.6.2 Numerical methods

Calculation of permeability values from raw pressure data  The pressure field in the preform is calculated using a finite element calculation (sLIP). Because there is no moving flow front for completely wetted preforms, the pressure field is constant over time dependent. A simulation program using the sLIP library has been developed, allowing easy adaption to arbitrary preform shapes and varying sensor positions.

In the iterative calculation loop for the actual permeability, the three values $K_1$, $K_2$ and $\varphi$ are the optimization variables. They can be modified by the optimization algorithm, and the sLIP calculation module can return the resulting pressure field. Additionally, the cavity height, the fiber volume content and the injected fluid flow are required by the simulation module to calculate the pressure field.

The target function is calculated considering the remaining error between measured and simulated pressure values. It is defined as

$$r = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{P_{i,\text{sim}} - P_{i,\text{measured}}}{P_{i,\text{measured}}} \right)^2$$  \hspace{1cm} (6.6)

giving the averaged, relative error of all pressure values measured.

As described in section 6.2.2 various optimization algorithms are considered for the estimation of the true permeability values. Swarm optimization can be used for multi-modal problem types but suffer from low convergence rate. For problems with a single optimum and a smooth topology, the gradient based Levenberg-Marquardt optimization can be used with fast convergence to the optimum. Equation 6.7 shows the Jacobian matrix used. It is simplified compared to definition 6.5 in the previous subsection, because instead of a time history of the pressure sensors, only one value per sensor and per simulation is available.

$$J = \begin{pmatrix}
\frac{\Delta p_1}{\Delta K_1} & \frac{\Delta p_1}{\Delta K_2} & \cdots & \frac{\Delta p_1}{\Delta K_j} \\
\frac{\Delta p_2}{\Delta K_1} & \frac{\Delta p_2}{\Delta K_2} & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\Delta p_k}{\Delta K_1} & \frac{\Delta p_k}{\Delta K_2} & \cdots & \frac{\Delta p_k}{\Delta K_j}
\end{pmatrix}$$ \hspace{1cm} (6.7)

with $p_j$ : value of sensor $j$

$\hspace{1cm}$

$j$ : number of tested gradients

$k$ : number of sensors
Similar results could be found for Levenberg-Marquardt calculations and for Particle Swarm optimization. However, a sufficiently fine finite element model for the calculation of the pressure field and the gradients had to be used for Levenberg-Marquardt optimization. This dependency on the mesh size may be due to the scattering of numerically calculated gradients, which decreases when finer calculation models are used. The Levenberg-Marquardt optimization showed a fast convergence for the investigated problems. Depending on start point 3..10 iterations were required, compared to $\approx 200…1000$ required iterations for swarm optimization.

**Influence of inlet pressure** The effective diameter of the inlet in the preform may be uncertain due to local disturbances from cutting and handling. For data evaluation three possible variants were considered, differing with respect to the inclusion of the pressure measured at the injection gate

1. $P_{\text{in, sim}} = P_{\text{in, measured}}$
   Sequential calculation of anisotropy ratio $\frac{K_1}{K_3}$ and $\varphi$, then the absolute values are determined by fitting the volume flow $q_{\text{in, sim}}$. Fast convergence can be expected, because the determination of the permeability values is split into two separated optimizations. However, including the pressure at the injection gate $P_{\text{in, measured}}$ the local disturbances around the gate can influence the evaluated permeability.

2. $q_{\text{in, sim}} = q_{\text{in, measured}}$
   The pressure measured at the injection gate is not considered, neither in the simulation, nor in the objective function.

3. $q_{\text{in, sim}} = q_{\text{in, measured}}$ and $P_{\text{in, sim}} = P_{\text{in, measured}}$
   The injection pressure is not applied in the simulation, but it is included in the objective function. So, an influence of the injection gate size on the evaluated values remains.

Variant 2 has been selected because the injection pressure does not have a direct influence. Thus, one sensor less is used, but the edge permeability around the inlet does not influence the evaluated permeability values.

**Remaining error after optimization** Parameters are calculated using a least square minimization. In practical applications of evaluated measured data, a residual error remains. This is because of the scattering
of volume flow measurement and the flow sensor data, but mostly due to differences between the flow model and the real flow. There may be (though probably small) discretization errors when using a numerical flow model. But other setups using analytical models show similar residual errors as well [2]. More important are locally varying permeabilities instead of a homogeneous permeability all over the preform. A second main source for the residual error result from differences in the geometry of the preform in experiment and simulation. Particularly, the dimensions of the outer edge of the preform are sensitive to handling.

**Influence of deviations of the preform position and dimensions**

Deviations in the actual shape of the tested preform from the simulation models will occur. Figure 6.6 shows an example of a preform of correct dimensions, but it is displaced in 90° direction (in the coordinate system used). Those disturbances affect the results and are caused by the measurement setup only. In this subsection the influence of several different disturbances is discussed. Investigated influences of deviations from the idealized preform shape are

- preform shifted 5 mm in 0° direction
- preform shifted 5 mm in 90° direction
- Edge of preform shifted 5 mm in 90° direction, injection port remains at its location
- Edge of preform shifted 5 mm in 0° direction, injection port remains at its location
- Diameter of injection port increased from 50 mm to 60 mm

Additionally, the influence of using a coarse simulation model for permeability calculation is investigated, to estimate the influence due to the finite element approximation. Therefore a small model of 602 nodes and 1082 elements has been used, which is \( \approx 1/9 \) of the model size used for the calculation.

**Calculation procedure:**

1. Assume permeability values (e.g. \( K_{1b} = 3 \cdot 10^{-10} m^2 \), \( K_{2b} = 1 \cdot 10^{-10} m^2 \), \( \varphi = 90° \))

2. Calculate sensor pressures for several differently disturbed simulation models including all deviations of interest
3. Re-calculate permeability for all disturbed cases.

All permeability evaluations were done using an undisturbed model with 5098 nodes and 9741 elements. The various disturbed simulations used models of very similar sizes. Optimizing one measured data point (injection pressure and FVC) on a Workstation (Intel Xeon 2.80 GHz, 1 GB RAM) takes approximately 100 s including all required simulations for gradient calculation and line search. Faster convergence can be achieved for most measurements, but the robustness of the estimation had the highest priority.

To assess the quality of evaluated permeability values, relative errors scaled by the actual permeability values are used

$$\epsilon_K = \frac{K_{dist} - K_0}{K_0} \tag{6.8}$$

where $K_{dist}$ is the permeability calculated with disturbed preform shape and $K_0$ is the actual permeability. For orientation of the main axis the absolute errors are considered, because no useful reference value can be defined to calculate relative errors

$$\Delta \varphi = \varphi_{dist} - \varphi_0 \tag{6.9}$$

with $\varphi_{dist}$: Calculated orientation
$\varphi_0$: Actual orientation

Relative errors are used, not absolute errors, as the pressure sensors close to the injection gate are subject to higher pressures and therefore higher error pressures than sensors far from the injection point. Thus, over-weighting of pressure values from the inner sensors can be avoided.
### Table 6.1: Influence of perform disturbances on calculated permeabilities.

<table>
<thead>
<tr>
<th>Disturbance Type</th>
<th>$\Delta \varphi$</th>
<th>$\epsilon K_1$</th>
<th>$\epsilon K_2$</th>
<th>Increased gate diameter of 60 mm</th>
<th>Outer edge shifted 5 mm in 0° direction</th>
<th>Outer edge shifted 5 mm in 90° direction</th>
<th>All edge shifted 5 mm in 90° direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse model with 602 nodes and 1082 elements</td>
<td>-</td>
<td>0.85%</td>
<td>0.26%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Increased gate diameter of 60 mm</td>
<td>-</td>
<td>0.00%</td>
<td>0.00%</td>
<td>2.89%</td>
<td>-3.76%</td>
<td>1.62%</td>
<td>-12.6%</td>
</tr>
<tr>
<td>Outer edge shifted 5 mm in 0° direction</td>
<td>-</td>
<td>10.2%</td>
<td>-1.62%</td>
<td>3.61%</td>
<td>-5.4%</td>
<td>13.7%</td>
<td>4.33%</td>
</tr>
<tr>
<td>Outer edge shifted 5 mm in 90° direction</td>
<td>-</td>
<td>10.2%</td>
<td>-1.62%</td>
<td>3.61%</td>
<td>10.2%</td>
<td>8.1%</td>
<td>-3.8%</td>
</tr>
<tr>
<td>All edge shifted 5 mm in 90° direction</td>
<td>-</td>
<td>10.2%</td>
<td>-1.62%</td>
<td>3.61%</td>
<td>10.2%</td>
<td>8.1%</td>
<td>-3.8%</td>
</tr>
</tbody>
</table>

Isotropic Preform with $K_1 = K_2 = 2 \cdot 10^{-9} m^2$
Table 6.2: Influence of perform disturbances on calculated permeabilities. Orthotropic Preform with $K_1 = 5 \cdot 10^{-9}m^2$, $K = 1 \cdot 10^{-9}m^2$ and $\varphi = 0$.
The tables 6.1 and 6.2 summarize the calculated results for an isotropic and an orthotropic preform. Errors in positioning the preform showed noticeable effects, depending on the orthotropy of the preform, the affected preform borders and the direction the preform was shifted. However, the deviations of 5 mm from the ideal shape and position that were tested in this investigation are a lot higher than the deviations that are typically achievable in experiments. In our experience deviations are in the range of 1 mm. So, assuming linearity, the approximate evaluation error will be below 3% for the worst deviation case investigated. The simulations also showed that the modified gate size had a very low influence on the results for orthotropic media and no influence for isotropic media. The effect of a coarse model is also very low, indicating that a sufficiently large finite element model is used for the permeability calculation.

Determination of preform permeability characteristics

The permeability values calculated for each measurement point are fitted to a Kozeny-Carman model (eq. 4.12) using a least squares fit. To fit the model relative errors are used as given by eq. 6.10.

$$\sum \frac{(K_{kc}(\phi) - K_{measured})^2}{K_{kc}(\phi)} \Rightarrow \text{min.}$$

(6.10)

An overall standard deviation over the various fiber volume contents will be calculated using the Kozeny-Carman model data as the average value at the corresponding FVC (eq. 6.11).

$$s_K = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (K_i - K_{kc}(\phi_i))^2}$$

(6.11)

6.6.3 Results of permeability measurements

In the results shown in this section, the principal directions of measured permeabilities are given with respect to warp direction. Laying the preform with the inside of the fabric role upside or downside into the measurement setup will turn the determined orientation $+\alpha$ to $-\alpha$. As looking from top or looking from bottom turns, there might be different results because of this. So attention has to be payed to ensure a reproducible and consequent layup.

Detailed measurement example

Figure 6.22 shows the raw data of one measurement. The corresponding permeabilities were measured
Figure 6.22: Pressures measured at the six sensors placed in the lower plate of the test setup

at five different fiber volume contents, testing three different injection pressures 0.35 bar, 1.05 bar and 1.75 bar. The effect of settling times can be observed as the pressures stabilize. During the experiments the pressures should be checked for a sufficient length of the settling times.

In figure 6.23 the evaluated permeability values for the pressure history given in figure 6.22 are presented. The solid line shows the Kozeny-Carman interpolation, which is a good approximation of the measured values.

With the developed setup about 15 different injection conditions of each preform were measured, varying in injection pressures and fiber volume content. As it can be seen in figure 6.22, one measurement takes about 30 minutes. Including the preparation and cleaning, approximately 1 hour is required per preform in total.

Results for E-Glass fiber woven fabric Five measurements were performed on Hexcel 01113 1000 TF970, a woven twill glass fabric with a nominal weight of 390 g/m². Eight layers have been used for each tested preform. Figure 6.24 shows the evaluated permeability values. Because the used settling times after increasing the compaction pressure were chosen too short, many measurements at low injection pressure could not be evaluated. The continuous lines shows the Kozeny-Carman approximation, which is used to convert the permeability values between different fiber volume contents.
Figure 6.23: Evaluation of one measurement, three different injection pressures were tested at each fiber volume content.

Figure 6.24: Measured Permeabilities of Hexcel G1113 as function of FVC.
Table 6.3 shows the determined data obtained from $N = 44$ measurements. The permeability values at any fiber volume content can be calculated with equation 4.12 using $c_{K_1}$ and $c_{K_2}$. Their scattering of 11% and 17% is low compared to literature data [77, 83, 93, 92, 48], even though the approximation with Kozeny-Carman does not perfectly match the results.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Determined value</th>
<th>Std. variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>4.5°</td>
<td>±1.5°</td>
</tr>
<tr>
<td>$K_1/K_2$</td>
<td>3.65</td>
<td>±0.59</td>
</tr>
<tr>
<td>$c_{K_1}$</td>
<td>$2.86 \cdot 10^{-10} m^2$</td>
<td>11%</td>
</tr>
<tr>
<td>$c_{K_2}$</td>
<td>$8.18 \cdot 10^{-11} m^2$</td>
<td>17%</td>
</tr>
<tr>
<td>$N$</td>
<td>44</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 6.3: Material data for Hexcel 01113

**Results for carbon fiber woven fabric** Eight measurements were performed on Hexcel G0986 D 1200, a woven twill carbon fabric with a nominal weight of $285 \, g/m^2$. Figure 6.25 shows the evaluated permeability values.
Only 5 layers were used per measurement, resulting in too thin preforms. The measured cavity heights at the maximum compaction pressure was only 1.1 mm, so low errors in the thickness measurement resulted in noticeable FVC deviations. As it can be seen in the graphic 6.25 a deviation of up to 5% in FVC was found at the maximum compaction pressure of 1.25 bar. At a compression pressure of 0.15 bar different preform heights between 1.42 and 1.61 mm were measured (12% deviation). As the preform should behave similar in each trial, and as FVC around 70% is hard to reach at the used compaction pressure, this high deviation is assumed to be due to an incorrect calibration of the displacement meter.

Correspondingly, the Kozeny-Carman fit in figure 6.25 is a rather poor approximation Likewise, the high standard deviations for the permeabilities in table 6.4 is obviously due to the insufficient quality of the approximation model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Determined value</th>
<th>Std. variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>-77.1$^\circ$</td>
<td>±5.8$^\circ$</td>
</tr>
<tr>
<td>$K_1/K_2$</td>
<td>2.64</td>
<td>±0.28</td>
</tr>
<tr>
<td>$c_{K_1}$</td>
<td>$1.45 \cdot 10^{-9} m^2$</td>
<td>32%</td>
</tr>
<tr>
<td>$c_{K_2}$</td>
<td>$5.53 \cdot 10^{-10} m^2$</td>
<td>32%</td>
</tr>
<tr>
<td>$N$</td>
<td>75</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 6.4: Material data for Hexcel G0986

**Results for non crimped carbon** Eight measurements were performed on Hexcel NBB00 HR1270 0400 G C1 TFX, a non-woven carbon fabric with a nominal weight of 410 g/m$^2$. Six layers have been used for each tested preform. Figure 6.26 shows the evaluated permeability values. Again, as for the previously presented glass fabric, the settling times were chosen too short so several measurements at low injection pressure could not be evaluated.

### 6.6.4 Discussion of measurement results

Obviously, the fiber volume content at a given compaction pressure varies significantly. For the non-crimped carbon variations up 5% have been found. Additionally, the fiber volume content of nearly 70%, reached for woven carbon fabric at 1.25 bar appears unrealistically high. This probability results from an insufficient mechanical setup. For better results,
224  Estimation of the actual permeability distribution

![Permeability vs Fiber Volume Content](image)

Figure 6.26: Measured Permeabilities of Hexcel NBB00 HR1270 as function of FVC.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Determined value</th>
<th>Std. variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>8.5°</td>
<td>±2.5°</td>
</tr>
<tr>
<td>$K_1/K_2$</td>
<td>2.47</td>
<td>±0.29</td>
</tr>
<tr>
<td>$c_{K_1}$</td>
<td>$1.60 \cdot 10^{-10} m^2$</td>
<td>15%</td>
</tr>
<tr>
<td>$c_{K_2}$</td>
<td>$6.58 \cdot 10^{-11} m^2$</td>
<td>17%</td>
</tr>
<tr>
<td>$N$</td>
<td>112</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 6.5: Material data for Hexcel NBB00 HR1270

an improved setup is necessary, where particularly a parallel cavity and precise thickness measurement have to be guaranteed.

In spite of these problems, the results are very promising. All data has been approximated by the Kozeny-Carman relation, and the standard deviation has been calculated with respect to this approximation. Although the errors in the measurements of the fiber volume content have a negative influence on the approximation quality, the resulting standard deviations are in the range known from literature.

If the permeability values are plotted over the compaction pressure instead of the measured fiber volume content, an even better reproducibility is obtained.
6.7 Conclusions for permeability estimation

General estimation of the permeability distribution  A permeability estimation algorithm has been developed which can be used to determine the actual permeability distribution in the mold during an injection. The estimation is based on the information from several sensors placed in the mold. In virtual experiments, the estimation algorithm was demonstrated to work and to reproduce the permeability distribution of the original model. In this way the real fill process can be monitored.

Obviously, the number and location of flow sensors affects the approximation quality. It could be shown that the number of estimated model parameters may be higher than the number of pressure sensors used. However, a design strategy to determine the number and the optimum locations of the flow sensors has to be developed. On the one hand, the uniqueness of the estimated permeability distribution leading to the measured sensor data should be guaranteed. On the other hand, the number of the sensors should be as low as possible to minimize the production costs.

In future work, the permeability estimation and flow monitoring should be run online with the real injection, which will require improved convergence rates of the model update. Along with the Swarm Optimization and Response Surface optimization which were used, further algorithms might be tested, such as Simplex Optimization or Simulated Annealing optimization. Additionally, the algorithm investigated in this work might be improved, for example, by preconditioned Levenberg Marquardt optimizations.

Beyond virtual experiments, the permeability distribution has to be estimated for experimentally obtained sensor data. In this work, experimentally determined sensor data has been used for continuous permeability measurement. However, general permeability estimations of the spatial permeability variations, were exclusively run in an experimental environment.

Based on the information of the real permeability distribution, a feedback control loop for the injection pressures and volume flows may be developed. Currently, the effort needed for a process optimization as it was presented in section 5 is too high because the calculation time will exceed the available time for online optimization. Simpler to realize, but although effective, will be feedback controlled injections that follow a fill pattern, which is optimized offline prior to the injection. Therefore, the flow front velocity might be used as control target. Besides the increased effort for flow sensing and online flow monitoring, additional gates have
to be placed on part, as the gate influence on the flow velocity is limited to a relative short distance.

**Continuous permeability measurement**  Continuous in-plane measurements can significantly speed up the experimental determination of permeability values. In one setup, the saturated permeability can be measured at various fiber volume contents and at various injection rates. Thus, costs for materials can be saved, and working time can be reduced. This is particularly useful for the simulation of VARI processes, for which the permeability is required at several fiber volume contents.

In the development of the continuous in-plane permeability measurement setup, simulations and experiments were successfully coupled. A fast routine to evaluate the permeability from measured sensor data has been developed. As the pressure distribution in an arbitrary preform shape can be calculated, the previous restriction of in-plane measurements to small injection gate diameters is overcome, and virtually any preform shape may be used. Errors from deviations between the real preform shape and the preform shaped assumed in the simulations were shown to be low as compared to the scattering of the permeability itself. A procedure has been developed and implemented, which controls the measurement setup automatically. Additionally, it records all data required to determine the in-plane permeability at various fiber volume contents and flow rates.

The mechanical setup used during the development of the method should be improved. Namely, the parallelism of the cavity, the stiffness against opening due to fluid pressure, and the measurement of the cavity height might be improved. To minimize the relative error in the measured fiber volume content, a minimum preform thickness has to be observed.

Measurements with air as test fluid are possible if the air volume flow is measured as well. In this way, long settling times and wetting of the fabric could be avoided, and the measuring procedure would therefore be considerably faster. Additionally, wetting of the fabric is not required. As only very low fluid pressures are required when measuring with airflow, the mechanical requirements of a stiff cell and the closing forces could be minimized. Using silicon oil as a test fluid results in similar Reynolds numbers such as they occur in real injections with resin. This is an advantage. Also, it is simpler to measure the higher pressures with accuracy. Additionally, the compaction behavior of dry preform differs from wet testing due to the missing lubrication effect of the test fluid.
Chapter 7

Conclusions and Outlook

This thesis is focused on numerical tools to improve the robustness of resin transfer molding processes. Although concepts and results for different topics presented in several chapters, the main idea developed in this work is the utilization of a parametric simulation model which allows to include stochastic process variations.

In the field of simulation, contributions like the detection and tracking of gas entrapments, as well as flow markers give valuable information in an early stage of the process designer. The parametric simulation model which allows to include process variations may be considered as a substantial contribution to improved simulations. Resulting process variations can be quantified and problem areas can be identified.

Further investigations will be necessary, to develop an improved simulation model for more precise prediction of voids. The heterogeneity of the fiber material will have to be considered on a macroscopic level.

Permeability disturbances of the parametric model need to be calibrated using statistical data, which should be determined under industrial manufacturing conditions. Eventually, correlations between variations should be introduced into the concept. Finally, the verification of the estimated process variations in serial production is desirable.

The presented process optimization with numerical methods benefits from the estimation of process robustness. Inappropriate processes as they are often suggested by numerical optimizations can be avoided when process reliability is considered (compare the optimization results in section 5.7.4). Additional gates and vents can be placed by the optimization at predefined locations as well as at free selected locations. A general definition of the optimization target has been developed, which needs only
very slight adaptations to fit for very different optimization problems. Thus, the effort to setup a numerical process optimization could be reduced to an absolute minimum.

Iterative and particularly stochastic optimization methods require a lot of computational resources. Therefore, efficient multilevel approaches and parallel computation have to be combined with the methods presented in this thesis.

For the investigated permeability estimation method, encouraging results have been found. A reasonable number of sensors appears to be sufficient the estimation, and the influence of the number of sensors on the estimation appears to be low. Particle Swarm Optimization achieved good estimations of the true permeability distribution. The topology of the residual function has been found to be smooth but multi-modal, so beside Particle Swarm Optimization stochastic and multi-start optimization methods may lead to successful estimations. Except for the continuous permeability measurement, transformation from the virtual environment to real experiments remains an open task.

A successful application of discussed permeability estimation is the presented method for continuous permeability measurement. Measurements of the permeability are significantly accelerated, restrictions of the conventional methods are removed, the required amount of material is reduced, and the saturated permeability can be determined. Although the mechanical realization has to be improved, excellent results have been obtained with this principle and the derived methods.

Applying simulations and process optimizations in the design of new processing parameters should become a standard approach, as it has become in other fields of mechanical engineering, like for example in structural design and sheet metal forming. Including stochastic variations appears to be an important step towards this goal. The determination of preform permeabilities needs therefore to become a standard procedure as well.
Danksagung

An dieser Stelle möchte ich mich bei all denen bedanken, die mich in den letzten Jahren unterstützt haben.
Allen voran Sarah, die mir in jeder Lage und bei allen Schwierigkeiten durch ping-pong, Wellen glätten, Drachensteigen und auf viele andere Arten weitergeholfen hat.
Prof. Paolo Ermanni für die Finanzierung dieser Arbeit und die Möglichkeit am Zentrum für Strukturtechnologien diese Arbeit durchführen zu können.
Prof. Francois Trochu from Ecole Polytechnique Montréal for being my co-examiner and talking to me in german.
Meinen Eltern für die Erklärung, warum $1 \cdot 1 = 1$ ist und nicht mehr als $1+1$, und dafür dass man immer Hilfe bekommen kann.
Fred und Regine Brückner für Wein, Weib und Gesang. Vielen Dank für die Gastfreundschaft!
Gion Barandun für die Psychohygiene, Balkonabende, und Rettung aus grosser Gefahr.
Hans Peter Eigenmann für die unermüdliche Unterstützung im Labor.
Dr. Gerald Kress, der immer mundgerecht erklärt hat, was man schon längst hätte wissen müssen. Und in fröhlicher Stimmung von neuen Missgeschicken berichtet hat.
David Keller für die Unterstützung beim Programmieren und bei der Optimierung.
Anna Brückner für die wirkungsvollen Knallerbsen “für gegen alles, was die Promotion torpedoirt”

Den Studenten, die Neues für mich herausgefunden haben: Stefan Klauser, Marcus Arnold, Roman Erne, Rolf Renggli

Und allen Anderen, die die vergangenen Jahren schöner gemacht haben: Vielen Dank!
Danke
Appendix A

Flow marker details

A.1 Detailed description of marker movement

A more detailed description of the developed moving algorithm presented in the previous section 3.4.1 is given here. Detailed interface objects and detailed functions that are required to move markers are introduced.

**Marker move procedure** Algorithm 2 controls the movement of a marker.

A marker can move within an element, which will be the most common case. When a marker reaches an element boundary it must be forwarded to a neighboring element if available. In most cases there will be no problem because a single neighbor is assigned at the element’s boundary. If

<table>
<thead>
<tr>
<th>Algorithm 2 marker move procedure (“slip::Marker::Move()”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. move in element, return if flight time is used completely</td>
</tr>
<tr>
<td>2. try volume move, return if succeeded</td>
</tr>
<tr>
<td>3. move marker on interface, return if flight time is used up completely</td>
</tr>
<tr>
<td>4. try volume move, return if succeeded</td>
</tr>
<tr>
<td>5. move marker on interface, return if flight time is used up completely</td>
</tr>
<tr>
<td>6. try volume move</td>
</tr>
</tbody>
</table>
there are several possible elements to which the marker may be forwarded, a stochastic forwarding algorithm of the `TryVolumeMove()` method (see algorithm 3) at a branching point will forward the marker to an element where it can move on. This function returns true if the marker could be moved inside a neighbor element and updates all marker fields like the marker location, the host element pointer and the remaining flight time. In case of a successful move of the marker inside a neighbor element, the `Move()` method is called recursively and the moving scheme starts again.

In case of an unsuccessful move inside a connected element, the algorithm tries to move the marker on the current element surface. This becomes necessary if a marker has reached an element border and no neighbor is available or if the time step length in all neighbor element is calculated as zero. If the marker is on an element surface it is moved in the surface plane until it reaches an edge. If it is on an edge the marker is moved until a node is reached. The algorithm tries to move the marker inside any of the elements connected to this node by calling `TryVolumeMove()`. If still no move inside a connected element is possible, it is tried to move the marker on an element connector of any element connected to the node. The element connector with the lowest angle between the velocity vector and the edge vector or element surface is chosen, provided that a time step \( \Delta t > 0.0 \) is possible on this connector.

Repetition of steps 2 and 3 in algorithm 2 is required for 3D simulations, only. In step 3 the marker can move on the surface between two three dimensional elements, and reaches an edge at their boundary. If it cannot be moved inside a neighbor element in step 4, it is tried in step 5 to move the marker on the edge reached. Step 6 is a final try to transfer
the marker to a neighbor element. If no appropriate element could be found the move method returns and the time budget is discarded.

**Intersection of marker and edge** The intersection point of a marker and an edge can be calculated using 3D vector geometry. In figure A.1 the calculation is derived considering the three vectors, velocity, edge vector and element normal vector. The current marker location B and the node location A are introduced. Two descriptions for the intersection point S can be found, leading to (A.1)

\[
\vec{A} + k_{edge} \cdot \text{EdgeVec} + k_{normal} \cdot \vec{n} = \vec{B} + \Delta t \cdot \vec{V}
\]  

(A.1)

which is equivalent to

\[
\begin{bmatrix}
\vec{V} \\
\text{EdgeVec} \\
\vec{n}
\end{bmatrix} \cdot 
\begin{bmatrix}
-\Delta t \\
k_{edge} \\
k_{normal}
\end{bmatrix} = \begin{bmatrix}
\vec{B} - \vec{A}
\end{bmatrix}
\]  

(A.2)

The values for \(k_{normal}\) should usually be around zero. An intersection within the edge area will result in \(0 \leq k_{edge} \leq 1\). Values outside that range indicate intersection points outside the edge area. Problems arising from parallel edge and flow velocity are treated in the solution algorithm of the equation system.

**Intersection of marker and surface** Analog to the intersection of marker and edge, an equation system can be build who’s solution gives the time \(\Delta t\) for the intersection of flow marker and surface. As illustrated in figure A.2 the intersection point is given by

\[
\vec{C} + \vec{P} + k_a \cdot \vec{a} + k_b \cdot \vec{b} = \vec{B} + \Delta t \cdot \vec{V}
\]  

(A.3)
This equation is equivalent to

\[
\begin{bmatrix}
\vec{V} \\
\vec{a} \\
\vec{b}
\end{bmatrix}
\cdot
\begin{bmatrix}
-\Delta t/k_a \\
k_a/k_b
\end{bmatrix}
= (\vec{B} - \vec{CP})
\]  

(A.4)

where \(k_a\) and \(k_b\) give the ratio of the vectors \(\vec{a}\) and \(\vec{b}\) that has to be added to \(\vec{CP}\) to reach the intersection point \(S\). Values for \(k_a < 0\) or \(k_b < 0\) indicate intersection locations outside the element’s surface area, as well as values resulting in \(k_a + k_b > 1\). The intersection is on the element’s surface for all other value combinations. Again, problems arising from parallel edge and flow velocity are treated in the solution algorithm of the equation system.
A.1 Detailed description of marker movement

Figure A.4: Marker forced back onto the element plane

**Moving on an edge**  A marker can be moved on an edge by calculating the projected flow velocity $\vec{V}_{proj} = \frac{\vec{v} \cdot \vec{E}}{\|\vec{E}\|}$ (figure A.3). Because the edge vector $\vec{E}$ points from node $C_1$ to $C_2$, the flight time to nodes $C_1$ and $C_2$ can be calculated as

$$
 t_1 = \frac{(\vec{C}_1 - \vec{Loc}) \cdot \vec{E}}{\vec{V} \cdot \vec{E}} 
$$

(A.5)

$$
 t_2 = \frac{(\vec{C}_2 - \vec{Loc}) \cdot \vec{E}}{\vec{V} \cdot \vec{E}} 
$$

(A.6)

One time is negative and one time is positive, the node with the positive flight time is reached next.

**Moving on a surface**  The movement of a marker on a surface is implemented in two steps. In the first step the marker moves with the elements velocity to a point A (figure A.4). If this point lies outside the surface plane, it has to be forced back onto the surface, point B.

Two conditions describe the point B on the element surface:

$$
 (\vec{B} - \vec{CP}) \cdot \vec{n} = 0 \land \vec{B} = \vec{A} + k \cdot \vec{n} 
$$

(A.7)

so a $3 \times 3$ equation system has to be solved in order to get the final location of B

$$
 (\vec{A} + k \cdot \vec{n} - \vec{CP}) \cdot \vec{n} = 0 \iff k \cdot (\vec{n} \cdot \vec{n}) = (\vec{CP} - \vec{A}) \cdot \vec{n} 
$$
\[ \vec{B} = \vec{A} + \left( \frac{(\vec{C} - \vec{A}) \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \right) \cdot \vec{n} \]  
(A.8)

**Numerical problems**  Numerical problems arise from discretization error of the representation of floating point numbers in digital computer systems. If an intersection point is calculated, the location is slightly scattered. So, it is not clear whether the intersection point is on an element connector or slightly beside it. In some cases this may lead to difficulties in the move algorithm because it is required to know whether a maker could be moved or not.

But checking for movement is only required if a marker is already located at an element connector. So a possible solution is to test whether the interface to which the marker is assigned has changed.

### A.2 Generation and storage of neighborship information

**Element Connector**  For storing neighborhood information two interface types (Edge and ElementSurface) are defined, derived from the virtual and abstract base class `ElementConnector`:

```cpp
class ElementConnector
{
public:
    virtual ~ElementConnector() {}

    virtual double Intersect( const DenseVector & P,
                              const DenseVector & V,
                              const DenseVector & ElementNormal,
                              double & dt ) = 0;

    virtual bool MoveOnInterface( double& dt,
                                  DenseVector& Loc,
                                  const DenseVector& V,
                                  vector<Element*>& NeighbourElements,
                                  ElementConnector*& EleCon ) = 0;

    void QRLinearSystem3x3( const DenseVector V1,
                            const DenseVector V2,
                            const DenseVector V3,
                            DenseVector & X );

    const static double Tolerance;

    std::vector< Element* > ConnectedElements;
};
```
The method **Intersect()** provides an interface to a function calculating the intersection point of the marker at location $P$ flowing with the velocity $V$. Additional information about the surface normal can be given by **ElementNormal** as it is required for edge intersection (A.2). The return value gives the position of the intersection point relative to the connector area. Values $> 0.0$ indicate locations on the connector, a value of $0.0$ indicates a location on the edge of the connector, and values $< 0.0$ indicate locations outside the connector. $dt$ gives the flight time of the marker to the intersecting point.

**MoveOnInterface()** moves the marker location with the velocity $V$ by a maximum time step of $dt$ and $dt$ is updated. The Pointer **EleCon** is updated in case a different **ElementConnector** is reached, otherwise it is set to **NULL** (pointer to nowhere). The vector **NeighbourElements** contains all elements that are connected to the final interface that has been reached.

**QRLinearSystem3x3()** is a specialized fast solver for equation systems of three vectors forming a linear combination.

Each **ElementConnector** provides a list of connected elements in **ConnectedElements**. **Tolerance** is a constant for comparison purposes, clearing numerical calculation errors as they occur in intersection calculation (A.1).

**Edge**  Shell elements are bounded by edges as described in subsection A.2. An **Edge** is defined by the straight line between the two nodes **Node1** and **Node2**. The geometrical information is stored in **EdgeVec**, a vector pointing from **Node1** to **Node2**.

```cpp
class Edge : public ElementConnector

Edge( Node* N1, Node* N2 );

double Intersect( const DenseVector & P,
                 const DenseVector & V,
                 const DenseVector & ElementNormal,
                 double & dt );

bool MoveOnInterface( double& dt,
                      DenseVector& Loc,
                      const DenseVector& V,
                      vector<Element*>& NeighbourElements,
                      ElementConnector*& EleCon );

Node* Node1;
Node* Node2;

DenseVector EdgeVec;
```
**ElementSurface**  Three dimensional elements like tetrahedral linear volume elements are bounded by surfaces of the type `ElementSurface`.

```cpp
class ElementSurface : public ElementConnector
{
    ElementSurface();
    void DetSurfaceDefinition();
    double Intersect( const DenseVector & P,
                      const DenseVector & V,
                      const DenseVector & ElementNormal,
                      double & dt );
    bool MoveOnInterface( double& dt,
                          DenseVector& Loc,
                          const DenseVector& V,
                          vector<Element*>& NeighbourElements,
                          ElementConnector**& EleCon );
    vector<Edge*> Edges;
    DenseVector CentralPoint;
    DenseVector BorderVecA, BorderVecB;
    DenseVector Normal;
};
```

Each `ElementSurface` has an element `Normal`, and three bounding `Edges`. The function `DetSurfaceDefinition()` reads the node information from those three edges and selects the location of one node as `CentralPoint` and defines the two vectors `BorderVecA` and `BorderVecB` which correspond to the two edges connected to `CentralPoint`.

So, two lists are build up

- `vector<ElementSurface*> Surfaces`
- `vector<Edge*> Edges`

storing all `ElementSurface` and `Edge` instances.

**Generation of edges** For linear triangular shell and volume elements a simple algorithm can be applied to create the edge list. As each node of an element is connected to every other node of this element, all combinations of node pairs must be pursued. This can easily be achieved by two nested for loops.

Before an new edge is created, it is necessary to check whether a node between the same pair of nodes already exists. In that case no new edge must be created, but the existing has to be updated (⇒ list of connected elements).
A.2 Generation and storage of neighborship information

<table>
<thead>
<tr>
<th>Algorithm 4</th>
<th>creation of element surfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;slip::Darcy3D4::FindNeighbours()&quot;</td>
<td></td>
</tr>
</tbody>
</table>

1. loop through all 4 element surfaces → i
   (a) if surface i already exist, go to 1f
   (b) pick three element nodes, without node i
   (c) find neighbor element by comparing element nodes and surface nodes
   (d) create new surface
   (e) inform neighbor element about neighborship
   (f) next

When an edge is created by an element, the element is added to the connection list of the edge and the edge is added to the elements edge list. Thus, both structures are informed about the connectivity.

**Generation of element surfaces** In sLIP, element surfaces are only required for *Darcy3D4* linear tetrahedral volume elements, because no other volume elements are available. While it is difficult to number element surfaces in general, for all element types available in sLIP a simple solution for numbering can be found. Each surface has three nodes, the fourth is outside the element surface. So, the number of the out-of-surface node gives the number of the corresponding surface.

While surface i is not yet assigned, it is created and appended to the list of *Surfaces*. To create a surface the bounding edges must be found first and than given to the new *ElementSurface* object. Additionally, the neighboring element has to be found if there is any. This is done by searching the other elements for an element that is connected to the same three nodes, too. If a neighbor element has been found, it is informed about this neighborship relation to avoid double creation of the element surface and to reduce the simulation time. The neighbor element finds the appropriate surface number by comparing the surface nodes and the element nodes, see algorithm 4

**Linear triangular shell elements** The element surface is bounded by three edges. An Edge is a structure between two nodes and contains
information about connected elements (see A.2). Additionally, it has a
field to store its normal vector.

To each edge of an element a different number of elements may be
connected:

- 0 at the edge of a part
- 1, this is the normal case
- 2 or more at joints of shell structures

**Volume Elements** Volume elements are bounded by surfaces, and sur-
faces are bounded by edges in turn. In contrast to edges, only two elements
can be connected to an element surface in maximum. It may have only
one connected element if the element surface structure is at the edge of
the cavity.
Appendix B

Resin model

B.1 Experimental campaign

Suggestions for an experimental campaign to measure the data to determine all required model parameters are summarized in table B.1.

Experimental campaign to determine the curing model parameters Differential Scanning Calorimetry (DSC) measurements are used to determine the curing behavior. Both, dynamic and isothermal DSC measurements are suggested in the experimental program. A least squares fit is used to determine the model parameters (table 3.1) describing the curing behavior. The present degree of cure is calculated as the released exothermal energy scaled by the total exothermal energy for complete cure.

\[
\alpha(t) = \frac{E_{\text{exo}}(t)}{E_{\text{exo, total}}} = \frac{1}{E_{\text{exo, total}}} \int_{t=0}^{t} P_{\text{exo}}(\dot{t}) dt
\]  

(B.1)

Accordingly, the curing rate is defined as

\[
\frac{d\alpha(t)}{dt} = \frac{P_{\text{exo}}(t)}{E_{\text{exo, total}}}
\]  

(B.2)

Dynamic DSC measurements at slow heating rates are used to determine the total exothermal energy that is released for complete curing (figure B.1). Problems may arise if the fast heating rates are too high. Slowly reacting resin systems may not be completely cured before the thermal degradation begins (figure B.2). Then the total exothermal energy cannot be determined properly. Therefore, a slow rate must be selected to
<table>
<thead>
<tr>
<th>Experiment</th>
<th>Objective</th>
<th>Parameter hints</th>
</tr>
</thead>
<tbody>
<tr>
<td>dynamic low temperature DSC (slow)</td>
<td>determination of $T_{\text{solid}}$</td>
<td>around solidification temp</td>
</tr>
<tr>
<td>dynamic DSC (slow)</td>
<td>total exothermal energy for full cure</td>
<td>—</td>
</tr>
<tr>
<td>dynamic DSC</td>
<td>$\frac{dT}{dt}$ as function of $T$ and $\alpha$</td>
<td>various heat rates $\frac{dT}{dt}$</td>
</tr>
<tr>
<td>isothermal DSC (t&lt;sub&gt;low&lt;/sub&gt;...t&lt;sub&gt;up&lt;/sub&gt;)</td>
<td>$\frac{d\alpha}{dt}$ as function of $T$ and $\alpha$</td>
<td>$T_{\text{injection min}}$...$T_{\text{curing}}$</td>
</tr>
<tr>
<td>isothermal rheometry</td>
<td>viscosity $\eta$ as function of $T$ and $\alpha$</td>
<td>$T_{\text{injection min}}$...$T_{\text{curing}}$</td>
</tr>
</tbody>
</table>
be able to separate the exothermal peaks resulting of resin cross-linking from degradation at high temperatures clearly. This measurement should be repeated to get reliable data for the total exothermal energy, because all further DSC measurements to determine the degree of cure (eq. B.1) and the curing rate (eq. B.2) will be related to it.

Measuring and processing isothermal DSC data is difficult as the exothermal energy that is released before the target temperature is reached cannot be evaluated. Therefore, the heating rate should be as high as possible to minimize this error. A temperature overshoot over the target temperature must be avoided, because even a short time at an increased temperature causes a lot undefined curing. This influence is worse than the error caused by a longer heating period.

Moreover isothermal DSC is very sensitive to a base line drift. To compensate drifting problems a subsequent dynamic DSC should be run for each sample, thus the remaining exothermal energy for complete cure can be measured and the degree of cure at the end of the isothermal measurement can be calculated. With the information of the degree of cure at the end of the isothermal measurement the base line drift can be corrected.

For reliable experimental determination of the model parameters, curing data should be measured in the relevant processing window (fig-
Figure B.3: Area in $T(\alpha)$ diagram covered by data points

ure B.3). Obviously, for slow curing resins such as RTM6, there are difficulties to cover certain process relevant areas. Then, measuring at the same conditions as they occur in the process becomes impossible due to the costs of long measurement times and technical problems including drift of the DSC device. Therefore, an extrapolation of curing data measured at higher temperatures becomes necessary.

**Experimental campaign to determine the viscosity model parameters** Isothermal viscosity measurements within the processing window (figure B.3) are suggested. In order to obtain the influence of curing on the viscosity measurements at increased temperatures up to the curing temperature might be necessary for slow curing resin systems. The influence of the shear rate on the measured viscosity and the reproducibility of viscosity measurements should be checked in additional tests.

**B.2 Evaluation of experimental data**

**Pre-processing of DSC data** DSC data needs to be processed before it can be used to determine the required model parameters. At first, all the data have to be shifted to compensate the base line offset due to base line drifting and heating power for dynamic DSC. The power values have to be converted into mass specific powers. The exothermic curing energy for total curing can be obtained as integral of the exothermic peak. In figure B.4 a dynamic DSC measurement is shown for RTM6. The exothermal
B.2 Evaluation of experimental data

Figure B.4: Specific exothermal power of RTM6 resin determined from DSC data at a heating rate of \(2 \frac{K}{min}\). The exothermal peak interferes with degradation peak starting at approx. 6500 s

power has been integrated from 2000 to 6500 s resulting in an exothermal energy for complete curing of \(E_{\text{exo,total}} = 507 \frac{J}{g}\). Figure B.5 shows the determined curing rate and degree of cure according to equations B.2 and B.1.

Isothermal DSC can be processed analogously, however the base line drift can only be corrected according to DSC values from the end of the measurement. When the maximum curing degree for the given temperature is reached, the exothermal power must reach zero. Figure B.6 shows a raw data plot of an isothermal DSC measurement with a slight deviation of the base line from zero. For further processing all data has to be shifted by the asymptotically reached offset value. Figure B.7 shows the evaluated degree of cure and the curing rate as a function of time.

Evaluation of curing parameters The pre-processed DSC data of RTM6 of curing degree and curing rate has been used to fit the kinetic model parameters. Eq. 3.5 has been used with the software MicroCal Origin for this evaluation. Recommendations for starting parameters for the least squares fit are given in [46].

Figure B.8 shows the resulting fit of the modeled values to the measured values. Figure B.9 shows all the data that has been used in a temperature vs. degree of cure diagram. All curing degrees from 0 to 1 are covered, but the temperature is above injection temperature. So, a slight extrapolation on the temperature scale has to be done when curing values for injection processes are calculated.
Figure B.5: Curing degree and curing rate during the dynamic DSC measurement shown in figure B.4

Figure B.6: Plot of isothermal DSC data for RTM6 resin at target temperature of 180°C

Figure B.7: Curing degree and curing rate during isothermal DSC measurement shown in figure B.6
B.2 Evaluation of experimental data

Figure B.8: Comparison of measured and modeled data.

Figure B.9: Temperature vs. degree of cure diagram showing the area that is covered by DSC curing data. The very most curing data have been measured outside the actual processing window.
Figure B.10: Obtained maximum degrees of cure for RTM6, depending on curing temperature and phenoxy content

The temperature dependent maximum degree of cure is shown in figure B.10. Increasing values could be found for higher temperatures.

**Evaluation of viscosity parameters**  A major problem in parameter fitting is the definition of an appropriate target. For this, the difference between measured and calculated viscosity could be used. But it will emphasis measured data points with a high viscosity compared to data points of low viscosity which is inconvenient. Better approximation results are achieved in this case by minimizing relative errors

\[
F = \sum \left\| \frac{\eta_{model} - \eta_{meas}}{\eta_{meas}} \right\| \Rightarrow \min \tag{B.3}
\]

Again, recommendations for starting parameters are given in [46]. The resulting approximation is shown in figure B.11, where all viscosity data are plotted over the degree of cure.

**Summary of determined model parameters**  All the model data of RTM6 resin is summarized in table B.2 for the curing model and in table B.3 for the rheological model. The given \(\alpha_0\) values depend strongly on the resin treatment, meaning storing conditions and in case of 2K-systems the time since mixing.
Figure B.11: Comparison of measured and modeled viscosities (logarithmic viscosity axis). Viscosities are plotted for phenoxy concentrations of 0..10%.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Determined Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>2.38311</td>
</tr>
<tr>
<td>$T_{\text{solid}}$</td>
<td>-20</td>
</tr>
<tr>
<td>$T_{\text{kin}}$</td>
<td>1327.11408</td>
</tr>
<tr>
<td>$m$</td>
<td>1.24279</td>
</tr>
<tr>
<td>$n$</td>
<td>0.86403</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.01693</td>
</tr>
<tr>
<td>$t_{\text{cure}}$</td>
<td>1.11279</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>0.0010 @ $c_{\text{Phenoxy}} = 0%$</td>
</tr>
<tr>
<td></td>
<td>0.0015 @ $c_{\text{Phenoxy}} = 5%$</td>
</tr>
<tr>
<td></td>
<td>0.0020 @ $c_{\text{Phenoxy}} = 10%$</td>
</tr>
</tbody>
</table>
Table B.3: RTM6 viscosity model data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Determined Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>462.89317</td>
</tr>
<tr>
<td>$\eta_\infty$</td>
<td>0.0005343853</td>
</tr>
<tr>
<td>$T_{rheo}$</td>
<td>562.51069</td>
</tr>
<tr>
<td>$f_{\text{stretch}}$</td>
<td>3.11129</td>
</tr>
<tr>
<td>$l_{\text{Rheo}}$</td>
<td>54.16094</td>
</tr>
</tbody>
</table>

Figure B.12: Comparison of measured and modeled curing degrees of RTM6, plotted over time

**B.3 Model accuracy for curing kinetics and rheology**

**Accuracy of the curing model**  Figures B.12, B.13 and B.14 summarize the predicted degree of cure as a function of time. The modeled data fit the measured data rather well, even though each curve shown here which is for a given temperature and phenoxy content, results from a single measurement. So, the generally high scattering of DSC measurements causes errors such as curing degrees slightly above 100% and higher final curing rates at lower temperatures than at higher temperatures. A shift on the time scale may also result from varying curing degrees at the beginning of the DSC measurements.
Figure B.13: Comparison of measured and modeled curing degrees of RTM6 + 5% Phenoxy

Figure B.14: Comparison of measured and modeled curing degrees of RTM6 + 10% Phenoxy
Accuracy of the rheological model  Figures B.15, B.16 and B.17 show measured and modeled viscosities. Although a good agreement of measured and predicted values is obtained, some problems in measuring the resin viscosity become obvious. It is difficult to set proper starting conditions for DSC measurements, when reactive samples are investigated. So, obviously, the resin viscosity decreases during the beginning of the measurement. This time dependent behavior probably results from heating until the resin reaches its target temperature. In figure B.15 this can be seen for the 80°C test, which reaches a stable viscosity after approx. 500 s. At higher temperatures the curing reaction starts and influences the resin viscosity. In the figures B.16 and B.17 the influence of solving phenoxy on the viscosity can be seen, which increases the viscosity.

A further aspect that has to be considered is the reproducibility of viscosity measurements. Several uncertainties like variations in temperature or phenoxy concentration, curing degree at the start of the test etc. 7 viscosity measurements under the same conditions are shown in figure B.18. Obviously, high variations up to a factor of ≈ 2 have to be accepted.

A linear dependency of the viscosity over the phenoxy concentration is given by the definition of the Phenoxy term in eq. 3.6. The model parameter $l_{\text{cure}}$, scaling the influence of the phenoxy concentration, has been determined as described above by a least fit, giving the slope of the resin’s viscosity with increasing phenoxy content.
Figure B.16: Comparison of modeled and experimental viscosity data for RTM6 + 5% Phenoxy.

Figure B.17: Comparison of modeled and experimental viscosity data for RTM6 + 10% Phenoxy.
Figure B.18: Reproducibility test of viscosity measurement with RTM6 at 120°C and a Phenoxy concentration of $c_{\text{Phen}} = 10\%$ over 7 runs.

Figure B.19: Influence of Phenoxy content on the viscosity of RTM6.
B.4 Exploration algorithm for resin marker assignment

Starting at each marker, the marker is assigned to its host element. All new assigned elements are listed in a private list of the marker. In the next recursive step, the neighbor elements of the elements listed in the previous step are assigned to the marker influence area. The new found elements are listed again. This assignment is done for all markers until the next exploration step starts (see algorithm 5).
Appendix C

Realization of local connector remeshing

Basic ideas of the developed remeshing algorithm are listed here:

- define sphere for mold connector at given location and with given diameter
- delete elements and nodes inside the injection sphere
- cut elements ranging over the connector edge, create new elements if appropriate
- create inner row of elements defined size between inner diameter $d_i$ and outer diameter $d_o$
- connect all inner nodes to the center node by tubes
- update neighborship information

Tolerances for node and element generation Some critical cases result from the suggested realization

- avoid duplicated nodes
- avoid very small elements and elements without volume
- connect central nodes if two connectors have to be created within a radius of $2 \cdot r_{nominal}$ where $r_{nominal}$ is the connector radius
• duplicated center nodes should get remeshed only once. They must not get connect by a tube (see previous item), because duplicated remeshing has no useful effect. The tube length would become zero, and therefore the calculation of its element matrix would fail.

If nodes are located very close to the cutting sphere, short sections of element edges may be cut. This leads to very small elements or element with at least one very short edge. Those elements may cause numerical problems because values of their conductivity matrix values can differ in order of magnitudes form the average sized elements. The resulting global conductivity matrix may have a poor condition number resulting in inaccurate solutions or, in the worst case, singularities.

To account for these problems, tolerance zones around nodes are used. Thus, very small elements can be avoided. A tolerance factor $\text{REMESH\_TOL}$ is introduced. Within a radius of $r_{\text{nominal}} \cdot \text{REMESH\_TOL}$ no new nodes may be created. In this cases the existing nodes have to be used, accepting small deviations ($< r_{\text{nominal}} \cdot \text{REMESH\_TOL}$) in the resulting connector diameter.

Additionally, nodes with a distance

$$(1 - \text{REMESH\_TOL}) \cdot r_{\text{nominal}} < r_{\text{node}} < (1 + \text{REMESH\_TOL}) \cdot r_{\text{nominal}}$$

from center are assumed to be on a “tolerance stripe”. Depending on the location of the other element nodes (whether they are inside or outside the connector sphere), they are considered as outside or inside nodes.

**Generation of suitable elements** Elements that range over the edge of the cutting sphere have to be divided or reshaped. This is described in detail for a linear triangular shell element, only. For all other elements it works analogously. Two basic cases for this element type are

1. one node lies outside the sphere and two are inside (figure C.1A)

2. or one node lies inside the sphere and two are outside (figure C.1B)

Because tolerances are used within which no new node can be generated, cutting an element near a node may return the same node twice. This case has to be specially treated to avoid elements without volume.
Figure C.1: Summary of possible situations when cutting a linear triangle element
Bibliography


## Curriculum Vitae

<table>
<thead>
<tr>
<th>Name:</th>
<th>Rene Arbter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of Birth:</td>
<td>July 11, 1977</td>
</tr>
<tr>
<td>Address:</td>
<td>Hirschgartnerweg 34, CH-8057 Zürich</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>10/2004 – 8/2008</td>
<td>Research assistant and PhD student</td>
</tr>
<tr>
<td>University:</td>
<td>ETH Zurich, Switzerland</td>
</tr>
<tr>
<td>Institute:</td>
<td>Centre of Structure Technologies</td>
</tr>
<tr>
<td>Field of work:</td>
<td>Resin Transfer Molding</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Studies:</td>
<td>Mechanical Engineering</td>
</tr>
<tr>
<td>Institute:</td>
<td>RWTH Aachen</td>
</tr>
<tr>
<td>Specialisation:</td>
<td>Aerospace engineering</td>
</tr>
<tr>
<td>Diploma thesis:</td>
<td>Untersuchung der Übertragbarkeit des</td>
</tr>
<tr>
<td></td>
<td>Puck’schen Festigkeitskriteriums auf</td>
</tr>
<tr>
<td></td>
<td>Faserverbundkunststoffe mit thermoplastischen</td>
</tr>
<tr>
<td></td>
<td>Matrices</td>
</tr>
<tr>
<td>Semester thesis:</td>
<td>Entwicklung eines Schüttelkolbens zur diffusiven</td>
</tr>
<tr>
<td></td>
<td>Fed-Batch-Betriebsweise</td>
</tr>
<tr>
<td>Semester thesis:</td>
<td>Experimentelle Ermittlung des Schubmoduls im</td>
</tr>
<tr>
<td></td>
<td>Nachrissbildungsbereich</td>
</tr>
<tr>
<td>8/2000 – 7/2002</td>
<td>Student research assistant</td>
</tr>
<tr>
<td>Institute:</td>
<td>Institut für Kunststoffverarbeitung, RWTH</td>
</tr>
<tr>
<td></td>
<td>Aachen</td>
</tr>
<tr>
<td>Tasks:</td>
<td>Injection molding and fiber reinforced parts</td>
</tr>
<tr>
<td>School:</td>
<td>Ferdinand Steinbeis Schule, Reutlingen</td>
</tr>
</tbody>
</table>
